

Learning and Applying Model-Based Visual Context

by

Vikash Gilja

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

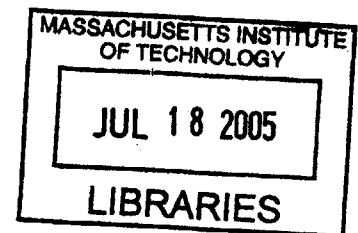
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2004 [September 2004]

Copyright 2004 Vikash Gilja. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis and to
grant others the right to do so.



Author ...

Department of Electrical Engineering and Computer Science

August 6, 2004

Certified by.....

Patrick Henry Winston

Ford Professor of Artificial Intelligence and Computer Science

Thesis Supervisor

Accepted by ..

Arthur C. Smith

Chairman, Department Committee on Graduate Theses

BARKER

Learning and Applying Model-Based Visual Context

by

Vikash Gilja

Submitted to the Department of Electrical Engineering and Computer Science
on August 6, 2004, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

I believe that context's ability to reduce the ambiguity of an input signal makes it a vital constraint for understanding the real world. I specifically examine the role of context in vision and how a model-based approach can aid visual search and recognition. Through the implementation of a system capable of learning visual context models from an image database, I demonstrate the utility of the model-based approach. The system is capable of learning models for "water-horizon scenes" and "suburban street scenes" from a database of 745 images.

Thesis Supervisor: Patrick Henry Winston

Title: Ford Professor of Artificial Intelligence and Computer Science

Acknowledgments

I extend a thank you to everyone who has contributed patience and guidance along the way. Professor Winston, thank you for all of your assistance and encouragement; of the many things I will take from MIT, your heuristics will be among the most treasured. Mom, Dad, and Dhedhe this page cannot hold all the contributions you have made to who I am; spirit is not meant to be contained. My friends, you inspire me to do great things with my life; if I act on only a small fraction of this inspiration, my life will never cease to be exciting.

I dedicate this thesis to the swiftness of time; if time ever disappears slower than goals, the aim is too low.

Contents

1	Introduction	11
1.1	Overview	11
1.2	Motivations	12
1.3	Preview	14
2	Visual Context	17
2.1	Definition	17
2.2	Applications	18
2.3	Existing Methods	19
2.3.1	Histograms	19
2.3.2	Qualitative Models	20
2.3.3	Natural Image Statistics	21
2.3.4	Query by Content	23
3	Model-Based Approach	25
3.1	Models Explained	25
3.1.1	Image Sub-Regions	28
3.1.2	Quantitative Measures	28
3.2	Applications	30
3.2.1	Search	30
3.2.2	Recognition	31
4	Architecture and Implementation	33

4.1	Generating the Blobverse	33
4.2	Matching and Clustering the Blobverse	35
4.2.1	Measuring Image Similarity	35
4.2.2	Matching Heuristics	37
4.2.3	Clustering	39
4.3	Learning Blobverse Models	40
5	Discussion	47
5.1	The Best of Both Worlds	47
5.2	How to make it work	50
6	Contributions	51

List of Figures

1-1	What is under the oval? Context provides enough constraint for us to make a reasonable guess.	12
1-2	Where are the monitors? The context of the scene can guide the visual search.	13
2-1	(A) was generated by randomly permuting the rows and columns of (B). (B) and (C) have inverted colors. Thus, (A) and (B) have identical histograms, but (B) and (C) have very different histograms.	20
2-2	Example of qualitative relationships.	21
2-3	The variance of the first 200 principal components of image structure space. The dashed line indicates the 25th principal component.	22
2-4	Starting with the image to the left, the images with the four closest and four farthest image statistics based contexts were retrieved.	23
3-1	Lipson's Snowy Mountain template consists of 3 parts: A for the sky, B for the snow, and C for the mountain. To match the template, an image must contain 3 groupings of pixels which match the template for pairwise qualitative relationships and quantitative values.	26
3-2	The images to the left and the right have similar contexts. If a template is generated with the correct set of qualitative relationships, it can tolerate both images.	27
3-3	If an image is an example of a common context, we can use our prior experiences with that context to speed up a visual search.	30

3-4	If an image is an example of a common context, we can use our prior experiences with that context to speed up a visual search.	31
4-1	Segmentation examples	34
4-2	Clustering with connected components	39
4-3	Clustering Example 1: The cluster is formed around the image with the red frame. Images were added to the cluster from left-to-right, top-to-bottom order. $k = 30$ and $\Theta = 0.66$	41
4-4	Clustering Example 2: The cluster is formed around the image with the red frame. Images were added to the cluster from left-to-right, top-to-bottom order. $k = 30$ and $\Theta = 0.66$	42
4-5	Model Learning Example 1: The following images match the model learned from cluster example 1. $\alpha = 0.75$	44
4-6	Clustering Example 2: The cluster is formed around the image with the red frame. Images were added to the cluster from left-to-right, top-to-bottom order. $\alpha = 0.9$	45
5-1	Complex classifiers may be necessary to learn from vector-based contexts	49

List of Tables

4.1	Feature Matching Techniques	37
4.2	Number of Blobs vs. Number of Correspondences	38

Chapter 1

Introduction

1.1 Overview

I believe that context is an important constraint for input processing; context can effectively reduce the ambiguity of an input signal. For example, if vision knows that it is looking at a kitchen, object recognition can reduce its search space from all-objects to kitchen-objects. Similarly, visual search can focus in on areas of a kitchen where specific items can be found, like counter-tops. In order to use context, a method for the extraction of context must be developed. Initial mapping of context should be fast, in order to quickly simplify the input processing problem. As input undergoes deeper processing, context can be refined. A partial understanding of the meaning can greatly simplify further processing of the input. Thus, I approach the problem of context by examining how a semantically rich context representation can be applied to visual scenes.

I have chosen to focus on the concept of visual context. As a first step, each image in a database of 750 images is segmented into 2-10 areas of roughly constant texture. This first step results in a huge information reduction, which facilitates comparison and classification. Matching methods are applied in order to cluster images with common contexts, such as “stadium scene” or “street scene.” Given a cluster of images sharing a context, a basic model of the context can be learned by examining the consistencies between image segments. This context model can be applied to

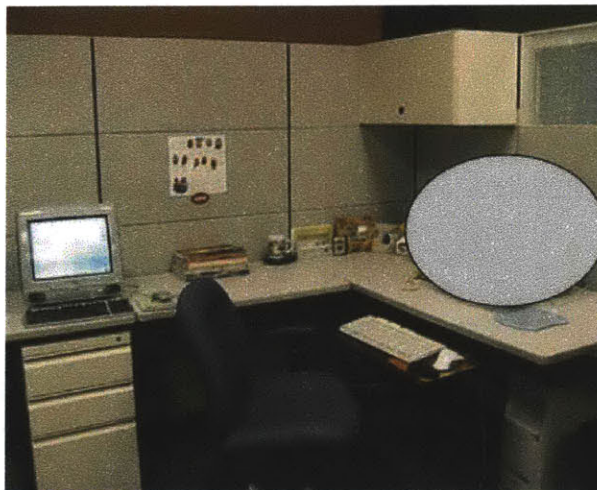


Figure 1-1: What is under the oval? Context provides enough constraint for us to make a reasonable guess.

classify new images. The implementation described in this thesis has learned to identify “water-horizon scenes” and “suburban street scenes.”

The visual context models proposed in this thesis are based upon pair-wise comparisons between areas of constant texture in a visual scene. These comparisons are qualitative in nature and can match a scene from different viewpoints or with a variable configuration. Existing models of context are dependent upon the physical structure of a scene.

I have contributed a motivation for visual context and have developed a representation for scene context. Through this exploration, I demonstrate methods by which images can be clustered and visual contexts can be learned. I also contribute methods by which semantics can be used to constrain visual search and object recognition.

1.2 Motivations

Visual context can be used to simplify the tasks of object recognition and visual search. Due to the regularity of the visual world, a visual context provides powerful constraints for both of these tasks. In the cubicle scene of figure 1-1, a region of the image is blocked out by a gray oval. However, the context around the oval makes it clear that there is probably a computer monitor under the oval.

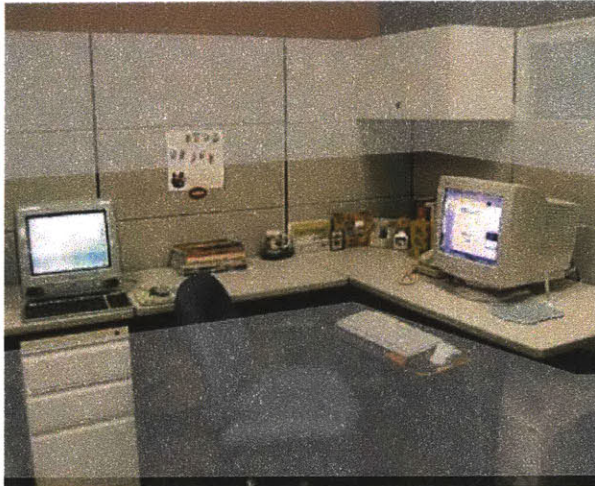


Figure 1-2: Where are the monitors? The context of the scene can guide the visual search.

Similarly, if asked to find computer monitors in the scene, it would be reasonable to start the visual search in a restricted area of the scene (figure 1-2). Because we are in an “office” context, the search space can be restricted to a specific region of the scene. Furthermore, context dictates it is highly likely that monitors are present with a specific orientation and scale in this portion of the scene. Such knowledge can reduce the computation necessary for matching.

Thus, context can reduce the amount of work necessary for visual search and object recognition by reducing their respective search spaces. Contextually primed visual search can focus on limited regions within an image, resulting in faster searches. Contextual priming can reduce the number of objects that object recognition needs to match against from all known objects to those that are consistent with the given context.

Work in the field of visual cognition provides evidence that context can provide assistance to both visual search and object recognition in humans. In a series of visual search studies by Chun and Jiang [2], consistent global scene context was found to greatly improve performance in visual search tasks. Subjects were presented with novel stimuli and were instructed to find targets within these stimuli. After training, targets presented in stimuli with consistent visual contexts were found more quickly.

Visual cognition studies have also provided evidence that context influences ob-

ject recognition performance. These studies involve two sets of stimuli: objects in consistent scenes and objects in inconsistent scenes. For example, a fire hydrant on a sidewalk is consistent, but a fire hydrant in a kitchen is inconsistent. The results suggest that humans are faster and more accurate at recognizing consistent stimuli [3].

In addition, Olivia and Schyns [5] have found that humans can accurately provide scene labels, like “kitchen” or “office”, within 45-135ms of image presentation. This labeling is relatively quick, considering that object detection generally takes about 150ms [7]. Thus, visual cognition provides evidence that visual contexts are vital for efficient visual search and object recognition and that it is possible to extract contexts quickly.

In addition to aiding visual search and object recognition, an effective visual context generator can be used to search and organize large databases of images. A visual context generator provides a semantic tag for an image based upon its content. Thus, if we want to find similar images, we simply need to search for images with similar tags. Most existing image database search systems rely on human entered tags or match on relatively simple image properties, like image color histograms. Human tagging requires a great deal of startup cost for large databases and is reliant on consistency between human tags. Histogram methods do not provide very useful matches, since they ignore the semantic content of an image. The context generator could provide the best of both worlds, a machine-based labeler that organizes a database by image content.

1.3 Preview

Chapter 1 provides an overview of the thesis and a motivation for studying visual context.

Chapter 2 defines visual context as it will be addressed throughout this thesis. The chapter follows with a discussion of possible applications for context and coverage of existing methods that have inspired this thesis.

Chapter 3 explains the model-based approach and offers some insights into possible implementations. The applications discussed in chapter 2 are expanded upon.

Chapter 4 describes a specific implementation of the model based approach. The implementation demonstrates how models of visual context can be learned from a database of images.

Chapter 5 is a discussion on the implementation described in chapter 4. This chapter focuses on the positive and negative aspects of the model-based approach and describes possible expansions and new directions.

The contributions of this thesis are outlined in chapter 6.

Chapter 2

Visual Context

2.1 Definition

Visual context can be defined in many ways. At the most basic level, a visual context is a subspace of all possible images. One can define such a subspace in any manner; two examples are all images of cities and all images that do not contain the color blue. The definition of a context is intrinsically linked to its utility.

Knowing that an image does not contain the color blue offers very little assistance in scene understanding. Perhaps, such knowledge can allow us to assume that certain objects, like a blue bird, cannot be in the scene. However, the context does not offer us any clues about the visual scene's structure and thus offers us few clues that could be used to speed up processing.

However, if we know that we are looking at a city scene, we can assume certain regularities. There are common large markers in a city scene, like streets and buildings. It is common to find people and cars on the streets and windows and awnings on the buildings. Cities scenes share common structure and content; these redundancies may allow us to apply heuristics that simplify processing of a scene.

I believe that semantically rich visual contexts are vital to scene understanding and learning. As discussed in the next section, such contexts can aid other modalities. Thus, in this thesis I focus on processes by which semantically rich visual contexts can be learned and applied to images. When I refer to visual context, you may assume

that the context is meant to link image to semantics.

Visual context can be refined by deeper processing, such as recognizing all of the images in the scene. As outlined in the previous chapter, however, visual context may be necessary to constrain such deeper processing steps. Thus, initial estimations of visual context must be made quickly, without the aid of object recognition.

2.2 Applications

A system for learning and/or extracting semantically linked visual context has many applications. These applications range from sorting and searching image databases to more complex image understanding to multi-modal perception. Context is a powerful information-processing tool that can be engineered to solve complex machine vision/perception problems and may be used by biological systems.

A system that identifies the visual context(s) of a scene can be used to search a database of images. If we query the database with an image that has a “city-scene-context,” the system can check if any other images in the database have a “city-scene-context.” By caching each image’s context labels, searches can be executed quickly since matches can be made with minimal image processing. In a biological system, such a system can be employed for localization. An organism could use visual context to quickly determine if it has been in a similar place or situation before. Speed is important, as a few milliseconds of hesitation may mean the difference between escaping a predator and becoming its lunch.

As mentioned previously, context can provide clues that greatly speed up both object recognition and visual search. If contexts are chosen correctly, images with similar contexts may share similar compositional structures. In a city scene, streets are below the skyline and people are often found on streets. People on the streets are generally upright. Additionally, the scale of a person on the street is proportional to the scale of the street. Thus, a visual search for streets or people can be constrained to a limited region of the scene and object recognition can be constrained to a specific class (such as people), orientation (upright), scale (proportional to street). In this

manner, context can provide goals for vision. For example, for a given context there may be areas of the scene that should be systematically analyzed for important details, such as faces or signs.

Visual context can provide constraints for other input modalities. For example, when you are in a city you are more likely to hear certain words, like “street,” “traffic,” and “building.” Being in a city increases the chances of hearing sounds originating from vehicles. Thus, if vision alerts us that we are in the city, constraints can be placed upon language and auditory processing. If an utterance or sound source is ambiguous, it would make sense to try to find the closest “city word” or “city sound” match.

2.3 Existing Methods

The image database search problem is closely related to visual context. In both scenarios, the goal is to provide a relatively quick approach for finding images that are similar in respect to some metric. The metric, like the definition of visual context, must be selected carefully so that it addresses the problem at hand. Thus, many methods for image database search exist. In the following section I present the image search and context methods that have influenced my work.

2.3.1 Histograms

Visual context requires a semantically derived image label. Images with similar semantic content should have similar contexts. In order to tease apart the semantic content of an image, we must examine its structure. Organizing images by their color or intensity histograms is not adequate. If we take an image and scramble it, the histogram will stay the same, but it will not have the same context (see figure 2-1) [4].

The way around this problem is to use structure to develop visual context. I will introduce two existing methods that utilize structure, Lipson’s qualitative models and natural image statistics methods.

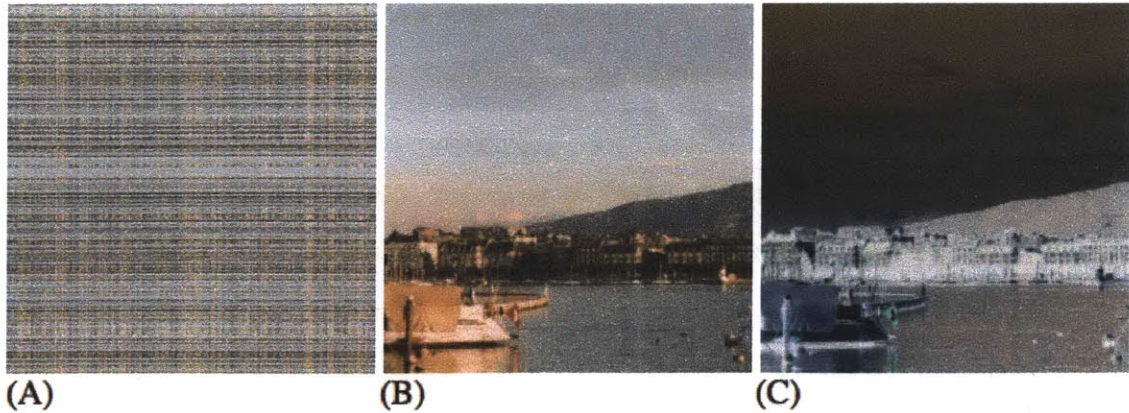


Figure 2-1: (A) was generated by randomly permuting the rows and columns of (B). (B) and (C) have inverted colors. Thus, (A) and (B) have identical histograms, but (B) and (C) have very different histograms.

2.3.2 Qualitative Models

Lipson uses qualitative models to represent and recognize image classes. The qualitative models are formed by generating pair-wise relationships between groups of pixels. An image class is represented as a graph specifying necessary relationships between pixel groups. If the graph for a given image class can be matched against an image, that image is part of that class.

Lipson defines four features that can be used in such a model: size, position, luminance, and color. The qualitative model uses greater than, equal to, and less than, to define the relationship of each of these features between two image patches. Figure 2-2 demonstrates the qualitative representation of a simple synthetic image.

Lipson uses this method to define a “snowy mountain template.” The template takes into account the sky, the snow-cap, and the snow-less mountain base. By using qualitative comparisons between these three elements Lipson can find images of snowy mountains in a database of 700 images with a 75% true-positive and 12% false-positive detection rate.

The advantage of qualitative models is that they can be used to detect an image class across multiple views and lighting conditions. A given scene viewed from different angles often retains the same relative spatial arrangements between objects. The relative luminance and color of two regions of an image do not change if the quality

Square vs. Triangle	
Red	=
Green	>
Blue	<
Size	>
Spatial-X	<
Spatial-Y	=
Luminance	<




Figure 2-2: Example of qualitative relationships.

of lighting across the entire scene is shifted. The disadvantage of this system is that learning new templates or image contexts boils down to graph matching. In a large database of images, even if all the qualitative relationships were precomputed, searching for images that match a given template is quite expensive. However, it is possible to speed up the process by using heuristics to reduce the number of candidates that are matched against.

2.3.3 Natural Image Statistics

Olivia and Torralba [6] develop a notion of visual context based upon natural image statistics. They utilize the statistical regularity of natural images to develop a low-dimensional holistic representation of an image. Images that map close together in this low-dimensional space are presumed to have similar contexts, while images that are far apart are presumed to have different contexts.

Olivia and Torralba assume that structure is key to context. Thus, they use orientation selective filter maps to produce a representation of the local oriented energy in an image. From there, they utilize principal components analysis (PCA) across the orientation selective filter maps of 1000 images to find a low-dimensional set of principal components. These components capture the majority of the variance in orientation selective filter maps. Thus, they correspond to the most informative axis of the orientation selective filter map data and form the “gist” of the structure of the image.

Using these methods and a set of 850 images, I have developed a 25-dimensional

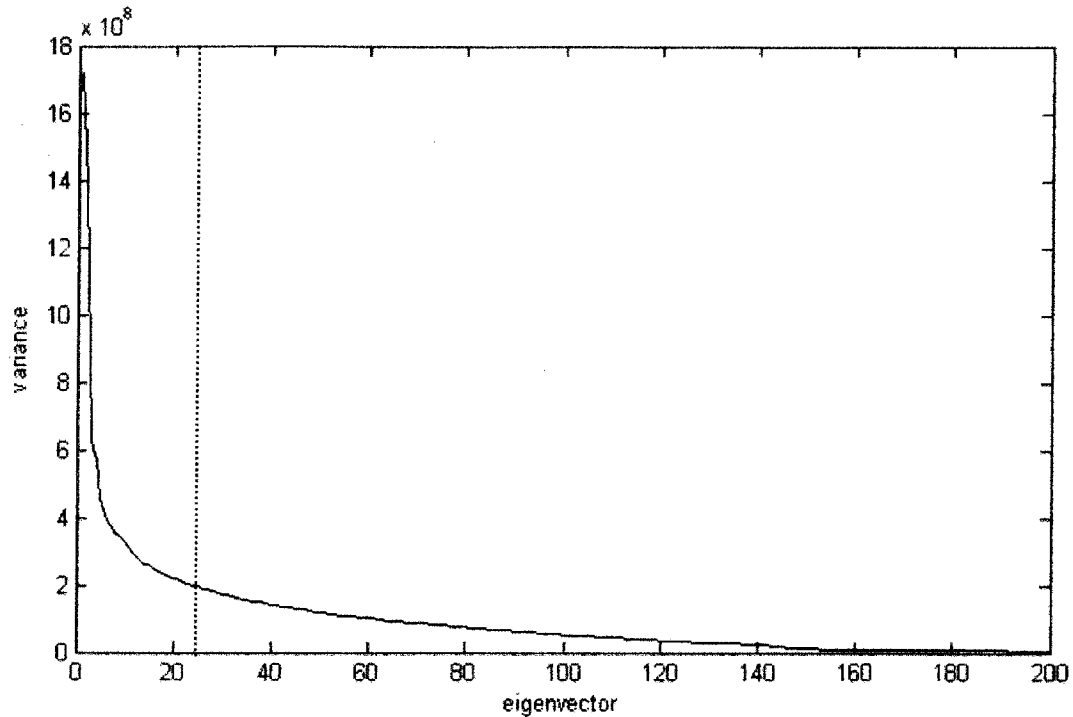


Figure 2-3: The variance of the first 200 principal components of image structure space. The dashed line indicates the 25th principal component.

representation of image context. The original images are 384 by 384 pixels and are mapped into 42 filter maps. Thus, each image represents a point in an approximately 6 million dimensional space ($384 \times 384 \times 42 = 6,193,152$). The first 25 principal components manage to capture over 50% of the variation in image structure (as defined by orientation selective filter maps). The graph in figure 2-3 shows the variance for the first 200 principal components.

This natural statistics based definition of context has quite a few advantages; it is parameter free and is fast. The model is parameter free because it is completely based upon image statistics. In a sense, it is derived from natural regularities. Identification of the principal components is computationally expensive, but once the components are isolated they can be applied very quickly. Mapping an image within the space requires only the computation of the filter maps and a set of cross products. With a database of images, one can search for similar contexts by finding nearest neighbors in the 25-dimensional space. However, the judgments are completely based upon image

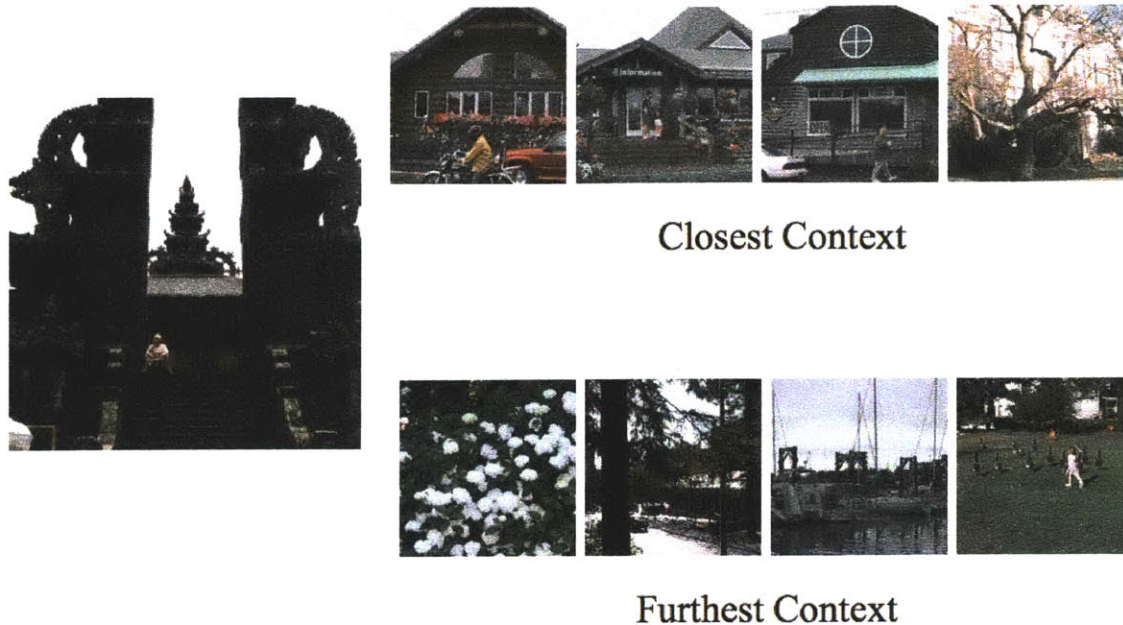


Figure 2-4: Starting with the image to the left, the images with the four closest and four farthest image statistics based contexts were retrieved.

structure. Thus, different views of the scene may map to very distant points in the 25-dimensional space. Also, very different contexts may have very similar structures, for example a pitched roof and a mountain peak have similar spatial structures but are very different scenes. Thus, this method is good for pairing together images that look alike in a structural sense, but may fail to match semantically related images.

2.3.4 Query by Content

The element missing from the natural image statistics approach is a notion of content. If a mountain is declared to be in a similar context to a house, examining the content of the image could repair the context. A conceptually simple approach to content based context is to recognize all of the objects in a scene and then to make a judgment based upon the types of objects in a scene. The mountain scene could be separated from a house because it lacks a door and windows. However, such an approach is impractical, because it requires the use of object recognition, a problem that is hard enough on its own! The Region-Based Image Querying technique offers an alternative.

Carson, Belongie, Greenspan and Malik [1] developed Region-Based Image Querying (RBIQ) as a method for retrieving images in a large database by using content. The method relies on image segmentation and region characterization. The process results in a blobworld representation of an image. Users can select a blob and search for similar blobs in images across the image library.

RBIQ sidesteps the issue of object recognition by using a very rough method of content characterization. Images are segmented by finding regions of roughly consistent texture. Texture consistency provides an estimate of object segmentation; the method is imperfect because neighboring objects with similar textures may be merged into a single segment and objects composed of varied textures will be split into multiple segments. Each segmented region is represented by its color, shape, location, and texture features, characterized by a 8-dimensional region classifier vector. The algorithm has been parameterized to segment approximately 4-6 regions per image.

Matches are made by querying for a specific region type. Thus, in order to make a match a blob is selected and the nearest neighbor blobs in the 8-dimensional region classifier space are found. Thus, the computational complexity of the search process is similar to the natural image statistics approach.

Chapter 3

Model-Based Approach

3.1 Models Explained

The model-based visual context approach described in this thesis is heavily influenced by the work of Lipson [4]. The core idea is that a context can be described by a template. This template describes the configuration and properties of subregions in the image. An image with subregions that match the template is an example of that context.

Lipson manually generated the set of models which she applies to the database. One of her main examples is of a “snowy mountain” template. As shown in figure 3-1, this model is defined by the presence of three regions, A, B, and C. The template defines specific constraints on quantitative values for individual regions and specifies qualitative pairwise relationships between these regions, such as “A is 1x2 pixels”, “B is lighter than A”, and “B” is above “C.”

This model-based approach offers flexibility. A template is resistant to deformations in the image. For example, shifts in configuration can be tolerated. We can move parts of the scene around and as long as the relationships described in the template are preserved, the deformed image is still a member of the context. For figure 3-2, a “building-street-water” context can be generated to accommodate both images. Similarly, a template will be robust to slight changes in viewing angle. If an observer move slightly, most pairwise relationship in the field of view will remain unchanged.

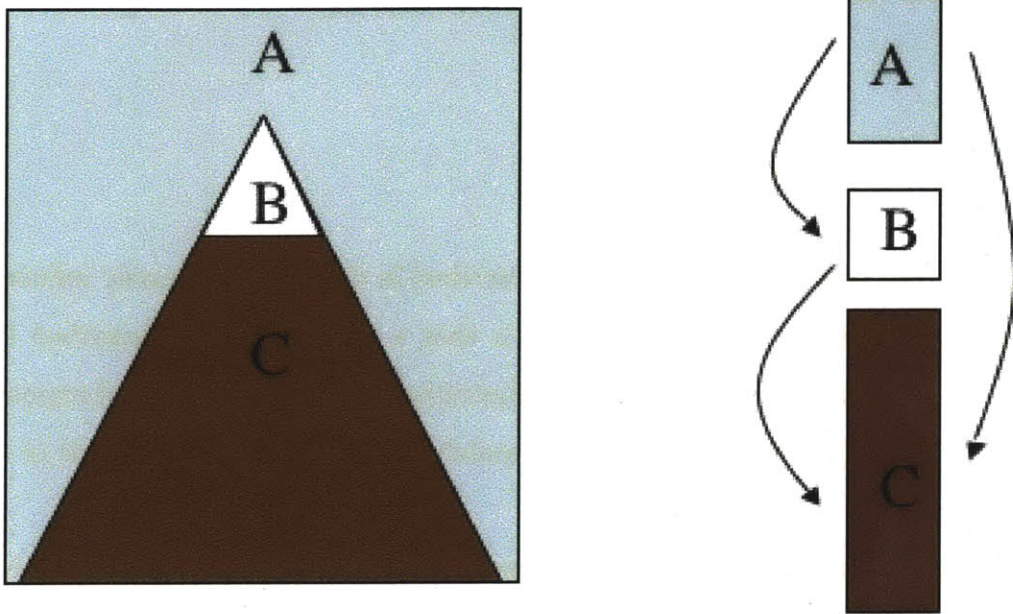


Figure 3-1: Lipson's Snowy Mountain template consists of 3 parts: A for the sky, B for the snow, and C for the mountain. To match the template, an image must contain 3 groupings of pixels which match the template for pairwise qualitative relationships and quantitative values.

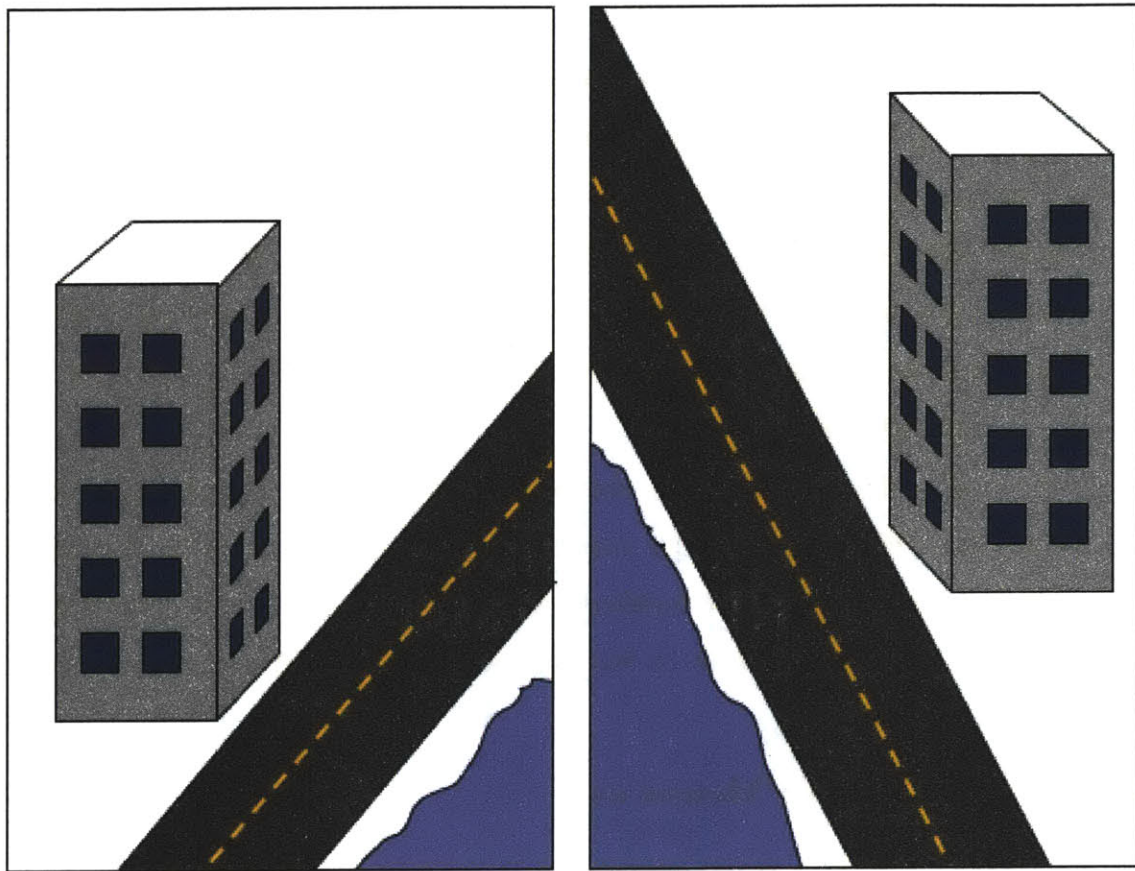


Figure 3-2: The images to the left and the right have similar contexts. If a template is generated with the correct set of qualitative relationships, it can tolerate both images.

3.1.1 Image Sub-Regions

The models of visual context I wish to develop are linked to image semantics. In order to extract the meaning of an image, we could recognize the objects within the image and examine all of the relationships between the objects. Such an approach is impractical, since we would like to use visual context to facilitate object recognition. Additionally, as a scene becomes increasingly cluttered the number of relationships between objects increases exponentially.

A more practical approach is to segment the image into areas of similar content. If we segment an image into areas of constant texture, it is highly likely that the region corresponds to a single object or a collection of similar objects. Additional information, like the presence of edges or occluding contours can be used to refine the segmentation process. Segmentation can be achieved in a completely bottom-up manner, simply relying on image properties. Thus, this step can occur before a context has been identified.

Given additional contextual clues, the segmentation procedure can be refined with a top-down bias. For example, if we expect to be in a city, the segmentor could be calibrated to be more sensitive to buildings and roads. In this manner, visual context can be applied recursively in order to find an increasingly specific context for a scene.

3.1.2 Quantitative Measures

The successful application of any vision system is reliant on the right choice of low-level features. For model-based visual context, we must compare sub-regions of an image. Thus, the features selected will represent a collection of pixels. The following list is not exhaustive, but is a start:

Color: With proper extraction, color is one of the most reliable properties. If color is isolated from intensity, it tends to remain constant under a wide variety of natural lighting conditions. In some applications, either mean or mean and variance may suffice. For more careful matching a color histogram may be necessary.

Intensity: From image to image the intensity of the same object can vary greatly.

Intensity is heavily dependent upon the brightness of a light source and the angle at which light is reflected. Thus, intensity may be more useful for comparing regions within a single image than as a quantitative measure. Like color, intensity can be represented as either a mean, mean and variance, or histogram. If represented as a histogram, the lighting constancy issue can be eliminated. Similar regions will have the same shape histograms under a variety of lighting conditions if the mean and variance are normalized.

Size: The size of a corresponding subregion can vary depending upon the distance at which the observer is standing from the target. Also, viewpoint shifts can cause changes to size. For example, viewing angle can shift the position of the horizon line and change the size of the “sky” in an image. However, size can help distinguish similar sub-regions. In the snow covered mountain example, the relative size of the snow and the mountain are vital. If the relationship was not included, a cloudy sky would be mistaken for snow.

Texture: Texture can be measured using a variety of methods, generally involving filter responses. Measures of texture offer a method for judging spatial patterns in pixel values within a sub-region. Texture can be vital for discriminating between sub-regions with similar color and intensity, like water and sky.

Position: The spatial organization of a set of sub-regions can be captured by comparing relative positioning in both the vertical and horizontal axis. Absolute measures are not that important, as slight changes to configuration or viewpoint can create large shifts in absolute position without altering the context.

Shape: Like texture, shape can be quantified using many different methods. The most simple approach could be a measure of height and width. More complicated methods include shape template matching and boundary fitting.

Distance/Scale: Using range data (from stereo or motion) can be used to tag regions with a z-range distance. These data can be used in a manner that is similar to x-y position.

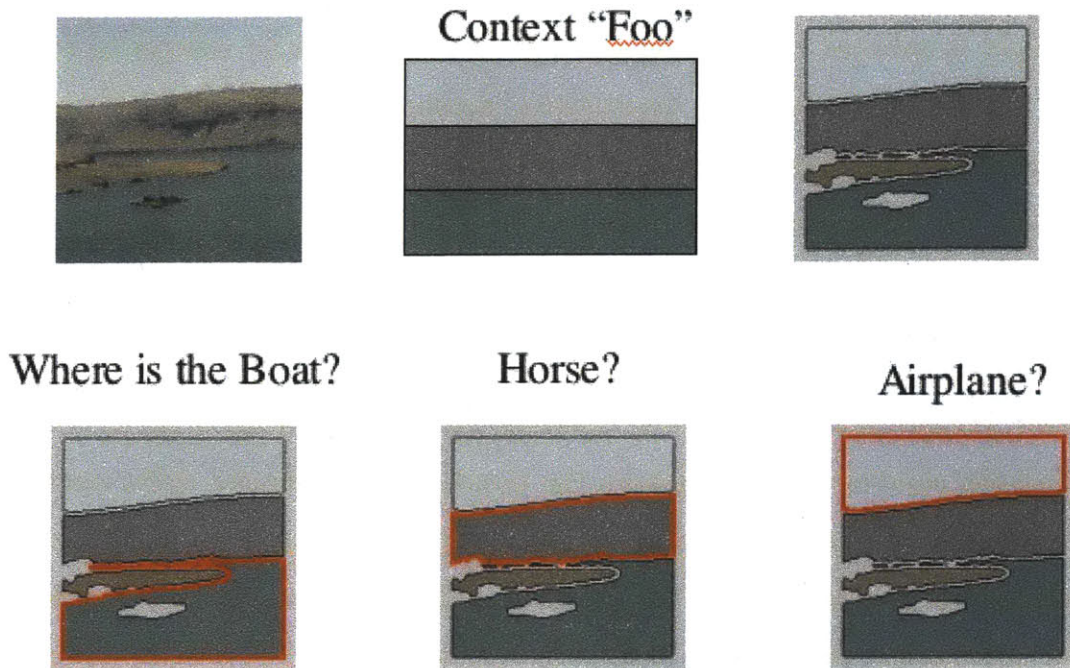


Figure 3-3: If an image is an example of a common context, we can use our prior experiences with that context to speed up a visual search.

3.2 Applications

In the previous chapter, I discussed how any form of visual context can be applied to facilitate visual search and object recognition. By using a model-based approach to visual context, these applications can be further refined. The model-based approach generates specific sub-regions of an image. We can use the sub-regions to simplify each of these tasks.

3.2.1 Search

If an image is associated with a commonly encountered context, we can use the previous experiences with the context to speed up processing of the image. For example, we encounter the image from figure 3-3. It happens to be an example of context "Foo." In previous encounters with context "Foo" we consistently found airplanes in the upper most sub-region, horses in the middle sub-region, and boats in the bottom sub-region. Thus, we have learned that if we want to find any of

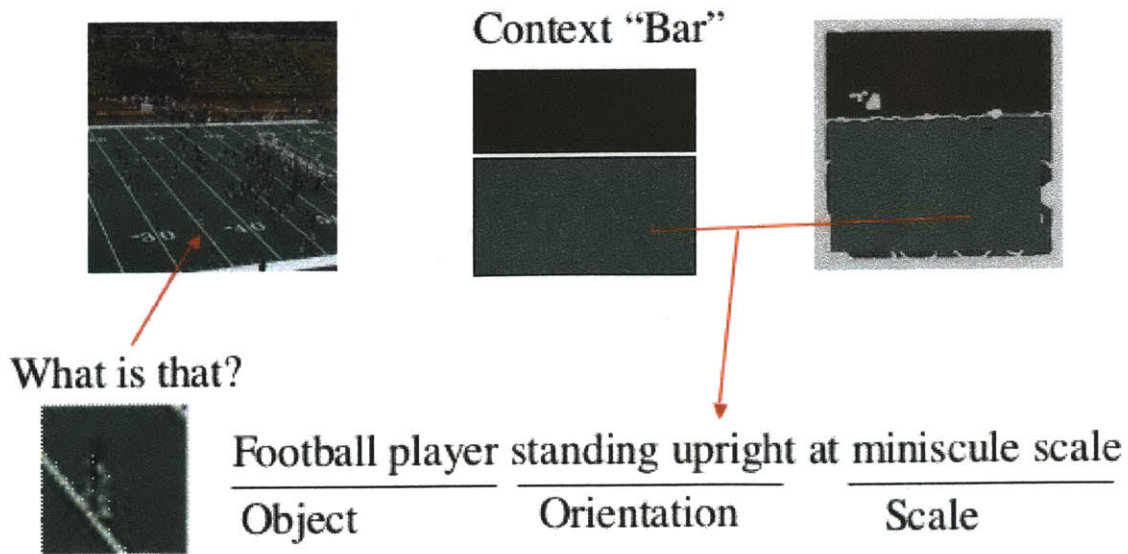


Figure 3-4: If an image is an example of a common context, we can use our prior experiences with that context to speed up a visual search.

these objects in the current image, we can restrict our search to one of these specific sub-regions.

With such a model, training will not require a large data set, because there are only a few sub-regions. A simple training scheme could involve conditioning a probability distribution to determine how likely it is for a given object to appear in a sub-region.

3.2.2 Recognition

Similar gains can be made for object recognition. The presence of an object in a sub-region provides evidence for future encounters. In figure 3-4, we can determine that the squiggly line is a football player because of the context. Prior experience has told us that such a squiggly lines present in the bottom region of the "bar" context are highly likely to be football players that are standing up and are at a really small scale. All of this prior knowledge simplifies the object recognition task. Instead of trying to match the squiggly line against all possible objects that we have ever encountered; we can start off by limiting ourselves to objects that are commonly found on a football field. Additionally, we can limit ourself to the common orientations and scales of

that object when found on a football field. Again, training is simplified because our context model consists of specific sub-regions.

Chapter 4

Architecture and Implementation

I have begun to explore the utility of model-based visual context by designing and implementing a system that allows models to be learned from a database of 745 images. The system attempts to use consistency amongst images to learn models without supervision. The process involves three subsystems, as described in the following sections. The system begins by segmenting images into regions of consistent texture. Then, these images are matched to find clusters of similar images. These clusters are analyzed to find a model that describes the cluster.

4.1 Generating the Blobverse

The “Blobverse” is a simplified version of U.C. Berkeley’s “Blobworld” [1]. I use the same method for image segmentation: expectation-maximization to find regions of consistent color, texture, and intensity (called “blobs”). This approach is appealing because it is entirely bottom-up, requiring no supervision. The blob representation is much more compact than an image consisting of pixel values. If segmentation is successful, then blobs can be tied to semantic components of the scene (such as “sky” or “tree”). The Blobverse uses a less weighty representation for each texture region than the Blobworld. The Blobworld represents each color channel using a 256-bin histogram; the Blobverse uses a single average value. This simplification is made to speedup the matching process. This speedup is important because the Blobverse

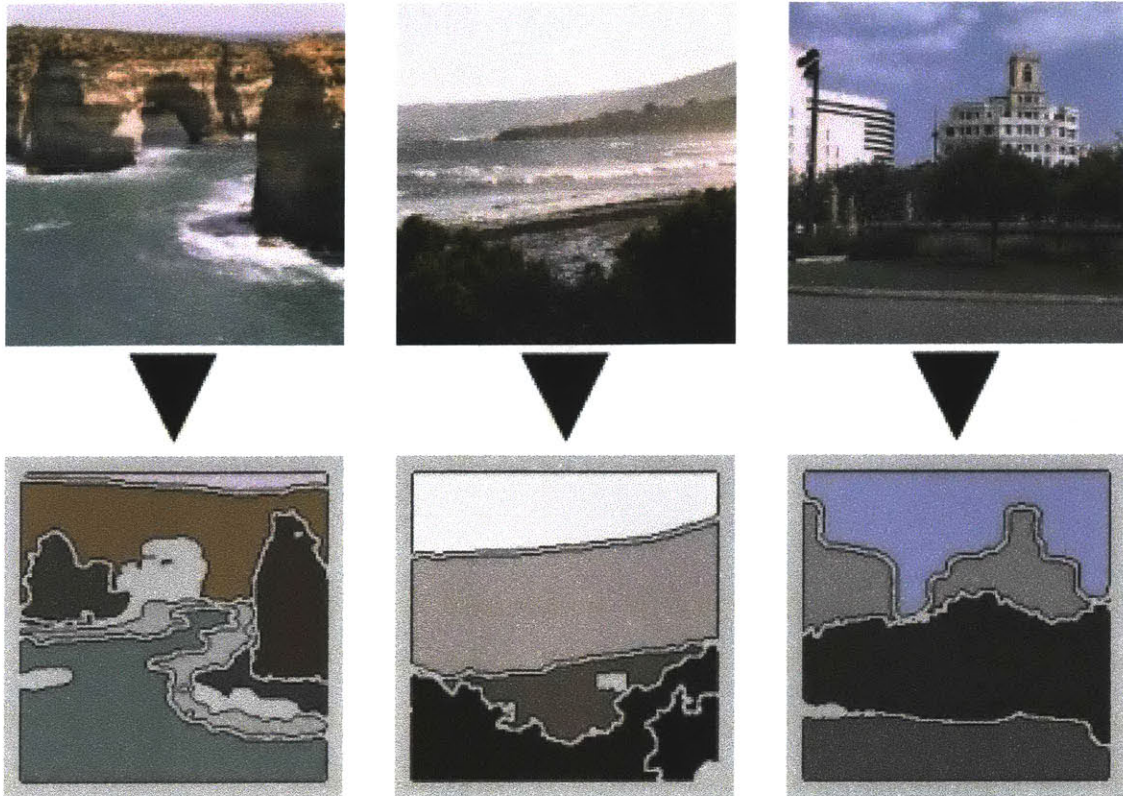


Figure 4-1: Segmentation examples

must test more candidate matches.

In order to carry out expectation-maximization (EM), each pixel in the image is represented by its location, color, and local texture information. EM is run with the assumption of two sources and is rerun with 3,4, and 5 sources. The result that best fits the data (based on a scoring heuristic), is selected. The regions are segmented by finding spatially connected pixels that are from the same source. Across images in the database, this process yields 1 to 10 segments. Figure 4-1 shows the results of segmentation for a few different images.

Each blob in the Blobverse has 8 features corresponding to color, intensity, size, position, and texture:

Color and Intensity are represented in the LAB colorspace. Color is converted from RGB to LAB for a number of reasons. Firstly, the L channel of lab contains only intensity information and the AB channels contain only color information. Independence between color and intensity information is vital. As mentioned in the

previous chapter, the color of an object tends to remain constant under a variety of conditions, while the intensity varies. Secondly, the color channels mimic the opponent color channels found in the human visual system; A is a red/green channel and B is a blue/yellow channel. Thus, the differences in the A and B channels map well to perceptual differences.

Size is computed by counting the total number of pixels that make up the blob. **Position** consists of two numbers representing the vertical and horizontal position in the 2-dimensional image plane. These numbers are calculated by finding the centroid of the blob.

Texture is represented by two measures, contrast and anisotropy. Contrast represents the change in intensity within the texture region. Anisotropy is a measure of how the texture is ordered. A high anisotropy indicates that the texture has a preferred orientation. The Blobworld uses a third measure, polarity; polarity measures the preferred orientation of the texture. The Blobverse does not use this measure, so that blobs are rotationally invariant.

4.2 Matching and Clustering the Blobverse

4.2.1 Measuring Image Similarity

In order to cluster similar images, a metric for image similarity must be used. Images with a similar context will share some or all of the same blobs. Thus, our matching algorithm should attempt to find the closest correspondence between blobs in each image. One possible matching algorithm scores each possible correspondence and chooses the highest scoring correspondence as the best match.

There are infinite possible scoring metrics; I have designed one with model-based visual context in mind. I use two types of matching, qualitative and quantitative. A good quantitative match occurs when two corresponding blobs share similar numeric values across a set of features. A good qualitative match occurs when the pairwise qualitative relationships between blobs in one image match the pairwise relationships

in the other image.

The quantitative measure should be highest for a perfect match and should falloff as the difference between feature values increases. A possible candidate metric is inverse Euclidean distance $\frac{1}{\Delta^2}$, where Δ is the difference between feature values. Unfortunately, such a metric causes an infinity at zero. Furthermore, the non-linearity around zero is incredibly strong; a correspondence with a single near-perfect match will overrule a correspondence with multiple less-than-near-perfect matches. A more effective metric is the Mahanabolis distance: $e^{-\frac{\Delta^2}{2}}$. The Mahanobilolis distance has a maximal value of one (at $\Delta = 0$) and it approaches zero as Δ approaches ∞ . The quantitative score for a blob correspondence is calculated using the following equation:

$$\Gamma_j = \sum_{i=0}^M \frac{(A_j^i - B_j^i)^2}{var_i}$$

$$score_{quant} = \sum_{j=0}^N e^{-\frac{\Gamma_j}{2}}$$

A_j^i and B_j^i are the values of feature i for j th pair of corresponding blobs. Notice that the squared difference is normalized by var_i in the first equation. var_i is the variance of feature i . This normalization factor is introduced so that each feature has the same weighting. Since the score is computed by summing the Mahanobilolis distance for each pair of corresponding blobs, the maximum possible score is N , the number of corresponding blob pairs.

For qualitative relationships the quality of each corresponding relationship is binary, either they do or do not match. Thus, qualitative matches are scored by finding the proportion of matches:

$$score_{qual} = \frac{\text{Number of matching pairwise relationships}}{\text{Total number of pairwise relationships}}$$

Table 4.1: Feature Matching Techniques

Matching Technique	Features
Quantitative	A-Channel (color), B-Channel (color), Anisotropy (texture), Contrast (texture)
Qualitative	Intensity, Size, X-Position, Y-Position

Either the quantitative or qualitative matching technique can be used for each of the eight features. I have assigned a technique to each feature based on the expected type of variation between scenes in a single context. See table 4.1 for a listing of these assignments.

Similar contexts will contain similar objects. Across a variety of conditions, similar objects should have consistent color and texture characteristics. Under natural illumination, color does not change very drastically. The texture extraction method used is designed to act in a manner that is insensitive to changes in intensity and scale. However, intensity, size and position can vary under different lighting conditions, such as a sunny versus a cloudy day, intensity will vary greatly. If we assume roughly constant illumination across the scene, the qualitative relationship between intensities of blobs will remain the same under different lighting conditions. Absolute size and position of blobs in a scene can change drastically as the observer shifts his/her point of view, but the qualitative relationships are more stable.

4.2.2 Matching Heuristics

An exhaustive search of all possible blob correspondences is possible but computationally expensive. For each pair of images there are ${}_n P_r = \frac{n!}{(n-r)!}$ different correspondences between blobs, where n and r are the number of blobs in each image and $n > r$. Given that each image is composed of up to 10 blobs, an exhaustive search must score up to $10! = 3,628,800$ correspondences. If we are only matching two images, such a time bound does not seem so bad. As we start clustering, the number of image comparisons increases quickly. Fortunately, we can employ a number of search heuristics to help prune out a few possibilities.

Table 4.2: Number of Blobs vs. Number of Correspondences

Number of Blobs	Exhaustive Search	Vertical Alignment Heuristic
1	1	1
2	2	5
3	6	19
4	24	69
5	120	251
6	720	923
7	5,040	3,431
8	40,320	12,869
9	362,880	48,619
10	3,628,800	184,755

I have implemented two types of search heuristics, alignment heuristics and greedy feature search. The alignment heuristic forces the ordering of a specific feature across blobs to be preserved. Assume we have two images, a base image and a matched image. Blobs A and B in the base image correspond to blobs α and β in the a matched image. With the vertical alignment heuristic if blob A is above blob B , then α must be above β . Such a heuristic can be applied to any feature to reduce the number of candidate matches. Through experimentation, I have found that vertical position is the best feature to use with the alignment heuristic. In general, scenes with similar contexts share a similar vertical alignment. As table 4.2 shows, this heuristic provides a computational savings as the number of blobs increases.

Greedy feature alignment uses the quantitative feature set to prune the tree of possible correspondences. A blob α is selected from the base image and the k blobs from the matched image that are the closest quantitative match to α each become a branch in the tree. Each of these branches is grown out with the k blobs which are the closest match to the next blob in the base image. The trees are constructed so that any path from root to leaf contains at most one instance of each blob from the matched image. Unfortunately, this technique is prone to error due to confusion between similar blobs, such as blobs that are composed of “sky” and “water.” The qualitative information disambiguates these blobs, but can only do so if k is set to a

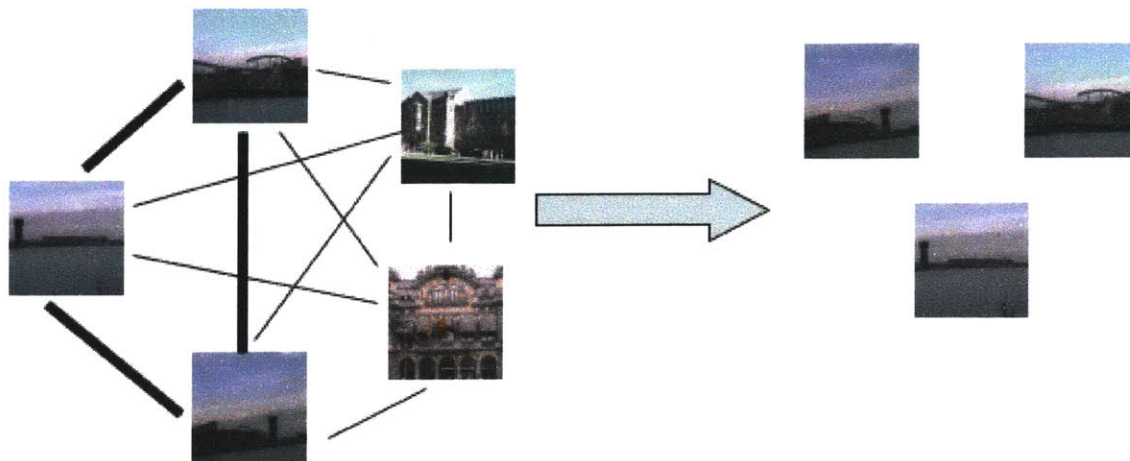


Figure 4-2: Clustering with connected components

sufficiently high value. As we increase k , we approach an exhaustive search. Thus, the results in the following sections use the vertical alignment heuristic to speed up the clustering process.

4.2.3 Clustering

The goal for clustering is to extract a group of images from the database that share a visual context. Assuming that the matching algorithm scores image pairs with a similar context highly, we can cluster images by finding groups of images for which every pair of images has a high match score. Such groupings can be found by building a graph and finding connected components. The process is sketched out in figure 4-2.

Every image is a node in the graph and there is a weighted edge between every pair of images. The weight of the edge is assigned by scoring how well the pair of images match. All of the edges below a threshold score, the thin lines in figure 4-2, are removed from the graph. Thus, every pair of images with an edge between them is a strong match. We can expect that these connected components share a common context.

Connected components is an appealing approach to clustering; it is simple, intuitive, and effective. Unfortunately, its computational complexity grows with $O(n^2)$, where n is the number of images in the database. By choosing to cluster around a

single image, we can reduce the complexity to $O(n)$. The algorithm works in the following manner:

1. Select an image, α , to cluster around.
2. Match all of the remaining images in the database against α
3. Add the top matching image to the cluster
4. Match each image in the cluster against the next highest match β . If all of the scores divided by the maximum possible score are above a threshold Θ , add β to the cluster.
5. Repeat step four with the top k images that match α .

The constants Θ and k can be set to alter the selectivity and maximum cluster size. There are a few disadvantages to this approach. Firstly, the approach requires the selection of an image to cluster around. However, in specific applications this image might be apparent (such as a database search, in which case the we would cluster around the query image).

Figures 4-3 and 4-4 are examples of clustering using the reduced complexity algorithm described above. The images in cluster example 1 are almost exclusively water, horizon, sky scenes. Four out of five images in cluster example 2 are suburban street scenes.

4.3 Learning Blobverse Models

The context models being learned will be entirely qualitative, based upon qualitative blob comparisons within image clusters. Given a cluster of images with a consistent context, blobs that correspond to the same object or scene component probably exist across images in the cluster. If these correspondences can be found, then the properties of these corresponding blobs can be analyzed to find consistencies. These consistencies will be used to generate context models. Some images may have extraneous blobs, blobs that do not correspond to the context. These blobs should not

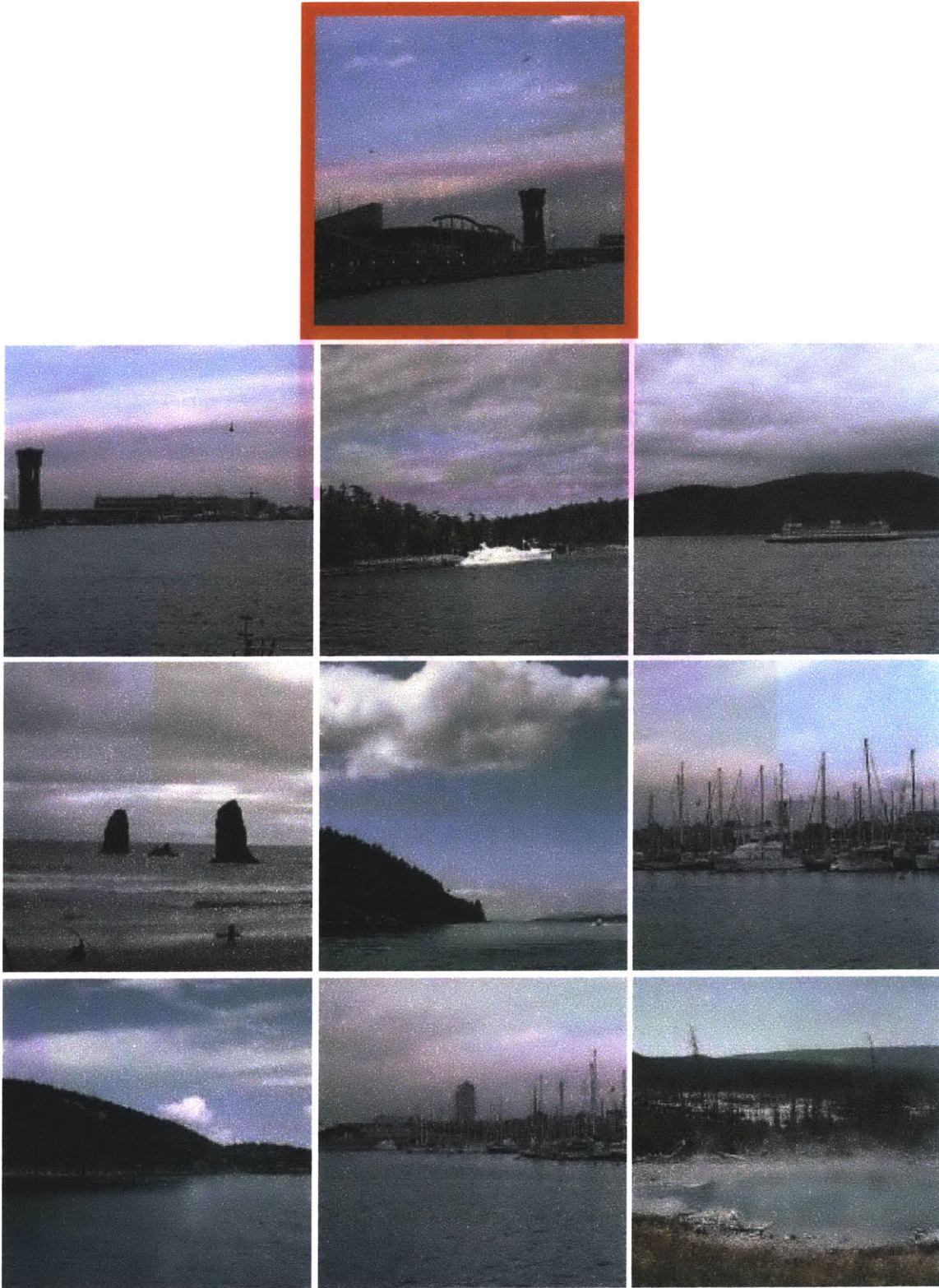


Figure 4-3: Clustering Example 1: The cluster is formed around the image with the red frame. Images were added to the cluster from left-to-right, top-to-bottom order. $k = 30$ and $\Theta = 0.66$.

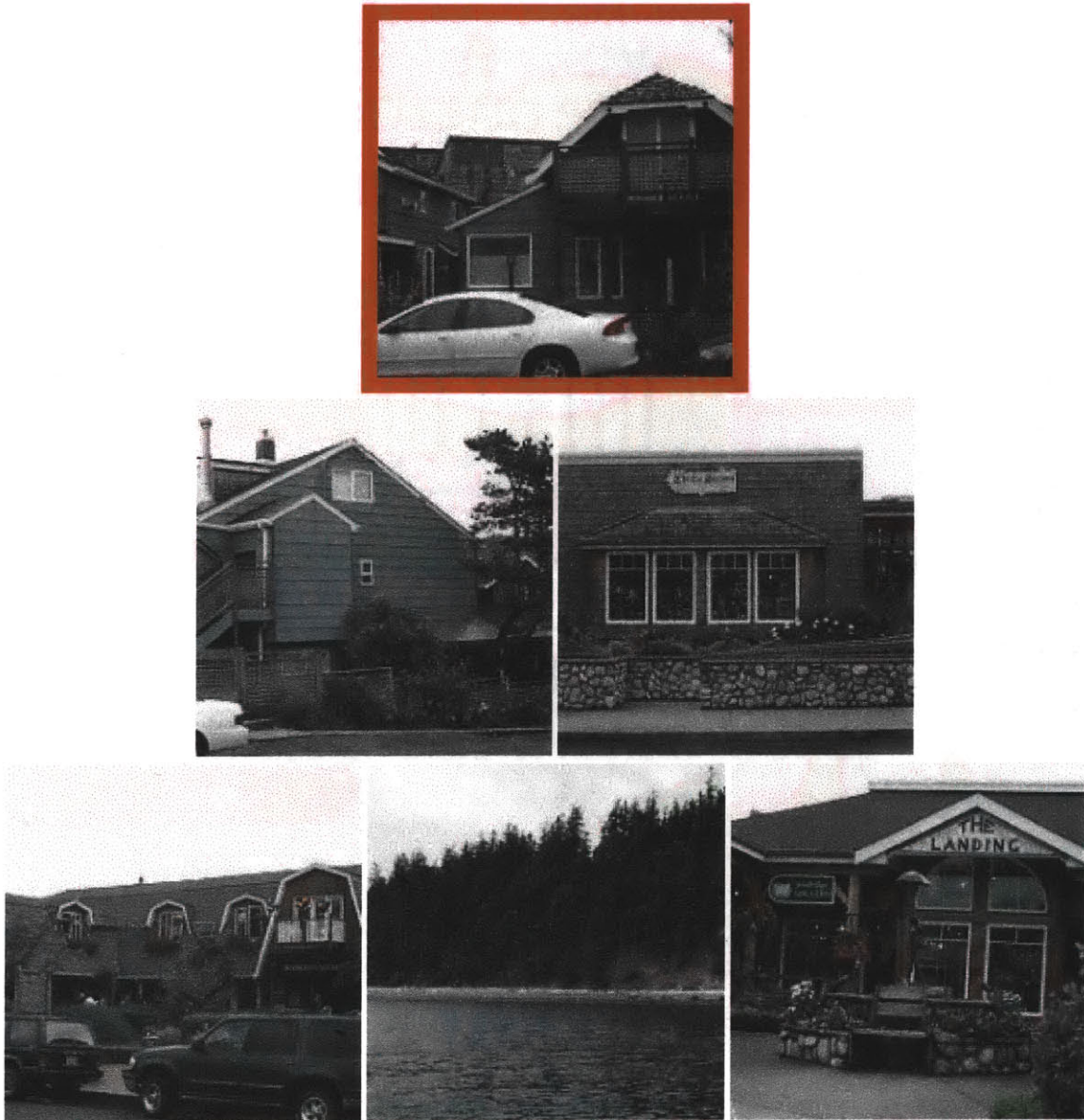


Figure 4-4: Clustering Example 2: The cluster is formed around the image with the red frame. Images were added to the cluster from left-to-right, top-to-bottom order. $k = 30$ and $\Theta = 0.66$.

be accounted for within the model. A learning algorithm should also be able to deal with a cluster that contains a few extraneous images, ones that do not fit the context.

The qualitative model learning algorithm that I have designed uses a prototype image to find blob correspondence. This image is meant to be the most representative image of the cluster. If we use the clustering algorithm discussed in the previous section, a logical choice is the image that the cluster was formed around. The learning algorithm uses another method for prototype selection. The prototype image should have blobs that correspond to the greatest number of blobs in each image. Thus, the algorithm find the image that is the best match to the most images in the cluster. Because the match score is based on the quality of blob correspondence, we expect such an image to accurately match the most blobs in each image across the image cluster. Since we must compare every image pair, prototype selection has a running time of $O(N^2)$, where N is the number of images in the cluster.

Once the prototype is selected, the blobs of each image are mapped to the blobs of the prototype. The mappings are found by running the blob matching algorithm between the prototype and each image in the cluster. Then, for each blob feature the algorithm counts each type of qualitative relationship ($<$, $>$, $=$) between each blob (as it maps to the prototype). If any of the counts exceed a threshold, $\alpha * N$ where N is the number of images in the cluster, the corresponding quantitative relationship is added to the model. The model is built in $O(N)$ time.

The threshold factor α must be greater than $\frac{1}{2}$ to insure that only one qualitative relationship is selected for every feature and pair of blobs. As α is increased, the model is simplified.

Applying the algorithm to the image clusters in figures 4-3 and 4-4, we yield a model for “water horizon scenes” and “suburban street scenes.” If we use this model to query the entire database we find the matches shown in figures 4-5 and 4-6.



Figure 4-5: Model Learning Example 1: The following images match the model learned from cluster example 1. $\alpha = 0.75$.



Figure 4-6: Clustering Example 2: The cluster is formed around the image with the red frame. Images were added to the cluster from left-to-right, top-to-bottom order. $\alpha = 0.9$.

Chapter 5

Discussion

5.1 The Best of Both Worlds

A context should be easy to learn. The reason for this is simple, context helps a learner to learn about the world. Context can start as a shallow understanding of a situation, place, scene, or conversation. This shallow understanding allows a learner to draw connections to previously acquired knowledge and to specialize a context if necessary. Learning must take advantage of redundancy and context allows redundancy to be identified.

Context can help one make a quick judgment of novelty. In a new situation, the learner should tread more carefully, observing minute details in order to make sense of the complex whole. When exposed to a similar situation, however, the learner can build upon the work done during the previous encounter. In order to maximize the benefit of reuse, the learner must be able to judge novelty as quickly as possible. In addition, such judgments should be robust to minor changes. For example, if the learner has visited many forests without flowers but never one with flowers, the learner should still be able to connect a forest with flowers with previous forest encounters. The ability to make this connection is important and allows the learner to quickly learn that forests can contain flowers.

The model based approach allows associations to be made between visual contexts without supervision. This is because the approach takes into account image structure

and content in a form that is invariant to changes with little or no effect on context, such as minor changes to scale and position. The complexity of the representation is also important for learnability. If we compare this approach to one used by Oliva and Torralba [6], the strengths and weaknesses of this complexity become clear.

Oliva and Torralba reduce context to an N -Dimensional feature vector, where $10 < N < 100$. The approach is appealing because the reduction is very straightforward, the outputs of filters are examined to find principal components. These principal components characterize almost all of the variation between natural scenes. The context is only one step away from low-level features.

The model-based approach requires an image to be reduced into components that correspond to elements of the scene. Unfortunately, segmentation is a problem that is far from being solved. To make matters worse, the problem is under constrained. Two different scenes can yield the same two-dimensional pictures. For example, any texture in a scene could be the result of small variations in depth or could be shading that is painted on to a surface. At best, we can make an educated guess by taking advantage of regularity in the world. The most effective segmentation techniques take advantage of color, texture, edges, and stereo data. For each image to be segmented, pixels must be classified into groups. Thus, context is many steps away from low-level features.

However, if we assume that all of these issues can be swept under the rug, the model-based approach offers learnability advantages because of its invariance properties. A context system based upon a feature vector will project to some Euclidean space. In order to learn a vector-based context, the space needs to be carved up. There are many existing methods for learning classifiers, such as support vector machines, neural networks, and decision trees. If the image space dimensionality is reduced incorrectly, certain context may overlap or images with the same context may be spread apart. As the sketch in figure 5-1 demonstrates, the required classifiers may need to be fairly complicated to fit the data. Even if all of these processes are easy, one problem still exists: we need tons of supervised data! In the real world, a child is not always given a label for a scene. Thus, they need a more effective way of tying

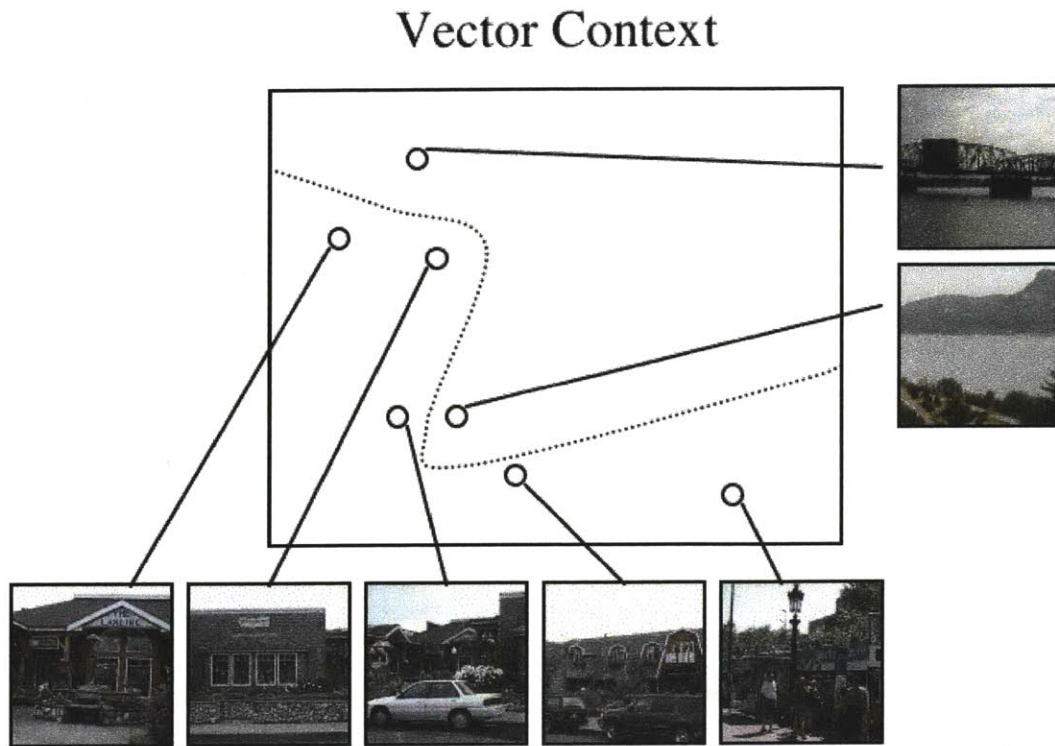


Figure 5-1: Complex classifiers may be necessary to learn from vector-based contexts contexts together.

The model-based approach does not require extensive supervision. Contexts are tied together by using scene content and structure. The models are descriptive enough that they can be compared. Additionally, the models lend themselves to introspection. As more is learned about the context, elements of the model can be given labels, like “sky” or “ground.” This opens up the way for explanation based learning.

The model-based representation is expensive to form. Texture segmentation is a hard problem that may require a great deal of computation. The vector-based representation is simple and is computationally cheap to build. However, model-based contexts are easier to learn than vector-based contexts. Clearly, each method has its tradeoffs. However, it may be possible to design a system in which these two methods bootstrap each other. Vector-based context can be used to find candidate context clusters by using nearest neighbors and the model-based approach can be used to edit these clusters. Also, the model-based approach can be used to tie together

similar contexts that are far apart in the visual context space.

5.2 How to make it work

Segmentation is key to the model-based approach. A poor segmentor will make learning impossible, a good segmentor will make learning easy. A good next step for producing effective model-based context is to develop a segmentation system specifically for the model-based approach. Such a segmentor would ignore small elements and would parse out large and salient scene components. The segmentor should be capable of amodal completion, if two areas of similar consistent texture are split by another region they should be merged into a single region. A few people standing on a field should not cut the field up into multiple image regions. Stereo and edge extraction can be used to find occluding contours, a particularly compelling division of scene segments.

An alternative approach is to specialize segmentation. This process would involve learning and applying texture detectors. These detectors could be used to find small neighborhoods of pixels that correspond to specific textures. The textures learned and applied by the system would be ones that occur commonly in scenes, such as “plant-like texture” or “building-like texture.”

Chapter 6

Contributions

Through this thesis, I have

- Implemented the Blobverse, a system that generates blobs from images and uses blob matching and image clustering to learn visual context models. The system has learned models for “water-horizon scenes” and “suburban street scenes.”
- Motivated the need to learn visual context and the utility of model-based visual context. In addition, I have discussed methods by which models can be learned.
- Discussed how visual context can be applied to visual search and recognition tasks and have provided arguments for how model-based visual context can improve these applications.

Bibliography

- [1] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. *Blobworld: A system for region-based image indexing and retrieval*. Springer, 1999.
- [2] M. Chun and Y. Jiang. Contextual cueing: Implicit learning and memory of visual context guides spatial attention, 1998.
- [3] J.M. Henderson and A. Hollingworth. High level scene perception. *Annual Review of Psychology*, 50, 1999.
- [4] Pamela Lipson. *Context and Configuration Based Scene Classification*. PhD thesis, MIT Electrical Engineering and Computer Science Department, 1996.
- [5] A. Oliva and P.G. Schyns. Coarse blobs or fine edges? evidence that information diagnosticity changes the perception of complex visual stimuli. *Cognitive Psychology*, 34, 1997.
- [6] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3), 2001.
- [7] Maximilian Riesenhuber and Thomaso Poggio. Stimulus simplification and object representation: A modeling study, 2000.