Life Patterns

Brian Clarkson and Alex Pentland

Abstract— In this thesis I develop and evaluate computational methods for extracting life's patterns from wearable sensor data. Life patterns are the reoccurring events in daily behavior, such as those induced by the regular cycle of night and day, weekdays and weekends, work and play, eating and sleeping. My hypothesis is that since a "raw, low-level" wearable sensor stream is intimately connected to the individual's life, it provides the means to directly match similar events, statistically model habitual behavior and highlight hidden structures in a corpus of recorded memories. I approach the problem of computationally modeling daily human experience as a task of statistical data mining similar to the earlier efforts of speech researchers searching for the building block that were believed to make up speech. First we find the atomic immutable events that mark the succession of our daily activities. These are like the "phonemes" of our lives, but don't necessarily take on their finite and discrete nature. Since our activities and behaviors operate at multiple time-scales from seconds to weeks, we look at how these events combine into sequences, and then sequences of sequences, and so on. These are the words, sentences and grammars of an individual's daily experience. I have collected 100 days of wearable sensor data from an individual's life. I show through quantitative experiments that clustering, classification, and prediction is feasible on a data set of this nature. I give methods and results for determining the similarity between memories recorded at different moments in time, which allow me to associate almost every moment of an individual's life to another similar moment. I present models that accurately and automatically classify the sensor data into location and activity. Finally, I show how to use the redundancies in an individual's life to predict his actions from his past behavior.

Index Terms—Keywords should be taken from the taxonomy (http://www.computer.org/mc/keywords/keywords.htm). Keywords should closely reflect the topic and should optimally characterize the paper. Use about four key words or phrases in alphabetical order, separated by commas.

1 Introduction

Imagine a device that can preserve our memories as we experience them and in the way we experience them. In order to be useful, the device must come with an environment to facilitate the remembering or browsing of stored experiences. A person's day-to-day activities are cyclical at some time-scales and follow slowly changing trends at other scales. The device's owner might have habits that structure a large part of his activities. This behavior should be readily portrayed and taken advantage of by the device, raising the basic question of how to provide a summarization to the casual browser. While this question has historically proven to be quite difficult in the fields of video and text summarization, we will argue that the very extended, intimate, and highly structured nature of the data that a prosthetic memory device is uniquely exposed to, makes it feasible to build statistical models of what events are commonplace and what events are rare.

The technology is available now to approximately capture and store the visual and auditory experiences of a person over a period of years and soon a lifetime. Since computational devices are gradually finding their way into more and more aspects of our daily lives, having these devices recognize and understand the events in our life is becoming important. A quick brainstorm will yield numerous uses for data of this type, from video diaries [13] to truly context-aware personal agents [45, 29]. However, just recording this data is not enough. It's not even enough for the simple task of re-experiencing or browsing one's stored experiences because of the sheer amount and variety of data involved. For these kinds of applications we at least need to be able to automatically search through the experiential data. For example, while browsing the user of an automatic diary comes across a kind of scene that he wishes to see more of. In this case it is necessary to be able to associate similar scenes to each other. Descriptive and predictive capabilities are necessary for agent-based applications that take actions based on the user's behavior. Knowing the habits of users and the difference between typical and atypical behavior are basic requirements for agents that work smoothly with humans. However, again, prediction is impossible unless we have a notion of the similarity between the scenes we are attempting to predict.

Quantitative analysis of someone's life can take place at many different time-scales. At each time-scale we expect to be able see some classes of phenomenon and not others (see Figure 0-1). Strapping sensors on an individual and sampling at the 1Hz time-scale will enable us to detect when someone is falling down the stairs. However, we will need to lift our view of the data to at least a daily time-scale if we want to predict when someone is going to fall down the stairs. In computational perception to date there has only been work on narrow, short time-scale domains. Long-term studies on individuals have been limited to the works of chronobiologists (researchers of long-term human physiology), psychologists, and clinical scientists. We now have the computational tools (storage space and

- *Brian Clarkson is with the Sony Computer Science Laboratory, Tokyo, Japan. E-mail: clarkson@csl.sony.co.jp.*
- *Alex Pentland is with the MIT Media Laboratory, Cambridge, MA, 02139. E-mail: sandy@media.mit.edu.*

Manuscript received (insert date of submission if desired). Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

computational power) to start considering modeling an individual's life at longer and longer time scales.

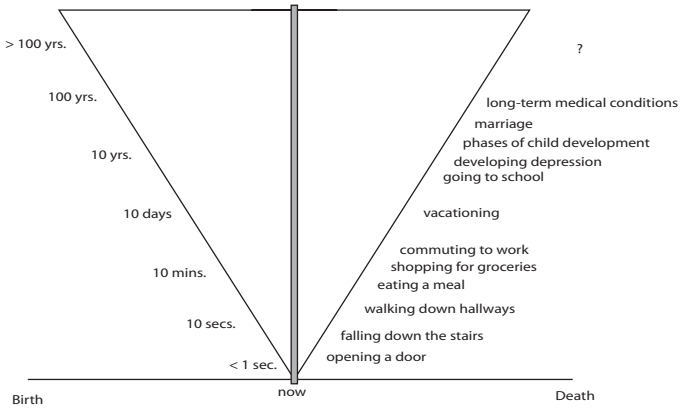

Figure 0-1: The different time-scales of life.

This work tackles the question of how to recognize and predict a person's day-to-day behavior from visual sensor data. Perhaps the hints that cognitive scientists are extracting about how we organize our own wet memories could provide clues on how to organize our machine's memories. Using ideas from episodic memory organization and insect-level perception, we develop an automatic framework for organizing sensor data that is intimately connected with an individual's daily activity. While this framework is designed and built with the application of a memory prosthesis (e.g. automatic diary) in mind, there is a direct application to context-aware agents and the frame problem in cognition.

Our first approach is to establish a similarity metric or method for assessing the similarity of pairs of time intervals in experiential sensor data. This similarity metric can then be used to group and align similar events together, structure the experiential sensor data into scene hierarchies, and classify situations. Our second approach is to statistically estimate the temporal models or statistical rules that describe the typical evolution of events in the experiential sensor data. These temporal models are a succinct description of the person's typical life pattern and can theoretically be used to identify anomalies or deviations from the norm that might signify novel events in the person's life. Since temporal models capture the habitual dynamics of the individual's life, they are also useful for prediction and summary.

2 Background & Motivation

Vannevar Bush has had his hand in many lines of academic thought, and ours is no exception. Even in 1945, he was imagining a wearable camera for the purposes of making the serendipitous record. Specifically he identifies three important properties that such a record needs to be useful:

- Continuous Recording
- Complete Storage
- Accessibility

Amazingly enough, Vannevar Bush was (correctly) unimpressed by the technical problems of taking the pictures, but instead points out that: "*The only fantastic thing about it is the idea of making as many pictures as would result from its use*." Of course he is referring to the development of the film, which he is assuming is still necessary, and the selection of which pictures will be lucky enough to receive attention for development. His observation underlines the necessity of indexing services for the growing store of images. Almost 50 years before Vannevar Bush wrote his prophetic article, inventors and tinkerers were already making wearable cameras in the form of scarves, walking sticks and pocket watches. In recent history, Steve Mann [31] has experimented with wearable cameras as a means of artistic expression (e.g. lookpaintings), online mediated reality, and as a means of personal record-taking with the same philosophy as Vannevar Bush's description above. However, there is a lack of experiments on what to do with the ever-increasing store of images obtained via a wearable camera.

Many have been inspired by Bush's description of the memex, and it is in fact considered by many to be the conceptual pre-cursor to the World Wide Web (attributed to another characteristic of the memex which is the set of links between objects that the memex contains). Many have interpreted Bush's memex concept as organizing the knowledge of humanity in general, but what if we interpret as organizing the memories of a single individual. In this case the memex becomes a kind of hyper-linked diary, much like the web logs, or blogs, that are turning into a recent WWW epidemic. However, no one has found a way to automatically include the real experiences, the visceral experiences, of the diary writer into the diary. Even more difficult is the creation of associating links amongst an individual's experiences. Let's take a look at how and why researchers have started tackling this and other related problems.

2.1 Multimedia Indexing

There is a large body of research on text classification and retrieval for organizing information that might be found in the textual components of a memex. However, this work tackles the analogous problem for perceptual sensor data recorded from an individual's life. How do we establish similarity between different sensor measurements or times of an individual's life? Undoubtedly, this similarity metric is task-specific. Thus, by virtue of the data being sight and sound, a closely related field is multimedia indexing where scientists and engineers are building systems that attempt to organize video and sound.

There has been a great deal of work in the last few decades concerning the problem of indexing databases of images and sound. The core problem in this field is to produce an appropriate similarity metric for comparing a given query example to objects in the database. Pentland et.al. [38] shows through an image sorting application, called PhotoBook, that in certain cases you can derive features from sets of related images (eigenfeatures) whose ordering in the Euclidean sense corresponds roughly to the way a

human would order a given set of images by similarity. Iyengar et. al. [22] extends this to video. Zhong et. al. [57] and later Lin et. al. [30] noticed that there is innate structure in a video's low-level characteristics that often corresponds to higher-level semantic structures of scenes. Zhong et. al. finds this innate structure in the low-level characteristics by clustering and heuristically grouping segments and shows that this relationship does exist in some cases. Independently, Saint-Arnaud [43], Foote et. al. [17] and [15] found that similar to image texture, you can define a concept of auditory texture and use it to classify and group audio clips based on similarity.

However, there is a key difference between the datasets of multimedia objects that the above researchers are considering and the database of daily experiences considered here. Day-to-day experiences are mostly routine or quasi-periodic. Theoretically the frequency of novel events is much lower than in TV newscasts or movies. Video surveillance researchers have noticed this about their data to great benefit. When you point a camera at a parking lot for long periods of time, with a little bit of domain knowledge you can easily cluster the usual from the unusual. [19] It seems our domain lies somewhere between movies and security video on the entropy scale.

How often novel events occur in someone's life is certainly different for each person, but ultimately the proportion of routine events to novel events is expected to be quite high. This translates into two important properties, redundancy and closure. Contrast this with a database of movies or images on the web. There is almost no limitation on the types of objects that could be present and hence a researcher using these databases can almost never assume that the queries will come from the same set of objects that are in the database. Nor will the apparent commonality of a pattern in the database necessarily have any relationship to the commonality of the pattern to the user.

2.2 Context-Aware Agents

How can person's day-to-day behavior be recognized or predicted by a computational agent? If we are going to build a personal agent (wearable or not) that anticipates its master's behavior we need to be able to build at least this basic level of understanding into the system. [29] Agents without these abilities can only act on explicit input thus limiting their usefulness to virtual environments such as the Internet. For software agents in wearable computers, PDA's, and cell-phones arguably most of the relevant context is contained in the physical world of the user and the user's environment. Hence, to say an agent in this situation is context-aware or situated means that it must have sensors into the user's physical world and an ability to learn the basic rules of the user's physical world.

Agents that recognize events in its master's surroundings and behavior can proactively react without explicit direction from the master, thus expanding their usefulness into new domains. Agents that don't anticipate can react and reconfigure based on the present 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, typical situations span the entire complex domain of real life where it is unreasonable to manually design such anticipatory behavior into an agent.

In the last 10 years there has been an explosion of efforts to bring context to computational agents. At Xerox, Lamming et. al. [28], used context in the form of location, encounters with others, workstation activity and telephone calls, as a way of keying information for recall. While some of the inputs to this system indirectly reflect the physical state of the user and his surroundings, they are limited to those physical activities that have a measurable effect on a system that is not designed to measure perceptual events (e.g. location corresponds to the user switching wireless hubs as he moves from room to room, typing at a workstation corresponds to a user activity, etc.). Complete multi-person systems (C-MAP [51] and The Conference Assistant [12]) using user location and history as the major context components, have also been built and tested for assisting participants at conferences, exhibitions, and other interaction- and information-rich events. The C-MAP system was unique in that one of its design goals was to also provide a useful record of the event and the user's actions during the event. Along similar lines is Brad Rhodes' Remembrance Agent [41] who uses limited context, text typed into a wearable prompt, to trigger just-in-time information. However, Rhodes was always the first to admit that in order to claim that an agent is truly context-aware that agent needs sensors into the real physical world.

The realization of the importance of sensing to context-awareness for computing applications has sparked intense interest in wearable sensors. Healey et. al. [21] constructed and experimented with a novel wearable sensor-driven agent called the StartleCam. It was a wearable camera integrated with a galvanic skin response (GSR) sensor who's measurements are generally considered to correspond to stress levels, especially when induced by a startle response. A wearable computer was programmed to monitor the GSR levels, detect a startle response, and respond by taking a picture via the worn camera. An alternate way of constructing this device that is more aligned with the ideas of this work, is to constantly record video and the GSR levels simultaneously. Later, the startle events detected in the GSR record can be used to highlight potentially interesting points in the video. Work by Starner et. al. [50], uses wearable cameras to extract information about the user's location (omni-directional camera) and task (camera oriented on the user's hands) as a user plays a mobile, multi-person game. Farringdon et. al. [14] uses sensors designed to monitor the user's motions (walking,

running) and posture (sitting, standing, lying) to determine user activity.

2.3 Memory Prosthesis

No one has yet been able to so completely record the experiences of one individual as to be able to go back to any moment, any second, of that person's life and invoke a remembrance of that moment. With such a recording, there are theoretically opportunities for understanding the structure of an individual's life for psychological, chronobiological, or personal agendas.

> What are the long- and short-term trends?
> What are the repeating or semi-repeating patterns?
> What part of your day is routine?
> What part of your day is novel?
> Are your current habits, for example, number of people you talk to per day, different from last year?
> During what periods are you the most active? Do they come in cycles?

In addition to these directed questions we can consider the use of this data in an environment that assists the user in effectively browsing his/her memories. A very compelling application is to use the structure extracted automatically by statistical analysis as scaffolding for contextualizing and compartmentalizing memorabilia that the user wishes to organize and inter-relate. This way the user is provided with an environment for browsing and exploring paths of memories along criterions other than time.

Lamming and Flynn [28], using the ParcTab [46] system, pioneered a portable episodic memory aid called the Forget-Me-Not system. They also noticed that the intimacy that a wearable or portable device has with its user enables it to consistently record certain aspects of its user's life. Since studies by [4] have confirmed that we group our memories into episodes, Lamming et. al. consider their device as an aid for recalling a particular memory episode, hence the name. However they do not attempt to organize the device's captured data into a similar episodic structure even though this could greatly assist the user in browsing the growing store of data.

2.4 The Frame Problem

As A.I. researchers built robots to perform increasingly complicated tasks at some point they found that even if they provide complete descriptions of the world and the rules that govern the robot's world there always remained the fundamental problem of choosing which pieces of this knowledge to consider when constructing a solution to a given problem. Unless the robot has some concept of relevancy, the exponential explosion of contingency plans and never-ending chains of induction will inevitably swamp it. Daniel Dennett [11] gives an excellent account of this illusive problem. Various researchers have since proposed mechanisms for alleviating this problem, but none are universally accepted as solutions yet, mainly due to the lack of convincing demonstrations on real world situations (consult [11] for listing on some of these approaches).

For example, in 1974, Minsky [33] published a memo titled "A Framework for Representing Knowledge". He outlined a structure, called a frame, that contained within it pointers to various pieces of knowledge that were expected to be useful in a given situation; not all pieces of information that could possibly be useful, only those expected to be useful. Thus, a frame also has a collection of constraints that need to be loosely satisfied in order for the frame to become "active" only in the correct situation. A frame specifies the expectations, predictions, or instructions about what should come result given that the frame's conditions are met.

Researchers in psychology [3] have also championed this idea of a frame (also referred to as schema) because of its apparent and compelling similarity to the episodic organization of human memory. While many competing theories are disagreeing on the details, the basic idea is that the processes associated with remembering perceptual events are intimately intertwined with the processes of concept formation and problem-solving. These frames are just collections of pointers to useful information (memories or even other frames) and can be seen as compartmentalizing or clustering an individual's concepts and memories, essentially for the dual-purpose of efficiency and generalization.

Researchers in computer vision and audition are familiar with the context problem since they routinely have to restrict their domains (i.e. *manually* specify a valid frame or set of valid frames) in order for their systems to work. For example, speech recognition has only been successful when the environment (car, office, wheelchair) and grammar (switchboard task, command-and-control, spontaneous conversation) are constrained. Face recognition benefits when we can constrain the face database and tracking benefits when the lighting conditions are known. All of these systems suffer from the *frame* problem because they need to know their current context in order to apply their context-sensitive algorithms.

2.5 Insect Perception

Rather than try to use perception techniques that are usually associated with high-level human-like intelligence such as speech recognition or face recognition, this work relies on insect-level perception. We can define what this means with an example from how insects, specifically Cataglyphis desert ants, are believed to navigate to previously visited locations. Studying how insects remember locations indicates what level-of-detail is necessary for recordings of environments during a matching task. It has been shown that a number of species of insects, from bees to ants, utilize landmark features in the surrounding scenery to navigate. [25] If landmarks are altered or moved, then the foraging insect will navigate as if their target location is in the new position implied by the altered landmarks. Lambrinos et. al. [27] has constructed robots that are based on a model of desert ant navigation. Desert ants seem to use landmark features recorded from various positions around the site to be able to find the site

again. This is very similar to the localization by panoramic views that researchers in robotic vision are developing [23].

3 DATA COLLECTION & METHODS

Our lives are not random. They certainly exhibit structure at all time-scales. How is this structure organized? What are its atomic elements? What is the network of dependencies connecting the past, present, and future moments? Limiting our analyses to an appropriate level-of-detail, enables us to reasonably tackle these questions. Since these questions need hard data to produce answers, we address how to collect measurements of an individual's experiences.

First-person, long-term sensor data is a guiding principle for our overall approach. It is inappropriate given the current state of the art to tackle the problem of how to give a machine human's level understanding of an individual's daily behavior without first granting it with an insect's level of understanding. Perhaps in certain cases, we can obtain near-human understanding by severely restricting the domain. However, in this work the completeness of the domain, that is an individual's day-to-day life, is a priority and hence we are guided to the more appropriate level of perception portrayed by insects. Similar to the representation-free approach of Rod Brooks, we avoid building complete models of the user's environment and instead rely on the redundancies in the raw sensor data to provide the structure. This philosophy implies the use of coarse level features and emphasizes robustness over detail (such as in the use of context-free methods over context-specific methods).

In this work we took a straightforward approach to addressing the issues of a similarity metric and temporal models of life patterns. We collected long-term sensor measurements of an individual's activity that enables the extraction of atomic elements of human behavior, and, the construction of classifiers and temporal models of an individual's day-to-day behavior.

3.1 The I Sensed Series: 100 days of experiences*

The first phase in statistically modeling life patterns is to accumulate measurements of events and situations experienced by one person over an extended period of time. The main requirement of learning predictive models from data is to have enough repeated trials of the experiment from which to estimate robust statistics. 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. 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).

We designed and followed a consistent protocol during the data collection phase. 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 orientation 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.

3.2 The Data Collection Wearable

The sensors chosen for this data set are meant to mimic insect 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 (vestibular), while taste and smell are not covered because the technology is not available yet. The left-right eye unit placement on insects differs from that front-back placement of the cameras in our system. However, they are qualitatively similar in terms of overall resolution and field-of-view. The properties of the 3 sensor modalities are as:

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 that 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 Logitech Quickcam 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 is housed inside the main compartment. Also in the main compartment are computer (PIII 400Mhz Cell Computer) with a 10GB hard drive (enough storage for 2 days) and batteries (operating time: ~10 hrs.). The polystyrene shell (see **Error! Reference**

source not found.) 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.

4 THE SIMILARITY MEASURE

Before we can answer any of the questions about classification, prediction or clustering, we first need to determine an appropriate distance metric with which to compare moments in the past. We will look at how to determine what are the appropriate intervals to be comparing and how to quantify their similarity. While doing so we present new methods for data-driven scene segmentation. We will then present methods for determining the similarity of pairs of moments that span time-scales from seconds to weeks. The tools we build up in this chapter provide the foundations for classification and prediction.

The Features

The first step in aligning sensor data is to decide on an appropriate distance metric on the sensor output. Possibly the simplest similarity measure on images is the L_1 norm on the vectorized image. Computer vision researchers typically avoid using such a simple metric because of its vulnerability to differences in camera position and orientation and opt instead for orientation-invariant representations, such as color histograms or image moments. However, as mentioned before there is clear evidence [54] that insects (and in many cases humans) store view-dependent representations of their surroundings for later recall and matching. In this case the dependency of the image and the camera position and orientation is an advantageous one. Throwing away the information that links an image to the state of the camera at the moment of capture doesn't make sense when the task is to situate the camera wearer.

Our distance metric between images in defined directly in terms of the pixels of the image:

$$D(x,y) = \sum_{i}^{W}\sum_{j}^{H}\sum_{c}^{3}\left|x_{ij}(c) - y_{ij}(c)\right|$$

$$x_{ij}(c) = \text{pixel's c-th channel value at (i,j) of image x}$$

Thus, it is directly influenced by the size, shape, color, and position of objects in view. Contrast this to color histogram-based metrics that are invariant to position and shape, but are sensitive to size and color. The L_1-norm on images is very good at discriminating different images, but probably one of the worst metrics for achieving any kind of generalization or robustness to noise. Our decision to use this metric for alignment rests on two observations.

First, given the size and coverage of our 100-day data set, finding a match for a particular image is literally like finding a "needle in a haystack". On the other hand, the larger our data set is, the closer the match will be. Thus the robust metrics, which aren't very discriminative, serve well when we are interested in finding matches that aren't very close (a requirement for sparse data sets). This comes at the cost of never being able to find that really close match. Of course, an optimized image matching technique would use the (supposedly less computationally intensive) histogram metric to achieve a coarse matching, and then finish off with the more discriminative metrics to find the best match. There is a great deal of comprehensive research on image features for the task of image matching, which doesn't need to be repeated. The conclusion so far seems to be that there is no one good set of features for all tasks. So we choose a generic metric that behaves well with respect to false alarms and instead rely on context for robustness to noise and generalization.

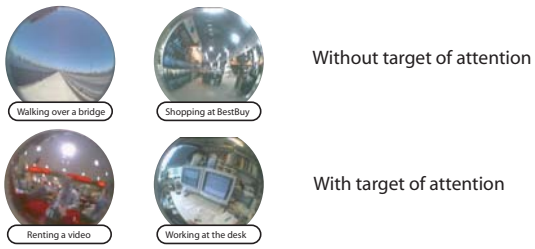

Figure 0-1: Two beneficial side-effect of the fish-eye lens. Objects receiving the wearer's visual attention cover more pixels. The wide-angle capture enables a complete but low resolution sampling of the periphery.

Second, the warping of our images by the fish-eye lens has some beneficial side-effects for the pixel-based metric. Since more resolution is given to the center of the image, objects that are being attended to tend to overwhelm the rest of the clutter (see Figure 0-1). This is qualitatively similar to how the human eye samples the light image falling on the retina. However, these foreground objects have to either be very close or very large for this to happen. Compare this to the case when there is no foreground object (again see Figure 0-1). Now some part of the background is being magnified, but since it is not receiving the wearer's attention, the center pixels will not persist as much as the pixels in the periphery. Schiele's [44] work on segmenting out attentional objects is based on this property. Also, since the

fish-eye lens captures the full periphery with low resolution, cluttering objects in the background (like this fellow pedestrian overtaking the wearer in Figure 0-2) will not affect many of the total peripheral pixels.

Figure 0-2: Generally, objects that are not attended to will only cover a small number of pixels. This is useful for achieving robust estimation of peripheral conditions.

The computational complexity of calculating the pixel-based metric is $O(3HW) = 3(320)(240) = 153,600$. This is unreasonable when we are processing days of video. Also not every pixel in the image has the same importance. For example there is the rim of the fish-eye lens visible in all images. These pixels don't really change from one image to the next. However, a principle components analysis (PCA) will take care of both these problems (please see [38] and [52] for a similar usage of PCA). As part of PCA we compute the eigenvalues and eigenvectors (or eigenimages) of the image covariance matrix. Since our computers couldn't hold a 153,600-by-153,600 element covariance matrix, we bilinearly subsampled the original 320-by-240 images to 32-by-24 pixels, resulting in a 2304-by-2304 covariance matrix. The eigenimages are the optimal (in the least-squares sense) modes or basis vectors for reconstructing the images that were used in estimating the covariance matrix.

The choice of how many eigenvectors to use was determined by a trade-off between reconstruction error and computational complexity incurred in the rest of the processing pipeline. We chose to project the front and rear views on to the subspace spanned by their top 100 eigenvectors. The reconstruction in these subspaces is 85% (front) and 87% (rear). This results in a 200-dimensional feature vector being passed to the next stage of alignment. Figure 0-3 summarizes the feature extraction step for the alignment.

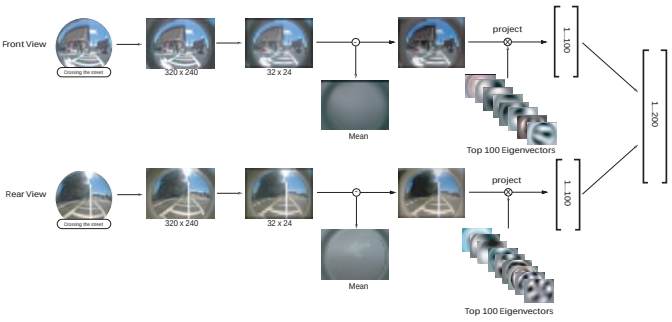

Figure 0-3: The processing pipeline for the alignment feature. The front and rear views are both subsampled and projected on to their respective top 100 eigenvectors. The result is concatenated into a 200-dimensional feature vector.

The Alignment Algorithm

The goal of this section is to be able to take any pair of sequences from the I Sensed data set and match each time step in the "source" sequence with a time step in the "destination" sequence. In other words, as we move linearly through the "source" sequence each moment is associated with a similar moment in the "destination" sequence. We can then use the cost of the match to represent the dissimilarity of the "source" and "destination" sequences. At the same time we are labeling the "source" sequence with the contents of the "destination" sequence. This answers both questions of how similar/dissimilar are two subsequences and why are they similar/dissimilar at the same time. In this section our main piece of technical machinery is the Hidden Markov Model (HMM) to represent constraints of a match and the Viterbi algorithm to perform the actual matching.

Time Constraints

We would like to bias the matching towards smooth transitions in the "destination" sequence from one time step to the next. This follows from the fact that if two points of time in someone's life are close than they should be semantically similar with respect to location, activity, etc. regardless of the sensor reading. For example, say an individual wearing a camera on his chest is walking down a brightly lit hallway. As he walks, he suddenly lifts his arm to rub his eyes, thus completely occluding the camera. The main activity (walking down a hallway) hasn't changed, nor has the location. However, without the smooth time constraint the times when the individual is rubbing his eyes would be matched with other dark moments. So it makes sense to try to limit the transitions in the "destination" sequence to be local in time. However there are two questions that need to be considered about this constraint. At the time-scales greater than a causal sequence, we can expect a longer sequence to be (almost) any permutation of causal subsequences. Hence large transitions in time should be allowed and in any direction in time.

The Alignment Hidden Markov Model

We now encode the constraints discussed above in the form of an HMM. Essentially, we represent the "destination" sequence as an HMM with state transition probabilities that encode the global and local transition constraints. We will call this HMM the alignment HMM. The features of the "source" sequence are the output observations for each state. Let $t = 1...T$ represent the index into the "source" sequence. Let x_t represent the feature of the "source" sequence at time t. Let $s = 1...N$ represent the index into the "destination" sequence, or equivalently, the s-th state of the alignment HMM. Let y_s represent the feature of the "destination" sequence at time s. The goal of alignment can be stated as determining the state sequence, $\{s_t^*\}$, that gives the best possible match to the input features, $\{x_t\}$, from the "source" sequence. This framework is equivalent to dynamic time-warping (DTW), except the cost functions

are represented probabilistically and thus more easily interpretable.

We encode the local and global time constraints discussed above into the transition probabilities of the alignment HMM:

$$p(s_t \mid s_{t-1}) = \begin{cases} Z\alpha^{|s_t - s_{t-1}|}, & 0 \le s_t - s_{t-1} \le K \\ Z\beta, & \text{otherwise} \end{cases}$$

The first case assigns the probabilities for transitions of at most K steps in the "destination" sequence. Its form is exponential to insure that the cost of a single transition that skips n time steps is the same as the cost of n transitions of one time step each. These transitions, which we will call the α-transitions, are the local transitions that try to maintain sequential continuity through momentary matching difficulties from minor insertions or deletions (e.g. those caused by rate differences, temporary occlusions, etc.). The second case assigns a constant probability, β, to global time transitions of any distance and in any direction. Generally, we would set $\beta << \alpha$. These transitions, which we will call the β-transitions, allow an alignment path to "teleport" instantly from any point in time to any other point in time all with the same associated cost. As mentioned before, this is useful when aligning sequences that consist of permuted subsequences, or have long insertions and/or deletions. Since Z is just a normalization constant, K, α and β are the only free parameters.

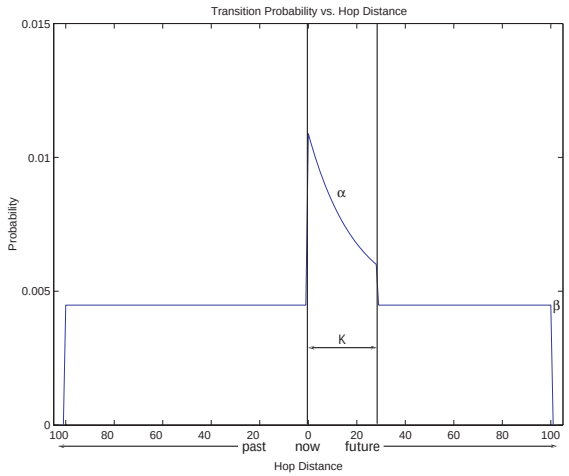

Figure 0-4: The parameterized form of the alignment HMM's transition probabilities.

The state emission probabilities are a function of similarity of the features in the "source" and "destination" sequences. We can define a Gaussian-like emission distribution for each state in the alignment HMM as follows:

$$p(x_t \mid s_t) = Ze^{D(x_t, y_{s_t})}$$

$D(\bullet)$ is the distance function on the features (L_1-norm in our case). Again, Z is a normalization constant. If $D(\bullet)$ were the Mahalanobis distance than this distribution would be exactly Gaussian. However, we use the faster L_1-norm appropriate to our pixel-based features. Also since our features are already decorrelated as a result of the projection on an eigenbasis there is no need to include scaling by the inverse covariance matrix.

Given values for the free parameters, K, α and β, we can compute the optimal alignment of the "source" and "destination" sequences by the Viterbi algorithm. The similarity of the sequences is appropriately measured by the likelihood score calculated during the course of the Viterbi algorithm. Recall that the computational complexity of Viterbi is $O(TN^2)$ in time and $O(TN + N^2)$ in space (we can reduce this by computing the distance and transitions probabilities on the fly but at a severe reduction in speed). Thus as the destination sequence gets longer the computational and storage loads increase quite rapidly. Typically, beam search is used to reduce the computational cost of Viterbi, however, this is not an option for us because the beam would prune all the alignments containing long jumps. This would prevent us from aligning sequences which contain similar causal subsequences but in differently permuted orders. If we keep all the parameters in memory for the fasted compute times, then the longest sequences we can align to are about 5000 steps long*. At a frame rate of 10Hz, this is about 8 minutes. So it is clear that if we are going to align sequences on the order of days or months, we have to use a multi-resolution approach.

A Taxonomy of Alignments

The source and destination sequences don't have to contain the same sequence of features to yield alignments that are useful. In fact the most interesting cases from the point of view of this work are those pairs of sequences that are between the two extremes of being well aligned at every step in time and not being alignable anywhere. Differences might arise due to the speed at which the subject is going through the activities represented in the sequences. There are cases when the sequences share similar parts but the parts are out of order. In these cases, the alignment score (i.e. likelihood of the Viterbi path) will be slightly lower (compared to monotonically match-able sequences) since β-transitions will be necessary to align the two sequences. We will discuss these cases more later on because they provide the means for scene segmentation.

Figure 0-5 shows two typical examples of alignment paths obtained when aligning sequences of quasi-similar content. The pair of sequences on the left are two examples of the subject walking from location A to location B. The sequences are highly similar thus only α-transitions are necessary to align them. This is what we will call an α-match. However, in the source sequence the trip took longer than it did in the destination sequence. The pair of sequences on the right both contain the subject's act of visiting three locations, A, B, and C. However, the order of these visits are different in each sequence. This is recognizable by presence of segments of continuous α-transitions punctuated occasionally by β-transitions. This is what we will call a β-match. The β-transitions occur when the user is transition from one scene (in this case

* This is assuming a 1GHz Pentium IV with about 500MB of RAM.

locations) to the next. Providing a taxonomy of alignments enables users of a search engine based on this work to use some interesting queries. For example, the user might point to an example of himself returning home after work by foot, and then ask for α-matches that occur at a faster speed, thus identifying those times when he returned home on rollerblades.

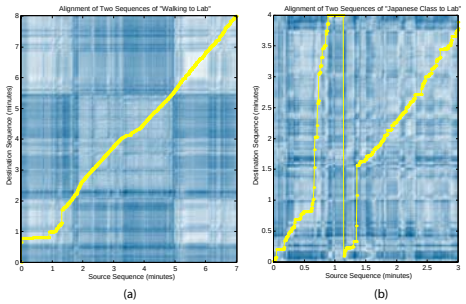

Figure 0-5: Alignment paths for (a) an α-matchable pair of sequences, and (b) a β-matchable pair of sequences.

Data-driven Scene Segmentation

A key capability required for browsing, classification, and prediction, is the segmentation of the data into manageable coherent chunks, or scenes. Segmentation into scenes is useful for browsing because it helps depict to the user what the main parts of a temporal sequence are and makes it easy to show how they relate to each other (e.g. scene transition graph). Scene segmentation is useful for classification because it guides the choice of labels and determines the intervals over which to integrate information from low-level features. Prediction becomes an insurmountable task with temporal sequences that have long and complicated dependencies between points in time, especially if those dependencies reach far back into the past. We will show that our method for scene segmentation is well-suited for compressing the past into a set of manageable chunks. In later chapters, we will show how this makes it possible for us to build prediction models that can use larger amounts of the past than previously possible.

Most of the difficulty researchers have faced while tackling this problem is the lack of a suitable definition of what a "scene" actually is. Many researchers base their algorithms for scene change detection on shot boundary detection [30]. Shot boundaries (the switching of camera views or edit points) are artificial artifacts introduced by the video's editor and algorithms for detecting them will usually fail on contiguous unedited video captured by a single camera. Certainly, this is one way to avoid having to define what a scene is, since the editor has already define them. On the other hand, some researchers define the scene as being a interval of time during which a pre-selected set of features are statistically constant, such as motion [47] or color [48]. Scenes changes are detected by building detectors on top of a time-derivative representation of these features. The main problems with this class of approach are that they only use local information (time-derivatives of the time-localized features) and it is very difficult to adapt to gradual changes in the feature statistics (non-stationarity). Another class of methods, model-based segmentation (train a model, use it to label the data), requires that you are able to define exactly what you mean by a scene, via feature-selection, rules or by labeling training data. For example, we might decide to equate location to scene, label a portion of data as such, train location models, and then use them to segment the rest of the data. These methods of course only work when your training set adequately covers the space of possible test inputs, a situation to which change detection methods are more robust.

We propose an alignment-based segmentation. Suppose we wish to find the scenes in a given sequence. Suppose also that our knowledge consists only of a set of previously seen sequences. First we proceed by aligning our given sequence with our entire bag of examples, so that every moment in the given sequence is matched to a moment in a past example. Let's assume there is a point where our current sequence is aligning nicely with a particular past sequence (indicated by α-transitions). So we keep traveling down our current sequence, watching the alignment path as we go. Eventually, during the alignment the past sequence that we have been aligning to will diverge and we will have to make a β-transition to another remote place in our bag of past examples. Since the alignment is the best possible, this means there are no other past examples that better align to our sequence for a longer period of time (there might be shorter ones). We have reached a point in our sequence beyond which all of our past examples don't extend. This is a natural place to deduce a scene break.

The basic principle being used is minimum description length (MDL), since we always choose longer scenes if there is evidence that a similar lengthy scene has occurred before. Since our alignment algorithm tries to minimize the number of β-transitions it can be thought of as computing the MDL labeling of the given sequence using the past examples as possible labels.

In essence, to support a scene in this framework, the system merely needs to find at least one match somewhere else in the data. The longer the match, the longer the scene, regardless of what happens inside. This way scenes are minimalistically defined by what sequences are repeated in the data and are independent of the nature of the scene.

We know give the full details of the segmentation algorithm.

1. Alignment: Let $x = \left(x_1, \ldots, x_T\right)$ be the source sequence in which we wish to find scene breaks. Let $\Upsilon = \left\{y^1 = \left(y_1^1, \ldots, y_{N_1}^1\right), \ldots, y^L = \left(y_1^L, \ldots, y_{N_L}^L\right)\right\}$ be the set of L destination sequences. To simultaneously align x

with all of the sequences in Υ we use an alignment HMM with a state-space that spans all of the destination sequences and thus has $N = \sum_i^L N_i$ states. We also need to slightly generalize the transition probabilities to this case so that the α-transitions are only between intra-sequence states,

$$p_{seg}(s_t \mid s_{t-1}) = \begin{cases} Z\alpha^{|s_t - s_{t-1}|}, & 0 \le s_t - s_{t-1} \le K \text{ and } s_t, s_{t-1} \in \text{same sequence} \\ Z\beta, & \text{otherwise} \end{cases}.$$

Thus, the $N \times N$ transition matrix will have a block diagonal structure. Distances need to be computed between all pairs of elements in x and Υ for a $T \times N$ distance matrix. Computing the Viterbi path of this HMM on the source sequence will yield an alignment path, $s^* = \left(s_1^*, \ldots, s_T^*\right)$ $s_t^* \in 1 \ldots N$, that best matches moments in x with any of the moments in sequences in Υ.

2. *Scene Change Score:* A scene break occurs when there is a β-transition. However, not all β-transitions are equal. So we score each moment in the alignment path as

$$c_t = \begin{cases} \left|s_t^* - s_{t-1}^*\right| & 0 \le s_t - s_{t-1} \le K \text{ and } s_t, s_{t-1} \in \text{same sequence} \\ N/L & 0 \le s_t - s_{t-1} \le K \text{ and } s_t, s_{t-1} \in \text{different sequences} \\ 0 & \text{otherwise} \end{cases}.$$

This allows longer jumps to have larger scores but assigns a constant score, N/L (the average sequence length), to jumps between sequences in Υ. Jumps less than size K (i.e. α-transitions) receive a minimal score of zero.

3. *Hierarchy of Scenes:* Finally if we sort the values of $\{c_t\}$ in descending order and successively split the sequence x at the associated times, a hierarchy of scenes is generated ordered by level-of-detail. Another way to describe the construction, is as sweeping a threshold from top to bottom down a graph of c_t, successively splitting the x sequence as the threshold encounters peaks.

Figure 0-6 shows an example of scene segmentation when Υ contains only one sequence that is locally similar to x but globally different. This way when aligned they yield a permuted path (see section 0). In order to achieve the best segmentation results it is desirable for the destination sequence to be a permuted version of the source sequence. Otherwise if there is no local similarity then this technique simplifies to pair-wise image clustering with temporal-smoothing. Thus it is important to include as much material in the set of destination sequences, Υ, as possible so as to increase the probability of find a good local match to each moment in the source sequence. However, computational requirements of the alignment will increase rapidly with the size of Υ. In the next section we show methods for alignment at coarser resolutions that will allow us to include more in Υ.

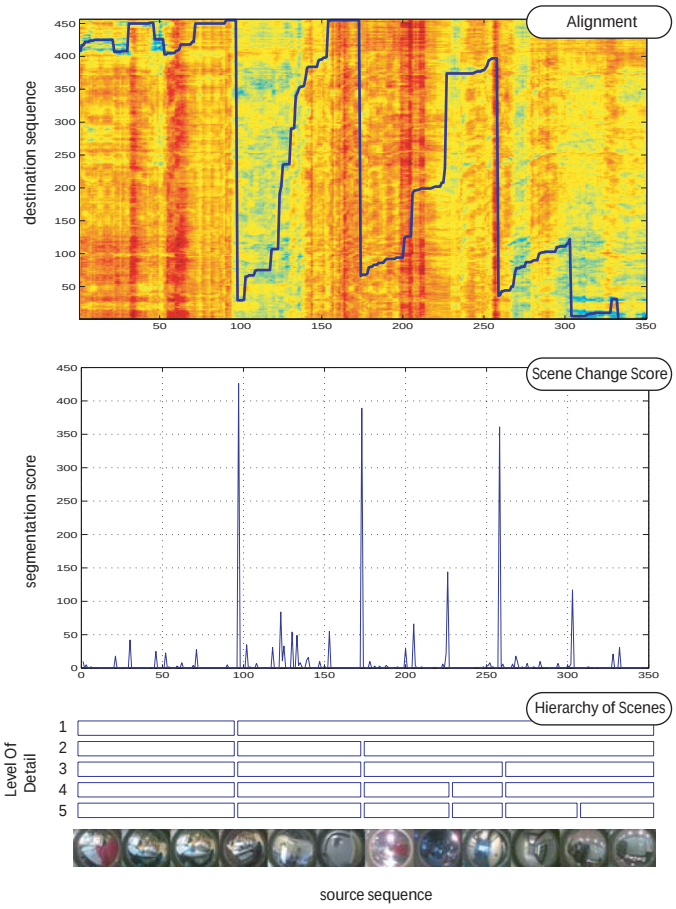

Figure 0-6: The algorithmic pipeline for segmenting a source sequence according to the contents of a destination sequence. Starting from top and proceeding to the bottom, (1) Alignment of the source sequence to the destination sequence, (2) Scoring each time step for the possibility of a scene change from the alignment path, (3) a Hierarchy of Scenes can be generated by sweeping a threshold across the scene change score.

Multi-scale Alignment

Alignment at the finest level of detail would consist of aligning each frame of a pair of sequences at the original recorded rate. However, since the computational cost for aligning a pair of sequences grows prohibitively with the length of the destination sequence, we need to adapt a multi-resolution method in order to align sequences on the order of days. In this section we show alignment at three different time-scales, fine (frame-rate), medium (run-length encoded signal), and coarse (5 minute chunks) alignment.

Fine-scale Alignment

With a fine-scale alignment on portions of the I Sensed data, it is possible to do very detailed comparison of two activity sequences. For example, it is possible to take two examples of the subject walking to the store and, after aligning at frame-rate, compare the matched images for

differences, missing objects, lighting changes, and so on. Figure 0-7 gives an example of the subject walking entering a building on campus and walking down a hallway. Notice that the alignment between the two sequences is exact down to what doorway and bulletin board he is passing by. In some frames you can see the presence of other people in the hallway (e.g. frame 14) that are not presence on May 10 but are on May 4.

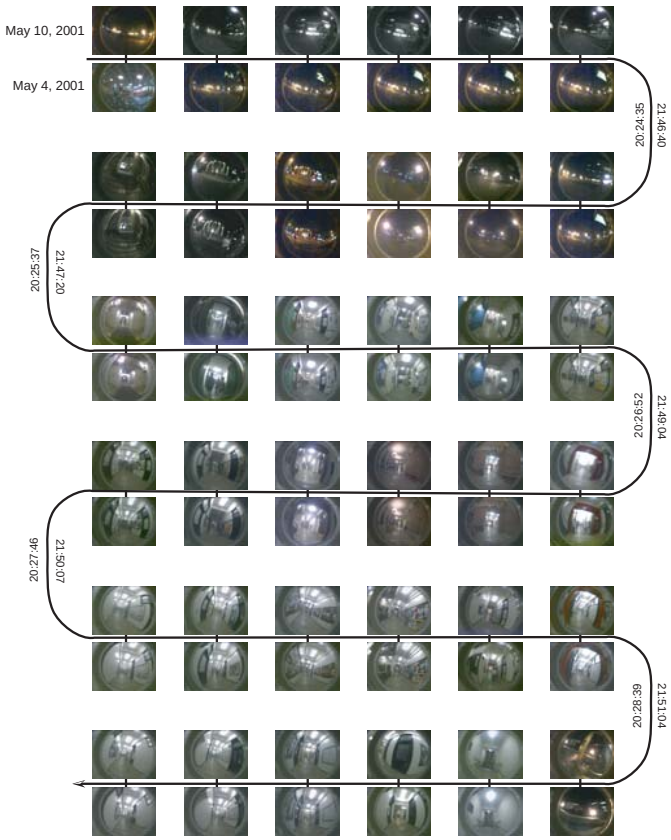

Figure 0-7: Fine-scale alignment of two similar scenes that happened on separate days: entering a building and walking down a hallway.

As mentioned before, it is too computationally intensive to do this kind of fine-scale alignment between every pair of moments in the I Sensed data. Sequences should be segmented into manageable chunks and evaluated for overall pair-wise similarity before they are chosen as candidates for fine-scale alignment.

Run-length Encoding

A useful tool for time-compressing the feature sequences is run-length encoding. As you might intuitively expect, video of an individual's life is full of long sequences where not very much is happening, punctuated by bursts of activity. This makes it an ideal candidate for run-length encoding (RLE). The procedure for RLE on video is as follows:

1. Choose a change threshold, τ and initialize $t, t^* = 0$.

   If $\frac{D(x_t, x_{t^*})}{D_{\max}} > \tau$ then add the current image, x_t, to the compressed sequence and set $t^* = t$. ($D_{\max}$ = the largest distance possible between a pair of images)

2. Set $t = t + 1$ and repeat step 2.

The resulting time-compressed sequence is irregularly subsampled where the sampling rate is proportional to the rate of change in the video. An entire day can be RLE'ed at a 15% change threshold from the original ~150,000 images to a manageable 3,000-5,000 images. The fact that such small threshold will nevertheless yield large compression rates is very fortunate. As we can see in **Error! Reference source not found.** the original video, even at such a short time-scale (one minute) and active period (shopping), contains long sequences of very little change as the user waits at the deli or browses through the beverages. At 5% the some of the long sequences are still exist due to small amounts of motion that is usually present when a camera is mounted on a person. However, at 15% no more repetitions exist but all of the major views are included.

Medium-scale Alignment

The RLE compression step at a 15% change threshold allows us to align a pair of days in about 5 minutes of a single PC's time. The average frame rate of RLE-15% compressed video in the I Sensed data set is 0.1 Hz or 1 frame every 10 seconds, but the instantaneous frame rate is highly variable, from 10 Hz to 0.001Hz.

In order to evaluate the use of the alignment score as a measure of similarity between days, we chose to compare the rate at which locations in the source and destination sequences were correctly matched by the alignment. We manually labeled the situation class of every 5 minute interval of May 9th and the 10 randomly chosen days. The situation labels and the categories that we chose to group them into are given in the next chapter on situation classification. The resulting labeling can be seen in **Error! Reference source not found.**. Visual inspection of the situation labeling without alignment doesn't clearly show why a pair of days would be similar or not. However, if we align the pair of days and then compare the situations that were matched then we can begin to see how the days are dissimilar or similar. In **Error! Reference source not found.** we show this aligned comparison of situation. Notice that the similar day succeeds in matching a number of outside and inside situations. Contrast this to the dissimilar day where the only matches were the ubiquitous "at work" situation and the "office" (sometimes).

Coarse-scale Alignment

When the goal of the alignment is provide either links of association to common moments in the past or derive good scene segmentations, then it is necessary to include as many days in the alignment HMM as possible. To this end we introduce our coarsest scale of alignment which allows us

to align a given day against 30 other days. The key component of the coarse alignment algorithm is to use the alignment scores of a medium-scale alignment on 5 minute chunks as the input into the coarse-scale alignment. The outline of the coarse-scale is as follows:

1. For every pair of 25 RLE-15% frames in $\{x = 1day, \Upsilon = 30days\}$ we align and store the alignment score in a TxN similarity matrix. We call these 25 frame sequences the coarse chunks. They vary in absolute time duration from 10 secs to 10 minutes, but average 5 minutes.
2. Then we align x against Υ using the inverse similarity matrix as the distance function $D(\bullet)$ and the same transition function, $p_{seg}(s_t \mid s_{t-1})$, that was defined in section 0 (Data-driven Scene Segmentation).

We chose a set of 32 days* to completely align with each other (i.e. 1 day vs. 31 days for each day). The computation of the alignment scores for all days in step 1 of the coarse-scale alignment was the most expensive, taking about 1 night to compute on a 1GHz computer. However, the result is a 3500-by-3500 similarity matrix that can aligned in under 10 minutes.

The coarse-level alignment can be used for a number of tasks:

- Deriving an associative network between moments in a large number of days
- Segmenting scenes for browsing
- Clustering similar days based on matching of similar moments rather than a global aggregrate score.
- Building prediction models that model dependencies over days
- Classifying situations

In the upcoming sections we evaluate a few of these.

5 Situation Classification

"Where are you and what are you doing?" are two of the most basic facts about your state. Many of your basic decisions, activities, and the events that happen to you are dependent on your location and the state of your location (e.g. turning down a hallway, meeting someone, turning on the light, eating at a restaurant). We believe that it is not location alone or activity alone that determines your context or influences your next action, but rather the interaction between location and activity. It doesn't make sense to model location irrespective of activity and vice versa. The two concepts are so highly correlated (certain locations are for certain activities, certain activities are for certain locations) that from a statistical point of view they must be modeled together. This coupling of location and activity is represented together in the concept of a situation.

Presumably at this moment you are sitting somewhere, perhaps your office or the library, reading this document. Let's assume that reading is one of the many activities that you conduct in your office. Arguably, reading only makes up a small portion of what could be called your office situation. Your office situation might also include speaking with colleagues, talking on the phone or typing at your computer. The office situation seems to be delineated by the physical boundaries of your office walls. However, it doesn't make sense to define all situations by the location they happen in. For example, the situation of "eating out" could and usually does happen across many locations (the local neighborhood café, the posh Italian restaurant in downtown, etc.).

In the upcoming sections we show how we can use the alignment similarity measure (given in 0) to classify situations in the I Sensed data set. We give results* for situation classification when using only short-term context (one RLE chunk vs. one RLE chunk alignment) and when using long-term context (one day vs. 30 day alignment). Naturally there are situations when one type of context is more appropriate than the other. In the last two sections of this chapter we give a method for combining the two types of context that improves classification accuracy over using either type of context alone.

The Situations

We labeled 20 days of the days used in the 30 day alignment (of section 0) for location every 5 minutes for a total of ~2000 labeled sections. If more than one location occurred in a given 5 minutes then that 5 minutes received multiple labels. To build our situations we grouped 58 locations by common activities. Table 0-1 gives the resulting 19 situations after the grouping. Naturally some of the locations contain other locations (e.g. the subject is always at the Media Lab if he is in his office).

Context-free Classification

In 0 we calculated the similarity between every pair of medium-level chunks (25 frames of RLE at 15%) in 30 days by aligning the frames and noting the log likelihood of the alignment. Our hypothesis is if different chunks are of the same situation (say both are from the street situation) then their alignments should give high scores relative to other chunks from different situations. Our earlier experiments had hinted at this possibility. So for any given chunk another chunk that has a high alignment score relative to it should be of the same situation class.

To test this hypothesis we took every chunk in the labeled 20 days and order the other chunks by their alignment score. The chunk was correctly classified if the chunk with the highest alignment score was from the same situation class and incorrectly classified if not. This is also the rank-1

* Actually 34 sequences since two of the days were split into two runs since we needed to briefly shut the data collection wearable off for maintenance.

* We will be giving the total accuracies in two flavors. Since the situations occur with vary different frequencies we need both. The accuracy (in the plots) is simply the number of correctly classified situations over the total number of situations seen in the test set. In the text we will also quote the average accuracy, which is the mean accuracy for all 19 situations. This accuracy is immune to the effects of varying situation frequency.

accuracy. The rank-2 accuracy is when we consider a chunk correctly classified if at least one correct match is in the top 2 scoring chunks. This is a completely unsupervised classifier since no knowledge of the labels was used to generate the similarity measure. Since the score is only dependent on the alignment of a pair of medium-level chunks (approx 1-5 minutes in duration), the classification is only affected by short-term memory (or context).

Figure 0-1 gives the results for matching situations of chunks with only short-term memory over 20 days of data (or about 2000 chunks). The chance recognition rate is the probability of a correct match if we just choose another chunk at random. Recognition rates vary quite a bit between class but all are many times larger than the chance recognition rate, indicating that the alignment score is a decent measure for similarity of situation. In fact the overall score for all situations is 89.4% (rank-1) and 95.0% (rank-2) over time. The average accuracy over the 19 situations is 82.4% (rank-1).

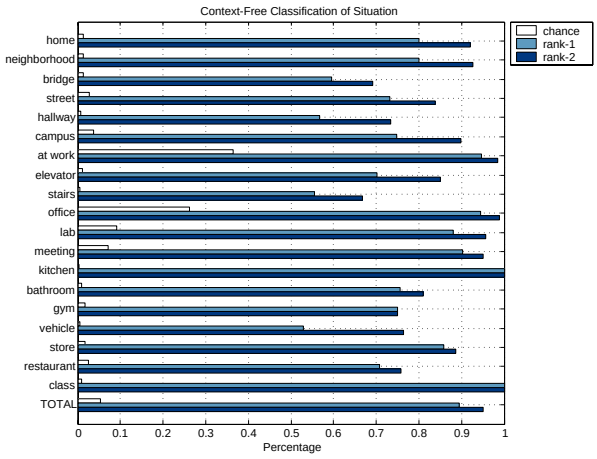

Figure 0-1: Rank-1 and rank-2 situation matching accuracy for the medium-level chunks via their alignment score. The figure gives the per situation accuracy and the total accuracy along with the chance recognition rates.

Far vs. Near Matches

We examined the errors that the short-term classifier makes in the experiment above. When the ranking of examples by the alignment score is unable to find a similar situation in the same day as the test chunk and is forced to choose one in another day. There is nothing wrong with choosing a match in a different day, but it turns out that the short-term classifier is not good at matching chunks that are far apart in time. Since only 57.9% of the closest matches in the experiment above are matches to other chunks within the same day of the test chunk, this weakness is expected to affect overall performance quite a bit. To quantify this intuition, we decided to compare matching accuracies for when we force the match to be in the same day (near) and in another day (far). Figure 0-2 gives the resulting recognition scores. Notice that the near accuracy (95.1%) is quite high compared to the far accuracy (72.2%). The average far accuracy over the 19 situations is 56.4% and the average near accuracy is 87.4%. This validates our intuition that near matches are easier for the context-free classifier than the far matches.

If we examine these far errors more closely we see that many of the mismatched chunks have high scores and are visually similar but don't make sense given the flow of events around the test chunk. This is a hint that context can help us correctly classify these "far" matches.

Classification with Long-term Context

Fortunately, we have an ideal tool for bringing long-term context to the classification problem – alignment. Recall in 0 we were able to align each day against 30 other days at the coarse level of detail (chunks). We can view this alignment in a different light. By aligning we matched chunks in a given day to chunks in the other 30 days. However, the each chunk-to-chunk match must contribute to a good alignment of the entire day to the other 30 days and not just be a good short-term match. Hence the coarse level alignment will smooth out matches that are good in isolation but don't follow the usual progression of events seen in the other days that we are trying to align with.

The long-term classifier is then constructed by matching every test chunk with the chunk that was aligned to it during the coarse daylong alignment. Figure 0-3 gives the results of the classification with context. The overall rank-1 accuracy* is 94.4% and the rank-2 accuracy is 96.6%. The average rank-1 accuracy is only 73.4% due to a few low performing classes (stairs, restaurant, bridge). This is an improvement over the context-free classifier by about 7 percentage points. However, recall that the context-free classifier is able to choose from the (easier) near matches while this classifier (by design) can only align chunks to chunks in different days. Hence the matches are all far matches. This means we should be comparing our accuracy to the context-free classifier's far performance of 72.2%. This is 24 percentage points below the contextualized classifier's performance showing that context indeed helps a great deal when we are forced to make matches between separate days.

Situations	Locations (grouped by activity)
home	home
neighborhood	Beacon St., Massachusetts Ave. (Boston-side)
bridge	Harvard Bridge, Longfellow Bridge
street	Kendall Square, Boston Downtown, Main St., Memorial Dr., Cambridgeside, 77 Massachusetts Ave.
hallway	Infinite Corridor
campus	inside & outside of bldg. 56,. 66, 7, 10
at work	Media Lab (entire building)
elevator	elevator (anywhere)

* Since the coarse alignment was done over a larger set than what was labeled, some labeled chunks are matched to unlabeled examples. We threw these out of the tabulation, resulting in the vehicle situation having no pairs of matches to count.

stairs	stairs (anywhere)
office	office (at Media Lab)
lab	Dismod, Garden, Interactive Cinema, copiers, CASR, Advisor's office
meeting	Facilitator Room, Black Couch Area, Bartos Auditorium
kitchen	kitchen (anywhere)
bathroom	bathroom (anywhere)
gym	DuPont Athletic Center
vehicle	taxi, bus, subway
store	Tower Records, Realtor, Graduate Housing Office, Medical Center, Color-Kinetics Inc.,The Food Trucks, Student Center, ATM
restaurant	Ginza, Cheesecake Factory, Kendall Foodcourt, Toscanini's, Bertuccis, AllAsia, Whitehead Cafeteria,Walker Cafeteria, Bio-Cafe, Penang
class	Japanese

Table 0-1: The situations and the actual location labels that they represent.

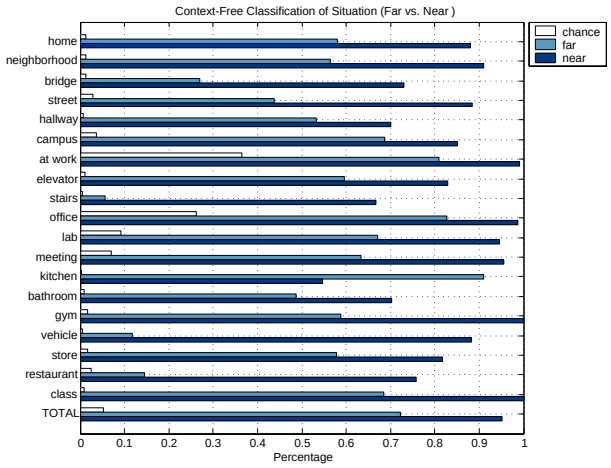

Figure 0-2: The performance of the short-term classifier when we force the match to be in the same day (near) and in another day (far).

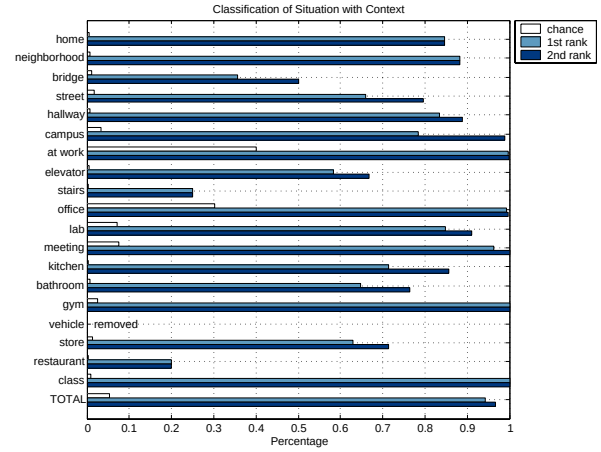

Figure 0-3: The performance of the contextualized classifier at matching situations. Rank-1 is the accuracy when only considering the actual chunk aligned to the test chunk. Rank-2 is when a correct match exists within one time step in either direction along the alignment path. The vehicle situation had no labeled pairs of matches to count.

Hybrid Classifier

Finally, we would like to combine our ability to find good matches within the same day with our ability to find matches between separate days. To do this we can use the following simple rule:

> *If a given test chunk's context-free match is in a separate day then classify this chunk with the contextualized classifier, otherwise it is a near match and thus we should use the context-free match.*

A situation classifier based on this rule will take advantage of the strengths of context-free and contextualized classification. Refer to Figure 0-4 for the per situation classification accuracies of this hybrid classifier. The overall accuracy is now 97.0% over 20 days of situations. The average accuracy is 85.5%.

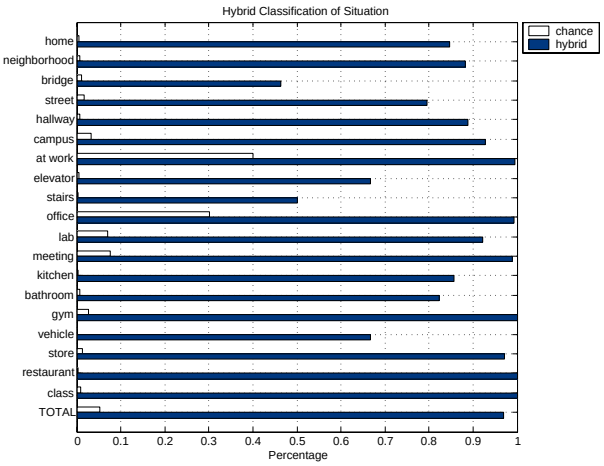

Figure 0-4: Performance of the hybrid classifier at situation classification. This classifier uses context-free classification on the near matches and contextualized classification on the far matches.

6 Life's Perplexity

"When you come to a fork in the road, take it." –Yogi Berra

At each moment in our lives, not every possible action is available for us to take. One cannot teleport in both time and space from breakfast at home to dinner at a restaurant in the blink of an eye. We can expect some moments to present numerous paths that smoothly diverge into radically future situations from the present situation while other moments may provide few alternatives. Since there is a natural tendency for us to limit the amount of variability

in our life, we might choose to habitually ignore certain alternative paths during the course of our day-to-day activities. There are an infinite number of possible routes to take from home to work, but out of habit and practicality we usually settle on a very small number like two or three. The concept we are referring to, which is concerned about the number of paths of action emerging from a given scene, is called the perplexity of a scene*.

In the previous chapter we developed a similarity measure that allows us to compare moments and intervals of video from an individual's life. By doing so we constructed an abstract space, one for each time-scale of the application of the similarity measure, in which the streams of sensor data are winding paths. Let's call this space a situation space since we showed in 0 that two similar intervals of video (and hence near to each other) are very likely to be of similar situations. Since no two moments in someone's life are exactly the same, the winding path never intersects itself unless we start to discretize or cluster the situation space. Once this is done, we can measure the perplexity (i.e. the number of forks in the road) at each point in the sensor stream. Places in the sensor stream that display a high fan-out can be thought of as *decision points*. In this chapter, we propose a method for finding these decision points and then go on to measure their perplexity and the consistency of the choices taken at those points (prediction accuracy).
Our approach is to first segment the sensor stream based on where we believe the decision points are. This process is based on the scene segmentation algorithm given in section 0. Then we assign discrete symbols to the sequences between the decision points by clustering with the similarity measure. After collapsing all runs of a symbol to a single symbol we can estimate the predictive accuracy of a 1st order Markov model and measure the perplexity of each symbol (see **Error! Reference source not found.**). We conclude the chapter by interpreting these results.

Clustering the situation space

Previously in section 0 we described a scene segmentation algorithm that essentially determined scene boundaries by where β-transitions occurred. Since these are the places where a given sequence diverges from the best-aligned past/future example, we can imagine that the individual has made a decision that is not typical (i.e. different from the past or future). In the following experiments we use the segmentation (842 scenes over 30 days) provided by the alignment of 1 day against 29 other days (section 0). Thus in this case, a β-transition signifies a point in one day where the experiences of the individual diverge from what was observed in all the other 29 days.

To assign symbols with each of these, we constructed a merge tree by successively merging the most similar pair of scenes in an agglomerative bottom-up manner. Similarity between clusters was calculated as the similarity of the least similar pair of examples in the clusters (e.g. this is the 'complete link' metric which favors compact clusters, as opposed to the 'single link' metric which favors long chains). The result is a binary cluster tree, which we show, fully depicted down to 200 clusters in **Error! Reference source not found.**. To obtain an N-clustering of the 842 scenes, we simply stop merging when we reach N clusters.

Choosing the number of situations

We determine the number of symbols by how predictive the symbols are. The naïve approach is to plot the prediction accuracy versus the number of clusters. We show this for the cluster tree on the 842 scenes from 5 to 200 clusters in **Error! Reference source not found.**. The predictive 1st order Markov model is,

$$x^n_{pred} = \arg\max\, p(x^n_t = i \mid x^n_{t-1})$$

where $x^n_t \in (1,\ldots,n)$ is the symbol of the n-cluster set at time, t. The probability distribution p is estimated empirically from co-occurrence counts on a training set after removing symbol repetitions. Accuracy is calculated by averaging the results over a 30-way cross-validation (leave 1 day out for test, train on the remaining 29 days).

Naturally, as the number of symbols increases, the probability of chance decreases, making the prediction task successively more difficult. Hence there is an unfair bias towards fewer symbols. So a straightforward use of prediction accuracy to choose the number of symbols is not appropriate. Instead we would like to measure how much information about the future, x^n_t, is extractable by a 1st order Markov model from the past, x^n_{t-1}. The standard measure for this is mutual information [10]. Mutual information between two variables yields the number of bits of information that one variable has about the other:

$$I(x^n_t; x^n_{t-1}) = \sum_{i=1}^{n}\sum_{j=1}^{n} p(x^n_t = i, x^n_{t-1} = j) \log \frac{p(x^n_t = i, x^n_{t-1} = j)}{p(x^n_t = i)\,p(x^n_{t-1} = j)}$$

In this case, $p(x^n_t, x^n_{t-1})$ is again estimated from co-occurrence accounts over a training set after removing symbol repetitions. Finally, in Figure 0-1 we plot the number of bits of mutual information per symbol,

$$B_n = \frac{I(x^n_t; x^n_{t-1})}{n}$$

versus the number of symbols. We notice that there are two opposing forces at work in this graph. When using too few symbols (<30), information about the underlying sensor stream and hence the actual scene is lost and severe perceptual aliasing blurs out the predictive cues from the past about the future. When using too many symbols (>30), less information is lost but the model is less able to generalize from its training examples. The result is that in between these two extremes (at around 30 symbols) there is an empirical optimum number of symbols that balances the trade-off between generalizability and perceptual aliasing.

* We use the word scene to generically refer to any interval of experiences in a person's life.

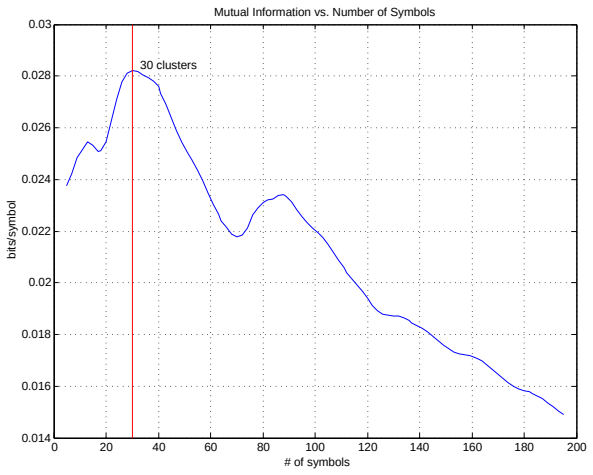

Figure 0-1: Number of bits of mutual information per symbol between a pair of successive scene symbols over 30 days.

Perplexity and prediction accuracy

Having settled on 30 symbols as the optimal number of clusters (for our given cluster tree) we can now answer questions about the predictive capacity and perplexity of the I Sensed data over a period of 30 days. As noted before not every moment presents the same number of alternatives for the future. As intuition would suggest, some symbols yield more consistent predictions than others. The rank-1 accuracies vary from 0% to 60%, but don't seem to have any relationship to the perplexity of the symbol. This independence of predictive accuracy from perplexity is rather anti-intuitive. This means that the high perplexity is caused by the occasional occurrence of an unusual symbol after a given symbol, but the top 4 (rank-4) predicted symbols do represent the most typical situation. For example approx 50% of all typical choices are in the top 4 choices made by the 1st order Markov model.

7 Conclusion

In our opinion the most important contribution of this work is not the specifics of the algorithms we presented but rather the proof of feasibility and the empirical results we show about the complexity of the sensing and modeling required to segment, classify, and predict events in an individual's day-to-day life. Of course, we expect many improvements on this work, especially in terms of more sophisticated models for prediction and (as always) more data with more subjects, but we believe that a few core ideas will survive this evolution for a long time to come.

First, insect-like perception via low-resolution but wide field-of-view sensors provides just the right level of robustness and just the right kind of information needed to recognize the large variety of situations over the course of an individual's day. The sensors don't just focus on the area in front of the subject but it captures the periphery and rear, thus recording information about the user's surroundings. We have found that by storing this kind of full-surround view-dependent information we can do very reliable situation matching (which subsumes location matching). These types of results are in agreement with the studies on insect navigation.

Second, no complicated models based on highly specific knowledge about geometry or physics are required to match sequences of views in timescales from minutes to days. It turns out that all the variations in orientation of the camera (caused by the subject's body movement) and the variations in lighting conditions (caused by weather, artificial lighting, AGC, etc.) are actually not so great when compared to the consistency displayed over many days. Truly debilitating variations in sensing conditions that prevent us from finding a reasonable match are rare and are simply indications of an unusual situation (something that is interesting in itself). A person's life is largely classifiable by simple alignment and matching techniques at the pixel level! Let's also not forget that all the experiments done were performed with a paltry 32x24 pixel image from each of the front and rear views*.

Third, a person's life is not an ever-expanding list of unique situations. There is a great deal of repetition and is evidenced by the success of the alignment and matching techniques used to define our similarity measure. Also we gave quantitative estimates of the actual perplexity of the various moments in the subject's day. This analysis is very dependent on the class of symbols used to describe the evolution of an individual's day. However, when we use situation-specific symbols the statistical perplexity we measured is 4 for 50% of the situations. This means that in 50% of our daily situations, we typically limit ourselves to, or, are typically limited to only 4 choices. We believe this will have deep ramifications for the feasibility of general-purpose agents.

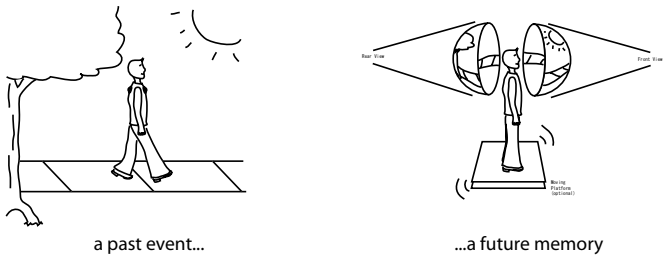

Figure 0-1: A proposed environment for re-experiencing the memories recorded by our I Sensed wearable. Front and rear views are projected onto hemispherical screens along with audio as the audience sits or stands on a motion platform.

We can be certain that we will have the technology available to record more and more of our lives for later *personal* exploration and use. If this evolution is

* This will hopefully please those concerned about the privacy issues surrounding ubiquitous cameras.

accompanied with a similar evolution in privacy protection then we can as a society and as individuals benefit from the availability of such records. The work in this thesis can be used to provide privacy filters on content (for example, sense but don't record in certain situations), but their actual use in practice will undoubtedly be dictated by larger forces.

There are many suggestive environments for re-experiencing past events recorded via wearable sensors (see Figure 0-1 for one possibility). As cameras become smaller and lower power and higher resolution, we can imagine the high quality recording of individual's memories. Again we don't need to limit ourselves to just the visual. These memories will become valuable commodities depending on the person and activity involved. Imagine training "memories" captured from fire fighters and police in real high-risk situations or Olympic athletes performing at their peak. These records can also be used for profiling processes such as the activities of doctors in hospitals to understand inefficiencies and the conditions that lead to errors. We have shown that at least we won't be stuck with rewind and fast-forward as our only interfaces into the years of our lives' recordings.

ACKNOWLEDGMENT

The authors wish to thank A, B, C. This work was supported in part by a grant from XYZ.

REFERENCES

[1] P. E. Agre, The Dynamic Structure of Everyday Life, 1988, Ph.D., Department of Electrical Engineering and Computer Science, Massachusetts Insitute of Technology, Cambridge

[2] F. Attneave, *Dimensions of similarity*, American Journal of Psychology, 63 (1950), pp. 516-556.

[3] L. Barsalou, *Frames, concepts, and conceptual fields*, in E. Kittay and A. Lehrer, ed. eds., *Frames, fields, and contrasts: New essays in semantic and lexical organization*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1992, pp. 21-74.

[4] L. W. Barsalou, *The content and organization of autobiographical memories*, in U. Neisser and E. Winograd, ed. eds., *Remembering reconsidered: Ecological and traditional approaches to the study of memory*, Cambridge University Press, Cambridge, 1988, pp. 193-243.

[5] A. J. Bell and T. J. Sejnowski, *The 'Independent Components' of Natural Scenes are Edge Filters*, Vision Research,: (1997), pp.

[6] V. Bush, *As We May Think*, The Atlantic Monthly, 176 (1945), pp. 101-108.

[7] C. Chesta, A. Girardi, P. Laface and M. Nigra, *Discriminative Training of Hidden Markov Models using a Classification Measure Criterion*, ICASSP,: (1998), pp.

[8] B. Clarkson and A. Pentland, *Unsupervised Clustering of Ambulatory Audio and Video*, in, ed. eds., *ICASSP'99*,, http://www.media.mit.edu/~clarkson/icassp99/icassp99.html, 1999, pp.

[9] B. Clarkson, N. Sawhney and A. Pentland, *Auditory Context Awareness via Wearable Computing*, in, ed. eds., *Perceptual User Interfaces*,, San Francisco, CA, 1998, pp.

[10] T. Cover and J. Thomas, *Elements of information theory*, Wiley,: (1991), pp. 183-223.

[11] D. C. Dennett, *Cognitive wheels: The frame problem of AI*, in C. Hookway, ed. eds., *Minds, Machines, and Evolution, Philosophical Studies*, Cambridge University Press, Cambridge, 1990, pp. 129-152.

[12] A. K. Dey, D. Salber, G. D. Abowd and M. Futakawa, *The Conference Assistant: Combining Context-Awareness with Wearable Computing*, The Third International Symposium on Wearable Computers, IEEE,: (1999), pp. 21-28.

[13] M. Eldridge, M. Lamming and M. Flynn, *Does a Video Diary Help Recall?*, in A. Monk, D. Diaper and M. D. Harrison, ed. eds., *People and Computers VII*, Cambridge University Press, Cambridge, 1992, pp. 257-269.

[14] J. Farringdon, A. J. Moore, N. Tilbury, J. Church and P. D. Biemond, *Wearable Sensor Badge & Sensor Jacket for Context Awareness*, in, ed. eds., *The Third International Symposium on Wearable Computers*,, San Francisco, CA, 1999, pp.

[15] B. Feiten and S. Gunzel, *Automatic Indexing of a Sound Database Using Self-organizing Neural Nets*, Computer Music Journal, 18:Fall (1994), pp. 53-65.

[16] D. J. Field, *What is the goal of sensory coding?*, Neural Computation, 6 (1994), pp. 559-601.

[17] J. Foote, *A Similarity Measure for Automatic Audio Classification*, in, ed. eds.,, Institute of Systems Science,, 1997, pp.

[18] G. R. Garner, *The processing of information and structure*, Wiley, New York, 1974.

[19] E. L. Grimson, C. Stauffer, R. Romano and L. Lee, *Using adaptive tracking to classify and monitor activities in a site*, in, ed. eds., *Computer Vision and Pattern Recognition*,,, 1998, pp.

[20] J. Han, M. Han, G.-B. Park, J. Park, W. Gao and D. Hwang, *Discriminative Learning of Additive Noise and Channel Distortions for Robust Speech Recognition*, in, ed. eds., *ICASSP*,,, 1998, pp.

[21] J. Healey and R. W. Picard, *StartleCam: A Cybernetic Wearable Camera*, The Second International Symposium on Wearable Computers, IEEE,: (1998), pp. 42-49.

[22] G. Iyengar and A. B. Lippman, *Videobook: An Experiment in Characterization of Video*, Intl. Conf. Image Processing, IEEE,: (1996), pp.

[23] M. Jogan and A. Leonardis, *Robust Localization Using Panoramic View-Based Recognition*, Internation Conference on Pattern Recognition, 4 (2000), pp. 136-139.

[24] J. John R. Deller, J. G. Proakis and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan, New York, 1993.

[25] S. Judd and T. Collett, *Multiple stored views and landmark guidance in ants*, Nature,:392 (1998), pp. 710-714.

[26] O. Kawara, *On Kawara: date paintings in 89 cities*, Museum Boymans-Van Beuningen, Rotterdam, 1992.

[27] D. Lambrinos, R. Moller, T. Labhart, R. Pfeifer and R. Wehner, *A mobile robot employing insect strategies for navigation*, Robotics and Autonmous Systems,:30 (2000), pp. 39-64.

[28] M. Lamming and M. Flynn, *"Forget-me-not" Intimate Computing in Support of Human Memory*,,: (1994), pp.

[29] H. Lieberman and D. Maulsby, *Instructible Agents: Software that just keeps getting better*, IBM Systems Journal, 35:3&4 (1996), pp. 539-556.

[30] T. Lin and H.-J. Zhang, *Automatic Video Scene Extraction by Shot Grouping*, International Conference on Pattern Recognition, 4 (2000), pp.

[31] S. Mann, *Wearable Computing: A First Step Toward Personal Imaging*, Computer, 30:2 (1997), pp.

[32] F. Mindru, T. Moons and L. v. Gool, *Recognizing color patterns irrespective of viewpoint and illumination*, CVPR99,: (1999), pp. 368-373.

[33] M. Minsky, *A Framework for Representing Knowledge*, in J. Haugeland, ed. eds., *Mind Design II*, MIT Press, London, 1974, pp. 111-142.

[34] M. Minsky, *The Society of Mind*, Simon & Schuster, New York, 1985.

[35] B. Moghaddam and A. Pentland, *Face Recognition using View-Based and Modular Eigenspaces*, SPIE, 2277:July (1994), pp.

[36] J. Orwant, Doppelganger Goes To School: Machine Learning for User Modeling, 1993, M.Sc., Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge

[37] D. O'Shaugnessy, *Speech Communication: Human and Machine*, Addison-Wesley,, 1987.

[38] A. Pentland, R. Picard and S. Sclaroff, *Photobook: Tools for Content-Based Manipulation of Image Databases*, SPIE Paper 2185-05, Storage and Retrieval of Image & Video Databases II,: (1994), pp. 34-47.

[39] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall,, 1993.

[40] B. Rhodes, Context-Aware Computing, 1999, http://www.media.mit.edu/wearables/lizzy/context.html

[41] B. J. Rhodes, Just-In-Time Information Retrieval, 2000, Ph.D., Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge

[42] B. Ronacher, *How do bees learn and recognize visual patterns?*, Biological Cybernetics, 79 (1998), pp. 477-485.
[43] N. Saint-Arnaud, Classification of Sound Textures, 1995, M.S.,, Massachusetts Institute of Technology,
[44] B. Schiele and A. Pentland, *Attentional Objects for Visual Context Understanding*, ISWC'99,: (1999), pp.
[45] B. Schilit, N. Adams and R. Want, *Context-Aware Computing Applications*, Proceedings of Mobile Computing Systems and Applications,: (1992), pp. 85-90.
[46] B. N. Schilit, N. Adams, R. Gold, M. Tso and R. Want, *The PARCTAB Mobile Computing System*, Proceedings of the Fourth Workshop on Workstation Operating Systems (WWOS-IV),: (1993), pp. 34-39.
[47] I. Sethi, V. Salari and S. Vemuri, *Image sequence segmentation using motion coherence*, in, ed. eds., *Proceedings of the First International Conference on Computer Vision*,, London, England, 1987, pp. 667-671.
[48] I. K. Sethi and N. V. Patel, *A Statistical Approach to Scene Change Detection*, Storage and Retrieval for Image and Video Databases,: (1995), pp. 329-338.
[49] R. N. Shepard, *Attention and the metric structure of the stimulus space*, Journal of Mathematical Psychology, 1 (1964), pp. 54-87.
[50] T. Starner, B. Schiele and A. Pentland, *Visual Contextual Awareness in Wearable Computing*, Second International Symposium on Wearable Computers,: (1998), pp. 50-57.
[51] Y. Sumi, T. Etani, S. Fels, N. Simonet, K. Kobayashi and K. Mase, *C-MAP: Building a Context-Aware Mobile Assistant for Exhibition Tours*, in, ed. eds.,, ATR Media Integration & Communications Research Laboratories, Kyoto, Japan, 1998, pp.
[52] M. Turk and A. Pentland, *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience, 3:March (1991), pp. 71-86.
[53] P. Viola and M. Jones, *Rapid Object Detection using Boosted Cascade of Simple Features*, Computer Vision and Pattern Recognition,: (2001), pp. 1-9.
[54] R. F. Wang and E. S. Spelke, *Human spatial representation: insights from animals*, TRENDS in Cognitive Sciences, 6:9 (2002), pp. 376-382.
[55] Z.-H. Wang and P. Kenny, *Speech Recognition in Non-Stationary Adverse Environments*, in, ed. eds., *ICASSP*,,, 1998, pp.
[56] Y. Zhao, *A Speaker-Independent Continuous Speech Recognition System using Continuous Mixture Gaussian Density HMM of Phoneme-Sized Units*, IEEE Transactions on Speech and Audio Processing, 1:3 (1993), pp. 345-361.
[57] D. Zhong, H. J. Zhang and S.-F. Chang, *Clustering Methods for video browsing and annotation*, SPIE,:Storage and Retrieval for Still Image and Video Databases IV (1996), pp.

First A. Author Biographies should be limited to one paragraph consisting of the following: sequentially ordered list of degrees, including years achieved; sequentially ordered places of employ concluding with current employment; association with any official journals or conferences; major professional and/or academic achievements, i.e., best paper awards, research grants, etc.; any publication information (number of papers and titles of books published); current research interests; association with any professional associations.

Second B. Author Jr. biography appears here. Degrees achieved followed by current employment are listed, plus any major academic achievements.

Third C. Author is a member of the IEEE and the IEEE Computer Society.