Multimodal Dynamics:
Self-Supervised Learning in Perceptual and Motor Systems

by

Michael H. Coen

Submitted to the Department of Electrical Engineering and Computer Science on May *x*, 2006 in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

ABSTRACT

This thesis presents a self-supervised framework for perceptual and motor learning based upon correlations in different sensory modalities. The brain and cognitive sciences have gathered an enormous body of neurological and phenomenological evidence in the past half century demonstrating the extraordinary degree of interaction between sensory modalities during the course of ordinary perception. We develop a framework for creating artificial perceptual systems that draws on these findings, where the primary architectural motif is the cross-modal transmission of perceptual information to enhance each sensory channel individually. We present self-supervised algorithms for learning perceptual grounding, intersensory influence, and sensory-motor coordination, which derive training signals from internal cross-modal correlations rather than from external supervision. Our goal is to create systems that develop by interacting with the world around them, inspired by development in animals.

We demonstrate this framework with: (1) a system that learns the number and structure of vowels in American English by simultaneously watching and listening to someone speak. The system then cross-modally clusters the correlated auditory and visual data. It has no advance linguistic knowledge and receives no information outside of its sensory channels. This work is the first unsupervised acquisition of phonetic structure of which we are aware, outside of that done by human infants. (2) a system that learns to sing like a zebra finch, following the developmental stages of a juvenile zebra finch. It first learns the song of an adult male and then listens to its own initially nascent attempts at mimicry through an articulatory synthesizer. In acquiring the birdsong to which it was initially exposed, this system demonstrates self-supervised sensorimotor learning. It also demonstrates afferent and efferent equivalence – the system learns motor maps with the same computational framework used for learning sensory maps.

Thesis Supervisor: Whitman Richards
Title: Professor of Brain and Cognitive Sciences

Thesis Supervisor: Howard Shrobe
Title: Principal Research Scientist, EECS

We have sat around for hours and wondered how you look. If you have closed your senses upon silk, light, color, odor, character, temperament, you must be by now completely shriveled up. There are so many minor senses, all running like tributaries into the mainstream of love, nourishing it.
The Diary of Anais Nin (1943)

He plays by sense of smell.
Tommy, The Who (1969)

Chapter 1

Introduction

This thesis presents a unified framework for perceptual and motor learning based upon correlations in different sensory modalities. The brain and cognitive sciences have gathered a large body of neurological and phenomenological evidence in the past half century demonstrating the extraordinary degree of interaction between sensory modalities during the course of ordinary perception. We present a framework for artificial perceptual systems that draws on these findings, where the primary architectural motif is the cross-modal transmission of perceptual information to structure and enhance sensory channels individually. We present self-supervised algorithms for learning *perceptual grounding*, *intersensory influence*, and *sensorimotor coordination*, which derive training signals from internal cross-modal correlations rather than from external supervision. Our goal is to create perceptual and motor systems that develop by interacting with the world around them, inspired by development in animals.

Our approach is to formalize mathematically an insight in Aristotle's *De Anima* (350 B.C.E.), that *differences in the world are only detectable because different senses perceive the same world events differently*. This implies both that sensory systems need some way to share their different perspectives on the world and that they need some way to incorporate these shared influences into their own internal workings.

A glossary of technical terms is contained in Appendix 1. Our usage of the word "sense" is defined in §1.5.

We begin with a computational methodology for *perceptual grounding*, which addresses the first question that any natural (or artificial) creature faces: *what different things in the world am I capable of sensing?* This question is deceptively simple because a formal notion of what makes things different (or the same) is non-trivial and often elusive. We will show that animals (and machines) can learn their perceptual repertoires by simultaneously correlating information from their different senses, even when they have no advance knowledge of what events these senses are individually capable of perceiving. In essence, by *cross-modally* sharing information between different senses, we demonstrate that sensory systems can be perceptually grounded by mutually bootstrapping off each other. As a demonstration of this, we present a system that learns the number (and formant structure) of vowels in American English, simply by watching and listening to someone speak and then cross-modally clustering the accumulated auditory and visual data. The system has no advance knowledge of these vowels and receives no information outside of its sensory channels. This work is the first unsupervised acquisition of phonetic structure of which we are aware, at least outside of that done by human infants, who solve this problem easily.

The second component of this thesis naturally follows perceptual grounding. Once an animal (or a machine) has learned the range of events it can detect in the world, *how does it know what it's perceiving at any given moment?* We will refer to this as *perceptual interpretation*. Note that grounding and interpretation are different things. By way of analogy to reading, one might say that *grounding* provides the dictionary and *interpretation* explains how to disambiguate among possible word meanings. More formally, grounding is an ontological process that defines what is perceptually knowable, and interpretation is an algorithmic process that describes how perceptions are categorized within a grounded system. We will present a novel framework for perceptual interpretation called *influence networks* (unrelated to a formalism know as *influence diagrams*) that blurs the distinctions between different sensory channels and allows them to influence one another while they are in the midst of perceiving. Biological perceptual systems share cross-modal information routinely and opportunistically (Stein and Meredith 1993, Lewkowicz and Lickliter 1994, Rock 1997, Shimojo and Shams 2001, Calvert et al. 2004, Spence and Driver 2004); *intersensory influence* is an essential component of perception but one that most artificial perceptual systems lack in any meaningful way. We argue that this is among the most serious shortcomings facing them, and an engineering goal of this thesis is to propose a workable solution to this problem.

The third component of this thesis enables sensorimotor learning using the first two components, namely, perceptual grounding and interpretation. This is surprising because one might suppose that motor activity is fundamentally different than perception. However, we take the perspective that motor control can be seen as perception *backwards*. From this point of view, we imagine that – in a notion reminiscent of a Cartesian theater – an animal can "watch" the activity in its own motor cortex, as if it were a privileged form of *internal* perception. Then for any motor act, there are two associated perceptions – the *internal* one describing the generation of the act and the *external* one describing the self-observation of the act. The perceptual grounding framework described above can then *cross-modally ground* these internal and external perceptions with respect to one another. The power of this mechanism is that it can learn mimicry, an essential form of behavioral learning (see the developmental sections of Meltzoff and Prinz 2002) where one animal acquires the ability to imitate some aspect of another's activity, constrained by the capabilities and dynamics of its own sensory and motor systems. We will demonstrate sensorimotor learning in our framework with an artificial system that learns to sing like a zebra finch by first listening to a real bird sing and then by learning from its own initially uninformed attempts to mimic it.

This thesis has been motivated by surprising results about how animals process sensory information. These findings, gathered by the brain and cognitive sciences communities primarily over the past 50 years, have challenged century long held notions about how the brain works and how we experience the world in which we live. We argue that current approaches to building computers that perceive and interact with the real, human world are largely based upon developmental and structural assumptions described by Piaget (1954) – although tracing back several hundred years – that are no longer thought to be descriptively or biologically accurate. In particular, the notion that perceptual senses are in functional isolation – that they do not internally structure and influence each other – is no longer tenable, although we still build artificial perceptual systems as if it were.

1.1 Computational Contributions

This thesis introduces three new computational tools. The first is a mathematical model of *slices*, which are a new type of data structure for representing sensory inputs. Slices are topological manifolds that encode dynamic perceptual states and are inspired by surface models of cortical tissue (Dale et al. 1999, Fischl et al. 1999, Citti and Sarti 2003, Ratnanather et al. 2003). They can represent both symbolic and numeric data and provide a natural foundation for aggregating and correlating information. Slices represent the data in a perceptual system, but they are also *amodal*, in that they are not specific to any sensory representation. For example, we may have slices containing visual information and other slices containing auditory information, but it may not be possible to distinguish them further without additional information. In fact, we can equivalently represent either sensory or motor information within a slice. This generality will allow us to easily incorporate the learning of motor control into what is initially a perceptual framework.

The second tool is an algorithm for *cross-modal clustering,* which is an unsupervised technique for organizing slices based on their spatiotemporal correlations with other slices. These correlations exist because an event in the world is simultaneously – but differently – perceived through multiple sensory channels in an observer. The hypothesis underlying this approach is that the world has regularities – natural laws tend to correlate physical properties (Thompson 1917, Richards 1980, Mumford 2004) – and biological perceptory systems have evolved to take advantage of this. One may contrast this with standard mathematical approaches to clustering, where some knowledge of the clusters, e.g., how many there are or their distributions, must be known a priori in order to derive them. Without knowing these parameters in advance, algorithmic clustering techniques may not be robust (Kleinberg 2002, Still and Bialek 2004). Assuming that in many circumstances animals cannot know the parameters underlying their perceptual inputs, how can they learn to organize their sensory perceptions? Cross-modal clustering answers this question by exploiting naturally occurring intersensory correlations.

The third tool in this thesis is a new family of models called *influence networks* (Figure 1). Influence networks use slices to interconnect independent perceptual systems, such as those illustrated in the classical view in Figure 1a, so they can influence one another during perception, as proposed in Figure 1b. Influence networks dynamically modify percepts within these systems to effect influence among their different components. The influence is designed to increase perceptual accuracy within individual perceptual channels by incorporating information from other co-occurring senses. More formally, influence networks are designed to move ambiguous perceptual inputs into easily recognized subsets of their representational spaces. In contrast with approaches taken in engineering what are typically called *multimodal systems*, influence networks are not intended to create high-level joint perceptions. Instead, they share sensory information across perceptual channels to increase local perceptual accuracy within the individual perceptual channels themselves. As we discuss in Chapter 7, this type of cross-modal perceptual reinforcement is ubiquitous in the animal world.

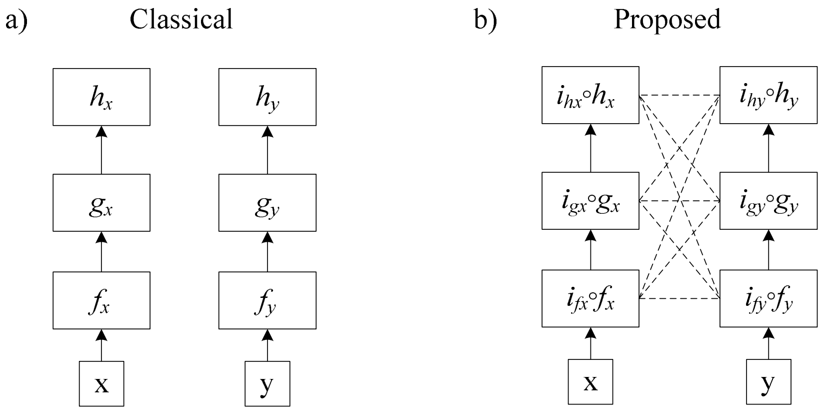

Figure 1 – Adding an influence network to two preexisting systems. We start in (a) with two pipelined networks that independently compute separate functions. In (b), we compose on each function a corresponding *influence function*, which dynamically modifies its output based on activity at the other influence functions. The interaction among these influence functions is described by an *influence network,* which is defined in Chapter 5. The parameters describing this network can be found via unsupervised learning for a large class of perceptual systems, due to correspondences in the physical events that generate the signals they perceive and to the evolutionary incorporation of these regularities into the biological sensory systems that these computational systems model. Note influence networks are distinct from an unrelated formalism called influence diagrams.

1.2 Theoretic Contributions

The work presented here addresses several important problems. From an engineering perspective, it provides a principled, neurologically informed approach to building complex, interactive systems that can learn through their own experiences. In perceptual domains, it answers a fundamental question in mathematical clustering: *how should an unknown dataset be clustered?* The connection between clustering and perceptual grounding follows from the observation that learning to perceive is learning to organize perceptions into meaningful categories. From this perspective, asking what an animal can perceive is equivalent to asking how it should cluster its sensory inputs. This thesis presents a *self-supervised* approach to this problem, meaning our sub-systems derive feedback from one another cross-modally rather than rely on an external tutor such as a parent (or a programmer). Our approach is also highly nonparametric, in that it presumes neither that the number of clusters nor their distributions are known in advance, conditions which tend to defy other algorithmic techniques. The benefits of self-supervised learning in perceptual and motor domains are enormous because engineered approaches tend to be ad hoc and error prone; additionally, in sensorimotor learning we generally have no adequate models to specify the desired input/output behaviors for our systems. The notion of *programming by example* is nowhere truer than in the developmental mimicry widespread in animal kingdom (Meltzoff and Prinz 2002), and this work is a step in that direction for artificial sensorimotor systems.

Furthermore, this thesis suggests that not only do senses influence each other during perception, which is well established, it also proposes that *perceptual channels cooperatively structure their internal representations*. This mutual structuring is a basic feature in our approach to perceptual grounding. We argue, however, that it is not simply an epiphenomenon; rather, it is a fundamental component of perception itself, because *it provides representational compatibility for sharing information cross-modally* during higher-level perceptual processing. The inability to share perceptual data is one of the major shortcomings in current engineered approaches to building interactive systems.

Finally, within this framework, we will address three questions that are basic to developing a coherent understanding of cross-modal perception. They concern both process and representation and raise the possibility that unifying (i.e. meta-level) principles might govern intersensory function:

1) Can the senses be perceptually grounded by bootstrapping off each other? Is shared experience sufficient for learning how to categorize sensory inputs?

2) How can seemingly different senses share information? What representational and computational restrictions does this place upon them?

3) Could the development of motor control use the same mechanism? In other words, can there be afferent and efferent equivalence in learning?

1.3 A Brief Motivation

The goal of this thesis is to propose a design for artificial systems that more accurately reflects how animal brains appear to process sensory inputs. In particular, we argue against *post-perceptual* integration, where the sensory inputs are separately processed in isolated, increasingly abstracted pipelines and then merged in a final integrative step as in Figure 2. Instead, we argue for *cross-modally integrated perception*, where the senses share information during perception that synergistically enhances them individually, as in Figure 1b. The main difficulty with the post-perceptual approach is that integration happens after the individual perceptions are generated. Integration occurs *after* each perceptual subsystem has already "decided" what it has perceived, when it is too late for intersensory influence to affect the individual, concurrent perceptions. This is due to information loss from both vector quantization and the explicit abstraction fundamental to the pipeline design. Most importantly, these approaches also preclude cooperative perceptual grounding; the bootstrapping provided by cross-modal clustering cannot occur when sensory systems are independent. These architectures are therefore also at odds with developmental approaches to building interactive systems.

Not only is the post-perceptual approach to integration biologically implausible from a scientific perspective, it is poor engineering as well. The real world is inherently multimodal in a way that modern artificial perceptual systems do not capture or take advantage of. Isolating sensory inputs while they are being processed prevents the lateral sharing of information across perceptual channels, even though these sensory inputs are inherently linked by the physics of the world that generates them. Furthermore, I will argue that the co-evolution of senses within an individual species provided evolutionary pressure towards representational and algorithmic compatibilities essentially unknown in modern artificial perception. These issues are examined in detail in Chapters 6 and 7.

Our work is computationally motivated by Gibson (1986), who viewed perception as an external as well as an internal event, by Brooks (1986, 1991), who elevated perception onto an equal footing with symbolic reasoning, and by Richards (1988), who described how to exploit regularities in the world to make learning easier. The recursive use of a perceptual mechanism to enable sensorimotor learning in Chapter 4 is a result of our exposure to the ideas of Sussman and Abelson (1983).

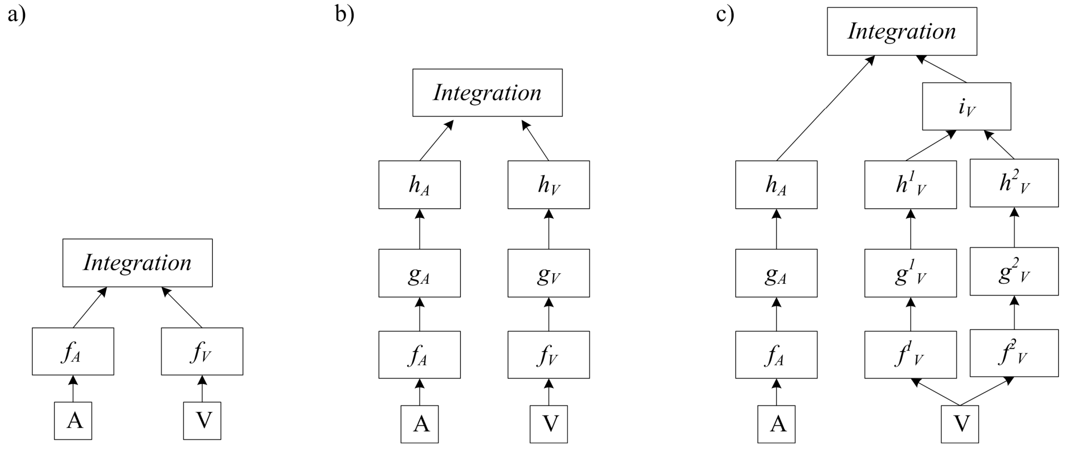

Figure 2 – Classical approaches to post-perceptual integration in traditional multimodal systems. Here, auditory (A) and visual (V) inputs pass through specialized unimodal processing pathways and are combined via an integration mechanism, which creates multimodal perceptions by extracting and reconciling data from the individual channels. Integration can happen earlier (a) or later (b). Hybrid architectures are also common. In (c), multiple pathways process the visual input and are pre-integrated before the final integration step; for example, the output of this preintegration step could be spatial localization derived solely through visual input. This diagram is modeled after (Stork and Hennecke 1996, p. 'xx').

1.4 Demonstrations

The framework and its instantiation will be evaluated by a set of experiments that explore *perceptual grounding*, *perceptual interpretation*, and *sensorimotor learning*. These will be demonstrated with:

1) **Phonetic learning**: We present a system that learns the number and formant structure of vowels (monophthongs) in American English, simply by watching and listening to someone speak and then cross-modally clustering the accumulated auditory and visual data. The system has no advance knowledge of these vowels and receives no information outside of its sensory channels. This work is the first unsupervised machine acquisition of phonetic structure of which we are aware.

2) **Speechreading**: We incorporate an *influence network* into the cross-modally clustered slices obtained in Experiment 1 to increase the accuracy of perceptual classification within the slices individually. In particular, we demonstrate the ability of influence networks to move ambiguous perceptual inputs to unambiguous regions of their perceptual representational spaces.

3) **Learning birdsong**: We will demonstrate self-supervised sensorimotor learning with a system that learns to mimic a Zebra Finch. The system is directly modeled on the dynamics of how male baby finches learn birdsong from their fathers (Tchernichovski et al. 2004, Fee et al. 2004). Our system first listens to an adult finch and uses cross-modal clustering to learn *songemes*, primitive units of bird song that we propose as an avian equivalent of phonemes. It then uses a vocalization synthesizer to generate its own nascent birdsong, guided by random exploratory motor behavior. By listening to itself sing, the system organizes the motor maps generating its vocalizations by cross-modally clustering them with respect to the previously learned *songeme* maps of its parent. In this way, it

learns to generate the same sounds to which it was previously exposed. Finally, we incorporate a standard hidden Markov model into this system, to model the temporal structure and thereby combine songemes into actual birdsong. The Zebra Finch is a particularly suitable species to use for guiding this demonstration, as each bird essentially sings a single unique song accompanied by minor variations.

We note that the above examples all use real data, gathered from a real person speaking and from a real bird singing. We also present results on a number of synthetic datasets drawn from a variety of mixture distributions to provide basic insights into the algorithms and *slice* data structure work. Finally, I believe it is possible to allow the computational side of this question to inform the biological one, and I will analyze the model, in its own right and in light of these results, to explore its algorithmic and representational implications for biological perceptual systems, particularly from the perspective of how sharing information restricts the modalities individually.

1.5 What Is a "Sense?"

Although Appendix 1 contains a glossary of technical terms, one clarification is so important that it deserves special mention. We have repeatedly used the word *sense*, e.g., sense, sensory, intersensory, etc., without defining what a *sense* is. One generally thinks of a sense as the perceptual capability associated with a distinct, usually external, sensory organ. It seems quite natural to say vision is through the eyes, touch is through the skin, etc. (Notable exceptions include proprioception – the body's sense of internal state – which is somewhat more difficult to localize and vestibular perception, which occurs mainly in the inner ear but is not necessarily experienced there.) However, this coarse definition of *sense* is misleading.

Each sensory organ provides an entire class of sensory capabilities, which we will individually call *modes*. For example, we are familiar with the *bitterness* mode of taste, which is distinct from other taste modes such as *sweetness*. In the visual system, *object* *segmentation* is a mode that is distinct from *color perception*, which is why we can appreciate black and white photography. Most importantly, individuals may lack particular modes *without other modes in that sense being affected* (e.g., Wolfe 1983), thus demonstrating they are phenomenologically independent. For example, people who like broccoli are insensitive to the taste of the chemical *phenylthiocarbamide* (Drayna et al. 2003); however, we would not say these people are unable to taste – they are simply missing an individual taste mode. There are a wide variety of visual agnosias that selectively affect visual experience, e.g., *simultanagnosia* is the inability to perform visual object segmentation, but we certainly would not consider a patient with this deficit to be blind, as it leaves the other visual processing modes intact.

Considering these fine grained aspects of the senses, we allow intersensory influence to happen between modes even within the same sensory system, e.g., entirely within vision, or alternatively, between modes in different sensory systems, e.g., in vision and audition. Because the framework presented here is *amodal*, i.e., not specific to any sensory system or mode, it treats both of these cases equivalently.

1.6 Roadmap

Chapter 2 sets the stage for the rest of this thesis by visiting an example stemming from the 1939 World's Fair. It intuitively makes clear what we mean by perceptual grounding and interpretation, which until now have remained somewhat abstract.

Chapter 3 presents our approach to perceptual grounding by introducing *slices*, a data structure for representing sensory information. We then define our algorithm for cross-modal clustering, which autonomously learns perceptual categories within slices by considering how the data within them co-occur. We demonstrate this approach in learning the vowel structure of American English by simultaneously watching and listening to a person speak.

Chapter 4 defines our architecture for sensorimotor learning, based on a Cartesian theater. Our system simultaneously "watches" its internal motor activity while it observes the effects of its own actions externally. Cross-modal clustering then allows it to ground its motor maps using previously clustered perceptual maps. This is possible because slices can equivalently contain perceptual or motor data, and in fact, slices do not "know" what kind of data they contain. The principle example in this chapter is the acquisition of species-specific birdsong.

Chapter 5 examines the temporal dynamics of perception by treating slices as phase spaces through which sensory inputs move. We define a dynamic activation model on slices and interconnect them through an *influence network*, which allows different modes to influence each other's perceptions dynamically. We then examine using this framework to disambiguate audio-visual speech inputs.

Chapter 6 examines the historic background of computational perception and provides further motivation for our approach. We also examine and critique related work in multimodal integration and perceptual categorization, outlining the similarities and differences.

Chapter 7 presents the biological motivations for this thesis, focusing on the past half-century of research in multimodal perception. We also examine several theoretical issues raised in earlier chapters and speculate on biological implications of our approach.

Chapter 8 contains a brief summary of the contributions of this thesis and outlines future work.

Chapter 2

Setting the Stage

We begin with an example to illustrate the two fundamental problems of perception addressed in this thesis:

1) *Grounding* – how are sensory inputs categorized in a perceptual system?

2) *Interpretation* – how should sensory inputs be classified once their possible categories are known?

The example presented below concerns speechreading, but the techniques presented in later chapters for solving the problems raised here are not specific to any perceptual modality. They can be applied to range of perceptual and motor learning problems, and we will examine some of their nonperceptual applications as well.

2.1 Peterson and Barney at the World's Fair

Our example begins with the 1939 World's Fair in New York, where Gordon Peterson and Harold Barney (1952) collected samples of 76 speakers saying sustained American English vowels. They measured the fundamental frequency and first three formants

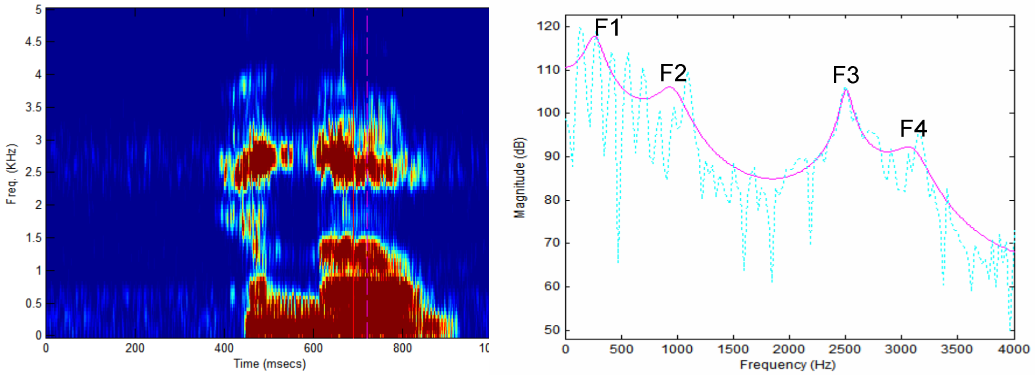

Figure 3 -- On the left is a spectrogram of the author saying, “Hello.” The demarcated region (from 690-710ms) marks the onset of phoneme /ao/, corresponding to the start of the vowel "o" in “hello.” The spectrum corresponding to this 20ms window is shown on the right. A 12th order LPC model is shown overlaid, from which the formants, i.e., the spectral peaks, are estimated. In this example: F1 = 266Hz, F2 = 922Hz, and F3 = 2531Hz. Formants above F3 are generally ignored for sound classification because they tend to be speaker dependent. Notice that F2 is slightly underestimated in this example, a reflection of the heuristic nature of formant determination.

(see Figure 3) for each sample and noticed that when plotted in various ways (Figure 4), different vowels fell into different regions of the formant space. This regularity gave hope that spoken language – at least vowels – could be understood through accurate estimation of formant frequencies. This early hope was dashed in part because co-articulation effects lead to considerable movement of the formants during speech (Holbrook and Fairbanks 1962). Although formant-based classifications were largely abandoned in favor of the dynamic pattern matching techniques commonly used today (Jelinek 1997), the belief persists that formants are potentially useful in speech recognition, particularly for dimensional reduction of data.

It has long been known that watching the movement of a speaker's lips helps people understand what is being said. (viz. Bender 1981, p41). The sight of someone's moving lips in an environment with significant background noise makes it easier to understand what the speaker is saying; visual cues – e.g., the sight of lips – can alter the signal-to-noise ratio of an auditory stimulus by 15-20 decibels (Sumby and Pollack 1954). The task of lip-reading has by far been the most studied problem in the computational multimodal literature (e.g., Mase and Pentland 1990, Huang et al. 2003, Potamianos et al.

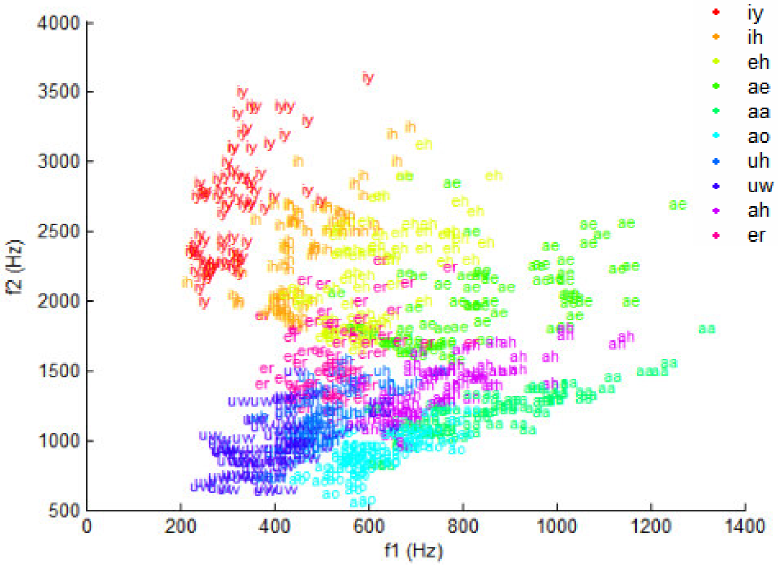

Figure 4 – Peterson and Barney Data. On the left is a scatterplot of the first two formants, with different regions labeled by their corresponding vowel categories.

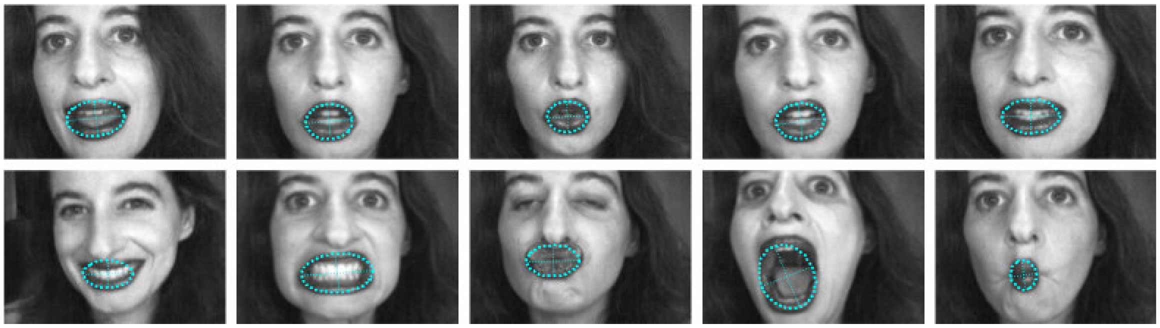

Figure 5 – Automatically tracking mouth positions of test subject in a video stream. Lip positions are found via a deformable template and fit to an ellipse using least squares. The upper images contains excerpts from speech segments, corresponding left to right with phonemes: /eh/, /ae/, /uw/, /ah/, and /iy/. The bottom row contains non-speech mouth positions. Notice that fitting the mouth to an ellipse can be non-optimal, as is the case with the two left-most images; independently fitting the upper and lower lip curves to low-order polynomials would yield a better fit. For the purposes of this example, however, ellipses provide an adequate, distance invariant, and low-dimensional model. The author is indebted to his wife for having lips that were computationally easy to detect.

2004), due to the historic prominence of automatic speech recognition in computational perception. Although significant progress has been made in automatic speech recognition, state of the art performance has lagged human speech perception by up to an order of magnitude, even in highly controlled environments (Lippmann 1997). In response to this, there has been increasing interest in non-acoustic sources of speech information, of which vision has received the most attention. Information about articulator position is of particular interest, because in human speech, acoustically ambiguous sounds tend to have visually unambiguous features (Massaro and Stork 1998). For example, visual observation of tongue position and lip contours can help disambiguate unvoiced velar consonants /p/ and /k/, voiced consonants /b/ and /d/, and nasals /m/ and /n/, all of which can be difficult to distinguish on the basis of acoustic data alone.

Articulation data can also help to disambiguate vowels. Figure 5 contains images of a speaker voicing different sustained vowels, corresponding to those in Figure 4. These images are the unmodified output of a mouth tracking system written by the author, where the estimated lip contour is displayed as an ellipse and overlaid on top of the speaker's mouth. The scatterplot in Figure 6 shows how a speaker's mouth is represented in this way, with contour data normalized such that a resting mouth configuration

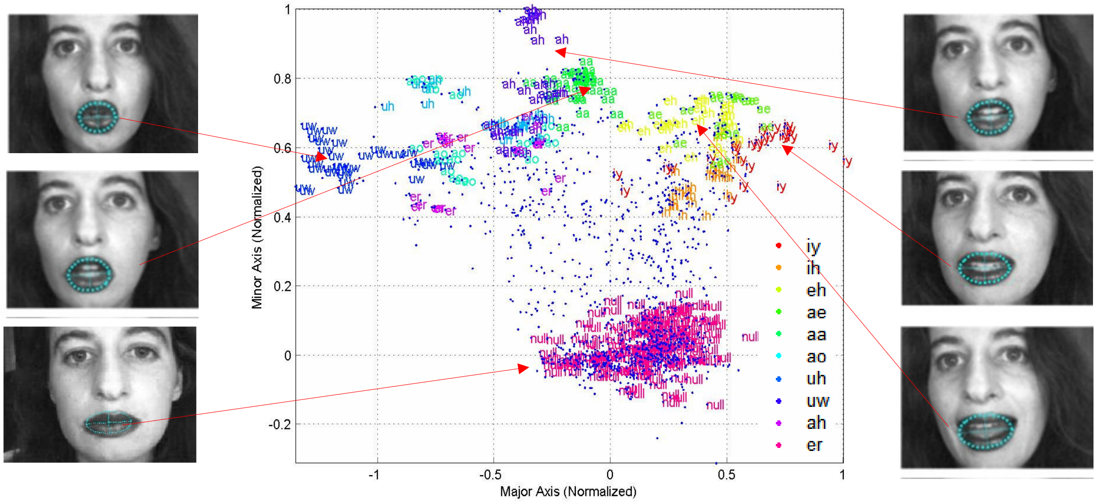

Figure 6 -- Modeling lip contours with an ellipse. The scatterplot shows normalized major (x) and minor (y) axes for ellipses corresponding to the same vowels as those in Figure 4. In this space, a closed mouth corresponds to a point labeled *null*. Other lip contours can be viewed as offsets from the null configuration and are shown here segmented by color. These data points were collected from video of this woman speaking.

(referred to as *null* in the figure) corresponds with the origin, and other mouth positions are viewed as offsets from this position. For example, when the subject makes an /iy/ sound, the ellipse is elongated along its major axis, as reflected in the scatterplot.

Suppose we now consider the formant and lip contour data simultaneously, as in Figure 7. Because the data are conveniently labeled, the clusters within and the correspondences between the two scatterplots are obvious. We notice that the two domains can mutually disambiguate one another. For example, /er/ and /uh/ are difficult to separate acoustically with formants but are easy to distinguish visually. Conversely, /ae/ and /eh/ are visually similar but acoustically distinct. Using these complementary representations, one could imagine combining the auditory and visual information to create a simple speechreading system for vowels.

2.2 Nature Does Not Label Its Data

Given this example, it may be surprising that our interest here is not in building a speechreading system. Rather, we are concerned with a more fundamental problem: how

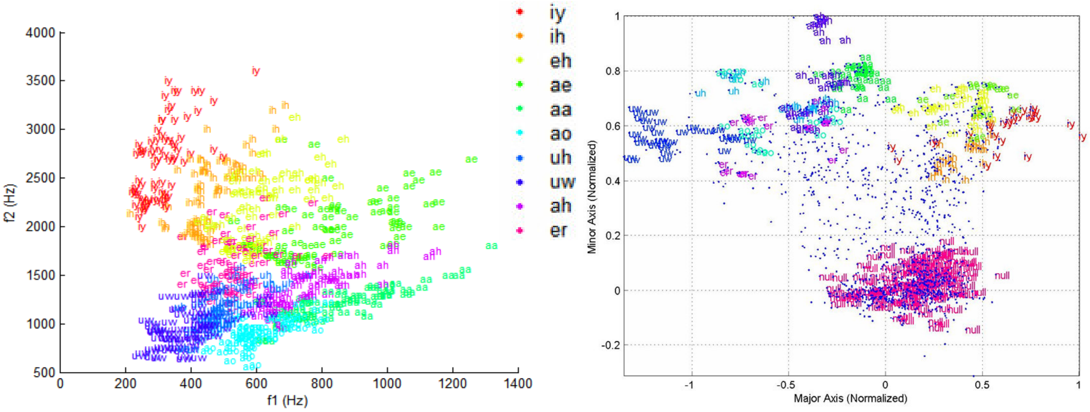

Figure 7 – Labeled scatterplots side-by-side. Formant data is displayed on the left and lip contour data is show on the right. Each plot contains data corresponding to the ten listed vowels in American English.

do sensory systems learn to segment their inputs to begin with? In the color-coded plots in Figure 7, it is easy to see the different represented categories. However, perceptual events in the world are generally not accompanied with explicit category labels. Instead, animals are faced with data like those in Figure 8 and must somehow learn to make sense of them. We want to know how the categories are learned in the first place. We note this learning process is not confined to development, as perceptual correspondences are plastic and can change over time.

We would therefore like to have a general purpose way of taking data (such as shown in Figure 8) and deriving the kinds of correspondences and segmentations (as shown in

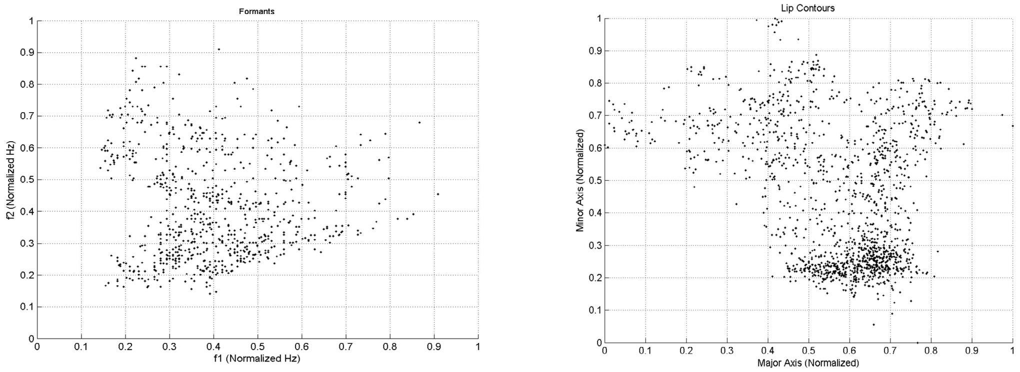

Figure 8 – Unlabeled data. These are the same data shown above in Figure 7, with the labels removed. This picture is closer to what animals actually encounter in Nature. As above, formants are displayed on the left and lip contours are on the right. Our goal is to learn the categories present in these data without supervision, so that we can automatically derive the categories and clusters such as those show above.

Figure 7) without external supervision. This is what we mean by *perceptual grounding* and our perspective here is that it is a clustering problem: animals must learn to organize their perceptions into meaningful categories. We examine below why this is a challenging problem.

2.3 Why Is This Difficult?

As we have noted above, Nature does not label its data. By this, we mean that the perceptual inputs animals receive are not generally accompanied by any meta-level data explaining what they represent. Our framework must therefore assume the learning is unsupervised, in that there are no data outside of the perceptual inputs themselves available to the learner.

From a clustering perspective, perceptual data is highly non-parametric in that both the number of clusters and their underlying distributions may be unknown. Clustering algorithms generally make strong assumptions about one or both of these. For example, the Expectation Maximization algorithm (Dempster et al. 1977) is frequently used a basis for clustering mixtures of distributions whose maximum likelihood estimation is easy to compute. This algorithm is therefore popular for clustering known finite numbers of Gaussian mixture models (e.g., Nabney 2002, Witten and Frank 2005). However, if the number of clusters is unknown, the algorithm tends to converge to a local minimum with the wrong number of clusters. Also, if the data deviate from a mixture of Gaussian (or some expected) distributions, the assignment of clusters degrades accordingly. More generally, when faced with nonparametric, distribution-free data, algorithmic clustering techniques tend not be robust (Fraley and Raftery 2002, Still and Bialek 2004).

Perceptual data are also noisy. This is due both to the enormous amount of variability in the world and to the probabilistic nature of the neuronal firings that are responsible for the perception (and sometimes the generation) of perceivable events. We will examine some of these phenomena in more detail in Chapter 6, but we note here that the brain itself introduces a great deal of uncertainty into many perceptual processes. In fact, one may perhaps view the need for high precision as the exception rather than the rule. For example, during auditory localization based on interaural time delays, highly specialized neurons known as the *end-bulbs of Held* – among the largest neuronal structures in the brain – provide the requisite accuracy by making neuronal firings in this section of auditory cortex highly deterministic (Trussell 1999). It appears that the need for mathematical precision during perceptual processing can require extraordinary neuroanatomical specialization.

Perhaps most importantly, perceptual grounding is difficult because there is no objective mathematical definition of "coherence" or "similarity." In many approaches to clustering, each cluster is represented by a prototype that, according to some well-defined measure, is an exemplar for all other data it represents. However, in the absence of fairly strong assumptions about the data being clustered, there may be no obvious way to select this measure. In other words, it is not clear how to formally define what it means for data to be objectively similar or dissimilar. In perceptual and cognitive domains, it may also depend on why the question of similarity is being asked. Consider a classic AI conundrum, "*what constitutes a chair?*" (Winston 1970, Minsky 1974, Brooks 1987). For many purposes, it may be sufficient to respond, "*anything upon which one can sit.*" However, when decorating a home, we may prefer a slightly more sophisticated answer. Although this is a higher level distinction than the ones we examine in this thesis, the principle remains the same and reminds us why similarity can be such a difficult notion to pin down.

Finally, even if we were to formulate a satisfactory measure of similarity for static data, one might then ask how this measure would behave in a dynamic system. Many perceptual (and motor) systems are inherently dynamic – they involve processes with complex, non-linear temporal behavior (Thelen and Smith 1994), as can been seen during perceptual bistability, cross-modal influence, habituation, and priming. Thus, one may ask whether a similarity metric captures a system's temporal dynamics; in a clustering domain, the question might be posed: *do points that start out in the same cluster end up in the same cluster?* We know from Lorentz (1964) that it is possible for arbitrarily small differences to be amplified in a non-linear system. It is quite plausible that a static

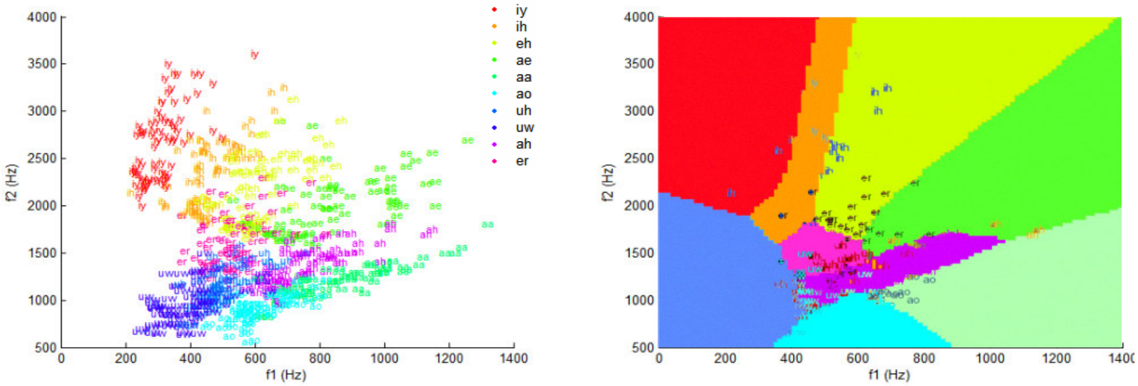

Figure 9 – On the left is a scatterplot of the first two formants, with different regions labeled by their corresponding vowel categories. The output of a backpropagation neural network trained on this data is shown on the right and displays decision boundaries and misclassified points. The misclassification error in this case is 19.7%. Other learning algorithms, e.g., AdaBoost using C4.5, Boosted stumps with LogitBoost, and SVM with a 5th order polynomial kernel, have all shown similarly lackluster performance, even when additional dimensions (corresponding to F0 and F3) are included (Klautau 2002). (Figure on right is derived from ibid. and used with permission.)

similarity metric might be oblivious to a system's temporal dynamics, and therefore, sensory inputs that initially seem almost identical could lead to entirely different percepts being generated. This issue will be raised in more detail in Chapter 5, where we will view clusters as fixed points in representational phase spaces in which perceptual inputs follow trajectories between different clusters.

In Chapter 3, we will present a framework for perceptual grounding that addresses many of the issues raised here. We show that animals (and machines) can learn how to cluster their perceptual inputs by simultaneously correlating information from their different senses, even when they have no advance knowledge of what events these senses are individually capable of perceiving. By *cross-modally* sharing information between different senses, we will demonstrate that sensory systems can be perceptually grounded by bootstrapping off each other.

2.4 Perceptual Interpretation

The previous section outlined some of the difficulties in unsupervised clustering of nonparametric sensory data. However, even if the data came already labeled and clustered, it would still be challenging to classify new data points using this information.

Determining how to assign a new data point to a preexisting cluster (or category) is what we mean by *perceptual interpretation*. It is the process of deciding what a new input actually represents. In the example here, the difficultly is due to the complexity of partitioning formant space to separate the different vowels. This 50 year old classification problem still receives attention today (e.g., Jacobs et al. 1991, de Sa and Ballard 1998, Clarkson and Moreno 1999) and Klautau (2002) has surveyed modern machine learning algorithms applied to it, an example of which is shown on the right in Figure 9.

A common way to distinguish classification algorithms is by visualizing the different spaces of possible decision boundaries they are capable of learning. If one closely examines the Peterson and Barney dataset (Figure 10), there are many pairs of points that are nearly identical in any formant space but correspond to different vowels in the actual data, at least according to the speaker's intention. It is difficult to imagine any accurate partitioning that would simultaneously avoid overfitting. There are many factors that

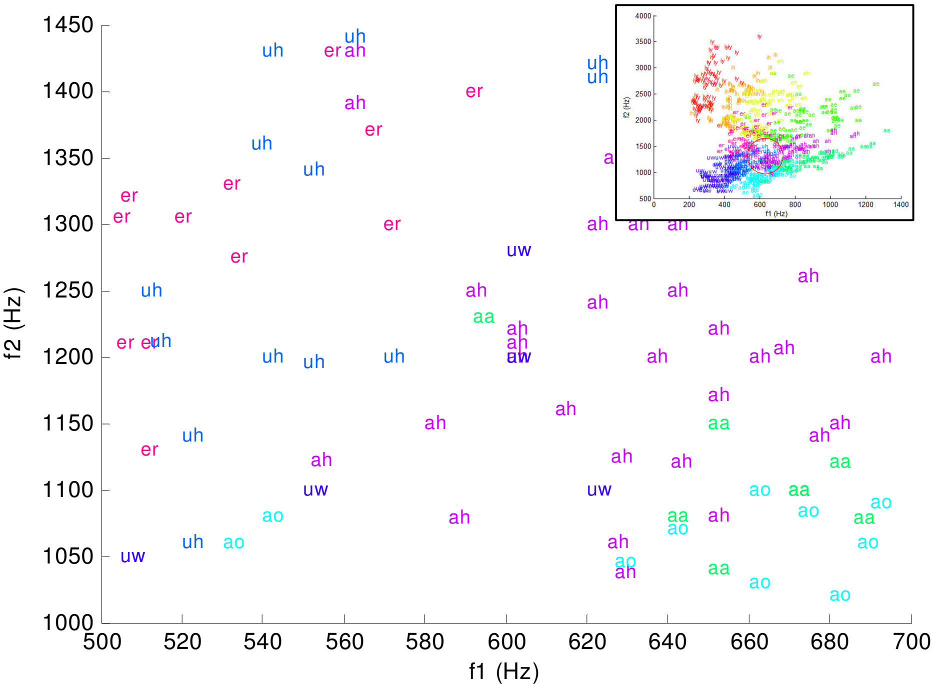

Figure 10 – Focusing on one of many ambiguous regions in the Peterson-Barney dataset. Due to a confluence of factors described in the text, the data in these regions are not obviously separable.

contribute to this, including the information loss of formant analysis (i.e., incomplete data is being classified), computational errors in estimating the formants, lack of differentiation in vowel pronunciation in different dialects of American English, variations in prosody, and individual anatomical differences in the speakers' vocal tracts. It is worth pointing out the latter three of these for the most part exist independently of how data is extracted from the speech signal and may present difficulties regardless of how the signal is processed.

The curse of dimensionality (Bellman 1961) is a statement about exponential growth in hypervolume as a function of a space's dimension. Of its many ramifications, the most important here is that many low dimensional phenomena that we are intuitively familiar with do not exist in higher dimensions. For example, the natural clustering of uniformly distributed random points in a two dimensional space becomes extremely unlikely in higher dimensions; in other words, random points are relatively far apart in high dimensions. In fact, transforming nonseparable samples into higher dimensions is a general heuristic for improving separation with many classification schemes. There is a flip-side to this high dimensional curse for us: ***low dimensional spaces are crowded***. It can be difficult to separate classes in these spaces because of their tendency to overlap. However, blaming low dimensionality for this problem is like the proverbial cursing of darkness. Cortical architectures make extensive use of low dimensional spaces, e.g., throughout visual, auditory, and somatosensory processing (Amari 1980, Swindale 1996, Dale et al. 1999, Fischl et al. 1999, Kaas and Hackett 2000, Kardar and Zee 2002, Bednar et al. 2004), and this was a primary motivating factor in the development of Self Organizing Maps (Kohonen 1984). In these crowded low-dimensional spaces, approaches that try to implicitly or explicitly refine decision boundaries such as those in Figure 10 (e.g., de Sa 1994) are likely to meet with limited success because there may be no good decision boundaries to find; perhaps in these domains, decision boundaries are the wrong way to think about the problem.

Rather than trying to improve classification boundaries directly, one could instead look for a way to move ambiguous inputs into easily classified subsets of their representational spaces. This is the essence of the *influence network* approach presented in Chapter 5 and is our proposed solution to the problem of perceptual interpretation. The goal is to use cross-modal information to "move" sensory inputs within their own state spaces to make them easier to classify. Thus, we take the view that perceptual interpretation is inherently a dynamic – rather than static – process that occurs during some window of time. This approach relaxes the requirement that the training data be separable in the traditional machine learning sense; unclassifiable subspaces are not a problem if we can determine how to move out of them by relying on other modalities, which are experiencing the same sensory events from their unique perspectives. We will show that this approach is not only biologically plausible, it is also computationally efficient in that it allows us to use lower dimensional representations for modeling sensory and motor data.

> It might be asked why we have more senses than one. [Had it been otherwise],… everything would have merged for us into an indistinguishable identity.
>
> Aristotle, *De Anima* (350 B.C.E)

Chapter 3

Perceptual Grounding

Most of the enormous variability in the world around us is unimportant. Variations in our sensory perceptions are not only tolerated, they generally pass unnoticed. Of course, some distinctions are of paramount importance and learning which are meaningful as opposed to which can be safely ignored is a fundamental problem of cognitive development. This process is a component of *perceptual grounding*, where a perceiver learns to make sense of its sensory inputs. The perspective taken here is that this is a clustering problem, in that each sense must learn to organize its perceptions into meaningful categories. That animals do this so readily belies its complexity. For example, people learn phonetic structures for languages simply by listening to them; the phonemes are somehow extracted and clustered from auditory inputs even though the listener does not know in advance how many unique phonemes are present in the signal.

Contrast this with a standard mathematical approach to clustering, where some knowledge of the clusters, e.g., how many there are or their distributions, must be known a priori in order to derive them. Without knowing these parameters in advance, algorithmic clustering techniques may not be robust (Fraley and Raftery 2002, Kleinberg 2002, Still and Bialek 2004). Assuming that in many circumstances animals cannot know the parameters underlying their perceptual inputs, how then do they learn to organize their sensory perceptions reliably?

This chapter presents an approach to clustering based on observed correlations between different sensory modalities. These cross-modal correlations exist because perceptions are created through physical processes governed by natural laws (Thompson 1917, Richards 1980, Mumford 2004). An event in the world is simultaneously perceived through multiple sensory pathways in a single observer; while each pathway may have a unique perspective on the event, their perspectives tend to be correlated by regularities in the physical world (Richards and Bobick 1988). We propose here that these correspondences play a primary role in organizing the sensory channels individually. Based on this hypothesis, we develop a new framework for grounding artificial perceptual systems.

Towards this, we will introduce a mathematical model of *slices*, which are topological manifolds that encode dynamic perceptual states and are inspired by surface models of cortical tissue (Dale et al. 1999, Fischl et al. 1999, Citti and Sarti 2003, Ratnanather et al. 2003). Slices partition perceptual spaces into large numbers of small regions (hyperclusters) and then reassemble them to construct clusters corresponding to the actual sensory events being perceived. This reassembly is performed by *cross-modal clustering*, which uses temporal correlations between slices to determine which hyperclusters within a slice correspond to the same sensory events. The cross-modal clustering algorithm does not presume that either the number of clusters in the data or their distributions is known beforehand. We examine the outputs and behavior of this algorithm on simulated datasets, drawn from a variety of mixture distributions, and on real data gathered in computational experiments.

The Simplest Complex Example

As in Chapter 2, we proceed here by first considering an example. We will return to using real datasets towards the end of this chapter, but for the moment, it is helpful to pare down the subject matter to its bare essentials.

Let us consider two hypothetical sensory modes, each of which is capable of sensing the same two events in the world, which we call the *red* and *blue* events. These two modes are illustrated below in Figure 11, where the dots within each mode represent its perceptual inputs and the blue and red ellipses delineate the two events. For example, if a "red" event takes place in the world, each mode would receive sensory input that (probabilistically) falls within its red ellipse. Notice that events within each mode

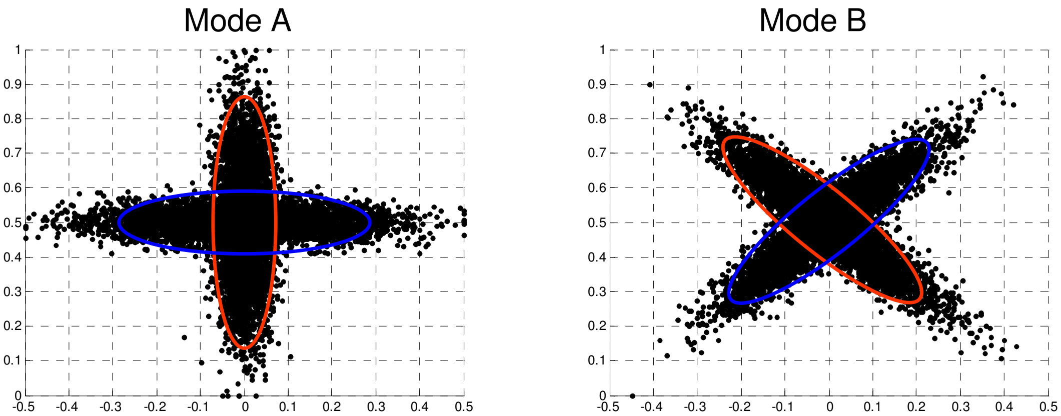

Figure 11 – Two hypothetical co-occurring perceptual modes. Each mode, unbeknownst to itself, receives inputs generated by a simple, overlapping Gaussian mixture model. To make matters more concrete, we might imagine Mode A is a simple auditory system that hears two different events in the world and Mode B is a simple visual system sees those same two events, which are indicated by the red and

overlap, and they are in fact represented by a mixture of two overlapping Gaussian distributions. We have chosen this example because it is simple – each mode perceives only two events – but it has the added complexity that the events overlap – meaning there is likely to be some ambiguity in interpreting the perceptual inputs.

Keep in mind that while *we* know there are only two events (red and blue) in this hypothetical world, the *modes* themselves do not "know" anything at all about what they can perceive. The colorful ellipses are solely for the reader's benefit; the only thing the modes receive is their raw input data. Our goal then is to learn the perceptual categories in each mode – e.g., to learn that each mode in this example senses these two overlapping events – by exploiting the temporal correlations between them.

Generating Codebooks

We are going to proceed by hyperclustering each perceptual space into a codebook. This simply means that we are going to generate far more clusters than are necessary for representing the actual number of perceptual events in the data. In this case, that would be two, but instead, we will employ a (much) larger number. For the rest of this discussion, we will refer to two different types of clusters:

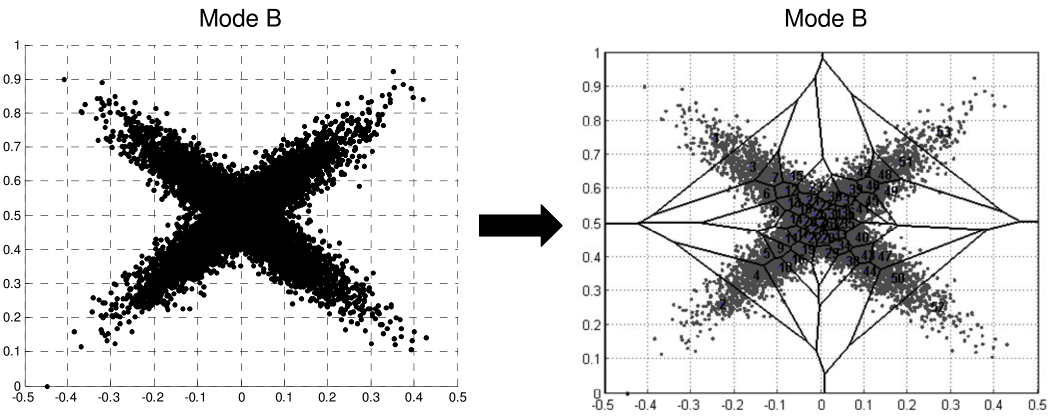

Figure 12 – Hyperclustering Mode B with the algorithm given below. Mode B is shown hyperclustered on the right. Here, we specified *k*=30 and the algorithm ended up generating 53 clusters after normalizing their densities.

1) *codebook clusters* (or *hyperclusters*) are generated by hyperclustering and are illustrated by the Voronoi regions show in Figure 12 on the right.

2) *perceptual clusters* refer to actual sensory events and are outlined with the colored ellipses in Figure 11.

Our goal will be to combine the *codebook* clusters to "assemble" the *perceptual* clusters. We note that while perceptual clustering is quite difficult, for reasons outlined in the previous chapter, hyperclustering is quite easy because there is no notion of perceptual correctness associated with it. Although we must determine how many codebook clusters to generate, we will show this number influences the amount of training data necessary rather than the correctness of the derived perceptual clusters. In other words, this approach is not overly sensitive to the hyperclustering: generating too many hyperclusters simply means learning takes longer, not that the end results are incorrect. Generating too few hyperclusters tends not to happen because of the density normalization described below. It is also sometimes possible to detect that too few clusters have been generated by using cross-modal information, a technique we examine later in this chapter.

To generate the codebooks, we will use a variant of the Generalized Lloyd Algorithm (GLA) (Lloyd 1982). We modify the algorithm to normalize the point densities within the hyperclusters, which otherwise can vary enormously. Many clustering algorithms, including GLA, optimize initially random codebooks by minimizing a strongly Euclidean distance metric between cluster centroids and their members. A cluster with a large numbers of nearby points may be viewed as equivalent to (from the perspective of the optimization) a cluster with a small number of distant points. It is therefore possible to have substantial variance in the number of points assigned to each codebook cluster. This is problematic because our approach will require that each *perceptual* cluster be represented by multiple *codebook* clusters, from which it is "assembled." The Euclidean bias introduced by the distance metric used for codebook optimization means that "small" perceptual events may be relegated to a single codebook cluster. This would prevent them from ever being detected.

There are many ways one could imagine achieving this density normalization. For example, we could explicitly add inverse cluster size to the minimization calculation performed during codebook refinement. This would leave the number of codebook clusters constant overall but introduce pressure against wide variation in the number of points assigned to each one. Rather than take an approach that preserves the overall number of clusters, we will instead modify the algorithm to recluster codebook regions that have been assigned "too many" points. This benefit of this is that we leave the GLA algorithm intact but now invoke it recursively on subregions where its performance is unsatisfactory. By keeping the basic structure of GLA, many of the mathematic properties of the generated codebooks remain unchanged. The downside of this approach is that the recursive reclustering increases the total number of generated hyperclusters. Thus, the algorithm generates at least as many codebook clusters as we specify and sometimes many more. This increase in codebook size can affect the computational complexity of algorithms operating over these codebooks, which we investigate later in this chapter. We note, however, that adding these additional clusters does not tend to require gathering more training data, an issue raised above. This is because the extra clusters are generated in regions that already have high point densities.

Our hyperclustering algorithm for generating (at least) k codebook regions over dataset $D \subseteq \mathbb{R}^N$ is:

1) Let $s = |D|/k$. This is our goal size for the number of data points per cluster.

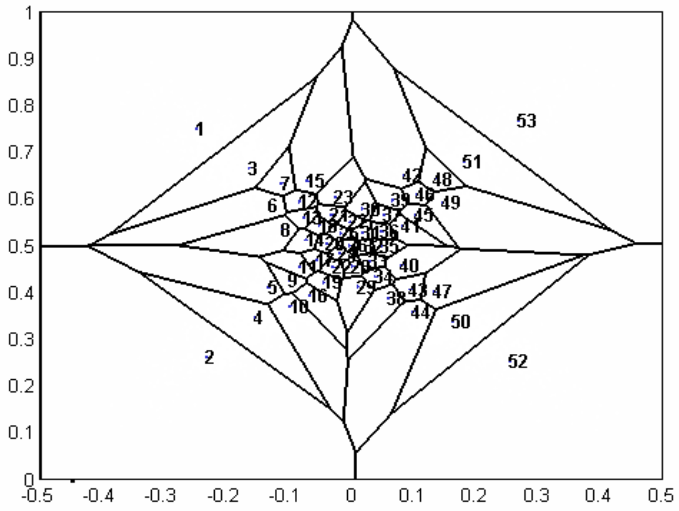

Figure 13 – The hyperclusters generated for the data in Mode B, with the data removed. The number identifying each cluster is located at its centroid. Notice how the number of clusters increases in the region corresponding to the overlap of the two Gaussian distributions, where the overall point density is highest.

2) Let $P=\{P_1,P_2,...,P_k\}$, $P_i \subset \mathbb{R}^N$ be a Lloyd partitioning of *D* over *k* clusters. This is the output of the Generalized Lloyd Algorithm.

3) For each cluster $P_i \in P$:

   If $|P_i| > s$ (the cluster has too many points), then recursively partition P_i:

   a. Let $Q=\{P_1,P_2\}$, $P_i \subset \mathbb{R}^N$ be a Lloyd partitioning of P_i over 2 clusters.

   b. Set $P=(P \cup Q)/P_i$. Add the two new partitions and remove the old one.

   End if statement

4) Repeat step 3 until no new partitions are added. Then, return the centroids of the sets in *P* as the final hyperclustering. Empirically, we find that $k<|P|<2k$.

The output of this algorithm on the data in Mode B is shown above in Figure 13. Notice how the number of clusters increases in the region corresponding to the overlap of the two Gaussian distributions, which is due to the density normalization. We note that any number of variations on this algorithm are possible. For example, in the reclustering step in (3), we might recursively generate $|P_i|/s$ rather than 2 clusters. We could also modify the goal size *s* to change the degree of density normalization. In any event, we have

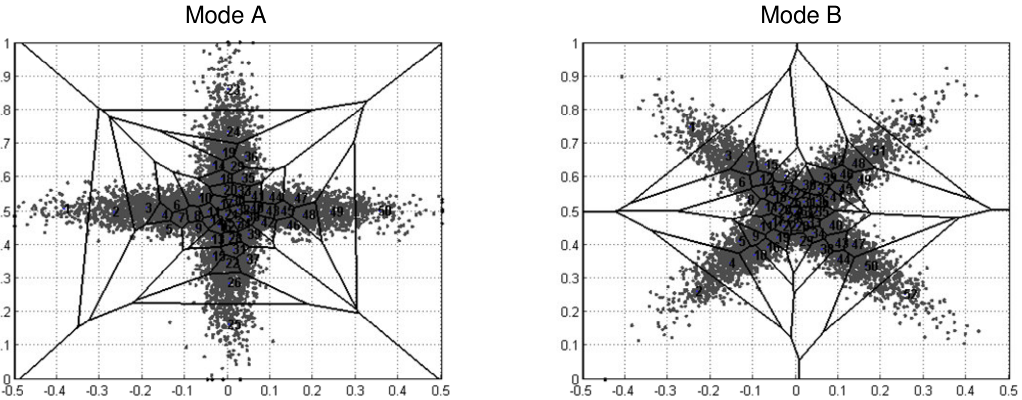

Figure 14 – Slices generated for Modes A and B using the hyperclustering algorithm in the previous section. Our goal is to combine the codebook clusters to reconstruct the actual sensory events perceived within the slices.

found that our approach is not particularly sensitive to the precise details of the codebook's generation; we confirm this statement later in this chapter, when we consider hyperclustering other mixture distributions. At present, the most important consideration is that the cluster densities are normalized, which minimizes the Euclidean bias inherent in the centroid optimization performed by the Lloyd algorithm.

Generating Slices

We now introduce a new data structure called *slices* that are constructed using the codebooks defined in the previous section. Figure 14 illustrates slices constructed for Modes A and B from our example above. Slices are topological manifolds that encode dynamic perceptual states and are inspired by surface models of cortical tissue (Citti and Sarti 2003, Ratnanather et al. 2003). They are able to represent both symbolic and numeric data and provide a natural foundation for aggregating and correlating information. Intuitively, a slice is a codebook with a non-Euclidean distance metric defined between its cluster centroids. In other words, distances within each cluster are Euclidean, whereas distances between clusters are not. A topological manifold is simply a manifold "glued" together from Euclidean spaces, and that is exactly what a slice is.

Our goal is to combine the codebook regions to "reconstruct" the larger perceptual regions within a slice. To do this, we will define a non-Euclidean distance metric

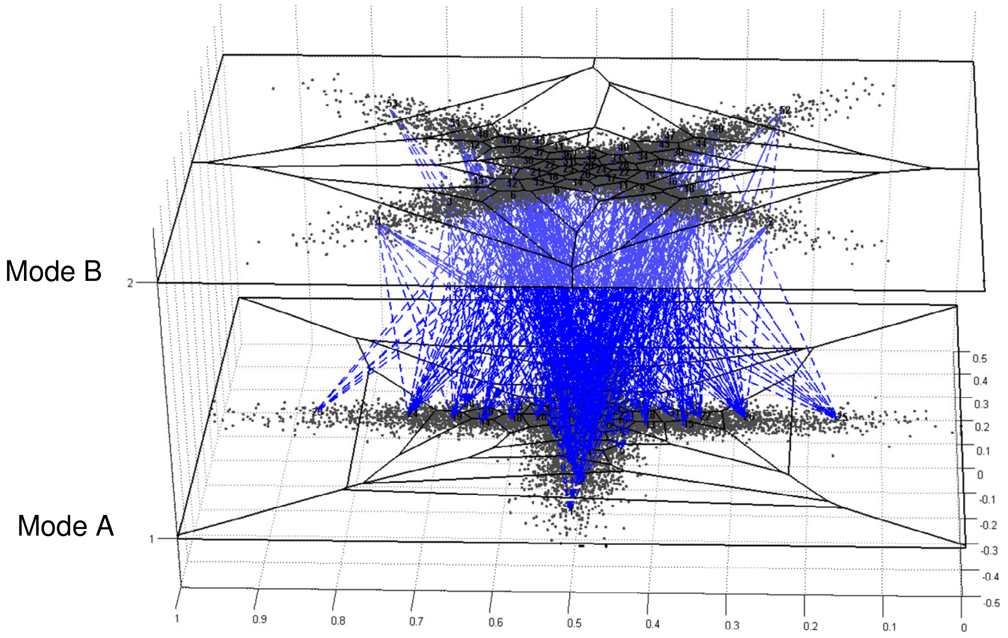

Figure 15 – Viewing Hebbian linkages between two different slices. The modes have been vertically stacked here to make the correspondences clearer. The blue lines indicate that two codebook regions temporally co-occur with each other. Note that these connections are weighted based on their strengths, which is not visually represented here, and that these weights are additionally asymmetric between each pair of connected regions.

between codebook regions that reflects how much we think they are part of the same perceptual event. In this metric, codebook regions corresponding to the same perceptual event will be closer together and those corresponding to different events will be further apart. Towards defining this metric, we first collect co-occurrence data between the codebook regions in different modes. We want to know how each codebook region in a mode temporally co-occurs with the codebook regions in other modes.

This data can be easily gathered with the classical sense of Hebbian learning (Hebb 1949), where connections between regions are strengthened as they are simultaneously active. The result of this process is illustrated in Figure 15, where the modes are vertically stacked to make the correspondences clearer. We will exploit the spatial structure of this Hebbian co-occurrence data to define the distance metric within each mode.

Hebbian Projections

In this section, we define the notion of a *Hebbian projection.* These are spatial probability distributions that provide an intuitive way to view co-occurrence relations between different slices. We first give a formal definition and then illustrate the concept visually.

Consider two slices $M_A, M_B \subseteq \mathbb{R}^n$, with associated codebooks $C_A = \{p_1, p_2, ..., p_a\}$ and $C_B = \{q_1, q_2, ..., q_b\}$, where cluster centroids $p_i, q_j \in \mathbb{R}^N$.

For some event *x*, we define $h(x) = \#$ of times event *x* occurs. Similarly, for events *x* and *y*, we define $h(x, y) =$ # of times events *x* and *y* co-occur. For example, $h(p_1)$ is the number of times inputs that belong to cluster p_1 were seen during some time period of interest. So, we see that $\Pr(x \mid y) = h(p, q) / h(p)$.

We define the *Hebbian projection* of a codebook cluster $p_i \in C_A$ onto mode M_B:

$$\vec{H}_A^B(p_i) = \left[\Pr(q_1 \mid p_i), \Pr(q_2 \mid p_i), ..., \Pr(q_b \mid p_i)\right] \tag{3.1}$$

When the modes are clear from context, we will simply refer to the projection by $\vec{H}(p_i)$.

A Hebbian projection is simply a conditional spatial probability distribution that lets us know what mode M_B probabilistically "looks" like when a region p_i is active in co-occurring mode M_A. This is visualized in Figure 16.

We can equivalently define a Hebbian projection for a region $r \subseteq M_A$ constructed out of a subset of its codebook clusters $C_r = \{p_{r1}, p_{r2}, ..., p_{rk}\} \subseteq C_A$:

$$\vec{H}_A^B(r) = \left[\Pr(q_1 \mid r), \Pr(q_2 \mid r), ..., \Pr(q_b \mid r)\right] \tag{3.2}$$

We will also define the notion of a *reverse Hebbian projection*, which projects a Hebbian projection back onto its source mode. It lets us measure – from the perspective of another modality – which other codebook regions in a slice appear similar to a reference region.

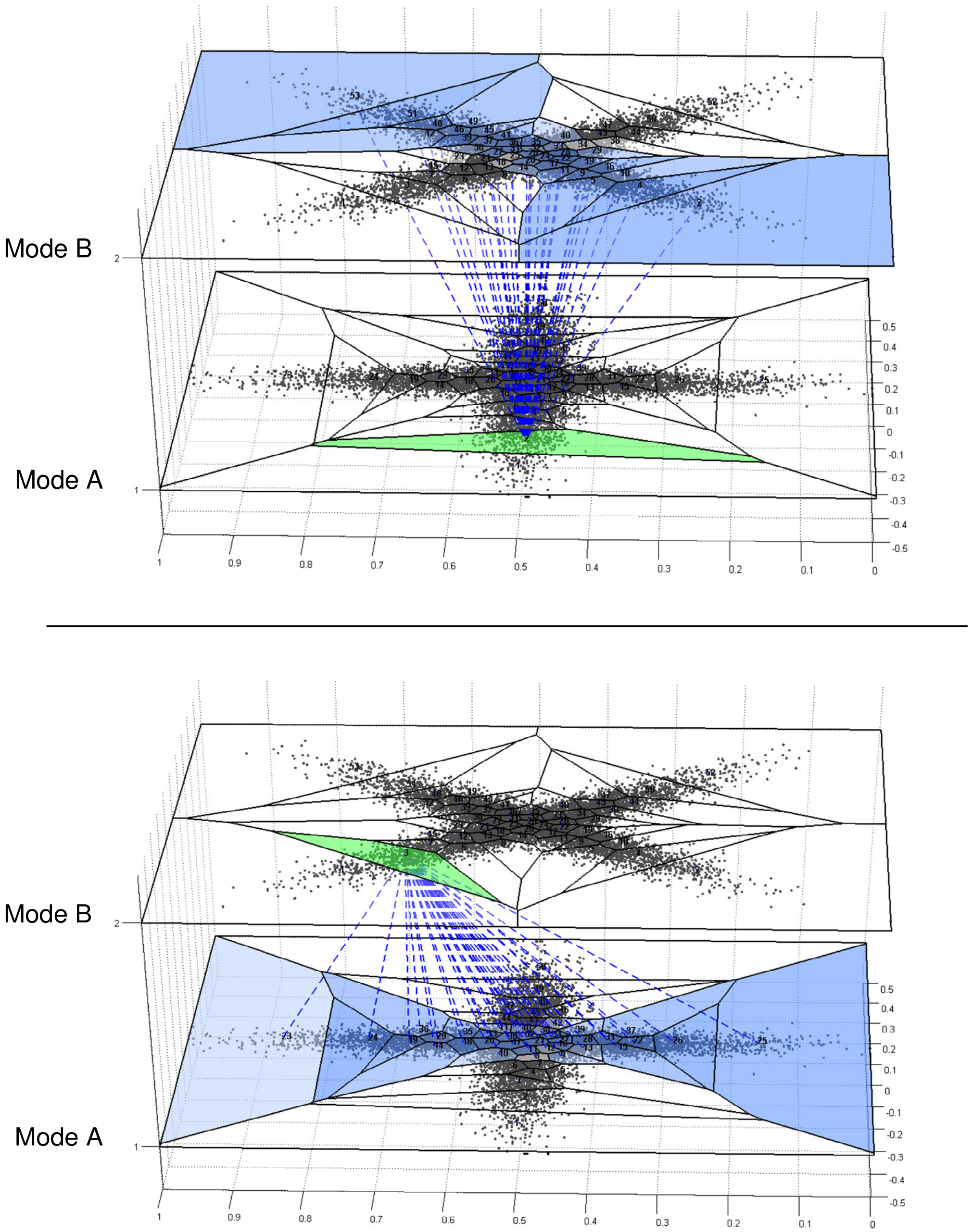

Figure 16 – A visualization of two Hebbian projections. On the top, we project from a cluster p_i in Mode A onto Mode B. The dotted lines correspond to Hebbian linkages and the blue shading in each cluster q_j in Mode B is proportional to $\Pr(q_j|p_i)$. A Hebbian projection lets us know what Mode B probabilistically "looks" like when some prototype in Mode A is active. On the bottom, we see a projection from a cluster in Mode B onto Mode A.

To do this, we first define *weighted* versions of the functions defined above for a set of weights ω. Consider a region r, $|r| = k$, where each cluster is assigned some weight ω_i. We assume that $\sum \omega_i = 1$.

$$\begin{aligned}
h_\omega(r) &= \sum_{p \in r} \omega_p h(p), \text{ where } p \text{ is a codebook cluster in region } r \\
\Pr_\omega(q,r) &= h_\omega(r,q) / h_\omega(r) = \sum_{p \in r} \omega_p h(p,q) \Big/ \sum_{p \in r} \omega_p h(p) \\
\vec{H}_\omega(r) &= \left[\Pr_\omega(q_1 \mid r), \Pr_\omega(q_2 \mid r), \ldots, \Pr_\omega(q_n \mid r)\right]
\end{aligned}$$

The reverse Hebbian projection $\widehat{H}_A^B(r)$ of a region $r \subseteq M_A$ onto mode M_B is then defined:

$$\widehat{H}_A^B(r) = \vec{H}_{\vec{H}(r)}(M_B) \tag{3.3}$$

$$= \left[\Pr_{H(r)}(p_1 \mid M_B), \Pr_{H(r)}(p_2 \mid M_B), \ldots, \Pr_{H(r)}(p_m \mid M_B)\right] \tag{3.4}$$

Again, when the modes are clear from context, we will simply refer to this as $\widehat{H}(r)$.

This distribution has a simple interpretation: the reverse Hebbian projection from mode M_A onto mode M_B for some region $r \subseteq M_A$ is the Hebbian projection of *all of mode* M_B onto mode M_A, weighted by the forward Hebbian projection of region *r*, as shown in equation (3.3). This process is visualized in Figure 17. Note that we are projecting an entire mode M_B here. This might seem initially surprising, but it simply corresponds to a projection of a region that contains all the codebook clusters for a given slice.

The reverse Hebbian projection $\widehat{H}(r)$ answers the question: *what other regions does mode M_B think region r is similar to in mode M_A?* It can therefore be viewed as a distribution that measures *cross-modal confusion.* For this reason, it provides a useful optimization tool, because we will only need to disambiguate regions that appear in each other's reverse Hebbian projections, i.e., they have a non-zero (or above some threshold) probability of being confused for one another by other modalities.

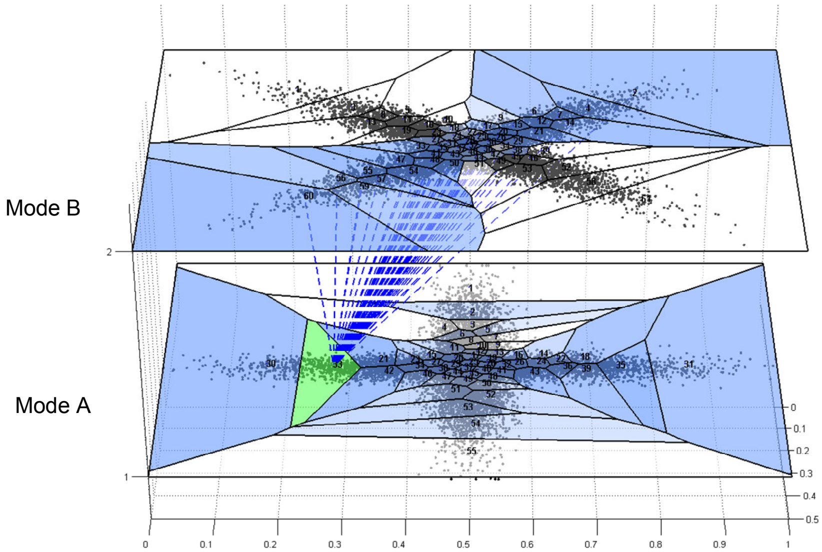

Figure 17 – Visualizing a reverse Hebbian projection. We first generate the Hebbian projection of the green cluster p_i in Mode A onto Mode B. This projection is represented by the shading of each region q_j in Mode B, corresponding to $\Pr(q_j|p_i)$. We then project all of Mode B back onto Mode A, weighting the contributions of each cluster q_i by $\Pr(q_j|p_i)$. This generates the reverse Hebbian projection, which is indicated by the shading of regions in Mode A.

Measuring Distance in a Slice

Let us briefly review where we stand at this point. We have introduced the idea of a *slice*, which breaks up a representational space into many smaller pieces that are generated by hyperclustering it. We would like to assemble these small hyperclusters into larger regions that represent actual perceptual categories present in the input data. In this section, we define the non-Euclidean distance metric between the hyperclusters that helps make this possible.

Consider the colored regions in Figure 18. We would like to determine that the blue and red regions are part of their respective *blue* and *red* events, indicated by the colored ellipses. It is important to recall that the colors here are simply for the reader's benefit. There is no labeling of regions or perceptual events within the slice itself. We will proceed by formulating a distance metric that minimizes the distance between codebook regions that are actually within the same perceptual region and maximizes the distance

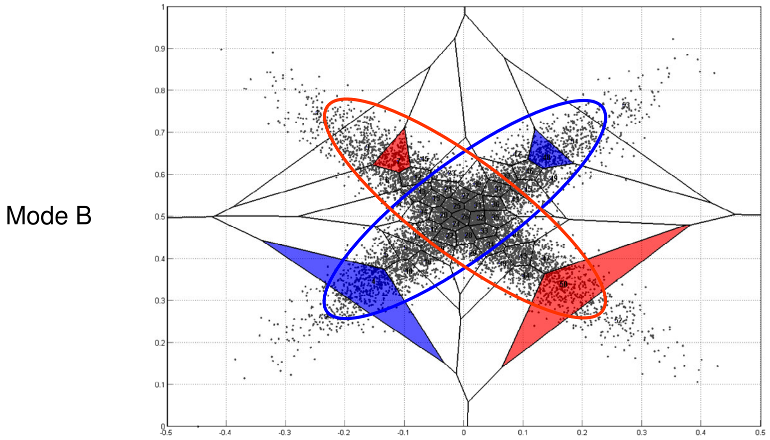

Figure 18 – Combining codebook regions to construct perceptual regions. We would like to determine that regions within each ellipse are all part of the same perceptual event. Here, for example, the two blue codebook regions (probabilistically) correspond the *blue* event and the red regions correspond to the *red* event.

between codebook regions that are in different regions. That this metric must be non-Euclidean is clear from looking at the figure. Each highlighted region is closer to one of a different color than it is to its matching partner.

We are going to use the Hebbian projections defined in the previous section to formulate this similarity metric for codebook regions. This will make the metric inherently cross-modal because we will rely on co-occurring modalities to determine how similar two regions within a slice are. Our approach is to compare codebook regions by comparing their Hebbian projections onto co-occurring slices. This process is illustrated in Figure 9.

The problem of measuring distances between prototypes is thereby transformed into a problem of measuring similarity between spatial probability distributions. The distributions are spatial because the codebook regions have definite locations within a slice, which are subspaces of $\mathbb{R}^n$. Hebbian projections are thus spatial distributions on n-dimensional data. It is therefore not possible to use one dimensional metrics, e.g., Kolmogorov-Smirnov distance, to compare them because doing so would throw away the essential spatial information within each slice.

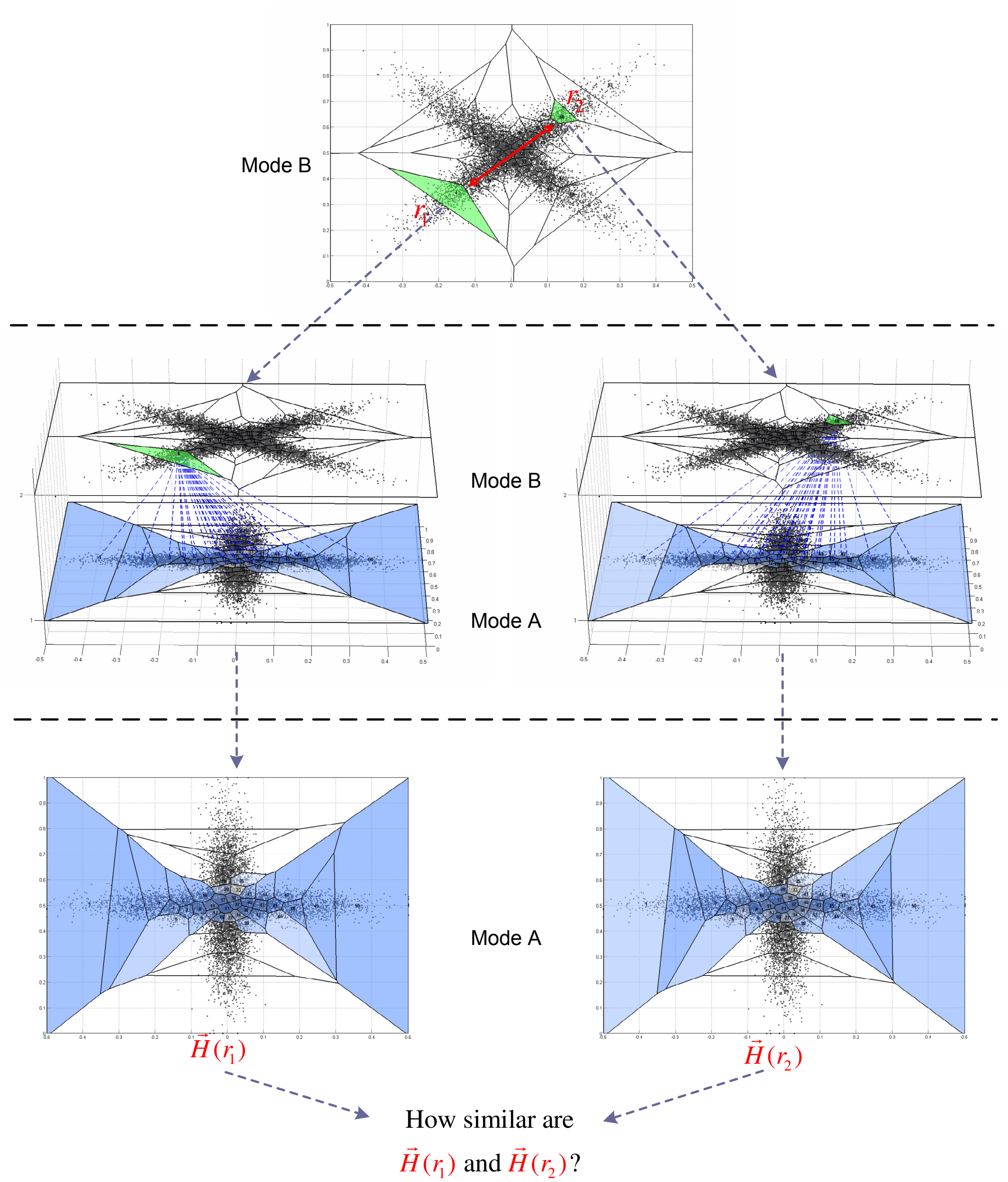

Figure 19 – Our approach to computing distances cross-modally. To determine the distance between codebook regions r_1 and r_2 in Mode B on top, we project them onto a co-occurring modality (Mode A) as shown in the middle. We then ask: *how similar are their Hebbian projections onto Mode A?*, as shown on the bottom. We have thereby transformed a question about distance between regions into a question of similarity between the spatial probability distributions provided by their Hebbian projections.

Defining Similarity

What does it mean for two things to be similar? This deceptively difficult question is at the heart of mathematical clustering and perceptual categorization and is common to a number of fields, including computer vision, statistical physics, and information and probability theory. The goal of measuring similarity between different things is often cast as a problem of measuring distances between multidimensional distributions on descriptive features. For example, in computer vision, finding minimum matchings between image feature distributions is a common approach to object recognition (Belongie et al. 2002).

In this section, we present a new metric for measuring similarity between spatial probability distributions, i.e., distributions on multidimensional metric spaces. We will use this metric to compute distances between codebook regions by comparing their Hebbian projections onto co-occurring modalities, as shown above in Figure 19. Our approach is therefore inherently multimodal – although we may be unable to determine a priori how similar two codebook regions are in isolation (i.e., unimodally), we can measure their similarity by examining how they are viewed from the perspectives of other co-occurring sensory channels. We therefore want to formulate a similarity metric on Hebbian projections that tells us not how far apart they are but rather, how similar they are to one another. This will enable perceptual bootstrapping by allowing us to answer a fundamental question:

> *Can any other modality distinguish between two regions in the same codebook? If not, then they represent the same percept.*

3.1.1 Introduction

There are a wide variety of metrics available to quantify distances between probability distributions (see the surveys in Rachev 1991, Gibbs and Su 2002). We may contrast these in many ways, including whether they are actually metrics (i.e., symmetric and satisfy the triangle inequality), the properties of their state spaces, their computational complexity, whether they admit practical bounding techniques, etc. For example, the common χ^2 distance is not a metric because it is asymmetric. In contrast, Kolmogorov-Smirnov distance is a metric but is defined only over $\mathbb{R}^1$. Choosing an appropriate metric for a given problem is a fundamental step towards solving it and can yield important insights into its internal structure.

In discussions of probability metrics, the notion of similarity generally follows directly from the definition of distance. Two distributions are deemed similar if the distance between them is small according to some metric; conversely, they are deemed dissimilar when the metric determines they are far apart. In our approach, we will reverse this dependency. We first intuitively describe our notion of similarity and then formulate a metric that computes it in a well-defined way. We call this metric the *Similarity distance* and it is the primary contribution of this section. Our approach is applicable to comparing distributions over any metric space and has a number of interesting properties, such as scale invariance, that make it additionally useful for work beyond the scope of this thesis.

3.1.2 Intuition

We begin by first examining similarity informally. Consider the two simple examples shown in Figure 20. Each shows two overlapping Gaussian distributions, whose similarity we would like to compare. Intuitively, we would say the distributions in Example A (on the left) are more similar to one another than those in Example B (on the right), because we will think of similarity as a measure of the overlap or proximity of spatial density. We are not yet concerned with formally defining similarity, but the intuition in these examples is exactly what we are trying to capture. Notice that the distributions in Example A cover roughly two orders of magnitude more area than those in Example B. Therefore, if we were to derive similarity from distance, the strong Euclidean bias incorporated into a wide variety of probability metrics would lead us to the opposite of our desired result. Namely, because the examples in B are much closer than those in A, we would therefore deem them more similar, thereby contradicting our desired meaning.

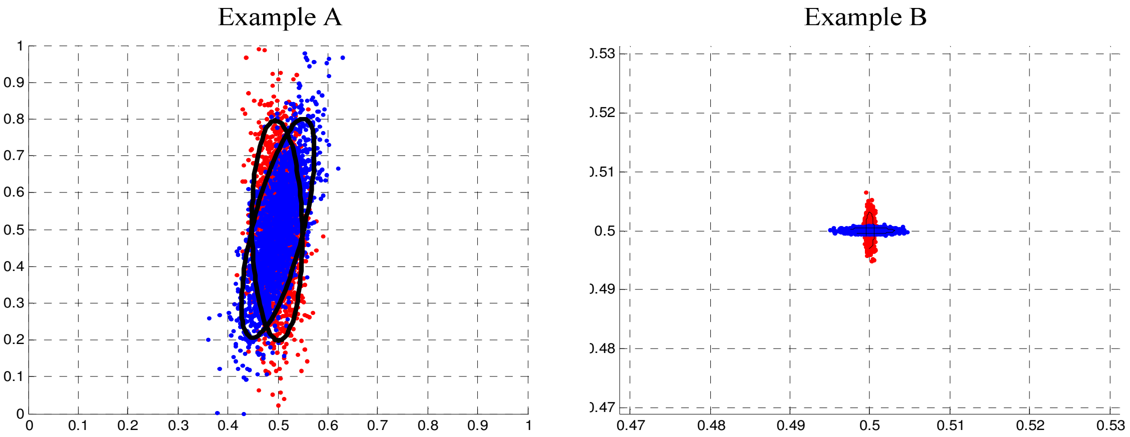

Figure 20 – Intuitively defining similarity. We consider the two distributions illustrated in Example A to be far more similar to one another than those in Example B, even though many metrics would deem them further apart due to inherent Euclidean biases. Notice that the distributions in Example A cover roughly two orders of magnitude more area than those in Example B. Note that simply normalizing the distributions before computing some metric on them would be ad hoc, very sensitive to outliers, and make common comparisons difficult.

Note that we cannot simply normalize pairs of distributions before computing some metric on them because our results would be highly sensitive to outliers. Doing so would also make common comparisons difficult, which is particularly important when demonstrating convergence in a sequence of probability measures. Finally, we want our similarity metric to be distribution-free and make no assumptions about the underlying data, which would make generalizing simple normalization schemes difficult.

3.1.3 Probabilistic Framework

We begin with some formal definitions. Our approach will be to define *Similarity distance* D_S as the ratio between two other metrics. These are the Kantorovich-Wasserstein distance and a new metric we introduce called the *one-to-many* distance. For each of these, we will provide a definition over continuous distributions and then present equivalent formulations for discrete weighted point sets. These are more computationally efficient for computing Similarity distance on the slice data structures introduced earlier. After this formal exposition, we intuitively explain and motivate these metrics in Section 3.6.4 and then show how Similarity distance is derived from them.

3.1.3.1 Kantorovich-Wasserstein Distance

Let μ and ν and be distributions on state space $\Omega = \mathbb{R}^n$. The Kantorovich-Wasserstein distance D_W (Kantorovich 1942, Gibbs and Su 2002) between μ and ν may be defined:

$$D_W(\mu,\nu) = \inf_J \{D(x,y):\ L(x)=\mu,\ L(y)=\nu\} \tag{3.5}$$

where the infimum is taken over all joint distributions *J* on *x* and *y* with marginals μ and ν respectively. For brevity, we will refer to D_W simply as the Wasserstein distance. Notice that in order to compute the Wasserstein distance, we already need to have a distance metric *D* defined to calculate the infimum. Where does *D* come from? In fact, in the approach described above, isn't *D* supposed to be the Similarity distance D_S, because we are proposing to use Similarity distance to measure distances within slices? Thus, we seem to have a chicken and egg problem from the start. We will sidestep this by defining *D* recursively through an iterative function system on D_S. This will allow us to compute Similarity distance by incrementally refining our calculation of it.

The definition in (3.5) assumes the distributions are continuous. Hebbian projections, however, are discrete distributions (i.e., weighted point sets) because they are over the codebooks within a slice. We may therefore simplify our computation by carrying it out directly over these codebooks. To do so, we define the Wasserstein distance on weighted point sets corresponding to discrete probability distributions. Consider finite sets $r_1, r_2 \subset \Omega$ with point densities φ_1, φ_2 respectively. Then we have:

$$D_W(\langle r_1,\varphi_1\rangle,\langle r_2,\varphi_2\rangle) = \inf_J \{D(x,y):\ L(x)=\langle r_1,\varphi_1\rangle,\ L(y)=\langle r_2,\varphi_2\rangle\}$$

which by (Levina and Bickel 2001) is equal to:

$$D_W(\langle r_1,\varphi_1\rangle,\langle r_2,\varphi_2\rangle) = \tfrac{1}{m}\min_{j_1,\dots,j_m}\sum_{i=1}^{m}\left[D\left(\langle r_1,\varphi_1\rangle_i,\langle r_2,\varphi_2\rangle_{j_i}\right)^2\right]^{1/2} \tag{3.6}$$

where *m* is the maximum of the sizes of r_1 and r_2, the minimum is taken over all permutations of $\{1,\dots,m\}$, and $\langle r_a,\varphi_a\rangle_i$ is the i^{th} element of set $\langle r_a,\varphi_a\rangle$. We note that (ibid.) has shown this is equivalent to the *Earth Mover's distance* (Rubner et al. 1998), a popular empirical measure used primarily in the machine vision community, when they are both computed over probability distributions.

We can now define the Wasserstein distance between Hebbian projections of $r_1, r_2 \subseteq M_A$ onto M_B as:

$$D_W\left(\vec{H}(r_1), \vec{H}(r_2)\right) = \inf_J \left\{ D(x,y) :\ L(x) = \vec{H}(r_1),\ L(y) = \vec{H}(r_2) \right\} \tag{3.7}$$

where the infimum is taken over all joint distributions J on x and y with marginals $\vec{H}(r_1)$ and $\vec{H}(r_2)$. By (3.6), we have this is equal to:

$$D_W\left(\vec{H}(r_1), \vec{H}(r_2)\right) = \tfrac{1}{m} \min_{j_1,\ldots,j_m} \sum_{i=1}^{m} \left[D\left(\vec{H}(r_1)_i, \vec{H}(r_2)_{j_i}\right)^2 \right]^{1/2} \tag{3.8}$$

where m is the number of codebook regions in M_B, the minimum is taken over all permutations of $\{1,\ldots,m\}$, and $\vec{H}(r)_i = i^{th}$ component of $\vec{H}(r)$.

We note that the Wasserstein distance presented above is not a candidate for measuring similarity. In fact, referring back to Figure 20, the red and blue distributions in Example A here are further apart as determined by the Wasserstein distance than those in Example B, i.e., $D_W(\text{Example A}) >\ D_W(\text{Example B})$, which does not capture our intended meaning of similarity.

3.1.3.2 Computational Complexity

The optimization problem in (3.6) was first proposed by Monge (1781) and is known as the Transportation Problem. It involves combinatorial optimization because the minimum is taken over $O(2^m)$ different permutations and can be solved by Kuhn's Hungarian method (1955, see also Frank 2004). However, by treating it as a flow problem, we instead use the Transportation Simplex method introduced by Dantzig (1951) and subsequently enhanced upon by Munkres (1957), which has worst case exponential time but in practice is quite efficient (Klee and Minty 1972).

To get some insight into the structure of this problem, we take a moment to examine the complexity of determining exact solutions to it according to (3.8). Although this is not necessary in practice, it is instructive to see how the choice of mixture distributions influences the complexity of the problem and the implications this has for selecting perceptual features. Notice that in the minimization in equation (3.8), the vast majority of permutations can be ignored because we only need examine regions that have non-zero probabilities in the Hebbian projections. In other words, we could choose to ignore any region $q_i \subseteq M_B$ where $\max(\vec{H}(r_1)_i, \vec{H}(r_2)_i) \leq \varepsilon$ for some small ε. A conservative approach would set $\varepsilon = 0$, however, one can certainly imagine using a slightly higher threshold to simultaneously reduce noise and computational complexity.

We may estimate the running time of calculating D_W exactly by asking how many non-zeros values we expect to find in the Hebbian projections onto mode M_B of two regions in mode M_A. Let us suppose that mode M_B actually has *d* events (of equal likelihood) distributed over *m* codebook regions. How many codebook regions are there within each event? If the events do not overlap, then we expect that each perceptual event is covered by *m/d* codebook regions, due to the density normalization performed during codebook generation. In this case, the minimization must be performed over $O(2^{1+m/d})$ permutations. Alternatively, it is possible for all of the sensory events to overlap, giving an upper bound, worst case of *m* regions per event and $O(2^m)$ running time. Thus, the running time is a function of the event mixture distributions as much as it is the number of codebooks. When Hebbian distributions are "localized," in the sense they are confined to subsets of the codebook regions, the worst-case running time is closer to $O(2^{1+m/d})$.

We can optimize the computation by taking advantage of the fact that the projections are over identical codebooks, i.e., their spatial distributions are over the same set of points generated by the hyperclustering of M_B. In the general statement of the Transportation Problem, this need not be the case. We can therefore reduce the number of codebook regions involved by removing the intersection of the projections from the calculation. Where they overlap, namely, the distribution described by a normalized

$\min\left(\vec{H}(r_1),\vec{H}(r_2)\right)$, we know the Wasserstein distance between them is 0. Therefore, let:

$$\vec{H}'(r_1) \quad = \vec{H}(r_1) - \min\left(\vec{H}(r_1),\vec{H}(r_2)\right) \tag{3.9}$$

$$\vec{H}'(r_2) \quad = \vec{H}(r_2) - \min\left(\vec{H}(r_1),\vec{H}(r_2)\right) \tag{3.10}$$

$$\Delta \qquad = 1 - \sum \min\left(\vec{H}(r_1),\vec{H}(r_2)\right) \tag{3.11}$$

We then have:

$$D_W\left(\vec{H}(r_1),\vec{H}(r_2)\right) = \Delta\, D_W\left(\vec{H}'(r_1),\vec{H}'(r_2)\right) \tag{3.12}$$

In other words, the Wasserstein distance computed over a common codebook (3.12) is equal to the distance computed on the distributions ((3.9) and (3.10)) over their non-intersecting mass (3.11). (Note that we must normalize (3.9) and (3.10) to insure they remain probability distributions, but the reader may assume this normalization step is always implied when necessary.) When the distributions overlap strongly, which we previously identified as the worst case scenario, we can typically use this optimization to cut the number of involved codebooks regions in half. When the distributions do not overlap, this optimization provides no benefit, but as we have already noted, this is a best case scenario and optimization is less necessary. As a further enhancement, we could also establish thresholds for Δ to avoid calculating D_W altogether. For example, in the case where their non-intersecting mass is extremely small, we might chose to define $D_W = 0$ or some other approximation .

In summary then, the computational complexity of exactly computing the Wasserstein distance very much rests on the selection of mixture distributions over which it is computed. These in turn depend upon the *feature selection* used in our perceptual algorithms, which directly determine the distributions of sensory data within a slice. We say that "good" features are ones that tend to restrict Hebbian projections to smaller subsets of slices and to reduce the amount of overlap among detectable perceptual events. (We suspect one can directly formulate a measure of the entropy in features based on these criteria but have not done so here.) Empirically, "good" features for computing the Wasserstein distance tend to be similar to the ones we naturally select when creating artificial perceptual systems. "Bad" features provide little information because their values are difficult to separate, i.e., they have high entropy. Later in the thesis, we will draw biological evidence for these theses idea from (Ernst and Banks 2002) .

3.1.3.3 The One-to-Many Distance

We now introduce a new distance metric called the *one-to-many* distance. Afterwards, we examine this metric intuitively and show how it naturally complements the Wasserstein distance. We will use these metrics together to formalize our intuitive notion of similarity.

Let *f* and *g* be the respective density functions of distributions μ and ν on state space $\Omega = \mathbb{R}^{\mathrm{n}}$. Then the one-to-many distance (D_{OTM}) between μ and ν is:

$$\begin{aligned} D_{OTM}(\mu,\nu) &= \int_{\mu} f(x) \cdot D_W(x,\nu)\, dx \\ &= \int_{\mu}\int_{\nu} f(x)\cdot g(y)\cdot d(x,y)\, dxdy \\ &= \int_{\nu} g(y)\cdot D_W(\mu,y)\, dy = D_{OTM}(\nu,\mu) \end{aligned}$$

We define this over weighted pointed sets as:

$$\begin{aligned} D_{OTM}\left(\langle r_1,\varphi_1\rangle,\langle r_2,\varphi_2\rangle\right) &= \sum_{p_i\in r_1} \varphi_i\, D_W\left(p_i,\langle r_2,\varphi_2\rangle\right) \\ &= \sum_{p_i\in r_1} \varphi_i \sum_{p_j\in r_2} \varphi_j\, d(p_i,p_j) \\ &= \sum_{p_j\in r_2} \varphi_j \sum_{p_i\in r_1} \varphi_i\, d(p_i,p_j) \\ &= D_{OTM}\left(\langle r_2,\varphi_2\rangle,\langle r_1,\varphi_1\rangle\right) \end{aligned}$$

The one-to-many distance is the weighted sum of the Wasserstein distances between each individual point within a distribution and the entirety of another distribution. It is straightforward to directly calculate D_W between a point and a distribution and computing D_{OTM} requires $O(m^2)$ time, where m is the maximum number of weighted points in the distributions. Also, we note from these definitions that D_{OTM} is symmetric.

From this definition, we can now formulate D_{OTM} between Hebbian projections of $r_1, r_2 \subseteq M_A$ onto M_B:

$$D_{OTM}\left(\vec{H}(r_1), \vec{H}(r_2)\right) = \sum_{i \in \vec{H}(r_1)} \omega_i \, D_W\left(i, \vec{H}(r_2)\right) \tag{3.13}$$

$$= \sum_{i \in \vec{H}(r_1)} \omega_i \sum_{j \in \vec{H}(r_2)} \omega_j D(i, j) \tag{3.14}$$

where $\omega_i = \vec{H}(r_1)_i$ and $\omega_j = \vec{H}(r_2)_j$. Recall that $\vec{H}(r)_i = i^{th}$ component of $\vec{H}(r)$.

We see this is symmetric:

$$D_{OTM}\left(\vec{H}(r_1), \vec{H}(r_2)\right) = D_{OTM}\left(\vec{H}(r_2), \vec{H}(r_1)\right)$$

The one-to-many distance is a weighted sum of the Wasserstein distances between each individual region in a projection and the other projection in its entirety. The weights are taken directly from the original Hebbian projections. This is represented by the term $\omega_i \, D_W\left(i, \vec{H}(r_2)\right)$ in (3.13). We note that the Wasserstein distance between a single region and a distribution is trivial to compute directly, as shown in (3.14), and it can be calculated in $O(m)$ time, where m is the number of codebooks in Mode B. Therefore, D_{OTM} can be computed in $O(m^2)$ time.

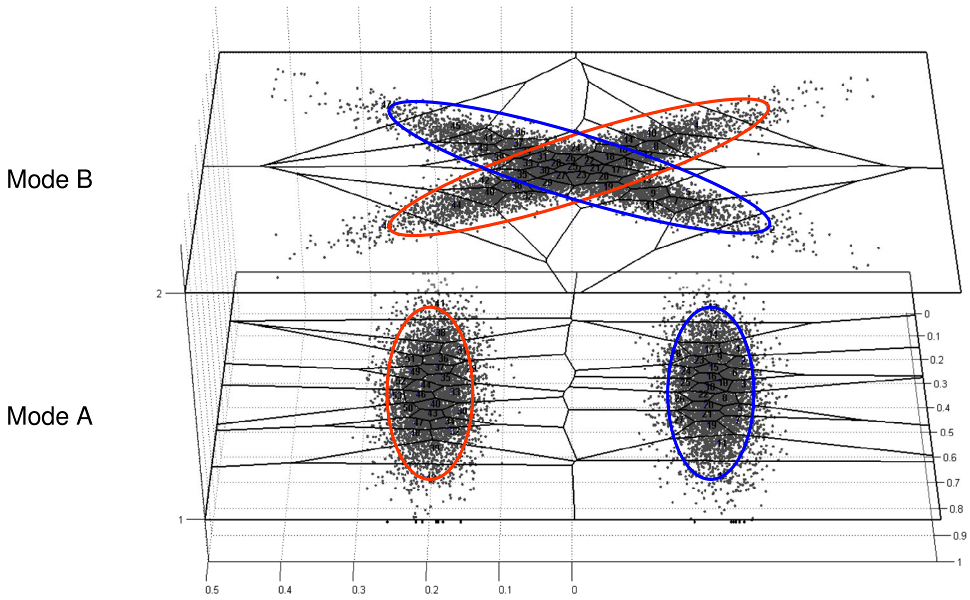

Figure 21 – A simpler example. Mode B is the same as in previous examples but the events in Mode A have been separated so they do not overlap. The colored ellipses show how the external *red* and *blue* events probabilistically appear within each mode.

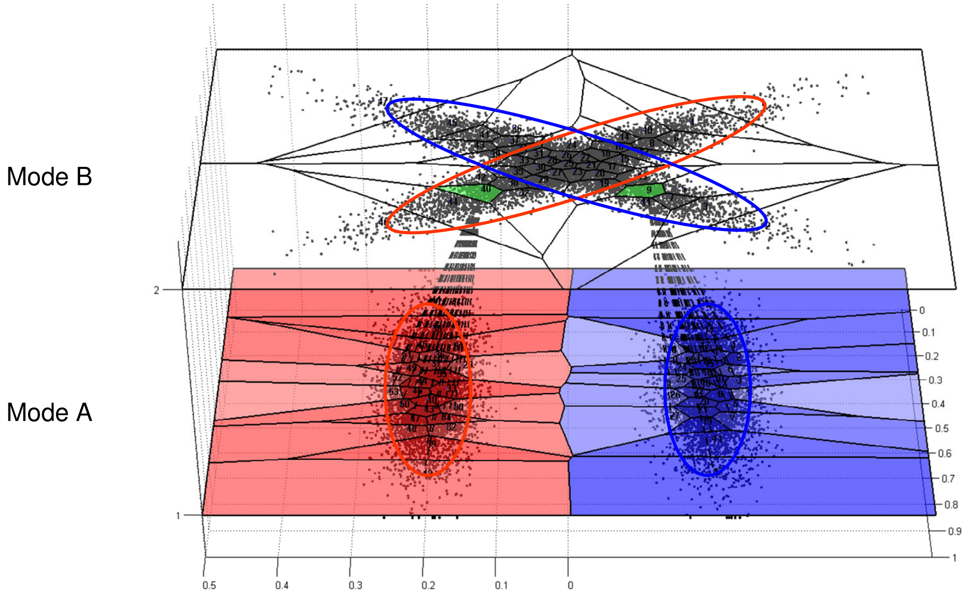

Figure 22 – Hebbian projections from regions in Mode B onto Mode A. Notice that the Hebbian projections have no overlap, which simplifies the visualization and discussion in the text. However, this does not affect the generality of the results.

3.1.4 Visualizing the Metrics: A Simpler Example

Consider the example in Figure 21. We have modified Mode A, on the bottom, so that its events no longer overlap. (Mode B on top remains unchanged.) This will simplify the presentation but does not affect the generality of the results presented here. As before, the two world events perceived by each mode are delineated with colored ellipses for the benefit of the reader but the modes themselves have no knowledge of them. The Hebbian projections from two codebook regions in Mode B are shown in Figure 22. We see this example was designed so that the projections have no overlap, making it easy to view them independently.

We now give an intuitive interpretation of the two distance metrics, D_W and D_{OTM}, based on the classic statement of the Transportation Problem (Monge 1781). This problem is more naturally viewed with discrete distributions, but the presentation generalizes readily to continuous distributions. Consider the Hebbian projections from our example in isolation, as show in Figure 23. On the left, $\vec{H}(r_1)$ is shown in red, and $\vec{H}(r_2)$ is shown in blue on the right. The shading within each Voronoi region is proportional to its weight (i.e., point density) within its respective distribution.

In the Transportation Problem, we imagine the red regions (on the left) need to deliver supplies to the blue regions (on the right). Each red region contains a mass of supplies proportional to its shading and each blue region is expecting a mass of supplies proportional to its shading. (We know that mass being "shipped" is equal to the mass being "received" because they are described by probability distributions.) The one-to-many distance is how much work would be necessary to deliver all the material from the red to blue regions, if each region had to independently deliver its mass proportionally to all regions in the other distribution. Work here is defined as $mass \times distance$.

The Wasserstein distance computes the minimum amount of work that would be necessary if the regions cooperate with one another. Namely, red regions could deliver material to nearby blue regions on behalf of other red regions, and blue regions could receive material from nearly red regions on behalf of other blue regions. Nonetheless, we maintain the restriction that each region has a maximum amount it can send or receive, corresponding to its point density. This is why the Wasserstein distance computes the solution to the Transportation Problem, which is directly concerned with this type of delivery optimization.

Thus, we may summarize that D_{OTM} computes an unoptimized Transportation Problem, where cooperation is forbidden, and that D_W computes the optimized Transportation Problem, where cooperation is required.

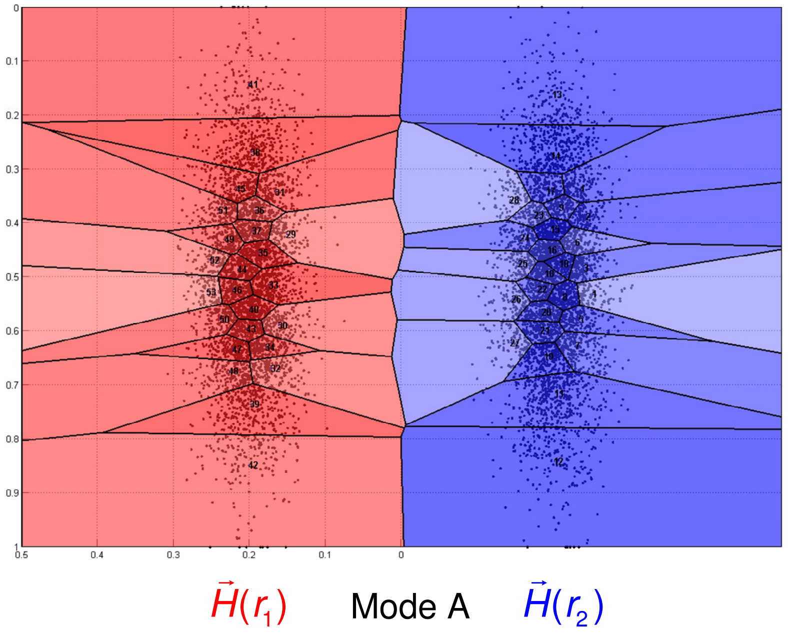

Figure 23 – Visualizing D_W and D_{OTM} through the Transportation Problem. We examine the Hebbian projections onto Mode A shown in Figure 22. $\vec{H}(r_1)$ is shown in red on the left and $\vec{H}(r_2)$ is shown in blue on the right. Each region is shaded according to its point density. In the Transportation Problem, we want to move the "mass" from one distribution onto the other. If we define $work = mass \times distance$, then D_{OTM} computes the work required if each codebook region must distribute its mass proportionally to all regions in the other distribution. D_W computes the work required if the regions cooperatively distribute their masses, to minimize the total amount of work required.

3.1.5 Defining Similarity

We are now in a position to formalize the intuitive notion of similarity presented above. We define a new metric called the *Similarity distance* (D_S) between continuous distributions μ and ν:

$$D_S(\mu,\nu)=\frac{D_W(\mu,\nu)}{D_{OTM}(\mu,\nu)}$$

and over weighted point sets:

$$D_S(\langle r_1,\varphi_1\rangle,\langle r_2,\varphi_2\rangle)=\frac{D_W\left(\langle r_1,\varphi_1\rangle,\langle r_2,\varphi_2\rangle\right)}{D_{OTM}\left(\langle r_1,\varphi_1\rangle,\langle r_2,\varphi_2\rangle\right)}$$

We thereby define the Similarity distance between Hebbian projections of $r_1,r_2\subseteq M_A$ onto M_B:

$$D_S\left(\vec{H}(r_1),\vec{H}(r_2)\right)=\frac{D_W\left(\vec{H}(r_1),\vec{H}(r_2)\right)}{D_{OTM}\left(\vec{H}(r_1),\vec{H}(r_2)\right)} \tag{3.15}$$

The Similarity distance is the ratio of the Wasserstein to the one-to-many distance. It measures the optimization gained when transferring the mass between two spatial probability distributions if cooperation is allowed. Intuitively, it normalizes the Wasserstein distance. It is scale invariant (see Figure 24) and captures our desired notion of similarity.

An important note to avoid confusion: Because D_S is a distance measure based on similarity – and not a similarity measure – *it is smaller for things that are more similar and larger for things that are less similar*. So, for any distribution ν, $D_S(\nu,\nu)=0$, expressing the notion that anything is (extremely) similar to itself.

We briefly examine the behavior of D_S at and in between its limits. Let $\vec{H}(r_1)$ and $\vec{H}(r_2)$ be identical Hebbian projections separated by some distance Δ. Then we have the following properties:

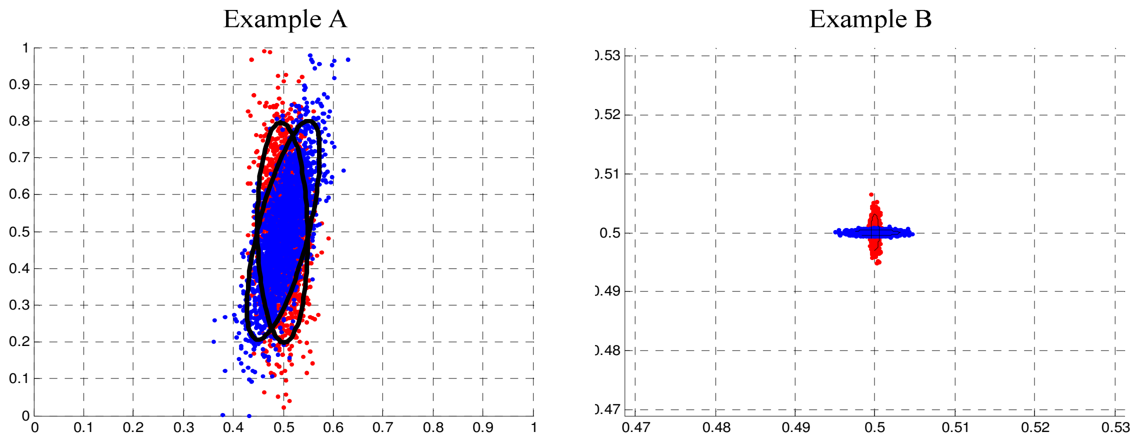

Figure 24 – Examining Similarity distance. Comparing the distributions in the two examples, we have $D_S\text{(Example A)} \ll D_S\text{(Example B)}$, which captures our intuitive notion of similarity.

1) We see that as the distributions are increasingly separated, the optimization provided in the Wasserstein calculation disappears:

$$\lim_{\Delta\to\infty} D_W\left(\vec{H}(r_1),\vec{H}(r_2)\right) = D_{OTM}\left(\vec{H}(r_1),\vec{H}(r_2)\right)$$

and therefore:

$$\lim_{\Delta\to\infty} D_S\left(\vec{H}(r_1),\vec{H}(r_2)\right) = 1$$

2) As the distributions are brought closer together, the Wasserstein distance decreases much faster than the One-To-Many distance:

$$\frac{\partial}{\partial\Delta} D_W\left(\vec{H}(r_1),\vec{H}(r_2)\right) > \frac{\partial}{\partial\Delta} D_{OTM}\left(\vec{H}(r_1),\vec{H}(r_2)\right)$$

3) As they approach, it eventually dominates the calculation:

$$\lim_{\Delta\to 0} D_W(\vec{H}(r_1),\vec{H}(r_2)) = 0 \text{ and therefore, } \lim_{\Delta\to 0} D_S(\vec{H}(r_1),\vec{H}(r_2)) = 0$$

4) So, we see that $0 \le D_S(\vec{H}(r_1),\vec{H}(r_2)) \le 1$ and D_S varies non-linearly between these limits.

On the next two pages, we visualize the dependence of D_S on the distance Δ and the angle θ between pairs of samples drawn from different distributions. We examine samples drawn from Gaussian and Beta distributions in Figure 25 and Figure 26 respectively.

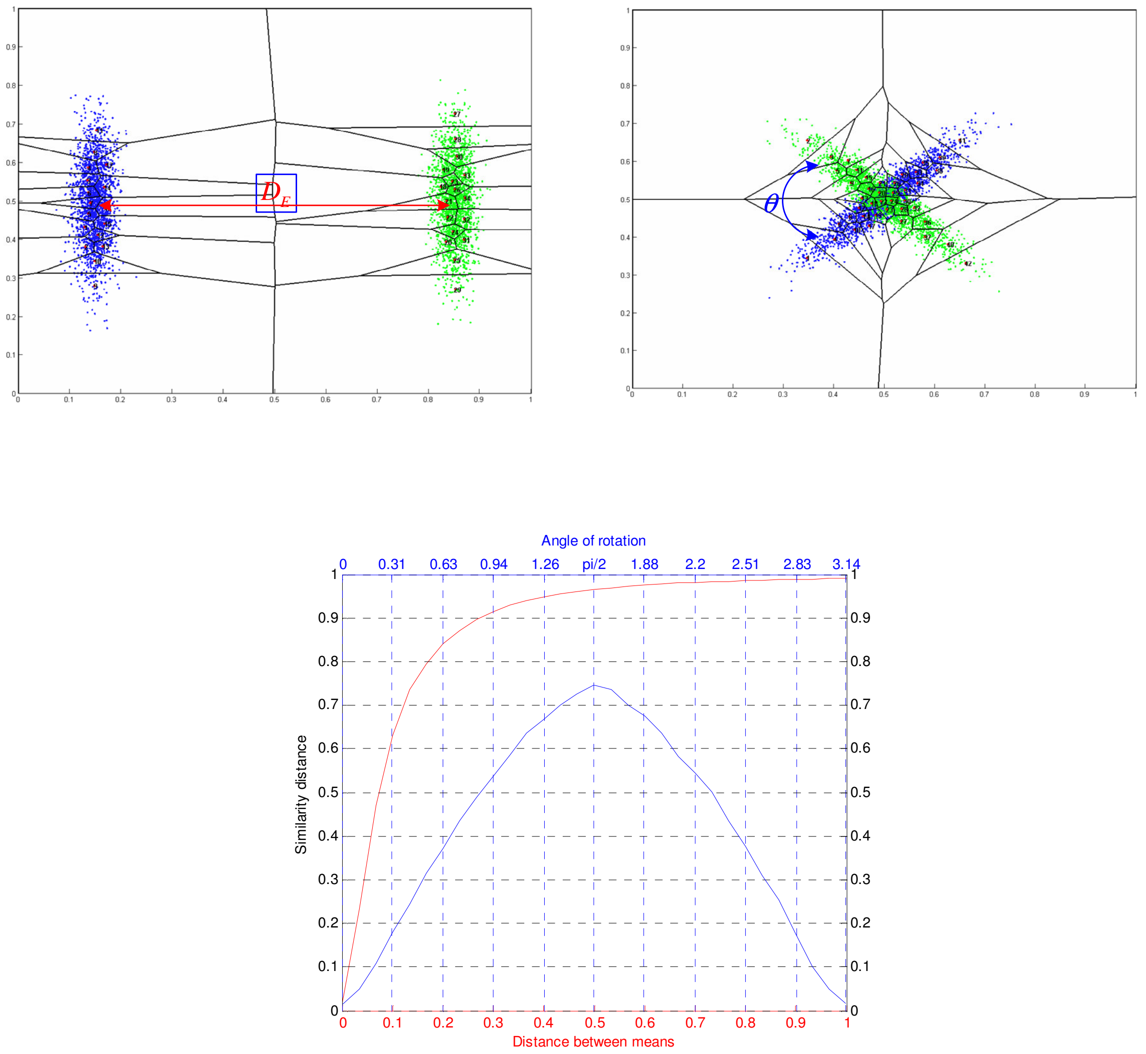

Figure 25 – Effects on D_S as functions of the distance D_E and angle θ between Gaussian distributions. As the distance D_E or angle between θ two Gaussian distributions decreases, we see how their corresponding similarity distance D_S decreases non-linearly in the graph on the bottom. D_E is shown in red and θ is shown in blue.

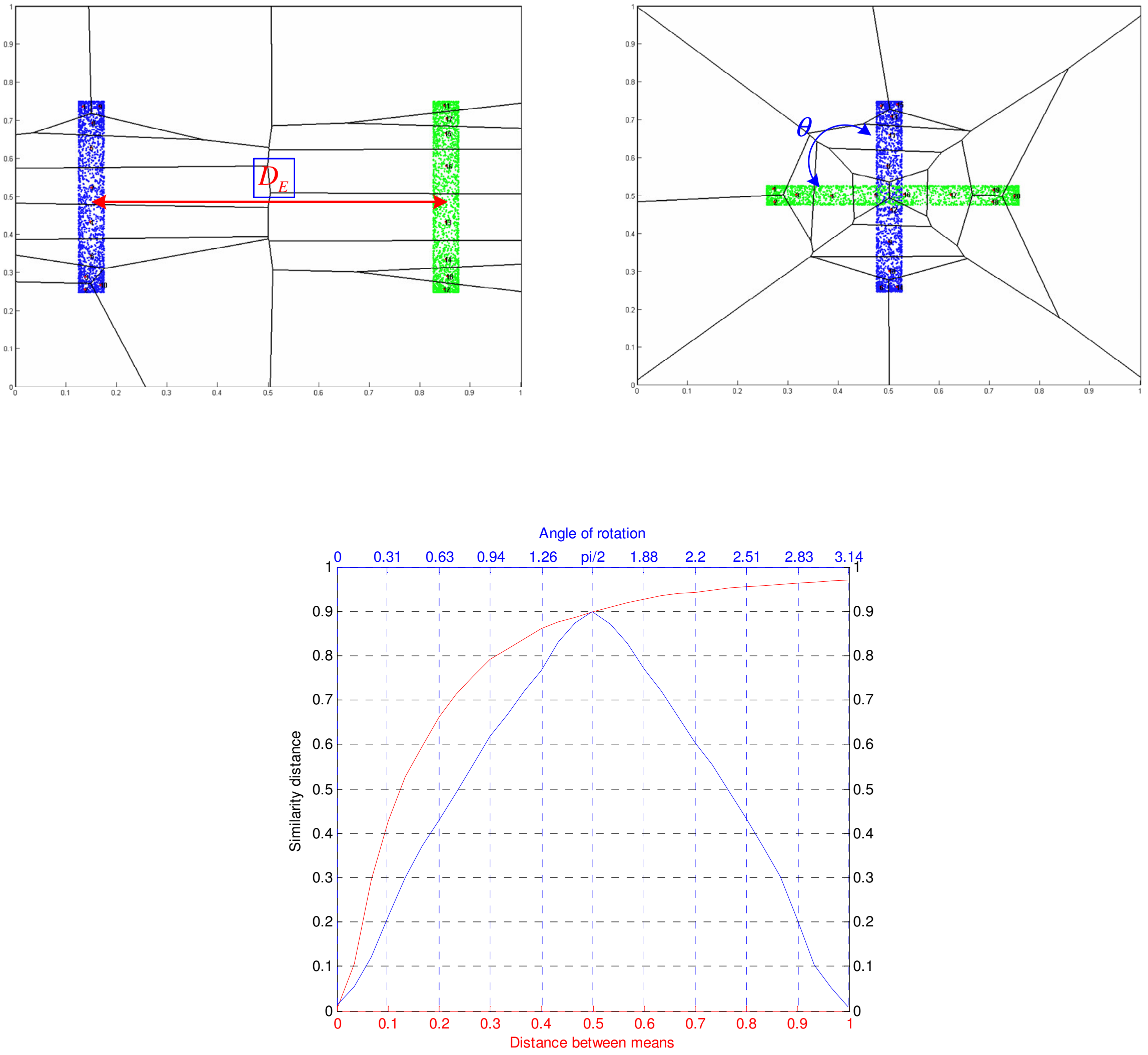

Figure 26 –Effects on D_S as functions of the distance D_E and angle θ between Beta distributions. As the distance D_E or angle between θ the two Beta distributions decreases, we see how their corresponding similarity distance D_S decreases non-linearly in the graph on the bottom. D_E is shown in red and θ is shown in blue.

3.1.5.1 A Word about Generality

We can use Similarity distance to compare arbitrary discrete spatial probability distributions. Several such comparisons are illustrated in Figure 27. These examples are important, because our being able to compute Similarity distances between these distributions means that we will be able to perceptually ground events drawn from their mixtures. That all our examples have so far involved mixtures of Gaussians has simply been for convenience. We will demonstrate later in this chapter that we can separate events corresponding to a wide assortment of mixture distributions such as the ones shown here.

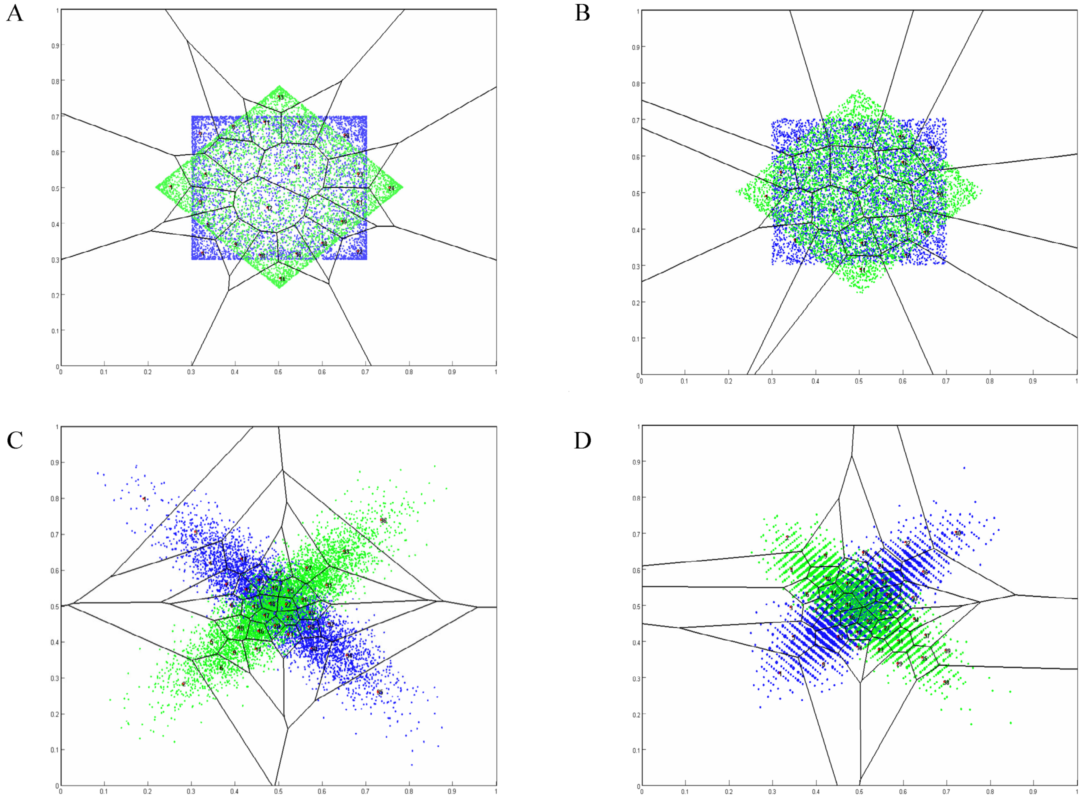

Figure 27 – Comparing spatial probability distributions. In each slice, the green and blue points represent samples drawn from equivalent but rotated 2-dimensional distributions. For each example, we identify the source distribution and the Similarity distance between the green and blue points. (A) A 2-D beta distributions with a, b = 4. Note the low density of points in the center of the distributions and the corresponding sizes of the codebook regions. $D_S = 0.22$. (B) A 2-D uniform distribution. $D_S = 0.11$. (C) A 2-D Gaussian distribution with $\sigma = (.15, .04)$. $D_S = 0.67$. (D) A 2-D Poisson distribution, with $\lambda = (50,30)$ and scaled by (.8, .3). $D_S = .55$.

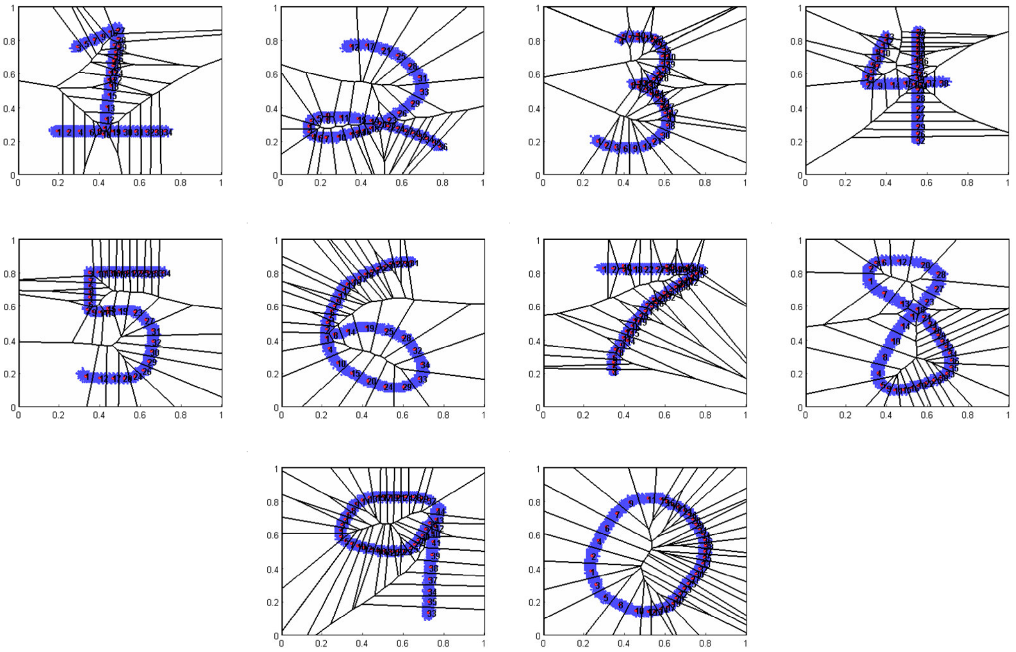

Figure 28 – Some familiar non-parametric probability distributions within codebooks. We explore hand writing recognition using D_S in Appendix 2.

We can also apply the Similarity distance to non-parametric distributions. For example, consider the familiar distributions shown in Figure 28. In Appendix 2, we will investigate using D_S for handwriting recognition and examine some of the interesting properties of codebooks constructed on contours.

Defining the Distance Between Regions

We now use Similarity distance to define the *Cross-Modal distance* (D_{CM}) between two regions $r_1, r_2 \in M_A$ with respect to mode M_B:

$$D_{CM}(r_1, r_2) \quad = \left[(1-\lambda)\left[D_E(r_1, r_2)\right]^2 \; + \; \lambda \left[\sqrt{2}\; D_S(\vec{H}(r_1), \vec{H}(r_2))\right]^2 \right]^{1/2} \tag{3.16}$$

$$= \left[(1-\lambda)\; \left[D_E(r_1, r_2)\right]^2 + 2\lambda \left(\frac{D_W\left(\vec{H}(r_1), \vec{H}(r_2)\right)}{D_{OTM}\left(\vec{H}(r_1), \vec{H}(r_2)\right)} \right)^2 \right]^{1/2} \tag{3.17}$$

where D_E is Euclidean distance and λ is the relative importance of cross-modal to Euclidean distance. Thus the distance between two regions within a slice is defined to have some component $(1-\lambda)$ of their Euclidean distance and some component (λ) of the Similarity distance between their Hebbian projections. This is illustrated in Figure 29.

In almost all uses of cross-modal distance in this thesis, we set $\lambda = 1$ and ignore Euclidean distance entirely. However, in some applications, e.g., hand-writing or drawing recognition, spatial locality within a slice is important because it is a fundamental component of the phenomenon being recognized. If so, we can use a lower of λ. Determining the proper balance between Euclidean and Similarity distances is an empirical process for such applications.

So far, we have only considered two co-occurring modes simultaneously to keep the examples simple. However, it is straightforward to generalize the definition to incorporate additional modalities, and the calculation scales linearly with the number of modalities involved. To define the cross-modal distance (D_{CM}) between two regions $r_1, r_2 \in M_A$ with respect to a set of co-occurring modes $M_I \in \mathrm{M}$, we define:

$$D_{CM}(r_1, r_2) = \left[(1-\lambda_E)\left[D_E(r_1, r_2)\right]^2 \; + \; 2\sum_{M_I \in \mathrm{M}} \lambda_I \; \left[\; D_S(\vec{H}_A^I(r_1), \vec{H}_A^I(r_2))\right]^2 \right]^{1/2} \quad (3.18)$$

where the contributions of each mode M_I is weighted by λ_I and we set $\lambda_E + \sum \lambda_I = 1$. For guidance in setting the values of the λ_I, we can turn to (Ernst and Banks 2002), who found that in intersensory influence, people give preference to senses which minimize the variance in joint perceptual interpretations, confirming an earlier prediction by (Welch and Warren 1986) about sensory dominance during multimodal interactions. This lends credence to our hypothesis in section 3.6.3.2 regarding the computational value of entropy minimization in the selection of perceptual features. We reexamine these issues in the dynamic model presented in Chapter 4.

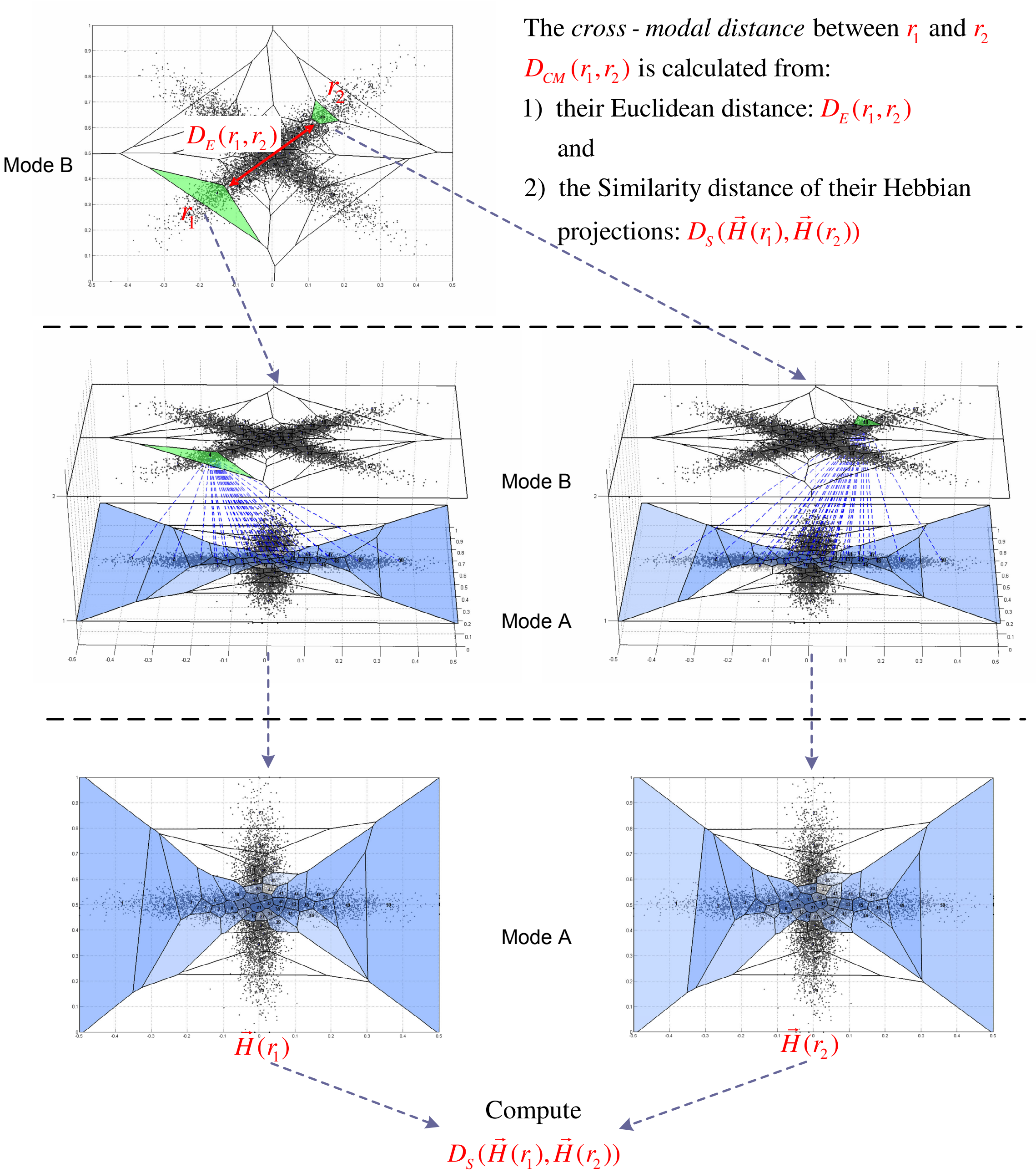
The *cross - modal distance* between r_1 and r_2
$D_{CM}(r_1, r_2)$ is calculated from:
1) their Euclidean distance: $D_E(r_1, r_2)$
and
2) the Similarity distance of their Hebbian projections: $D_S(\vec{H}(r_1), \vec{H}(r_2))$
Mode B
$D_E(r_1, r_2)$
r_1
r_2
Mode B
Mode A
Mode A
$\vec{H}(r_1)$
$\vec{H}(r_2)$
Compute
$D_S(\vec{H}(r_1), \vec{H}(r_2))$

Figure 29 – Calculating the *cross-modal distance* between codebook regions in a slice. The distance is a function of their local Euclidean distance and the how similar they appear from the perspective of a co-occurring modality. To determine this for regions r_1 and r_2 in Mode B on top, we project them onto Mode A, as shown in the middle. We then compute the Similarity distance of their Hebbian projections, as shown on the bottom.

3.1.6 Defining a Mutually Iterative System

In this section, we show how to use the cross-modal distance function defined above to calculate the distances between regions within a slice. This statement may seem surprising. Why is any elaboration required to use D_{CM}, which we just defined? There are two remaining issues we must address:

1) We have yet to specify the distance function *D* used to define the Wasserstein distance in equations (3.7) and (3.8), which was also "inherited" in our definition of the one-to-many distance in equation (3.14).

2) By defining distances cross-modally, we have created a mutually recursive system of functions. Consider any two regions r_1, r_2 in mode M_A. When we calculate $D_{CM}(r_1, r_2)$, we are relying on knowing the distances between regions within another mode M_B, which are used to calculate $D_S\left(\vec{H}(r_1), \vec{H}(r_2)\right)$. However, the distances between regions in M_B are calculated exactly the same way but with respect to M_A. So, every time we calculate distances in a mode, we are implicitly changing the distances within every other mode that relies upon it. And of course, this means its own inter-region distances may change as a result! How do we account for this and how do we know such a system is stable?

We will approach both of these issues simultaneously. Suppose we parameterize the distance function *D* in all of our definitions:

$$D_W\left(\vec{H}(r_1), \vec{H}(r_2), D\right) = \tfrac{1}{m} \min_{j_1,\ldots,j_m} \sum_{i=1}^{m} \left[D\left(\vec{H}(r_1)_i, \vec{H}(r_2)_{j_i}\right)^2 \right]^{1/2} \tag{3.19}$$

$$D_{OTM}\left(\vec{H}(r_1), \vec{H}(r_2), D\right) = \sum_{i \in \vec{H}(r_1)} \omega_i \, D_W\left(i, \vec{H}(r_2), D\right) \tag{3.20}$$

$$D_S\left(\vec{H}(r_1),\vec{H}(r_2),D\right) \quad = \frac{D_W\left(\vec{H}(r_1),\vec{H}(r_2),D\right)}{D_{OTM}\left(\vec{H}(r_1),\vec{H}(r_2),D\right)} \tag{3.21}$$

$$D_{CM}\left(r_1,r_2,D\right) = \left[(1-\lambda)\left[D_E(r_1,r_2)\right]^2 + 2\lambda\left[D_S(\vec{H}(r_1),\vec{H}(r_2),D)\right]^2\right]^{1/2} \tag{3.22}$$

We now define an iterative function system on modes M_A and M_B that *mutually* calculates D_{CM} over their regions:

Let $\Delta_t^X = D_{CM}$ in mode M_X at time *t*. Recall that D_E is Euclidean distance.

For all pairs of regions $r_i, r_j \in M_A$ and $q_i, q_j \in M_B$, we define:

$$\Delta_0^A(r_i,r_j) \quad = D_E(r_i,r_j) \tag{3.23}$$

$$\Delta_0^B(q_i,q_j) \quad = D_E(q_i,q_j) \tag{3.24}$$

$$\Delta_t^A(r_i,r_j) \quad = D_{CM}\left(r_i,r_j,\Delta_{t-1}^B\right) \tag{3.25}$$

$$\Delta_t^B(q_i,q_j) \quad = D_{CM}\left(r_i,r_j,\Delta_{t-1}^A\right) \tag{3.26}$$

Thus, we are start by assuming in (3.23) and (3.24) that the distances between regions in a slice are Euclidean, in the absence of any other information. (We later eliminate this assumption in the intermediate steps of cross-modal clustering, where we have good estimates on which to base the iteration.) The iterative steps are shown in (3.25) and (3.26) , where at time *t*, we recalculate the distances within each slice based upon the distances in the other slice at time $t-1$. For example, notice how the definition of $\Delta_t^A(r_i,r_j)$ calculates D_{CM} using Δ_{t-1}^B in (3.25). After all pairs of distances have been computed at time *t*, we can then proceed to compute them for time $t+1$. As we did in equation (3.18), we can easily generalize this system to include any number of mutually recursive modalities. The complexity again scales linearly with the number of modalities involved.

We stop the iteration when Δ_t^A and Δ_t^B begin to converge, which empirically tends to happen very quickly. Thus, we stop iterating on mode M_X at time *t* when:

$$\max_{r_i, r_j \in M_X} \frac{\left|\Delta_t^X(r_i, r_j) - \Delta_{t-1}^X(r_i, r_j)\right|}{\Delta_t^X(r_i, r_j)} < \kappa, \text{ for } \kappa = .9, \text{ we typically have } t \leq 4.$$

We will refer to this final value of Δ_t^X for any regions $r_i, r_j \in M_X$ as $\tilde{D}_{CM}\left(r_i, r_j\right)$.

With this, we can complete our formal definition of the *slice* data structure. The final component necessary for specifying the topological manifold defined by a slice was the non-Euclidean distance metric between the hyperclustered regions. We now define this distance to be $\tilde{D}_{CM}$.

Cross-Modal Clustering

Recall that our goal has been to combine codebook regions to "reconstruct" the larger perceptual regions within a slice. The definition of the iterated cross-modal distance $\tilde{D}_{CM}$ in the previous section allows us to proceed, because it suggests how to answer the following fundamental question:

> *Can any other modality distinguish between two regions in the same codebook? If not, then they represent the same percept.*

Because $\tilde{D}_{CM}$ represents the distance between two regions *from the perspective of other modalities*, we will use it to define a metric that determines whether to combine them or not. If $\tilde{D}_{CM}\left(r_i, r_j\right)$ is sufficiently small for two regions $r_1, r_2 \subseteq M_A$, then we will say they are *indistinguishable* and therefore part of the same perceptual event. If $\tilde{D}_{CM}\left(r_i, r_j\right)$ between two regions is large, we will say they are *distinguishable* and therefore, cannot be part of the same perceptual event. These criteria suggest the general structure of our cross-modal clustering algorithm. One important detail remains: how small must $\tilde{D}_{CM}\left(r_i, r_j\right)$ be for us to say it is "sufficiently" small? How do we define the threshold for merging two regions? An earlier version of this work appears in (Coen 2005).

3.1.7 Defining Self-Distance

We define the notion of *self-distance*, which measures the internal value of $\tilde{D}_{CM}$ within an individual region. Thus, rather than measure the distance between two different regions, which has been our focus so far, *self-distance* measures the internal cross-modal distance between points within a single region. It is a measure of internal coherence and will allow us to determine whether two different regions are sufficiently similar to merge. We first define self-distance within an individual codebook cluster and then generalize this for regions composed of these clusters.

Consider a slice M_A with associated codebook $C_A = \{p_1, p_2, ..., p_a\}$, generated from training dataset $T \subseteq \mathbb{R}^N$. Note the points in T represent the perceptual inputs to the slice that must be gathered before a codebook can be generated – it is not possible to hypercluster a slice that has no data within it. Let T_i be the Voronoi partitioning of T with respect to the codebook clusters in C_A; in other words, each T_i contains the points in the training dataset that are assigned to the cluster defined by p_i.

We are going to further partition each T_i into two sets, T_i^+ and T_i^-, by fitting a linear orthogonal regression onto it. For $T \subseteq \mathbb{R}^N$, this will generate an $(N-1)$-dimensional hyperplane that divides T_i into two sets, T_i^+ and T_i^-, minimizing the perpendicular distances from them to the hyperplane. Note that because the data are drawn from independent distributions, there is no error-free predictor dimension that generates the other dimensions according to some function. This is equivalent to the case where all variables are measured with error, and standard least squares techniques do not work in this circumstance. We therefore perform principal components analysis on T_i and generate the hyperplane by retaining its $N-1$ principal components. This computes what is known as the orthogonal regression and works even in cases where all the data in T_i are independent.

We use this hyperplane to partition T_i into T_i^+ and T_i^-, as shown in Figure 30. We define the *self-distance* (D_{self}) of codebook cluster p_i:

$$D_{self}(p_i) = \tilde{D}_{CM}\left(T_i^+, T_i^-\right) \tag{3.27}$$

The co-occurrence data for T_i^+ and T_i^- are collected simultaneously with that of their parent region p_i. They are gathered with respect to whole codebook regions in other slices — not with respect to those region's internal partitions. Therefore, there is minimal overhead in gathering Hebbian co-occurrence data for these subclusters and doing so

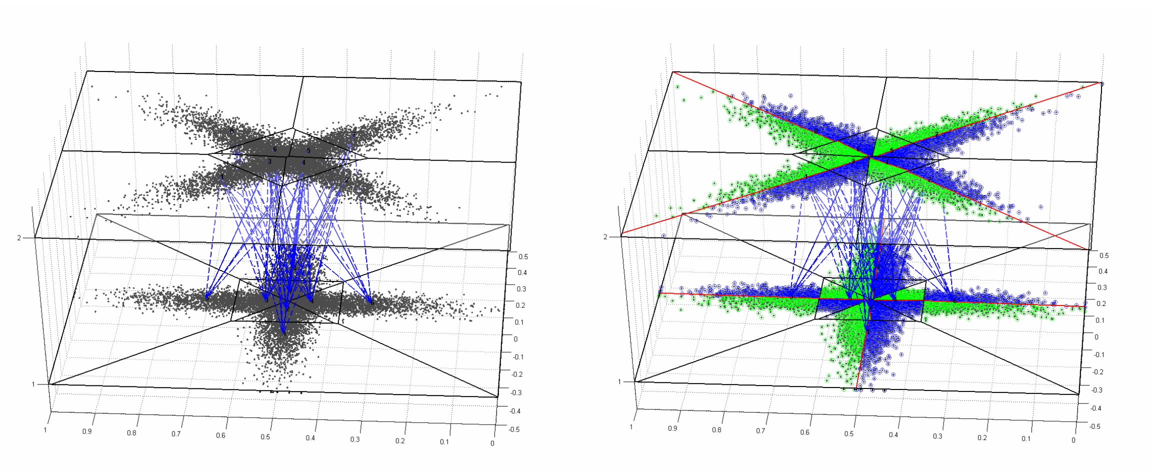

Figure 30 -- Visualizing the subpartitioning of codebook regions. The display on the right shows the internal partitioning of codebook regions in the slices on the left. Each cluster is divided into two components T_i^+ and T_i^- which are arbitrarily colored blue or green. The line that partitions each cluster is shown in red.

does not impact the computational complexity of this framework. The self-distance of a region is simply the weighted sum of the self-distances of its codebook clusters.

For a region $r = \{p_a, ..., p_b\}$, a subset of codebook clusters, let φ_i be the relative weight of each $p_i \in r$, e.g., $\varphi_i = |T_i| / \sum_{p_j \in r} |T_j|$. We define the *self-distance* of region r:

$$D_{self}(r) = \sum_{p_i \in r} \phi_i \, D_{self}(p_i) \tag{3.28}$$

3.1.8 A Cross-Modal Clustering Algorithm

We now present an algorithm for combining codebook clusters into regions that represent the sensory events within a slice. This is done in a greedy fashion, by combining the closest regions according to $\tilde{D}_{CM}$ within each slice. We use the definition of *self-distance* to derive a threshold for insuring regions are sufficiently close to merge. Afterwards, we examine the algorithm and some examples of its output.

Cross-Modal Clustering:

Given: A set of slices M and λ, the parameter for weighting Euclidean to Similarity distances. For each slice $M_i \in M$, we will call its codebook $C_i = \left\{ p_1, ..., p_{k_i} \right\}$.

Initialization: For each slice $M_i \in M$, initialize a set of *regions* $R_i = C_i$. Each slice will begin with a set of regions based on its codebook. We will merge these regions together in the algorithm below.

Algorithm:

```
Calculate $\tilde{D}_{CM}$ over the slices in set M.

While (true) do:
    Calculate $\tilde{D}_{CM}$ over the slices in set M. Use current $\tilde{D}_{CM}$ as t=0 value
    For each slice $M_i \in M$:
        Sort the pairs of regions in $M_i$, $r_a, r_b \in R_i$, by $\tilde{D}_{CM}(r_a, r_b)$
            For each pair $r_a, r_b \in R_i$, in sorted order:
                If $D_{self}(r_a) + D_{self}(r_b) > \tilde{D}_{CM}(r_a, r_b)$:
                    Merge($r_a, r_b$)
                    Exit inner for loop.

                For each codebook cluster $p_i$ in $C_i$:
                    Let $r = \min_{r \in R_i} \arg\left[ \tilde{D}_{CM}(p_i, r) \right]$
                    Move $p_i$ into region $r$

        If no regions were merged in any slice
            Either wait for new data or stop

Procedure Merge($r_a, r_b$):
    $r_a = r_a \cup r_b$.
    $R_i = R_i / r_b$.
    For all $p_i, p_j \in r_a$, set $\tilde{D}_{CM}(p_i, p_j) = 0$.
```

The cross-modal clustering algorithm initially creates a set of regions in each slice corresponding to its codebook. The goal is to merge these regions based on their cross-modal distances. The algorithm proceeds in a two-step greedy fashion:

1) For each slice, consider its regions in pairs, sorted by $\tilde{D}_{CM}$. If we find two regions satisfying $D_{self}\left(r_a\right)+D_{self}\left(r_b\right)>\tilde{D}_{CM}\left(r_a,r_b\right)$, we merge them and move onto the next step.

2) If as a result of this merger, some cluster p_i is now closer to another region, we simply move it there.

When we merge two regions, we set the pairwise distances between all codebooks clusters within them to 0, because we now view them as all part of the same underlying perceptual event and therefore equivalent to one another. At the end of each loop, we recompute $\tilde{D}_{CM}$ using the current value as the starting point in the iteration, which propagates the effects of mergers to the other slices in M. In the event no mergers are made in any slices, we can choose to either wait for new data, which will update the Hebbian linkages, or we can terminate the algorithm, if we assume sufficient training data has already been collected.

Most clustering techniques work by iteratively refining a model subject to an optimization constraint. The iterative refinement in our algorithm occurs in the recalculation of $\tilde{D}_{CM}$, which is updated after each round of mergers within the slices. This spreads the effect of a merger within a slice by changing the Similarity distances between Hebbian projections onto it. This in turn changes the distances between regions in other slices, as discussed in section 3.7.1. The optimization constraint is that we do not merge regions whose internal self-distances sum to less than their $\tilde{D}_{CM}$ distance. If we think of self-distance as measure of how much variation is permitted with an individual region, we only merge regions when the sum of their internal variations can account for their distance, as specified by the condition $D_{self}\left(r_a\right)+D_{self}\left(r_b\right)>\tilde{D}_{CM}\left(r_a,r_b\right)$.

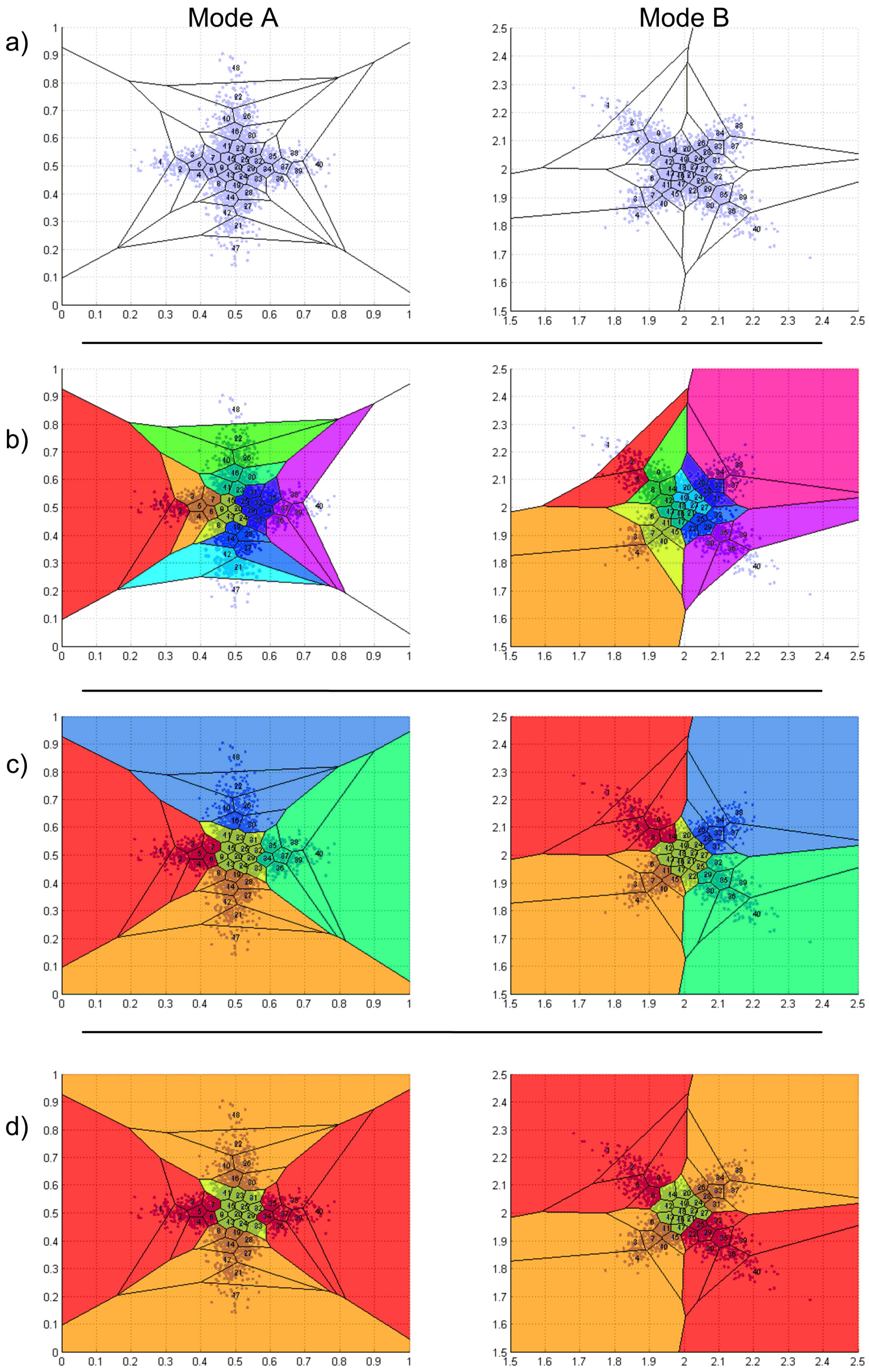

Figure 31 – The progression of the cross-modal clustering algorithm. (A) shows the initial codebook creation in each slice. (B) and (C) show intermediate region formation. (D) shows the correctly clustered outputs, with the confusion region between the categories indicated by the yellow region in the center. Note in this example, we set $\lambda = .7$ to make region formation easier to see by favoring spatial locality. The final clustering was obtained by setting $\lambda = 1$.

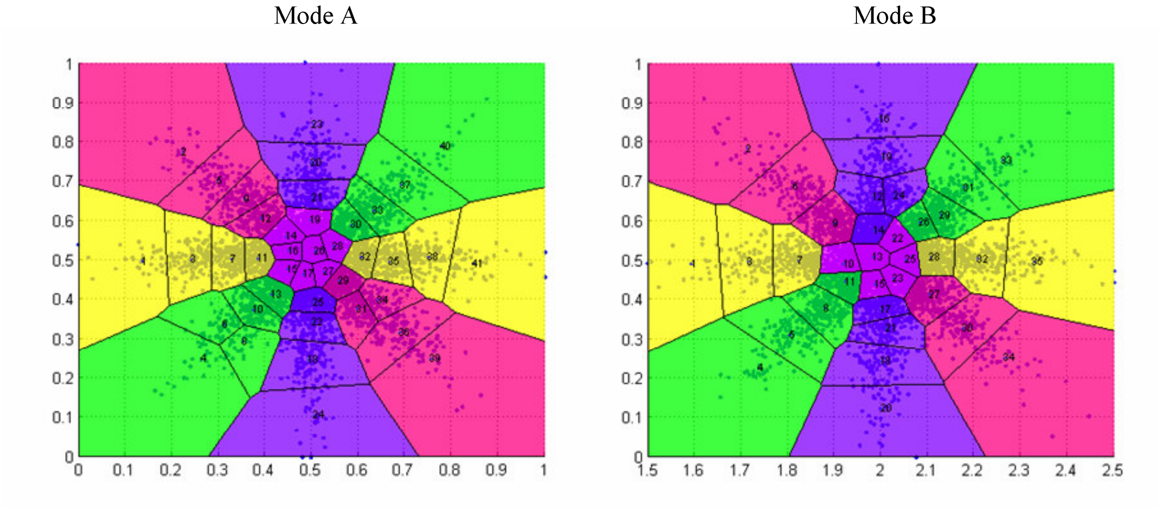

Figure 32 – The output of cross-modally clustering four overlapping Gaussian distributions in each slice. The confusion region between them is indicated in the center of the clusters.

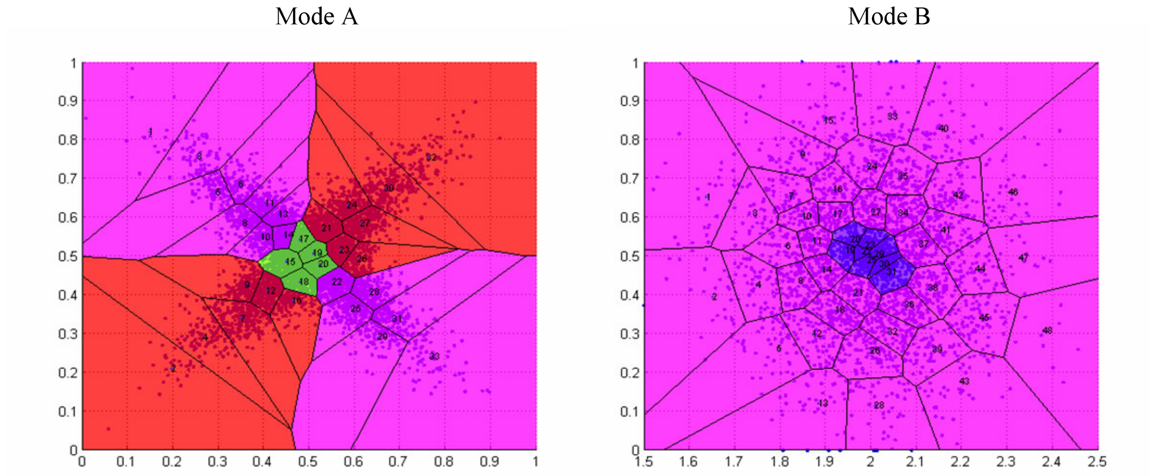

Figure 33 – Finding one cluster embedded in another. In mode B, cross-modal clustering is able both to detect the small cluster embedded in the larger one and to use this separation of clusters to detect those in mode A. This is due to the non-Euclidean scale invariance of Similarity distance, which is used for determining the cross-modal distance between regions. Thus, region size is unimportant in this framework, and "small" regions are as effective in disambiguating other modes as are "large" regions.

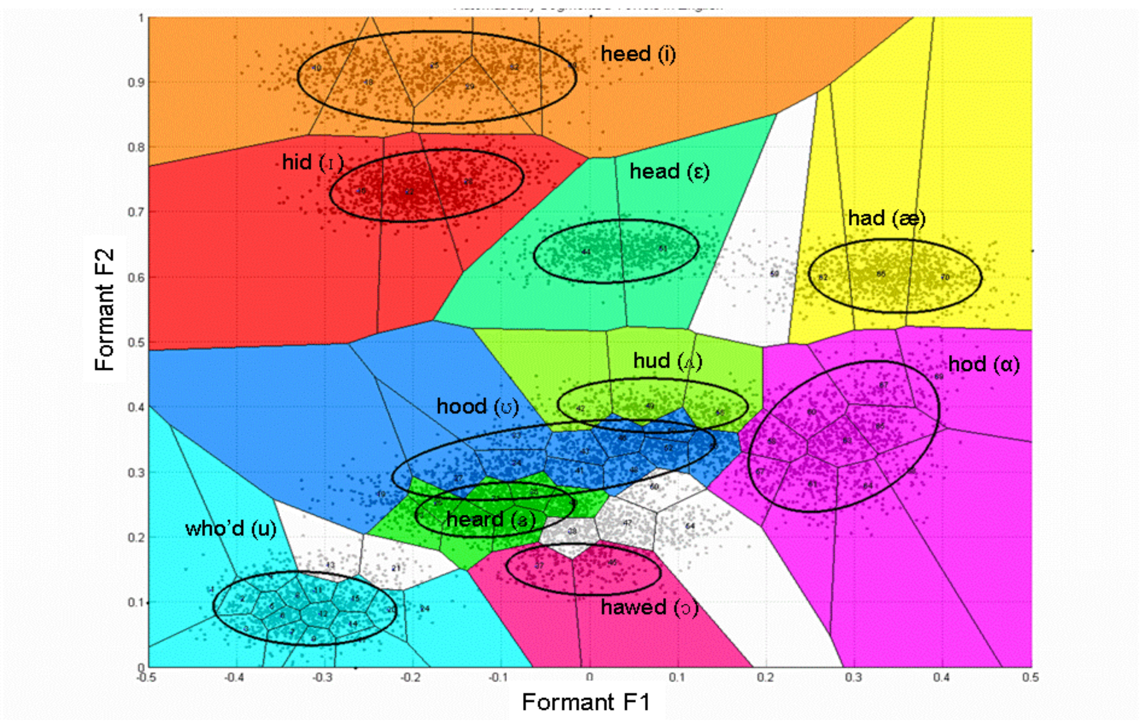

Figure 34 – Self-supervised acquisition of vowels (monophthongs) in American English. This work is the first unsupervised acquisition of human phonetic data of which we are aware. The identifying labels were manually added for reference and ellipses were fit onto the regions to aid visualization. All data have been normalized. Note the correspondence between this and the Peterson-Barney data show below.

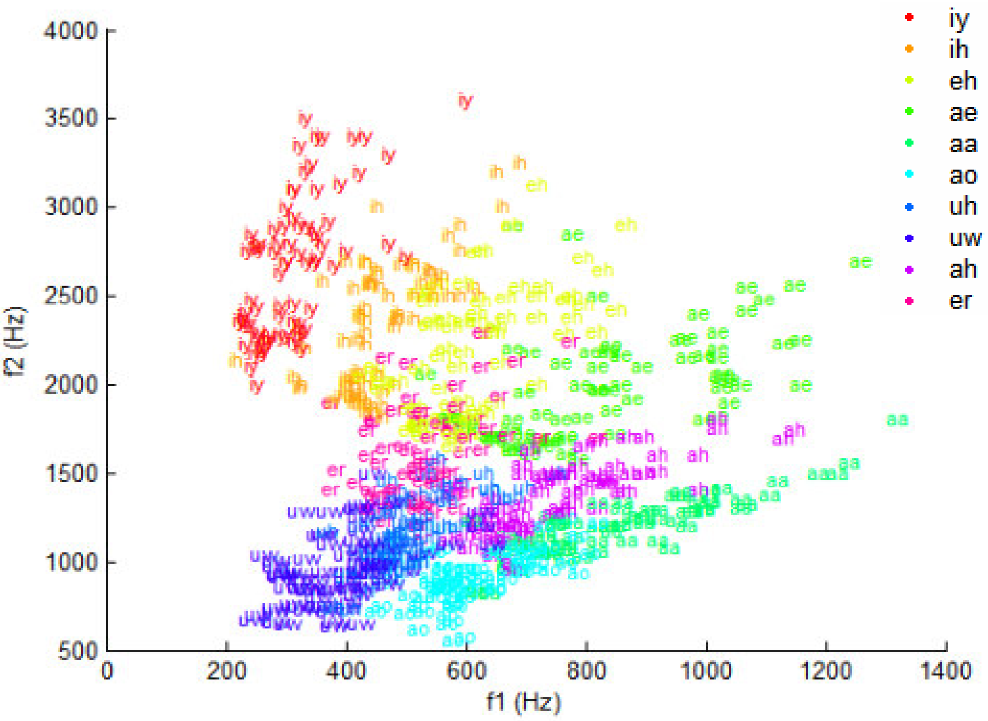

Figure 35—The Peterson-Barney dataset. Note the correspondence between this and Figure 34.

DRAFT NOTE: UPDATE THE COLORS AND IPA LABELS TO MATCH!!!

The progression of the algorithm starting with the initial codebook is shown in Figure 31. Setting $\lambda < 1$ includes Euclidean distances in the calculation of $\tilde{D}_{CM}$. This favors mergers between adjacent regions, which makes the algorithm easier to visualize. At the final step, setting $\lambda = 1$ and thereby ignoring Euclidean distance allows the remaining spatially disjoint regions to merge. Had we been uninterested in visualizing the intermediate clusterings, we would have set $\lambda = 1$ at the beginning. Doing so yields an identical result with this dataset, but the regions merge in a different, disjoint order. Note in general, however, it is not the case that different values of λ yield identical clusterings. All other examples in this thesis use $\lambda = 1$ exclusively.

Figure 32 demonstrates that the algorithm is able to resolve multiple overlapping clusters, in this case, two mixtures of four Gaussian distributions. Figure 33 show an important property of the Similarity distance, namely, it is scale invariant. The smaller cluster in Mode B is just as "distinct" as the larger one in which it is embedded. It is both detected and used to help cluster the regions in Mode A.

3.1.9 Clustering Phonetic Data

In Chapter 2, we asked the basic question of how categories are learned from unlabelled perceptual data. In this section, we provide an answer to this question using cross-modal clustering. We present a system that learns the number (and formant structure) of vowels (monophthongs) in American English, simply by watching and listening to someone speak and then cross-modally clustering the accumulated auditory and visual data. The system has no advance knowledge of these vowels and receives no information outside of its sensory channels. This work is the first unsupervised machine acquisition of phonetic structure of which we are aware.

For this experiment, data was gathered using the same pronunciation protocol employed by (Peterson and Barney 1952). Each vowel was spoken within the context of an English word beginning with [h] and ending with [d]; for example, /ae/ was pronounced in the context of "had." Each vowel was spoken by an adult female approximately 90-140 times. The speaker was videotaped and we note that during the recording session, a small number of extraneous comments were included and analyzed with the data. The auditory and video streams were then extracted and processed.

Formant analysis was done with the Praat system (Goedemans 2001, Boersma and Weenink 2005), using a 30ms FFT window and a 14th order LPC model. Lip contours were extracted using a system written by the author described in Chapter 2. Time-stamped formant and lip contour data were fed into *slices* in an implementation of the work in this thesis written by the author in Matlab and C. This implementation is able to visually animate many of the computational processes described here. This capability was used to generate most of the figures in this thesis, which represent actual system outputs.

Figure 34 shows the result of cross-modally clustering formant data with respect to lip contour data. Notice the close correspondence between the formant clusterings in Figures 22 and 23, which displays the Peterson-Barney dataset introduced earlier. We see the cross-modal clustering algorithm was able to derive the same clusters with the same spatial topology, without knowing either the number of clusters or their distributions.

The formant and lip slices are shown together in Figure 36, where the colors show region correspondences between the slices. This picture exactly captures what we mean by mutual bootstrapping. Initially, the slices "knew" nothing about the events they perceive. Cross-modal clustering lets them mutually structure their perceptual representations and thereby learn the event categories that generated their sensory inputs. The black lines in the figure connect neighboring regions within each slice and the red line connect corresponding regions in different slices. They show a graph view of the clustering within each slice and illustrate how a higher-dimensional manifold may be constructed out of lower-dimensional slices. This proposes an alternative view of the structures created by cross-modal clustering, which we hope to explore in future work.

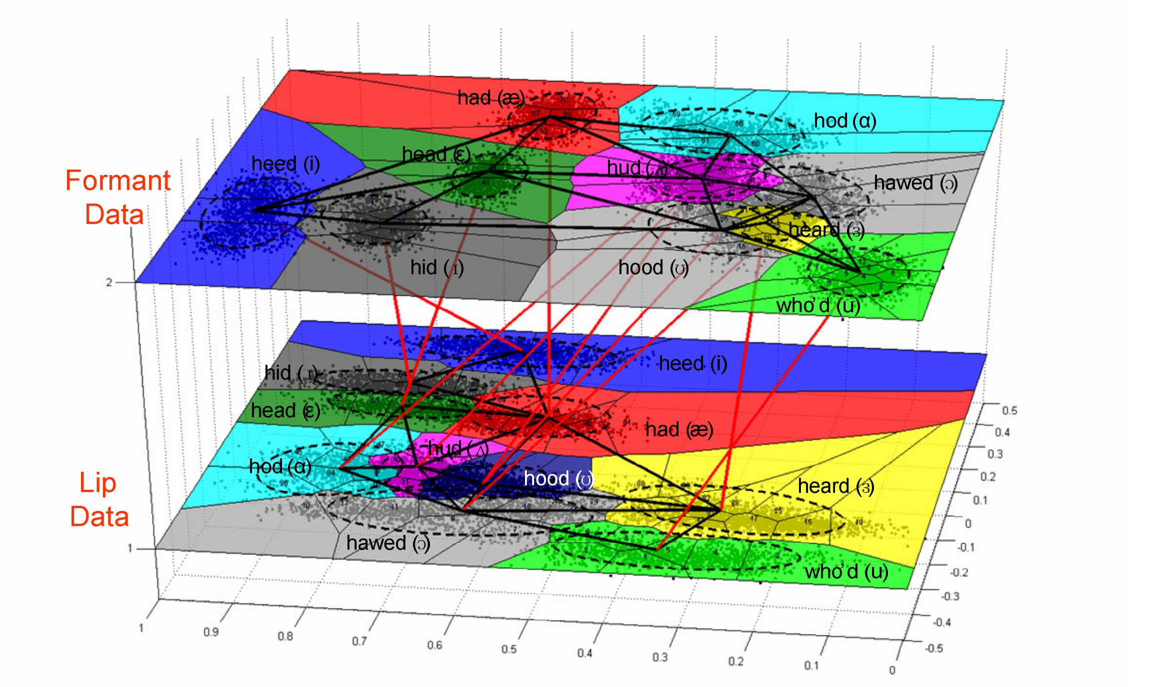

Figure 36 – Mutual bootstrapping through cross-modal clustering. This displays the formant and lip slices together, where the colors show the region correspondences that are obtained from cross-modal clustering. Initially, the slices "knew" nothing about the events they perceive. Cross-modal clustering lets them mutually structure their perceptual representations and thereby learn the event categories that generated their sensory inputs.

Summary

This chapter introduced *slices*, a neurologically inspired data structure for representing sensory information. Slices partition perceptual spaces into codebooks and then reassemble them to construct clusters corresponding to the actual sensory events being perceived. To enable this, we defined a new metric for comparing spatial probability distributions called *Similarity distance*; this allows us to measure distances within slices through cross-modal Hebbian projections onto other slices. We then presented an algorithm for *cross-modal clustering*, which uses temporal correlations between slices to determine which hyperclusters within a slice correspond to the same sensory events. The cross-modal clustering algorithm does not presume that either the number of clusters in the data or their distributions is known beforehand. We also examined the outputs and behavior of this algorithm on simulated datasets and on real data gathered in computational experiments. Finally, using cross-modal clustering, we have shown that sensory systems can be perceptually grounded by bootstrapping off each other.

DRAFT NOTE: To do for this chapter:

Replace Figures 22 – 24 with new versions

Include high-level Matlab code in an appendix, at least for cross-modal clustering and coherence determination.

References – (covers the whole thesis, not just this excerpt)

1. Agin, G.J. Representation and description of curved objects. Ph.D. Thesis, Stanford University, Stanford, CA. 1972.
2. Alcock, J. Animal Behavior: An Evolutionary Approach, 7th edition. Sinauer Associates, Inc., Sunderland, Massachusetts. 2001.
3. Amari, S. Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42:339-364. 1980.
4. Arbib, M.A., Schemas for the temporal organization of behavior. *Human Neurobiology*, 4, 63-72. 1985
5. Aristotle. De Anima. 350 BCE. Translated by Tancred, H.L. Penguin Classics. London. 1987.
6. Atkeson CG, Hale J, Pollick F, Riley M, Kotosaka S, Schaal S, Shibata T, Tevatia G, Vijayakumar S, Ude A, Kawato M: Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, Special Issue on Humanoid Robotics, 46-56. 2000.
7. Auroy P., Irthum B., Woda A. Oral nociceptive activity in the rat superior colliculus. *Brain Res*. 549:275-284. 1991.
8. Bangalore, S. and Johnston, M. Integrating multimodal language processing with speech recognition. In *ICSLP-2000*, vol.2, 126-129. 2000.
9. Bednar, J.A., Choe, Y., De Paula, J., Miikkulainen, R., Provost, J., and Tversky, T. Modeling Cortical Maps with Topographica, *Neurocomputing*. 58—60 pp. 1129-1135. 2004.
10. Bellman, R. *Adaptive Control Processes*. Princeton University Press, 1961.
11. Bender, R. E. The Consquest of Deafness (3rd Ed.). Danville, Il: Interstate Publishers. 1981.
12. Binford, T. O., Visual perception by computer. In the *Proceedings of the IEEE Systems Science and Cybernetics Conference*. Miami, FL. *IEEE*, New York. 1971.
13. Bishop, C.M., Neural Networks for Pattern Recognition. Oxford University Press, 1995.
14. Blake, R., Sobel, K. V., and Gilroy, L. A., Visual Motion Retards Alternations between conflicting Perceptual Interpretations. *Neuron*, (39):1–20, August, 2003.
15. Bloedel, JR, Ebner, TJ, and Wise, SP (eds.), Acquisition of Motor Behavior in Vertebrates, MIT Press: Cambridge, MA 1996.
16. Bobick, A.; Intille, S.; Davis, J.; Baird, F.; Pinhanez, C.; Campbell, L.; Ivanov, Y.; Schütte, A.; and Wilson, A. The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment. M.I.T. Media Laboratory Perceptual Computing Section 398. 1996.
17. Boersma, P., and Weenink, D., PRAAT, a system for doing phonetics by computer. (Version 4.3.14) [Computer program]. Glot International 5(9/10): 341-345.http://www.praat.org/. May 26, 2005.
18. Bolt, R. A., Put-That-There: Voice and gesture at the graphics interface. *Computer Graphics*. Vol. 14, No. 3. pp. 262 – 270. July, 1980.

19. Brainard, M.S. and Doupe, A.J. Auditory feedback in learning and maintenance of vocal behaviour. *Nat Rev Neurosci*, 2000.

20. Brenowitz, E.A., Margoliash, D., Nordeen, K.W. (Eds.) Special issue on birdsong. *Journal of Neurobiology*. Volume 33, Issue 5, pp 495-709. 1997.

21. Brooke N. M., & Scott S. D. PCA image coding schemes and visual speech intelligibility. *Proc. Institute of Acoustics*, **16**(5), 123–129. 1994.

22. Brooks, R., A Robust Layered Control System For A Mobile Robot. *IEEE Journal of Robotics and Automation.* 2:14-23. March, 1986.

23. Brooks, R. Intelligence without representation, *Artificial Intelligence.* 47:139–159. 1991a.

24. Brooks, R. Intelligence Without Reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (IJCAI-91). Sydney, Australia. 1991b.

25. Brooks, R.A., C. Breazeal (Ferrell), R. Irie, C. Kemp, M. Marjanovic, B. Scassellati and M. Williamson, Alternate Essences of Intelligence. *In Proceedings of The Fifteenth National Conference on Artificial Intelligence.* (AAAI98). Madison, Wisconsin. 1998.

26. Bruner, J.S. and Postman, L. On the Perception of Incongruity: A Paradigm. In *Journal of Personality*, *18*, 206-223. 1949.

27. Bushara, K.O., Hanakawa, T., Immisch, I., Toma, K., Kansaku, K., and Hallett, M., Neural correlates of cross-modal binding. *Nature Neuroscience* 6, 190-195. 1 Feb 2003.

28. Butterworth, G. The origins of auditory-visual perception and visual proprioception in human development. In *Intersensory Perception and Sensory Integration*, R.D. Walk and L.H. Pick, Jr. (Eds.) New York. Plenum. 1981.

29. Calvert, A.G., Spence, C., and Stein, B.E. The Handbook of Multisensory Processes. Bradford Books. 2004

30. Calvert, G.A., Bullmore, E., Brammer, M.J., Campbell, R., Iversen, S.D., Woodruff, P., McGuire, P., Williams, S., and David, A.S., Silent lipreading activates the auditory cortex. *Science*, 276, 593-596. 1997.

31. Calvert, GA., Campbell R., and Brammer, MJ., Evidence from functional magnetic resonance imaging of crossmodal binding in the human heteromodal. *Curr. Biol.*, 2000.

32. Camarillo D.B., Krummel T.M., Salisbury J.K., Robotic technology in surgery: past, present, and future. *Am J Surg*. 188:2S-15S. 2004.

33. Cheng, G., and Kuniyoshi**,** Y. Complex Continuous Meaningful Humanoid Interaction: A Multi Sensory-Cue Based Approach *Proc. of IEEE International Conference on Robotics and Automation* (ICRA 2000), pp.2235-2242, San Francisco, USA, April 24-28, 2000.

34. Cherry, E. C., Some experiments on the recognition of speech with one and two ears. *J. Acoust. Soc. Am*. 25, 975–979. 17, 227–246. 1953.

35. Chomsky, N. Language and the Brain. Quandaries and Prospects. Hans-Lukas Teuber Lecture. Massachusetts Institute of Technology. October 13, 2000.

36. Chomsky, N., and Halle, M. *The sound pattern of English.* New York: Harper and Row. 1968.

37. Citti, G. and Sarti, A. A cortical based model of perceptual completion in the roto-translation space. In *Proceeding of the Workshop on Second Order Subelliptic Equations and Applications.* Cortona. 2003.

38. Clark B, and Graybiel A., Factors Contributing to the Delay in the Perception of the Oculogravic Illusion. *Am J Psychol 79*:377-88. 1966.

39. Clarkson, P. and Moreno, P.J. On the use of support vector machines for phonetic classification. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99).* 1999.

40. Clemins, P.J. and Johnson, M.T. Application Of Speech Recognition To African Elephant (Loxodonta Africana) Vocalizations. In Proceedings of the *International Conference on Acoustics, Speech, and Signal Processing*. Vol. I p. 487. 2003.

41. Coen, M.H. Cross-modal clustering. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*. (*AAAI-05*) Pittsburg, Pennsylvania. 2005.

42. Coen, M.H. Design Principles for Intelligent Environments. In *Proceedings of The Fifteenth National Conference on Artificial Intelligence.* (*AAAI-98*). Madison, Wisconsin. 1998.

43. Coen, M.H. Multimodal interaction: a biological view. In *Proceedings of 17th International Joint Conference on Artificial Intelligence*. (*IJCAI-01*) Seattle, Washington. 2001.

44. Coen, M.H. The Future Of Human-Computer Interaction or How I learned to stop worrying and love My Intelligent Room. *IEEE Intelligent Systems*. March/April. 1999.

45. Coen, M.H., and Wilson, K. Learning Spatial Event Models from Multiple-Camera Perspectives in an Intelligent Room. In *Proceedings of MANSE'99.* Dublin, Ireland. 1999.

46. Coen, M.H[, Phillips, B., Warshawsky, N., Weisman, L., Peters, S., Gajos, K., and Finin, P. Meeting the computational needs of intelligent environments: The Metaglue System. In *Proceedings of MANSE'99*. Dublin, Ireland. 1999.

47. Coen, M.H., The Future of Human-Computer Interaction, or How I Learned to Stop Worrying and Love My Intelligent Room, *IEEE Intelligent Systems*, vol. 14, no. 2, Mar./Apr., pp. 8--19. 1999.

48. Cohen, P. R., Johnston, M., McGee, D., Smith, I. Oviatt, S., Pittman, J., Chen, L., and Clow, J. QuickSet: Multimodal interaction for simulation set-up and control. 1997, *Proceedings of the Applied Natural Language Conference,* Association for Computational Linguistics. 1997.

49. Collins, A. M. & Loftus, E. F., A spreading activation theory of semantic processing. *Psychological Review*, *82*, 407-428. 1975.

50. Collins, A. M. & Quillian, M. R., Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, *8*, 240-247. 1969.

51. Cotzin, M. and Dallenbach, K. (1950). Facial Vision: the role of pitch and loudness in the perception of obstacles by the blind. *The American Journal of Psychology.* 63: 485-515

52. Cotzin, M. and Dallenbach, K., Facial Vision: the role of pitch and loudness in the perception of obstacles by the blind. *The American Journal of Psychology*, 63: 485-515. 1950.

53. Cytowic, R. E., Synesthesia: A Union of Senses. New York. Springer-Verlag. 1989.

54. Dale AM, Fischl B, and Sereno MI., Cortical surface-based analysis. I. Segmentation and surface reconstruction. *Neuroimage*. Feb; 9(2):179-94. 1999.

55. Dantzig, G.B. *Application of the simplex method to a transportation problem*. In Activity Analysis of Production and Allocation, Koopmans, T.C. (Ed.) pp. 339--347. Wiley, New York, 1951.

56. Darrell, T., Gordon, G., Harville, M., and Woodfill, J., Integrated person tracking using stereo, color, and pattern detection, *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '98),* pp. 601-609, Santa Barbara, June, 1998.

57. Dautenhahn, K. and Nehaniv, C. L. (eds.), Imitation in Animals and Artifacts. MIT Press: London, 2002.

58. de Sa, V.R. *Unsupervised Classification Learning from Cross-Modal Environmental Structure*. Doctoral Dissertation, Department of Computer Science, University of Rochester. 1994.

59. de Sa, V.R., & Ballard, D. Category Learning through Multi-Modality Sensing. In *Neural Computation* 10(5). 1998.

60. Dempster, A., Laird, N. and Rubin, D., Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39(1):1–38, 1977.

61. Dennett, D.C. Consciousness Explained, Little, Brown and Company. Boston, MA. 1991.

62. Dennett, D. C., and Haugeland, J. Intentionality. The Oxford Companion to the Mind. Gregory, R.L. (Ed.)., Oxford University Press. 1987.

63. Dobbins, A. C., Jeo, R., and Allman, J., Absence of spike frequency adaptation during binocular rivalry [Abstract]. *Society for Neuroscience Abstracts, 21.* 1995.

64. Donoho, D.L., and Grimes, C., Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS*; vol. 100; no. 10;5591-5596. May 13, 2003.

65. Elbert, T., Pantev, C., Wienbruch, C., Rockstroh, B., and Taub, E. Increased Cortical Representation of the Fingers of the Left Hand in String Players, *Science*: 270: 305-307. 1995.

66. Ernst, M.O., and Banks, M.S., Humans integrate visual and haptic information in a statistically optimal fashion. *Nature.* 415, 429-433; doi: 10.1038/415429a. 24 January 2002.

67. Fahlman, S.E. NETL: A System for Representing and Using Real-World Knowledge, MIT Press, Cambridge MA. 1979.

68. Ferrell, C. Orientation behavior using registered topographic maps. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96).* Society of Adaptive Behavior. 1996.

69. Finney, E.M., Fine, I., and Dobkins, K.R., Visual stimuli activate auditory cortex in the deaf. *Nature Neuroscience* 4, 1171-1173, 2001.

70. Fischl B., Sereno, M.I. and Dale, A.M., Cortical Surface-Based Analysis. II: Inflation, Flattening, and a Surface-Based Coordinate System. *Neuroimage*. 9(2):195-207. 1999.

71. Fisher, J. and Darrell, T., Signal-Level Audio Video Fusion Using Information Theory. *Proceedings of Workshop on Perceptive User Interfaces*. 2001.

72. Fitzgibbon, A., Pilu, M., and Fisher, R.B., "Direct Least Square Fitting of Ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476-480, May, 1999.

73. Fitzpatrick, R. and McCloskey, D.I., Proprioceptive, visual and vestibular thresholds for the perception of sway during standing in humans. *The Journal of Physiology*, Vol 478, Issue 1 pp. 173-186, 1994.

74. Flynn, A., Redundant sensors for mobile robot navigation, M.S. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA, July 1985.

75. Frank, A. On Kuhn's Hungarian Method - a tribute from Hungary. *Naval Research & Logistics*. 52. 2005.

76. Frisby, J. P., and Davies, I. R. L., Is the Haptic Müller–Lyer a Visual Phenomenon? *Nature*. 231, 463-465. 18 Jun 1971.

77. Galef, B. G., Imitation in animals: History, definition, and interpretation of data from the psychological laboratory. In: Social learning: Psychological and biological perspectives, eds. T. R. Zentall & B. G. Galef, Jr. Erlbaum. 1988.

78. Garnicia, O. K., Some prosodic and paralinguistic features of speech to young children. In C. E. Snow and C. A. Ferguson (Eds.) Talking to children. Cambridge University Press. 1977.

79. Gazzaniga, M. (Ed.) The New Cognitive Neurosciences. MIT Press. 2nd edition. Cambridge, MA. 2000.

80. Gerstner, W and Kistler, W.M., Spiking Neuron Models. Single Neurons, Populations, Plasticity Cambridge University Press. 2002.

81. Gerstner, W. and Kistler W., Spiking Neuron Models. Single Neurons, Populations, Plasticity.
Cambridge University Press, 2002.

82. Gibbon, J. Scalar expectancy theory and Weber's law in animal timing. *Psychol. Review* 84(3):279-325. 1977.

83. Gibbs, A.L and Su, F.E. On choosing and bounding probability metrics. *International Statistical Review*, vol. 70, number 3, 419-435. 2002

84. Gibson. J.J. The Ecological Approach to Visual Perception. Lawrence Earlbaum Associates. Hillsdale, N.J. 1986.

85. Gold, B. and Morgan, N. *Speech and Audio Signal Processing*. Wiley Press, New York. 2000.

86. Gross, R., Yang, J., and Waibel, A. Face Recognition in a meeting room. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, March 2000

87. Grunfeld, E.A., Okada, T, Jauregui-Renaud, K., and Bronstein, A.M., The effect of habituation and plane of rotation on vestibular perceptual responses. *J Vestib Res*. 10(4-5):193-200. 2000.

88. Haesler, S., Wada, K., Nshdejan, A., Morrisey, E.E., Lints, T., Jarvis, E.D., and Scharff, C. FoxP2 Expression in Avian Vocal Learners and Non-Learners. *J. Neurosci*. 24: 3164-3175. 2004

89. Hall, A.C. and Moschovakis, A. eds., The superior colliculus: new approaches for studying sensorimotor integration. (Boca Raton: CRC Press) 2004.

90. Hamill, N.J., McGinn, M.D. and Horowitz, J.M., Characteristics of auditory brainstem responses in ground squirrels. *Journal of Comparative Physiology* B: Volume 159, Number 2. Pages: 159 – 165. March 1989.

91. Hebb, D.O. 1949. *The Organization of Behaviour*. John Wiley & Sons, New York.

92. Heil, P. and Neubauer ,H., A unifying basis of auditory thresholds based on temporal summation. *PNAS*, Vol. 100, no. 10. May 13, 2003.

93. Held, R. Shifts in binaural localization after prolonged exposures to atypical combinations of stimuli. *Am. J. Psychol.* 68L526-266. 1955.

94. Helmholtz, H. v. *Handbook of Physiological Optics.* 1856. as reprinted. in James P.C. Southall. (Ed.) 2000.

95. Hennecke M. E., Prasad K. V., & Stork D. G. Using deformable templates to infer visual speech dynamics. In *Proc. 28th Annual Asilomar Conf. on Signals, Systems and Computers*. 1994.

96. Hinton, G.E. and Sejnowski, T.J. Learning and relearning in Boltzman machines. In *Parallel Distributed Processing*, Rumelhart, D.E. and McClelland, J.L. (eds), Volume 1:Foundations. MIT Press, Cambridge, MA. 1986.

97. Holbrook, A., and Fairbanks, G. Diphthong Formants and their Movements. *J. Speech Hear. Res*. 5, 38-58. 1962

98. Horn, B.K.P. Shape from Shading. MIT Artificial Intelligence Laboratory Technical Report. AITR 232. November 1970.

99. Hoshino, O. and Kuroiwa, K., Echo sound detection in the inferior colliculus for human echolocation. *Neurocomputing*: *An International Journal Special Issue*, 38-40, 1289-1296. 2001.

100. Howard, I. P. & Templeton, W. B. *Human Spatial Orientation*, Wiley, New York. 1966.

101. Hsu, W.H., and Ray, S.R., Construction of recurrent mixture models for time series classification. *International Joint Conference on Neural Networks* (IJCNN'99), 1999.

102. Huang, X, Acero, A., and Hon, H.W. Spoken Language Processing. Prentice Hall, New Jersey. 2001.

103. Hubel, D. H. and Wiesel, T. N., *J. Physiol.,* London. 160, 106−154. 1962.

104. Hughes, J. W. The threshold of audition for short periods of stimulation, *Proc. R. Soc. London Ser. B* 133, 486-490. 1946.

105. Intille, S. S., Larson, K., and Tapia, E. M., Designing and evaluating technology for independent aging in the home, in *Proceedings of the International Conference on Aging, Disability and Independence*. 2003.

106. J. Kleinberg. An Impossibility Theorem for Clustering. *Advances in Neural Information Processing Systems (NIPS)* Whistler, British Columbia, Canada. 2002.

107. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., and Hinton, G.E. (1991) Adaptive mixtures of local experts. *Neural Computation,* 3, 79-87

108. Jacoby, L.L. Remembering the data: analyzing interactive processes in reading. *Journal of Verbal learning and Verbal Behaviour, 22,* 485-508. 1983.

109. James, W. 1890. Principles of Psychology. Vol. 2. Dover. 1955.

110. Jelinek, F. Statistical Methods for Speech Recognition. MIT Press. Cambridge, MA. 1997.

111. Kaas J.H., and Hackett T.A. Subdivisions of auditory cortex and processing streams in primates. In *PNAS.* Oct 24;97(22):11793-9. 2000.

112. Kaas, J. The Reorganization of Sensory and Motor Maps after Injury in Adult Mammals, in M. Gazzaniga (ed.), The New Cognitive Neurosciences, 2nd ed. (Cambridge: MIT Press: 223-236). 2000.

113. Kardar, M. and Zee, A. Information optimization in coupled audio-visual cortical maps. In *PNAS* 99: 15894-15897. 2002

114. Kautau, A. Classification of the Peterson & Barney vowels using Weka. Technical Report. UCSD. 2002.

115. Keele S.W. and Summers, J. J., The Structure of Motor Programs. In Motor Control – Issues and Trends. Stelmach, G. E., (Ed.) Academic Press. San Diego. 1976.

116. Klee, V., and Minty, G. J., *How good is the simplex algorithm*?, in Inequalitites, III. Shisha, O. (Ed.), Academic Press, New York, pp. 159-175. 1972.

117. Kogan, J.A. and Margoliash, D. Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study. *Journal of the Acoustic Society of America*. Vol. 103, No. 4, April 1998.

118. Kohler, I. The formation and transformation of the perceptual world. *Psychological Issues* 3(4):1-173. 1964.

119. Kohonen, T. Self-Organization and Associative Memory. Springer-Verlag, Berlin. 1984.

120. Kuhl, P.K. Human adults and human infants show a 'perceptual magnet effect' for the prototypes of speech categories, monkeys do not. *Perception & Psychophysics*, 50, pp. 93-107. 1991.

121. Kuhn, H. The Hungarian method for the assignment problem. *Naval Res. Logist. Q*. 2:83—97. 1955.

122. Kumar, V. 2002. Towards Trainable Man-machine Interfaces: Combining Top-down Constraints with Bottom-up Learning in Facial Analysis. Ph.D. Thesis. Department of Brain and Cognitive Sciences. MIT.

123. Leopold, D. A., Wilke, M., Maier, A. and Logothetis, N.K. Stable perception of visually ambiguous patterns. *Nature*. Volume 5, Number 6, pp 605 – 609. June 2002.

124. Levina, E., and Bickel P.J., The Earth Mover's Distance is the Mallows Distance: Some Insights from Statistics. *Proceedings of ICCV*, Vancouver, Canada, pp. 251-256. 2001.

125. Lewkowicz, D.J. and Lickliter, R. (eds.) The Development of Intersensory Perception. Lawrence Erlbaum Associations. Hillsdale, N.J. 1994.

126. Lippmann, R.P. Speech recognition by machines and humans. *Speech Communication* 22, 1-15. 1987.

127. Lloyd, S. P., `Least squares quantization in PCM,' *IEEE Trans. Inform. Theory*, vol. 28, pp. 129-137, 1982.

128. López, M.E., Barea, R., Bergasa, L.M. and Escudero, M.S., Visually Augmented POMDP for Indoor Robot Navigation (Spain) *From Proceedings of* Applied Informatics. 2003.

129. MacKay, D.H.J. Information Theory, Inference, and Learning Algorithms. Cambridge University Press. 2003.

130. Mackey, M.C., and Glass, L. Oscillation and Chaos in Physiological Control Systems. *Science*,

131. Mackie, G.O. and Singla, C.L. Studies on Hexactinellid Sponges. I. Histology of Rhabdocalyptus dawsoni (Lambe, 1873). *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, Vol. 301, No. 1107. (Jul. 5, 1983), pp. 365-400. 1983.

132. Marler, P. Three models of song learning: Evidence from behavior. *Journal of Neurobiology.* Volume 33, Issue 5 , Pages 501 – 516. 1997.

133. Marr, D. Vision. WH Freeman, San Francisco, 1982.

134. Mase, K., and Pentland, A. Automatic Lipreading by Computer. *Trans. IEEE.*, vol. J73-D-II, No. 6, pp. 796-803, June 1990.

135. Massaro, D.W. The fuzzy logical model of speech perception: A framework for research and theory. In Y. Tohkura, E. Vatikiotis-Bateson, and Y. Sagisaka (Eds.), *Speech Perception, Production and Linguistic Structure* (pp.79-82). Ohmsha Ltd, Tokyo. 1992.

136. Massaro, D.W. and Cohen, M.M. Fuzzy logical model bimodal emotion perception: Comment on The perception of emotions by ear and by eye by de Gelder and Vroomen *Cognition and Emotion, 14(3),* 313-320. 2000.

137. Massaro, D.W., Cohen, M.M., Gesi, A. and Heredia, R. Bimodal Speech Perception: An Examination across Languages. *Journal of Phonetics, 21*, 445-478. 1993.

138. Massie, T. M. and Salisbury, J. K.. *The PHANToM Haptic Interface: A Device for Probing Virtual Objects*. In ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems 1994, Dynamic Systems and Control 1994, volume 1, pages 295--301, Nov. 1994.

139. Massone, L., and Bizzi, E., On the Role of Input Representations in Sensorimotor Mapping, *Proc. IJCNN*, Washington DC, 1:173-176. 1990.

140. Mataric, M.J., Studying the Role of Embodiment in Cognition. *Cybernetics and Systems* 28(6):457-470. 1997.

141. Maunsell, J.H.R, Nealy, T.A., Sclar, G., and DePriest, D.D. Representation of extraretinal information in mokey visual cortex. In *Neural mechanisms of visual perception. Proceedings of the Second Retina Research Foundation Symposium*, 14-15 April 1989. D.M Lam and C.D. Gilbert (eds). Woodlands, Texas. Portfolio Pub. Co. 1989.

142. McGraw, M. B., The Neuromuscular Maturation of the Human Infant. New York: Institute of Child Development. 1939.

143. McGurk, H., and MacDonald, J. Hearing lips and seeing voices. *Nature*. 264:746-748. 1976.

144. Mellinger, D.K. and Clark. C.W. Bioacoustic transient detection by image convolution. *Journal of the Acoustical Society of America*. -- Volume 93, Issue 4, p. 2358. April 1993.

145. Meltzoff, A.N. and Moore, M.K. Imitation of facial and manual gestures by human neonates. *Science* 198:75-78. 1977.

146. Meltzoff, A. N., and Prinz, W., The imitative mind: Development, evolution, and brain bases. Cambridge, England: Cambridge University Press. 2002.

147. Metcalfe, J.S., . Chang, T.Y., Chen, L.C., McDowell, K., Jeka, J.J., and Clark, J. E. Development of somatosensory-motor integration: An event-related analysis of infant posture in the first year of independent walking. *Developmental Psychobiology* 46(1) 19-35. 2005.

148. Meyer, D. E., and Schvaneveldt, R. (1971). Facilitation in recognizing pairs of words: evidence of a dependence between retrieval operations. *Journal of Experimental Psychology* 90:227--235.

149. Miller, G. and Chomsky, N. Finitary models of language users. In Luce, R.; Bush, R. and Galanter, E. (eds.) *Handbook of Mathematical Psychology, Vol 2.* New York: Wiley. 419-93. 1963.

150. Miller, D. B. The acoustic basis of mate recognition by female zebra finches (Taeniopygia guttata). *Anim. Behav*. 27, 376–380. 1979.

151. Minnen, D., Starner, T., Ward, J.A. ,Lukowicz, P., and Troester, G., Recognizing and Discovering Human Actions from On-Body Sensor Data ICME 2005, Amsterdam, NL, July 6-8, 2005.

152. Minsky, M., A Framework for Representing Knowledge. Reprinted in *The Psychology of Computer Visio*n, Winston, P.H. (Ed.), McGraw Hill, 1975.

153. Mitchell, T., Machine Learning. McGraw Hill, 1997.

154. Moody, J. and Darken, C. J. Fast learning in networks of locally tuned processing units. *Neural Computation*, 1:281-294. 1989.

155. Moran, T.P., and Dourish, R., Introduction to This Special Issue* on Context-Aware Computing. *Human-Computer Interaction*, Volume 16, 2001.

156. Müller, C. M. and Leppelsack, H. J., Feature extraction and tonotopic organization in the avian auditory forebrain. *Experimental Brain Research* (Historical Archive), Volume 59, Issue 3, pp. 587 – 599, Aug, 1985.

157. Mumford, S. Laws in Nature. London, England: Routledge. 2004.

158. Naeaetaenen, R., Tervaniemi, M., and E Sussman, P., Primitive intelligence in the auditory cortex. *Trends Neurosci*. 2001.

159. Nakayama, K. and Silverman, G. H., Serial and parallel processing of visual feature conjunctions. *Nature 320*, 264–265. 1986.

160. Nefian, A., Liang, L., Pi, X., Xiaoxiang, L., Mao, C. and Murphy, K. *ICASSP '02 (IEEE Int'l Conf on Acoustics, Speech and Signal Proc.)*, 2:2013--2016. 2002

161. Newell A. and Rosenbloom P.S., Mechanisms of skill acquisition and the law of practice. In: Cognitive skills and their acquisition (Anderson JR, Ed.), pp. 1–51. Hillsdale, NJ: Erlbaum. 1981.

162. Nilson, N. (Ed.), Shakey the Robot. SRI A.I. Center Technical Note 323. April, 1984.

163. Nordeen KW, Nordeen EJ. Auditory feedback is necessary for the maintenance of stereotyped song in adult zebra finches. *Behav Neural Biol*. Jan;57(1):58-66. 1992.

164. Ohnishi T, Matsuda H, Asada T, Aruga M, Hirakata M, Nishikawa M, Katoh A, Imabayashi E. Functional anatomy of musical perception in musicians. *Cereb Cortex.* 11(8):754-60. 2001.

165. Ölveczky B.P., Andalman A.S., Fee M.S. Vocal Experimentation in the Juvenile Songbird Requires a Basal Ganglia Circuit. PLoS Biol 3(5): e153. 2005

166. Oviatt, S. Multimodal interfaces, The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, Lawrence Erlbaum Associates, Inc., Mahwah, NJ. 2002

167. Pearson, P. and Kingdom, F.A.A., On the interference of task-irrelevant hue variation on texture segmentation. *Perception*, 30, 559-569. 2001.

168. Peterson, G.E. and Barney, H.L. Control methods used in a study of the vowels. *J.Acoust.Soc.Am.* 24, 175-184. 1952.

169. Piaget, J. Construction of reality in the child. London: Routledge & Kegan Paul, 1954.

170. Piaget, J. Genetic Epistemology. W. W. Norton & Co. Scranton, Pennsylvania. 1971.

171. Picard, R. Affective Computing. MIT Press. Cambridge, MA. 1997

172. Pierce, W.D., and Cheney, C.D. Behavior Analysis and Learning. 3rd edition. Lawrence Erlbaum Associates. NJ. 2003.

173. Poppel, E. A hierarchical model of temporal perception. *Trends in Cognitive Sciences,* Volume 1, Number 2, pp. 56-61(6). May 1997.

174. Poppel, E., Held, R., and Frost. D. Residual visual function after brain wounds involving the central visual pathways in man. *Nature*, 243, 295-296. 1973.

175. Potamianos, G., Neti, C., Luettin, J., and Matthews, I. Audio-Visual Automatic Speech Recognition: An Overview. In: *Issues in Visual and Audio-Visual Speech Processing*, G. Bailly, E. Vatikiotis-Bateson, and P. Perrier (Eds.), MIT Press (In Press), 2004.

176. Ratnanather, J. T., Barta, P. E., Honeycutt, N. A., Lee, N. G., Morris, H. M., Dziorny, A. C., Hurdal, M. K., Pearlson, G. D., and Miller, M. I. Dynamic programming generation of boundaries of local coordinatized submanifolds in the neocortex: application to the Planum Temporale, *NeuroImage*, vol. 20, pp. 359-377. 2003.

177. Remagnino, P., Foresti, G., and Ellis, T.J. (Eds.), Ambient Intelligence a Novel Approach. Springer-Verlag New York Inc. 2004.

178. Richards, W. (Ed.) Natural Computation. Cambridge, MA. The MIT Press. 1988.

179. Rubin, P., Baer, T. and Mermelstein, P., An articulatory synthesizer for perceptual research. *Journal of the Acoustical Society of America*, 70, 321-328. 1981.

180. Rubner, Y., Tomasi, C., and Guibas, L. J., A Metric for Distributions with Applications to Image Databases. *Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India*, ,pp. 59-66. January 1998.

181. Ryle, G. The Concept of Mind. Chicago: The University of Chicago Press. Chicago, IL. 1949.

182. Saar, S. Sound analysis in Matlab. http://ofer.sci.ccny.cuny.edu/html/sam.html. 2005.

183. Sams, M., Aulanko, R., Hamalainen, M., Hari, R., Lounasmaa, O., Lu, S., and Simola, J. Seeing speech: Visual information from lip movements modified activity in the human auditory cortex. *Neurosci. Lett.* 127:141-145. 1991.

184. Sandini G., Metta G. and Konczak J. Human Sensori-motor Development and Artificial System. In *Proceedings of AIR&IHAS '97*, Japan. 1997.

185. Schmidt, R. A., Motor control and learning, 2nd edition. Human Kinetics Publishers. 1988.

186. Shimojo, S., and Shams, L. Sensory modalities are not separate modalities: plasticity and interactions. Current Opinion in Neurobiology. 11:505-509. 2001.

187. Slaney, M. A Critique of Pure Audition. In *Proceedings of Computational Auditory Scene Analysis Workshop*. International Joint Conference on Artificial Intelligence, Montreal, Canada. 1995.

188. Slater, P. J. B., Eales, L. A., & Clayton, N. S. Song learning in zebra finches: Progress and prospects. *Advances in the Study of Behavior*, 18, 1–34. 1988.

189. Smolensky, P. . Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations. Cambridge, MA: MIT Press/Bradford Books. 194-281. 1986.

190. Starner, T., Wearable Computing and Contextual Awareness. Ph.D. Thesis Massachusetts Institute of Technology. June 1999.

191. Stein, B.E. and Dixon, J.E. Superior colliculus neurons respond to noxious stimuli. *Brain Research*. 158:65-73. 1978.

192. Stein, B.E., and Meredith, M. A. 1994. The Merging of the Senses. Cambridge, MA. MIT Press.

193. Stein, R.B., The frequency of nerve action potentials generated by applied currents. *Proc. R. Soc. Lond B. Biol. Sci.*, 1967.

194. Stevens, S.S., On the psychophysical law. *Psychol. Review* 64(3):153-181, 1957.

195. Still, S., and Bialek, W. How many clusters? An information theoretic perspective, *Neural Computation*. 16:2483-2506. 2004.

196. Stork, D.G., and Hennecke, M. Speechreading: An overview of image processing, feature extraction, sensory integration and pattern recognition techniques", *Proc. of the Second Int. Conf. on Auto. Face and Gesture Recog.* Killington, VT pp. xvi--xxvi. 1996.

197. Sumby, W.H., and Pollack, I. Visual contribution to speech intelligibility in noise. *J. Acoust. Soc. Am.* 26:212-215. 1954.

198. Summerfield, Q. Some preliminaries to a comprehensive account of audio-visual speech perception, in Dodd, B. and Campbell, R., editors, Hearing by Eye: The psychology of lip-reading. Lawrence Erlbaum Associates, Hillsdale NJ. pgs 3-52. 1987.

199. Sun. http://www.javasoft.com/products/java-media/speech/. 2001.

200. Sung, Michael and Pentland, Alex (Sandy)., Minimally-Invasive Physiological Sensing for Human-Aware Interfaces. *HCI International.* 2005.

201. Sussman E., Winkler I., Ritter W., Alho K., and Naatanen, R., Temporal integration of auditory stimulus deviance as reflected by the mismatch negativity. *Neuroscience Letters*, Volume 264, Number 1, 2, pp. 161-164(4). April 1999.

202. Sussman, G. Abelson, H. and Sussman, J. Structure and Interpretation of Computer Programs. MIT Press. Cambridge, MA. 1983.

203. Takeuchi, A. and Amari S-I., Formation of Topographic Maps and Columnar Microstructures in Nerve Fields. 1999.

204. Tchernichovski O, Nottebohm F, Ho C, Pesaran B, Mitra P. A procedure for an automated measurement of song similarity. *Anim Behav* 59: 1167-1176. 2000.

205. Tchernichovski, O. Private communication. 2005.

206. Tenenbaum, J. B., de Silva, V. and Langford, J. C., A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science.* 290 (5500): 2319-2323, 22 December 2000.

207. Thelen, E., and Smith, L. A Dynamic Systems Approach to the Development of Cognition and Action. Cambridge, MIT Press. 1998.

208. Thompson, DW. On Growth and Form. New York: Dover Publications. 1917 revised 1942.

209. Thorndike, E. L., Animal intelligence: An experimental study of the associative process in animals. *Psychological Review Monograph* 2(8):551–53. 1898.

210. Thrun, S., Burgard, W., and Fox, D., Probabilistic Robotics. MIT Press. Cambridge, MA. 2005.

211. Tinbergen, N., The Study of Instinct, Oxford University Press, New York, NY. 1951.

212. Treisman, A., Preattentive processing in vision. *Computer Vision, Graphics and Image Processing 31*, 156–177. 1985.

213. Trussell LO. PDF Physiological mechanisms for coding timing in auditory neurons. *Ann. Rev. Physiol.* 61:477-496. 1999.

214. Ullman, Shimon. High-level vision: object recognition and visual cognition. Cambridge. MIT Press. 1996.

215. van der Meer A.L., van der Weel F.R. and Lee D.N., The functional significance of arm movements in neonates. *Science*. 267(5198):693-5. 1995.

216. Vignal, C., Mathevon, N. & Mottin, S. Audience drives male songbird response to partner's voice. *Nature* 430, 448–451. 2004.

217. vol. 197, pp. 287, 1977.

218. Von Neumann, J. The Computer and the Brain, *Silliman Lectures Series*, Yale Univ. Press, New Haven, CT. 1958.

219. Waibel, A., Vo, M.T., Duchnowski, P., and Manke, S. Multimodal Interfaces. *Artificial Intelligence Review*. 10:3-4. p299-319. 1996.

220. Wang, F., Ma, Y.F., Zhang, H.J., Li, J.T., A Generic Framework for Semantic Sports Video Analysis Using Dynamic Bayesian Networks, pp. 115-122, 11th International Multimedia Modelling Conference (MMM'05), 2005.

221. Wang, L., Walker, V.E., Sardi, H., Fraser, C. and Jacob, T.J.C., The correlation between physiological and psychological responses to odour stimulation in human subjects. *Clinical Neurophysiology* 113, 542-551. 2002.

222. Warren, D.H., Welch, R.B. and McCarthy, T.J. The role of auditory-visual `*compellingness'* in the ventriloquism effect: Implications for transitivity amongst the spatial senses. *Perception and Psychophysics,* 30(6), pp 557- 564. 1981

223. Watanabe, S. *Pattern Recognition*. Human and Mechanical. John Wiley, New York. 1985

224. Watson, A.B., Temporal sensitivity in *Handbook of perception and human performance*. K. Boff, L. Kaufman and J. Thomas, (Eds.) Volume 1 (A87-33501 14-53). New York, Wiley-Interscience, p. 6-1 to 6-43. 1986.

225. Watts, D., and Strogatz, S. *Collective dynamics of 'small-world' networks*. Nature 393:440-442. 1998.

226. Webb, D. M., and Zhang, J. FoxP2 in Song-Learning Birds and Vocal-Learning Mammals. *Journal of Heredity*. 96: 212-216. 2005.

227. Weiner, N. Cybernetics: On Control and Communication in the Animal and the Machine. John Wiley, 1948.

228. Weiser, Mark., The Computer for the Twenty-First Century. *Scientific American.* 265(3):94—104, September 1991.

229. Weiser, Mark., The world is not a desktop. *Interactions*. pp. 7—8, January 1994.

230. Welch, R. B., and Warren, D. H. 1986. "Intersensory Interactions." In *Handbook of Perception and Human Performance*, edited by K. R. Boff, L. Kaufman, and J. P. Thomas, chap. 25. New York: Wiley.

231. Wertheimer, M. Laws of Organization in Perceptual Forms. First published as Untersuchungen zur Lehre von der Gestalt II, in *Psycologische Forschung*, *4*, 301-350. Translation published in Ellis, W. *A source book of Gestalt psychology* (pp. 71-88). London: Routledge & Kegan Paul. 1938

232. Wiener, N., Cybernetics: or the Control and Communication in the Animal and the Machine, Cambridge: MIT Press. 1948.

233. Wiggs, C.L., and Martin, A., Properties and mechanisms of perceptual priming. Current Opinion in Neurobiology *Cognitive Neuroscience*, (8):227-233, 1998.

234. Williams, H., and Staples, K. Syllable chunking in zebra finch (Taeniopygia guttata) song. *J Comp Psychol*. 106(3):278-86. 1992.

235. Winston, P.H. Learning Structural Descriptions from Examples. MIT Artificial Intelligence Laboratory Technical Report. AITR-231. September 1970.

236. Witten, I, and Frank, E. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann. 2005.

237. Wolfe, J.M. Hidden visual processes. *Scientific American, 248(2),* 94-103. 1983.

238. Wolfe, J.M., & Cave, K.R. The psychophysical evidence for a binding problem in human vision. *Neuron, 24*, 11-17. 2000.

239. Wren, C., Azarbayejani, A., Darrell, T., and Pentland, P., "Pfinder: Real-Time Tracking of the Human Body ", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1997.

240. Wu, Lizhong, Oviatt, Sharon L., Cohen, Philip R., Multimodal Integration -- A Statistical View, *IEEE Transactions on Multimedia*, Vol. 1, No. 4, December 1999, pp. 334-341.

241. Zann R. The Zebra Finch: A Synthesis of Field and Laboratory Studies. Oxford: Oxford University Press, 1996

242. Ziegler, P. & Marler, P. (Eds.) Special issue: Neurobiology of Birdsong, *Annals of the New York Academy of Science.* 1016: 348–363. 2004.

243. Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., and Hetherington, L. JUPITER: A Telephone-Based Conversational Interface for Weather Information. *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 1, January 2000.