Life Patterns: structure from wearable sensors

by

Brian Patrick Clarkson

B.Sc., Mathematics, MIT 1997

B.Sc., Electrical Engineering and Computer Science, MIT 1997

M.Eng., Electrical Engineering and Computer Science, MIT 1999

Submitted to the Program of Media Arts and Sciences,

School of Architecture and Planning,

in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY IN MEDIA ARTS AND SCIENCES

at the

Massachusetts Institute of Technology

September 2002

Signature of Author_____

Program in Media Arts and Sciences
September 19, 2002

Certified by_____

Alex P. Pentland
Toshiba Professor of Media Arts and Sciences
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by_____

Andrew B. Lippman
Chair
Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Life Patterns: structure from wearable sensors

by

Brian Patrick Clarkson

Submitted to the Program of Media Arts and Sciences,

School of Architecture and Planning,

on September 19, 2002

in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

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.

Thesis Advisor: Alex P. Pentland

Title: Toshiba Professor of Media Arts and Sciences

Life Patterns: structure from wearable sensors

by

Brian Patrick Clarkson

Thesis Committee:

Advisor: _____

Prof. Alex P. Pentland
Toshiba Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis Reader: _____

Prof. Trevor Darrell
Dept. of Electrical Engineering & Computer Science
Massachusetts Institute of Technology

Thesis Reader: _____

Prof. Joseph Paradiso
Program in Media Arts and Sciences
Massachusetts Institute of Technology

Acknowledgements

5 years ago when I started working at the Media Lab as a part-time undergraduate researcher the capacity to produce the ideas in this thesis was not in me. Since then I have had the privilege of working with the best of the best.

My advisor, Prof. Alex Pentland, while wading through his ocean of responsibilities typically bestowed on overly successful academics always found the time to guide me and reassure me that my ideas were good ones. I am thankful to him.

For a graduate student, his fellow graduate students are the key influencers on his academic development. I owe my ideas on wearables to Thad Starner (my first office-mate). I owe my education of machine-learning and statistics to the long discussion and reading groups I've had with Sumit Basu, Tanzeem Choudhury, Tony Jebara and Ali Rehimi, and of course the wise tutelage of Tom Minkas and Kris Popat.

I thank Prof. Joseph Paradiso and Prof. Trevor Darrell (my readers) for their invaluable feedback on this document and the final preparation of this work.

I want to thank my mother, Anneliese Clarkson, for the high standards she raised me with. I thank her for every night she made me go to bed at 9pm but let me read for as long as I wished. I thank my dad, Thomas Clarkson, who is responsible for instilling me with my love for learning. My dad never got a fair chance to nurture his love for learning, hopefully he sees that this is what would have happened if he did.

Contents

List of Figures

Chapter 1: Introduction

Our memories are the unreliable denizens of our brains. From the habits we form over time to the vignettes that compose our dreams, memories are a sort of compression of our physical experiences. An elderly man with a century of experiences has nothing but his memory to prove to himself that he has lived a full life, unless he has strewn the signs of his life outside of his mind through family, friends, a house, picture albums, scents, diaries, and accomplishments. Thus everyone has an instinctive urge to capture experiences and preserve them before they fade away.

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, the user of an automatic diary while browsing the data in a linear fashion 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 amongst because otherwise every event looks new and unique.

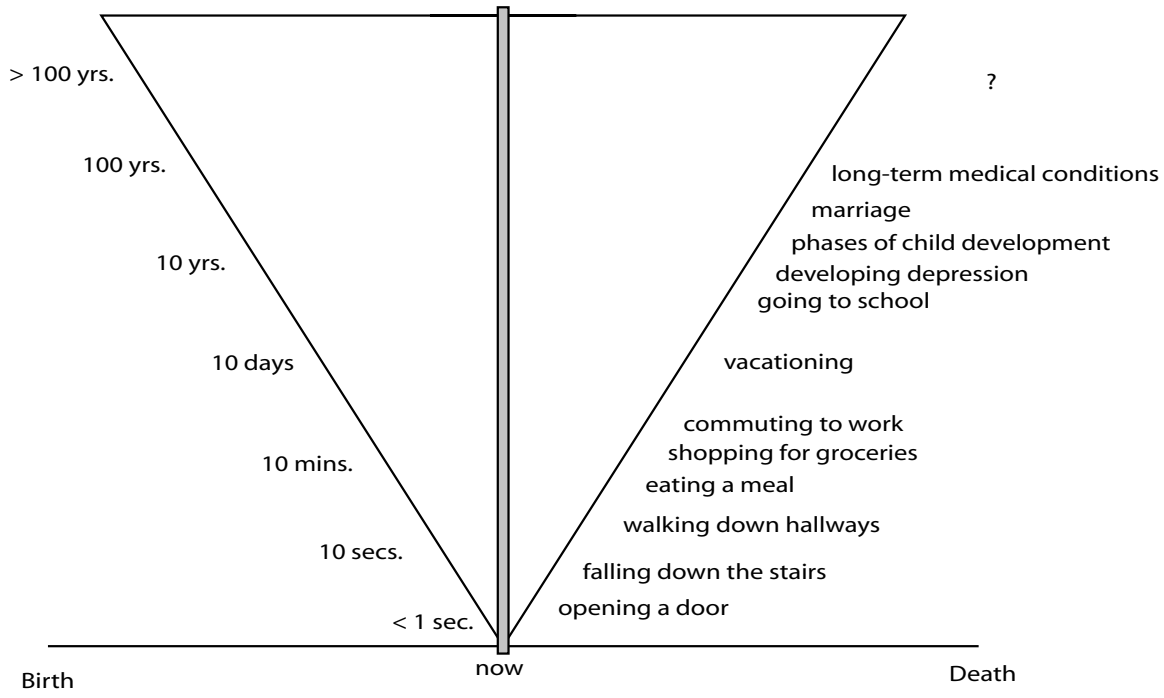


Figure 1-1: The different time-scales of life.

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 1-1). Strapping sensors on an individual and processing at the 1 second 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 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 computational power) to start considering modeling an individual's life at longer and longer time scales.

The main hypothesis of this thesis proposal is that consistent, repeatable structure exists in long-term wearable sensor data. We believe this because the sensor data comes from the patterns in an individual's life. The fact that we have common sense concepts like "daily routine" and "a typical day" are hints that a certain part of our life is repetitive or routine and hence an ideal candidate for statistical modeling. Philip Agre notes in his Ph.D. dissertation in the field of symbolic A.I., *The Dynamics of Everyday Life*:

> *"Everyday life is almost wholly routine, an intricate dance between someone who is trying to get something done and a fundamentally benign world that is continually shaped by the bustle of human activity." [1]*

Machine learning methods rely on repeatable patterns in the data in order to make statistical estimates in the presence of noise. While this structure, or life pattern, will naturally have many manifestations, we approach the extraction of life patterns from two directions.

This work tackles the question of how to recognize and predict a person's day-to-day behavior from visual, auditory, and motion sensor data. Perhaps the hints that cognitive scientists are extracting about how we organize our own wet memories could provide clues on how to do organize our machine memories. Using ideas from episodic memory organization and insect-level perception, we provide 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.

This document is laid out in a simple and causal fashion. In chapter 2, we situate this work amongst the current efforts of researchers in artificial intelligence, human-computer interaction and computational perception. In this context we motivate the goals of this thesis. In chapter 3, we summarize our earlier experiments on smaller data sets that highlight the basic methods in principles explored later on. In chapter 4 we describe the conditions under which we collected 100 days of experiential data from wearable sensors. Chapter 5 is the meat-and-potatoes section upon which the rest of the work lies. Here we detail the basic machinery that allows us to draw comparisons between and align exemplars of experience. Chapter 6 exhibits our ability to classify the exemplars into situations, then in chapter 7 we show how to extract the grammar of experience and use it to predict future situations or detect unusual ones.

Chapter 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:

> *"A record if it is to be useful to science, must be continuously extended, it must be stored, and above all it must be consulted. Today we make the record conventionally by writing and photography, followed by printing; but we also record on film, on wax disks, and on magnetic wires. Even if utterly new recording procedures do not appear, these present ones are certainly in the process of modification and extension.*
>
> *Certainly progress in photography is not going to stop. Faster material and lenses, more automatic cameras, finer-grained sensitive compounds to allow an extension of the minicamera idea, are all imminent. Let us project this trend ahead to a logical, if not inevitable, outcome. The camera hound of the future wears on his forehead a lump a little larger than a walnut. It takes pictures 3 millimeters square, later to be projected or enlarged, which after all involves only a factor of 10 beyond present practice. The lens is of universal focus, down to any distance accommodated by the unaided eye, simply because it is of short focal length. There is a built-in photocell on the walnut such as we now have on at least one camera, which automatically adjusts exposure for a wide range of illumination. There is film in the walnut for a hundred exposures, and the spring for operating its shutter and shifting its film is wound once for all when the film clip is inserted. It produces its result in full color. It may well be stereoscopic, and record with two spaced glass eyes, for*

striking improvements in stereoscopic technique are just around the corner.

The cord which trips its shutter may reach down a man's sleeve within easy reach of his fingers. A quick squeeze, and the picture is taken. On a pair of ordinary glasses is a square of fine lines near the top of one lens, where it is out of the way of ordinary vision. When an object appears in that square, it is lined up for its picture. As the scientist of the future moves about the laboratory or the field, every time he looks at something worthy of the record, he trips the shutter and in it goes, without even an audible click. Is this all fantastic? The only fantastic thing about it is the idea of making as many pictures as would result from its use." [6]

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 (see Figure 2-1). 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 an ever-increasing store of images obtained via a wearable camera.

Ben Akiba (1903) | Bloch's Detective Photo Scarf (1890) | Photoret Watch Camera (1894) | Ticka Pocket Watch Camera (1905)

Figure 2-1: Some of the earliest "wearable" cameras.

Vannevar Bush further describes a device, the memex, a concept that is well known to us:

> *Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, "memex" will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory. [6]*

This device seems somehow separated or unsuited for the recording of our visceral experiences, because he describes it as a device for recording "*books, records, and communications*". However, he next muses about intercepting what we sense here:

> *All our steps in creating or absorbing material of the record proceed through one of the senses -- the tactile when we touch keys, the oral when we speak or listen, the visual when we read. Is it not possible that some day the path may be established more directly? [6]*

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). However, let's take a look at Bush's last comment in his article (and this is the last time I will quote from him):

> *Presumably man's spirit should be elevated if he can better review his shady past and analyze more completely and objectively his present problems. [6]*

Many have interpreted Bush's word as referring to humanity in general, but what if we interpret them as referring to 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 WWW epidemic. However, no one has found a way to 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. Jon Orwant has addressed some of these issues as they apply to the graphical user-interface in his Doppelganger system [36].

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:

> *"Unlike desktop computers, wearable computers have the potential to ``see'' as the user sees, ``hear'' as the user hears, and experience the life of the user in a ``first-person'' sense. They can sense the user's physical environment much more completely than previously possible, and in many more situations. This makes them excellent platforms for applications where the computer is working even when you aren't giving explicit commands. Health monitors, communications systems, just-in-time information systems, and applications that control realworld devices for you are all examples of these contextually aware / agent applications. Wearables also need these new kinds of applications more than desktop computers do. When sitting at a desktop computer you can expect your user to be interacting with the screen directly. The user's primary task is working with the computer. With wearables, most of the time the user is doing something besides interacting with the computer. They might be crossing the street, or enganged in conversation, or fixing a boeing 777 jet engine. In most cases the wearable is there in a support role at best, and may even be an active distraction from the user's primary task. In these situations the computer can't rely on the user to tell it everything to do, and so it needs information from the wearer's environment. For example, imagine an interface which is aware of the user's location: while being in the subway, the system might alert him with a spoken summary of an e-*

mail. However, during a conversation the wearable computer may present the name of a potential caller unobtrusively in the user's head-up display, or simply forward the call to voicemail." [40]

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.

If we record 10 Hz video at 640x480 with JPEG compression and audio (16kHz, 16bit and no compression) then we can expect 1 day to require about 5GB, 1 year → 1.8TB, 10 years → 18TB, and 100 years → 180TB. Based on current trends in storage density we can expect to fit a whole life time of video in 1 sq. in. by the year 2040 (refer to Figure 2-2).

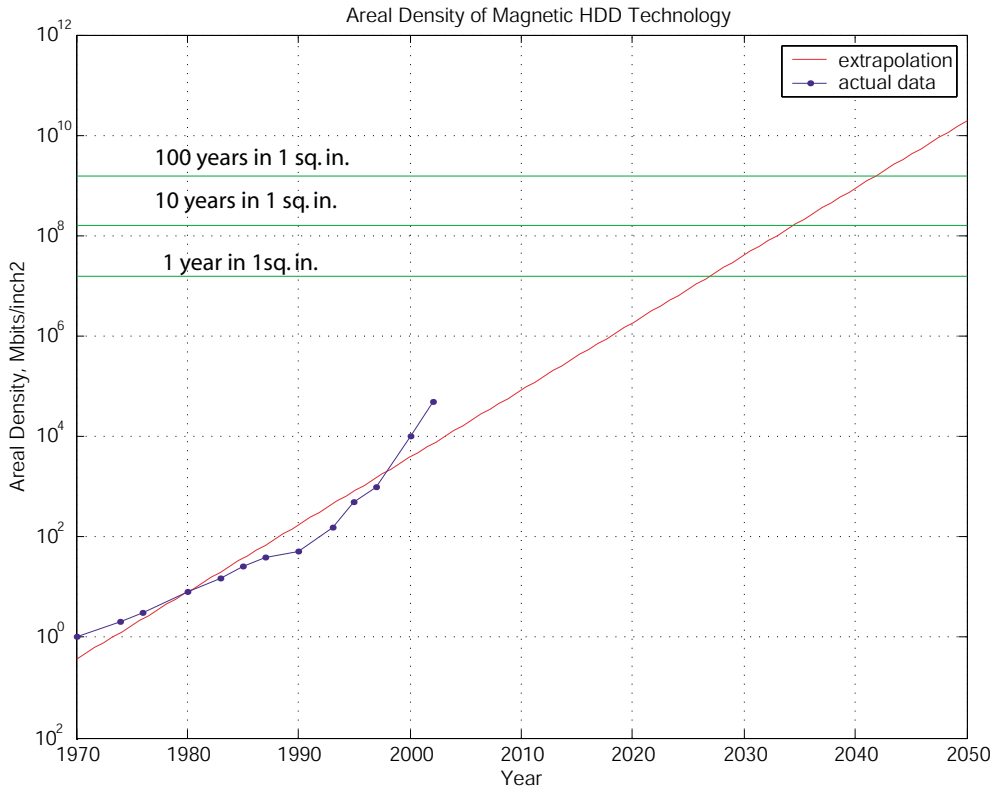


Figure 2-2: Extrapolation of the storage density of magnetic medium. We show the data storage requirements for recording an individual's life for 1 year, 10 years and 100 years based on the storage requirements for the I Sensed data set. (Data source: IBM 2002)

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. Frames provide stereotypical, but perhaps imprecise, knowledge. The system-level purpose of these frames is to provide relevant context and domain-specific knowledge to a context-dependent problem-solver. The expectations that these frames encode are largely considered to be derived from experience. Thus if in the process of completing a task (e.g. making a sandwich) a specific problem comes up (e.g. where is the mayonnaise), then the currently active frame can be consulted. Frames generally encode the expected solution of a query (e.g. the mayonnaise is expected to be in the refrigerator, based on past experience) that is only usually correct. However, at least the agent can continue completing the task until an actual problem or inconsistency occurs (e.g. you are out of mayonnaise). This avoids lengthy and likely irrelevant pondering over solutions to possibly non-existent problems.

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.

While this is an entirely valid and scientific approach to doing research on perceptual systems, it still remains to explore the mechanism for combining say a number of these constrained modules into a system that can smoothly and robustly handle a large variety of situations. This is similar to the bottom-up structures of Minsky's Society of Mind [34], where simpler task-specific agents are combined to create more complex higher-level agents. This work addresses the issue of building a perceptual system that can serve as the higher-level glue between more constrained task-specific perceptual agents, such as speech and face recognizers, or head and gesture tracking systems.

A specific example of this would be as follows. Say we had available to us a speech recognition module that only gave reliable results when the vocabulary and auditory front-end processing matched the input conditions. Recognizing when to switch between different frames/schemata, thus affecting vocabularies or auditory front-ends, can be the task of a higher-level agent that is responsible for operating correctly in a large variety of contexts (maybe all of them), not just one. However, this higher-level agent has the same inputs as the speech recognizer (e.g. microphone and camera), but the nature of its task, recognizing situation, requires that it operate in larger variety of contexts than the speech recognizer. This work provides a framework for finding and recognizing the physical contexts that make up a person's life.

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]. Figure 2-3 portrays the *snapshot* model of Lambrinos et. al. The *snapshot* model hypothesizes that the desert ant stores snapshots of the landmark locations in from various viewpoints around the site. Then displacement vectors are calculated using a weighted combination of contributions from each landmark based on apparent difference in size and bearing between the current landmark appearance and the remembered landmark appearance. In Figure 2-4, a computational pipeline for extracting the landmark features for use in the *snapshot* model is given. The key point here is that the desert ant is able recognize previously visited locations and even navigate towards them using very simple (almost pixel-based) features extracted from the visual scene. Throughout the perceptual pipeline of this work I also attempt to use insect-level complexity in choice of features and processing, avoiding the complexity of detailed visual and auditory scene analysis, which generally firsts attempts to map scene constituents onto semantically relevant concepts.

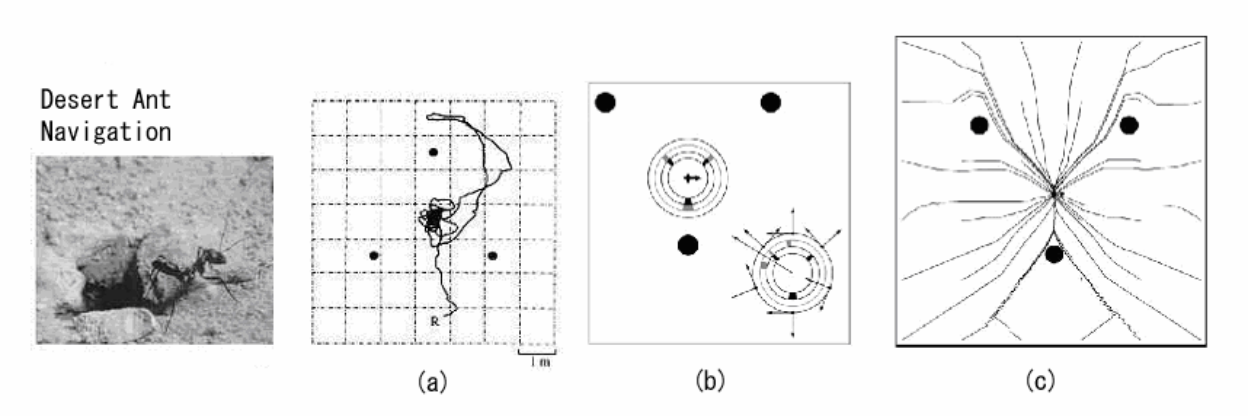


Figure 2-3: The desert ant finds its home by orienting on previously remembered landmarks, (a) shows a typical path that desert ant takes when it has been trained to find its nest at the center of 3 columns, (b) shows the naviagation vectors calculated from 2 example views of the landmarks using Lambrinos snapshot model, (c) trajectories generated by the snapshot model. (figures are reproduced from [27]**)**

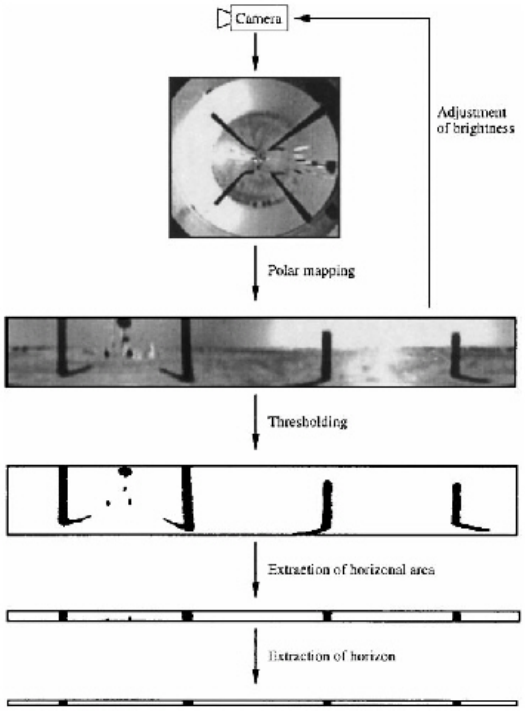


Figure 2-4: The snapshot model's feature pipeline used by Lambrinos et. al. to construct a navigational robot. Starting from an omnidirectional view, then thresholding and masking to create a 1D bitmap specifying landmark features in radial coordinates around the robot. (figures reproduced from [27]**).**

Chapter 3: Earlier Experiments

During the infancy of this work our vague notions of how to proceed at the large scale needed guidance from some experiments done on a smaller scale. At the time we were considering what it would take to build a truly personal and reactive computer system, useful to its user all day long not just while doing specific tasks. We concluded that a basic requirement for such a system is that it has access to the same kind of information as its user, for example the ambient sights and sounds in the users environment. Also at the time computers were rapidly evolving from large desktop systems to miniature, wearable devices. The wearable computer pioneers were calling for systems that live in the same sensory world as its human user. [50]

However, there was no previous work on how tractable data recorded from an individual's day-to-day behavior is. What kinds of things could classifiers be reliably trained for and used to classify the sensor data? What sensors are necessary? At what time-scale should the analysis take place? In this intellectual environment we took a microphone and mounted it on the author's shoulder, an Elmo matchstick camera with its lens replaced with a door-peeper[*] mounted on the author's backpack pointing backwards and wore them for a day.

3.1 Unsupervised Clustering of Wearable Sensor Data

We figured that if we could cluster this data and the clusters corresponded to semantically relevant concepts then we would have a good idea of the feasibility of the modeling task. So, once the data was collected we extracted features from the audio and video with the

[*] We used the 180-degree field-of-view kind installed in hotel doors that let you see who is at your door even if they are trying to hide against the wall.

goal of clustering in mind. This meant global, coarse features that measure the visual and aural ambiance of the situation but gloss over the specific contents the video and audio.

3.1.1 The Features

The visual features were the parameters of 9 Gaussians fit to the luminance and two chrominance channels of the video in 9 regions. The regions were manually determined based on qualitative evaluation of the typical optical flow patterns of the video, and an eye towards roughly separating the foreground and background pixels. The 9 regions are shown in Figure 3-1. This resulted in 81 features (9 Gaussians each with 9 parameters: 3 mean plus 6 covariance). The means of the Gaussians measure the overall color and illumination in each region while the covariance features represent the shape of the color distribution.

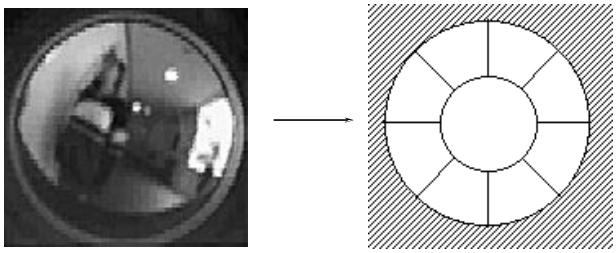


Figure 3-1: The 9 regions used to aggregate the visual features.

The auditory features were 25 filter-banks arranged on a Mel-scale in frequency to match the capabilities of humans to discriminate adjacent frequencies. [37] Both sets of features, audio and video, were calculated at a rate of 10Hz.

3.1.2 Time Series Clustering

The algorithm we used to cluster time series data is a variation on the Segmental K-Means algorithm [39] [24]. This algorithm is typically used during the training pipeline of a continuous speech recognition system when time-aligned labels at the phonemic level are not available. It tries to align a given sequence of symbols to raw feature data. In our case we don't even have the sequence of symbols so we have adapted the procedure as follows:

Given: N, the number of models, T the number of samples allocated to a state, S, the number of states per model, f the expected rate of class changes.

Initialization: Select N segments of the time series each of length T*S, spaced approximately 1/f apart. Initialize each of the N models with a segment, using linear state segmentation.

Segmentation: Compile the N current models into a fully connected grammar. A nonzero transition connects the final state of every model to the initial state of every model. Using this network, re-segment the cluster membership for each model.

Training: Estimate the new model parameters using the Forward-Backward algorithm on the segments from step 3. Iterate on the current segmentation until the models converge in likelihood and then go back to step 3 to re-segment. Repeat steps 3 and 4 until the segmentation converges.

3.1.3 Time Hierarchy

Varying the frame-state allocation number directs the clustering algorithm to model the time-series at varying time scales. In the *Initialization* step, this time scale is made explicit by T, the frame-state allocation number, so that each model begins by literally modeling S*T samples. Of course, the re-estimation steps adaptively change the window size of samples modeled by each HMM. However, since EM is a local optimization the time scale will typically not change drastically from the initialization. Hence, by increasing the frame-state allocation we can build a hierarchy of HMMs where each level of the hierarchy has a coarser time scale than the one below it.

3.1.4 Representation Hierarchy

There are still important structures that just clustering at different time scales will not capture. For example, suppose we wanted a model for a supermarket visit, or a walk down a busy street. As it stands, clustering will only separate specific events like supermarket music, cash register beeps, walking through aisles, for the supermarket, and cars passing, crosswalks, and sidewalks for the busy street. It will not capture the fact that

these events occur together to create scenes, such as the supermarket scene, or busy street scene[*].

We address this shortcoming by adapting a hierarchy of HMMs much a like a grammar. So beginning with a set of low-level HMMs, which we will call event HMMs (like phonemes), we can encode their relationships into scene HMMs (like words). The process is as follows:

Detect: By using the Forward algorithm with a sliding window of length $\forall t$, obtain the likelihood, $L_\lambda(t) = P(O_t,...,O_{t+\Delta t} \mid \lambda)$ for each object HMM, λ, at time, t.

Abstract: Construct a new feature space from these likelihoods,

$$F(t) = \begin{bmatrix} L_1(t) \\ \text{M} \\ L_N(t) \end{bmatrix}$$

Cluster: Now cluster the new feature space into scene HMMs using the time-series clustering algorithm.

For the event HMM layer, we constrained ourselves to left-right HMMs with no jumps and single Gaussian states. The reason for this is that we would like to restrict the HMMs to find distinct sequences that occur over and over again. Using an ergodic HMM here would cause the clustering to group sequences that might be permuted versions of each other but with the same overall dynamics. However, our concepts of scenes as collections of event sequences that happen together fits well with the modeling bias of an ergodic HMM, hence we use ergodic HMMs to represent scenes.

[*] Notice that simply increasing the time scale and model complexity to cover the typical supermarket visit is not feasible for the same reasons that speech is recognized at the phoneme and word level instead of at the sentence and paragraph level.

We evaluated our performance by noting the correlation between our emergent models and a human-generated transcription. Each cluster plays the role of a hypothesis. A hypothesis is verified when its indexing correlates highly with a ground truth labeling. Hypotheses that fail to correlate are ignored, but kept as ``garbage classes". (Hence, it is necessary to have more clusters than ``classes" in order to prevent the useful models from having to model everything.) In the following experiments we restricted the system to two levels of representation (i.e. a single object HMM layer and a single scene HMM layer). The time scales were varied from 3 secs to 100 secs for the object HMMs, but kept at 100 secs for the scene layer.

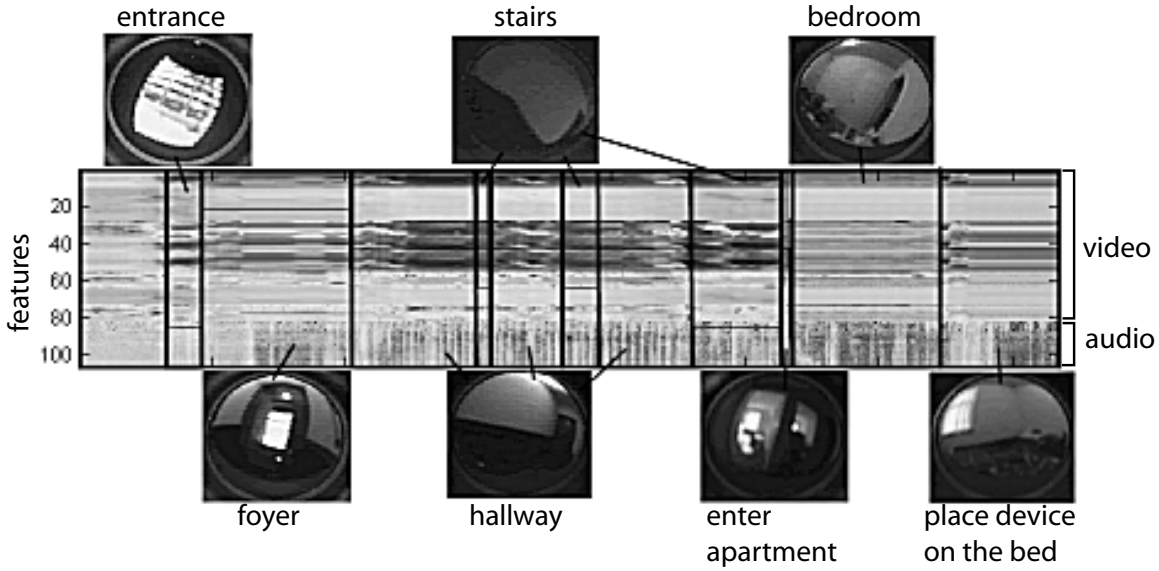


Figure 3-2: Coming Home: this example shows the user entering his apartment building, going up 3 stair cases and arriving in his bedroom. The vertical lines depict the system's segmentation. Manually selected key frames are shown with each segment.

Short Time Scale Object HMMs: In this case, we used a 3 sec time-scale for each object HMM and set the expected rate of class changes, *f*, to 30 secs. As a result, the HMMs modeled events such as doors, stairs, crosswalks, and so on. To show exactly how this worked, we give the specific example of the user arriving at his apartment building. This example is representative of the performance during other sequences of events. Figure 3-1 shows the features, segmentation, and key frames for the sequence of events in question. The middle plot represents the raw feature vectors (top 81 are video, bottom are

audio). Notice that you can even see the users steps in the audio section of the features (81-106).

Long Time-scale Object HMMs: Here we increase the time-scale of the object HMMs to 100 secs. The results are that HMMs model larger scale changes such as long walks down hallways and streets.

As a measure of the validity of our clustered HMMs we present the correlation coefficients between the independently hand-labeled ground truth and the output likelihood of the highest correlating model. The table below shows the classes that the system was ably to reliably model from only 2 hours of data:

Label	Correlation Coefficient
office	0.91
lobby	0.72
bedroom	0.86
cashier	0.83

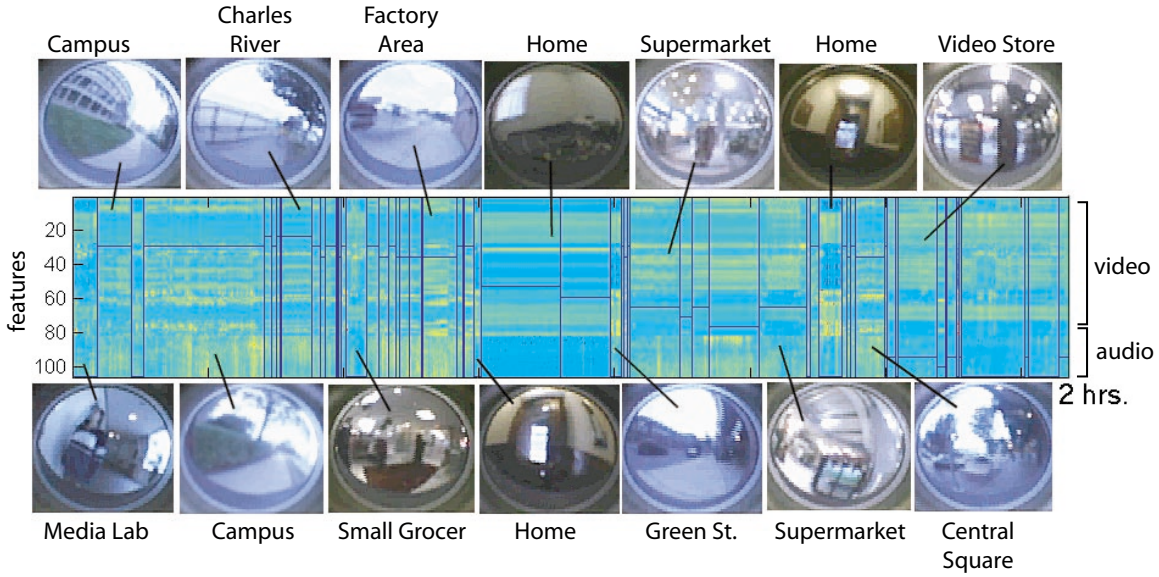


Figure 3-3: The clustering result at the scene level.

Long Time-scale Scene HMMs: We also constructed a layer of scene HMMs that are based on the outputs of the Short Time-scale Object HMMs from above. Where before we were unable to clean classes for more complex events, like the supermarket visit and walk down a busy street, now this level HMMs is able to capture them. The following table gives the correlations for the best models:

Label	Correlation Coefficient
dormitory	0.80
Charles River	0.70
factory area	0.75
sidewalk	0.78
video store	0.98

Figure 3-4 and Figure 3-5 show the likelihood traces for the models that correlated with ``walking down a sidewalk" and ``at the video store". While the video store and sidewalk scenes have elements that overlap with other scenes, their clustered models are able to select only the occurrences of their associated scenes. The fact these scenes could be clustered from data using HMMs is an indication that the concept of a scene corresponds in some manner to the statistical structure in the data.

It turns out that similarly scene-like clusters were not obtainable if we just use simple K-means clustering that ignores temporal statistics. Furthermore, trying different features such as color histograms, FFTs, image moments, etc. all led to basically the same results. This leads us to deduce two important characteristics of our task. First, it is the dynamics of the features that provides the discriminative power of our models. The supermarket is dissimilar from the video store not because they have different ambiance (in fact their time-averaged feature statistics are identical) but rather the subject behaves differently in each scene thus inducing different temporal dynamics. In the video store there is no long periods of no motion punctuated by motion as the subject moves from one wall of videos to the next. In the supermarket the subject is essentially always in motion. The second characteristic is that the pertinent scale of features is peripheral as opposed to attentive.

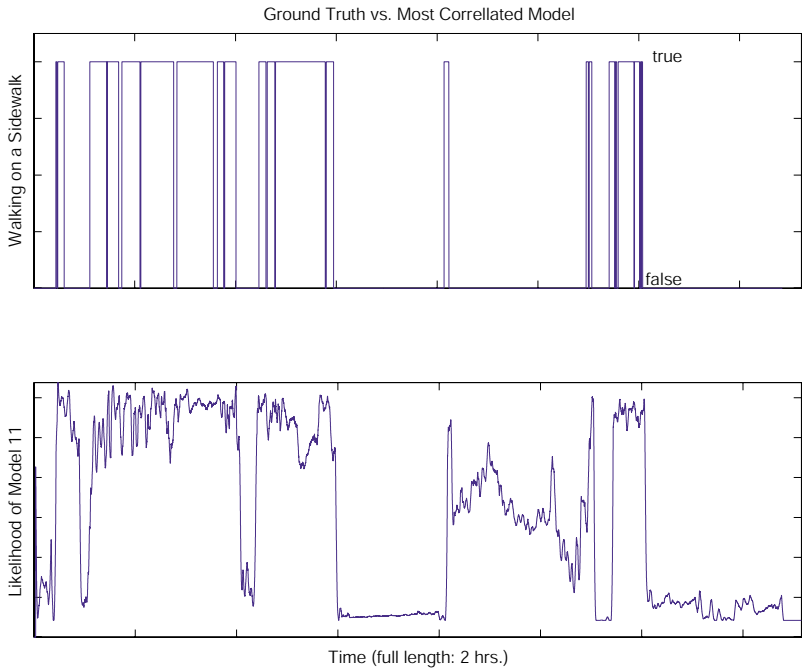


Figure 3-4: The ground truth and corresponding likelihood trace of the most correlated HMM for the sidewalk scene.

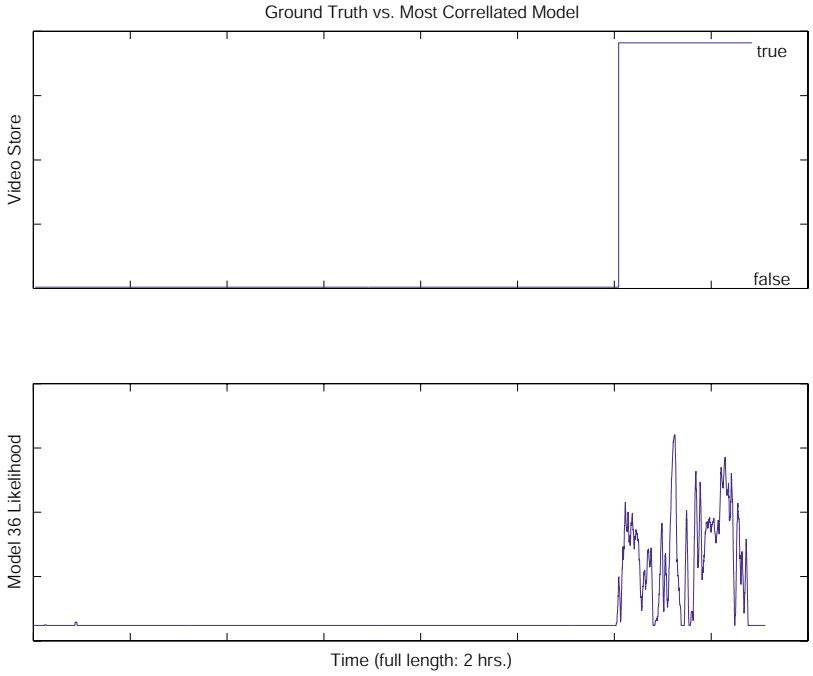


Figure 3-5: The ground truth and corresponding likelihood trace of the most correlated HMM for the vide store scene.

3.2 Capturing the Dynamics of Situations

Supervised methods are useful when the application at hand has a well-defined set of situation classes that must be accurately classified. Examples of each situation are provided and used to build models for automatically transcribing other examples of unknown situations. We now show that situation can be successfully classified by likelihood ratio tests with Hidden Markov Models (HMM) on color image moments features (visual ambience) and long-term auditory features (auditory ambience) collected from each situation. These features serve a similar role that the landmark features do for the desert ants in that they provide a noisy discriminatory function for matching places without context. However, differently from the ant, we do not have control over how the user moves, so an HMM is used to model the temporal variation of the situation's ambiance.

In these supervised learning experiments, we show that accurate situation classification with coarse features is possible and determine the complexity of the statistical model required for the task. The supervised learning of situation models worked well enough that we were guided to cluster the visual features and search for correspondences between the clusters and situation classes. The results were enticing and led us to the work you will see later on in the chapters to follow.

Figure 3-6: The wearable used in the experiments on situation dynamics and transitions. It is a primitive data collection system consisting of a pinhole camera mounted on a shoulder strap and a handheld mouse pad for data entry and labeling.

For the remaining experiments in this chapter we used the data collection wearable shown in Figure 3-6. This is a precursor to the more advanced wearable we will introduce later on for the I Sensed data set. The compute power and onboard storage is provided by a PII Sony Picturebook.

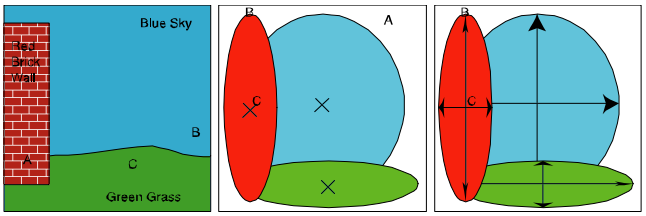
3.2.1 Video Feature Set

For the video images, we calculated spatial moments in each of the color channels, Y, Cb and, Cr. Mindru et. al. [32] has shown that image moments are useful for recognizing spatial patterns of colors while being robust to angle of approach. Thus we use them here too, however we will see later on that many types of features could be suitable for this task. The specific image moments used to build our visual features are:

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

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

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



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

The 4 types of image moments (as determined by the exponent of the pixel location) measure 3 aspects of the spatial pixel distribution: mass, geometric center, and geometric spread. For example the figure shows an abstract image with its dominating pixel distributions in each color channel. The red brick wall will tend to draw the geometric center (middle panel) of its color channel to the right and the last moment will measure how spread out or concentrated most of the energy in each channel is. The visual features were calculated at 5Hz.

3.2.2 Audio Feature Set

For the audio, we simply calculate a spectrogram using a 1024-pt FFT at a 15Hz frame-rate. The spectrogram was passed through a bank of 11 Mel-scale filters to map the linear frequency sampling of the FFT to the more perceptually relevant log-like scale. [37] This yields 11 coefficients per unit time. The resulting time sequence of spectral coefficients was then low-pass filtered with a single-pole IIR filter with a time constant of 0.4 seconds:

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

and subsequently sampled at 5Hz. A similarly low-passed filtered estimate of energy is also calculated for a grand total of 12 auditory features. These kind of coarse features for auditory scene analysis (ASA) are thematically similar to the features used in non-speech specific works such as [43] and [17]. The goal of such features is to capture the ambience or textural characteristics of the auditory environment. We had shown in [9] that long-term auditory characteristics tend to identify with distinct situations and that we can in fact detect scene changes by noting changes in the long-term spectral distribution (see Figure 3-7).

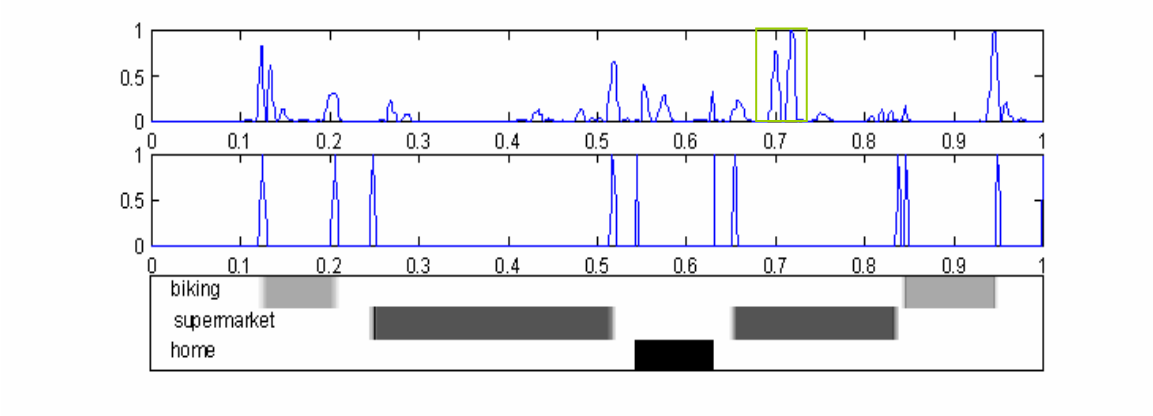
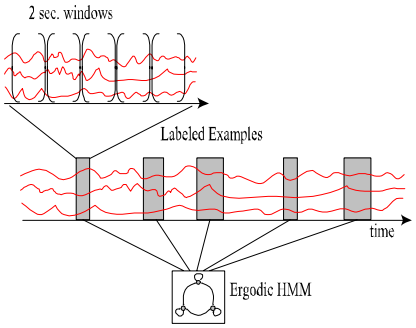


Figure 3-7: Scene change detection via auditory clustering (top), hand-labeled scene changes (middle), scene labels (bottom). The large error (marked in green) during the supermarket scene is due to noise from construction work happening on a section of the store interior.

3.2.3 Situation Dynamics

As our initial experiments into situation modeling indicated, we cannot discriminate sitatutions based solely on their static feature statistics. Many will appear the same from a time-average point of view. The user's motion and activity in the environment are both particular to a situation and combine to give a situation its own temporal signature. The ergodic HMM is a natural choice to model a temporal signature. Each situation class is assigned an HMM which is trained on a labeled set of data in a way that emphasizes the short-term dynamic texture of each location.



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

chosen, as opposed to say a left-right HMM, since we can expect highly variable dynamics at such a short time-scale. Using a more specific topology would imply that we know what kind of dynamics to expect in the situation features.

3.2.4 Model Evaluation

We now separately evaluate each model's accuracy of detecting and rejecting the location it was trained for. So, to determine for each time, *t*, the probability of a given location, the Forward-Backward algorithm was used to yield,

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

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

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

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

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

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

and that is by the Receiver-Operator Characteristic (ROC). The log likelihood threshold that achieves the Equal Error Rate (EER) point on the ROC curve is the value that should be mapped to a probability of 0.5. The mapping for the 2 remaining intervals, [0 0.5) and (0.5, 1], by histogramming the log likelihoods in each set and calculating the cumulative

distribution function (cdf). We took these 2 cdf's and concatenated them to produce one continuous mapping function, ϕ, that assigned all log likelihoods to [0, 1] with $\phi(l_{threshold}) = 0.5$. The effect of this mapping is to whiten the distribution of likelihoods observed for each model based on the training set. The result is a normalized score that mimics $P(y \mid x_{N-t}, ..., x_t)$, and is appropriate for inter-model comparison.

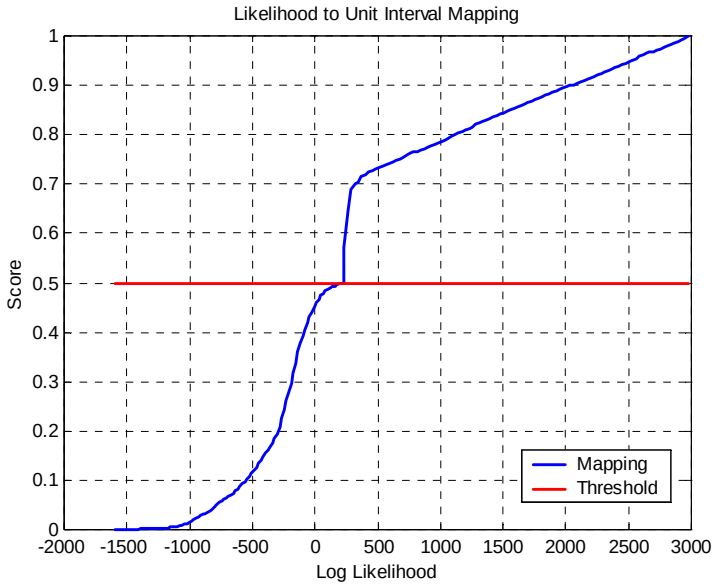


Figure 3-8: The histogram based mapping from raw log likelihood to a classification score.

	Accuracy (EER)		
Locations	A+V	A	V
BorgLab	95.9	56.2	97.1
BTLab	97.3	63.8	98.8
Courtyard	92.2	64.9	93.0
Elevator	99.8	58.0	98.4
Lower Atrium	95.7	88.7	87.3
Upper Atrium	95.0	56.3	96.0
Office	96.0	87.3	93.5

Table 3-1: EER results for correct detection of 7 situations using audio and video features together and separately.

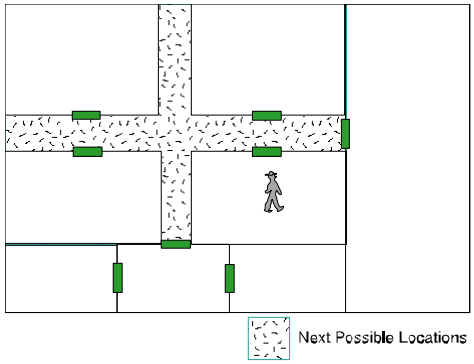
3.2.5 Results

Table 3-1 lists the recognition rates for situation classification at the equal error rate (EER). While video clearly outperforms the audio, it is intuitively pleasing to see that even audio by itself provides quite a bit of information for recognizing the situation. In fact in situations like the "Lower Atrium" the audio and video

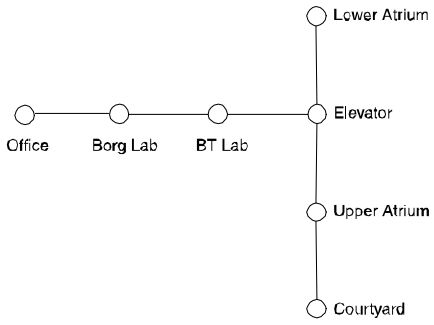
separately can only discriminate the situation less than 90% of the time. However, combining the two modalities gives nearly 96% accuracy. These results are based on equal weighting of the audio and video. Theoretically, Bayesian selection of these weights would prevent the reduction of performance by combining the audio and video modalities.

3.2.6 Temporal Constraints

Consecutive situations are typically related in a probabilistic manner. If our situations correspond with locations then there is a definite constraint on how one can transition from situation to the next.



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



3.3 Between Situations

The previous experiments discriminated situations by modeling the dynamics of the features during a given situation. However, in doing so we are ignoring another source of information, which is how we move from one situation to the next. The transition between to distinct situations can be much more distinctive then either situation itself. We test this by trying to recognize with high temporal accuracy the act of entering/leaving 3 separate locations:

1. Enter Office
2. Leave Office
3. Enter Kitchen
4. Leave Kitchen
5. Enter Black Couch Area (BCA)
6. Leave BCA

Please see Figure 3-9 for a map of the area. The 3 boxes refer to the areas that were labeled whenever the user entered or left them. The path connecting these areas is not an actual path but just an estimation of the usual route that the user took to get between these 3 areas.

Other than the selection of the 3 areas for labeling, the conditions of this experiment were quite freeform. No effort was made to control for spontaneous situations since we wished to collect data under the most natural conditions possible.

- Natural movement and posture
- Spontaneous conversations in the hallway
- Constant shifting of the sensor package on the user's body.

Basically, we want to emphasize that except for the user having a handheld device for labeling the exact moment when the transition occurred, all other conditions were kept as natural as possible.

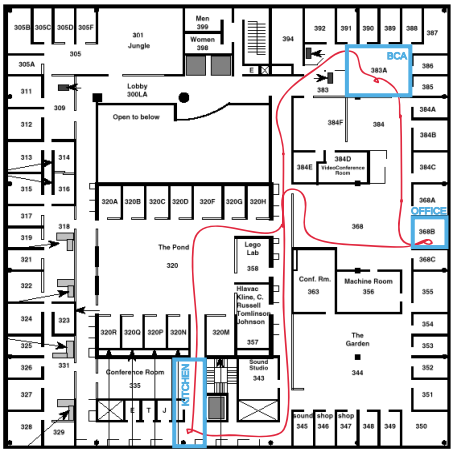


Figure 3-9: Location Map

EVENTS	# OF EXAMPLES
Leave Office	31
Enter Office	27
Leave BCA	21
Enter BCA	22
Leave Kitchen	21
Enter Kitchen	22

Table 3-2: The Data Set

The transitions (1-6) were labeled with impulses in time. For example when the user entered the kitchen, he marked the moment he passed through the doorway by pressing the label button on the handheld touch pad. See Table 3-2 for the number of labels collected for each event. For each of the events we partitioned the sets into separate training sets and testing sets.

3.3.1 The Models

The models used for determining the occurrence of events from the sensor stream were again ergodic HMMs. We trained an HMM on each of the six events separately. Classification was achieved by using the Viterbi algorithm to obtain an estimate of the event likelihood for a window of features. If the likelihood exceeded a threshold then the event was triggered.

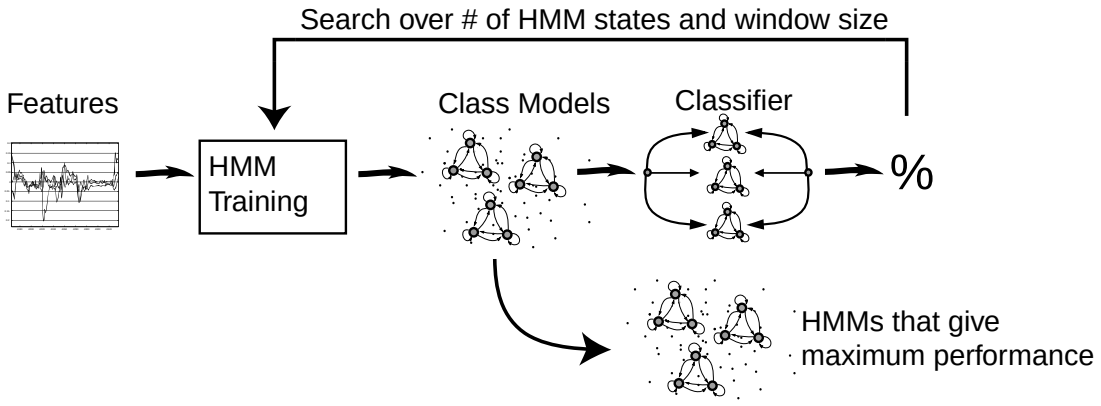


Figure 3-10 Overview of the Baseline System: 24 dimensional A/V features sampled at 5Hz enter the training pipeline at the left. HMMs are trained with varying numbers of states and window sizes. The HMMs that give maximum testing performance are selected.

To construct the training examples we took a time window of features around each of the impulse labels in the training set. This same window size was used in the Viterbi algorithm during classification. The window size represents the model's use of context, so that larger window sizes mean more context is taken into account. Each state in the HMMs were constrained to have a single Gaussian. However, this still leaves the number of states and the window size undetermined. Since we would like to know what the necessary complexity of model and length of time integration is required we decided to search for these parameters based on per-class performance.

We selected the parameters using brute force search over a range of state counts and window sizes. Using classification accuracy as the selection criterion, we iterated over state counts from 1-10 and window sizes from 2-20 secs. See Figure 3-10 for a flow diagram of the training procedure just described and Figure 3-11 for an example of a criterion surface.

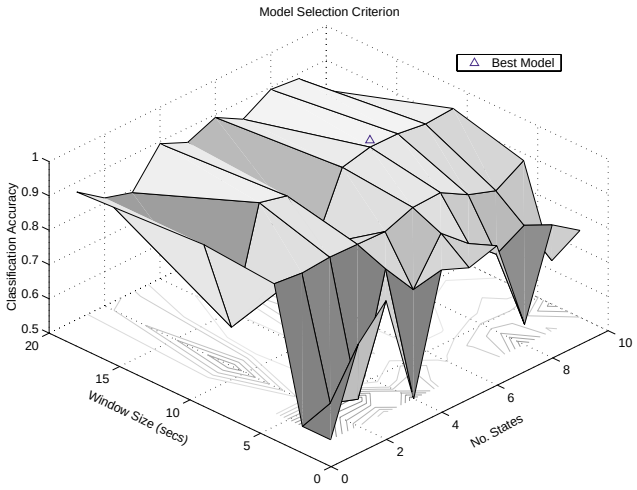


Figure 3-11 Accuracy vs. Free Model Parameters: this plot shows classification accuracy for the *Enter Kitchen* task for different HMM sizes and window sizes.

3.3.2 Results

We evaluated the models on a separate test set and obtained the following classification results. Since the thresholds for each model still needed to be determined we calculated Receiver-Operator Characteristic (ROC) curves for each model. The resulting curves are shown in Figure 3-12. We used the Equal Error Rate (EER) criterion (i.e. cost of false acceptance and of correct acceptance are the same) to choose optimal points on the ROC curves. Table 3-3 gives the resulting accuracies and the associated model parameters for each event.

Events	# of States	Window Size (secs)	Accuracy (%)
Leave Office	8	20	85.8
Enter Office	2	11	92.5
Leave BCA	3	20	92.6
Enter BCA	7	20	95.7
Leave Kitchen	1	4	99.7
Enter Kitchen	7	11	94.0

Table 3-3: The resulting model parameters and accuracies (based on EER) for each event/class.

The next plots (Figure 3-13, Figure 3-14, Figure 3-15, and Figure 3-16) give the actual likelihood traces for the best and worst performing event models overlaid with the ground truth labels. Although in both cases the likelihood traces are quite noisy, the peak

separation near actual labels is quite good (as supported by the ROC curves). *Leave Kitchen* (Figure 3-13) had the best performance with 99.7% accuracy achieved with just a 1 state HMM trained on 4 sec. feature windows. *Leave Office* (Figure 3-15) had the worst performance with 85.8% achieved with an 8 state HMM trained on 20 sec. feature windows. Notice that the window sizes exhibit themselves in the time resolution of the classifiers (Figure 3-14 and Figure 3-16).

These results are actually quite surprising considering the lack of context (at most only 20 seconds of features are used) and the coarse features. The classification accuracy is high in both acceptance and rejection. In fact the temporal resolution for some of the classes is pleasantly high (4 secs for the act of leaving the kitchen).

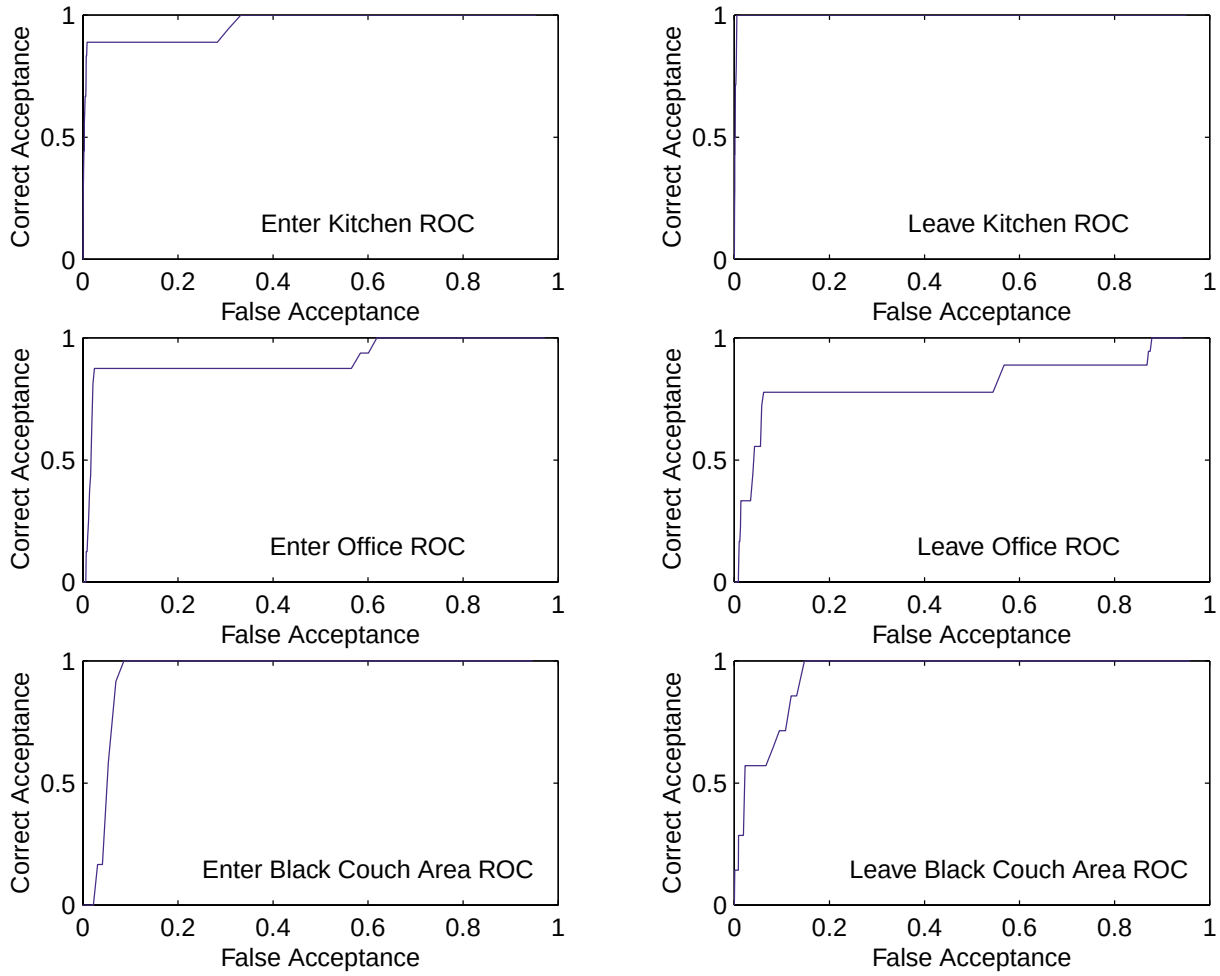


Figure 3-12 Receiver-Operator Characteristic (ROC) Curves for each model when tested on the test set and varying the threshold on the likelihood.

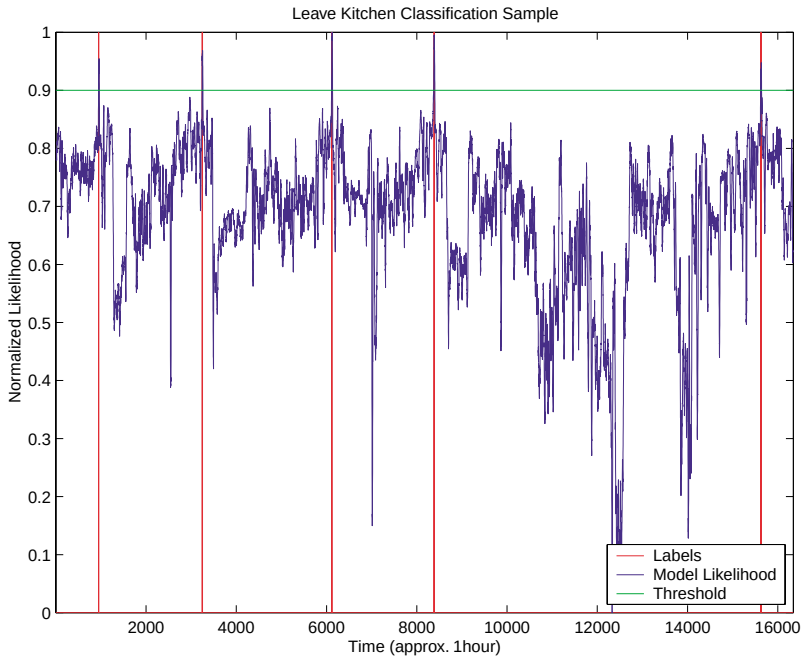


Figure 3-13 *Leave Kitchen* Classification: This class achieved the best classification performance of all the classes and it used a 1 state HMM trained on 4 sec feature windows. This figure shows approx. 1 hr. of performance on the test set.

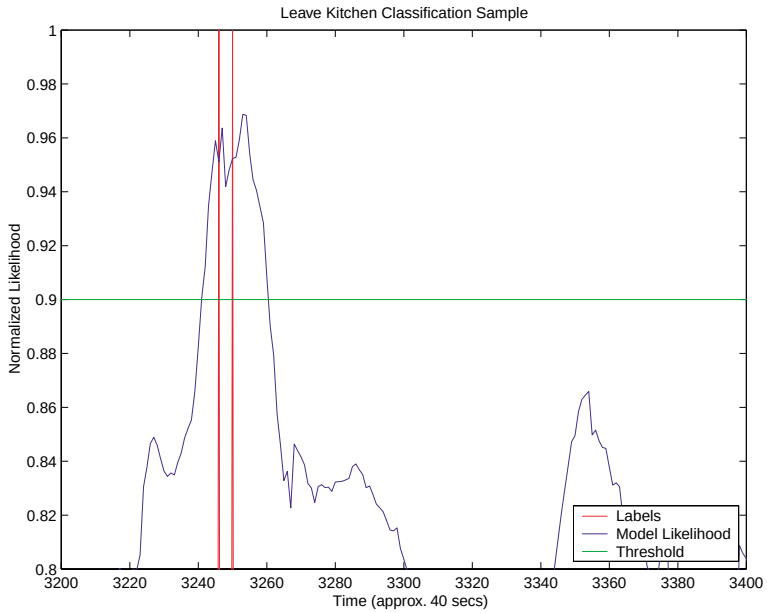


Figure 3-14 *Leave Kitchen* Classification (Zoom): This figure zooms in on a particular event in Figure 3-13. Notice the width of the likelihood spike is similar to the window size of this model (i.e. 4 secs).

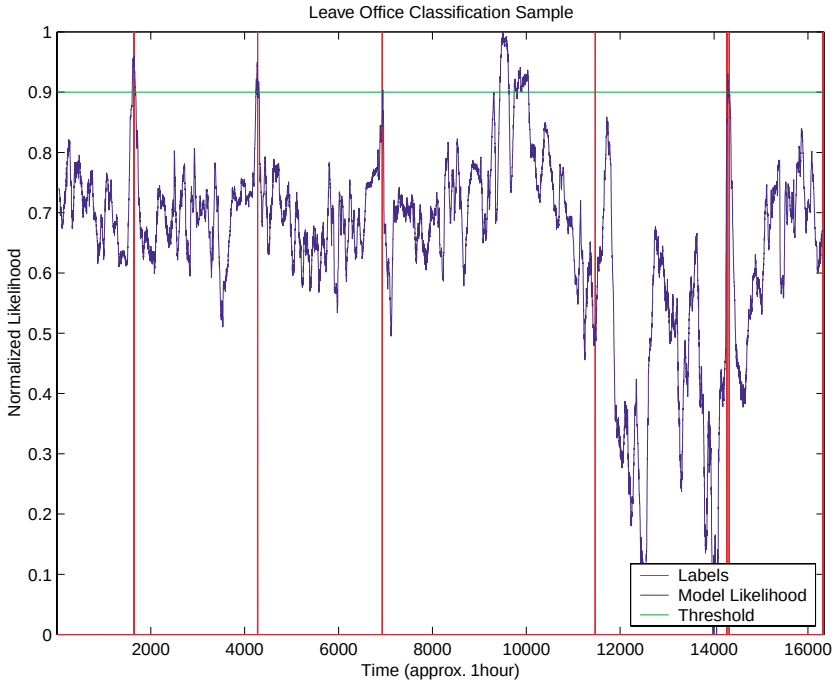


Figure 3-15 *Leave Office* Classification: This class achieved the worst classification performance of all the classes and it used a 8 state HMM trained on 20 sec feature windows. This figure shows approx. 1 hr. of performance on the test set.

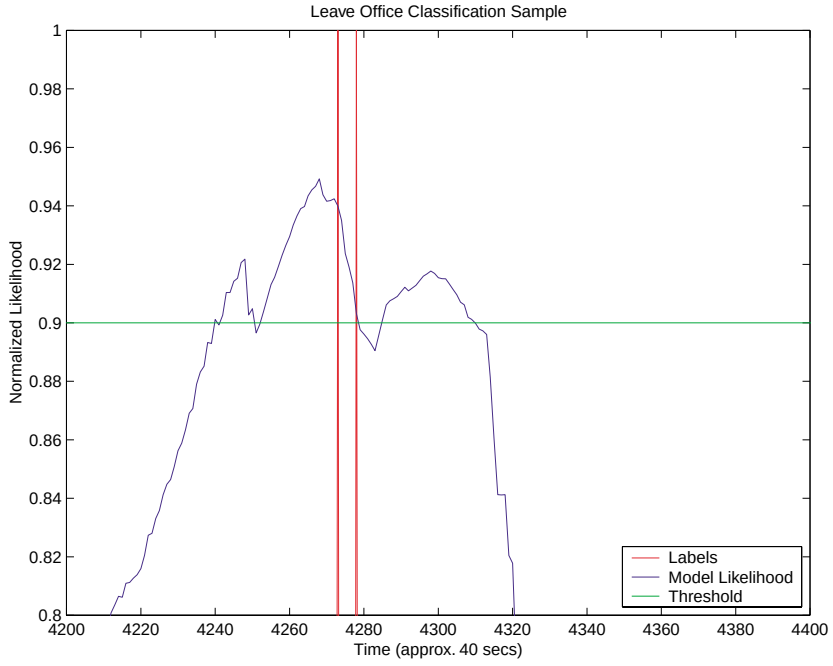


Figure 3-16 *Leave Office* Classification (Zoom): This figure zooms in on a particular event in Figure 3-15. Notice the width of the likelihood spike is similar to the window size of this model (i.e. 20 secs).

3.4 Discussion

We have now gone over a few small-scale experiments in clustering and recognizing a user's situation. We can believe now that coarse features that capture the global state of the surroundings are sufficient to discriminate situations. It seems complex methods for obtaining variance to lighting and orientation are not required. However, at this point considering the limited size of the data sets used, we can only claim these as good rules of thumb that have not been sufficiently validated. So, in the following experiments in this thesis we take these concepts and apply them to a much more extensive data set to search for the limit of the ability of statistical models to extract experiential structure from raw low-level sensory data.

Chapter 4: 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? These are big questions, which we cannot address completely. However, through a few guiding principles (which we describe next) we can limit our analyses to an appropriate level-of-detail, thus enabling us to reasonably tackle the above questions. Since these questions need hard data to produce answers, we also address how to collect measurements of an individual's experiences and describe how we overcame this hurdle.

When the detective tries to understand the mind of the criminal, he attempts to place himself in the criminal's state of mind, duplicating the experiences and encounters that the criminal might have had up to and including the scene of the crime. This makes it possible for the detective to infer the missing pieces of evidence and perhaps predict the criminal's next move. Mapping this intuition to the case of a computational agent (the detective) and its user (the criminal), means that we should provide the computational agent the same inputs the user is receiving so as to allow the agent to understand or habituate to the experiences and thus perhaps predict future experiences of its user. This implies that we use wearable sensors that are unobtrusively integrated into the user's clothing. Furthermore, we will concentrate our efforts on sensors that parallel biological perception: vision, audition, and vestibular. Lastly, we must be able to capture the subject's experiences for as long as is reasonably possible. First-person, long-term sensor data is a guiding principle for our overall approach.

The second principle guiding this work is the use of peripheral or context-free perceptual methods. The definition of a context-free method is an algorithm or system that is effective in any context, thus independent of lighting, background auditory conditions,

etc. Context-free methods generally rely on global features such as color histograms or optical flow in vision and spectrograms and peak tracking in the audio to provide useful descriptions of the raw sensor data. Thus if we have a speech detection module that operates robustly in most conditions, we can include it as a context-free perceptual method. Contrast this with the use of attentive or context-specific perceptual modules, such as today's state-of-the-art speech recognizers [55] [56] [7] [20] or face recognition systems [35] [53] that require knowledge of the current context in order to operate.

The third guiding principle of this work owes its inspiration to insect-level perception. 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. I will describe this data set and then describe in more detail methods for building coarse descriptions of the world, and thus a similarity metric. Last, we describe methods for extracting temporal models based on these coarse event descriptions.

4.1 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).

* The term "I Sensed" comes from a piece of historical conceptual art that has played a part in inspiring this thesis. In the 70's there was a Japanese conceptual artist named Kawara On [26] O. Kawara, *On Kawara: date paintings in 89 cities*, Museum Boymans-Van Beuningen, Rotterdam, 1992., who was in a way obsessed with time and the (usually) mundane events that mark its passage. His works such as the *I Met* and *I Went* series explored the kind of day-to-day events that tend to fall between the cracks of our memories. For years, everyday Mr. On would record the exact time he awoke on a postcard and send it to a friend or create lists of the people he met each day or trace on maps where he went each day. Other relevant works are his *I Got Up At*, *I Am Still Alive*, and the *I Read* series. His work raises a few interesting questions. If we had consistent records of some aspects of our day-to-day lives over a span of a lifetime, what trends could we find? What kinds of patterns or cycles would reveal themselves? Interestingly, we wouldn't need highly detailed memories to find these trends and patterns, just a consistent sampling in time. One of my dreams is to build a device that can capture these life patterns automatically and render them in a diary-like structure.

Figure 4-1: The Data Collection wearable when worn.

The wearable was worn from mid-April to mid-July of 2001 by the author. Refer to Figure 4-5 for actual excerpts from this data set during 4 example situations: eating lunch, walking up stairs, in a conversation, and rollerblading.

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.

Persons who normally interact with the subject on a day-to-day basis and have a possibility of having a potentially private conversation recorded are asked to sign a consent form in which we formally agree to not disclose recordings of them in anyway without further consent. This way my data collection experiments were in full accordance with the Massachusetts state laws on recording audio & video in public.

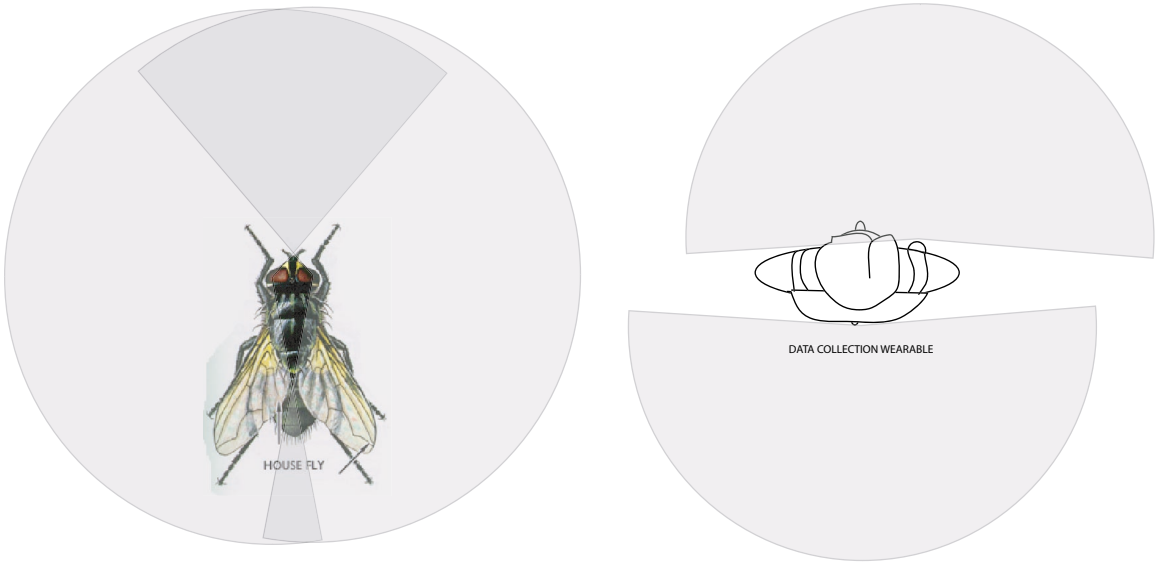


Figure 4-2: Comparison of the field-of-view for the common household fly and the data collection wearable used in the I Sensed series.

4.1.1 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 (see Figure 4-2). Other possibilities for sensors that have no good reason for being excluded are temperature, humidity, accelerometers, and bio-sensors (e.g. heart-rate, galvanic skin response, glucose levels). The properties of the 3 sensor modalities are as follows (see Figure 4-4):

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 Figure 4-1) 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.

4.1.2 The Data Journal

Organizing, accessing, and browsing such a large amount of video, audio, and gyro data is a non-trivial engineering task. So far we have a system that allows us to fully transcribe the “I Sensed” series and to access it arbitrarily in a multi-resolution and efficient manner. This ability is essential for learning and feature extraction techniques talked about later in this paper. All data (images, frames of audio, button presses, orientation vectors, etc.) are combined and time synchronized in our data journaling system to millisecond accuracy (see Figure 4-3).

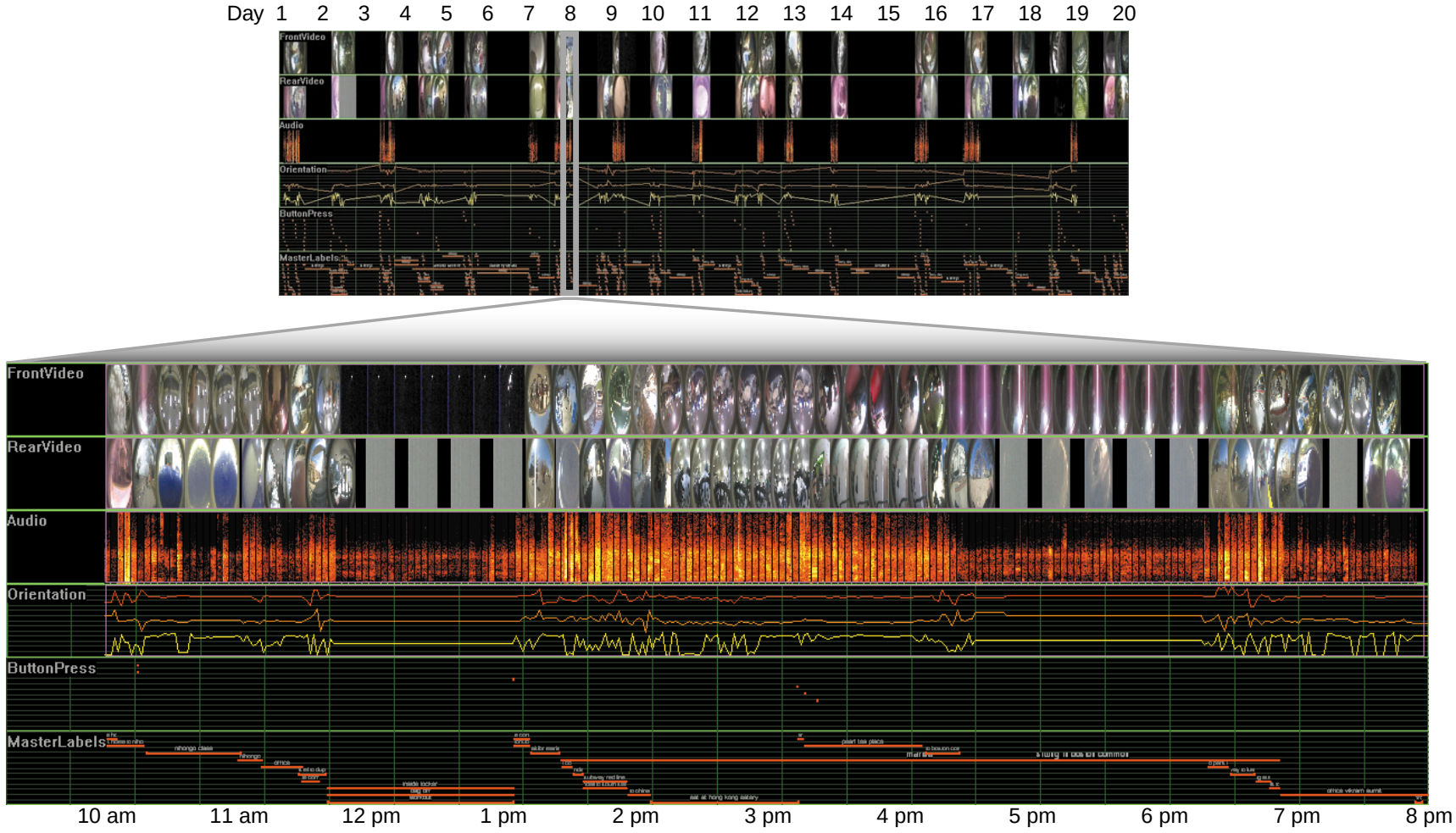


Figure 4-3: The Data Journal System: provides a multiresolution representation of the time-synchronized sensor data.

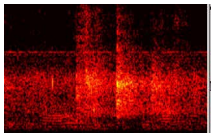
The Data Collection Wearable

Figure 4-4: The Data Collection Wearable Schematic

Rear View

Orientation

Front View

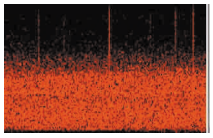


Audio Spectrogram

Scene 1: Eating Lunch

Scene 2: Walking Up Stairs

Front View



Audio Spectrogram

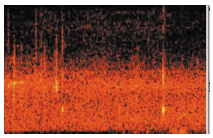
Rear View

Orientation

Rear View

Orientation

Front View

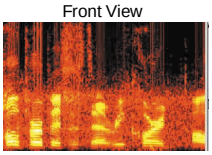


Audio Spectrogram

Scene 3: Rollerblading

Scene 4: In A Conversation

Front View



Audio Spectrogram

Rear View

Orientation

Figure 4-5: Some excerpts from the "I Sensed" series

Chapter 5: 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.

5.1 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.

There is an interesting side-note on the choice of the exponent in the Minkowski metric. Researchers in biological perception have noticed repeatedly that simple creatures such as insects (particularly bees) appear to use an L_1 norm on visual discrimination tasks, but as the creature gets more complex (say humans) they discretely switch between the L_1

and L_2 norms (see Figure 5-1 from [42] for more details.) depending on the preconceptions they bring to the task. There is also evidence of humans using the L_4 norm. A few researchers [18] [49] have even gone so far as to say these two metrics are indicative of two types of mental processing, city-block indicates "analytic" processing, while the Euclidean indicates "holistic" processing. A more practical suggestion is that the choice of metric is dependent on the how the features combine to produce the perceived input. [18] [2] Attributes that are perceived in aggregated form, such as the hue, saturation and brightness of a color, are appropriate for the Euclidean metric. Attributes that are separable, such as the amount of light falling on an array of photo-sensors, are ideal for the city-block metric.

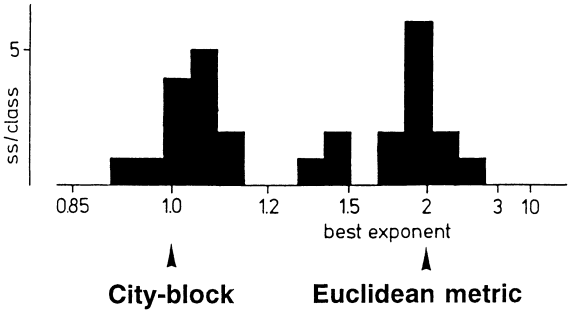


Figure 5-1: Histogram of the exponent of the Minkowski metric that best fit the performance of 27 human subjects on a size- and brightness-discrimination task. (reproduced from [42])

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

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

$x_{ij}(c)$ = 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 5-2: 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 5-2). 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 5-2). 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 5-3) will not affect many of the total peripheral pixels.

Figure 5-3: 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 eigenimages of the front and rear views, ordered by their contribution to reconstruction, is given in Figure 5-4 and Figure 5-5. Notice that the mean image contains the rim of the lens and a monochrome gradient consistent with idea that light usually comes from above. As is typical, the mean was removed from each image before PCA analysis. The first rear view's eigenimage is definitely the ghost of a chair back, which is a very common view to the rear. The third eigenimage of both the front and rear views could possible account for varying overhead

illumination. Eigenimages like rear view's #18 and #19, and, front view's #14 and #15 could be representing motion blurred images. However, the rest of the eigenimages have a distinct resemblance to the 2-D Fourier basis, ordered in frequency, as is expected for images with stationary statistics [16]. An interesting (but very computationally intensive in the training phase) alternative is to find the basis via independent components analysis (ICA). Bell and Sejnowski [5] finds that the independent components of natural gray-scale scenes (much like ours) are localized edge filters ordered by spatial location and orientation (i.e. oriented Gabor filters).

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 error (see Figure 5-6) 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 5-7 summarizes the feature extraction step for the alignment.

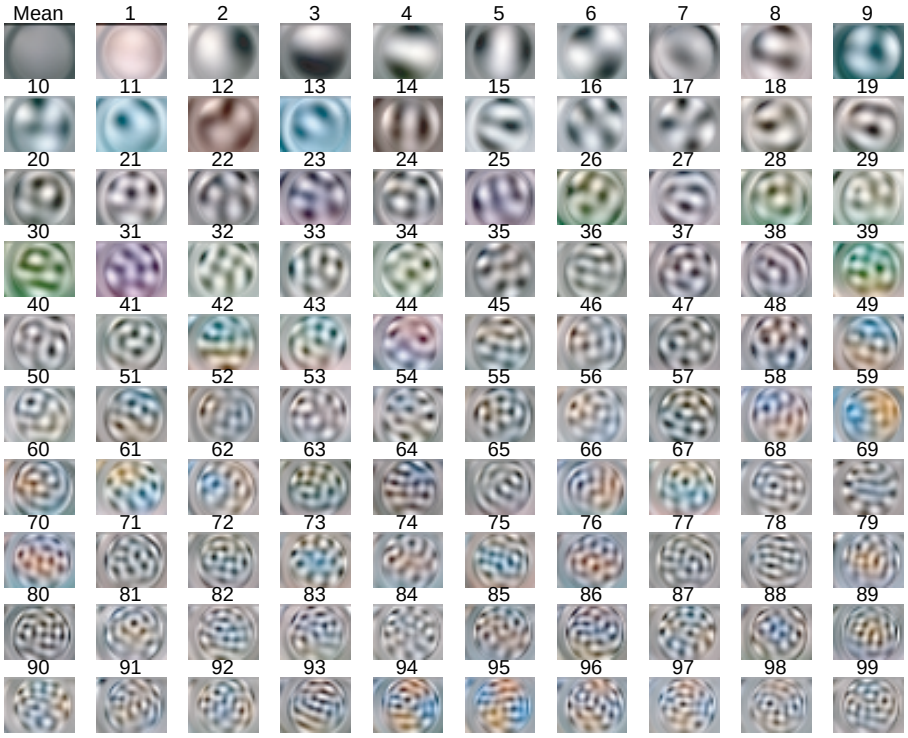


Figure 5-4: The mean and first 99 eigenvectors of the front view.

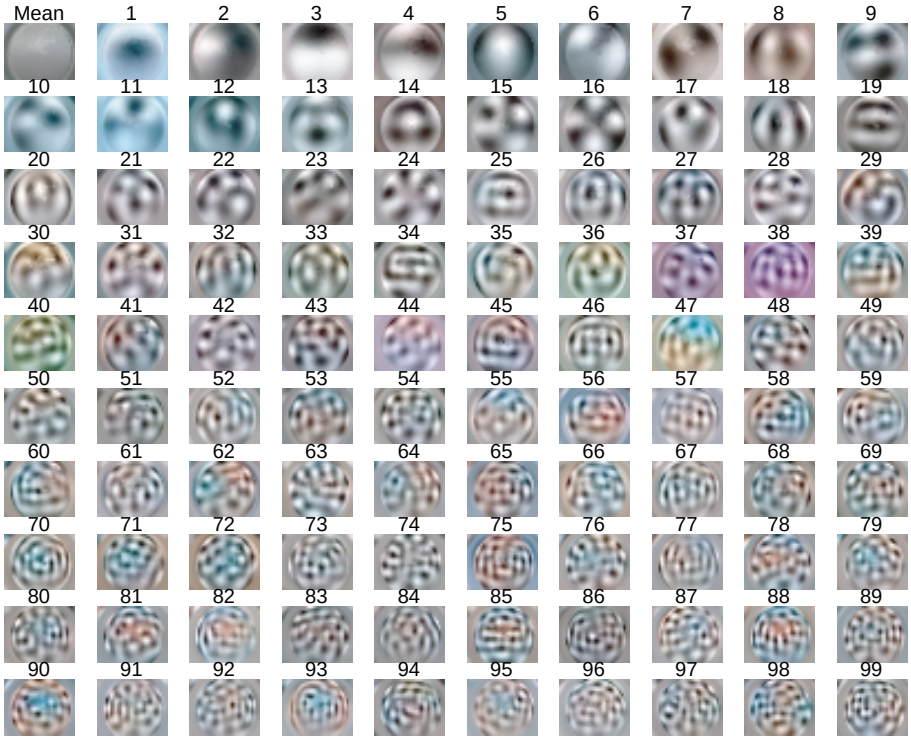


Figure 5-5: The mean and first 99 eigenvectors of the rear view.

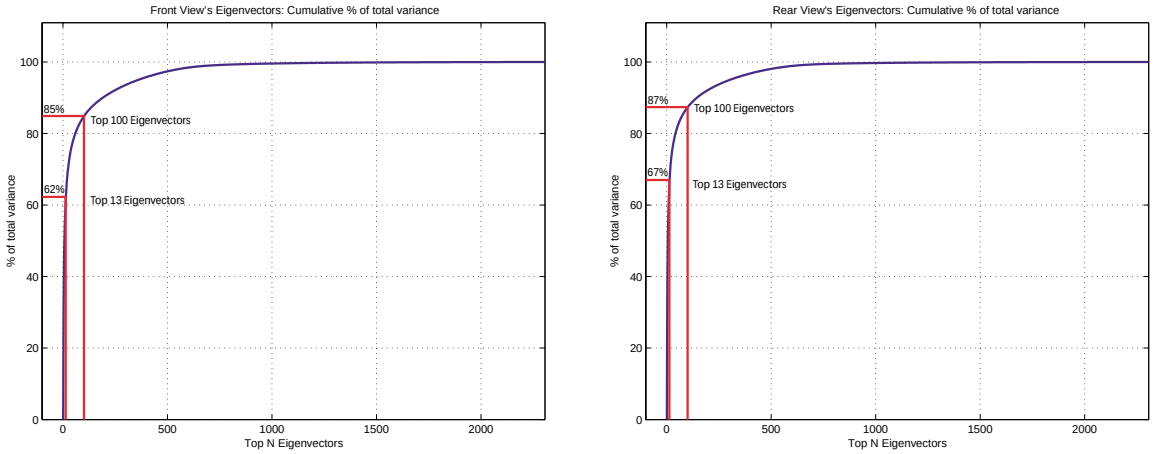


Figure 5-6: Percentage of the total variance captured by the top N eigenvectors (ordered by their eigenvalues). These curves show that these image space's are actually quite difficult to compress with just PCA. At least 400 eigenvector's are needed to reach the 95th percentile.

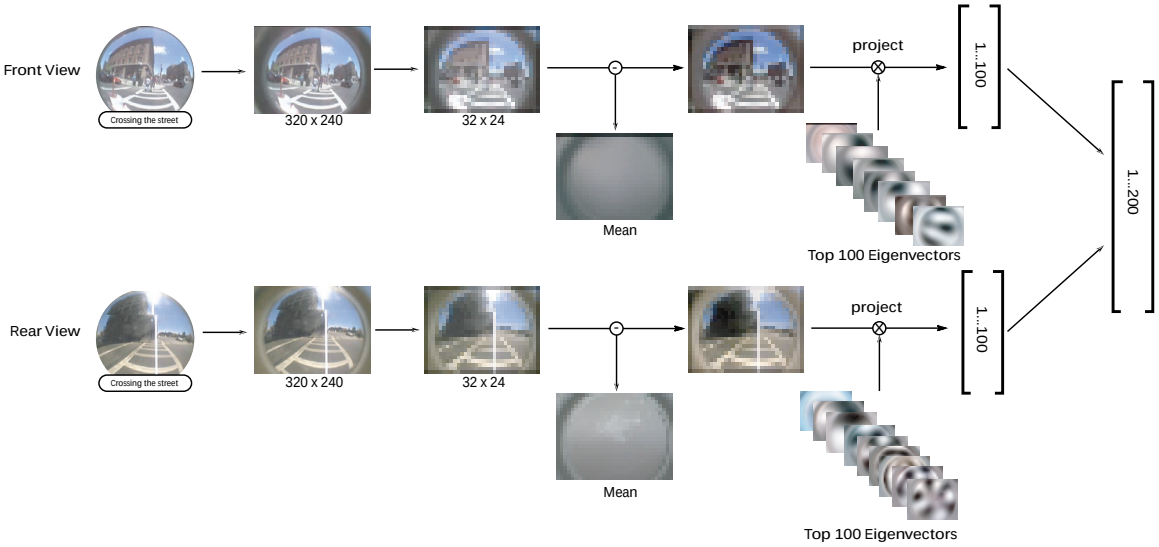


Figure 5-7: The processing pipeline for the alignment feature. The front and rear views are both subsampled and projected on to their respective top 100 eigenvectors. The result is concatenated into a 200-dimensional feature vector.

5.2 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.

5.2.1 Local 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. If we are lucky it is matched to other moments during which he is rubbing his eyes, but most likely it will simply match to another random time of some night making it seem as if the user has suddenly teleported in both space and time. In reality, this smoothness property of someone's temporal state applies at all time-scales in some form or another from seconds (e.g. locations, activity) to even months (e.g. seasonal changes which do add a systematic bias to environmental sensor readings and possibly even the subject's activities). 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.

Are transitions backwards in time appropriate? In addressing this let's consider the example of passing through a doorway. The typical sequence is: approaching the threshold → passing through it → moving away from the threshold. There is something about this sequence that won't allow this from ever happening with its subsequences permuted (e.g. moving away from the threshold → passing through it → approaching the threshold). This is what we call a causal sequence[1]. For speech the analogous concept is the string of phonemes that make up a word. It is obvious then in this case that transitions backwards in time are inappropriate. In general, since our features preserve orientation (front view features are not mixed up with rear view features, such as via the histogram and Radon-like transforms), all sequences are locally causal except in a few rare cases, which are what we call reversible. We discuss these next.

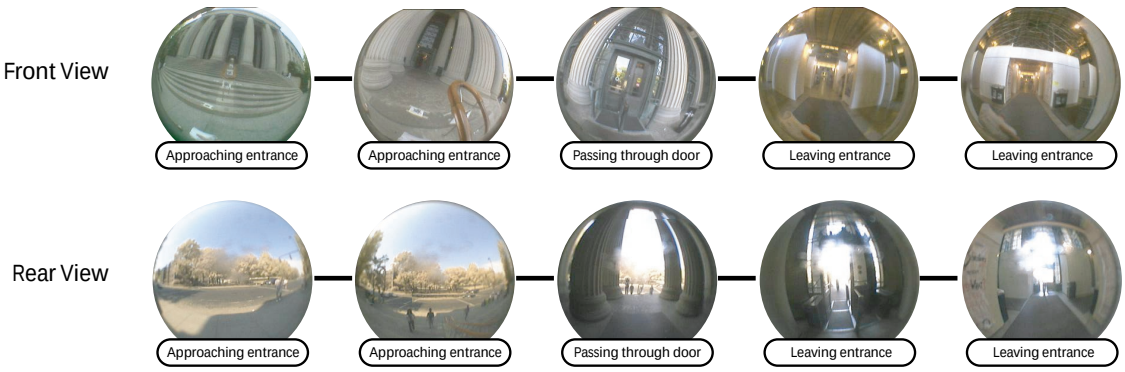


Figure 5-8: Entering a doorway is a causal sequence.

Another type of sequence is the reversible (or non-causal) sequence. This type of sequence might occur in either direction (forward or reverse) but it will never be seen with its parts permuted. Under normal circumstances, most atomic sequences of video are

[1] These causal sequences are related to the "event" concept of the author's previous work, [8]. The event in this case was clustered from similar data (collected via wearable microphone and camera) using a left-right HMM.

also irreversible. This is because a camera is an oriented sensor moving through space. However, even with an oriented camera, sequences can be reversed in two scenarios. There is the situation when the person carrying the camera walks backwards. This never happens in the I Sensed data set. A slightly more common but still rare situation is when approaching an object looks very similar to leaving it. We can detect these situations with our system because it would imply that the front and rear views are the same. Take for example the act of walking down a long monotonous hallway as shown in Figure 5-9. Since the hallway receding in the distance looks the same in both directions, there are no local clues as to which direction in time is the forward direction. In general reversible sequences are quite rare and we would gain so much more from keeping the constraint that time moves forward rather than supporting matches between sequences and their time-reversed counterparts.

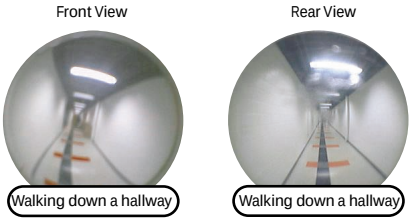


Figure 5-9: A long monotonous hallway is a reversible sequence.

So far we have been ignoring the actual definition of what is local in time. How big is the largest transition in time that is still local? This complication arises from the fact that even for causal sequences, substitutions and deletions are still possible. If we were guaranteed that sequences in the data will re-occur at exactly the same rate with out any noise (for example, caused by occlusions) then we should only need to consider transitions of one time unit. However, obviously this is not going to be the case with our data.

5.2.2 Global Time Constraints

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.

5.2.3 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.

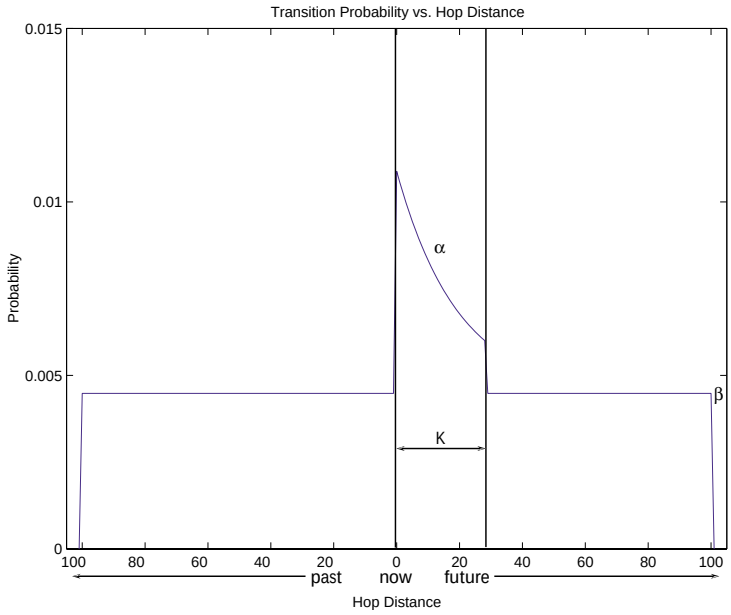


Figure 5-10: The parameterized form of the alignment HMM's transition probabilities.

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:

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

$D(\cdot)$ is the distance function on the features (L_1-norm in our case). Again, Z is a normalization constant. If $D(\cdot)$ 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.

5.3 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 5-11 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.

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

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 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.

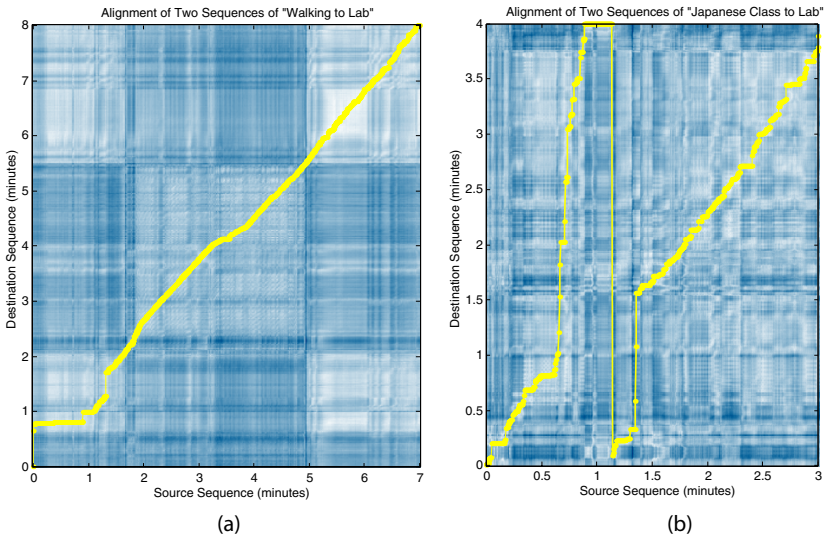


Figure 5-11: Alignment paths for (a) an α-matchable pair of sequences, and (b) a β-matchable pair of sequences.

5.4 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 that change detection methods are more robust to.

As clustering methods have become more popular for video indexing, researchers are agreeing that it is better to let the data define what a scene is rather than choosing beforehand the conditions for a scene. Zhong et. al. [57] has found that clustering features

extracted globally from shorter video segments (gotten via some other method) can alleviate some of these problems, but using time-averaged features to represent longer and longer segments of video will unnecessarily discard vital discriminative information. In our earlier experiments, we discussed a way around this by summarizing feature segments with HMMs, which encode feature dynamics, rather than averaging them out. Cluster-based segmentation is attractive because it allows the data to determine the scenes by how the data “clumps” in feature space. However, as always with clustering methods, the work is hidden in the similarity measure. For example, we can't find scenes in video just by clustering frame-based features. Most scenes are usually *not* a set of similar images, which is what clusters in an image space would give. In Figure 5-12, we give an example of what this typically looks like when clustering with frame-based features. Most scenes turn out to exhibit complex modes in image-feature space.

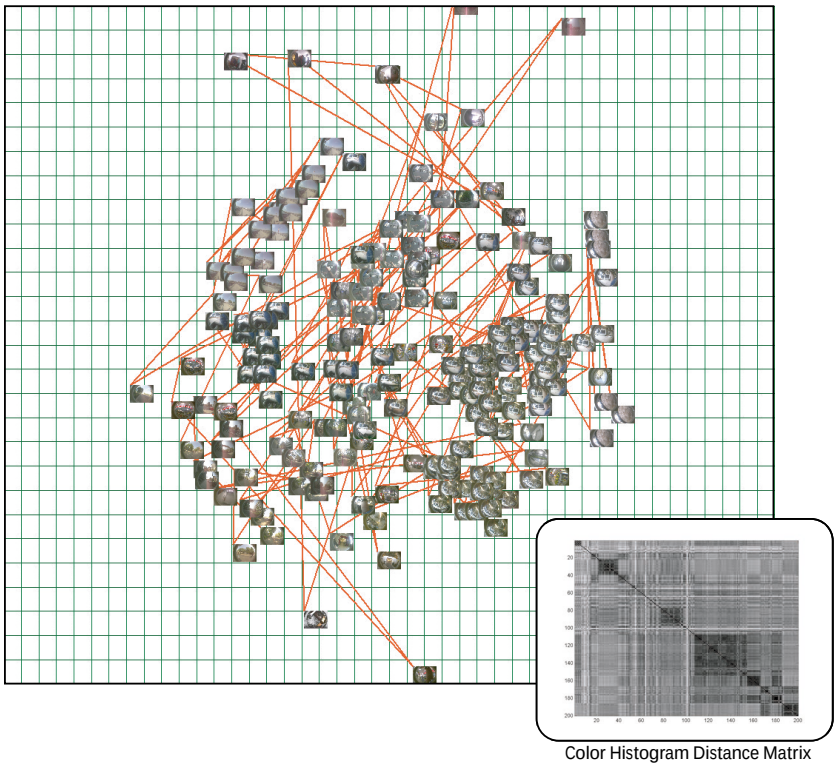


Figure 5-12: Multi-dimensional scaling (Sammon mapping) of a color histogram feature on the rear view images taken from a single day. The images nicely cluster based on their visual similarity, but the temporal continuity (red line) required by scene segmentation is not preserved.

A more accurate description of a scene is a path through the image space. (We can replace the word image with feature of course.) The hierarchy of HMMs (see Earlier Experiments) was one way to tackle this problem by clustering path dynamics instead of individual feature points. Hence, the clusters that were discovered were able to support scenes with highly variable image feature statistics. However, the problem with the clustering is that clusters will only form around high-density portions of the space. Even adding more centroids will tend to just divide up the densely clumped portions of the cluster space. In other words, there has to be a lot of supporting examples to form a cluster and hence scenes tend to be defined by the norm. We have been finding that with very long inhomogeneous data like the I Sensed data, this can be a very debilitating effect. The longer scenes completely overwhelm the shorter scenes. In order to achieve a clustering that adequately covers the observed set of path dynamics, we need to match the order of centroids with the number of scenes. So instead of estimating prototypical examples, why not just use the sequences themselves?

We now propose an alignment-based segmentation. Before giving the algorithm let's map it out qualitatively. Suppose we wish to find the scenes in a given sequence. Suppose also that our knowledge consists only of a huge bag 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 reader might ask: what about all the stuff that was grouped into one scene just because we found another matched sequenced in the past? The basic principle at work here is a minimum description length (MDL) one, 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 $x = (x_1, ..., x_T)$ be the source sequence in which we wish to find scene breaks. Let $\Upsilon = \left\{ y^1 = (y_1^1, ..., y_{N_1}^1), ..., y^L = (y_1^L, ..., y_{N_L}^L) \right\}$ 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, $s^* = (s_1^*, ..., s_T^*)$ $s_t^* \in 1...N$, 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 5-13 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 5.3). 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.

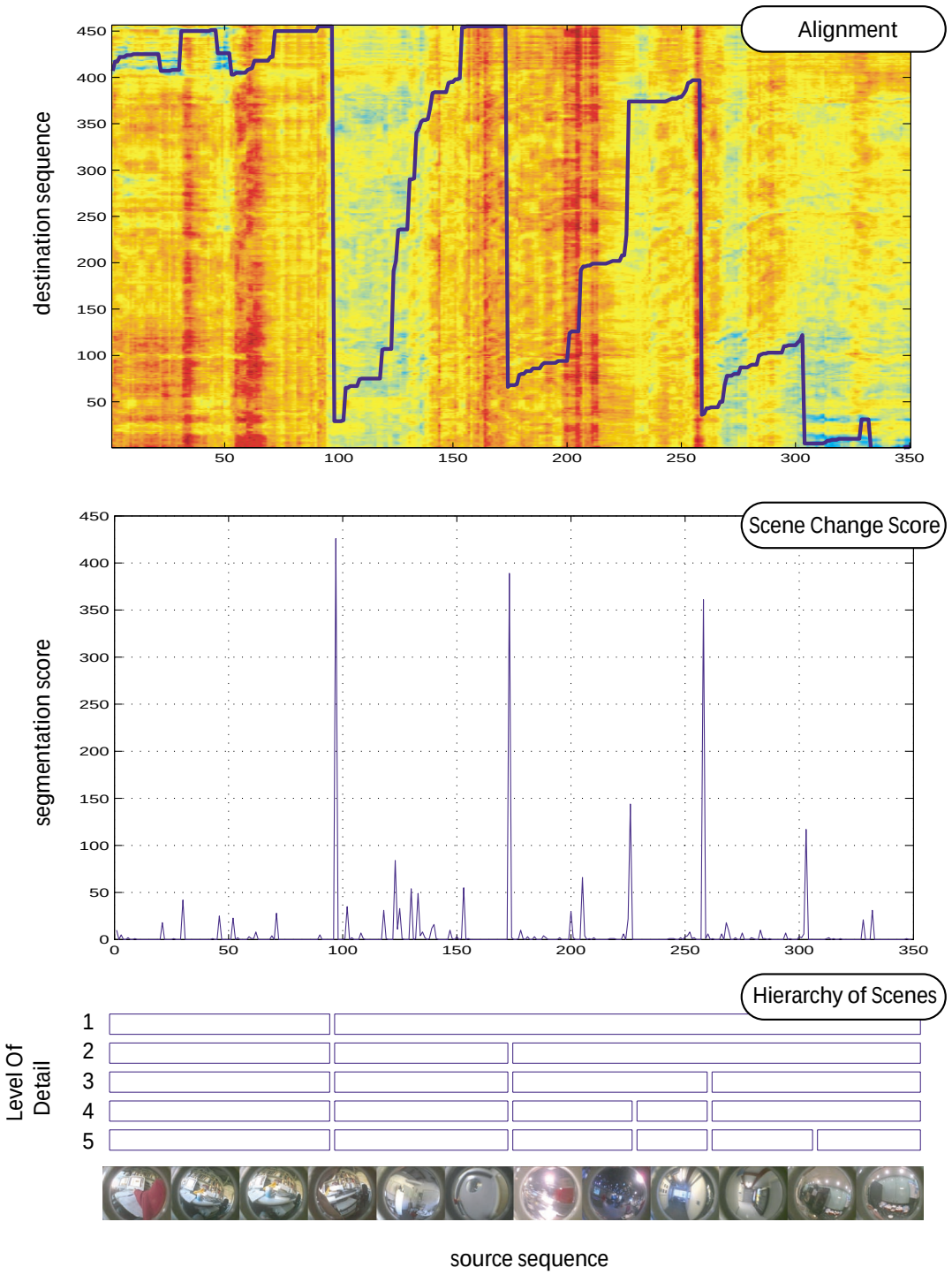


Figure 5-13: The algorithmic pipeline for segmenting a source sequence according to the contents of a destination sequence. Starting from top and proceeding to the bottom, (1) Alignment of the source sequence to the destination sequence, (2) Scoring each time step for the possibility of a scene change from the alignment path, (3) a Hierarchy of Scenes can be generated by sweeping a threshold across the scene change score.

5.5 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. Given that the similarities are pre-computed and stored in memory the typical lengths of time that we can align at each scale of detail in 5 minutes are given in Table 5-1. For longer times, recall that the alignment procedure is linear in the length of the source sequence but squared in the length of the destination sequence.

Alignment Scale of Detail	Sequence Length
Fine	6-7 minutes
Medium	1 day
Coarse	approx. 30 days

Table 5-1: Alignments lengths at the three levels of detail that a 1GHz computer with about 500MB of memory can compute in 5 minutes (given a pre-computed similarity matrix). One day is on average 8 hours of data for the I Sensed data set.

5.5.1 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 5-14 gives an example of the subject walking entering a famous building on campus and walking down a well-known 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.

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.

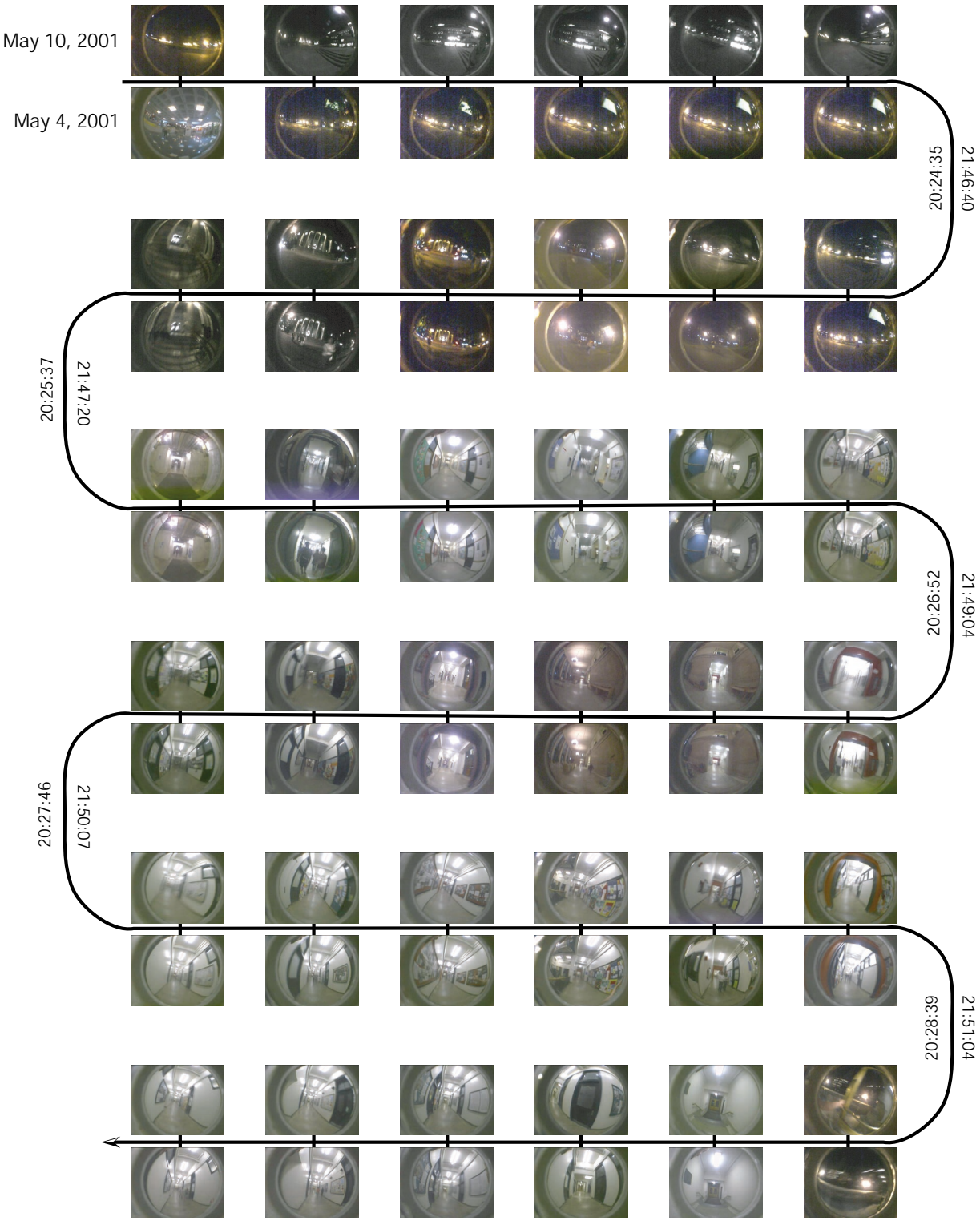


Figure 5-14: Fine-scale alignment of two similar scenes that happened on separate days: entering a building and walking down a hallway.

5.5.2 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 $t, t^* = 0$.
2. 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)
3. 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. In Figure 5-15, we give an examples of the RLE compression on a minute of video (in this case the subject is shopping at a convenience store so there is relatively a lot of activity) at two different settings of the change threshold. There are about 600 frames in the original sequence, which at 5% RLE compression is reduced to 189 frames and at 15% RLE compression is reduced to 12 frames. A 5% change threshold means that if less than 5% of the pixels change between the last image in the compressed sequenced and the current image under consideration then it will not be added to the compressed sequence. 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 Figure 5-15 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 pretty all of the major views are included.

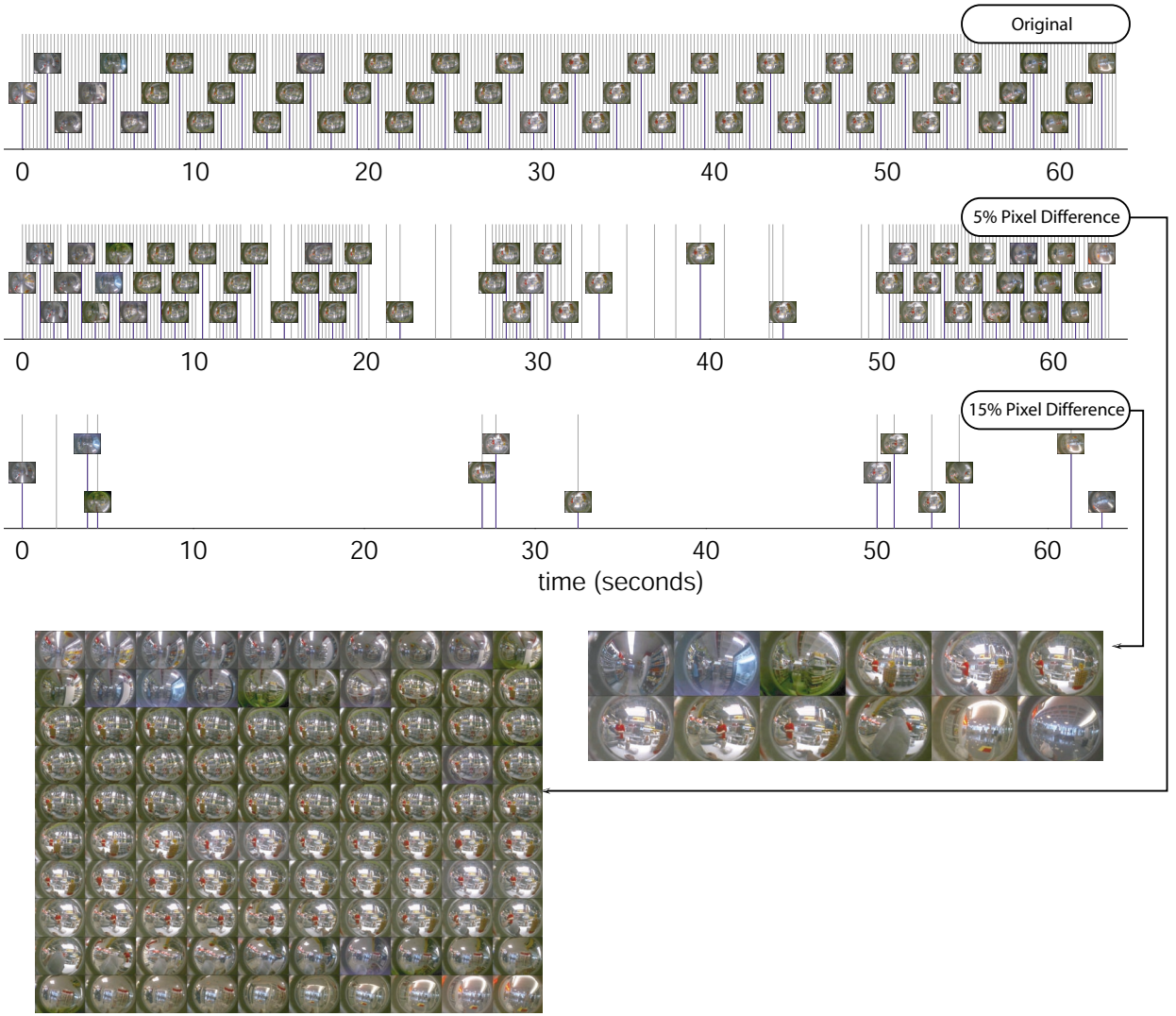


Figure 5-15: Different levels of run length encoding for a minute of video. The gray lines denote the actual sample rate of the video.

5.5.3 Medium-scale Alignment

The RLE compression step at a 15% change threshold allows us to align a pair of days in about 5 minutes. 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 Figure 5-16 we see 10 example paths gotten from aligning Weds. May 9, 2001 to 10 randomly chosen days after RLE-15% compression. Figure 5-17 and Figure 5-18 show video (highly subsampled for printing) of the source sequence (May 9) to the most similar (June 15) and least similar (May 10) days of the ten randomly chosen days in Figure 5-16, respectively. The alignment with a similar days is much more successful in finding moments in the destination sequence (June 15) that match the source sequence, but even the dissimilar day matches the source at a few moments. Particularly notice that in Figure 5-17 it looks like the alignment consistently matches outdoor moments in the source sequence with outdoor moments in the destinations sequence, meetings with meetings (even if they are in different rooms), office with office, and so on. The alignment with the dissimilar day is much less successful because it has little appropriate material to match with. This suggests that the alignment score could be used to find similar days. Figure 5-20 shows the alignment score for May 9th aligned with each of the days in the randomly chosen set. This score was used to choose the most similar and dissimilar days that were given in Figure 5-17 and Figure 5-18.

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 Figure 5-19. 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 Figure 5-21 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).

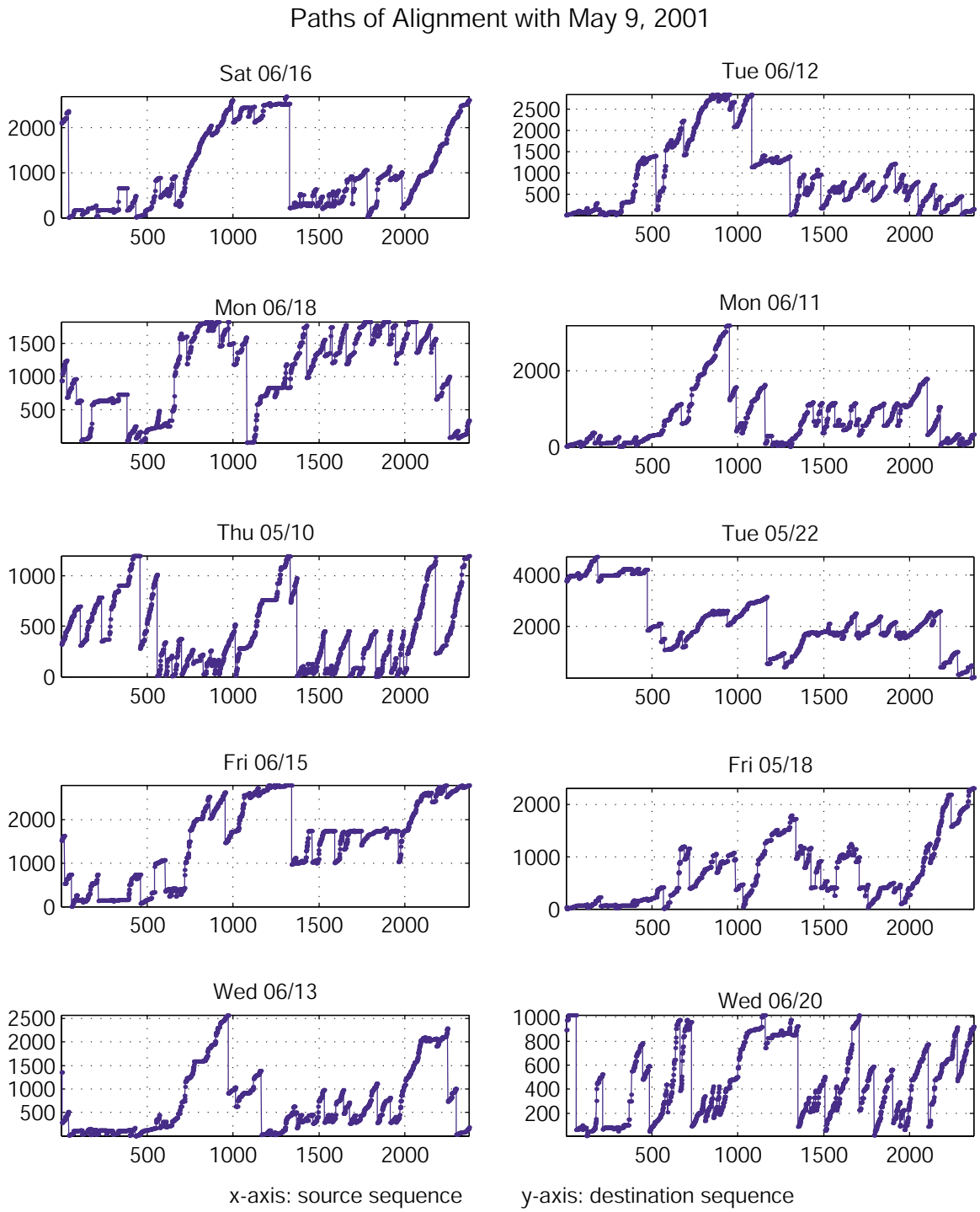


Figure 5-16: Examples of alignment paths at the medium level of detail. The source sequence is always May 9, 2001 and the destination sequences are 10 randomly chosen days. Each plot maps the source sequence to the destination sequence. The long hops are β -transitions.

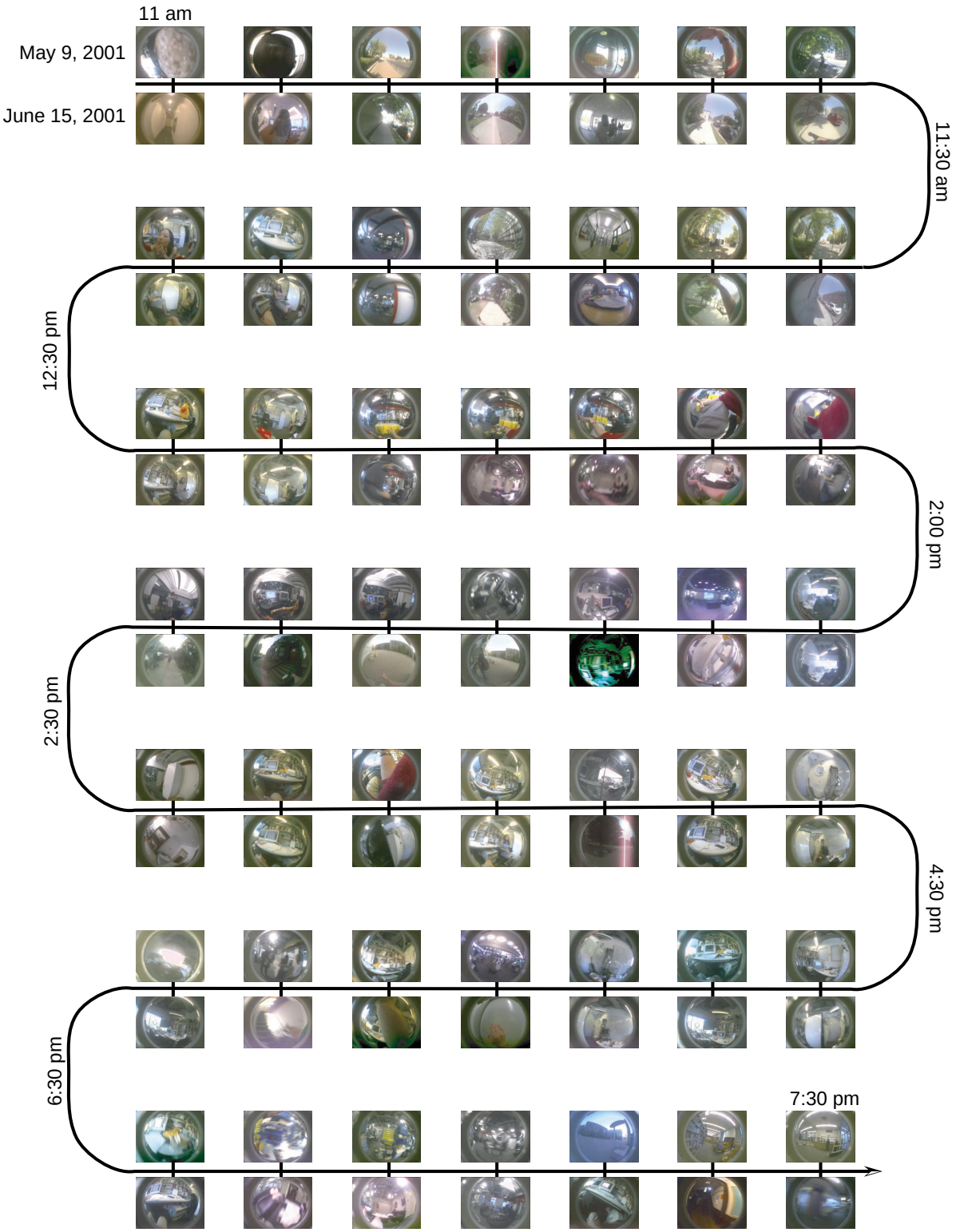


Figure 5-17: An alignment of Wednesday May 9 to a similar day (Friday June 15) at the medium level of detail (RLE-15%). The source and destination sequences have about 3000 frames, this figure only shows a few. Also notice the non-uniform sampling due to RLE.

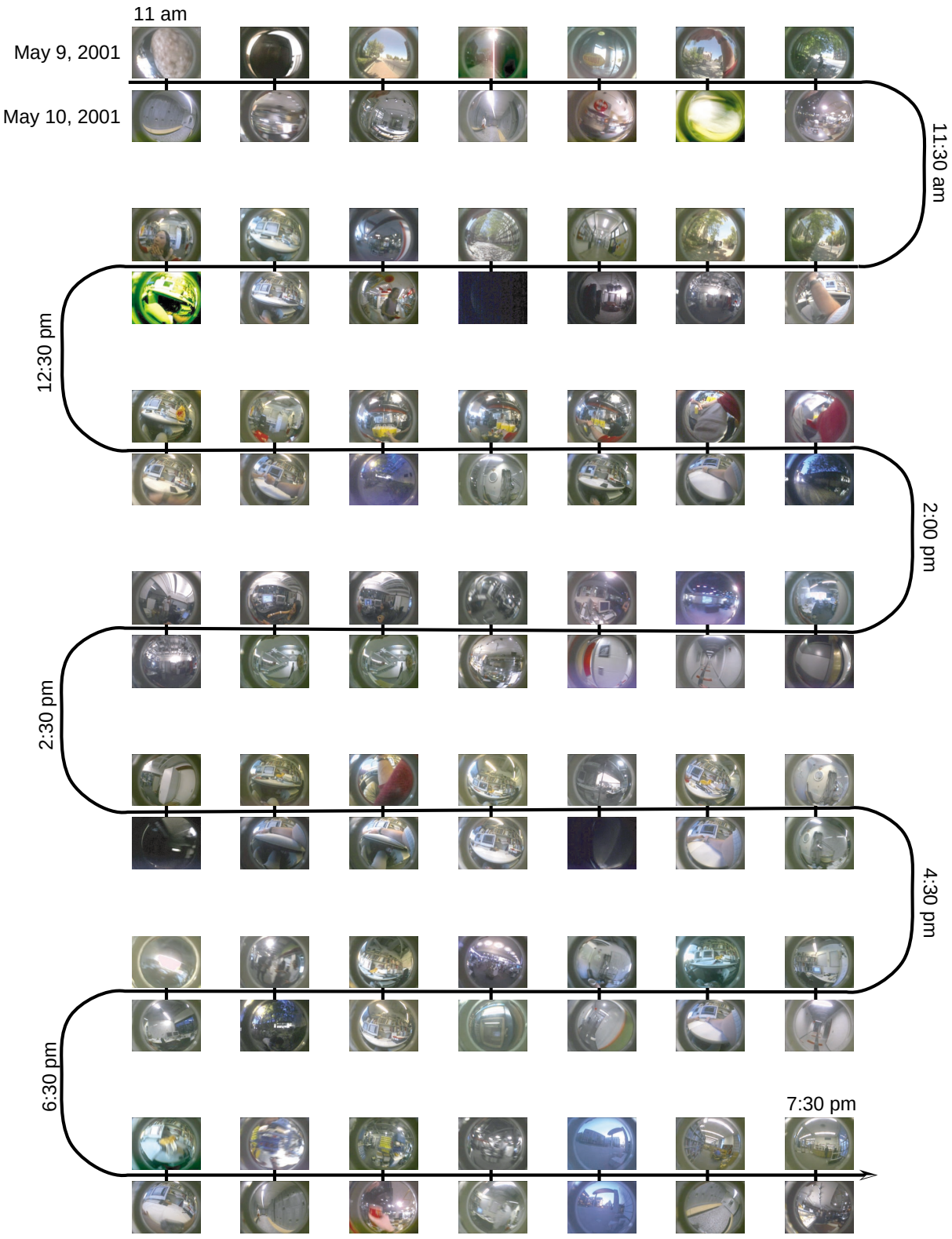


Figure 5-18: An alignment of Wednesday May 9 to a dissimilar day (Thursday May 10) at the medium level of detail (RLE-15%). The source and destination sequences have about 3000 frames, this figure only shows a few. Also notice the non-uniform sampling due to RLE.

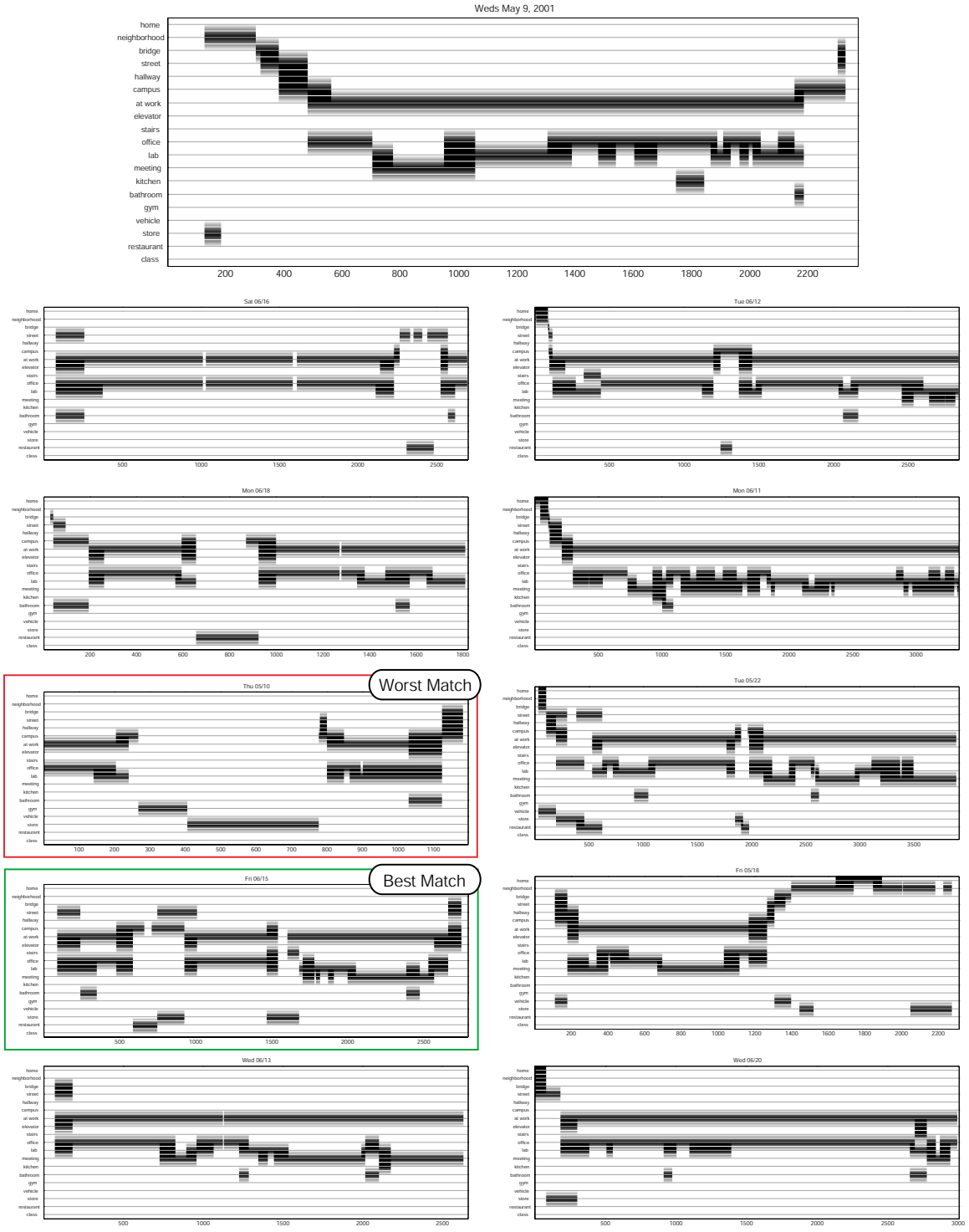


Figure 5-19: May 9th and the 10 randomly chosen days are given here with their situations (y-axis) hand-labeled every 5 minutes (x-axis). Each location category is represented by a horizontal track with dark areas indicating when a situations was occurring. There is overlap because the situation categories are not exclusive and more than one situation can occur in a 5 minute segment.

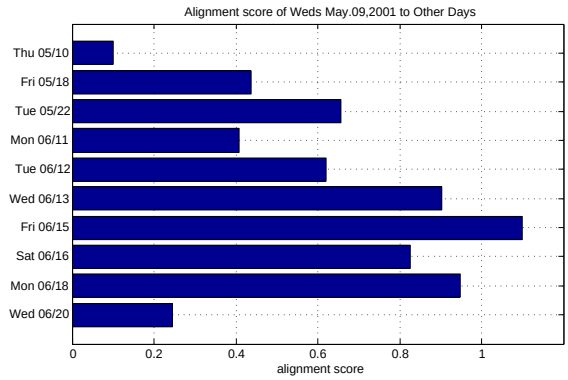


Figure 5-20: The alignment scores (normalized log likelihood) of May 9th to 10 randomly chosen days.

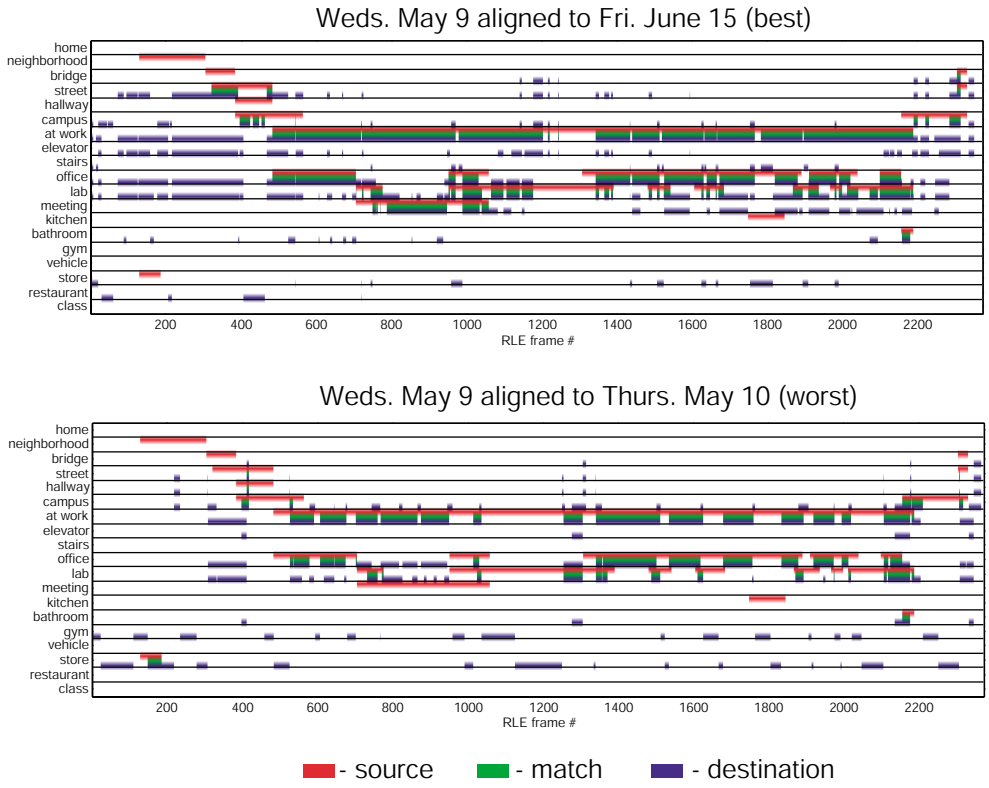


Figure 5-21: Comparison of situations (hand-labeled) for May 9 to the most similar and dissimilar days after alignment. The red bar is the location of the subject on May 9th. The blue bar is the location of the subject on the destination day at the matched time. The green bar denotes correct matching of the situation by the alignment. Notice that the situation categories are not exclusive. See Figure 5-19 for the situation labeling of the non-aligned versions.

5.5.4 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 $T \times N$ 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 $D(\mathrm{g})$ and the same transition function, $p_{seg}(s_t \mid s_{t-1})$, that was defined in section 5.4 (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 entire similarity matrix (stacked together to show the similarity between every pair of 5-minute intervals in the 32 days) is shown in Figure 5-22 with the alignment overlaid in yellow. There are no alignment matches inside the white areas along the diagonal because the day being aligned was itself naturally left out of the alignment. This forces the alignment procedure to find matches in other days.

* 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.

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 chapters we evaluate a few of these.

5.6 Summary

Defining the similarity between tracks of video requires that you first identify the pairs images that should be compared. We have solved this correspondence problem with a straightforward alignment technique. It turned out that the alignment also gave us an excellent method for scene segmentation because the alignment identified the points in the video where keeping temporal continuity with an example in the past was difficult. Hence there is a good chance (which increases with more data) that the video had reached a branching point in the scene transition graph, indicating a scene transition. Finally, we showed how to apply our alignment methods on a wide-range of time-scales.

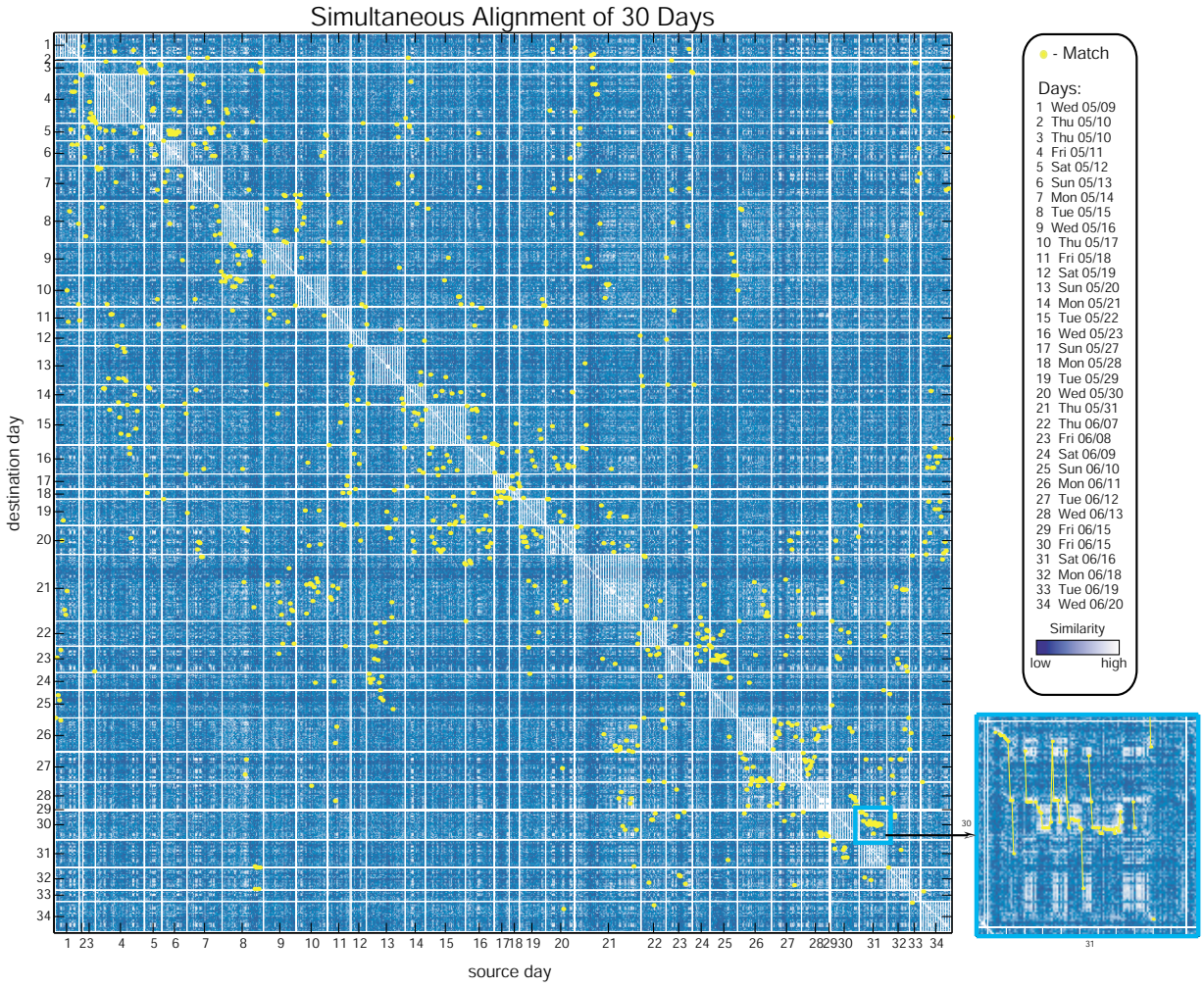


Figure 5-22: The result of coarse-scale alignment on 30 days. Each source day is aligned with the remaining days leaving itself out of the alignment (shown in white). The matches are depicted in yellow which when shown magnified (inset) reveals that they are paths. The backdrop depicts the similarity value

Chapter 6: 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 Chapter 5:) 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.

6.1 The Situations

We labeled 20 days of the days used in the 30 day alignment (of section 5.5.4) 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 6-1 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).

Note: 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.

6.2 Context-free Classification

In 5.5.4 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 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 6-1 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).

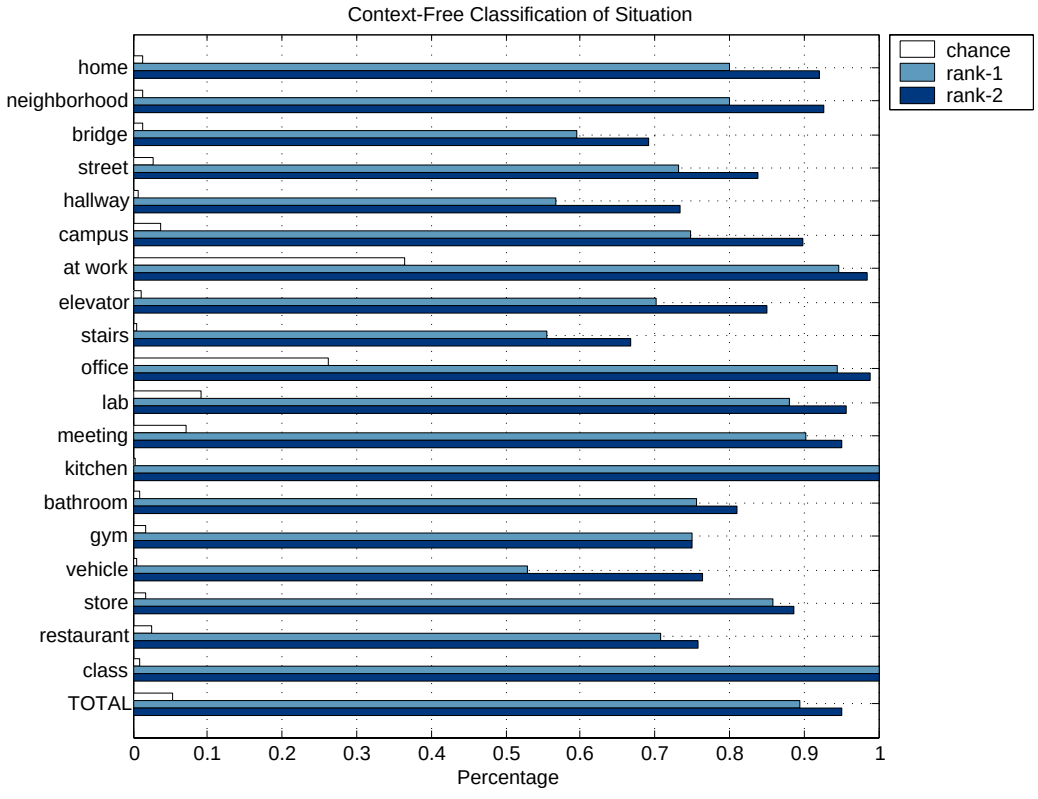


Figure 6-1: Rank-1 and rank-2 situation matching accuracy for the medium-level chunks via their alignment score. The figure gives the per situation accuracy and the total accuracy along with the chance recognition rates.

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, bldg. 66, bldg. 7, bldg. 10
at work	Media Lab (contains any of the other locations that are at work)
elevator	elevator (anywhere)
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 6-1: The situations and the actual location labels that they represent.

6.3 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 6-2 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.

6.4 Classification with Long-term Context

Fortunately, we have an ideal tool for bringing long-term context to the classification problem – alignment. Recall in 5.5.4 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 6-3 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.

[*] 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.

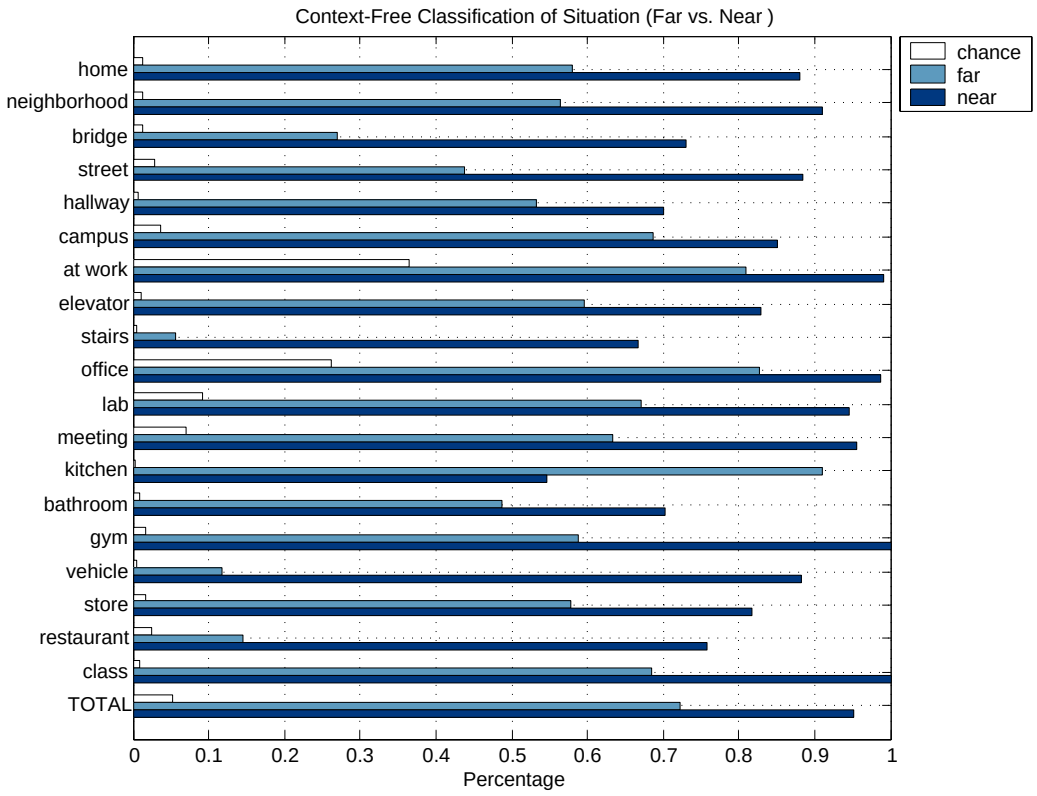


Figure 6-2: The performance of the short-term classifier when we force the match to be in the same day (near) and in another day (far).

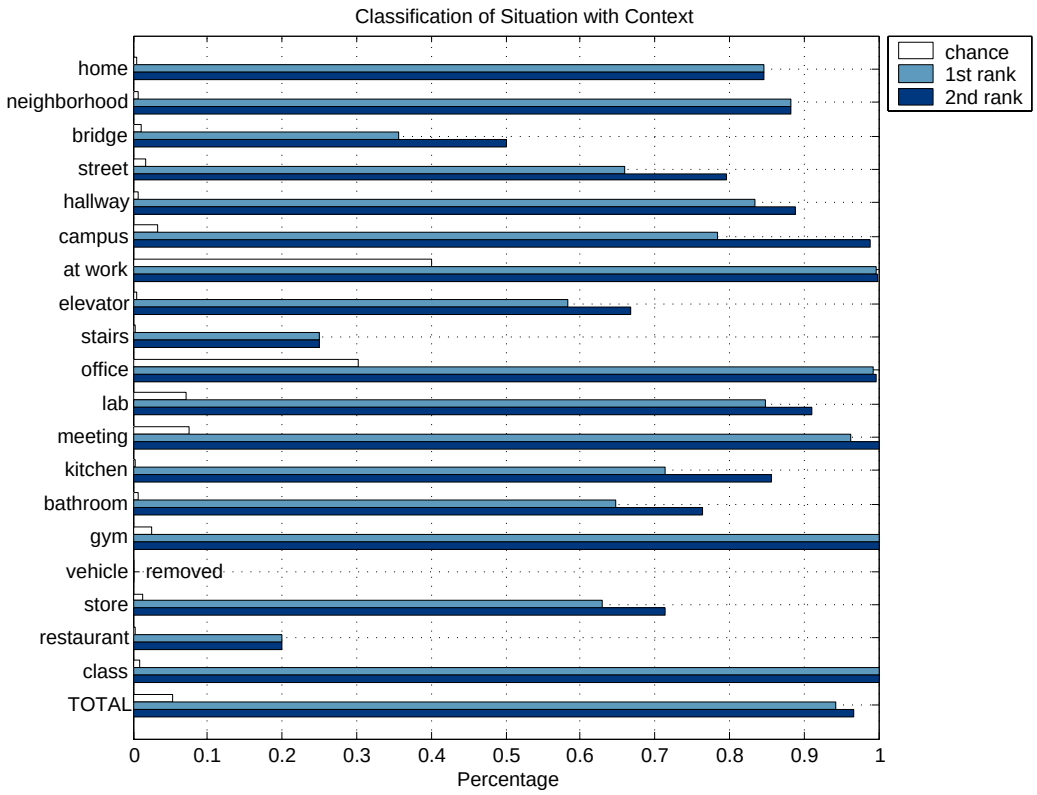


Figure 6-3: The performance of the contextualized classifier at matching situations. Rank-1 is the accuracy when only considering the actual chunk aligned to the test chunk. Rank-2 is when a correct match exists within one time step in either direction along the alignment path. The vehicle situation had no labeled pairs of matches to count.

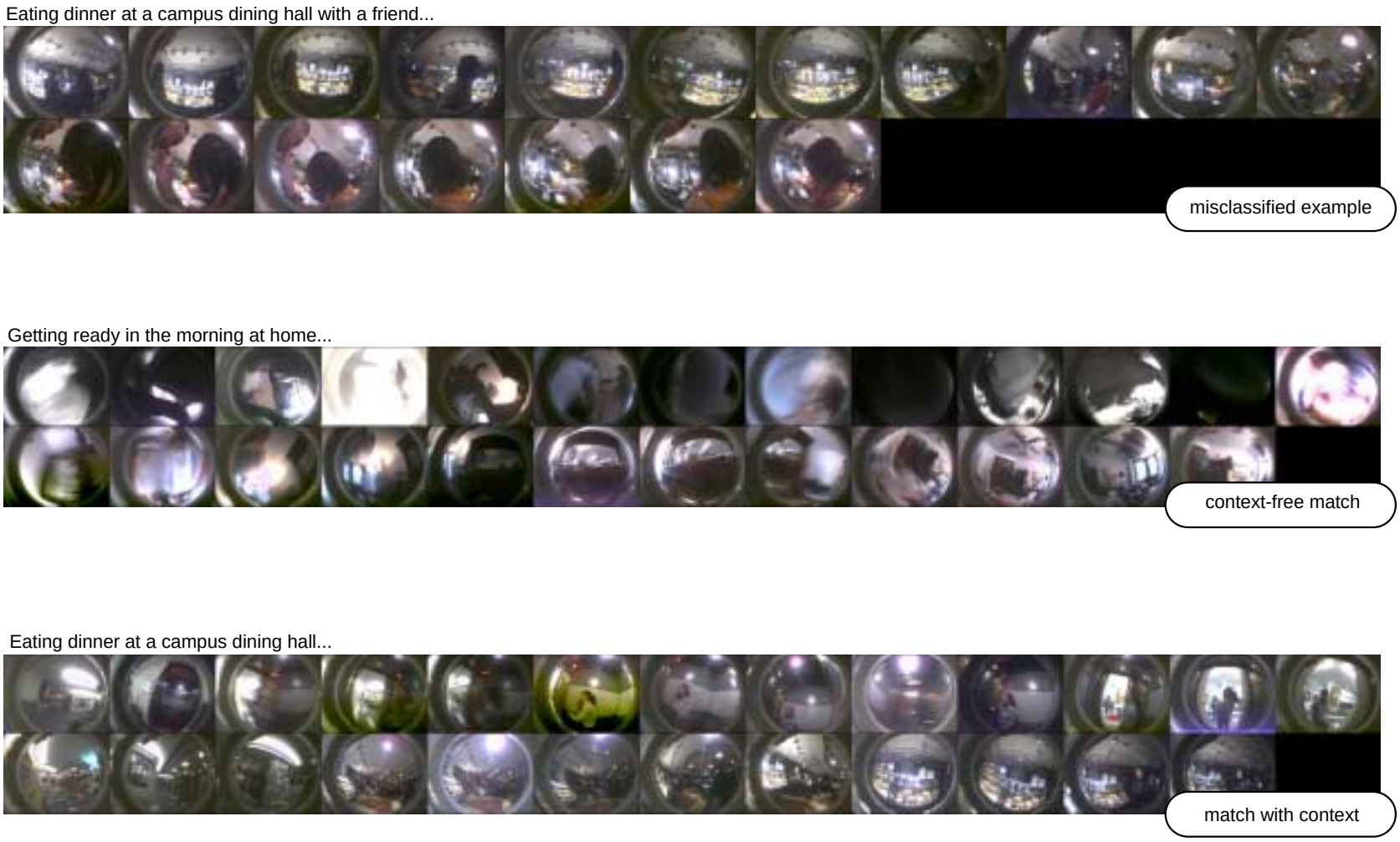


Figure 6-4: This restaurant situation was misclassified by the context-free classifier but correctly matched with context. The example was misclassified due to the protracted occlusion of the camera by another person's head (last 7 frames) and matched to a highly varying sequence which has a high likelihood of producing decent alignments with many types of situations.

6.5 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 6-5 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%.

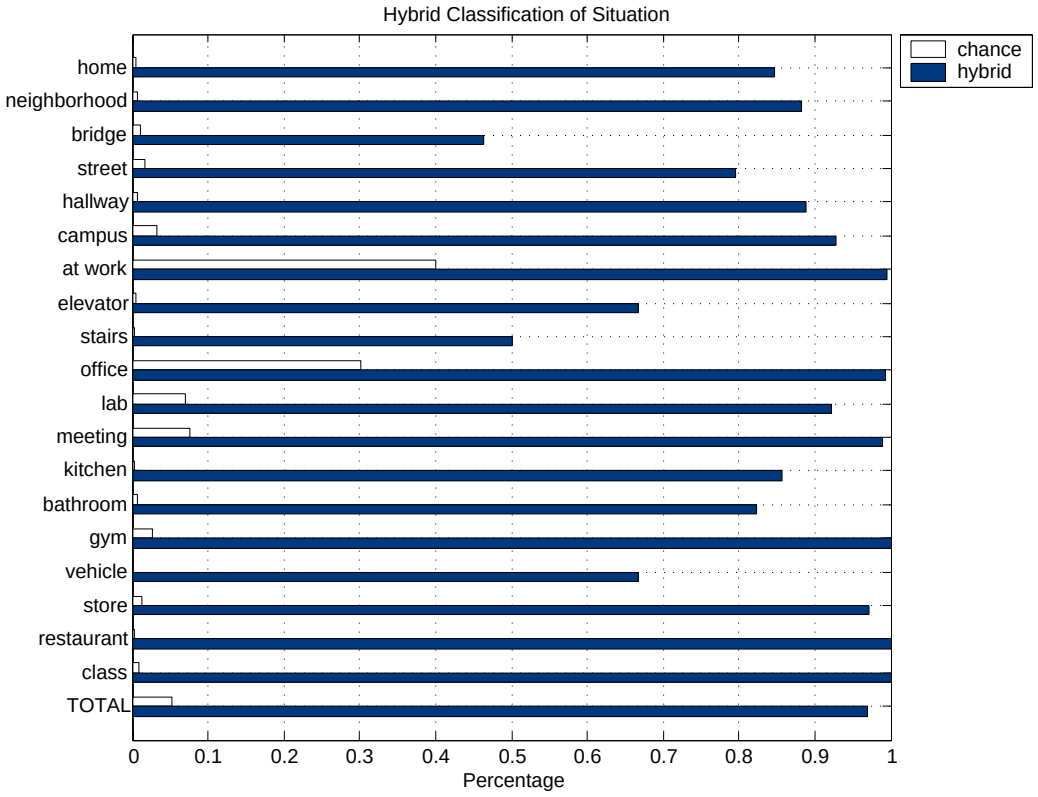


Figure 6-5: Performance of the hybrid classifier at situation classification. This classifier uses context-free classification on the near matches and contextualized classification on the far matches

Chapter 7: 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 Chapter 5: 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

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

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).

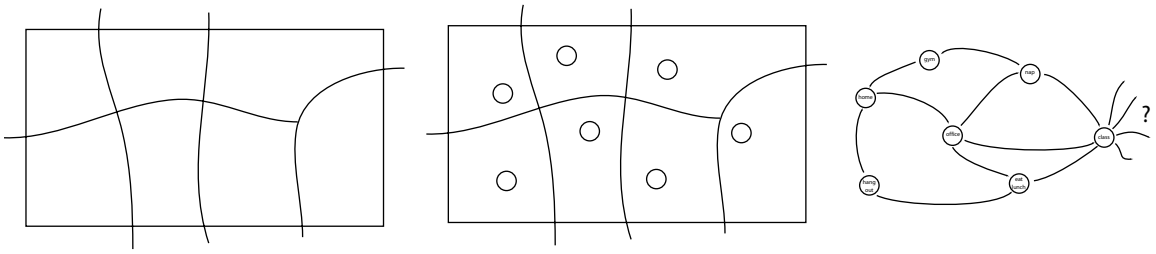


Figure 7-1: By cutting up (e.g. clustering) a situation space (left) into discrete regions, we can tabulate the transitions that occur between regions over the course of an individual's day (right).

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 5.4. 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 Figure 7-1). We conclude the chapter by interpreting these results.

7.1 Clustering the situation space

Previously in section 5.4 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 5.5.4). 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.

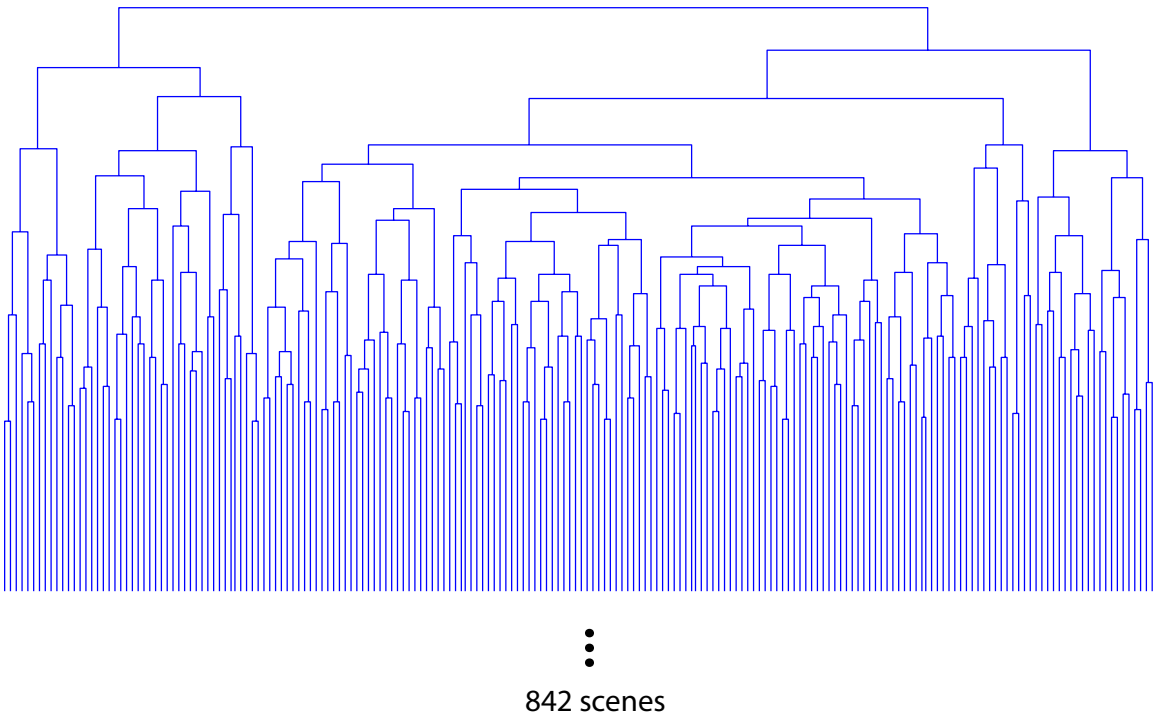


Figure 7-2: The hierarchical cluster tree on the 842 scenes (arranged horizontally) segmented from 30 days. Clusters are merged successively to form compact larger clusters.

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 Figure 7-2. To obtain an N-clustering of the 842 scenes, we simply stop merging when we reach N clusters.

7.2 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 Figure 7-3. The predictive 1st order Markov model is,

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

where $x_t^n \in (1,...,n)$ 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).

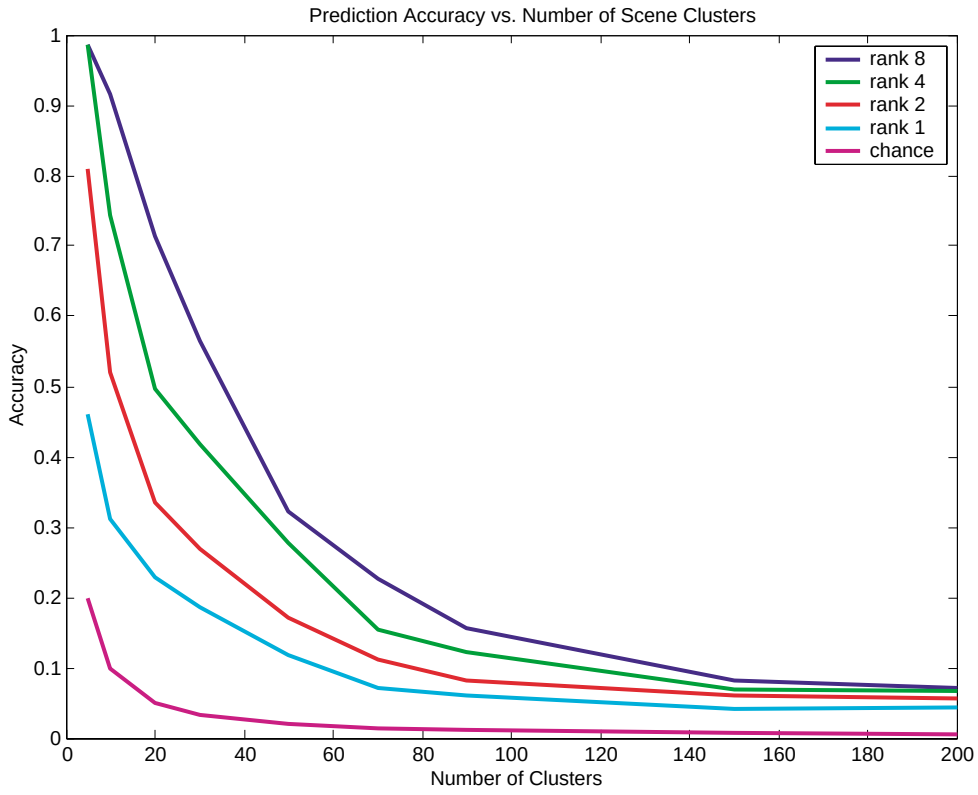


Figure 7-3: A plot of how 1st order Markov prediction accuracy varies with the number of scene clusters.

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_t^n, is extractable by a 1st order Markov model from the past, x_{t-1}^n. 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:

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

In this case, $p(x_t^n, x_{t-1}^n)$ is again estimated from co-occurrence accounts over a training set after removing symbol repetitions. Finally, in Figure 7-4 we plot the number of bits of mutual information per symbol,

$$B_n = \frac{I(x_t^n; x_{t-1}^n)}{n}$$

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.

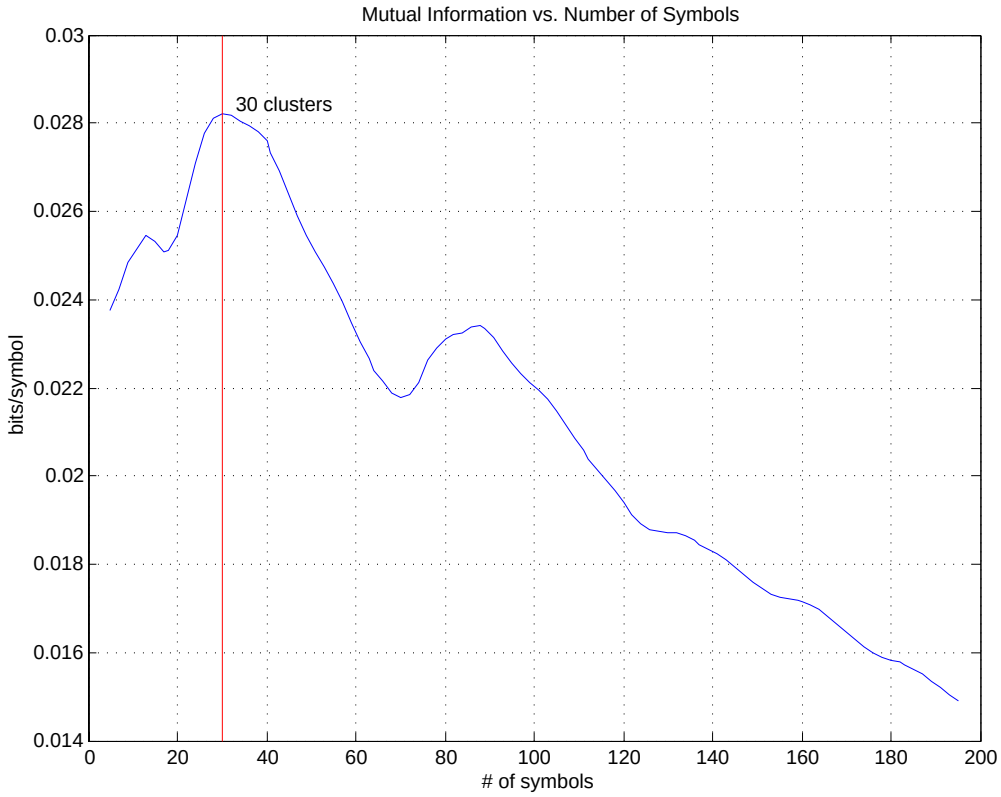


Figure 7-4: Number of bits of mutual information per symbol between a pair of successive scene symbols over 30 days.

7.3 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. This is experimentally verified in Figure 7-5 where we plot perplexity versus symbol. This chart shows that the bottom five symbols have perplexities below 5 (i.e. at most 5 different symbols are seen after this symbol) and the top five symbols have perplexities over 14. If we go back to the data and see what these symbols are corresponding to then we note interestingly things. The low perplexity symbols denote scenes such as leaving home. Overwhelming the next scene is a commuting to work scene (i.e. taking the typical route to work), but there are a few a variations typically seen on weekends of the subject leaving home but not going to work.

The highest perplexity scene (perplexity of 22) was the office scene. Typically this was the base of operations for the subject because he could transition to almost any other major scene from here (home, restaurant for lunch or dinner, gym, go out with friends, etc.). A perplexity of 22 means that there were 22 distinct symbols that were observed to follow the given symbol. It gives us no indication of which of those symbols typically followed or rarely followed. We can look to predictive accuracy to measure these things.

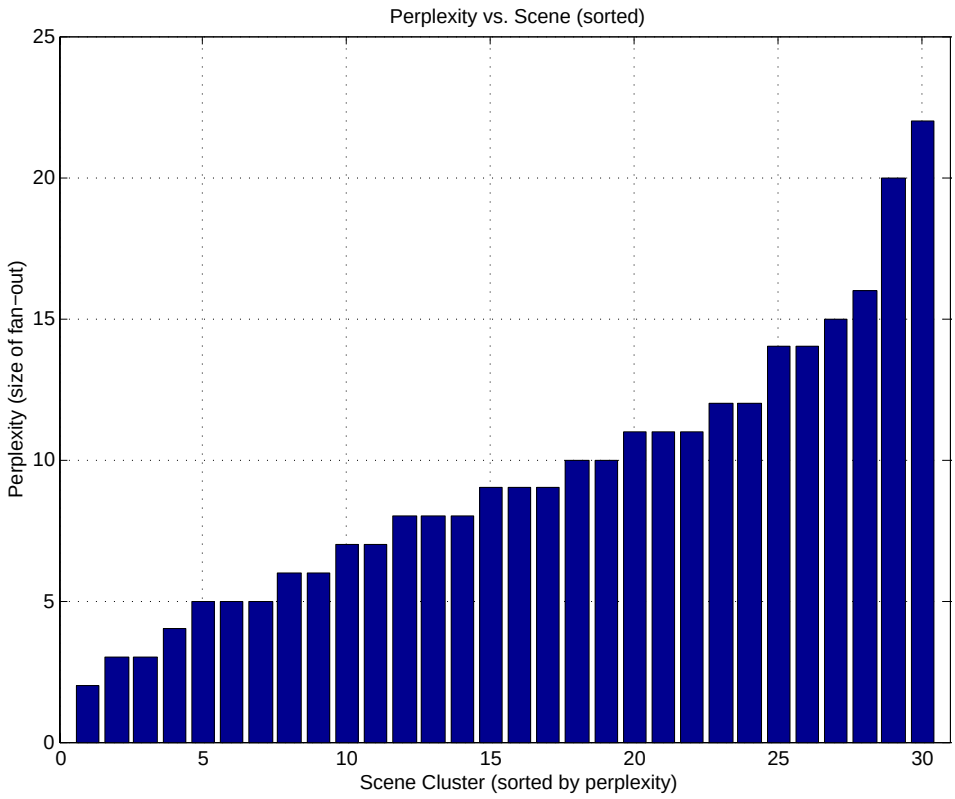


Figure 7-5: Perplexity is plotted for each scene cluster (or symbol). The scene clusters are sorted from low to high perplexity.

We show the prediction accuracy for each symbol in Figure 7-6. 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 becomes even more obvious if we plot the total predictive accuracy after throwing away the high perplexity parts of the data (Figure 7-7). 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. So we might conclude that statistical perplexity (i.e. number of choices weighted by occurrence) of all symbols is much lower than the hard perplexity depicted in Figure 7-5. For example approx 50% of all typical choices are in the top 4 choices made by the 1st order Markov model.

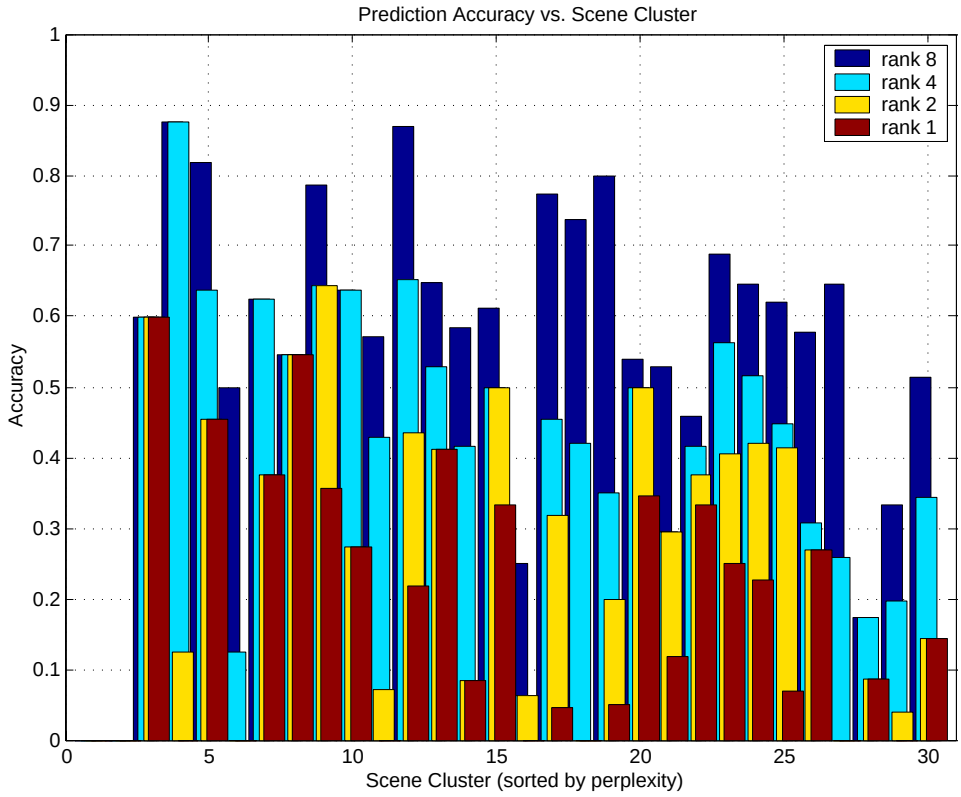


Figure 7-6: This plot shows the ability of a 1st order Markov model to predict the next scene from the previous scene. The prediction accuracy varies widely depending on the scene.

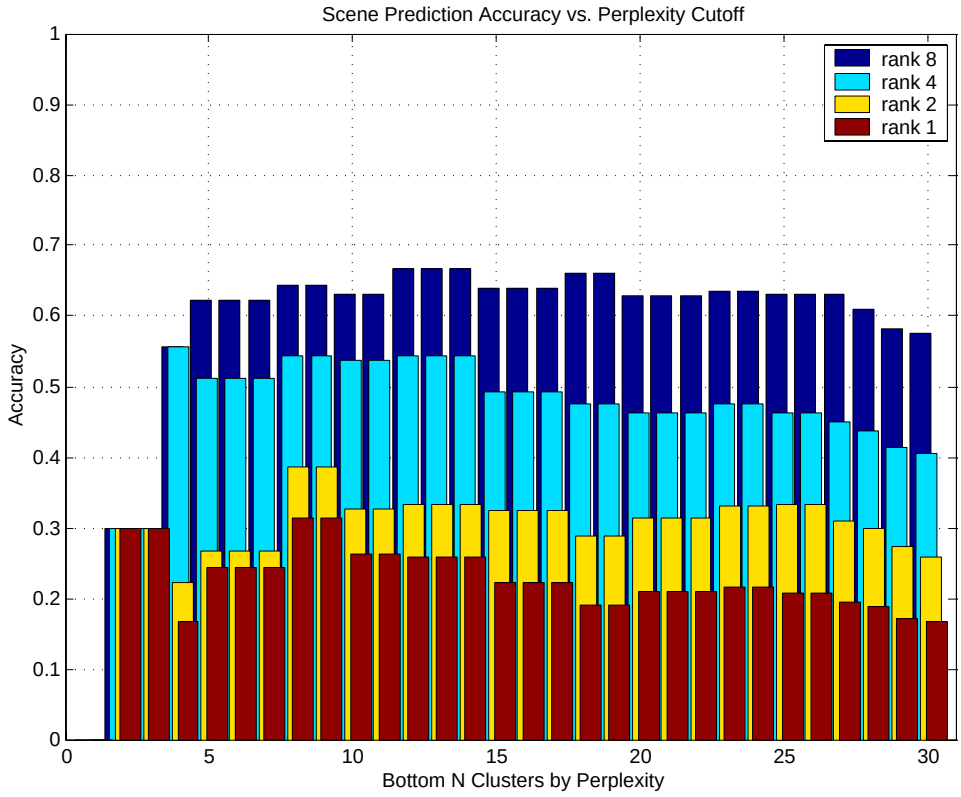


Figure 7-7: Total prediction accuracy for the N symbols with the smallest perplexity (N=1…30).

Chapter 8: Conclusions

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.

It is exciting how many ways this work can be extended and improved. There is a lot of wide-open territory in the composition of the sensors used. We collected auditory and orientation data in addition to the visual, but didn't require them in this work. Optical flow coupled with gyros and accelerometers in a similarly simple framework could theoretically capture the various telekinetic situations (sitting, running, gait, speed, etc.). The sensors used in this work are forever fixed on the outside and can only observe the *results* of the complex phenomenon happening inside the human body. Bio-sensors such as heart rate, breathing, galvanic skin response, hormone levels, and more are representative of the internal state of a person. These sensors could possibly provide us with a window into the why a person acts.

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

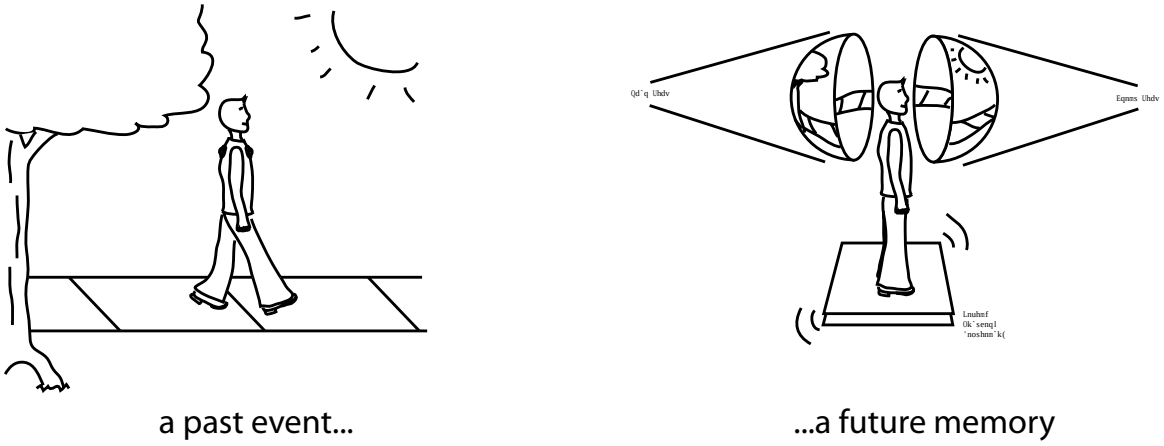


Figure 8-1: A proposed environment for re-experiencing the memories recorded by our I Sensed wearable. Front and rear views are projected onto hemispherical screens along with audio as the audience sits or stands on a motion platform.

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 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 8-1 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.

Chapter 9: Bibliography

[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.

The End