The Discovery of Perceptual Structure from Visual Co-occurrences in Space and Time

by

Phillip Isola

B.S., Computer Science, Yale University, 2008

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Cognitive Science
at the Massachusetts Institute of Technology

September 2015

Signature of Author: Signature redacted
Department of Brain and Cognitive Sciences
June 30, 2015

Certified by: Signature redacted
Edward H. Adelson
John and Dorothy Wilson Professor of Vision Science
Thesis Supervisor

Accepted by: Signature redacted
Matthew A. Wilson
Sherman Fairchild Professor of Neuroscience and Picower Scholar
Director of Graduate Education for Brain and Cognitive Sciences

The Discovery of Perceptual Structure from Visual Co-occurrences in Space and Time

by Phillip Isola

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

Although impressionists assure us that the world is just dabs of light, we cannot help but see surfaces and contours, objects and events. How can a visual system learn to organize pixels into these higher-level structures? In this thesis I argue that perceptual organization reflects statistical regularities in the environment. When visual primitives occur together much more often than one would expect by chance, we may learn to associate those primitives and to form a perceptual group.

The first half of the thesis deals with the identification of such groups at the pixel level. I show that low-level image statistics are surprisingly effective at higher-level segmentation. I present an algorithm that groups pixels by identifying meaningful co-occurrences in an image's color statistics. Consider a zebra. Black-next-to-white occurs suspiciously often, hinting that these colors have a common cause. I model these co-occurrences using pointwise mutual information (PMI). If the PMI between two colors is high, then the colors probably belong to the same object. Grouping pixels with high PMI reveals object segments. Separating pixels with low PMI marks perceived boundaries.

If simple color co-occurrences can tell us about object segments, what might more complex statistics tell us? The second half of the thesis investigates high dimensional visual data, such as image patches and video frames. In high dimensions, it is intractable to directly model co-occurrences. Instead, I show that modeling PMI can be posed as a simpler binary classification problem in which the goal is to predict if two primitives occur in the same spatial or temporal context. This allows us to model PMI associations between complex inputs. I demonstrate the effectiveness of this approach on three domains: discovering objects by associating image patches, discovering movie scenes by associating frames, and discovering place categories by associating geotagged photos.

Together, these results shed light on how a visual system can learn to organize raw sensory input into meaningful percepts.

Thesis Supervisor: Edward H. Adelson
Title: John and Dorothy Wilson Professor of Vision Science
Thesis Committee: William T. Freeman, Joshua B. Tenenbaum

Acknowledgments

For me, the most enjoyable part of grad school has been getting to interact with so many creative and talented people. I've been fortunate to work with dozens of collaborators, friends, and mentors, and I'm grateful to them all. Here I will specify just a few who I worked especially closely with.

Thanks to Ted, my advisor, for his constant support and endless wisdom. I have very much appreciated the freedom he gave me to explore an eclectic array of projects. His disciplined approach and honest outlook on the limits of our knowledge has given me perspective on what it means to be a good scientist.

Thanks to Aude, for her mentorship throughout grad school, especially when I was just getting started. Her excitement is infectious, and her grand vision kept me inspired whenever day to day progress seemed slow.

Thanks to the other members of my committee, Bill and Josh. Thanks to Bill for showing me how an artistic creativity can be combined with rigorous engineering. Thanks to Josh for always providing deep and enjoyable conversation, and forcing me to think about the theoretical implication of my work.

Thanks to Antonio, who, along with Aude, fueled so much excitement and progress in my first few years working on memorability. Antonio's humor and creativity made working together always fun.

Thanks to Ce, who mentored me as an intern at Microsoft Research, for driving me not just to make things work, but also to understand why they work.

Thank you Daniel and Dilip, who were my co-authors on all the work described in this thesis. Daniel, thanks for always seeing the best side of each idea, for your excitement and ability to develop a half baked thought into a thorough experiment.

Dilip, thanks for your cool, collected, and kind approach to all things, and your skill at cutting away the fluff and getting at the main point.

Thank you to my all my collaborators on memorability. It has been gratifying to watch this research program develop beyond me. Thank you Devi for your clarity of thought, Wilma for your drive and ambition, and Aditya for your boundless ability to get cool things done. Thank you Zoya, not only for our work on memorability but also for sparking many fun projects beyond. Your ability to energize and organize people has been inspiring.

Thank you Joseph and Andrew, who have been constant friends as well as colleagues. I will miss our late night discussions at lab, even when they got a bit philosophical. Thank you Joseph for encouraging me to dream big. Thank you Andrew for keeping me grounded, and always applying rigor and healthy skepticism to our work.

Thanks to all members of the two labs I have been part of, the T-Ruth group and the Oliva lab, and to the CSAIL vision group. It has been a pleasure working in such a welcoming and friendly environment.

Thanks to all my other friends during my time here, too numerous to name. Thanks for the conversations, the support, and the adventures.

Lastly, thanks to my family, for everything.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Clown fish live next to sea anemones, lightning is always accompanied by thunder. When looking at the world around us, we constantly notice which things go with which. These associations allow us to segment, interpret, and understand the world.

Why do we consider thunder and lightning to be associated? How did we learn the association in the first place? Through evolution and development, our hereditary line has been bombarded with a huge amount of raw visual data. Somewhere along the way we became able to make sense of it all. We began to see objects and events, rather than just of bunch of photons. How did we get from point A to point B? Put simply: how, and why, do we group physical "stuff" into the perceptual "things"?

The focus of this thesis is on the unsupervised discovery of perceptual structure. I will describe how visual associations can be learned based on how often different colors, textures, and other patterns occur next to each other in our natural environment. Then I will show how to use these associations to uncover meaningful visual groups, including segments and contours, objects and movie scenes. Along the way, we will explore the utility of using statistical models that are highly specific to the subset of the world in which they will be applied. In addition, we will see how several problems in perceptual organization – including object segmentation, contour detection, and similarity measurements – can be posed as particular kinds of statistical association and dissociation.

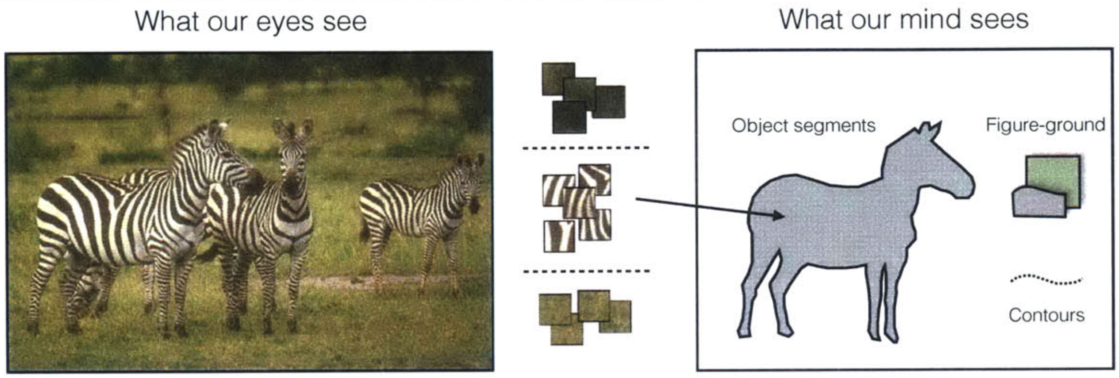


Figure 1.1: Illustration of perceptual organization. Statistical regularities in the world can be used to group raw pixels and patches into a variety of perceptual structures, such as object segments and contours.

This thesis deals with classical questions in perceptual science, about how and why we represent the world the way we do. A vast amount has been written on this. In order to situate us, I briefly review the prior work below.

■ 1.1 Perceptual organization and the role of structure in vision

When we look at the world, we cannot help but see coherent structure. A bunch of birds forms a "flock", the black and white stripes of a zebra are grouped as a "texture", and, despite the idiom, we certainly can see the "forest" for the trees. Even if we look at unfamiliar visual worlds, like a Jackson Pollock painting or a fluffy cumulus cloud, we still see familiar shapes, symmetries, and groupings. This ability is known as perceptual organization: we organize the visual data around us into coherent percepts. The eyes take in a bunch of haphazard photons yet somewhere deep in the brain we arrange them into objects, contours, and layers (Figure 1.1). This thesis explores how we can replicate and explain some of these abilities with computational models.

An early attempt to systemize the rules of perceptual organization was given by the Gestalt psychologists. These scientists came up with a set of heuristic rules to explain how raw visual data gets bound up into coherent percepts. These rules deal mostly with different notions of similarity and other heuristics, e.g., similarly colored circles are seen as a set, similarly oriented edge fragments are grouped into contours. This thesis adheres to an alternate and singular grouping principle: that perceptual groups are inferences about casual structure in the world. This idea also has a long history, going back at least to Helmholtz's notion of perception as unconscious inference [115], and has since been developed by many researchers, e.g., [67, 90, 109, 120].

Rather than simply describing perceptual phenomenology, these researchers asked: what is perceptual organization for? Witkin and Tenenbaum proposed that the structures we perceive can be viewed as explanations of the cause of the visual data [109, 120]. Certain visual patterns are highly unlikely to occur by accident, for example an arrangement of dots forming a perfect circle. We infer that someone, or some object, must have arranged the dots just so. The idea is that our visual system is on the lookout for non-accidental patterns because such patterns are clues toward the underlying physics and semantics of the world.

The idea of non-accidentalness has been widely applied in the vision literature. Among many other examples, non-accidentalness has been used to explain contour grouping [67], to disambiguate shape perception [40], and for image segmentation and clustering [35, 36].

This thesis presents a simple method for identifying non-accidental visual structure. Most previous works inferring non-accidental structures look for specific types of structure, e.g., smooth contours [67] or specific 3D shapes [11]. In contrast, we look for a more general class of non-accidents: any visual events that occur together more often than you would expect by chance. An advantage of our approach is that we do not need to develop domain specific models for each new problem we consider. Chapter 3 demonstrates this property by using the same generic modeling steps for three rather different problems: object segmentation, movie scenes segmentation, and the discovery of place types.

The idea of using co-occurrence rates to learn perceptual groups has some precedent. In an influential paper on auditory learning, Saffran et al. demonstrated that infants naturally pick up on regularities in the co-occurrence of sounds they hear, and they can use these to learn auditory groups [94]. The authors suggested this might explain how infants learn to segment speech. Subsequent researchers showed that this kind of "statistical learning" also works in vision, with participants able to learn groupings of nonsense shapes based on their rates of co-occurence [39].

This approach to perceptual organization rests on modeling the pairwise statistics of visual primitives. Many previous works have used other pairwise visual relationships to uncover structure. I review those methods next.

■ 1.2 Linking up visual data: similarity versus association

No pixel is an island. It is connected to the rest of the visual visual by a web of similarities, associations, and other relationships. Many approaches to identifying visual structure start by modeling the edges in this web.

Special focus has been devoted to appearance similarity. As far back as 1890, William James wrote that "a sense of sameness is the very keel and backbone of our thinking" [53]. Subsequent researchers have argued that similarity reveals how the brain organizes information, and underlies how we can generalize between alike stimuli (e.g., [98, 99, 111, 113]). Neuroscientists also use similarity to study cognitive organization. A currently popular technique is to measure "representational similarity": if two stimuli evoke similar patterns of brain activity, researchers infer that the brain represents those stimuli as alike [62]. This method can reveal visual object categories represented in inferotemporal cortex [20].

Similarity is also central to the Gestalt theory. Indeed, most of the Gestalt grouping rules describe grouping based on some notion of similarity: similarity of appearance (law of similarity), of position (law of proximity), or of motion (law of common fate) [112].

In computer vision and machine learning, similarity is also a powerful organizing principle. Popular similarity-based methods include nearest-neighbor inference algorithms and kernel methods (see, e.g., [24] for a review), which model visual data as a matrix of pairwise similarity measurements. Most computational work on image similarity either hand-specifies a similarity function (e.g., [119]) or learns similarity from supervision (e.g., [72, 103, 108, 117]).

Nonetheless, the problem of visual similarity is far from solved. The majority of image similarity models in computer vision focus on "look-alikes" – i.e. images that have similar colors, textures, edges, and other perceptual features (see left-most panel of Figure 1.2). However, many pairs of images may be considered similar despite being very different at the pixel level (e.g., right-most panel of Figure 1.2). Other pairs will have many properties in common but nonetheless be considered dissimilar. Murphy & Medin give the example of a plum and lawnmower: both are less than 10,000 kg, both can be dropped, etc [82]. Psychological theories of similarity and categorization have long confronted this paradox: that two things can be wildly dissimilar in some ways yet still be considered alike [41, 76, 79].

A key idea from this line of research is that for things to be considered alike, they need not be similar not in all aspects, but just must be similar in "ways that matter" (see [86] for further discussion). How can we identify which kinds of similarity matter and which do not?

While similarity may be a quite sensible grouping rule, this thesis proposes statistical association as a more fundamental principle. Statistical associations can be seen as a way of learning which kinds of similarity matter in a given domain. Indeed, I will show how certain kinds of visual similarity can be understood as approximate measures

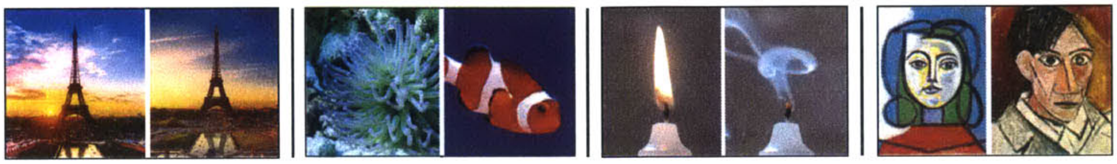


Figure 1.2: Visual associations come in a wide variety of forms. Some associations are perceived as visual similarity. Elsewhere two images may be associated but actually look quite dissimilar. The association may be semantic sameness, contextual proximity, a temporal link, stylistic similarity, or something else. Little work has modeled visual relationships beyond simple forms of similarity, such as the appearance similarity that relates the two photos of the Eiffel Tower above. While context models do exist, they usually operate over hand-designed semantic labels, rather than operating directly on raw images. This thesis looks for meaningful statistical associations in the raw data itself.

of association (Chapter 2 Section 2.7.1 and Section 2.8, and throughout Chapter 3). This approach does away with the need to define what it means for two things to be similar and instead measures affinities purely as a function of objectively measurable information in the environment.

The central technical product of this thesis is an affinity measure, based on statistical association, that can be used as an alternative to similarity-based affinities. Such an affinity measure could be used to discover many kinds of visual structure, including low dimensional subspaces [92, 110] or graphs [57]. However, I will focus just on visual grouping. I follow a standard approach to using pairwise affinities for grouping [95]. First a graph is constructed in which visual primitives are the nodes and edge weights are given by the affinity between each pair of primitives. Then the graph is partitioned so that primitives with high affinity are assigned to the same partition and those with low affinity are assigned to separate partitions. There are several ways performance such a partitioning. Throughout this thesis, the particular method I use is spectral clustering (see [116] for a review).

■ 1.3 An information-theoretic approach

With this backdrop, we can now give a preview of the technical meat. How do we actually measure visual associations? Our measure can be motivated as follows. Suppose you see two common events occur together, say a streetlight flickers right as a crow flies overhead. In all likelihood, you will discount this as mere coincidence. But when two *rare* events happen at once, perhaps hundreds of bats fill the midday sky as the sun enters an eclipse, we become suspicious that the events must be causally linked. Horace Barlow coined the term "suspicious coincidence" to describe events like this [8]. He defined these as cases in which the joint probability of two events is much higher than it would be if those events were independent (i.e., $P(A, B) \gg P(A)P(B)$). Such cases suggest that there is underlying structure that links the two events.

Barlow's idea is closely related to a measure of statistical association known as *pointwise mutual information* (PMI) [37]:

PMI for two events A and B is defined as:

$$\mathrm{PMI}(A, B) = \log \frac{P(A, B)}{P(A)P(B)}. \tag{1.1}$$

Taking expectation over A and B results in the regular mutual information:

$$\mathrm{MI}(A, B) = \mathbb{E}[\mathrm{PMI}(A, B)]. \tag{1.2}$$

This quantity is the log of the ratio between the observed joint probability of $\{A, B\}$ and the probability of this tuple were the two events independent. Equivalently, the ratio can be written as $\frac{P(A|B)}{P(A)}$, that is, how much more likely is event A given that we saw event B, compared to the base rate of event A.

What does PMI tell us about underlying causal structure? PMI will be zero if and only if A and B are independent events (as long as both A and B have non-zero probability). Therefore, non-zero values of PMI indicate there is a dependency between the events. The sign of PMI also conveys information. Negative values indicate that observing A predicts that B will *not* occur. For example, if A and B are mutually exclusive, PMI will go to negative infinity. Negative values of PMI indicate dissociations, whereas positive values indicate associations. Two dissociated events are unlikely to be caused by the same underlying process. To group data, we therefore use the signed value of PMI as an affinity. This serves to group associated events and separates events that are either independent or dissociated.

Although not as prevalent as regular multual information, PMI does pop up here and there in the literature. In language modeling, it is commonly used to find collocations [15, 19]. Similar notions been have used to describe sparse coding in the brain [96] and to explain inferences of everyday cognition [42]. Closely related are methods that weigh the relevance or informativeness of some data based on how much more likely it is in a context of interest than in a background context. Such methods have been applied for information retrieval [106], mid-level visual learning [55, 104], and for visual clustering [35, 36].

■ 1.4 Contributions

This thesis makes the following contributions. Our work:

1. Operationalizes classical ideas about non-accidentalness in perceptual organization.

2. Introduces a model of perceptual affinity based on statistical association rather than similarity.

3. Demonstrates the utility of local statistics for modeling these associations.

4. Shows that low-level pixel statistics can be surprisingly informative toward mid-

level grouping.

5. Derives an efficient way to scale up models of statistical association, simply using binary classifiers.

6. Provides general purposes machinery for applying the proposed framework to a variety of problems, by using deep convolutional neural nets.

■ 1.5 My other work

During grad school I've had the opportunity to work on a wide variety of projects. A motivating question for me has been how do we represent knowledge about the world, and why do these representations exist? This thesis investigates representations in the form of visual groups – segments, objects, movie scenes, etc.

In parallel, I have been studying knowledge representation in the form of memories. What do we remember, and why do we remember it? In [48], [47], and [49], we found that people tend to remember photos of people, social scenes, and close up, interactive objects. Surprisingly, aesthetic photos were forgettable on average, and other high level assessments of quality, such as "interestingness", were also negatively correlated with memorability. This might be explained by the fact that most beautiful photos in our dataset were landscapes, and landscapes are devoid of memorable objects. Nonetheless, it became clear that memorability is measure distinct from other measures of visual impact, and, in contrast to measures like interestingness, memorability can be objectively quantified using behavioral experiments.

We extended our work to study the memorability of faces [6, 7], infographics [12], and words [68], and the topic has been produced a vibrant body of research from other students and groups as well, e.g., [58] and [73].

Of particular relevance to the topic of this thesis is our recent work modeling memorability from a statistical perspective. The general hypothesis is that the statistics of our natural environment affect what we remember. This hypothesis is related to the classical idea in the literature that items that stand out from their context tend to be better remembered [59]. However, it has remained difficult to measure exactly what it means for a natural image to "stand out." In [13], we modeled this notion with a particular statistical model, measuring how unlikely it would be to see that image under the distribution images given by the surrounding context. This model can predict not only which images will be remembered, but also how a given image will change in memorability depending on the environment in which it is shown. It appears there, as it will in the rest of this thesis, that the statistics of the environment affect what knowledge our brains decide to represent.

I have also been interested in visual representations from an engineering perspective. What are useful representations for 3D shapes, and for complex scenes? These questions lead to a line of work on collage-based representations. In [21] my co-authors and I introduced a representation for 3D shape that models the world as a collage of remembered surface fragments. In [46], I extended the approach to represent a full scene as a collage of remembered objects.

Most recently, I have been interested in representing data beyond a single image. What's an effective way to represent a whole collection of photos? In [51], my co-authors and I modeled a collection of photos of an object class, e.g., 'tomatoes', in terms of the states (e.g., 'unripe', 'ripe') depicted in that collection. We additionally looked at the transformations that link the states together (e.g., 'ripening'). The idea is that each image of a tomato is related to each other tomato image through a structured transformation. These transformations – 'ripening', 'aging', 'breaking' – describe highly specific kinds of visual relationships, imbued with rich semantics. The present thesis is also occupied with visual relationships, but of a much more basic class: co-occurrence rates of visual events in space and time.

For a full list of my work during grad school please see my webpage at `mit.edu/phillipi`.

Chapter 2

Finding objects through suspicious coincidences in low-level vision

Partitioning images into semantically meaningful objects is an important component of many vision algorithms. In this chapter, we describe a method for detecting boundaries between objects based on a simple underlying principle: pixels belonging to the same object exhibit higher statistical dependencies than pixels belonging to different objects. We show how to derive an affinity measure based on this principle using pointwise mutual information, and we show that this measure is indeed a good predictor of whether or not two pixels reside on the same object. Using this affinity with spectral clustering, we can find object segments and boundaries in the image – achieving state-of-the-art results on the BSDS500 dataset. Our method produces pixel-level accurate boundaries while requiring minimal feature engineering.

This chapter is an extended version of [50].

■ 2.1 Introduction

Semantically meaningful contour extraction has long been a central goal of computer vision. Such contours mark the boundary between physically separate objects and provide important cues for low- and high-level understanding of scene content. Object boundary cues have been used to aid in segmentation [4, 5, 63], object detection

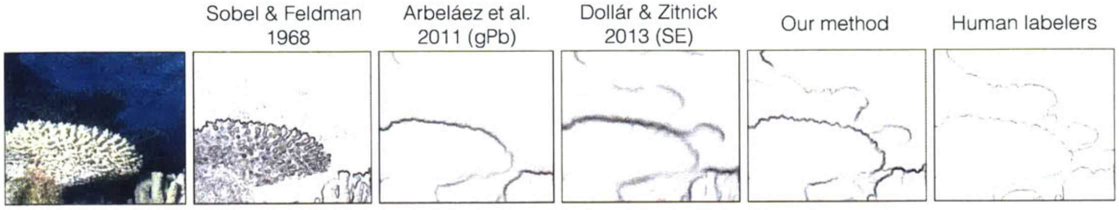


Figure 2.1: From left to right: Input image; Contours recovered by the Sobel operator [105]; Contours recovered by Dollár & Zitnick 2013 [28]; Contours recovered by Arbeláez et al. (gPb) [4]; Our recovered contours; Contours labeled by humans [4]. Sobel boundaries are crisp but poorly match human drawn contours. More recent detectors are more accurate but blurry. Our method recovers boundaries that are both crisp and accurate. Our method does so by suppressing edges in highly textured regions such as the coral in the foreground. Here, white and gray pixels repeatedly occur next to each other. This pattern shows up as a suspicious coincidence in the image's statistics, and our method infers that these colors must therefore be part of the same object. Conversely, pixel pairs that straddle the coral/background edges are relatively rare and our model assigns these pairs low affinity.

and recognition [83, 102], and recovery of intrinsic scene properties such as shape, reflectance, and illumination [9]. While there is no exact definition of the "objectness" of entities in a scene, datasets such as the BSDS500 segmentation dataset [4] provide a number of examples of human drawn contours, which serve as a good objective guide for the development of boundary detection algorithms. In light of the ill-posed nature of this problem, many different approaches to boundary detection have been developed [4, 28, 65, 121].

As a motivation for our approach, first consider the photo on the left in Figure 2.1. In this image, the coral in the foreground exhibits a repeating pattern of white and gray stripes. We would like to group this entire pattern as part of a single object. One way to do so is to notice that white-next-to-gray co-occurs suspiciously often. If these colors were part of distinct objects, it would be quite unlikely to see them appear right next to each other so often. On the other hand, examine the blue coral in the background. Here, the coral's color is similar to the color of the water behind the coral. While the change in color is subtle along this border, it is in fact a rather unusual sort of change

– it only occurs on the narrow border where coral pixels abut background water pixels. Pixel pairs that straddle an object border tend to have a rare combination of colors.

These observations motivate the basic assumption underlying our method, which is that the statistical association between pixels *within* objects is high, whereas for pixels residing on different objects the statistical association is low. We will use this property to detect boundaries in natural images.

One of the challenges in accurate boundary detection is the seemingly inherent contradiction between the "correctness" of an edge (distinguishing between boundary and non-boundary edges) and "crispness" of the boundary (precisely localizing the boundary). The leading boundary detectors tend to use relatively large neighborhoods when building their features, even the most local ones. This results in edges which, correct as they may be, are inherently blurry. Because our method works on surprisingly simple features (namely pixel color values and very local variance information) we can achieve both accurate *and* crisp contours. Figure 2.1 shows this appealing properties of contours extracted using our method. The contours we get are highly detailed (as along the top of the coral in the foreground) and at the same time we are able to learn the local statistical regularities and suppress textural regions (such as the interior of the coral).

It may appear that there is a chicken and egg problem. To gather statistics within objects, we need to already have the object segmentation. This problem can be bypassed, however. We find that natural objects produce probability density functions (PDFs) that are well clustered. We can discover those clusters, and fit them by kernel density estimation, without explicitly identifying objects. This lets us distinguish common pixel pairs (arising within objects) from rare ones (arising at boundaries).

In this paper, we only look at highly localized features – pixel colors and color variance in 3x3 windows. It is clear, then, that we cannot derive long feature vectors with sophisticated spatial and chromatic computations. How can we hope to get good performance? It turns out that there is much more information in the PDFs than one might at first imagine. By exploiting this information we can succeed.

Our main contribution is a simple, principled and unsupervised approach to contour detection. Our algorithm is competitive with other, heavily engineered methods. Unlike these previous methods, we use extremely local features, mostly at the pixel level, which allow us to find crisp and highly localized edges, thus outperforming other methods significantly when more exact edge localization is required. Finally, our method is unsupervised and is able to adapt to each given image independently. The resulting algorithm achieves state-of-the-art results on the BSDS500 segmentation dataset.

The rest of this chapter is organized as follows: we start by presenting related work, followed by a detailed description of our model. We then proceed to model validation, showing that the assumptions we make truly hold for natural images and ground truth contours. Then, we compare our method to current state-of-the-art boundary detection methods. Finally, we will discuss the implications of this work.

■ 2.2 Related work

Contour/boundary detection and edge detection are classical problems in computer vision, and there is an immense literature on these topics. It is out of scope for this paper to give a full survey on the topic, so only a small relevant subset of works will be reviewed here.

The early approaches to contour detection relied on local measurements with linear filters. Classical examples are the Sobel [31], Roberts [89], Prewitt [85] and Canny [14] edge detectors, which all use local derivative filters of fixed scale and only a few orientations. Such detectors tend to overemphasize small, unimportant edges and lead to noisy contour maps which are hard to use for subsequent higher-level processing.

The key challenge is to reduce gradients due to repeated or stochastic textures, without losing edges due to object boundaries.

As a result, over the years, larger (non-local) neighborhoods, multiple scales and orientations, and multiple feature types have been incorporated into contour detectors. In fact, all top-performing methods in recent years fall into this category. Martin et al. [78] define linear operators for a number of cues such as intensity, color and texture. The resulting features are fed into a regression classifier that predicts edge strength; this is the popular Pb metric which gives, for each pixel in the image the probability of a contour at that point. Dollár et al. [30] use supervised learning, along with a large number of features and multiple scales to learn edge prediction. The features are collected in local patches in the image.

Recently, Lim et al. [65] have used random forest based learning on image patches to achieve state-of-the-art results. Their key idea is to use a dictionary of human generated contours, called Sketch Tokens, as features for contours within a patch. The use of random forests makes inference fast. Dollár and Zitnick [28] also use random forests, but they further combine it with structured prediction to provide real-time edge detection. Ren and Bo [87] use sparse coding and oriented gradients to learn dictionaries of contour patches. They achieve excellent contour detection results on BSDS500.

The above methods all use patch-level measurements to create contour maps, with non-overlapping patches making independent decisions. This often leads to noisy and broken contours which are less likely to be useful for further processing for object recognition or image segmentation. Global methods utilize local measurements and embed them into a a framework which minimizes a global cost over all disjoint pairs of patches. Early methods in this line of work include that of Shashua and Ullman [97] and Elder and Zucker [32]. The paper of Shashua and Ullman used a simple dynamic programming approach to compute closed, smooth contours from local, disjoint edge fragments.

These globalization approaches tend to be fragile. More modern methods include a Conditional Random Field (CRF) presented in [88], which builds a probabilistic model for the completion problem, and uses loopy belief propagation to infer the closed contours. The highly successful gPb method of Arbeláez et al. [4] embeds the local Pb measure into a spectral clustering framework [70, 101]. The resulting algorithm gives long, connected contours higher probability than short, disjoint contours.

The rarity of boundary patches has been studied in the literature before, e.g. [127]. We measure rarity based on pointwise mutual information [37] (PMI). PMI gives us a value per patch that allows us to build a pixel-level affinity matrix. This local affinity matrix is then embedded in a spectral clustering framework [4] to provide global contour information. PMI underlies many experiments in computational linguistics [15, 19] to learn word associations (pairs of words that are likely to occur together), and recently has been used for improving image categorization [10]. Other information-theoretic takes on segmentation have been previously explored, e.g., [81]. However, to the best of our knowledge, PMI has never been used for contour extraction or image segmentation.

■ 2.3 Information-theoretic affinity

Consider the zebra in Figure 2.2. In this image, black stripes repeatedly occur next to white stripes. To a human eye, the stripes are grouped as a coherent object – the zebra. As discussed above, this intuitive grouping shows up in the image statistics: black and white pixels commonly co-occur next to one another, while white-green combinations are rarer, suggesting a possible object boundary where a white stripe meets the green background.

In this section, we describe a formal measure of the *affinity* between neighboring

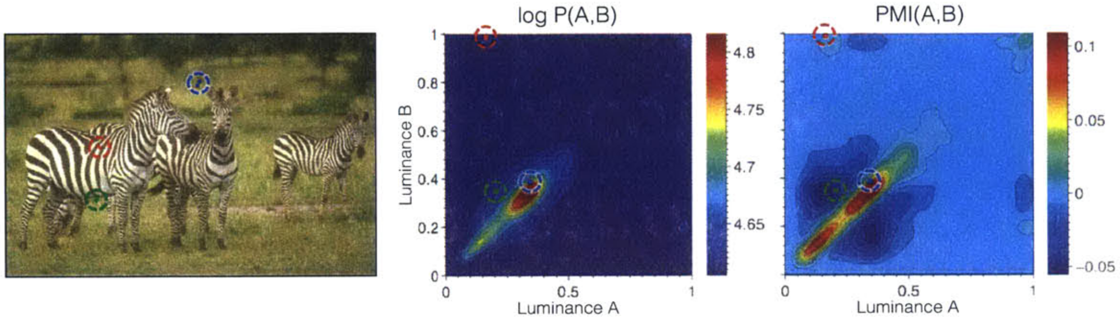


Figure 2.2: Our algorithm works by reasoning about the pointwise mutual information (PMI) between neighboring image features. Middle column: Joint distribution of the luminance values of pairs of nearby pixels. Right column: PMI between the luminance values of neighboring pixels in this zebra image. In the left image, the blue circle indicates a smooth region of the image where all points are on the same object. The green circle shows a region that contains an object boundary. The red circle shows a region with a strong luminance edge that nonetheless does not indicate an object boundary. Luminance pairs chosen from within each circle are plotted where they fall in the joint distribution and PMI functions.

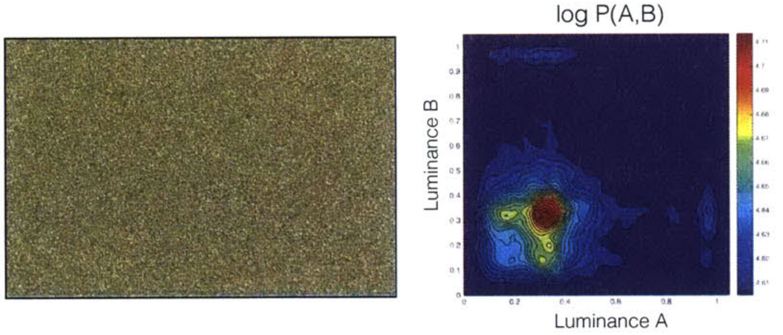


Figure 2.3: A randomly scrambled version of the zebra image in Figure 2.2. The joint density for this image factors as $P(A, B) \approx P(A)P(B)$. Non-symmetries in the distribution are due to finite samples; with infinite samples, the distribution would be perfectly separable. PMI is essentially comparing the hypothesis that the world looks like this scrambled image to the reality that the world looks like the zebra photo in Figure 2.2.

image features, based on statistical association. We denote a generic pair of neighboring features by random variables A and B, and investigate the joint distribution over pairings $\{A, B\}$.

Let $p(A, B; d)$ be the joint probability of features A and B occurring at a Euclidean distance of d pixels apart. We define $P(A, B)$ by computing probabilities over multiple distances:

$$P(A, B) = \frac{1}{Z} \sum_{d=d_0}^{\infty} w(d)p(A, B; d), \tag{2.1}$$

where w is a weighting function which decays monotonically with distance d (Gaussian in our implementation), and Z is a normalization constant. We take the marginals of this distribution to get $P(A)$ and $P(B)$.

In order to pick out object boundaries, a first guess might be that affinity should be measured with joint probability $P(A, B)$. After all, features that always occur together probably should be grouped together. For the zebra image in Figure 2.2, the joint distribution over luminance values of nearby pixels is shown in the middle column. Overlaid on the zebra image are three sets of pixel pairs in the colored circles. These pairs correspond to pairs $\{A, B\}$ in our model. The pair of pixels in the blue circle are both on the same object and the joint probability of their colors – green next to green – is high. The pair in the bright green circle straddles an object boundary and the joint probability of the colors of this pair – black next to green – is correspondingly low.

Now consider the pair in the red circle. There is no physical object boundary on the edge of this zebra stripe. However, the joint probability is actually lower for this pair than for the pair in the green circle, where an object boundary did in fact exist. This demonstrates a shortcoming of using joint probability as a measure of affinity. Because there are simply more green pixels in the image than white pixels, there are more chances for green accidentally show up next to any arbitrary other color – that is, the joint probability of green with any other color is inflated by the fact that most pixels in the image are green.

In order to correct for the baseline rarity of features A and B, we instead model affinity with a statistic related to *pointwise mutual information*:

$$\mathrm{PMI}_\rho(A,B) = \log \frac{P(A,B)^\rho}{P(A)P(B)}. \tag{2.2}$$

When $\rho = 1$, PMI_ρ is precisely the pointwise mutual information between A and B discussed in the introduction of this thesis. To recap, this quantity is the log of the ratio between the observed joint probability of $\{A, B\}$ in the image and the probability of this tuple were the two features independent. Equivalently, the ratio can be written as $\frac{P(A|B)}{P(A)}$, that is, how much more likely is observing A given that we saw B in the same local region, compared to the base rate of observing A in the image. When $\rho = 2$, we have a stronger condition: in that case the ratio in the log becomes $P(A|B)P(B|A)$. That is, observing A should imply that B will be nearby and vice versa. As it is unclear a priori which setting of ρ would lead to the best segmentation results, we instead treat ρ as a free parameter and select its value to optimize performance on a training set of images (see Section 2.4).

In the right column of Figure 2.2, we see the pointwise mutual information over features A and B. This metric appropriately corrects for the baseline rarities of white and black pixels versus gray and green pixels. As a result, the pixel pair between the stripes (red circle), is rated as more strongly mutually informative than the pixel pair that straddles the boundary (green circle). In Section 2.7.1 we empirically validate that PMI_ρ is indeed predictive of whether or not two points are on the same object.

Figure 2.3 shows what the zebra image would look like if adjacent colors were actually independent. The log joint distribution over this scrambled image is plotted to the right, which is approximately equal to $\log P(A)P(B)$. PMI is essentially measuring how far away the statistics of the actual zebra image are from this unstructured version of the zebra image. Specifically, the PMI function for the zebra image (right column of Figure 2.2) is equal to the distribution in the middle column of Figure 2.2 minus the distribution on the right of Figure 2.3 (up to sampling approximations).

2.4 Learning the affinity function

In this section we describe how we model $P(A, B)$, from which we can derive $\text{PMI}_\rho(A, B)$. The pipeline for this learning is depicted in Figure 2.4(a) and (b). For each image on which we wish to measure affinities, we learn $P(A, B)$ specific to that image itself. Extensions of our approach could learn $P(A, B)$ from any type of dataset: videos, photo collections, images of a specific object class, etc. However, we find that modeling $P(A, B)$ with respect to the internal statistics of each test image is an effective approach for unsupervised boundary detection. The utility of internal image statistics has been previously demonstrated in the context of super-resolution and denoising [125] as well as saliency prediction [75].

Because natural images are piecewise smooth, the empirical distribution $P(A, B)$ for most images will be dominated by the diagonal $A \approx B$ (as in Figure 2.2). However, we are interested in the low probability, off-diagonal regions of the PDF. These off diagonal regions are where we find changes, including both repetitive, textural changes and object boundaries. In order to suppress texture while still detecting subtle object boundaries, we need a model that is able to capture the low probability regions of $P(A, B)$.

We use a nonparametric kernel density estimator [84] since it has high capacity without requiring an increase in feature dimensionality. We also experimented with a Gaussian Mixture Model but were unable to achieve the same performance as kernel density estimators.

Kernel density estimation places a kernel of probability density around every sample point. We need to specify on the form of the kernel and the number of samples. We used Epanechnikov kernels (i.e. truncated quadratics) owing to their computational efficiency and their optimality properties [33], and we place kernels at 10000 sample points per image. Samples are drawn uniformly at random from all locations in the image. First a random position x in the image is sampled. Then features A and B are sampled from image locations around x, such that A and B are d pixels apart. The sampling is done with weighting function $w(d)$, which is monotonically decreasing and gives maximum weight to $d = 2$. The vast majority of samples pairs $\{A, B\}$ are within distance $d = 4$ pixels of each other.

Epanechnikov kernels have one free parameter per feature dimension: the bandwidth of the kernel in that dimension. We select the bandwidth for each dimension through leave-one-out cross-validation to maximize the data likelihood. Specifically, we compute the likelihood of each sample given a kernel density model built from all the remaining samples. As a further detail, we bound the bandwidth to fall in the range $[0.01, 0.1]$ (with features scaled between $[0, 1]$) – this helps prevent overfitting to imperceptible details in the image, such as jpeg artifacts in a blank sky. To speed up evaluation of the kernel density model, we use the kd-tree implementation of Ihler and Mandel [44]. In addition, we smooth our calculation of PMI_ρ slightly by adding a small regularization constant to the numerator and denominator of Eq. 2.2.

Our model has one other free parameter, ρ. We choose ρ by selecting the value that gives the best performance on a training set of images completely independent of the test set, finding $\rho = 1.25$ to perform best.

■ 2.5 Boundary detection

Armed with an affinity function to tell us how pixels should be grouped in an image, the next step is to use this affinity function for boundary detection (Figure 2.4 (c) and (d)). Spectral clustering methods are ideally suited in our present case since they operate on affinity functions.

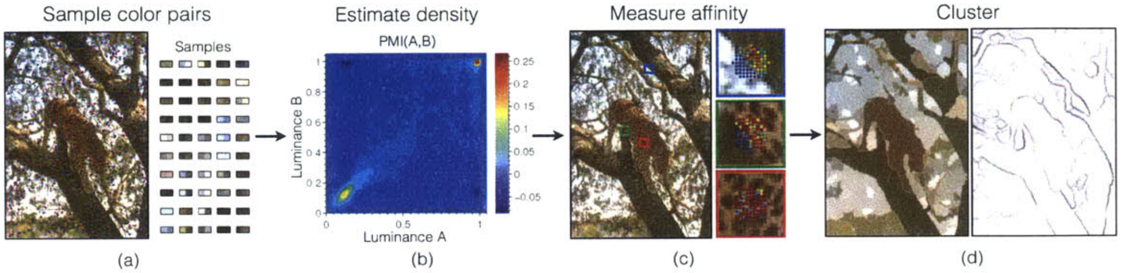


Figure 2.4: Boundary detection pipeline: (a) Sample color pairs within the image. Red-blue dots represent pixel pair samples. (b) Estimate joint density P(A,B) and from this get PMI(A,B). (c) Measure affinity between each pair of pixels using PMI. Here we show the affinity between the center pixel in each patch and all neighboring pixels (hotter colors indicate greater affinity). Notice that there is low affinity across object boundaries but high affinity within textural regions. (d) Group pixels based on affinity (spectral clustering) to get segments and boundaries.

Spectral clustering was introduced in the context of image segmentation as a way to approximately solve the Normalized Cuts objective [100]. Normalized Cuts segments an image so as to maximize within segment affinity and minimize between segment affinity. To detect boundaries, we apply a spectral clustering using our affinity function, following the current state-of-the-art solution to this problem, gPb [4].

As input to spectral clustering, we require an affinity matrix, $\mathbf{W}$. We get this from our affinity function PMI_ρ as follows. Let i and j be indices into image pixels. At each pixel, we define a feature vector $\mathbf{f}$. Then, we define:

$$\mathbf{W}_{i,j} = e^{\mathrm{PMI}_\rho(\mathbf{f}_i,\mathbf{f}_j)} \tag{2.3}$$

The exponentiated values give us better performance than the raw PMI_ρ values. Since our model for feature pairings was learned on nearby pixels, we only evaluate the affinity matrix for pixels within a radius of 5 pixels from one another. Remaining affinities are set to 0.

In order to reduce model complexity, we make the simplifying assumption that different types of features are independent of one another. If we have M subsets of features, this implies that,

$$\mathbf{W}_{i,j} = e^{\sum_{k=1}^{M} \mathrm{PMI}_{\rho}(\mathbf{f}_i^k, \mathbf{f}_j^k)} \tag{2.4}$$

In our experiments, we use two feature sets: pixel color (in L*a*b* space) and the diagonal of the RGB color covariance matrix in a 3x3 window around each pixel. Thus for each pixel we have two feature vectors of dimension 3 each. Each feature vector is decorrelated using a basis computed over the entire image (one basis for color and one basis for variance).

Given $\mathbf{W}$, we compute boundaries by following the method of [4]: first we compute the generalized eigenvectors of the system $(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda \mathbf{D}\mathbf{v}$, where $\mathbf{D}_{i,i} = \sum_{j \neq i} \mathbf{W}_{i,j}$. Then we take an oriented spatial derivative over the first N eigenvectors with smallest eigenvalue ($N = 100$ in our experiments). This procedure gives a continuous-valued edge map for each of 8 derivative orientations. We then suppress boundaries that align with image borders and are within a few pixels of the image border. As a final post-processing step we apply the Oriented Watershed Transform (OWT) and create an Ultrametric Contour Map (UCM) [4], which we use as our final contour maps for evaluation.

In addition to the above approach, we also consider a multiscale variant. To incorporate multiscale information, we build an affinity matrix at three different image scales (subsampling the image by half in each dimension for each subsequent scale). To combine the information across scales, we use the multigrid, multiscale angular embedding algorithm of [69]. This algorithm solves the spectral clustering problem while enforcing that the edges at one scale are blurred versions of the edges at the next scale up.

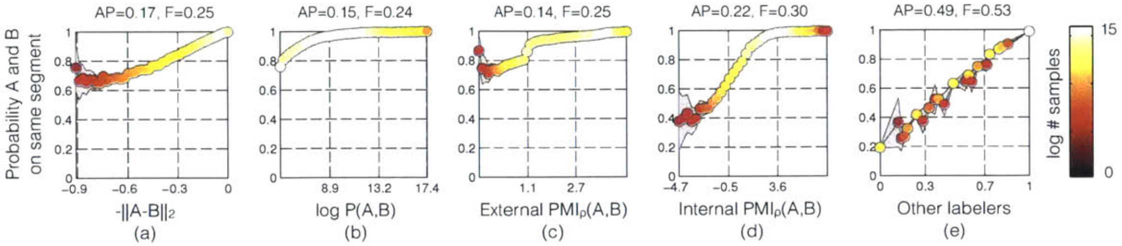


Figure 2.5: Here we show the probability that two nearby pixels are on the same object segment as a function of various cues based on the pixel colors A and B. From left to right the cues are: (a) color difference, (b) color co-occurrence probability based on internal image statistics, (c) PMI based on external image statistics, (d) PMI based on internal image statistics, and (e) theoretical upper bound using the average labeling of $N-1$ human labelers to predict the Nth. Color represents number of samples that make up each datapoint. Shaded error bars show three times standard error of the mean. Performance is quantified by treating each cue as a binary classifier (with variable threshold) and measuring AP and maximum F-measure for this classifier (sweeping over threshold).

■ 2.6 Segmentation

Segmentation is a complementary problem to edge detection. In fact, our edge detector automatically also gives us a segmentation map, since this is a byproduct of producing an Ultrametric Contour Map [4]. This ability sets our approach, along with gPb-owt-ucm, apart from many supervised edge detectors such as SE, for which a segmentation map is not a direct byproduct.

■ 2.7 Experiments

In this section, we present the results of a number of experiments. We first show that PMI is effective in detecting object boundaries. Then we show benchmarking results on the BSDS500 dataset. Finally, we show some segmentation results that are derived using our boundary detections.

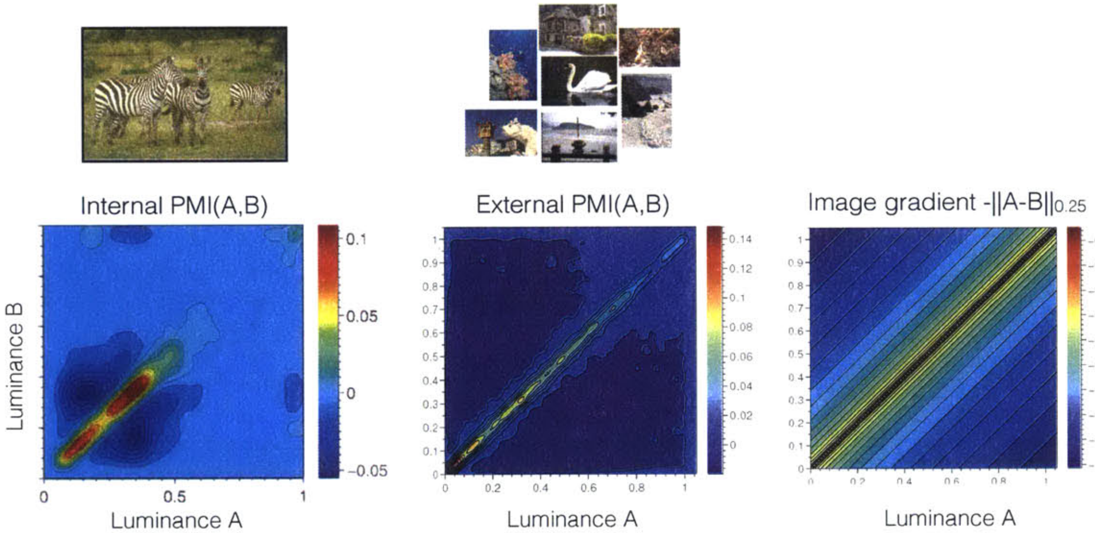


Figure 2.6: The PMI function, over luminance values A and B, learned from internal image statistics (left) versus external statistics (middle). Internal statistics refers to measuring luminance co-occurrences in a particular image, in this case the zebra photo at the top left. External statistics refers to measuring luminance co-occurrences aggregated over the entire BSDS500 training set (example images from which are shown top middle). The internal statistics suggest we should group black and white pixels, since these pairs show up with suspicious frequency on zebras. On most images, however, we certainly should not group black and white pixels since usually dark and light things are separate objects. The external statistics convey this, assigning high affinity only to pixels with similar values (along the diagonal). External PMI therefore results in a function quite similar to the classical Sobel filter [105], which detects edges where the image gradient is high (rightmost plot). From this perspective, the Sobel filter measures statistical dissociations with respect to the statistics of average images, whereas our method detects edges with respect to the statistics of the specific test image we apply it to.

2.7.1 Is PMI_ρ informative about object boundaries?

Given just two pixels in an image, how well can we determine if they span an object boundary? In this section, we analyze several possible cues based on a pair of pixels, and show that PMI_ρ is more effective than alternatives.

Consider two nearby pixels with colors A and B. In Figure 2.5 we plot the probability that a random human labeler will consider the two pixels as lying on the same object segment as a function of various cues based on A and B.

To measure this probability, we sampled 20000 nearby pairs of pixels per image in the BSDS500 training set, using the same sampling scheme as in Section 2.4. For each pair of pixels, we also sample a random labeler from the set of human labelers for that image. The pixel pair is considered to lie on the same object segment if that labeler has placed them on the same segment.

A first idea is to use color difference $||A - B||_2$ to decide if the two pixels span a boundary (Figure 2.5(a); note that we use decorrelated L*a*b* color space with values normalized between 0 and 1). Color difference has long been used as a cue for boundary detection and unsurprisingly it is predictive of whether or not A and B lie on the same segment.

Beyond using pixel color difference, boundary detectors have improved over time by reasoning over larger and larger image regions. But is there anything more we can squeeze out of just two pixels?

Since boundaries are rare events, we may next try $\log P(A, B)$. As shown in Figure 2.5(b), rarer color combinations are indeed more likely to span a boundary. However, $\log P(A, B)$ is still a poor predictor.

Can we do better if we use PMI? In Figure 2.5(c) and (d) we show that, yes, $\mathrm{PMI}_\rho(A, B)$ (with $\rho = 1.25$) is quite predictive of whether or not A and B lie on the same object. Further, comparing Figure 2.5(c) and (d), we find that it is important

Algorithm	ODS	OIS	AP
Canny [14]	0.60	0.63	0.58
Mean Shift [22]	0.64	0.68	0.56
NCuts [23]	0.64	0.68	0.45
Felz-Hutt [38]	0.61	0.64	0.56
gPb [4]	0.71	0.74	0.65
gPb-owt-ucm [4]	0.73	0.76	0.73
SCG [121]	**0.74**	0.76	0.77
Sketch Tokens [65]	0.73	0.75	0.78
SE [28]	**0.74**	0.76	0.78
Our method – SS, grayscale	0.67	0.69	0.71
Our method – SS, color	0.72	0.75	0.77
Our method – SS, color + var	0.73	0.76	**0.79**
Our method – MS, color + var	**0.74**	**0.77**	0.78

Table 2.1: Evaluation on BSDS500

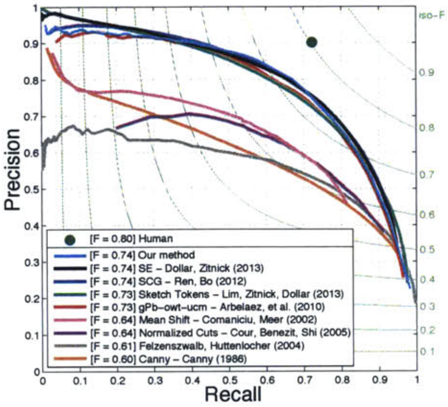


Figure 2.7: Precision-recall curve on BSDS500. Figure copied from [28] with our results added.

that the statistics for PMI_ρ be adapted to the test image itself. Figure 2.5(c) shows the result when the distribution $P(A, B)$ is learned over the entire BSDS500 training set. These *external* statistics are poorly suited for modeling individual images. On the other hand, when we learn $P(A, B)$ based on color co-occurrences *internal* to an image, PMI_ρ is much more predictive of the boundaries in that image (Figure 2.5(d)). Figure 2.6 explores this point further, showing that external statistics end up learning a function quite similar to color similarity. From this perspective, edge detectors based on color similarity, like the classical Sobel [105] and Canny filters [14] are approximately implementing a PMI decision rule just with respect to average images rather than the specific test image they are applied to.

■ 2.7.2 Benchmarks

We run experiments on four versions of our algorithm: single scale using only pixel luminance as the feature (labeled as *SS, grayscale*), single scale using pixel colors as the features (*SS, color*), single scale using color and variance features (*SS, color + var*), and multiscale with both color and variance features (*MS, color + var*). Where possible, we compare against the top performing previous contour detectors. We choose the Structured Edges (SE) detector [28] and gPb-owt-ucm detector [4] to compare against more extensively. These two methods currently achieve state-of-the-art results. SE is representative of the supervised learning approach to edge detection, and gPb-owt-ucm is representative of affinity-based approaches, which is also the category into which our algorithm falls.

BSDS500: The Berkeley Segmentation Dataset [4, 77] has been frequently used as a benchmark for contour detection algorithms. This dataset is split into 200 training images, 100 validation images, and 200 test images. Although our algorithm requires no extensive training, we did tune our parameters (in particular ρ) to optimize performance on the validation set. In Table 2.1 and Figure 2.7, we report our performance on the test set. ODS refers to the F-measure at the optimal threshold across the entire dataset. OIS refers to the per-image best F-measure. AP stands for area under the precision-recall curve. On each of these popular metrics, we match or outperform the state-of-the-art. It is also notable that our *SS, color* method gets results close to the state-of-the-art, as this method only uses *pixel pair colors* for its features. We believe that this result is noteworthy as it shows that with carefully designed nonlinear methods, it is possible to achieve excellent results without using high-dimensional feature spaces and extensive engineering.

In Figure 2.8 we show example detections by our algorithm on the BSDS500 test set. These results are with our *MS, color + var* version with $\rho = 1.25$. We note that our results have fewer boundaries due to texture, and crisper boundary localization. In Figure 2.9, we compare results of segmentations with our contours to those of gPb contours. Notice that in the coral image, our method recovers the precise shape of the bottom, reddish coral, while gPb-owt-ucm misses some major features of the contour. Similarly, in the bird image, our method captures the beak of the top bird, whereas

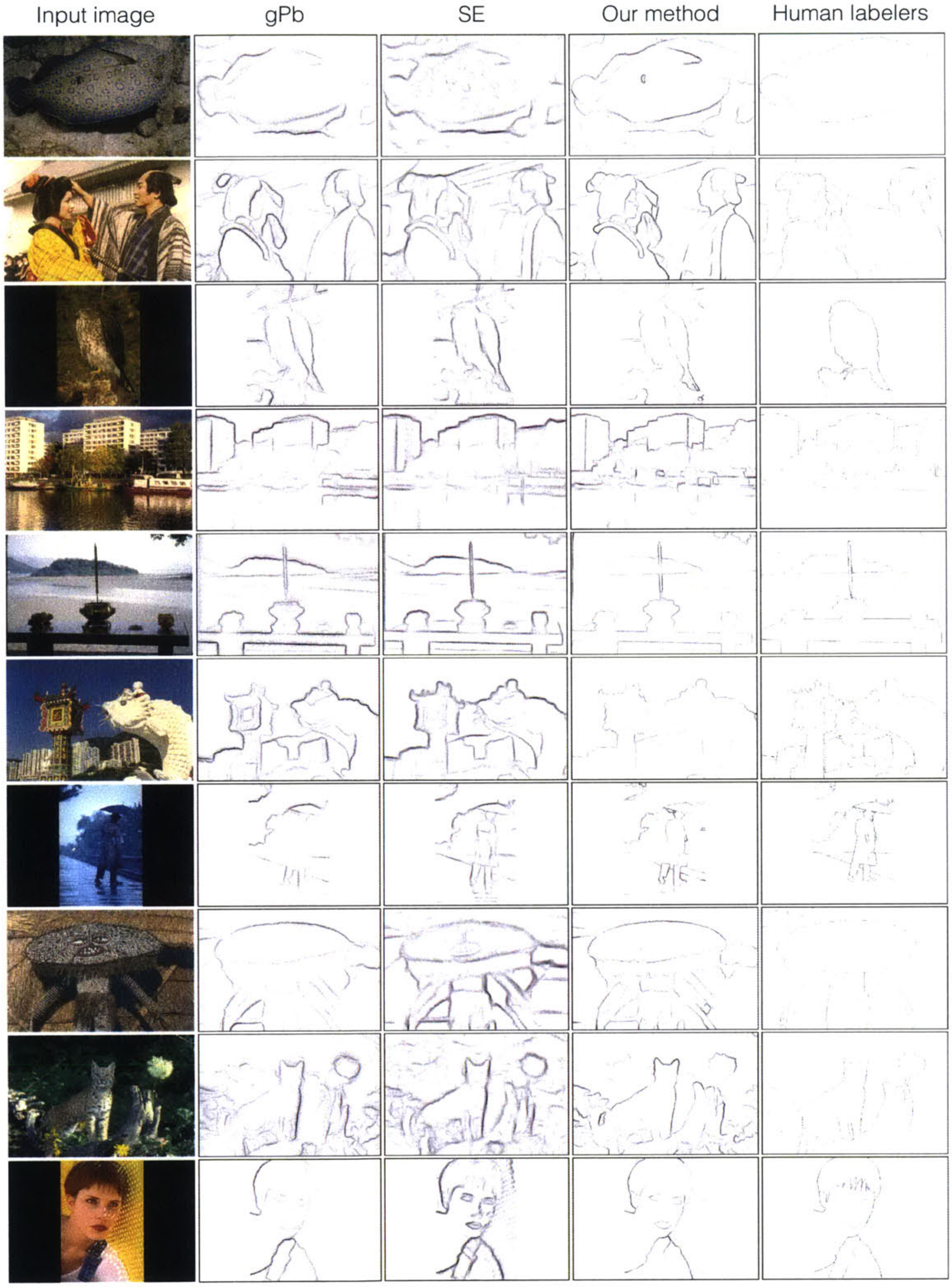


Figure 2.8: Contour detection results for a few images in the BSDS500 test set, comparing our method to gPb [4] and SE [28]. In general, we suppress texture edges better (such as on the fish in the first row), and recover crisper contours (such as the leaves in upper-right of the fifth row). Note that here we show each method without edge-thinning (that is, we leave out non-maximal suppression in the case of SE, and we leave out OWT-UCM in the case of gPb and our method).

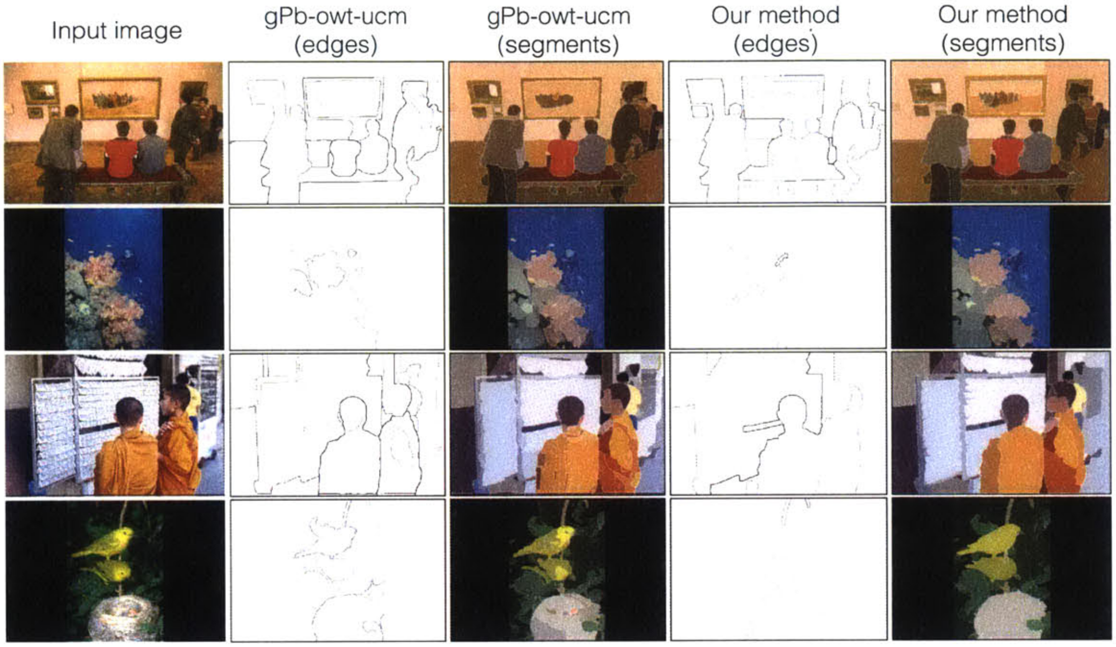


Figure 2.9: Example segmentations for a few images in the BSDS500 test set, comparing the results of running OWT-UCM segmentation on our contours and those of gPb [4].

gPb-owt-ucm smooths it away.

Is color necessary? Different objects tend to have different color compositions, and this is why color dissociations can be used to find boundaries between objects. If we remove color, and only look at luminance statistics, do objects still cluster in the same way? Interestingly, while numerical performance does drop somewhat (Table 2.1), many meaningful groupings can still be found just in luminance space. We motivated PMI using the example of the black-next-to-white pixels on a zebra. Figure 2.10 shows that our method can indeed group this pattern without additional color information, but adding color certainly helps as well.

High resolution edges: One of the striking features of our algorithm is the high resolution of its results. Consider the white object in Figure 2.11. Here our algorithm is able to precisely match the jagged contours of this object, whereas gPb-owt-ucm incurs much more smoothing. As discussed in the introduction, good boundary detections

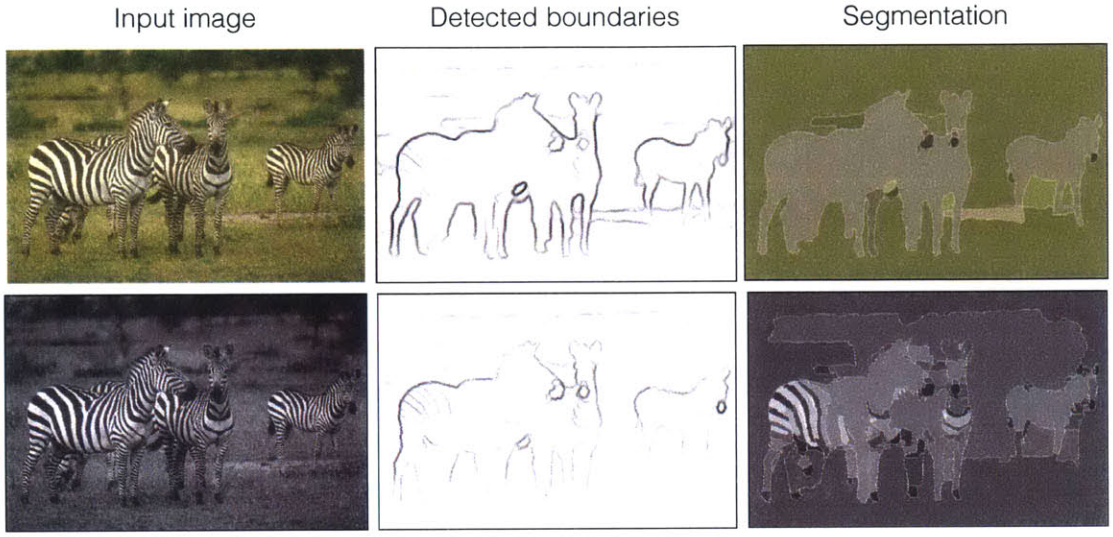


Figure 2.10: Color helps but many meaningful groupings can be found just in grayscale images.

should be both "correct" (detecting real object boundaries) and "crisp" (precisely localized along the object's contour). The standard BSDS500 metrics do not distinguish between these two criteria.

However, the benchmark metrics do include a parameter, r, related to crispness. A detected edge can be r pixels away from a ground truth edge and still be considered a correct detection. The standard benchmark code uses $r = 4.3$ pixels for BSDS500 images. Clearly, this default setting of r cannot distinguish whether or not an algorithm is capturing details above a certain spatial frequency. Varying r dramatically affects performance (Figure 2.12). In order to benchmark on the task of detecting "crisp" contours, we evaluate our algorithm on three settings of r: r_0, $r_0/2$, and $r_0/4$, where $r_0 = 4.3$ pixels, the default setting.

In Figure 2.12, we plot our results and compare against SE (with non-maximal suppression) and gPb-owt-ucm. While all three methods perform similarly at $r = r_0$, our method increasingly outperforms the others when r is small. This quantitatively

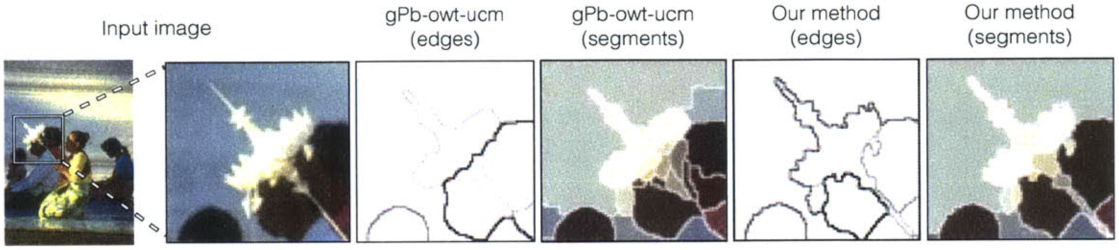


Figure 2.11: Here we show a zoomed in region of an image. Notice that our method preserves the high frequency contour variation while gPb-owt-ucm does not.

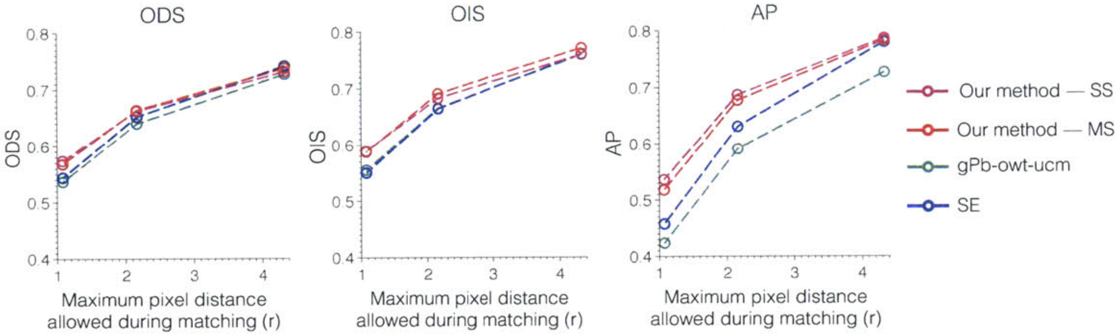


Figure 2.12: Performance as a function of the maximum pixel distance allowed during matching between detected boundaries and ground truth edges (referred to as r in the text). When r is large, boundaries can be loosely matched and all methods do well. When r is small, boundaries must be precisely localized, and this is where our method most outperforms the others. For our method in these plots, we use both color and variance features.

demonstrates that our method is matching crisp, high resolution contours better than other state-of-the-art approaches.

Consensus labels: Recently, Hou et al. [43] pointed out that many of the "ground truth" contours in BSDS are perceptually quite weak, and may not be indicative of real object boundaries. To account for this issue, Hou et al. suggested benchmarking algorithms against a subset of "consensus labels" in the BSDS dataset. Each BSDS image was labeled by several humans. Consensus labels are those pixels that *all* labelers for a given image mark as a contour. To see how well we match boundaries that all labelers agree on, we benchmark our algorithm against consensus labels. Results

Algorithm	ODS	OIS	AP
gPb-owt-ucm [4]	0.59	0.65	0.44
SE [28]	0.59	0.62	**0.58**
Our method – MS, color + var	**0.61**	**0.68**	0.56

Table 2.2: Evaluation on BSDS500 consensus edges [43]. We achieve a substantial improvement over the state-of-the-art in the ODS and OIS measures.

are listed in Table 2.2. On the ODS and OIS measures, our algorithm significantly outperforms SE and gPb-owt-ucm, while on AP, SE achieves the best results.

Speed: Recently several edge detectors have been proposed that optimize speed while also achieving good results [28, 65]. The current implementation of our method is not competitive with these fast edge detectors in terms of speed. To achieve our *MS, color + var* results above, our highly unoptimized algorithm takes around 15 minutes per image on a single core of an Intel Core i7 processor.

However, we can tune the parameters of our algorithm for speed at some cost to resolution. Doing so, we can match our state-of-the-art results (ODS=0.74, OIS=0.77 AP=0.80 on BSDS500 using the standard $r = 4.3$) in about 30 seconds per image (again on a single core of an i7 processor). The tradeoff is that the resulting boundary maps are not as well localized (at $r = 1.075$, this method falls to ODS=0.52, OIS=0.53, AP=0.43, which is well below our full resolution results in Figure 2.12). The speed up comes from 1) downsampling each image by half and running our *SS, color + var* algorithm, 2) approximating $\mathrm{PMI}_\rho(A, B)$ using a random forest prior to evaluation of $\mathbf{W}$, and 3) using a fixed kernel bandwidth rather than adapting it to each test image. Code for both fast and high resolution variants of our algorithm is available at `mit.edu/pmi-boundaries`.

■ 2.7.3 Effect of ρ

In Figure 2.13, we investigate the effect of ρ (Equation 2.2) on finding object boundaries. ρ is the only explicitly trained parameter of our algorithm. Can we do without it? The

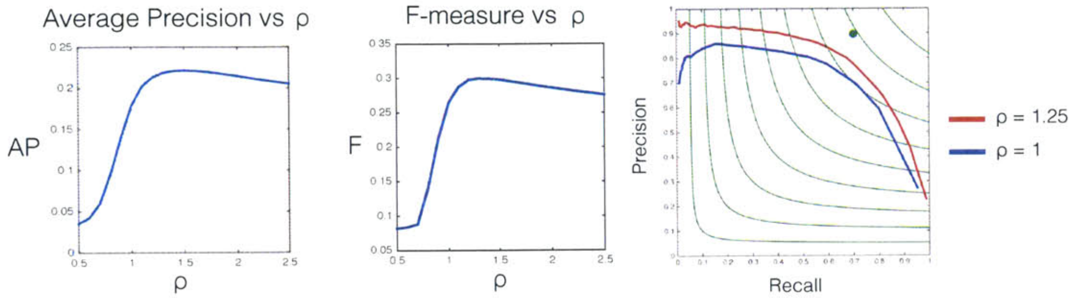


Figure 2.13: Left and middle: the effect of ρ (Equation 2.2) on performance at predicting whether or not two pixels are on the same object segment. Right: Boundary detection performance on BSDS500, comparing the parameter setting used in our benchmarks, $\rho = 1.25$, against using regular pointwise mutual information, i.e. $\rho = 1$.

the left two panels of Figure 2.13, we measure performance by thresholding PMI_ρ to produce a binary classification of whether or not two pixels lie on the same object segment. Sweeping over all possible thresholds gives an Average Precision (AP) and F-measure for this classifier. As can be seen, setting ρ slightly above 1 maximizes performance on these measures. Using $\rho = 1$, in which case Equation 2.2 exactly measures PMI, is somewhat worse.

We further test the necessity of learning ρ by setting $\rho = 1$ and benchmarking boundary detection performance on BSDS500 (Figure 2.13, right). Using the *MS, color + var* version of our algorithm, this setting of ρ achieves ODS=0.70, OIS=0.74 AP=0.70. These numbers are higher than older algorithms such as [14, 22, 23, 38], but substantially below our best results (using $\rho = 1.25$). Clearly ρ plays an important role. However, it remains unclear why PMI_ρ, with $\rho = 1.25$, should be more effective than regular old PMI, with $\rho = 1$. Future research should be directed at better understanding this parameter.

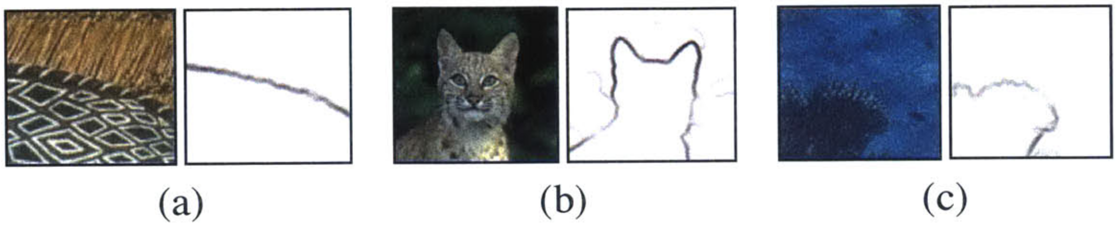


Figure 2.14: Three important properties of our approach: (a) it groups textures, (b) it also groups non-repetitive but mutually diagnostic patterns, (c) and it can pick up on subtle shifts in color that are nonetheless statistically prominent.

2.8 Discussion

In this chapter, we have presented an intuitive and principled method for contour detection which achieves state-of-the-art results. We have shown that, contrary to recent trends, it is possible to achieve excellent boundary detection results using very local information and low-dimensional feature spaces. This is achieved through the use of a novel statistical framework based on pointwise mutual information.

This statistical approach to perceptual grouping contrasts in several important ways from classical methods. Whereas many previous methods used *similarity* as the grouping rule (e.g., [3, 14, 105]), we use statistical *association*. We note that pixels can be associated but dissimilar in appearance, as in the case of the black and white stripes of a zebra.

The new approach has several important properties, displayed in Figure 2.14. Repetitive patterns show up as statistical regularities and can be grouped. This has the effect of suppressing patterns within textures while still highlighting the boundaries between different textures (Figure 2.14(a)). But textures are not the only kind of statistical regularity that can be detected. The facial features of the bobcat in Figure 2.14(b) are not textural but are nonetheless quite predictive of one another, and therefore will be grouped by PMI. One further way to understand our method is that it tells us which kinds of changes in an image matter and which do not. The boundary between reef and water in Figure 2.14(c) is marked by a subtle shift in shade of blue. While the change in color is small, it is nonetheless statistically significant, and detected by our method. Sometimes a small change will be statistically important, as in this example of turquoise next to blue. Other times what looks like a big change, e.g., a black stripe next to white, does not, in fact, indicate any real change in the underlying structure of the scene. By adapting to the internal structure of each test image it encounters, our method can handle both cases.

Chapter 3

Learning to group high-dimensional visual data

Chapter 2 showed that statistical associations between adjacent pixel colors, measured by PMI, can be very effective at localizing object boundaries. However, because we used a generative model for PMI, we were only able to learn associations at a very small scale: between pixel-level primitives. In contrast, in this chapter we show how to model PMI discriminatively, which allows us to apply PMI grouping to much higher-dimensional and in more generic settings.

The basic idea in this chapter is that we want to learn an affinity metric to organize visual primitives into semantic groups. We will use a Siamese neural network that can be trained to take two visual primitives as input and output an affinity between them [18]. One option would be to train the network to exactly predict whether or not two primitives are part of the same semantic group. However, that approach only will work if we already have a training set annotated with semantics. Instead we would like to discover meaningful groupings directly from sensory experience, in an unsupervised fashion. Therefore, instead of predicting whether or not two primitives are semantically similar, we predict whether or not they are spatially or temporally nearby. Because the world is smooth, with semantics changing slowly over space and time, predicting spacetime adjacency turns out to be a good approximation to predicting

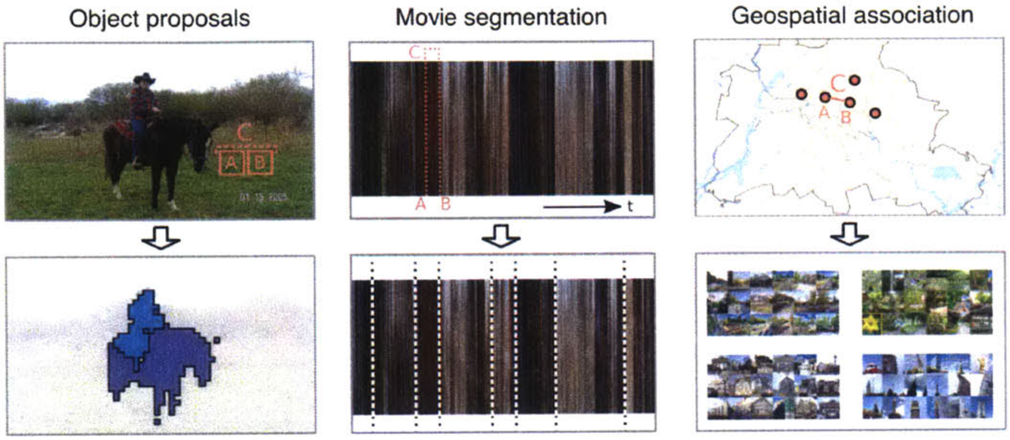


Figure 3.1: We model the dependencies between two visual primitives – e.g., patches, frames, photos – occurring in the same spatial or temporal context. In each example above, the primitives are labeled A and B and the context is labeled by $\mathcal{C}$. By learning which elements tend to appear near each other, we can discover visual groups such as objects (left), movie scenes (middle), and semantically related places (right).

semantic similarity.

Interestingly, while this story appears quite different on the surface from the story about PMI in Chapter 2, it turns out there is a close relationship: classifying proximity is equivalent to measuring certain PMI dependences, as we will see below. In this way, we can turn the problem of learning PMI from a hard probabilistic modeling problem into a much easier discriminative problem. Using the discriminative approach we are able to calculate PMI for high dimensional visual primitives such as patches and images. We apply the method to generate fast and accurate object proposals which are competitive with state-of-the-art supervised methods, as well as to automatic movie scene segmentation, and to grouping semantically related photographs.

This chapter is adapted from [52].

■ 3.1 Related work

Studies of perceptual grouping and associations have deep roots in the cognition and vision literature [8, 42, 45, 109]. Barlow [8] postulated that "suspicious coincidences" are a central feature by which the cerebral cortex discovers underlying structure in the world; this was further studied by Griffiths and Tenenbaum in the context of everyday cognition [42]. Witkin and Tenenbaum [109] argued that perceptual organization is a primitive level of visual inference, being fundamental to higher processes such as object recognition or depth perception.

Visual grouping is also a central problem in computer vision, showing up in the tasks of edge/contour detection [3, 14, 50] , (semantic) segmentation [71, 101], and object proposals [2, 61, 124], among others. Many papers in this field take the approach of first modeling the affinity between visual elements, then grouping elements with high affinity (e.g., [101]). Our work follows this approach. However, rather than using a hand-engineered grouping cue [101, 124], or learning to group with direct supervision [29, 61], we use spatial and temporal statistical association as the affinity. Grouping based on other notions of statistical association has received some prior attention [35, 36].

This chapter is especially related to recent work that uses "natural supervision" to train models [1, 26, 27, 64, 80, 93, 107, 118]. In these works, the common theme is to exploit spatial and/or temporal coherency as supervisory signals. These papers demonstrate that space and time can be powerful cues for learning visual structure. Our work justifies this approach by grounding it in information-theoretic principles. Further, while most of these past works are highly application specific, we show how to make the approach generic and applicable to a wide variety of settings.

■ 3.2 Model

We model the association between two visual primitives A and B, which may, for example, be image patches or video frames. This is defined with respect to a context variable $C \in \{0, 1\}$, which measures whether or not A and B "go together" in some sense to be specified. For example, C may indicate whether or not patches A and B

were sampled from adjacent locations in an image.

As in Chapter 2, we use PMI to measure statistical association. In the previous chapter, we defined a joint distribution $P(A, B)$ over adjacent colors in an image. Here, we make the notion of adjacency explicit through the variable $\mathcal{C}$. This way, we can treat A and B as i.i.d. random variables that only become dependent on one another when we condition on them both coming from the same context $\mathcal{C}$.

PMI for A, B, given context $\mathcal{C}$, is defined as:

$$\text{PMI}(A, B|\mathcal{C}) = \log \frac{P(A, B|\mathcal{C})}{P(A|\mathcal{C})P(B|\mathcal{C})} \tag{3.1}$$

Building an explicit probabilistic model for A, B and $\mathcal{C}$ is a challenging task, intimately related to the vast field of natural scene statistics. Many different models for natural image statistics have been proposed in recent years [56, 91, 126], but they are almost always limited in dimensionality. The reason is that even for 8×8 grayscale patches, building a probabilistic model requires modeling a 64 dimensional probability space.

We pose the modeling of PMI as a discriminative problem, allowing for easier learning by making weak assumptions about the data. Instead of estimating a full distribution over the values of A, B and $\mathcal{C}$, we predict $\mathcal{C}$ directly given A and B. This results in a binary classification problem, where the labels are obtained naturally from the context of A and B.

Observation: Let $A \perp\!\!\!\perp B$, $A \perp\!\!\!\perp \mathcal{C}$, and $B \perp\!\!\!\perp \mathcal{C}$, $\perp\!\!\!\perp$ denoting independence of two variables, then:

$$\log P(\mathcal{C}|A, B) \propto \text{PMI}(A, B|\mathcal{C}) \tag{3.2}$$

Proof: We start with the definition of the conditional distribution of $\mathcal{C}$ given A and B:

$$\log P(\mathcal{C}|A,B) = \log \frac{P(A,B,\mathcal{C})}{P(A,B)} = \log \frac{P(A,B|\mathcal{C})P(\mathcal{C})}{P(A,B)} \tag{3.3}$$

then using the independence relations above, we get:

$$\begin{aligned} \log \frac{P(A,B|\mathcal{C})P(\mathcal{C})}{P(A,B)} &= \log \frac{P(A,B|\mathcal{C})}{P(A)P(B)}P(\mathcal{C}) && (3.4) \\ &= \log \frac{P(A,B|\mathcal{C})}{P(A|\mathcal{C})P(B|\mathcal{C})}P(\mathcal{C}) && (3.5) \\ &= \mathrm{PMI}(A,B|\mathcal{C}) + \log P(\mathcal{C}) && (3.6) \end{aligned}$$

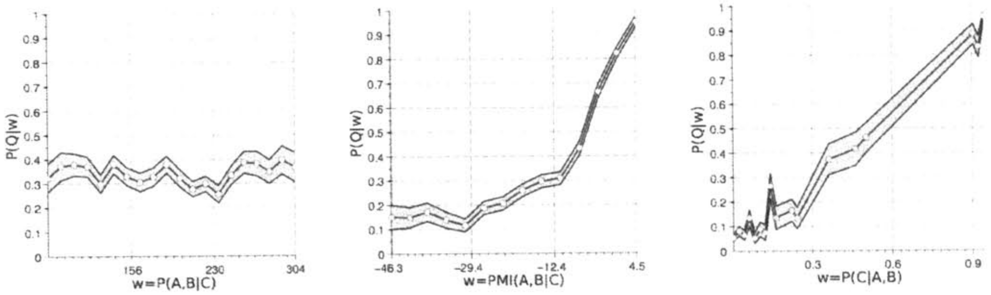
and finally, assuming a uniform prior over $\mathcal{C}$ we get the required result. ■

The uniform prior over $\mathcal{C}$ may instead be easily learned. Since now the output space of the problem is just a binary label, there is a large arsenal of tractable tools available to tackle the modeling problem.

The independence relationships we assume state that when randomly sampled, the visual primitives are independent from one another, and that A and B do not depend on whether they appear within a particular context. These assumptions are weaker than they may seem. Independence between A and B is easy to satisfy by construction. Specially, to train a classifier $P(\mathcal{C}|A,B)$, we sample primitives A and B i.i.d. from their the empirical distribution in some dataset. By construction then, A and B will be independent random variables. [1]

Independence between A and $\mathcal{C}$, and between B and $\mathcal{C}$, says that each primitive, on its own, tells us nothing about $\mathcal{C}$. This condition requires careful choice of $\mathcal{C}$. Not all choices will satisfy our assumptions. For example, if $\mathcal{C}$ indicates "are both from the

[1] In our exact implementation, we actually always train classifiers with 50% samples where $\mathcal{C} = 1$ and 50% where $\mathcal{C} = 0$. Since this modification only changes the prior $P(\mathcal{C})$, the resulting learned function is proportional to $P(\mathcal{C}|A,B)$ where A and B were sampled i.i.d., and therefore, the independence between A and B holds.



(a) $\mathcal{Q}$ vs. $P(A, B|\mathcal{C})$ (b) $\mathcal{Q}$ vs. $\mathrm{PMI}(A, B|\mathcal{C})$ (c) $\mathcal{Q}$ vs. $P(\mathcal{C}|A, B)$

Figure 3.2: Predictive qualities of different models for context and visual primitives. We calculate the output of each model for a test set of unseen pairs of points. We then bin the outputs (x-axis values) and for each bin calculate the average value of the desired grouping principle $\mathcal{Q}$ (y-axis values, along with standard deviation). The context function for these plots is spatial adjacency and the grouping principle is segmentation (not used in training, obtained from Pascal VOC 2012). As can be seen (left), the *joint* distribution of visual primitives conditioned on context is non-informative about grouping. PMI, however, is highly informative about grouping (middle), as is the output of the discriminative model $P(\mathcal{C}|A, B)$ (right).

bottom half of the image" then each patch on its own would be predictive of the answer and the independences would be violated (if one patch is a blue sky patch, we would immediately know the pair cannot be in the lower half of the image). On the other hand, if $\mathcal{C}$ indicates "are these two patches A and B adjacent", then independences holds. Seeing one patch tells us nothing about how likely an unobserved second patch is to be adjacent, since all patches have an equal number of adjacent patches [2].

Motivated by this observation, in each of our applications, we define $\mathcal{C}$ to measure spatial or temporal proximity. Each resulting classifier, $P(\mathcal{C}|A, B)$, is then a pure measure of the statistical association between its inputs conditioned on its output.

[2] note that this is only strictly true in an unbounded domain, such as a panorama; in regular photos, boundary patches have fewer neighbors than central patches, making independence only an approximation

■ 3.2.1 Modeling PMI explicitly with a GMM

In order to show that PMI is informative about interesting groupings, we build an explicit probabilistic model for PMI using Gaussian Mixture Models (GMMs). GMMs have been recently shown to be very effective at modeling image statistics and are therefore a natural choice for this task [126]. The context we model here is spatial adjacency: $\mathcal{C} = 1$ means that A and B come from adjacent locations within an image and we want to show that $\text{PMI}(A, B|\mathcal{C} = 1)$ is informative about image segmentation.

We train a GMM PMI model on 100,000 grayscale 8×8 patch pairs from the Pascal VOC 2012 dataset [34]. This is a dataset of everyday images, annotated by humans with object masks. We sample patch pair comes from non-overlapping adjacent locations in images in the Pascal training set. We generated a separate test set of unseen patch pairs from unseen images. For each test pair we have a label $\mathcal{Q}$ denoting whether or not the patches come from the same annotated object. Each point on the graph in Figure 3.2(b) is the average value of $\mathcal{Q}$ (y axis), averaged over all pairs that produced PMI values that fall in a certain bin (x-axis). As can be seen, PMI is highly informative about $\mathcal{Q}$ – low PMI values predict that $\mathcal{Q}$ will be 0, and high values predict that $\mathcal{Q} = 1$. Figure 3.2(a) shows that merely modeling the joint distribution $P(A, B|\mathcal{C})$ is not informative about $\mathcal{Q}$. The reason for this is that PMI normalizes the joint by the marginal probabilities of A and B.

■ 3.2.2 Modeling PMI with a CNN

Motivated by the results thus far and the relation in Eq. 3.2, we now show that training a classifier to predict $\mathcal{C}$ directly is as effective at predicting $\mathcal{Q}$ as a full probabilistic model such as the one in Section 3.2.1. Using the Caffe framework [54], we train a deep convolutional neural network (CNN) on the same training data as in Section 3.2.1. The network architecture is depicted in Figure 3.3. The structure is similar to Siamese nets

[18]: one branch of the CNN processes patch A while another processes patch B, with weight sharing between the two branches. The output of each branch is a feature vector. The output vectors are concatenated then passed through two fully connected layers. A final softmax layer produces predictions over $\mathcal{C}$.

All the 100,000 pairs from Section 3.2.1 were used as the positive examples ($\mathcal{C} = 1$), and we add another 100,000 of patch pairs sampled from random locations in the dataset as negative examples ($\mathcal{C} = 0$). We train the network to predict $\mathcal{C}$ given A and B. Figure 3.2(c) shows a plot of average $\mathcal{Q}$ as a function of predicted probability of $\mathcal{C} = 1$ (in exactly the same way as Figure 3.2(b), but the x-axis is now network output). As can be seen, the network output is highly informative about $\mathcal{Q}$, the ground truth grouping of objects in the dataset, even though the training signal was just spatial adjacency. Indeed the network produces values that correlate quite well with the GMM model. Spearman's rank correlation between outputs from the network and GMM is $\rho = 0.68$.

As noted above, such a classification framework scales much better than GMMs as A and B increase in dimension. We can now use this to learn models which employ much richer information than is possible with GMMs, such as larger patch sizes, color information, and so on. In the subsequent sections we show that these are useful and allow for tasks which would be currently impossible to tackle with full probabilistic modeling.

3.3 Learning visual associations

We apply the following generic approach to learning visual associations and groups:

1. Define A, B, and $\mathcal{C}$ based on the domain.

2. Learn $P(\mathcal{C}|A, B)$ using a CNN.

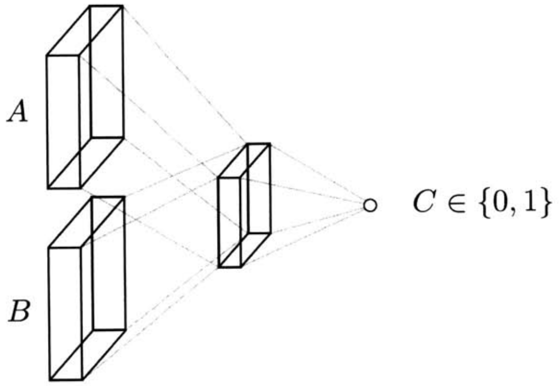


Figure 3.3: General structure of our CNNs for modeling functions of the form $P(\mathcal{C}|A, B)$.

Table 3.1: Our method compared to baselines at predicting $\mathcal{C}$ (spatial or temporal adjacency) and $\mathcal{Q}$ (semantic sameness) for three domains: image patch associations, video frame associations, and geospatial photo associations. In each column the first number is Average Precision and the second is F-measure at the threshold that maximizes this value.

	Patch associations		**Frame associations**		**Photo associations**	
Affinity measure	$\mathcal{C}$	$\mathcal{Q}\|\mathcal{C} = 1$	$\mathcal{C}$	$\mathcal{Q}\|\mathcal{C} = 1$	$\mathcal{C}$	$\mathcal{Q}$
Raw color	0.80 / 0.71	0.73 / 0.68	0.82 / 0.69	0.70 / 0.67	0.57 / 0.67	0.58 / 0.67
Mean color	0.84 / 0.74	0.73 / 0.70	0.83 / 0.73	0.72 / 0.68	0.55 / 0.67	0.57 / 0.67
Color histogram	0.91 / 0.82	0.78 / 0.71	**0.92** / **0.83**	**0.77** / 0.71	0.62 / 0.67	0.61 / 0.67
HOG	0.66 / 0.67	0.63 / 0.67	0.80 / 0.68	0.73 / 0.68	0.63 / 0.67	**0.76** / **0.75**
PMI	**0.94** / **0.86**	**0.81** / **0.74**	0.91 / 0.82	0.76 / **0.72**	**0.66** / **0.68**	**0.76** / 0.73

3. Setup a graph in which primitives A and B are nodes and edge weights are given by $P(\mathcal{C}|A, B)$. Then partition the graph into visual groups using spectral clustering.

This section will describe steps 1 and 2. The next section will describe step 3.

We examine three domains: 1) learning patch associations based on their spatial adjacency in images, 2) learning video frame associations based on their temporal adjacency in movies, and 3) learning geotagged photo associations based on their proximity in a city[3]

[3] All data and code will be made available online at www.anonymous.edu

Each task corresponds to a different choice of A, B, $\mathcal{C}$, and $\mathcal{Q}$. In each case, we analyze performance at predicting $\mathcal{C}$ and at predicting $\mathcal{Q}$, comparing our CNN to baseline grouping cues. Each baseline corresponds to a measure of the similarity between the primitives. Similarity measures like these are commonly used in visual grouping algorithms [3, 35]. The results are given in Table 3.3.

Image patch associations: We start with the same task as in Section 3.2.2, but scaled up so that A and B are now 17×17 pixel patches (with circular masks) and are in color. The context function is spatial adjacency. Positive examples ($\mathcal{C} = 1$) are pairs of adjacent patches (specifically, whose centers are within 48 pixels of each other, with no overlap between the patches) and negative examples ($\mathcal{C} = 0$) are pairs sampled from random locations across the dataset. We generate the dataset such that it will have 50% positive and 50% negative examples, with 200,000 patch pairs in total. The patches are sampled from the Pascal VOC 2012 training set.

We train a CNN (two convolutional layers, two fully connected layers) to model $P(\mathcal{C}|A, B)$. Figure 3.4 shows the most associated, but dissimilar patch pairs according to this network. Since very similar patches (in the L_2 distance sense) usually have high

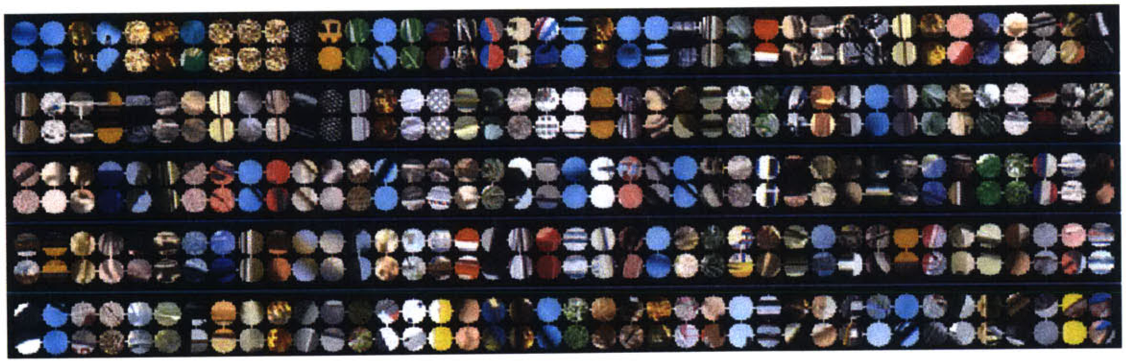


Figure 3.4: Highly associated, but not trivially similar, patches discovered by the model that learns patch associations. Each row shows pairs of highly associated patches. The patches in each pair are arranged one above the other. See text for details on how these patches are computed.

association we do not display very similar pairs. As can be seen, the network learns to associate patches with different kinds of structure such as texture, local features, and color similarities even if at the pixel level the patches are quite different.

To evaluate performance, we sample 10,000 patches from the Pascal VOC 2012 validation set, 50% with $\mathcal{C} = 1$ and 50% with $\mathcal{C} = 0$. As shown in Table 3.3 our network outperforms the baselines. In addition we measured performance at predicting $\mathcal{Q}$, where $\mathcal{Q}$ indicates whether or not the center pixel of the two patches lies on the same labeled object instance. Since $\mathcal{Q}$ is held out from training, we evaluate predicting $\mathcal{Q}$ on the training set. Even though it was only trained to predict $\mathcal{C}$, our method is effective at predicting $\mathcal{Q}$ as well, achieving an average precision (AP) of 0.81. This validates that spatial proximity, $\mathcal{C}$, is a good surrogate for "same object", $\mathcal{Q}$.

Movie frame associations: Our framework can also be applied to learning temporal associations. To test this, we set A and B to be frames, cropped and down sampled 32×32 pixels, from a set of seven movies sampled from the top 100 rated movies on IMDB[4]. In this setting, $\mathcal{C}$ indicates temporal adjacency – specifically, two frames are assigned $\mathcal{C} = 1$ if they are within 10 seconds from each other, and $\mathcal{C} = 0$ otherwise.

Again we train a CNN to model $P(\mathcal{C}|A, B)$ (three convolutional layers, two fully connected layers). To evaluate predicting $\mathcal{C}$, we train on four of the movies and test on the remaining three. Our method can learn this task fairly well, but is slightly outperformed by color histogram similarity. This shows that color histogram metrics is a good task-specific solution here. Note, however, that our method was able to learn this automatically, and in an unsupervised fashion.

How do our learned temporal associations relate to semantic visual scenes? To test this, we compared against DVD chapter annotations, setting $\mathcal{Q}$ to be "do these two

[4] http://www.imdb.com/

frames occur in the same DVD chapter?" We evaluate predicting $\mathcal{Q}$ on two movies from the training set since $\mathcal{Q}$ was held out during training (only two training movies had DVD chapter annotations). Our network achieves an AP of 0.76 on this task. Similar to above, we can then see that temporal adjacency, $\mathcal{C}$, is an effective surrogate for learning about semantic sameness, $\mathcal{Q}$.

Geospatial photo associations: A tourist in New York, walking from Time Square to Central Park, will experience a striking transition as she steps into the park. Dense glassy skyscrapers will suddenly be replaced by row after row of leafy elm. Like images and videos, the earth's surface varies slowly, punctuated by a few quick transitions here and there. Just as an object is a collection of associated patches, and a movie scene is a collection of associated frames, a visual *place* can be viewed a collection of associated photographs. Here we set A and B to be geotagged photos, cropped and down sampled to 32×32 pixels, and $\mathcal{C}$ indicates whether or not A and B are taken within 11 meters of one another.

Using the same CNN architecture as for the movie frame network, we again learn $P(\mathcal{C}|A, B)$, but for this new setting of the variables. We train on five cities selected from the MIT City Database [123], which is a collection of tourist photos of major cities such as New York, Paris, and Tokyo. We test predicting $\mathcal{C}$ on a held out set of three more cities from that dataset. We also test how well the network predicts place semantics. For this, we define $\mathcal{Q}$ as "do these two photos belong to the same place category?" We test this task on the LabelMe Outdoors dataset [66] for which each photo was assigned to one of eight place categories (e.g., "coast", "highway", "tall building"). Our network shows promising performance on this task, reaching 0.76 AP on predicting $\mathcal{Q}$. HOG similarity reaches the same performance, which corroborates past findings that HOG is effective at grouping related photos [25].

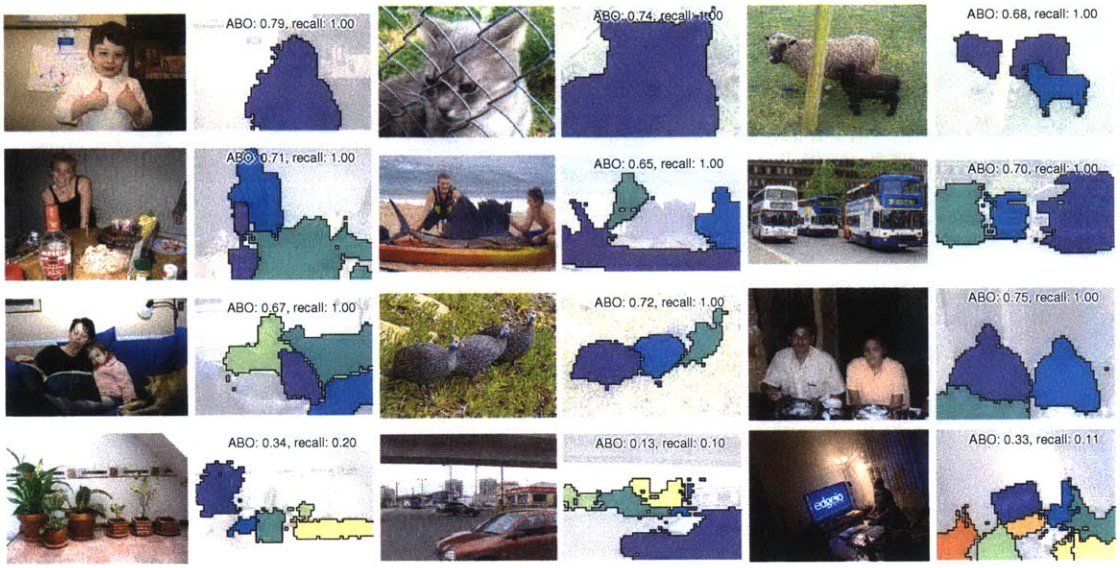


Figure 3.5: Example object proposals. Out of 100 proposals per image, we show those that best overlap the ground truth object masks. Average best overlap [60] and recall at a Jaccard index of 0.5 are superimposed over each result.

Notice that while HOG does well on associating photographs, it does not do well at associating movie frames nor image patches. On the other hand, color histogram similarity does well on associating movie frames, but fails at grouping everyday photographs – while frames in a movie scene all use the same color palette, tourist photos of the same place will have high color variance. Different grouping rules will be effective at different tasks. Our learning based approach has the advantage that it automatically figures out the appropriate grouping cue for each new domain, and thereby achieves good performance on all our tasks.

■ 3.4 From associations to visual groups

■ 3.4.1 Finding objects using patch associations

As demonstrated in Section 3.3, patches with high PMI usually belong to the same object. This suggests that we can localize objects in an image by grouping patches

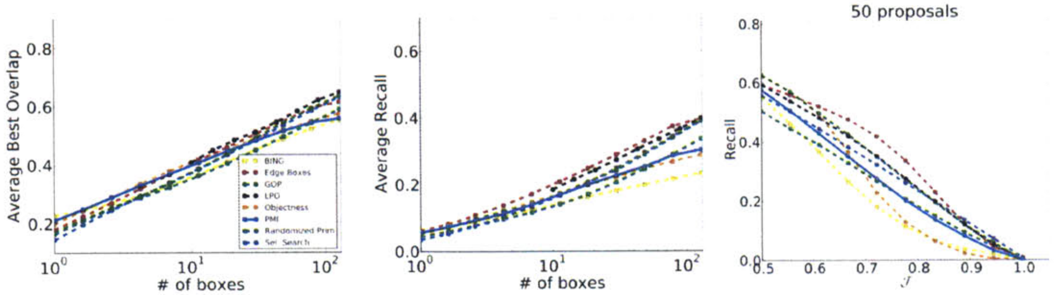


Figure 3.6: Object proposal results, evaluated on bounding boxes. Our unsupervised method (labeled PMI) is competitive with state-of-the-art supervised algorithms at proposing up to around 100 objects. The far-right figure is for 50 proposals per image. ABO is the average best overlap metric from [60], $\mathcal{J}$ is Jaccard index. The papers corresponding to the the labels in the left plot are: BING [17], EdgeBoxes [124], LPO [61], Objectness [2], GOP [60], Randomize Prim [74], Sel. Search [114].

with high PMI. We focus on the specific problem of "object proposals" [61, 124], where the goal is to localize all objects in an image.

We use the patch-associations $P(\mathcal{C}|A, B)$ defined in Section 3.3. To provide fair comparison against competitor methods, we only train on the Pascal VOC 2012 training set and test on the validation set. Given a test image, we sample all 17×17 patches at a stride of 8 pixels. We construct a graph in which each patch is a node and nodes are connected by an edge if the spatial distance between the patch centers is at least 17 pixels and no more than 33 pixels. Each patch is multiplied by a circular mask so that no two patches connected by an edge see any overlapping pixels. Each edge, indexed by i, j, is weighted by $\mathbf{W}_{i,j} = (P(\mathcal{C}_{i,j}|A_i, B_j))^{\alpha}$, resulting in the affinity matrix $\mathbf{W}$, where we use the value $\alpha = 20$ in our experiments.

To globalize the associations, we apply spectral clustering to the matrix $\mathbf{W}$. First we create the Laplacian eigenmap L for $\mathbf{W}$, using eigenvectors numbered from $2-16$. Each eigenvector is scaled by $\lambda^{-\frac{1}{2}}$ where λ is the corresponding eigenvalue. We then generate object proposals simply by applying k-means to the Laplacian eigenmap. To generate more than a few proposals, we run k-means multiple times with random restarts.

Qualitative results from our method are shown in Figure 3.5. In each case we show

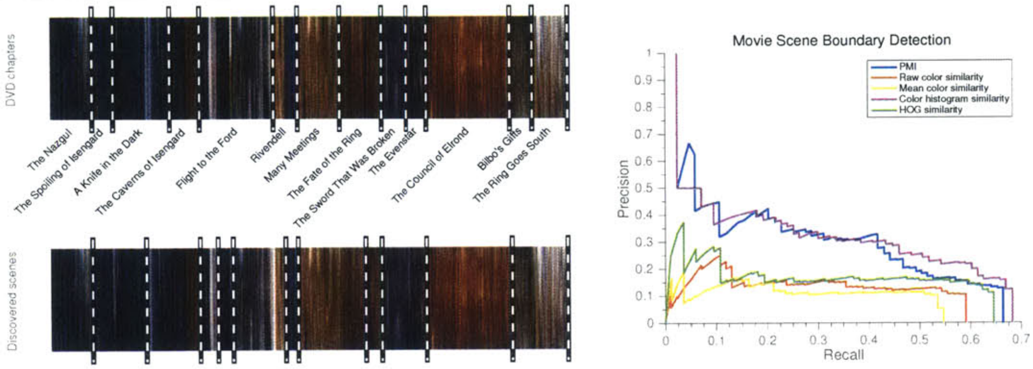


Figure 3.7: Movie scene segmentation results. On the left, we show a "movie barcode" for *The Fellowship of the Ring*; the top shows the DVD chapters and the bottom our recovered scene segmentation. On the right, we quantify our performance on this scene segmentation task; see text for details.

the proposals that have best overlap with the ground truth object masks for 100 proposals. We quantitatively compare against other state-of-the-art methods in Figure 3.6. Even though our method is unsupervised, it reaches performance comparable to recent supervised methods at proposing up to 100 objects per image. Our implementation runs in about 4 seconds per image on a 2015 Macbook Pro.

■ 3.4.2 Segmenting movies using frame associations

Just as objects are composed of sets of associated patches, scenes in a movie are composed of sets of associated frames. Here we show how the frame associations can be used to break a movie into coherent scenes, a problem that has received some prior attention [16, 122].

To segment a movie, we build a graph in which each frame is a node and all frames within ten seconds of one another are connected by an edge. We then weight the edges using the frame-associations $P(\mathcal{C}|A, B)$ (Section 3.3), and partition the graph using spectral clustering.

To evaluate, we use DVD chapter annotations as ground truth. We use the temporal-associations model from Section 3.3, evaluating on the two annotated movies from the

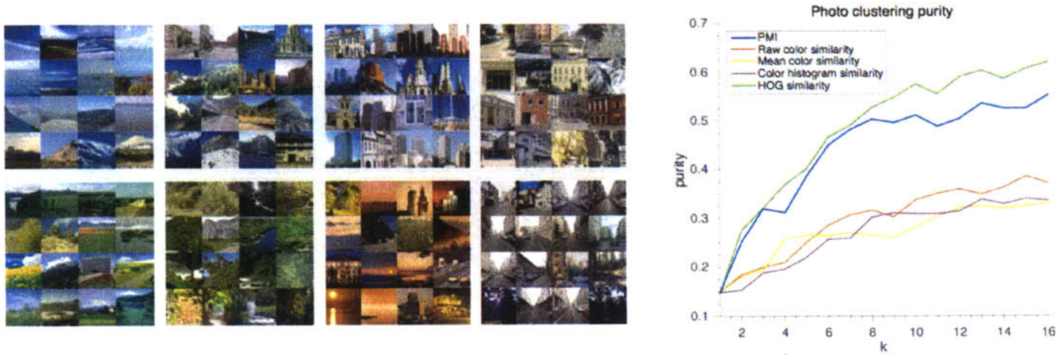


Figure 3.8: Left: Clustering the LabelMe Outdoor dataset [66] into 8 groups using PMI affinities. Random sample images are shown from each group. Right: Photo cluster purity versus number of clusters k. Note that PMI was learned on an independent dataset, the MIT City dataset [123].

training set (chapter annotations were held out during training, only temporal proximity was used as the training signal). Following a standard evaluation procedure in image boundary detection [3], we measure performance on the retrieval task of finding all ground truth boundaries. Figure 3.7(right) quantifies our performance at this task, compared to the baselines from Section 3.3. For each method, we scale the affinities as $\mathbf{W}^{\alpha}$, where α is selected to optimize test time performance, thereby showing an upper bound on achievable performance (setting α properly has a large effect on performance, so factoring its effect out in this way better reflects the real merit of each affinity measure). Our approach finds more boundaries with higher precision than these baselines. Figure 3.7 (left) shows an example segmentation of a section of *The Fellowship of the Ring*. The movie is displayed as a "movie barcode"[5] in which each frame is squished into a single column and time advances to the right. On top are the DVD chapter annotations, and on the bottom are our inferred boundaries.

■ 3.4.3 Discovering place types using geospatial associations

Taking the geospatial-associations model from Section 3.3, we cluster photos into coherent types of places. Here we create a fully connected graph between all photos in a

[5] http://moviebarcode.tumblr.com/

given collection, weight the edges with $P(C|A,B)$ and then apply spectral clustering to partition the collection. We test the purity of the clusters on LabelMe Outdoors dataset [66]. Clustering purity versus number of clusters k is given in Figure 3.8 (right), showing that our method is effective at discovering semantic place categories. Figure 3.8 (left) shows random sample images from each cluster after clustering into 8 categories. This clustering has 51% purity.

■ 3.5 Discussion

We have presented a general and principled approach to learning visual groupings, which requires no pre-defined labels. Instead our framework uses "natural" context as a supervisory signal by learning to predict which entities appear together in space or time. By doing so, we learn different clustering mechanisms for a variety of tasks.

Our approach achieves competitive results on object proposal generation, even when compared to supervised methods. Additionally, we demonstrated that the same method can be used to segment movies into scenes by learning to predict if two frames are close to one another in time. Finally, by learning to predict if two photographs were taken in nearby locations, we are able uncover semantic place categories. The principles underlying the framework are quite general and may be applicable to data in other domains, when there are natural context signals and groupings.

This chapter suggests an alternate explanation of the effectiveness of PMI at visual grouping. We showed that our PMI measures are equivalent to a classifier of spatial or temporal proximity. An intuitive explanation of their success is then: the world is smooth, so nearby stuff is often produced by the same underlying thing. Predicting whether or not two observations are nearby should therefore serve as a good proxy for predicting whether or not the two observations come from the same thing. Our analysis links this qualitative story to precise statements about information and dependence.

Chapter 4

Conclusion

In this thesis, I have explored how perceptual structure can arise as a natural consequence of statistical regularities in the world. We saw that objects and other semantic groups contain mutually informative patterns. I showed that the boundaries we perceive between objects can be explained as statistical dissociations between the stuff on either side. Finally, I have suggested that certain kinds of perceptual similarity can be viewed as particular measures of statistical association.

Chapters 2 and 3 offered two different ways to model statistical associations between visual data. Both models were related to pointwise mutual information, but the models differed in many other ways. Chapter 2 used a generative model over pixel co-occurrences within a single image whereas Chapter 3 employed a discriminative model over patches (and other high-dimensional data) using statistics measured across a dataset.

The intuitive story for the Chapter 2 is that colors that co-occur together much more often than chance should be grouped. The simplest story for Chapter 3 is rather different: we want to learn a similarity metric between different patches. We could do so by training a Siamese network to predict if two patches are semantically the same or different. But that would require expensive annotations to provide the supervision. Instead we use a cheap proxy, training the network to just decide if the two patches are adjacent (in space or time). This proxy learns almost the same thing as what we

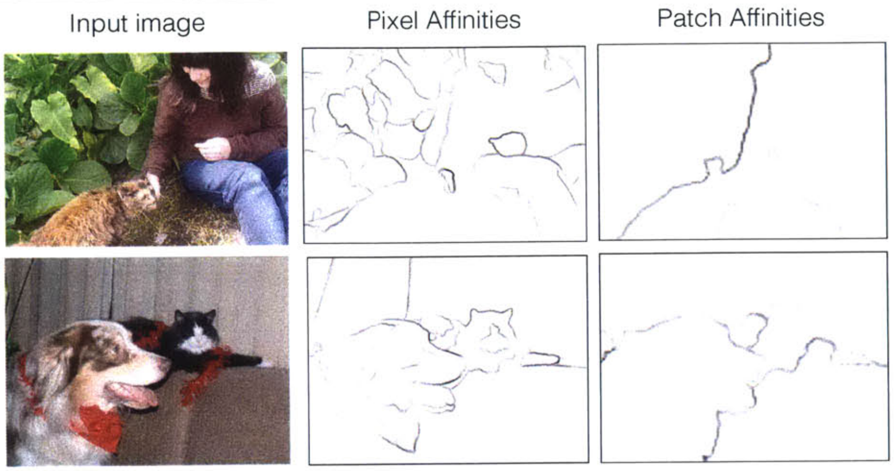


Figure 4.1: Boundary detection using pixel affinities (Chapter 2) versus patch affinities (Chapter 3). Notice that the pixel affinities result in a finer-grained segmentation than the patch affinities.

really want, since adjacent patches usually are semantically the same and non-adjacent patches are usually semantically different.

It is interesting that these two stories boil down to modeling almost the same measure, variants of PMI, but there are certainly big differences between the approaches. Here I discuss some of the similarities and differences, and questions that remain for future research.

■ 4.1 Pixels versus patches

Chapters 2 and 3 both present segmentation algorithms, the former using affinities at a pixel level and the latter at a patch level. Figure 4.1 provides a visual comparison of the two methods. Qualitatively, the pixel affinities produce more precisely localized boundaries, but somewhat over-segment the image into mid-level regions, like textures and object parts. The patch affinities, on the other hand, result in a coarser segmentation that may be more appropriate for finding entire objects. The former works well at matching the partitioning given by humans in the Berkeley Segmentation Dataset

[77]. The later is effective at the problem of object proposals, where the goal is to put bounding boxes around entire objects. Quantitatively comparing the two methods on the same tasks – pixel affinities on object proposals and patch affinities on image segmentation – will be an important step for future research.

■ 4.2 The effect of scale

Central to the difference between the pixel-level results and the patch-level results is the issue of scale. Recall the image of a coral reef in Figure 2.1. Subjects did not draw boundaries around each and every limb of that coral. The subjects who made these and other drawings in the Berkeley Segmentation Dataset were instructed to divide each image into something between 2 and 20 pieces. At the scale of the coral image, that meant outlining entire corals rather than individual pieces. But if we were to zoom in to just a single limb of coral, it is likely a subject would then draw a boundary around it, so as to fill the quota of dividing the image into at least two pieces. If we instead were to zoom out, revealing many more corals in the reef, it is possible subjects would have considered the whole collection of corals to be a single segment. Clearly, the scale of structure relative to the size of the image matters.

Interestingly, our models using PMI are also dependent on scale, and this may explain some of their success at matching the Berkeley data and other human percepts. For example, in Chapter 2, we defined a joint distribution over colors that appear "adjacent" to one another. The specific notion of adjacency was a few pixel radius, defined in Equation 2.1. This radius defines a scale of interest. We are looking for events that occur together in a tiny window of the image. Such co-occurrences capture processes that look like texture to an human viewing the whole image.

More generally, PMI can be understood as comparing the probability of a joint observation under one context – we can call it the *local context* – to its probability under some other broader context – a *global context* (note that in Chapter 3, we made local context explicit with the random variable C). In Chapter 2, we were measuring how much more often two colors occur within a few pixels of each other in an image to how often they occur within image as a whole. In Chapter 3, we were comparing how much more often two patches occur within a few *dozen* pixels of each other to how often they occur within a large dataset of images.

As shown in Figure 4.1, the scale of the two contexts being compared affects the results. Regularities like 'face' is near 'hand' may only appear in the affinities if the scale of the local context is quite large (and likely this is not a regularity even our patch-level method picks up on). Distinguishing between one leaf and another leaf, on the other hand, may require that the local context is very small so that most adjacent measurements are on just a single leaf. This can help explain why boundaries between leaves are discovered in Figure 4.1 using pixel affinities but not using patch affinities.

The types of structures discovered will depend on the scale of the local context and the scale of the global context. To find an association between two events with PMI, we require that those events co-occur more often in the local context than in the global context. Studying the role of each context, and designing better context functions, will be an exciting direction for future research.

■ 4.3 Internal versus external statistics

The two chapters presented two ways of aggregating statistics for PMI, the first using internal statistics and the second using external statistics. The internal option is to learn a stimulus-specific PMI function based on the statistics of co-occurrences in just the single test stimulus on which that function will be applied. The external option is to learn a stimulus-generic PMI function based on the statistics of co-occurrences averaged over an entire dataset of stimuli.

In Chapter 2, I argued that it helps to use internal statistics. Chapter 3 may appear to contradict this point; we instead learned associations based on *external* statistics, i.e. we learned a single PMI function aggregated over a dataset rather than learning a separate PMI function per image.

The issue becomes clearer when we examine why external statistics did not work in Chapter 2. Consider again the zebra. Based on internal statistics we were able to learn to group black-and-white pixels. Such a grouping rule would never be found on the basis of external statistics. This is because in most images dark and light pixels belong to separate objects, and will usually not be found next to each other. A generic color grouping rule, learned from external statistics, will learn to only group alike colors (see Figure 2.6).

External statistics were ineffective at the pixel scale because the external world is so complex that color co-occurrence statistics cannot capture all of it – there is no generic rule for whether or not black and white pixels go together. We need a very small world indeed, e.g., a single image, to make pixel-level statistics able to capture all that is going on in that small world. But as we move toward bigger and bigger patches, each patch conveys more and more information. A model over patches has greater capacity than a model over pixels. Big enough patches should be able to model associations that are general to an entire dataset rather than just to a single image. For example, if we learn that stripe patches are associated with other stripe patches, that grouping rule will apply well regardless of the specific image we apply it to. This is fortunate since it becomes hard to use internal statistics for big patches – there may not enough independent big patches within a single image for us to learn from them, and learning a high-dimensional patch model per image would be exceedingly slow.

Future work should explore combining external and internal statistics. For example, externally-learned priors could be combined with local adaptation based on internal statistics.

4.4 Generative versus discriminative models of association

In Chapter 2, we used a generative model of PMI while in Chapter 3 we switched to a discriminative model. The generative model is only tractable over low-dimensional features, like color. However, when tractable, the generative approach may be preferable. This is because modeling the full joint distribution gives us more flexibility over choosing different information and dependence measures. In particular, in Chapter 2, we were able to add an exponent ρ to the numerator of the PMI equation (Equation 2.2), and we saw that this was important for achieving maximum performance (Section 2.7.3). It is unclear how to set ρ to have a value other than 1 using the discriminative approach.

We therefore see a several important differences between the two halves of this thesis. At the pixel-level, internal statistics work well whereas external statistics fail. At the patch level (and beyond) associations can be effectively modeled with external statistics. An advantage of using pixels and internal statistics is crispness, since we work at a very fine scale. An advantage of using patches and external stats is that we do not have to relearn the model for each image we come across. In addition, high-level associations, like 'head with body', likely only exist at a patch level and beyond, and such associations may only be learnable across a large dataset where many repetitions of the association can be observed.

Combining the best of both methods should be a fruitful direction for future work. The human brain has the ability to see structure and groupings at multiple levels of the visual hierarchy. A bunch of pixels may be seen as a texture, a bunch of textures may become an object, a bunch of objects form a scene. A hierarchical model of associations, that operates on multiple scales, might be able to capture this richness.

We are left with perhaps more questions than when we started. I focused on perceptual groups learned from visual co-occurrences. But groups are hardly the only kind of structure our minds represent, and spacetime co-occurrence is not the only signal from which we can learn. We also see hierarchies and continua, analogies and causal chains, among much else. Do these other kinds of representations also reflect simple structure in the environment? The questions that motivate this thesis remain largely, and excitingly, unanswered: why do we have the mental representations we have, and how did we learn them in the first place?

Bibliography

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. *arXiv preprint arXiv:1505.01596*, 2015.

[2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *PAMI*, 2012.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[4] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.161. URL `http://dx.doi.org/10.1109/TPAMI.2010.161`.

[5] Pablo Arbeláez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3378–3385. IEEE, 2012.

[6] Wilma A. Bainbridge, Phillip Isola, Idan Blank, and Aude Oliva. Establishing a database for studying human face photograph memorability. In *Proceedings of the Annual Conference of the Cognitive Sciences Society*, 2012.

[7] Wilma A. Bainbridge, Phillip Isola, and Aude Oliva. The intrinsic memorability of face images. *Journal of Experimental Psychology: General*, 142(4):1323–1334, 2013.

[8] Horace Barlow. Cerebral cortex as model builder. *Models of the visual cortex*, pages 37–46, 1985.

[9] Jonathan Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. Technical report, Berkeley Tech Report, 2013.

[10] Samy Bengio, Jeff Dean, Dumitru Erhan, Eugene Ie, Quoc Le, Andrew Rabinovich, Jon Shlens, and Yoram Singer. Using web co-occurrence statistics for improving image categorization. *arXiv preprint arXiv:1312.5697*, 2013.

[11] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[12] Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2013)*, 2013.

[13] Zoya Bylinskii, Phillip Isola, Constance Bainbridge, Antonio Torralba, and Aude Oliva. Intrinsic and extrinsic effects on image memorability. *Vision Research*, 2015. in press.

[14] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[15] Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, pages 789–797, 2008.

[16] Liang-Hua Chen, Yu-Chun Lai, and Hong-Yuan Mark Liao. Movie scene segmentation using background information. *Pattern Recognition*, 41(3):1056–1065, 2008.

[17] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.

[18] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[19] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[20] Radoslaw M. Cichy, Dimitrios Pantazis, and Aude Oliva. Resolving human object recognition in space and time. *Nature Neuroscience*, 17(3):455–462, 2014.

[21] Forrester Cole, Phillip Isola, William T Freeman, Frédo Durand, and Edward H Adelson. Shapecollage: Occlusion-aware, example-based shape interpretation. In *ECCV*. 2012.

[22] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[23] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.

[24] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods.* Cambridge university press, 2000.

[25] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[26] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Context as supervisory signal: Discovering objects with predictable context. In *ECCV*. 2014.

[27] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015. URL `http://arxiv.org/abs/1505.05192`.

[28] P Dollár and C.L. Zitnick. Structured Forests for Fast Edge Detection. *ICCV*, 2013.

[29] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[30] Piotr Dollár, Zhuowen Tu, and Serge Belongie. Supervised learning of edges and object boundaries. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1964–1971. IEEE, 2006.

[31] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[32] James H Elder and Steven W Zucker. Computing contour closure. In *ECCV'96*, pages 399–412. Springer, 1996.

[33] Vassiliy A Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.

[34] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88 (2):303–338, 2010.

[35] Alon Faktor and Michal Irani. Clustering by composition–unsupervised discovery of image categories. In *ECCV*. 2012.

[36] Alon Faktor and Michal Irani. Co-segmentation by composition. In *ICCV*, 2013.

[37] Robert M Fano. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29:793–794, 1961.

[38] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[39] József Fiser and Richard N Aslin. Unsupervised statistical learning of higher-order spatial structures from visual scenes. *Psychological science*, 12(6):499–504, 2001.

[40] William T Freeman. The generic viewpoint assumption in a framework for visual perception. *Nature*, 368(6471):542–545, 1994.

[41] Nelson Goodman. Seven strictures on similarity. 1972.

[42] Thomas L Griffiths and Joshua B Tenenbaum. From mere coincidences to meaningful discoveries. *Cognition*, 103(2):180–226, 2007.

[43] Xiaodi Hou, Alan Yuille, and Christof Koch. Boundary Detection Benchmarking: Beyond F-Measures. *CVPR*, 2013.

[44] Alex Ihler and Mike Mandel. http://www.ics.uci.edu/ ihler/code/kde.html.

[45] Nathan Intrator and Leon N Cooper. Objective function formulation of the bcm theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks*, 5(1):3–17, 1992.

[46] Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *ICCV*, 2013.

[47] Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva. Understanding the intrinsic memorability of images. In *Advances in Neural Information Processing Systems*, 2011.

[48] Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What makes an image memorable? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–152, 2011.

[49] Phillip Isola, Jianxiong Xiao, Devi Parikh, Antonio Torralba, and Aude Oliva. What makes a photograph memorable? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1469–1482, 2014.

[50] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H. Adelson. Crisp boundary detection using pointwise mutual information. In *ECCV*, 2014.

[51] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1383–1391, 2015.

[52] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H. Adelson. Learning visual groups from statistical association in space and time. *Under review*, 2015.

[53] William James. *The principles of psychology*. 1890.

[54] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[55] Mamta Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 923–930. IEEE, 2013.

[56] Yan Karklin and Michael S Lewicki. Learning higher-order structures in natural images. *Network: Computation in Neural Systems*, 14(3):483–499, 2003.

[57] Charles Kemp and Joshua B Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008.

[58] Aditya Khosla, Wilma A. Bainbridge, Antonio Torralba, and Aude Oliva. Modifying the memorability of face photographs. In *International Conference on Computer Vision (ICCV)*, 2013.

[59] Talia Konkle, Timothy F Brady, George A Alvarez, and Aude Oliva. Conceptual distinctiveness supports detailed visual long-term memory for real-world objects. *Journal of Experimental Psychology: General*, 139(3):558, 2010.

[60] Philipp Krahenbuhl and Vladlen Koltun. Geodesic object proposals. In *ECCV*. 2014.

[61] Phillip Krahnenbuhl and Vladlen Koltun. Learning to propose objects. In *CVPR*, 2015.

[62] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis–connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2, 2008.

[63] Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In *Computer Vision–ECCV 2006*, pages 581–594. Springer, 2006.

[64] Xiaodan Liang, Si Liu, Yunchao Wei, Luoqi Liu, Liang Lin, and Shuicheng Yan. Computational baby learning. *arXiv preprint arXiv:1411.2861*, 2014.

[65] Joseph J Lim, C Lawrence Zitnick, and Piotr Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. *CVPR*, pages 3158–3165, 2013.

[66] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009.

[67] David Lowe. *Perceptual organization and visual recognition*, volume 5. Springer Science & Business Media, 2012.

[68] Kyle Mahowald, Phillip Isola, Evelina Fedorenko, Edward Gibson, and Aude Oliva. What makes a word memorable? In prep.

[69] Michael Maire and Stella X Yu. Progressive Multigrid Eigensolvers for Multiscale Spectral Segmentation. *ICCV*, 2013.

[70] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001.

[71] Tomasz Malisiewicz and Alexei A Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007.

[72] Tomasz Malisiewicz and Alexei A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, June 2008.

[73] Matei Mancas and Olivier Le Meur. Memorability of natural scenes: The role of attention. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 196–200. IEEE, 2013.

[74] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013.

[75] Ran Margolin, Ayellet Tal, and Lihi Zelnik-Manor. What makes a patch distinct? In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1139–1146. IEEE, 2013.

[76] Arthur B Markman and Dedre Gentner. Nonintentional similarity processing. *The new unconscious*, pages 107–137, 2005.

[77] D Martin, C. Fowlkes, and D Tal. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV*, 2001.

[78] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):530–549, 2004.

[79] Douglas L Medin, Robert L Goldstone, and Dedre Gentner. Respects for similarity. *Psychological review*, 100(2):254, 1993.

[80] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In Léon Bottou and Michael Littman, editors, *ICML*, pages 737–744, Montreal, June 2009. Omnipress.

[81] Hossein Mobahi, Shankar Rao, Allen Yang, Shankar Sastry, and Yi Ma. Segmentation of natural images by texture and boundary compression. *International Journal of Computer Vision*, 95:86–98, 2011. URL `http://dx.doi.org/10.1007/s11263-011-0444-0`.

[82] Gregory L Murphy and Douglas L Medin. The role of theories in conceptual coherence. *Psychological review*, 92(3):289, 1985.

[83] Andreas Opelt, Axel Pinz, and Andrew Zisserman. A boundary-fragment-model for object detection. In *Computer Vision–ECCV 2006*, pages 575–588. Springer, 2006.

[84] Emanuel Parzen et al. On estimation of a probability density function and mode. *Annals of mathematical statistics*, 33(3):1065–1076, 1962.

[85] Judith MS Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970.

[86] Daniel Reisberg. *Cognition. Exploring the Science of the Mind. Third media edition.* New York: Norton, 2007.

[87] Xiaofeng Ren and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. In *NIPS*, 2012.

[88] Xiaofeng Ren, Charless C Fowlkes, and Jitendra Malik. Scale-invariant contour completion using conditional random fields. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1214–1221. IEEE, 2005.

[89] Lawrence Gilman Roberts. *Machine Perception of Three-Dimensional Solids.* PhD thesis, Massachusetts Institute of Technology, 1963.

[90] Irvin Rock. The logic of perception. 1983.

[91] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.

[92] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[93] Bryan Russell, Alyosha Efros, Josef Sivic, Bill Freeman, and Andrew Zisserman. Segmenting scenes by matching image composites. In *NIPS*, 2009.

[94] Jenny R Saffran, Richard N Aslin, and Elissa L Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, 1996.

[95] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[96] Terry Sejnowski and Tobi Delbruck. The language of the brain. *Scientific American*, 307(4):54–59, 2012.

[97] Amnon Sha'ashua and Shimon Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. *ICCV*, 1988.

[98] Roger N Shepard. Multidimensional scaling, tree-fitting, and clustering. *Science*, 210(4468):390–398, 1980.

[99] Roger N Shepard. Toward a universal law of generalization for psychological science. *Science*, 237(4820):1317–1323, 1987.

[100] Jianbo Shi and J Malik. Normalized cuts and image segmentation. *PAMI*, 22(8): 888–905, 2000.

[101] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[102] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Contour-based learning for object detection. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 503–510. IEEE, 2005.

[103] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM Transaction of Graphics (TOG) (Proceedings of ACM SIGGRAPH ASIA)*, 30(6), 2011.

[104] Saurabh Singh, Abhinav Gupta, and Alexei Efros. Unsupervised discovery of mid-level discriminative patches. *Computer Vision–ECCV 2012*, pages 73–86, 2012.

[105] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing, 1968.

[106] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[107] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.

[108] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. *International Conference on Machine Learning (ICML)*, 2011.

[109] Jay M Tenenbaum and AP Witkin. On the role of structure in vision. *Human and machine vision*, pages 481–543, 1983.

[110] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[111] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022): 1279–1285, 2011.

[112] D. Todorovic. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008. revision 91314.

[113] Amos Tversky. Features of similarity. *Psychological review*, 84(4):327–352, 1977.

[114] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[115] Hermann Von Helmholtz. *Handbuch der physiologischen Optik*, volume 9. Voss, 1867.

[116] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[117] Catherine Wah, Subhransu Maji, and Serge Belongie. Learning localized perceptual similarity metrics for interactive categorization. *Human-Machine Communication for Visual Recognition and Search (HMCV) Workshop*, 2014.

[118] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. *arXiv preprint arXiv:1505.00687*, 2015.

[119] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[120] Andrew P Wilkin and Jay M Tenenbaum. What is perceptual organization for? *From Pixels to Predicates*, 1985.

[121] Ren Xiaofeng and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. *NIPS*, pages 593–601, 2012.

[122] Yun Zhai and Mubarak Shah. Video scene segmentation using markov chain monte carlo. *Multimedia, IEEE Transactions on*, 8(4):686–697, 2006.

[123] B. Zhou, Liu. Liu, A. Oliva, and A. Torralba. Recognizing City Identity via Attribute Analysis of Geo-tagged Images. *ECCV*, 2014.

[124] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*. 2014.

[125] M Zontak and M Irani. Internal Statistics of a Single Natural Image. *CVPR*, 2011.

[126] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011.

[127] Daniel Zoran and Yair Weiss. Natural images, gaussian mixtures and dead leaves. *NIPS*, 2012.