A Probabilistic Approach to Image Feature Extraction, Segmentation and Interpretation

By

Christopher Joseph Pal

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Mathematics

in

Computer Science

Waterloo, Ontario, Canada, 1999

Author's Declaration for Electronic Submission of a Thesis

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis describes a probabilistic approach to image segmentation and interpretation. The focus of the investigation is the development of a systematic way of combining color, brightness, texture and geometric features extracted from an image to arrive at a consistent interpretation for each pixel in the image. The contribution of this thesis is thus the presentation of a novel framework for the fusion of extracted image features producing a segmentation of an image into relevant regions. Further, a solution to the sub-pixel mixing problem is presented based on solving a probabilistic linear program. This work is specifically aimed at interpreting and digitizing multi-spectral aerial imagery of the Earth's surface. The features of interest for extraction are those of relevance to environmental management, monitoring and protection. The presented algorithms are suitable for use within a larger interpretive system. Some results are presented and contrasted with other techniques. The integration of these algorithms into a larger system is based firmly on a probabilistic methodology and the use of statistical decision theory to accomplish uncertain inference within the visual formalism of a graphical probability model.

Acknowledgements

First I would like to thank Dr. David Swayne, who has provided a great place for research at the Computing Research Laboratory for the Environment CRLE. I am grateful for your support throughout my time at the CRLE. You have afforded me the freedom to pursue my research interests, provided me with the necessary financial support, computing resources and opportunities for collaboration necessary to investigate solutions to real environmental problems. I am grateful to Dr. Brendan Frey who has helped me by providing an essential sounding board for many of the ideas presented in this thesis. Your financial support during the last portion of my study is also greatly appreciated. Thanks to Dr. Don Cowan who has provided me with a friendly home at the Computer Systems Group at the University of Waterloo. Thanks to Dieter Lehmann who provided some great collaboration an insights into the types of features that people use to detect patterns in aerial photography as well as guidance as to the types of landscape features that are of ecological importance. Thank you to Dr. Chunshen Pan for your comments on some of the sub-pixel derivations in this thesis. Thanks to my parents for their constant pestering (moral support), helping me get this thesis written. Finally, thanks to Sarah for putting up with my tunnel vision work practices used while writing this thesis.

Contents

Table of Figures

1 Introduction

Numerous systems and algorithms have been constructed for the processing, analysis and interpretation of digital images. Indeed, much of the research in methodologies and algorithms in the area of Statistical Pattern Recognition (SPR) and Artificial Intelligence (AI) have dealt with this task. Further, in industry many systems have been developed for various specialized tasks related to computer vision and image processing. In the past, many AI researchers have focused on the use of logic for solving "higher level" computer vision problems. In this context, "high level" can be considered as the interpretation of "lower level" extracted features. In contrast, SPR research has tended to focus more heavily on "lower level" issues more related to image processing and feature extraction. However, the difficulties involved with managing uncertainty using a logical formalism have led many AI researchers to investigate the use of more statistically grounded techniques. In particular, recent research in the AI uncertain-reasoning community has led to the development of a probabilistic way to encode a computer program that must deal with uncertain information in a sound manner. This formalism is often called a graphical probability model (GPM). These models have been shown to be able to express the typical programs that have in the past been called Expert Systems. These types of programs were initially encoded using Expert System shells or interpreters that essentially managed the control structure involved with executing an extremely large number of "if, then statements". Further, GPMs also allow many commonly used statistical procedures to be illustrated within an intuitive visual framework that can be used for the specification of a complicated probability model. Such graphs can thus be interpreted as either *probabilistic programs* or *statistical models*. As such, a number

of the more complex mathematical procedures associated with these graphs can be hidden from a user with little background in statistical theory. However, the same graph can also provide an intuitive visual framework for a researcher looking at improving the underlying algorithms used to perform learning and inference in the graph. In this way the use of probability graphs allows a rigorous and principled formalism for reasoning about uncertain information in a way that facilitates the incorporation of the model into a larger system.

The focus of this investigation is the fusion of features extracted over multiple scales of conceptual abstraction and scales of image resolution. The emphasis is on the integration of "higher level" reasoning techniques that have in the past been dealt with using logical techniques and "lower level" techniques that in the past have been dealt with using filter theory, linear algebra and statistics. Further emphasis is placed on the integration of these techniques into a larger system for interpreting and extracting features from relatively high-resolution (10cm to 10m per pixel) aerial imagery of the Earth. In particular, the features of interest in this investigation are those of environmental relevance. As such, the goal of this thesis is to automate the procedure of simultaneously classifying and segmenting features in aerial imagery to produce three general types of features consisting of:

- Segmented boundaries of objects suitable for vectorized storage
- Segmented regions consisting of the pixel level materials of objects
- Where appropriate, a measure of the most likely sub pixel components, for desired regions of the image

Two examples of typical hand generated region segmentations are shown in Figure 1.1.1 and Figure 1.1.2. A 512x512-pixel color image of an urban area is illustrated in Figure 1.1.1 and a 512x512 black and white image of farmland is shown Figure 1.1.2. The legend for the features in the respective segmentations is given following Figure 1.1.2. Both images have a resolution of 1m per pixel. In many cases, a second level of detail for an image may be desired. As such, similar images could be drawn for the more detailed elements in an image as illustrated in the finer scale classification urban image in Figure 1.1.3. Finally within the boundary of some of these regions, sub-pixel information may be desired. For example, one might be looking for a particular weed (possibly an illicit substance) that may be growing in an open grassed area and this weed might be detectable from the subtle variations in color measurements.

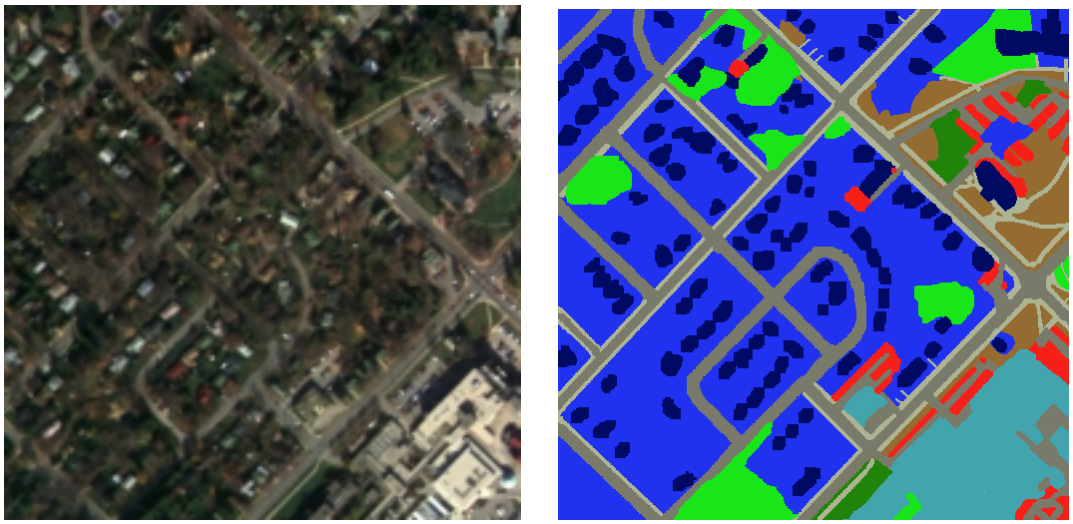


Figure 1.1.1 Left: An example of color aerial imagery of an urban area (image from GEOREF System Ltd.). Right: an example of object level regions of interest.

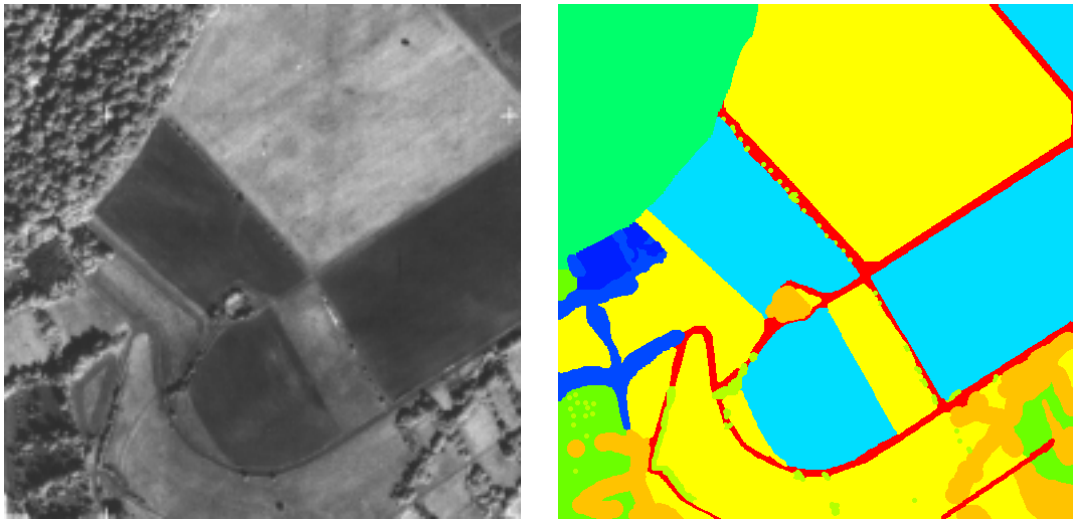


Figure 1.1.2 Left: An example of black and white aerial photography of farmland (image courtesy of Dieter Lehmann). Right: an example of object level regions of interest.

Legend for the Urban Area Classification of Figure 1.1.1

Color	Interpretation
	Sidewalk Area
	Dense Trees
	Industrial Area
	Road
	Sparse Trees
	Parking Lot
	Open Grass Field
	Residential Yard
	House Roof

Legend for Farmland Area Classification of Figure 1.1.2

Color	Interpretation
	Field Boundaries
	Enclosed Grass Area
	Deciduous Forest
	Small Sparse Trees
	Deciduous Tree Line
	Open Field
	Dense Coniferous
	Mixed Tree line
	Crops

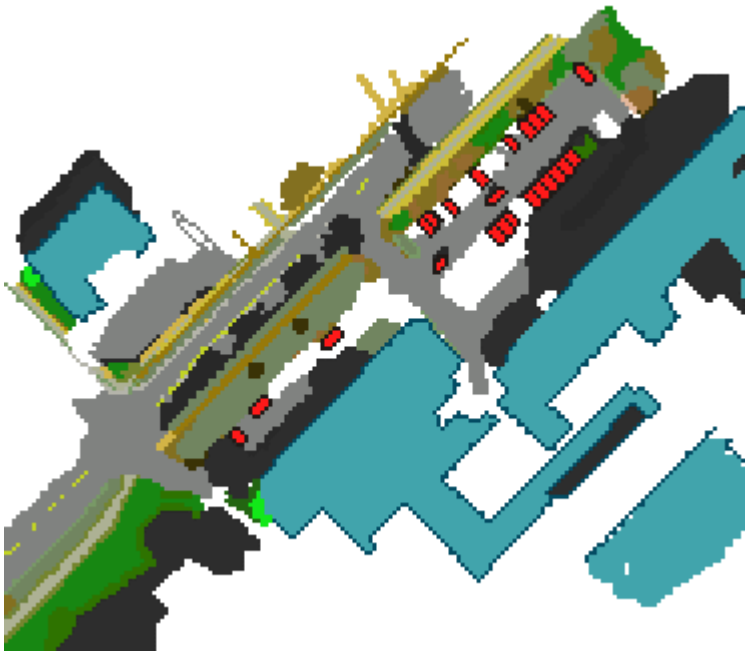


Figure 1.1.3 An example of a finer scale segmentation from the bottom right corner of the 512x512 image of an urban area. White areas were labeled as being unknown.

Legend for the Fine Scale Urban Area Classification of Figure 1.1.3

RGB Color	Interpretation
45 45 45	Unknown Deep Shadow
46 115 6	Shadowed Grass
172 155 44	Burnt Grass
24 135 19	Green Grass
117 133 47	Sidewalk and Grass
204 189 54	Sidewalk Grass Tree
173 177 148	Sidewalk Concrete
41 41 41	Shadowed Pavement
128 130 128	Pavement
121 153 102	Grass Pavement Concrete
181 181 170	Gravel Shoulder
194 212 37	Pavememt and Yellow Line
110 129 62	Pavement Red Tree Grass
152 122 15	Red Tree Sidewalk Shadow
210 180 70	Red Tree and Sidewalk

RGB Color	Interpretation
114 133 96	Red Tree Grass and Shadow
59 51 13	Red Tree and Shadow
131 113 33	Red Tree and Grass
227 184 157	Red Tree and Pavement
147 107 48	Red Tree
31 26 23	Car Boundary
255 25 25	Car Materials
12 76 99	Industrial Roof Boundary
65 165 172	Industrial Roofing
12 20 99	Residential Roof Boundary
27 45 238	Residential Roofing
35 101 20	Gree Tree and Shadow
19 231 26	Green Tree
255 255 255	Unlabeled

1.1 Motivation

Advances in technology have increased the availability and resolution of aerial imagery of the Earth's surface. This fact has great potential for the management, monitoring and protection of the environment. However, for many of these types of tasks raw raster based images must be painstakingly interpreted to extract relevant features for subsequent environmental analysis. The problem of extracting these features is closely linked to the somewhat higher level and more abstract task of actually interpreting what these features represent. There are many extremely useful applications of aerial image interpretation that can be found in the literature. Not surprisingly, many of these applications have military applications. For example, a great deal of research in the area of "high level" image interpretation of aerial photography has focused on the interpretation of military bases consisting of buildings, aeroplanes, vehicles, tanks and other military objects. Often this task is known as site reconstruction [49]. This can be contrasted with similar environmental applications where the digital interpretation of pixel color and the spatial organization of colored pixels into tree canopy shapes has been used to partially automate procedures for tree species identification [51]. This type of procedure is relevant for forest inventory and the monitoring of forest ecosystem health on a large scale. Often in the past, environmentally relevant applications of image analysis have focused on sub-pixel spectral analysis. This analytic need stems from the fact that there are large amounts of satellite data available, but at coarse resolution. A typical example is discussed in [32], in which sub-pixel analysis techniques are employed to classify wetland plants from Landsat (30-meter resolution) imagery. In this application, a large area of wetland where cypress was highly mixed with other species was classified. This classification was relevant to ecosystem analysis, as

cypress is an indicator species used to delineate water flow gradients. Some more modern technologies essentially involve mounting satellite-sensing technologies on airplanes and then gathering high-resolution color imagery in a manner similar to the more common black and white photography. These techniques allow even higher resolutions to be obtained. Road extraction is another extremely important research area for obvious military reasons. However, there are many instances where the detection of roads in imagery is important for environmental applications such as determining the alteration of natural watershed boundaries. Such information is useful for soil conservation and pollution transport modeling.

1.2 Thesis Overview

This thesis describes a system for image segmentation, feature extraction and sub-pixel inference. The algorithms used in the system allow the interpretation of images in the form of homogeneous regions suitable for storage in a compressed format. Features extracted from the spatial organization of pixel measurements are grouped into classes corresponding to more abstract, higher level concepts. This information is then used probabilistically to allow each separate pixel to be given a separate classification. In this way, the "context" of a given pixel is established through the fusion of abstract information derived over a larger scale with more localized analysis (relationships between neighboring pixels) and single pixel measurements (e.g. color). This analysis is used to establish a set of probabilities of particular materials being the primary subject of a given pixel. Given this probabilistic information, a form of probabilistic constraint satisfaction can be used to derive consistent sub-pixel interpretations in regions where this information is desired. Results and

conclusions are drawn for each independent section, while more general conclusions and recommendations appear at the end of this thesis. The sections of this thesis are organized in the following manner. First, the remainder of Chapter 1 provides an overview of existing systems and techniques for incorporating "higher level" abstract knowledge into the task of image interpretation. An overview of the common formalisms used in different research communities is also discussed. The techniques used in this thesis are grounded heavily in the use of probability and statistical decision theory. Thus, Chapter 2 provides an introduction to probability theory and the computation of probabilities within large probability graphs. Chapter 3 discusses some relevant aspects of Color Perception and Spectral Analysis. The notion of clustering colors is presented and described using a probabilistic model developed in earlier in Chapter 2. The limitations of such a model are discussed and the notions of local and global context are introduced. Chapter 4 introduces the concept of "local context" or the local relationships between pixels. A probability model is constructed for the inference procedure involved with reasoning about the local context of pixels. The model is constructed in such a way that it can be applied to all the pixels in an image in a consistent way. The limitations of modeling local context relationships are discussed and the notion of more global context is introduced. Chapter 5 discusses how the model developed in Chapter 4 can be used to allow the local context of small squares (of 16x16 pixels for this investigation) to be modeled in a similar way to the pixel level context model. Additionally the choice of an initial linear transformation for the squares of pixels is discussed. Then, the coarser scale information is combined with the pixel level information to improve the classification and segmentation procedure. Chapter 6 introduces some techniques for extracting curves from greyscale images and justifies the selection of the algorithm used in

this investigation. Chapter 7 discusses the relevance of perceptual relationships between curves for high level reasoning about image features. Past techniques for finding relevant perceptual relationships are discussed and a principled, systematic way to compute perceptual relationships between curves is presented. Chapter 8 discusses past approaches to sub-pixel inference. It is shown that given probabilistic estimates of the most likely sub-materials within a given pixel, a form of probabilistic constraint satisfaction can be used as a principled alternative to the standard techniques used in the literature for solving the mixed pixel problem. Chapter 9 presents some conclusions and a discussion of this investigation.

1.3 Knowledge Fusion and Image Interpretation

This section provides a literature review of the techniques that have been used in the past to incorporate features extracted over multiple scales of resolution from an image and the interpretation of these features using multiple levels of abstraction. First, there are numerous examples of early research in this area involving the use of rule-based expert systems to find consistent explanations for lower level features extracted from images. Some examples of this approach are presented below. Then, a discussion is given of the general trend in the expert system community to formalize uncertain inference. The resulting impact on image interpretation systems is discussed with some examples of more modern systems. Finally, a discussion of some general trends in Statistical Pattern Recognition with respect to multi-resolution analysis is given.

Logical Techniques and AI

Expert systems are essentially just a large collection of "if-then" statements. However, although these systems are conceptually simple, managing the control structure of a large expert system is rather difficult. For this reason, expert system shells were developed to manage the execution of rules, freeing the user from having to chart or codify all the possible ways in which a rule could become activated [28]. Some examples of these types of shells include OPS5 and NASA's CLIPS or C language integrated production rule system. There also exists a computational problem involved with pattern matching in systems constructed to execute a large number of "if then" statements. To address this type of problem, most of the commonly used expert systems of this nature use some variation of what is known as the Rete Pattern matching algorithm [22]. The early AI approach to image interpretation involved the use of such systems. Some of my own research presents a more detailed view of the software architecture of such a Rule-based Interpreter [55]. In this work the dynamic execution of a rule-based interpreter is illustrated using a call graph between functions that have been grouped into more abstract components.

One of the best early examples of the fusion of high level knowledge and low level feature extraction for image interpretation comes from the Digital Mapping Laboratory at the Computer Science Department of Carnegie-Mellon University. Here, a large system for the interpretation of primarily black and white aerial images known as SPAM [46] was under intense development from 1985 to the early 1990s. The system was developed to test the hypothesis that the interpretation of aerial imagery requires substantial (high level) knowledge about the scene under consideration [27]. High level knowledge about the

characteristics of various types of scenes was encoded into a production rule system and then used to constrain the search for plausible consistent scene models. The SPAM system accomplished the interpretation task by successively transforming so-called image *interpretation primitives* to primitives at a higher level of abstraction with a rule based system. For a typical scene interpretation hundreds of thousands of rules would be executed. Four classes of interpretation primitives were defined. The classes were given the following names: *regions*, *fragments*, *functional-areas* and *models*. The phases of interpretation took the following form. First, regions were created from low level segmentation algorithms. These regions were then converted into fragments based on the local properties of the segmentation such as shape, texture and height. For example, rules such as: *runways are typically 50 to 80 meters wide or houses are typically 8 to 10 meters high* would be encoded into appropriate production rules. Then a form of constraint satisfaction was applied to the generated fragment interpretations in which the relationships between different pairs of objects or fragments were checked for *local-consistency*. For example, such constraint rules might take the following form *runways have perpendicular taxiways*. Of significant importance is the fact that multiple constraints exist between pairs of objects and these constraints interact jointly on the fragment hypotheses. These constraints were thus used to generate the so-called functional areas, represented using the convex hull of features within the functional area. For example the terminal functional area was defined to contain only terminal building, parking lot, road and parking apron hypotheses. A final model of the scene was then generated using heuristic applied to the various hypothesis quality measures of the functional areas. This type of rule based approach is still used as a means to fuse low level features with high level knowledge in more modern designs. A more modern example (late

1990s) is given in [18] where a system for model-based object recognition in perspective aerial images is described using a similar rule-based interpretive process. This paper also reviews a number of other rule-based approaches following a similar design pattern.

Managing Uncertainty: AI and Statistics

As one might imagine, rule based systems are a natural way of encoding high level knowledge. However, large expert systems are difficult to manage. Consistency problems can easily arise in the system and finding these inconsistencies can become a troubling problem. For example Digital corporations XCON expert system used for configuring the VAX required an extremely large number of people for maintaining the underlying rule base. The SPAM group has reported similar difficulties with respect to rule consistency [27]. Yet, the issue of how to use high level knowledge in a useful way is one of the central issues involved with automating image interpretation. David McKeown Jr., principle researcher at CMU's Digital mapping lab stated in [48] (his top ten list of lessons learned in automated cartography): "Simply talking about lots of knowledge doesn't get you very far. [Organizing knowledge and] showing how vision systems work better with knowledge than without knowledge is a hugely difficult and largely open problem."

Further, one of the key difficulties with the use of rule based systems to manage and manipulate knowledge for the fusion of low level features extracted using any algorithm, is the management of the uncertainty involved with combining these features. This problem is not unique to the image understanding research community. The general issue of reasoning

with uncertain information has been the focus of research in the expert system community. As a result of this research, a graphical formalism has been developed to represent programs that were once coded using rule based programming languages. Using this formalism one can think of the graph as a probabilistic program that also corresponds to the specification of a complicated probability distribution. The program is executed not by evaluating rules within a rule interpreter, but by passing probabilistic messages in a graph. The use of such formalisms for solving real world problems was demonstrated extremely well by the Pathfinder project [29] at Stanford's Medical Informatics Department. Here a large medical expert system was developed is a series of stages starting with a production rule system, experimenting with different strategies for managing uncertain inference. This research lead to the conclusion that the probability graph was the best underlying formalism for managing, specifying and updating uncertain information for this application. Further, the system that was constructed using this formalism currently outperforms its expert creators, with respect to its diagnosis task.

The now classic, simple example of such a probability graph originally proposed by Lauritzen and Spiegelhalter [43] is shown in Figure 1.3.1. Each node in the graph is a random variable taking on a set of states. The probability of the variable's state is shown numerically and visually using a bar graph.

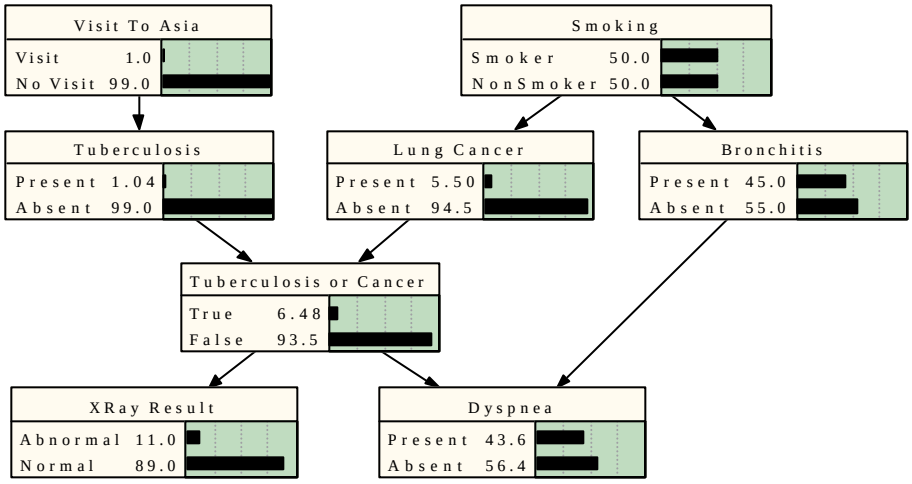


Figure 1.3.1 A simple probability graph for diagnosing chest pain (from Lauritzen and Spiegelhalter [43]).

Figure 1.3.1 illustrates what is known as the *marginal probability distribution* of the variables in the graph. The marginal probabilities are not specified at design time. What must be specified are *unconditional probabilities* for variables with no parents and *conditional probabilities* for variables with parents. These graphs and their generalization to variables outside the medical domain are very useful for working with large probability models.

Probability Graphs: AI and Pattern Recognition

Many of the techniques employed by researchers in the area Remote Sensing employ analysis techniques that can also be viewed as instances of graphical probability models. Typically they look at the fusion of features extracted from different scales using a "fractal like" approach based more on linear algebra than on an encoding of uncertain logical relationships as in the AI expert system example. These techniques involve the *multi-resolution analysis of localized features* in which images are analyzed using the same general

technique, but applied to the image at different levels of resolution. Examples of this type of analysis include Wavelet analysis, Fourier Analysis, and Image Pyramid filters (like wavelets but not an orthogonal basis). These filters are then combined to analyze the image. These techniques are closely related to our probabilistic models and will be discussed in further detail later in Chapter 5. Finally, one other related approach based heavily on information theory involves the management of higher level concepts using attributed hyper-graphs [74]. This approach is beyond the scope of this thesis and will not be discussed in further detail. As it is clear that the formalism used in this thesis involves the use of graphical probability models, Chapter 2 begins with a review of some probability theory and discusses the computation of probabilities in large graphs using this generalized graphical framework.

2 Probability Theory, Graphs and Density Estimation

2.1 Basic Probability Theory and Bayes' Theorem

A fundamental rule for working with probabilities of discrete events using probability calculus is the following. For events A and B with probability $P(A)$ and $P(B)$, the *joint* probability $P(A,B)$ and the *conditional* probability $P(A|B)$ are related by:

$$P(A \mid B)P(B) = P(A,B)$$

Eq. 2.1.1

The probability of the *joint* event A and B also denoted $P(A \wedge B)$ in some conventions. From this initial rule one derives the well-known *Bayes' rule*

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}$$

Eq. 2.1.2

The calculation of $P(B \mid A)$ in this way is often referred to as finding a *posterior probability.* Further, in this context $P(B)$ is often referred to as a *prior probability.* Finally, $P(A)$ is known as an *unconditional* probability.

Often one wishes to describe a probabilistic model in terms of variables that take on a finite number of states. Thus, if A is a variable with some states a_1, a_2, …, a_n then $P(A)$ is a *probability distribution* over these states

$$P(A) = (x_1, x_2, \ldots, x_n); \qquad x_i \geq 0; \qquad \sum_{i=1}^{n} x_i = 1$$

and x_i is the probability of A being in state a_i. A state variable that has only two states can be referred to as a *binary variable*. A state variable that has more than two states can be referred to as a *categorical variable*. Next, given the joint probability for some variables $P(A, B, C,$

etc…) finding the unconditional probability distribution of one of these variables can be performed in a calculation know as *marginalization*. A variable is marginalized out of the joint probability by summing over all of the states of the *other* variables. This produces an unconditional probability, for example

$$P(A) = \sum_b \sum_c P(A, B = b, C = c)$$

Eq. 2.1.3

Variables do not necessarily have to be represented by discrete states. Continuous variables obey the same generalized rules. For continuous variables which are (in some sense) integrable, summations become integrals. Given a *conditional distribution* $P(x \mid C)$ over continuous variable x and given the class variable C, with a prior for the class $P(C)$ one can find the *posterior distribution* using Bayes' theorem in the usual way

$$P(C \mid x) = \frac{P(x \mid C)P(C)}{P(x)}$$

Eq. 2.1.4

Conditional distributions over continuous variables e.g. $P(x \mid C)$ are often parameterized in some way for computational efficiency. Further, if one views a class conditional distribution of *probability density* (i.e. $P(x \mid C)$) as a function of the parameters (e.g. x and C) these types of probabilities are referred to as *likelihood functions* for the observed value of x. This leads to the often used summary of Bayes' theorem as

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalizat ion}}$$

Eq. 2.1.5

2.2 Probability Graphs

One way to model complex probability distributions that is popular in the expert system community uses conditional distributions e.g. $P(x|C)$ and unconditional distributions of random variables e.g. just $P(C)$. For the purposes of creating software systems, specifying distributions using only these types of probabilities can be equivalently expressed using a graphical formalism. In this formalism, arrows are used between nodes in a graph to denote the required conditional probability distributions between the random variables. Figure 1.3.1 is a commonly used example of such a graph. Variables with no arrows pointing to them must thus be associated with an unconditional probability. Specifying a complicated probability distribution in terms of unconditional probability distributions for variables with no parents and conditional distributions for variables with parents can thus be equivalently expressed as a directed acyclic graph or a DAG. A very simple example of such a graph is illustrated in Figure 2.3.1. This type of formalism has been extremely popular in the medical expert system community [29]. The use of this directional arrow notation was popularized as doctors commonly think of diseases as causing symptoms. Models specified using this formalism have also been shown useful for solving some environmental problems [30]. However, a more generalized form of probability graphs is discussed in section 2.3 in which arrows are not necessarily used. Early research into constructing such systems involved specifying the structure of the graph and required the specification of the conditional and unconditional probabilities at design time. More current research involves "learning" both the structure and the probabilities of the graph from data [71].

There are two widely used families of algorithms for updating the probabilities of DAGs when new evidence becomes available [62]. Both of these algorithms are based on the intuitive notion of passing probabilistic "messages" between nodes in a graph. The first family, popularized in the expert system community by Pearl [58] is based on probabilistic message passing directly in the DAG. However, these algorithms are only provably correct for *singly connected* directed acyclic graphs (DAGs). Singly connected DAGs are of the form that each link is a bridge (i.e. removal of a link will disconnect the network) [37]. These types of DAGs are alternatively known as *tree structured* graphs or graphs with *no cycles regardless of the direction of the edges.* The second family of algorithms derive from the propositions of Lauritzen and Spiegelhalter [43] and apply to general DAGs. Here message passing is performed not in the graph directly, but in a transformation of the graph based on converting DAGs into trees consisting of clusters of variables with undirected links between the clusters. The next few sections describe the updating process involved when incorporating new probabilistic information within networks that have been constructed with the appropriate conditional and unconditional probabilities.

2.3 Updating Probabilities in Tree Structured DAGs

Consider the following simple medical example, in which there is a binary disease variable. The disease variable may cause a binary symptom variable to be present and also may cause the result of a particular test to be positive or negative. A probabilistic model can be constructed for this simple situation in the following way.

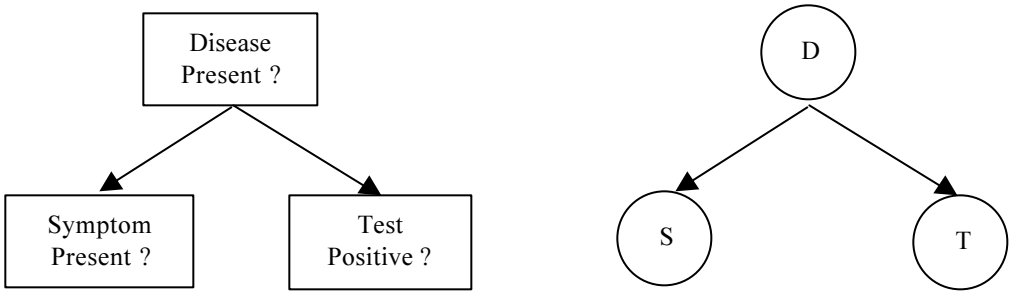


Figure 2.3.1 A simple medical example with one disease one symptom and one test.

Recall, for this model to be completely specified, one must supply the probabilities $P(D)$, $P(S|D)$, $P(T|D)$. Once these probabilities have been entered into the network, the joint probability of the entire network can be calculated in this or any other tree-structured graph by simply labeling the nodes in ascending order and summing over the states of all the variables. The probability of any node can then be found by marginalizing the given node out of the joint probability of the entire network. In the example of Figure 2.3.1 these calculations would proceed as follows.

$$P(D,S,T) = \sum_d \sum_s \sum_t P(S = s \mid D = d) P(T = t \mid D = d) P(D = d)$$

Eq. 2.3.1

$$\begin{aligned} P(S) &= \sum_d \sum_t P(D,S,T) \\ &= \sum_d \sum_t P(S = s \mid D = d) P(T = t \mid D = d) P(D = d) \end{aligned}$$

Eq. 2.3.2

The calculations of Eq. 2.3.1 and Eq. 2.3.2 can be performed with one large summation over all possible joint assignments. However, the time for these summations can be greatly reduced if one observes that the sums can be "pushed right" over terms that do not contain the variable of the summation. This forms the basis of the somewhat more intuitive

explanation of Pearl's message passing algorithm known as the *sum-product algorithm* [41]. Finding a marginal probability for a variable using the somewhat complicated message-passing notation of Pearl is thus simply equivalent to evaluating the factorized summation of a marginal calculation. To make this clear the *factor graph* notation can be used in which both the conditional and unconditional probability distributions are made explicit as nodes in the graph. A factor graph is a bipartite graph that expresses how a global function of many variables factors into a product of local functions [41]. In such a graph there is a variable node for each variable in the graph (commonly indicated by a circle) and a function node for each factor (commonly indicated by a rectangle). A variable node is connected to a function node if and only if the variable is an argument of the function. As the sum-product algorithm is general and not specific to probability distributions, each probability distribution can be regarded as a general function of the variables for the distribution.

Message passing proceeds in the graph as follows. A node sends a message to its neighbors when it has received a message from each of its other neighbors. Function nodes send messages to variable nodes consisting of the marginal probabilities for the states of that variable from their local function multiplied by the incoming messages. If a function node only has one connection, it sends an unconditional probability for the connected variable found in its local table. Variable nodes send messages consisting of the product of their incoming messages for each of their states from the neighboring function nodes. If a variable node is only connected to one function node, it sends a "dummy" message consisting of a 1 (one) for each state of the variable. If the graph is a tree, each node is guaranteed to send a message to every neighbor and receive a message from every neighbor. The marginal

probabilities for the states of a variable node are found from the product of all the incoming messages. The derivation leading to Eq. 2.3.3 illustrates the messages used for a calculation of the marginal probabilities of variable S. Figure 2.3.2 illustrates the messages graphically.

$$
\begin{aligned}
P(S) &= \sum_d \sum_t P(S, D=d, T=t) \\
&= \sum_d \sum_t P(S \mid D=d)\, P(T=t \mid D=d)\, P(D=d) \\
&= \sum_d P(S \mid D=d) \underbrace{P(D=d)}_{\text{message 1a}} \underbrace{\sum_t P(T=t \mid D=d)}_{\text{message 1b}} \\
&= \sum_d P(S \mid D=d) \underbrace{P(D=d) \sum_t P(T=t \mid D=d)}_{\text{message 2}} \\
&= \underbrace{\sum_d P(S \mid D=d)\, P(D=d) \sum_t P(T=t \mid D=d)}_{\text{message 3}}
\end{aligned}
$$

Eq. 2.3.3

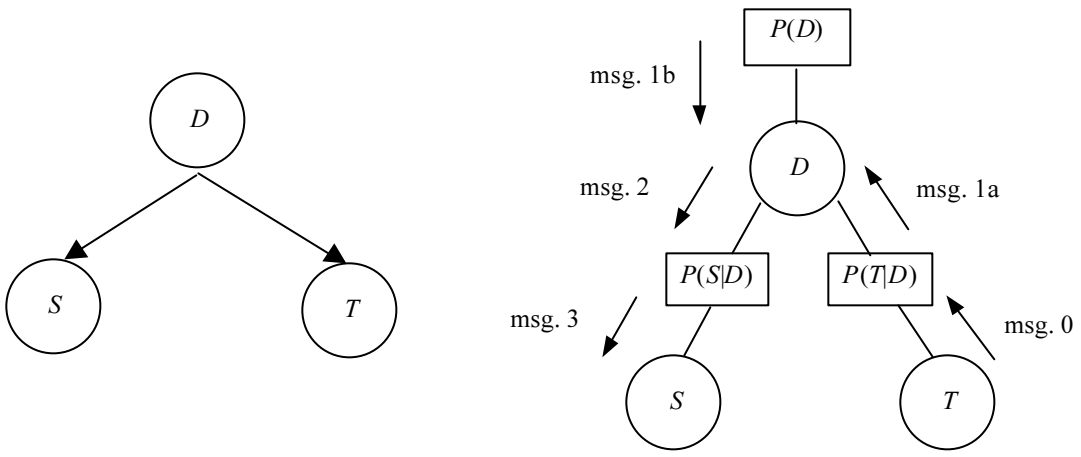


Figure 2.3.2 A DAG (left) and a factor graph (right) for the simple medical example.

For more complicated graphs, if one wishes to find the marginal probabilities for each variable, the computation is greatly simplified using this message passing scheme. In fact, the computation is linear in the number of connections using this message passing strategy as two messages get send down each connection. If new evidence is entered into the network,

the algorithm is modified so that variable to function messages are always set to the value of the probabilistic observation. At the end of the message passing, the probabilities for the nodes with new evidence can either be held constant or updated with their appropriate posterior probability derived from the product of all the incoming messages.

This type of message passing algorithm can be used to find approximate solutions in some probability graphs. In fact, for complicated graphs such as those used to model certain coding algorithms, approximate inference using message passing in DAGs with undirected cycles has been shown to produce acceptable approximate solutions in linear time approaching the theoretically fastest solution. A recent result has shown that some of the fastest decoding algorithms currently known can be viewed as message passing in a DAG with undirected cycles [40]. However, there are certain circumstances when no amount of message passing in a DAG will produce a correct result. The following section discusses when this is the case and introduces the family of algorithms that can be used to solve the problem of probabilistic inference in a DAG exactly.

2.4 Updating Probabilities Exactly in Unrestricted DAGs

The fundamental problem with cycles is that evidence on variables may be coupled [37]. Message passing of probabilities with respect to single variables in the DAG decouples the evidence with respect to multiple variables and this information is thus lost at the meeting point. To address this problem, a second family of algorithms based on the insights of Lauritzen and Spiegelhalter [43] have been developed to convert the DAG into a tree

structured *Markov Random Field* (MRF) or a graph with no arrow, consisting of clusters of variables. Such a graph is known as a *junction tree*. Message passing in such a graph is then accomplished by using the fact that a Markov distribution on a such a graph can be written as a product of the distribution on the cliques of the graph divided by the product of the distributions on their intersections [62]. Let X_G represent all the variables in the graph, Q represent the set of all cliques, X_q represent all the variables in a clique, S represent the set of all intersections and X_s represent all the variables in an intersection. The symbolic form of the decomposition can then be written as follows.

$$P(X_G) = \frac{\prod_{q \in Q} P(X_{q_i})}{\prod_{s \in S} P(X_s)}$$

Eq. 2.4.1

This representation is known as a *marginal representation* and is a special case of the more general potential representation written using positive functions (not necessarily probability distributions) in the following way

$$P(X_G) = \frac{\prod_{q \in Q} \phi_q(X_q)}{\prod_{s \in S} \psi_s(X_s)}$$

Eq. 2.4.2

Converting a general potential representation into a marginal representation is a key step in Lauritzen-Spiegelhalter style message passing algorithms. Importantly, the clusters or cliques derived from the DAG are determined by analyzing the network and creating clusters in such a way as to ensure that no coupled information is lost.

Consider entering new evidence about a test into the directed graph used in the simple medical example. The probability of S (a symptom) can be found by replacing the dummy message used in the marginal calculation with an evidence message. The equation form of the calculation is similar to Eq. 2.3.3 but with the evidence added in place of the dummy message of ones at the end of the equation.

$$P^*(S) = \sum_d P(S = s \mid D = d) P(D = d) \sum_t P(T = t \mid D = d) P^*(T = t)$$

Eq. 2.4.3

However, this probability can also be calculated using the fact that $P(S,D)=P(S|D)P(D)$ and $P(T,D)=P(T|D)P(D)$ and then using these joint probabilities explicitly during the computation. The computation is illustrated in the following way, where the $X = x$ notation within probability calculations has been replaced with simply X, for clarity.

$$\begin{aligned} P^*(S) &= \sum_d P^*(S,D) \\ &= \sum_d P(S \mid D) P^*(D) \\ &= \sum_d \frac{P(S,D)}{P(D)} P^*(D) \\ &= \sum_d \frac{P(S,D)}{P(D)} \sum_t P^*(D,T) \\ &= \sum_d \frac{P(S,D)}{P(D)} \sum_t P(T \mid D) P^*(T) \\ &= \sum_d \frac{P(S,D)}{P(D)} \sum_t \frac{P(T,D)}{P(T)} P^*(T) \end{aligned}$$

Eq. 2.4.4

The updating strategy in Eq. 2.4.4 is such that the two joint probability distributions communicate by passing messages on new probabilities for their intersection. New evidence is entered into a joint probability distribution similarly with an initial message on the variable. Graphically, the message-passing scheme on the joint distributions and their

intersections can be illustrated in an undirected graph known as a *junction graph* indicating clusters of variables found from the *cliques* of the graph and the intersections of the cliques, commonly known as *separators.* Such a junction graph can also be illustrated in the factor graph notation where the probability tables used to encode the information for the cliques and separators are illustrated explicitly. Figure 2.4.1 and Figure 2.4.2 illustrate four equivalent ways to represent the same probability distribution. Figure 2.4.2 illustrates the updating strategy for factor graphs of DAGs and this is contrasted with the slightly different updating strategy proposed by Jensen [37] for junction trees, but here it is illustrated in a factor graph. The junction graph terminology that is commonly used to describe each operation involved with this message passing scheme is also shown as labels above the arrows and variables in the graph of Figure 2.4.2.

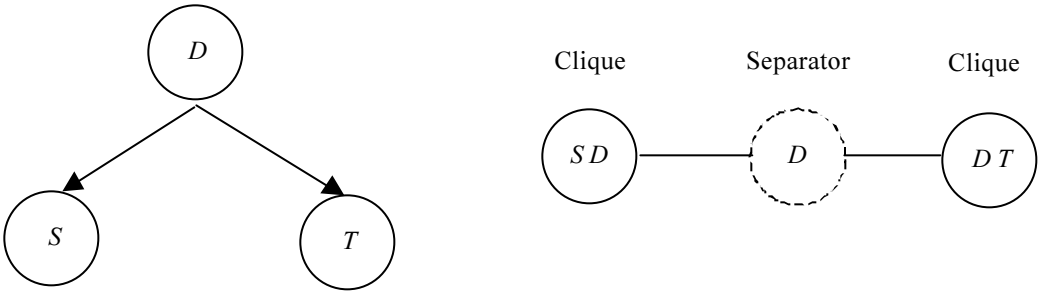


Figure 2.4.1 A DAG and a Junction Graph for a simple medical example.

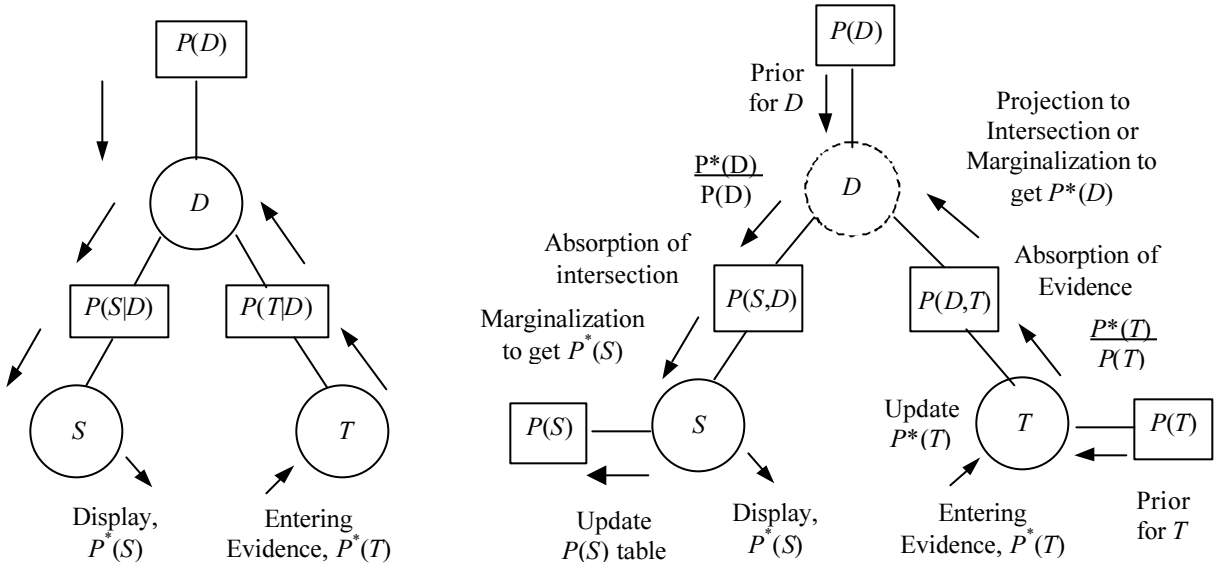


Figure 2.4.2 Left: Message passing in a factor graph for a DAG. Right: Message passing in a factor graph for a junction tree.

$$
\begin{aligned}
P^*(S) &= \sum_d \frac{P(S,D)}{P(D)} \underbrace{\sum_t \frac{P(T,D)}{P(T)} P^*(T)}_{\text{absorption of evidence}} \\
&= \sum_d \frac{P(S,D)}{P(D)} \underbrace{\sum_t \frac{P(T,D)}{P(T)} P^*(T)}_{\text{projection to the intersection or marginalization to get } D^*} \\
&= \sum_d \underbrace{\frac{P(S,D)}{P(D)} \sum_t \frac{P(T,D)}{P(T)} P^*(T)}_{\text{absorption of intersection or P(S,D) calibrates to P(D,T)}} \\
&= \underbrace{\sum_d \frac{P(S,D)}{P(D)} \sum_t \frac{P(T,D)}{P(T)} P^*(T)}_{\text{marginalization to get the updated probability for S}}
\end{aligned}
$$

Eq. 2.4.5

Tree structured DAGs can be converted into Junction graphs by constructing an intermediate graph known as a *moral graph*. A moral graph is constructed by introducing undirected links between each pair of variables that appear together in a relation. The result of this procedure is that:

1. All directed edges will be replaced by undirected edges and
2. Variables with a common child are linked together (giving this graph its name)

The distribution then takes the form of a MRF on the moral graph (a moral graph consists of only single variables, not clusters). From the moral graph, the clusters of variables used to construct the junction graph are found from the *cliques* in the moral graph. The cliques of a graph are the maximal complete sub-graphs (the usual definition). However, in some cases the Junction graph for a corresponding tree structured DAG will contain cycles and message passing in a cyclical junction graph may not be exact. But, in general any cycles in the junction graph of a tree structured DAG will have the same intersection node on all the links of the cycle [37]. Removing any of the links can break the cycle and produce a *Junction Tree* in which message passing between cliques and their intersections will result in exact probabilistic inference. For DAGs that are not tree structured, the corresponding junction graph may contain cycles that *do not* have the same intersection node on all the links of the graph. For these situations other algorithms [37] have been developed to convert the junction graph into a junction tree so that the message-passing scheme in the junction tree can be used to produce exact solutions. However these solutions can be quite computationally expensive as the size of the cliques may get quite large. There are other ways to compute exact solutions in multiply connected DAGs that do not involve converting the graph into a junction tree such as Shacter's arch reversal algorithm [69] or the conditioning procedure proposed by Pearl [57]. Further details of all of these algorithms can be found in the associated references.

The following sections will deal with the construction of a relatively large graphical probability model approaching the size in which the distinctions between a probability model and a probabilistic program is blurred. In this model, bayesian inference and these message-passing principles will be used to update probabilities in the graphical representation of the model. But, first a note must be made about probability distributions over continuous variables (e.g. $P(\bar{\mathbf{x}} \mid k)$). One way to treat a probability distribution over a continuous variable or vector is to discretize the variable into "bins" or states of a discrete variable. However, there are many cases where the discovery of the optimal discretization is critical and it is much more efficient to use some form of parameterized function to specify the probability distribution. The next section deals with constructing such parameterized distributions.

2.5 Mixture Models for Continuous Probability Density Estimation

One way to represent an unconditional probability distribution for continuous data is in terms of a *mixture model*. In this context, a model is constructed and parameterized to estimate a *probability density.* Thus, when the mixture model takes on an appropriate functional form, certain parameters of the model can be interpreted as higher-level class variables in a probability diagram. Consider that we could construct a density function as a linear combination of basis functions where the number m, of basis functions is much less than the number of data points. In this case, one can write the model for the density of a given class as a linear combination of a set of m independent and parameterized, component or "class"

densities $P(\vec{\mathbf{x}} \mid k)$ such that $\vec{\mathbf{x}} \in R^n$, where n is the number of dimensions. Such a model takes the following equation form.

$$P(\vec{\mathbf{x}}) = \sum_{k=1}^{m} P(\vec{\mathbf{x}} \mid k) P(k)$$

Eq. 2.5.1

In graphical form, such a model may be illustrated as

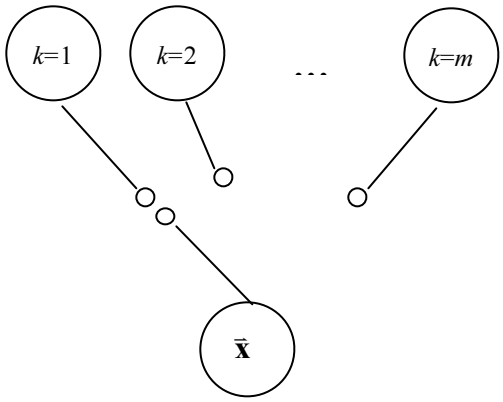


Figure 2.5.1 A switch like model for $P(\vec{\mathbf{x}})$.

The probabilities $P(k)$ can be called prior probabilities of a data point having been generated from component k of the mixture, they are often called the mixing parameters and in the literature they are commonly denoted by π_k. These priors are constrained in the following way

$$\sum_{k=1}^{m} P(k) = 1$$
$$0 \le P(k) \le 1$$

Eq. 2.5.2

These constraints mean that one can write the graphical probability model in a way that collapses the top layer of nodes in Figure 2.5.1 into a single categorical variable K taking on states $k \in [1,2,\ldots,m]$, where m is the number of classes.

Figure 2.5.2 A model of a categorical class variable *K* causing a distribution over $\vec{\mathbf{x}}$.

Further, as one might expect, the component density functions must also be normalized so that

$$\int_{-\infty}^{\infty} P(\vec{\mathbf{x}} \mid k) d\vec{\mathbf{x}} = 1$$

Eq. 2.5.3

With these constraints on the components, $P(\vec{\mathbf{x}} \mid k)$ can thus be regarded as a class conditional density [4]. Note that the unconditional density $P(\vec{\mathbf{x}})$ is also normalized correctly as a result of these constraints, i.e. the following is also true.

$$\int P(\vec{\mathbf{x}}) = \sum_{k} P(k) \int_{-\infty}^{\infty} P(\vec{\mathbf{x}} \mid k) d\vec{\mathbf{x}} = 1$$

Eq. 2.5.4

Data can be generated from such a model by selecting a point at random with probability $P(k)$ and then generating a data point from the corresponding component density $P(\vec{\mathbf{x}} \mid k)$. Now if we wish to find the probability of a component or class, then we can use Bayes' theorem

$$P(k \mid \vec{\mathbf{x}}) = \frac{P(\vec{\mathbf{x}} \mid k)P(k)}{P(\vec{\mathbf{x}})}$$
$$= \frac{P(\vec{\mathbf{x}} \mid k)P(k)}{\sum_{k=1}^{m} P(\vec{\mathbf{x}} \mid k)P(k)}$$

Eq. 2.5.5

This represents the *posterior* probability that a particular subclass k was responsible for generating the data point $\vec{\mathbf{x}}$.

2.6 A Simple Example of a Mixture Model

Consider the following two simple examples in two dimensions where there are $m = 2$ and $m = 3$ classes respectively. In both cases the classes have equal unconditional probabilities, that is $P(k) = 1/2, \forall k \in K$ in the first example and $P(k) = 1/3, \forall k \in K$ in the second example. In the first case the conditional distributions i.e. $P(\vec{\mathbf{x}} \mid k)$ for two of the classes are Gaussian distributions. The Gaussians have means at $(x, y) = (.25, .5)$ and $(.75, .5)$ for $k = 1$ and $k = 2$ respectively. In the second case, a third conditional distribution is added that is uniform over the range illustrated. The following figures illustrate the contour lines of the associated probability distributions with these models. In all cases the figures have been normalized so that white is the maximum value obtained in the range displayed. Notice that the addition of the uniform distribution "distorts" the posterior probability from a relatively simple logistic shaped discrimination surface into a slightly more complex distorted Gaussian.

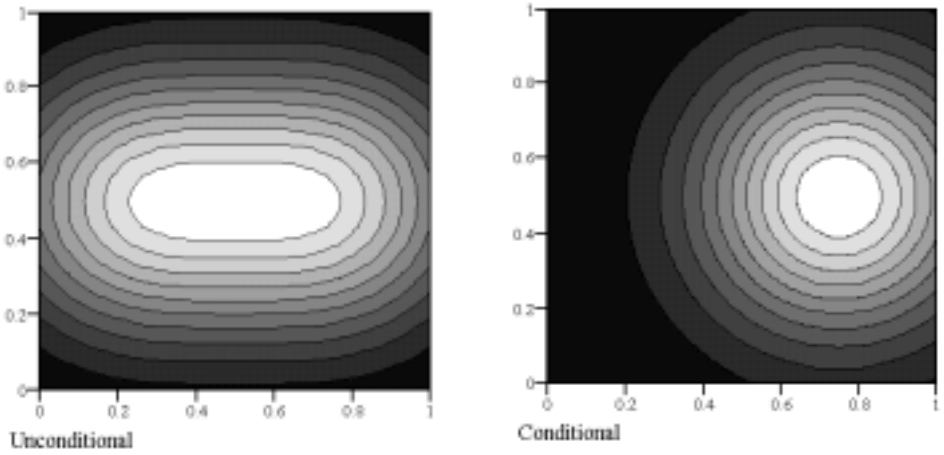


Figure 2.6.1 $P(\vec{\mathbf{x}})$ (left) and $P(\vec{\mathbf{x}} \mid k)$ (right) for k=2 have the same form for both the case where there is and where there is not a third uniform class.

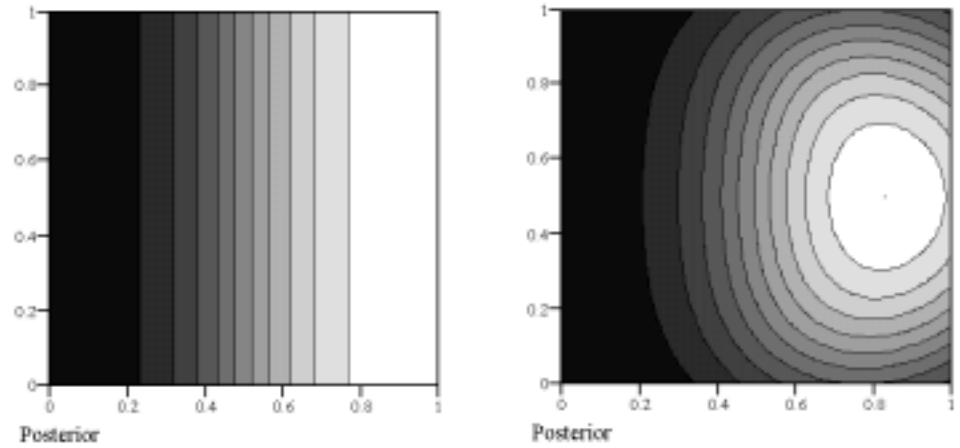


Figure 2.6.2 $P(\vec{\mathbf{x}} \mid k)$ for k=2 for the case where there is no uniform distribution (left) and the case where there is a third uniform distribution (right).

An interesting feature of modeling conditional probability densities with Gaussian distributions is that the posterior distributions (i.e. $P(k \mid \vec{\mathbf{x}})$) that are produced tend to have a somewhat sigmoidal morphology. This morphology should be contrasted with types of distributions that are characteristic of artificial neural networks constructed using logistic activation functions. As such one can see how artificial neural networks with logistic activation functions can be interpreted as posterior probability estimators. However, estimating the probability $P(k \text{ or "the class"} \mid \vec{\mathbf{x}})$ directly using a neural network within a

larger system makes it difficult to incorporate new information, i.e. information concerning the class that could turn the unconditional probability $P(k)$ in the graphical description of the mixture model into a conditional probability e.g $P(k|C)$ or the probability of k given some additional context. These reasons constitute significant drawbacks of using neural networks to construct large software systems [20]. Thus, density estimation using Gaussian distributions for the classification of continuous vectors into discrete classes offers a more principled and rigorous alternative to the use of neural networks from *both* a statistical and software engineering point of view.

2.7 A Mixture Model With Classes and Subclasses

The simple mixture models developed so far can be expanded slightly to represent an even more useful model. Consider a scheme in which $P(\bar{\mathbf{x}})$ is modeled by using a "higher-level" class variable C taking on states $c \in [1,2,\ldots,n]$, where n is the number of states or "*classes*". In this model, with each class is associated a set of "*sub-classes*" encoded as a set of states K_c of the larger sub-class variable K. In this model the union of all the states of each of the sets K_c produces the states of the variable K (i.e. $\bigcup_c K_c = K$). K_c thus takes on m_c states $k_c \in [1,2,\ldots,m_c]$ for each class. The total number of states in K is thus $m = \sum_{c \in C} m_c$ states. The probability $P(k|c)$ for a given state $k \in K$ is thus zero if $k \notin K_c$ and all the states for which $k \in K_c$ are normalized for each among themselves for each c. Thus, $P(K|C) = \pi^c_{k'}$ can be written as a *sparse matrix*. Where k' represents the subclass indices in the original mixtures. The matrix takes the following form, with zero elements of the matrix left as blank:

$$P(K \mid C) = \begin{bmatrix} \pi_1^1 & \pi_2^1 & \cdots & \pi_{m_1}^1 & & & & & & & & \\ & & & & \pi_1^2 & \pi_2^2 & \cdots & \pi_{m_2}^2 & & & & \\ & & & & & & & & \ddots & & & \\ & & & & & & & & & \pi_1^n & \pi_2^n & \cdots & \pi_{m_n}^n \end{bmatrix}$$

Eq. 2.7.1

Here the rows of the matrix represent $P(K \mid C)$ for each class c of C. The columns represent each subclass state k of K. Thus, the unconditional probability, $P(\vec{\mathbf{x}})$ can be calculated by exploiting the structure of the sparse representation in the following way

$$P(\vec{\mathbf{x}}) = \sum_{c \in C} \sum_{k_c \in K_c} P(\vec{\mathbf{x}} \mid k_c) P(k_c \mid c) P(c)$$

Eq. 2.7.2

If $P(C)$ is constructed appropriately (i.e. normalized), it is easy to see that the unconditional probability is still normalized in this expanded model, i.e.

$$\int P(\vec{\mathbf{x}}) d\vec{\mathbf{x}} = \sum_{c \in C} P(c) \sum_{k_c \in K_c} P(k_c \mid c) \int P(\vec{\mathbf{x}} \mid k_c) d\vec{\mathbf{x}} = 1$$

Eq. 2.7.3

One now has a probabilistic model in which there is a class variable that causes a distribution over a subclass variable, that in turn cause a (parameterized) distribution over the continuous variable $\vec{\mathbf{x}}$. This probability model is illustrated in Figure 2.7.1.

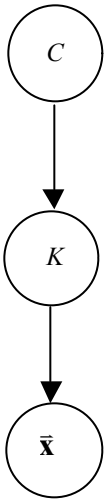


Figure 2.7.1 A Class-Subclass Mixture Model

Consider now a calculation for the posterior probability of a particular state of the class variable given an observation of $\vec{\mathbf{x}}$, $P(c^* \in C \mid \vec{\mathbf{x}} = \vec{\mathbf{x}}_{obs})$. Using Bayes' rule, the calculation is simple,

$$P\left(c^* \mid \vec{\mathbf{x}}_{obs}\right) = \frac{\left[P\left(\vec{\mathbf{x}}_{obs} \mid c^*\right)\right]P\left(c^*\right)}{P(\vec{\mathbf{x}}_{obs})}$$

$$P\left(c^* \mid \vec{\mathbf{x}}\right) = \frac{\sum_{k_{c^*} \in K_{c^*}} P\left(\vec{\mathbf{x}}_{obs} \mid k_{c^*}\right)P\left(k_{c^*} \mid c^*\right)P\left(c^*\right)}{\sum_{c \in C}\sum_{k_c \in K_c} P(\vec{\mathbf{x}}_{obs} \mid k_c)P(k_c \mid c)P(c)}$$

Eq. 2.7.4

The probability model of Figure 2.7.1 is a simple, but powerful building block for constructing complicated probability distributions over continuous variables with discrete parents using a relatively few number of parameters. This type of model will be used in a number of different probability models later in this thesis.

2.8 "Learning" or Fitting Mixture Models from Data

Mixture Models can be "learned" using the common statistical procedure of *maximizing the likelihood* of a data set consisting of values for the $\bar{\mathbf{x}}$ vector and the associated class labels. Practically, each labeled data set can be "clustered" or modeled separately and then combined as described above. If the subclass variables are given, then the parameters for Gaussian probability distributions $P(\bar{\mathbf{x}} \mid k)$ are found simply by calculating the mean and variance (or for higher dimensional vectors the variance-covariance matrix) for each subclass.

When the subclass variables are not given but the number of states m_c for each K_c are given then, the Expectation Maximization algorithm or EM algorithm [19] can be used to essentially "fill in" the missing values with their expected value and iteratively fit the model with these "hidden" variables. The algorithm will find a local minimum in the Maximum Likelihood solution space.

Interestingly, a number of authors have shown that the commonly used k-means algorithm is a special case of this form of EM clustering in the limit where the variance-covariance matrix for each cluster approaches zero [4]. The observation is insightful as it makes the underlying assumption as to the nature of the clusters explicit. When the number of states for each K_c is not given, one way to determine the appropriate number of states is to introduce a penalty term for the complexity of the model into the maximum likelihood calculation. This term is often constructed by measuring the number of bits used to encode the floating-point coefficients of the associated Gaussian probability distribution for a single subclass variable.

Finding a model is then often accomplished by searching through the penalized likelihood function by either adding additional Gaussians or starting with a large number of Gaussians and successively combining classes. This second approach is taken in [5] and is also used for Gaussian mixture clustering within this thesis.

3 Color Perception and Spectral Analysis

3.1 Key Issues

Before discussing the use of color measurements as a means to interpret or classify an image it is important to make clear just what color is. This section gives an overview of the important aspects of color science with respect to perception, clustering and the goal of extracting patterns and interpreting the most likely objects and materials comprising the subject of a given image. Then a simple probabilistic model for the objects and materials in an image is developed using the probabilistic formalism discussed in Chapter 2. The limitations of the model are discussed and as such, constitute the motivation for the following chapters of the thesis.

3.2 The Perception, Measurement and Representation of Color

This section introduces the basic concepts of color perception and discusses the common ways of characterizing color. A review of color coordinate systems is provided, as knowledge of color coordinates systems is important for three main reasons. Firstly, multi-spectral remote-sensing data often comes from narrow band "light" sensors mounted on satellites or airplanes that are not tuned to the standard RGB primaries used in image coding, based on the characteristics of color monitors. Although it is possible to gather data over a large number of narrow band channels, the correlation between channels in the range of human vision, with respect to typical objects of interest is usually so high that only a few, (in

many cases three) bands are necessary. Thus, if classification systems are constructed based on "colors" (or "spectral measurements" of light intensity within the human visual range), then if different models are to be re-used for different data sources then a coordinate transformation will be necessary. Second, when clustering colors with Gaussian distributions it is somewhat preferable to have data that is fairly Gaussian by its very nature. Choosing an appropriate coordinate system can address this problem. For example, the CIE xyY coordinate system has been shown to produce relatively Gaussian shaped distributions when just-noticeable color difference studies have been performed and analyzed in this coordinate system [75]. Third, the commonly used sub-pixel linear spectral mixing model discussed in Chapter 8 is more understandable given a simple introduction to color coordinate systems.

Color is essentially a perceptual response of the human retina to incident light having wavelength in the range of 400 nm to 700 nm. The radiance or power of an image can be expressed in terms of a spectral power distribution or SPD. The science of *colorimetry* deals with the measurement of this energy and the study of this energy's effect on the human visual system. The organization that has set many of the standards for this study is known as the Commission International de L'Eclairage or the CIE.

It is generally accepted that the human visual system has three dominant color photoreceptors known as cone cells. As discussed earlier, there is extremely high correlation between narrow band measurements of energy in the visual range from the reflectance of natural objects. This correlation in the context of evolutionary theory, somewhat explains why we have only three classes of photoreceptors. Each cone cell has a different frequency response

to *all* of the colors in the visual spectrum and the CIE has performed statistical studies to determine the form of these response curves for an average person or what is known as the *standard observer*. As there are exactly three color receptors in the human visual system, only three numerical measurements are required to encode a color. Importantly, the standard Red Green Blue (RGB) values used in most image formats do not encode all possible colors that may be observed by the human visual system. These values usually correspond to some standardized emission properties for color monitors known as RGB_{709} [35]. However, the CIE has defined a number of coordinate systems for the specification of color in which all possible visible colors can be specified. In particular, the CIE XYZ system [60] describes a color in terms of *luminance* Y and two other components, X and Z. Luminance corresponds to spectral power weighted by a sensitivity function generated by measurements of the human visual system, while the X and Z values correspond to spectral responses determined based on a statistical study of experiments on human observers. It is possible to transform from standard RGB_{709} coordinates (where the coordinates range over [0, 1]) to the CIE XYZ coordinates using the following linear matrix operation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix}$$

Eq. 3.2.1

Note that white or RGB = [1,1,1] produces $Y = 1$. Additionally, the CIE has defined a normalization process to compute "little" x and y known as *chromaticity cordinates*. Chromaticity coordinates are useful as they are a measure of "pure" color in the absence of brightness and are calculated as follows:

$$x = \frac{X}{X+Y+Z} \quad , \quad y = \frac{Y}{X+Y+Z}$$

Eq. 3.2.2, 3.2.3

Finally, *lightness* is the human perceptual response to luminance. Lightness is denoted by the CIE system using L^*. L^* can be calculate from Y and Y_n, where Y_n is known as the white reference (basically the color your eyes think is white in your perceptual field). The CIE defines L^* as:

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16; \qquad 0.008856 < \frac{Y}{Y_n}$$

Eq. 3.2.4

If, for example, a laser beam or narrow band light source were swept across the visible range of light (i.e. 400 nm to 700 nm) a curve known as the *spectral locus* would be observed in this x, y coordinate system. The sensation of purple cannot be synthesized from a single wavelength and requires a mixture of the longest and shortest wavelengths (red and blue). Thus, there exists what is known as *the line of purples* on such x, y chromaticity diagrams. The area within the spectral locus and the line of purples bounds all visible colors. Further, given any set of primary colors, the area defined by the convex hull of these colors represents all the possible colors that can be expressed using an additive mixture of those colors. Using such additive mixtures, each color component can take on values in the range [0,1]. Thus, for example given all possible combinations of the three RGB_{709} primaries one can convert these values to the CIE xyY coordinate system and the colors that can be expressed in this coordinate system trace out a triangle in the x, y plane. Figure 3.2.1 illustrates the main features of the CIE x,y plane.

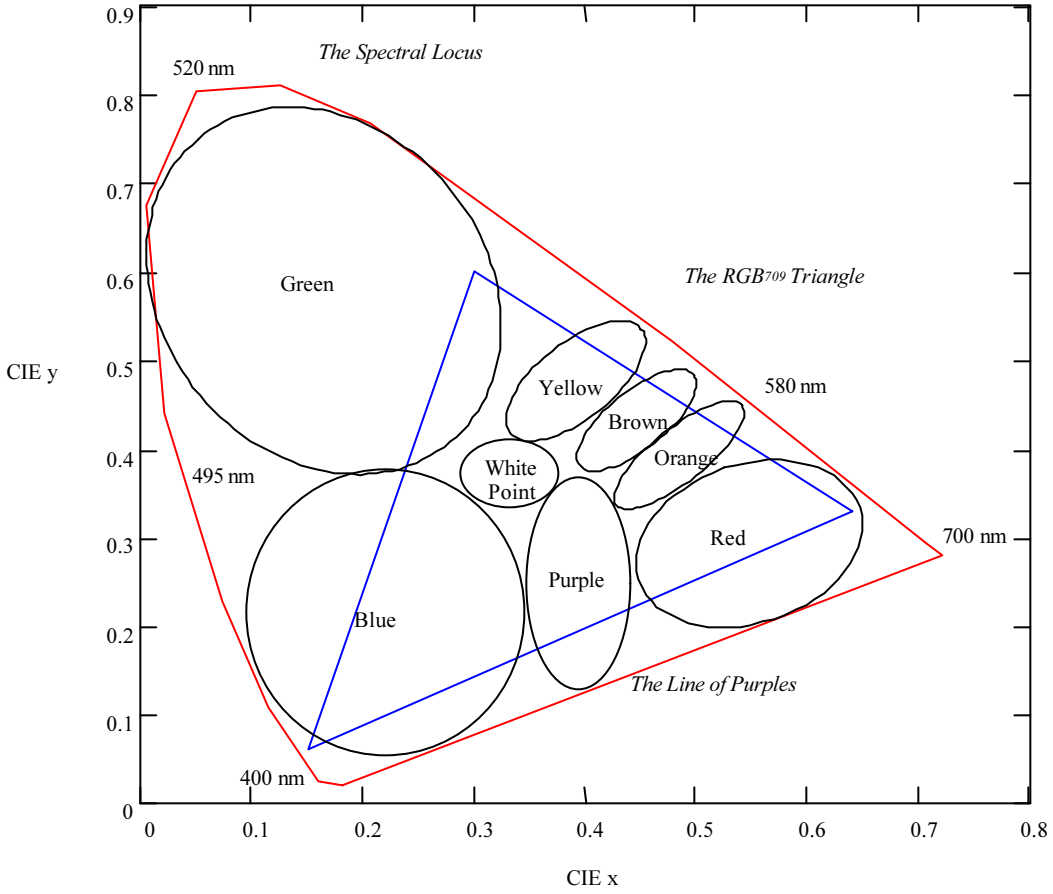


Figure 3.2.1 The CIE x,y chromaticity diagram with common colors labeled and the RGB_{709} triangle indicated

3.3 Finding Color Clusters and Constructing Probability Models

There are a number of possible ways in which we could specify and learn a probability model for the color of a pixel given the materials that are present in the given pixel. Consider that we could construct a data set using simply a number of samples from a portion of an image. As a specific example, consider a small patch of an RGB image consisting of a sidewalk, grass and the pavement of a road. The RGB components can be converted to CIE xyY and

then the x,y chromaticity coordinates for each pixel can be plotted in two dimensions. This is illustrated in Figure 3.3.2.

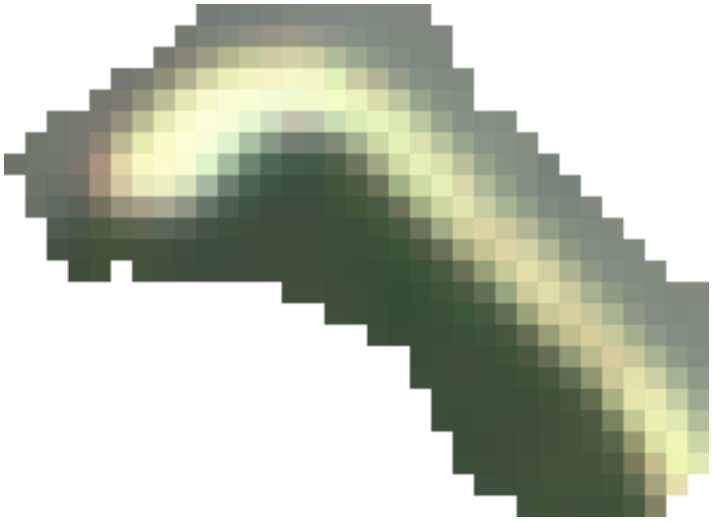


Figure 3.3.1 A portion of an aerial image consisting of road pavement (top), sidewalk (middle) and grass (bottom).

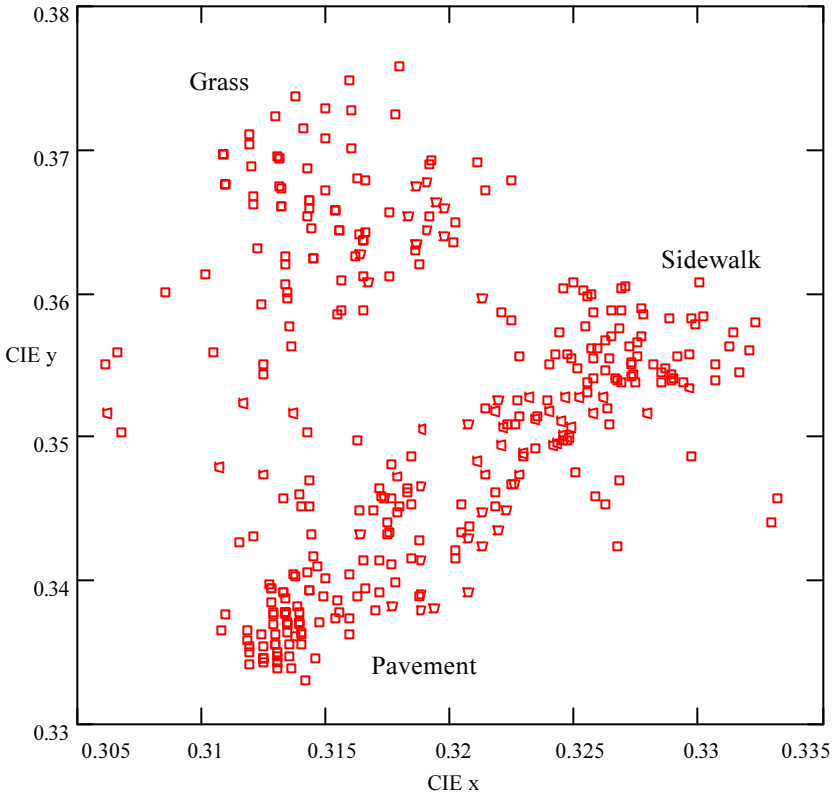


Figure 3.3.2 The CIE x,y coordinates for each of the pixels in the image.

From Figure 3.3.2 one can see three primary clusters of color that correspond to the three materials observed in the image. Now, if one presents this entire data set (CIE Y included) with no class labels to the clustering algorithm described earlier, one might expect the algorithm to produce roughly three "blobs" or Gaussians. Graphically, such a model corresponds to a variable *K* causing a probability distribution over the CIE xyY vector.

Figure **3.3.3** illustrates the contours of the unconditional probability model in the CIE x,y chromaticity diagram found by fitting a Gaussian Mixture Model using EM with a MDL penalty term. However, this model and algorithm in fact find four classes. The contour lines of the unconditional probability distribution and the ellipse for the conditional probability distribution of this fourth class are illustrated in Figure 3.3.3.

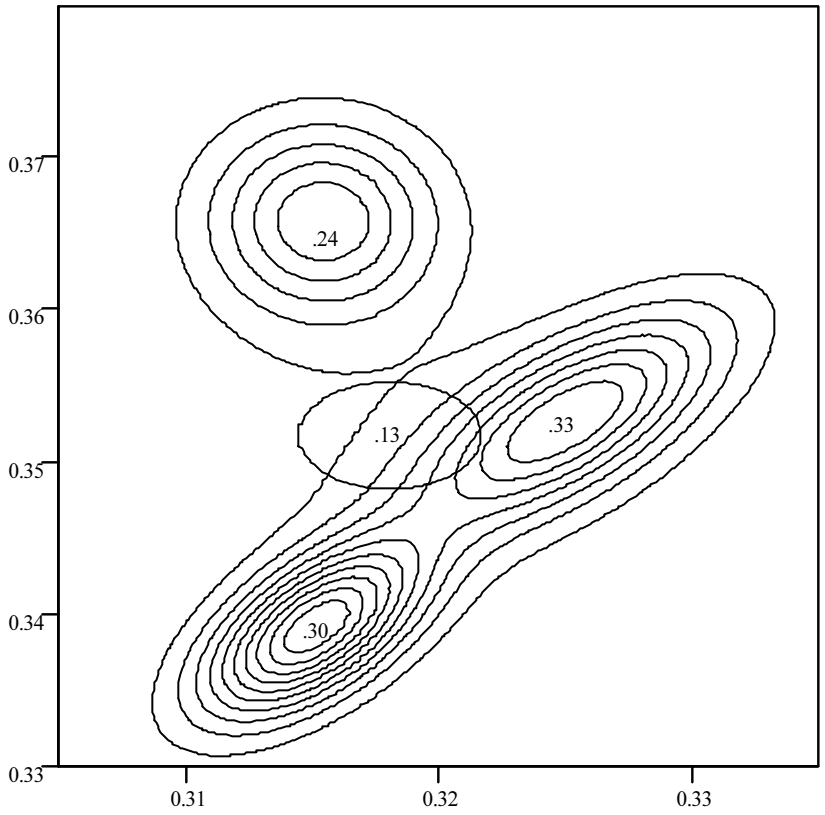


Figure 3.3.3 Contour plot for a Mixture Model of $P(\bar{\mathbf{x}})$, where the central ellipse illustrates $P(\bar{\mathbf{x}} \mid k = 4)$

The contour map found for the unconditional probability is centered on three clusters with some distortion due to the fourth cluster. The unconditional probabilities for each cluster (i.e. $P(k)$) are illustrated near the mean of the respective clusters. From Figure 3.3.3 one can visualize the true underlying process that gives rise to the observed color values for the image. The fourth class that was found using our clustering algorithm provides a good starting point for discussing these processes.

The fourth class found using this algorithm is modeling two main effects. First, the fourth class models a relatively large number of less dominant, potential materials that may be present in some quantity within a given pixel. These less dominant pixels could be considered as "noise". Secondly, this fourth class models a pixel mixing effect at the boundaries of the objects (i.e. field, sidewalk and road) in the image (along with any mixing that might occur between neighboring sensors when the image is captured by the associated electronics). The three higher probability classes do correspond roughly to the materials grass, pavement and concrete. However the blobs are shifted slightly from the "true" means and squashed from the "true" variance-covariance (i.e. for a data set consisting of only "pure" examples of a given material). This is due to the sub-pixel boundary and sensor mixing effects. If one wishes to use a clustering algorithm to find a set of "pure" materials that are responsible for the observed colors in an image, these mixing effects should be captured.

One way to "sharpen" up the clusters that are the pure classes and allow the differentiation between relatively pure classes and mixed classes, is to incorporate gradient information into

the vector being clustered. However, the standard gradient calculation is not appropriate in this situation, as we wish to estimate the degree to which all the neighboring pixels differ from the current pixel. For this reason the 3x3-pixel *edge filters* optimized to minimize orientation bias proposed by Wilson [73] and described in further detail in Section 6.2 offer an attractive filtering technique for measuring edges or "delta" CIE xyY information. The result of applying our model to the CIE xyY vectors plus the magnitude and phase of the Wilson edge filters is illustrated in Figure 3.3.4.

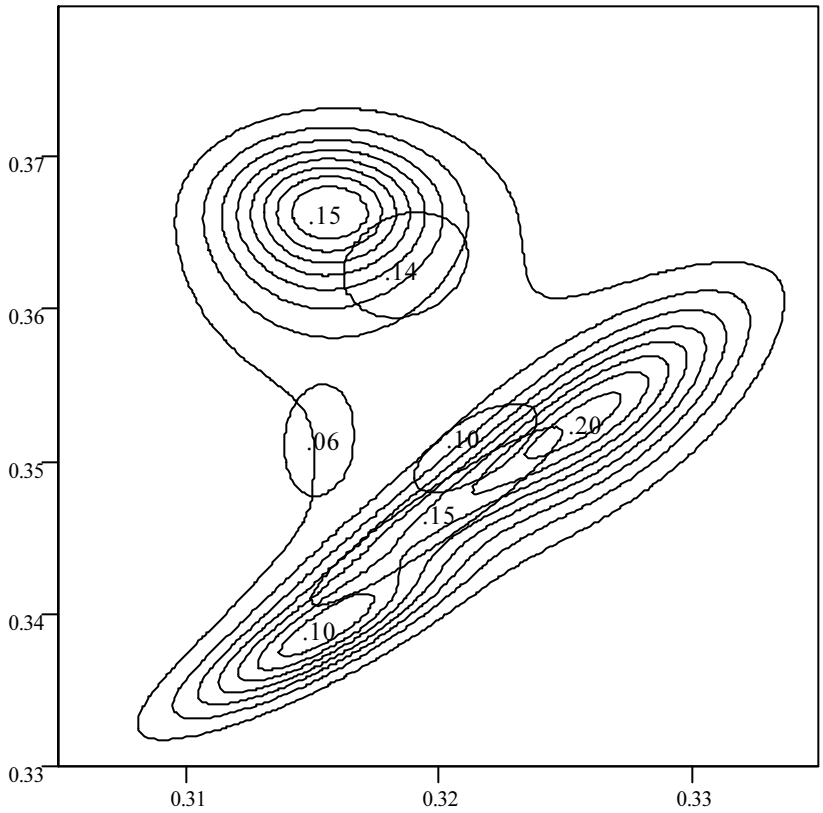


Figure 3.3.4 A Contour plot for $P(\bar{\mathbf{x}})$ with ellipses for the conditional distributions for the less dominant (due to higher variance), "mixture" classes also illustrated.

Notice that the three dominant classes found using the edge information are “sharper” or are of less variance as illustrated by the relative density of the contour lines with respect to the color only density model.

To mitigate this mixing effect at object boundaries, one can construct a model in which each material and a set of *boundary* classes are clustered independently using a labeled data set for each class. For practical purposes, an image can be labeled so that the boundaries of objects are explicitly specified. The pixels of an image can be labeled using a color palette and specified on a second layer of an image, using standard drawing software. This task is somewhat simplified as most drawing packages allow users to draw a line based outline in one color and then fill in the enclosed shape with a different color.

Additionally, in some cases mixing may occur elsewhere than at the boundaries of objects. For example, trees at one meter per pixel resolution will produce a non-boundary mixing effect or a mixture of green grass and burnt grass might occur within the boundary of the object “grass field”. Thus where necessary, *mixture classes* not arising from object boundaries can be given explicit labels and not left to be found by the underlying clustering algorithm. By labeling mixed material classes and relatively pure material classes one can construct a much more accurate model of the true physical effect. An example of such a labeling is given in Figure 3.3.5 for a portion of the image of the urban area within a loop of the road. The original image has been enhanced using an L^* power function applied to each color coordinate and unlabeled pixels in the second image are shown as white.

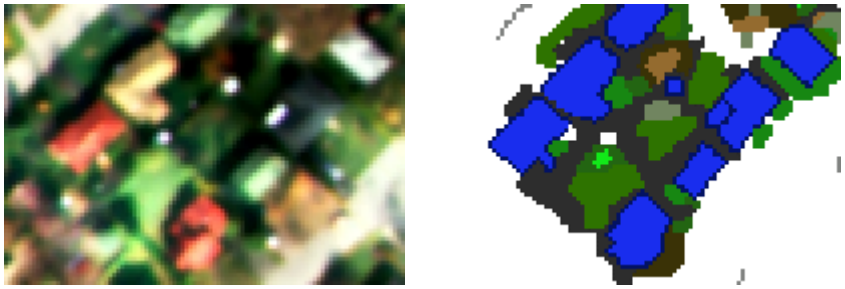


Figure 3.3.5 An image and a corresponding pixel material class labeling.

Once the sets of pixel level material classes have been defined, (possibly including boundary classes mixed material classes and pure material classes) data from a labeled image can be independently clustered for each class. The probability density models may then be combined into a larger model. The combined model can take the form of the class-subclass model developed in Chapter 2. The model is illustrated graphically in Figure 3.3.6.

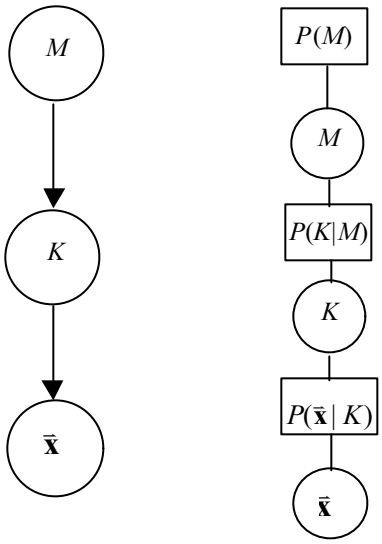


Figure 3.3.6 Left: A Class-Subclass Mixture Model for Image Materials M causing a set of Gaussian cluster sub-classes *K* over CIE xyY and delta xyY vector space $\vec{\mathbf{x}}$. Right: The equivalent factor graph.

Figure 3.3.7 illustrates the result of a classification using the class-subclass mixture model applied to a labeling of 13% of the urban image. The labeling used boundary classes for

houses and mixture classes for trees. The image on the right is the most likely class using a model with no derivative information while the image on the right does contain derivative information. Both models use equal priors (unconditional probabilities) on the highest-level material class variable.

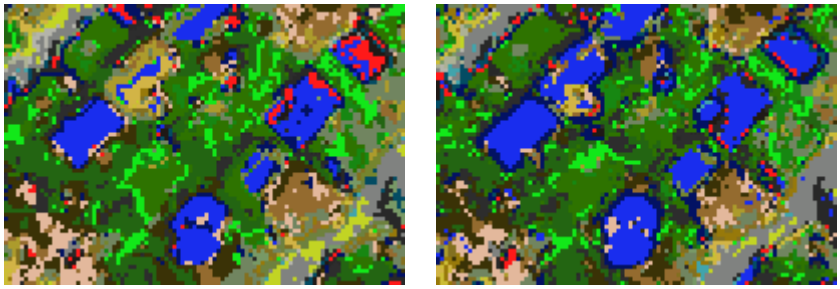


Figure 3.3.7 Left: The most likely class for each pixel based only on color. Right: The most likely class with edge filter information included.

The classification with the derivative information allows a slightly more accurate classification, the effect is made apparent by the relatively homogeneity of the house roofs on the right image. However, for both of these classifiers the image consisting of the most likely classification for each pixel looks "spotted" or "noisy". There are two main explanations for these effects. First, there is a great deal of intermixing of the underlying materials that comprise the image. The combination of materials mixed at the sub-pixel level can produce an extremely large gamut or set of potential observed colors in an image. Second, materials with different reflective profiles can in principal produce the identical "color" and lightness measurements. For example, a green car could produce the same observed color as a green tree even though the underlying spectral characteristics are slightly different in the unobserved "spectral bands". Further, when only intensity information is available it becomes extremely difficult to classify an image based on only pixel level measurements. To address these issues, the next section discusses some common ways of

quantifying and incorporating “context” into a pixel level classification procedure and an extended probabilistic model for the classification of pixels is proposed and examined.

4 Incorporating Localized "Context" to Classify Pixels

4.1 Overview

Numerous strategies have been proposed to deal with the issues of the local context or the task of reasoning about relationships between features in Computer Vision. One way to deal with the local context of pixels is to classify groups or small squares of pixels together. However, the choice of the size of the groups is extremely important. If one simply breaks up the image into a grid then one can effectively classify a coarser scale version of the image. But what about the relationships between the coarser scale blocks? The same context information among blocks is missing. Further, what if a pixel level classification is desired? Thus, a strategy must be devised to encode contextual constraints while allowing a pixel level classification to be performed.

A Markov Random Field (MRF) is a probabilistic model can be used to encode contextual constraints while allowing a pixel level classification to be performed. MRFs are commonly used in the Remote Sensing community, the Artificial Neural Network research community and the Image Processing research community. A special from of a MRF known as Gibbs Markov Random Field is particularly popular in these research communities. However, in these research communities this form of a MRF draws heavily from Statistical Physics and accordingly, the analogy commonly used by researchers in these areas involves treating images as lattice systems and energies are discussed as opposed probabilities. This analogy is contrasted with the bayesian interpretation used for the underlying formalism in this thesis. Section 4.2 introduces the Gibbs-MRF. Then in Section 4.3 logical morphological filters are

briefly discussed. Then in Section 4.4, a probability model is developed to encode spatially localized pixel relationships or the local contextual constraints of pixel classification states.

4.2 Gibbs-Markov Random Field Models

Markov Random Field Models or MRFs are commonly used in Remote Sensing and Computer Vision for such applications as image restoration [25], segmentation [42] and texture characterization [15]. Image restoration involves the detection and correction of noise in an image. While, segmentation in this context usually involves finding homogeneous regions of texture in an image. The basic idea behind MRF modeling involves constructing a parameterized probabilistic model consisting of only the local relationships between pixel grayscale levels or discretized pixel states. When combined together, relationships between pixels interact and provide a probability model for the entire image. When such models are constructed to characterize grayscale relationships, texture patterns are often associated with the information encoded in the model. When such models are constructed for discretized states of pixels, the relationships between a given pixel and its neighbors are encoded probabilistically in such a way that some relationships are more likely than other relationships. If these states correspond to discretized grayscale values then textures are a natural interpretation for the encoded information. If the states are classification states of groups of pixels, then the information can be interpreted as relationships between feature classes analogous to a probabilistic version of the "rules" used in an expert system.

The widely referenced paper by Geman and Geman [25] seems to have sparked the more widespread use of the MRFs with Gibbs distributions in image modeling research and also

influenced the commonly used terminology. This paper drew together many ideas rooted in Statistical Physics (MRFs originated with Ising's 1925 thesis on ferromagnetism [34]), used a parameterized form for a MRF known as the Gibbs distribution and showed how they could be applied to image restoration Further, this paper introduced the sampling procedure for investigating MRFs by generating realizations of the model, known as *Gibbs Sampling* (a specialized version of the metropolis algorithm [50]). Accordingly, it seems that much of the terminology and mathematical formulation found in the literature for these models draws heavily from this lattice system analogy even in cases where the model is somewhat inappropriate. These methods are also quite closely related to so-called Maximum Entropy Methods or MEMs where again this statistical physics analogy arises. Not surprisingly, similar questions arise as to just what the associated entropy and energy terms used in these techniques represent. More modern research has begun to interpret these techniques more intuitively as being bayesian inference methods [59]. The probabilistic model developed later in this thesis takes a form very similar to the standard MRF model. However, in this thesis investigation probabilistic message passing is used to perform inference in the model as opposed to a Monte Carlo Methodology. The probability propagation scheme also emphasizes the clearly more realistic bayesian analogy. A commonly used methodology for image modeling is presented below.

First, a *neighborhood system* η is defined indicating the local interaction of pixel sites (often taken as the closest neighbors but more complex systems are possible). Cliques are found from the definition of the neighborhood system as all the sets of pixels for which each pixel in the clique is a neighbor of each other pixel in the clique when the neighborhood is applied

to the whole image. With each clique $q \in Q$ (the set of all cliques) is associated a *potential function* that is a parameterization of the variables in the clique usually denoted as $V_q(\mathbf{x}_q)$. Already the analogy with the junction tree algorithm discussed earlier becomes apparent, but at this point the Statistical Physics terminology comes into play. In the commonly used terminology, the sum of all the potential functions is known as the *energy function*. The energy function is used as the argument to a simple exponential calculation that is appropriately normalized to produce a parameterized form for the probability of the given *lattice point* (pixel in the image) being in a particular state. The probability model can then be written using the form of the well-known Gibbs distribution from Statistical Physics. Accordingly, such models are commonly referred to as Gibbs Markov Random fields or GMRFs. The probability model for *all the pixel sites* (i.e. the joint probability) of the image X_G, given the neighborhood system or set of variables η then takes the following form.

$$P(X_G) = \frac{1}{Z} exp\left(-\frac{1}{T} U(X_G)\right)$$

Eq. 4.2.1

Where T represents the temperature in a Lattice System but is usually taken as a constant for image modeling. $U(X_G)$ is discussed as the *energy function* and is decomposed into a summation over the set of all *clique potential functions* $V_q(X_q)$ associated with each clique $q \in Q$, the set of all cliques on the lattice. The decomposition can thus be written as:

$$U(X_G) = \sum_{q \in Q} V_q(X_q)$$

Eq. 4.2.2

Note that these functions $V_q(X_q)$ depend only on the states of the pixels in the clique, X_q and their form is not subject to any additional constraint. The specification of the form of these

functions is thus what determines the nature of the probabilistic model. These functions are often set heuristically at "design time". Z is known as the partition function and is intuitively understood as a function that normalizes the distribution. Z is found from:

$$Z = \sum_{\text{all x}} exp\left(-\frac{1}{T}U(X_G)\right)$$

Eq. 4.2.3

One can thus summarize the process of constructing such a model in four steps:

1. Specify the neighborhood geometry
2. Determine the cliques in the graph
3. Specify the potential functions for each clique
4. Form P(**x**) from the Gibbs distribution

Figure 4.2.1 illustrates three commonly used neighborhood systems. While Figure 4.2.2 illustrates the cliques associated with the first order and second order neighborhood systems. Notice that there is a so-called self-interaction effect due to the fact that each variable is in a clique with itself.

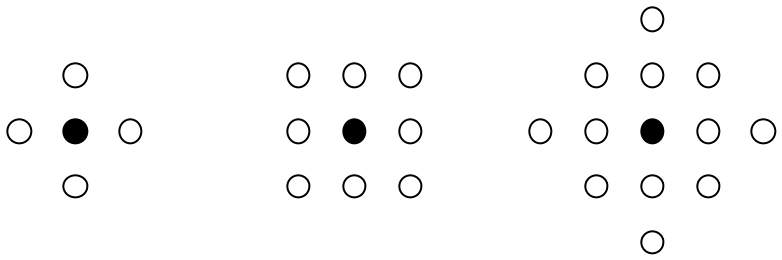


Figure 4.2.1 MRF Neighborhood systems of order 1, 2, 4.

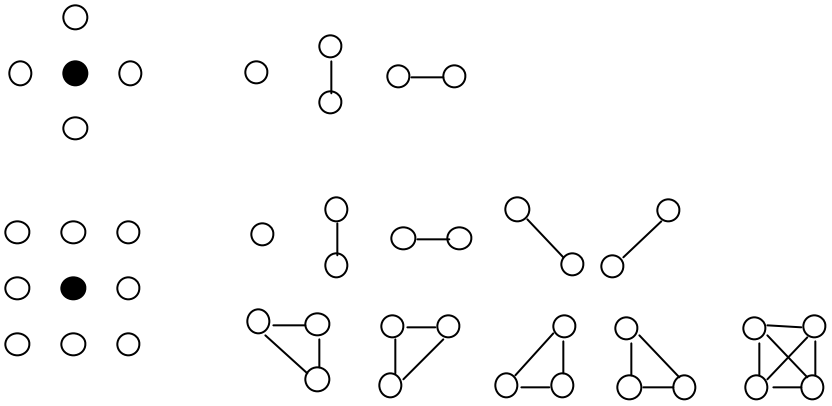


Figure 4.2.2 MRF cliques for order one (top) and two (bottom) neighborhood systems.

4.3 Morphological Filters

Image processing algorithms developed for industrial vision problems are typically computationally very simple and thus inexpensive, as the algorithms are used for real-world systems that often run in "real-time" or are used for very large images. A fast and simple strategy that is often used to "clean up" an initial color based segmentation involves the use of logically defined *morphological filters*. Consider that one could generate a series of binary image planes consisting of a classification for a given pixel using some initial algorithm based on pixel color alone. Often, the technique of applying a simple bounding box in RGB space is used to derive such classification planes. A filter can then be defined so that a pixel classified as a particular class is removed unless at least one of its neighbors has also been classified as belonging to that same class. Applying such a filter tends to "erode" a binary image, removing pixels that are not connected to other pixels. Such operations are used for generating the skeleton of an object by successively applying the filter. Now, consider a second type of filter. Here, when a given pixel is a negative classification but more than half of the other surrounding pixels positively classified, change the centre pixel to a positive

classification. Such a filter tends to have a "growing" effect on the image. The successive application of an erosion filter over a few iterations followed by a growing filter will tend to remove small clusters of positively classified pixels while also "filling-in" any negatively classified holes in larger regions classified as being positive. The model developed in the next section has a similar effect on images. However, this model is treated in a probabilistic framework and thus is not restricted to binary images. Further, the relationships between pixel states are "learned" and not pre-specified.

4.4 A Probability Model for Spatially Localized Relationships

In this section it is shown how a model for relationships between image features can be encoded using the graph formalism of Chapter 2. The motivation and intuition for constructing such a model is to find a principled and systematic way to correct the initial hypotheses generated by a probabilistic pixel level classification. The resulting procedure could be thought of as a form of hypothesis refinement. The similarities of the model with respect to MRF techniques and more simplistic morphological filters will be apparent. However, in contrast to the more common relaxation techniques (e.g. Gibbs sampling), iterative probability propagation is used to perform inference in the model. The message passing procedure also emphasizes the Bayesian interpretation involving inference in a graphical probability model as opposed to the lattice system and associated "simulated annealing" analogy that is often associated with relaxation procedures. In this way one can think of modeling the "higher level" knowledge of relationships between classification states as opposed to the coupling energies of lattice points.

Consider that for a given pixel location, using a probability model based on color alone we can infer the material class of that particular pixel. Looking at the neighboring pixels one can then assess the local context of that pixel. Intuitively, one might encode this information in terms of the probability of a neighboring pixel being of a particular material class *given* the centre pixel's class. This type of model illustrated in Figure 4.4.1, using the directed arrow notation. Here, the each pixel is classified as being of a particular *material type*, denoted by $M_{i,j}$ where the x,y positions of the pixel in the image are indicated by i,j. Recall that each pixel has associated a corresponding Gaussian probability model for color, but this model sub-structure has not been shown for clarity.

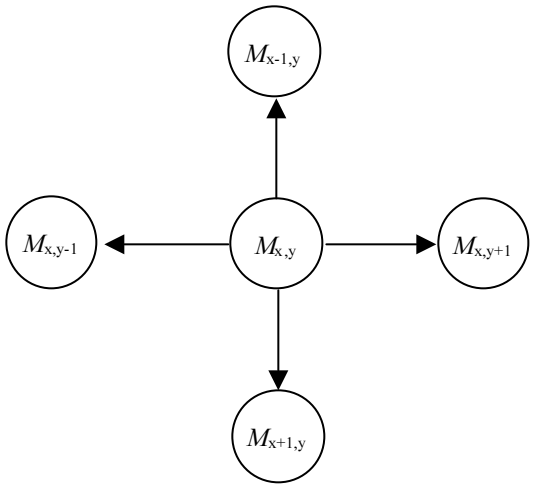


Figure 4.4.1 A probability model for classifying a pixel at some location x,y in an image as a material, $M_{x,y}$ in the context of neighboring pixels $M_{i,j}$.

Here we begin to see the problems associated with trying to model these relationships using the directed arrow notation. The DAG notation and the associated probability tables are often associated with the notion of one variable causing an effect on another variable. However, in many cases one does not think of a real cause and effect relationship existing between classification states of neighboring pixels. For example, one does not attribute a causal

relationship between two green pixels both arising from a grassed area, but one might associate a causal relationship between a tree colored pixel and a neighboring shadow colored pixel.

To construct a general model for the image, relationships between pixels can be more accurately modeled as correspondence rather than cause and effect relationships. Such a model can be constructed using a MRF. Figure 4.4.2 illustrates a MRF for the classified state of a pixel in the context of the states of the surrounding pixels in a first order neighborhood system using the factor graph notation. The previously developed model for pixel color is then incorporated into this model as illustrated in Figure 4.4.3.

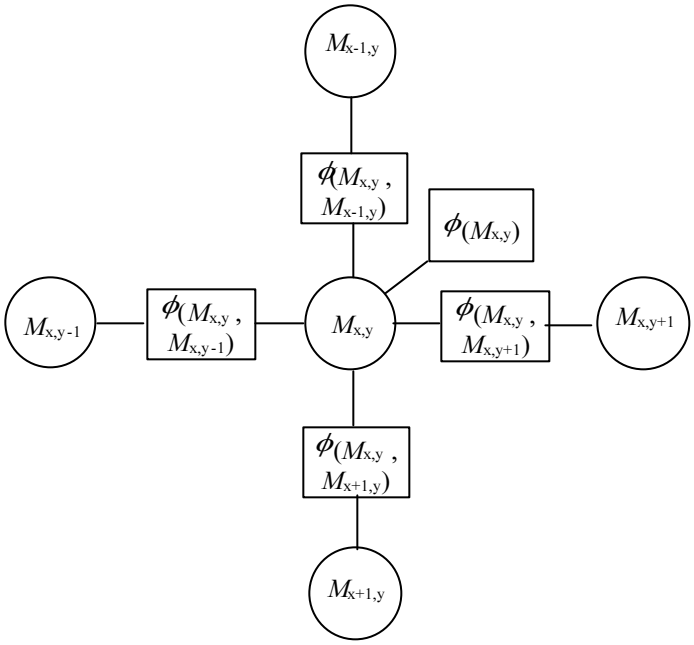


Figure 4.4.2 A probability model for the local context of a pixel.

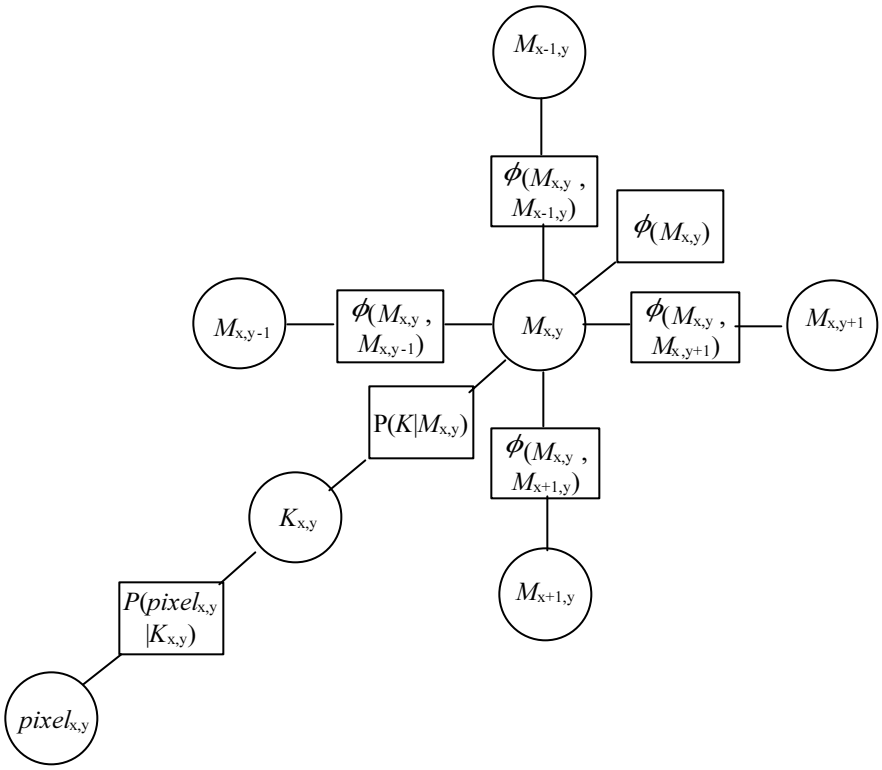


Figure 4.4.3 A probability model for the classification of pixels into materials incorporating pixel color and the local context of variables.

If one ensures that there is horizontal and vertical symmetry (i.e. all cliques of horizontal relationships are equal and all cliques of vertical relationships are equal), then this localized probability model for each pixel and its neighbors can be combined into a consistent probability model for the entire image. The result of constructing such a probability model over the entire image using this locally developed model produces a lattice. In the lattice, each pixel is connected to its neighbor through a potential function of it and the corresponding neighbor. A segment of the complete model for an image using the factor graph notation is illustrated graphically in Figure 4.4.4. The substructure relating each material variable to the corresponding pixel color measurements has been removed for clarity.

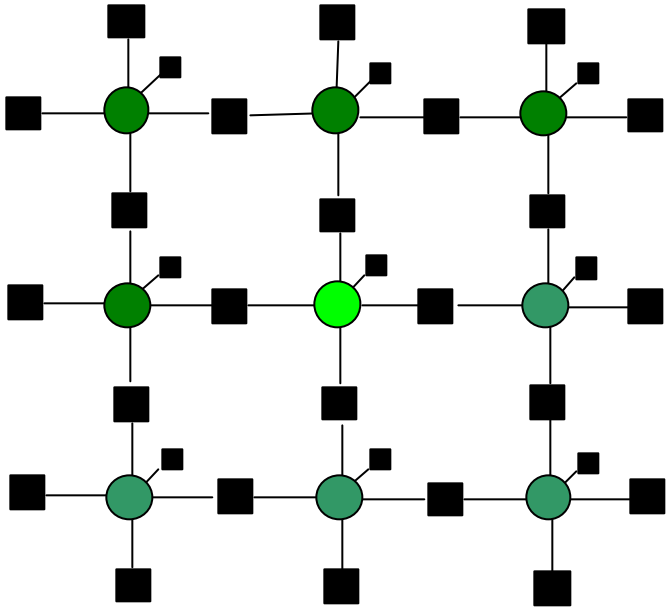


Figure 4.4.4 A factor graph a MRF model for an image (substructure is removed for clarity). Different colors have been used for material variables to illustrate the currently most probable material and probability tables are illustrated with black rectangles. Figure 4.4.5 illustrates the complete model using the factor graph notation.

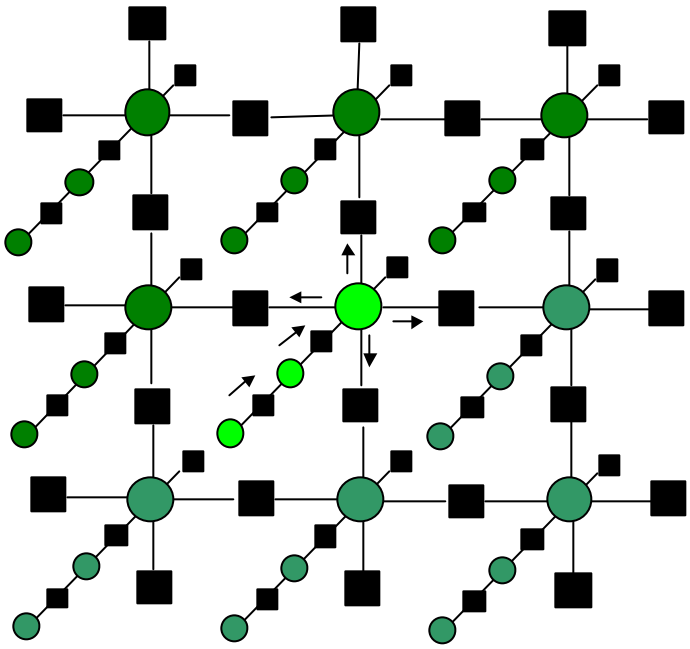


Figure 4.4.5 The larger probability model illustrated as a factor graph, arrows indicate messages passed from variables to their neighbors in the graph.

For this investigation, the probabilistic message passing procedures discussed in Chapter 2 are used to update the probabilities for each variable in the graph given its neighbors. For simplicity the potential functions are set to be equal to the joint probability tables of the variables in the function. These probability tables relating variables are "learned" from a labeled data set. For this investigation Dirichlet priors are used for learning the tables. In this context, probability messages are *sent* from each pixel to each of its neighbors as illustrated by the arrows in Figure 4.4.5. Thus, each pixel level material variable also *receives* a message from each of its neighbors, except at the boundary. Pixel material variables communicate by sending messages to their neighbors through their common cliques. Of significant note is the fact that these messages can be computed in parallel on appropriate computational hardware. The probabilistic message passing procedure differs from the more commonly used stochastic relaxation techniques in that no sampling from probability distributions takes place. A more detailed explanation of this message passing procedure and the relationship between this model and the more common Gibbs-MRF is described in Appendix I.

The model that has been developed above represents a way to encode information concerning the types of states neighboring pixels tend to be in together. The information can be used to refine an initial pixel or feature level hypothesis based on information such as color or lightness. As such, this procedure could be thought of as a filter. For example, consider the situation where after color analysis the most likely material class of the centre pixel in Figure 4.4.4 is "car material" (bright green) and the most likely class of the other surrounding pixels are "green tree" (light green) or "shadowed green tree" (dark green). Intuitively, such a

probability model will tend to reduce the probability of the centre pixel being “car material” and increase the probability of the centre pixel being either “green tree” or “shadowed green tree”. This is the case as it is most likely that within a training set there are very few instances of green cars mostly obscured (except for one pixel) by green tree. A concrete example of applying this model to the color image of an urban area is illustrated in Figure 4.4.6.

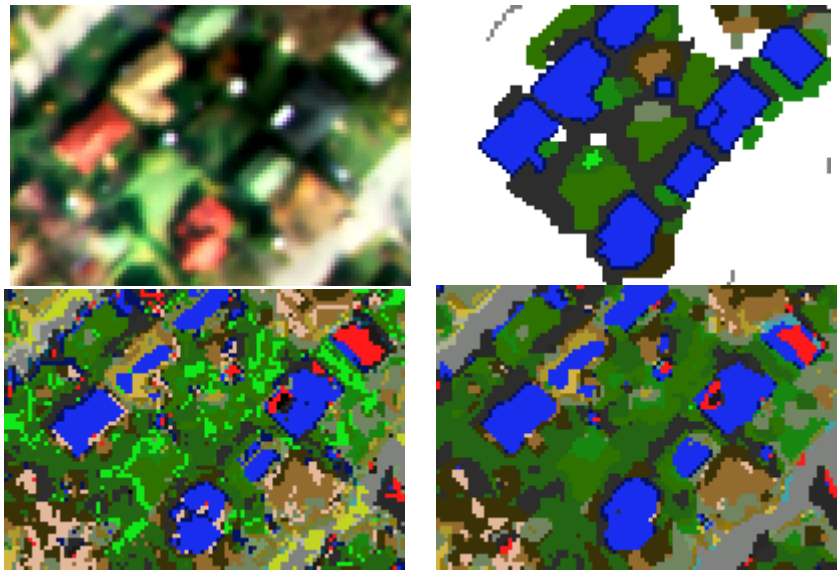


Figure 4.4.6 Top Left: The original image, Top Right: The labeled image, Bottom Left: Most likely class using only color, Bottom Right: A single iteration of message passing.

This model of local pixel context is useful for segmenting an image into homogeneous regions. The probability tables relating pixel classes to their local neighbors encodes localized contextual relationships in a distributed manner and will refine the “initial hypothesis” from the pixel level analysis. However, if only grayscale information is available for pixel level measurements then the level of ambiguity from the initial analysis will be extremely high leading to relatively unrealistic segmentations. This is illustrated in Figure 4.4.7.

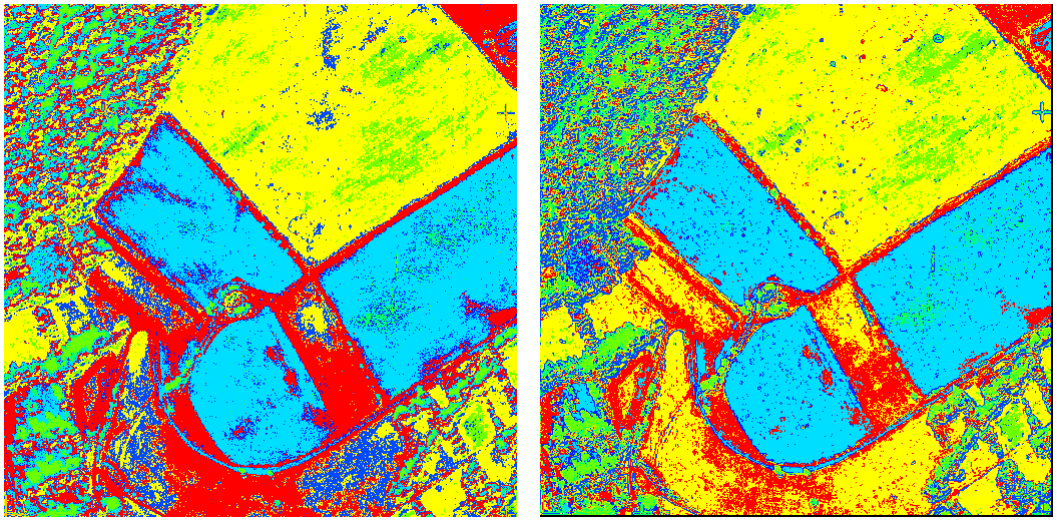


Figure 4.4.7 Left: A pixel-level Greyscale classification of the farmland image. Right: a model incorporating pixel color plus derivative information.

Figure 4.4.7 confirms what one might have expected, namely that the context of pixels is essential for constructing a classifier for greyscale images. The derivative information is helpful, but there is still a very high level of ambiguity. Further, simple concepts such as the notion that trees are at least a certain size and have specific textures or that forests consist of groups of trees are not modeled with such a low-level approach.

To address this problem, the model can be expanded to operate over relatively large groups of pixels. Thus the probability relationships between these groups of pixels will encode relationships between feature classes at a higher level of abstraction. The use of MRFs for encoding relationships between "higher-level" features has been investigated by other researchers but less common than pixel level models [61]. The selection of the appropriate size and configuration of these groups of pixels can be extremely important. However, in a simplified model one can break the image up into a pattern of repeating blocks or overlapping "windows". Then, relationships between these windows can be encoded using

the model that has just been developed for encoding pixel level relationships. Such a model will effectively allow a coarser scale version of the image to be segmented and classified in a manner similar to the pixel level procedure. The size of these windows and the resolution of the image will determine the abstract classification states or interpretations of these blocks of pixels. The next section discusses the construction of such a model.

5 Classifying Larger Groups of Pixels

There are two very important issues involved with the classification of relatively large groups of pixels within an even larger image. First, one must choose the size and structure of the groups. Second, one must ensure that an accurate probability model can be constructed for the relatively high dimensional pattern. When color was discussed earlier it was shown that the CIE xyY linear transformations of standard RGB values was a relatively good starting point for pattern classification. Similarly, an appropriately chosen initial change of basis can be useful for detecting patterns within groups of pixels. The next few sections discuss some past approaches to the problem of selecting an initial basis and the related issues involved with selecting appropriate groups of pixels to begin detecting patterns. Then a scheme is presented allowing ideas developed from the previous section to be incorporated into a classification and segmentation procedure.

5.1 Fourier Methods

Research in the area of Applied Mathematics has produced some very popular techniques for the analysis of images involving the construction of filters encoded as matrix transformations. Most of the research on the application of these methods by applied mathematicians is focused on greyscale images where the images are treated as matrices and the analysis is treated as an exercise in matrix algebra. Fourier analysis is usually achieved using the Fast Fourier Transform (FFT). The FFT reduces the matrix form of the discrete Fourier Transform computation in one dimension into an $O(N\log_2 N)$ from an $O(N^2)$ computation. Further, of significant note is the observation that FFT algorithms can be

viewed using our probabilistic formalism as instances of probabilistic message passing algorithms as described in [41]. The transform decomposes an image into an orthogonal set of basis functions consisting of sine and cosine functions resulting in a set of two-dimensional global frequency filters. The computation for one-dimensional signals is often written as:

$$H_n = \sum_{k=0}^{N-1} W^{nk} h_k, W \equiv e^{\frac{2\pi i}{N}}$$

Eq. 5.1.1

H_n represents the transformed signal and the input signal is represented by h_k. Further, the original continuous version of the Fourier transform involves an integral over all space (i.e. -∞ to ∞) and the matrix version of the transformation is both discretized and limited to finite extent. As the matrix form of the computation is a finite version of the original transformation, a rectangular *window function* is *implicitly* applied to the underlying signal. Thus, the resulting transformation always contains convolutional effects in the frequency domain signal. This problem resulted in a great deal of research into the time-frequency tradeoffs involved with various non-rectangular window functions. The transformation and the computational properties of recursive computations of the transform are interesting from a mathematical point of view. However, when viewed in terms of filter theory, if the basic transformation as defined is applied to an entire image, frequency filters global over the entire image are produced. When applied to large images, this limits its utility for detecting spatially localized patterns in complex scenes. Thus, for practical applications, the transform can be applied to smaller blocks of the image, 32x32 pixels for example. A Fourier transform can then be applied to each of the square blocks independently, possibly with a non-rectangular window function. The coarse scale version of image can then be classified for

each square based on the coefficients from the transformation for each square. The technique thus produces a classification of a coarser scale version of the image. If a finer scale classification is desired, then a "sliding window" approach can be used at a greater computational expense as the windowed Fourier transform must be recomputed covering part of area that has already been analyzed. This type of analysis is commonly used in one dimension as a first step in speech recognition from audio signals. A good description of feature extraction from audio using these types of sliding window techniques is given in my United States Patent [56].

5.2 Image Pyramids

Image pyramids were extremely popular in early computer vision research as they offered a means of constructing computationally efficient, spatially localized filters responsive to patterns over various scales. Typically, the image is smoothed or blurred using a smoothing filter and a then an image of one half the size is created by re-sampling the smoothed image. Then, the image is subjected to a second filter. This second filter often takes the form of either some form of frequency filter or some form of derivative-like calculation estimating discontinuities or edges over a larger area than is typically associated with exact numerical derivative calculations. Such techniques are known as pyramid calculations as when illustrating the filtering and re-sampling process the successively smaller images produce a pyramid when stacked one on top of one another. Such a pyramid diagram is shown in Figure 5.2.1.

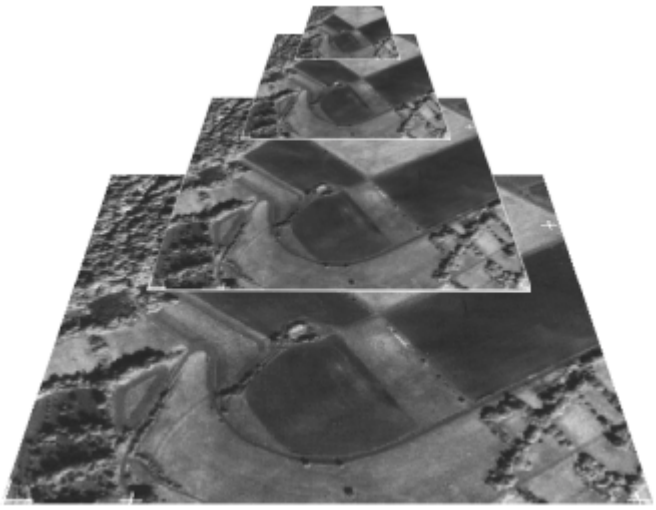


Figure 5.2.1 A pyramid for the farmland image.

The blurring process may seem counterintuitive to the goal of extracting patterns. However, if one considers the task of looking for an object or generating hypotheses concerning objects within a very large image these techniques seem less strange. The blurring and re-sampling process is analogous to the act of physically taking a step back from an image and can be thought of as a model of the physical effect of light interacting on a light sensor (e.g. the human eye). Alternatively, such techniques can be considered as a form of dimensionality reduction with respect to scale. For example, if one wishes to classify and segment forests, it is much easier to construct probability models based on small images or templates of trees in a coarser image rather than on images of the same size in the original image consisting of single high resolution leaves. *One easily sees the forest from the trees, but it takes too long when you are looking at the detail of the leaves.*

5.3 Orthogonal Wavelet Transformations

Wavelet transformations are the more modern way of viewing these filtering operations. However, they also include the set of pyramid filters that also correspond to orthogonal linear matrix transforms. Like the Fourier Transform they can be computed recursively resulting in the so-called Fast Discrete Wavelet Transform or FDWT [59]. However, here a sparse matrix transform is constructed from a set of convolutional filter coefficients consisting of a set of the smoothing coefficients and a set of wavelet coefficients. The result is that the image is wavelet filtered and then smoothed successively. Practically, the complete FDWT is accomplished by applying the matrix transform permuting the result and then applying the transformation again on one half of the result (corresponding to a smoothed version of the image). The matrix below illustrates the structure of the 4^{th} order Daubechies wavelet transformation matrix. The inverse transform is just the transpose of the matrix (i.e. it is orthogonal). The first row represents the smooth coefficients and the second row represents the wavelet coefficients. The pattern is repeated but shifted over two columns to create the complete matrix structure for a one-dimensional transform.

$$\begin{bmatrix}
c_0 & c_1 & c_2 & c_3 & & & & & & & & \\
c_3 & -c_2 & c_1 & -c_0 & & & & & & & & \\
 & & c_0 & c_1 & c_2 & c_3 & & & & & & \\
 & & c_3 & -c_2 & c_1 & -c_0 & & & & & & \\
\vdots & \vdots & & & & & \ddots & & & & & \\
 & & & & & & & c_0 & c_1 & c_2 & c_3 & \\
 & & & & & & & c_3 & -c_2 & c_1 & -c_0 & \\
c_2 & c_3 & & & & & & & & c_0 & c_1 & \\
c_1 & -c_0 & & & & & & & & c_3 & -c_2 &
\end{bmatrix}$$

Eq. 5.3.1

The result can be thought of as a set of spatially localized frequency filters of increasing scale resembling windowed Fourier filters. Figure 5.3.1 contrasts a 4^{th} order Daubechies wavelet with a Fourier Filter of similar "frequency". Both images were synthesized by applying inverse transformations to matrices of all zeros except for a single one in the transformation domain.

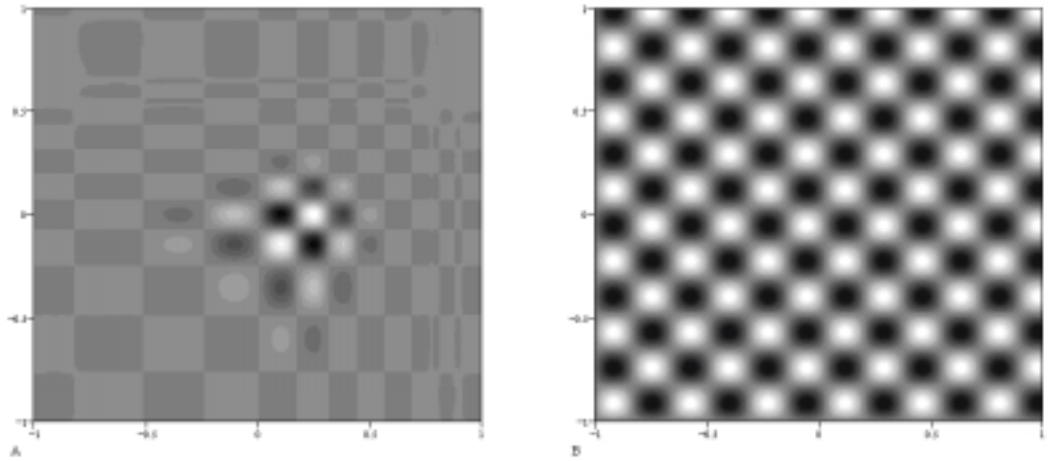


Figure 5.3.1 Left: A Wavelet filter. Right, a Fourier Filter. Both are illustrated on a 256x256 pixel image.

The spatially localized nature of wavelet analysis offers a computationally attractive solution to the sliding window computational redundancy associated with sliding window Fourier Analysis. Recall that the wavelet hierarchy consists of spatially localized wavelets of increasing size. As they are spatially localized, each wavelet filter in the hierarchy is responsive to a localized region in the image. One can thus create a vector for each location in the image using only the set of vectors that are responsible for constructing that location in the image. Thus, unlike the sliding window approach using windowed Fourier analysis one does not need to re-compute the window for each step. A wavelet transform can be applied to the whole image, recursively up to the largest desired wavelet resolution scale. This largest wavelet scale thus determines the sliding window size. Then, vectors for each window in the

image can be cut out of the wavelet transformation matrix. For example, a three level recursive application of the 4^{th} order anisotropic Daubechies wavelet transform to a 512x512-pixel image defines a 16x16-pixel sliding-window.

Figure **5.3.2** illustrates the optimal response of each of the wavelets from each filter bank. Each filter bank is illustrated by the gridlines. Each 16x16-pixel location in the sliding window block is thus responsive to one filter of each of the types illustrated. At the top left is the DC component vector. The response of all the 64x64 DC blocks put together produces a smoothed image. The relatively symmetric filters run down the diagonal of the image and have been combined into groups to illustrate the way in which the filters combine to reconstruct spatially localized frequency patterns. Notice also that this anisotropic wavelet transform also contains filters that look very much like classic edge detectors. Similar filters or initial linear transformations of image data have been found by a number of authors using empirical techniques for finding statistically independent components of large data-sets of natural images [2].

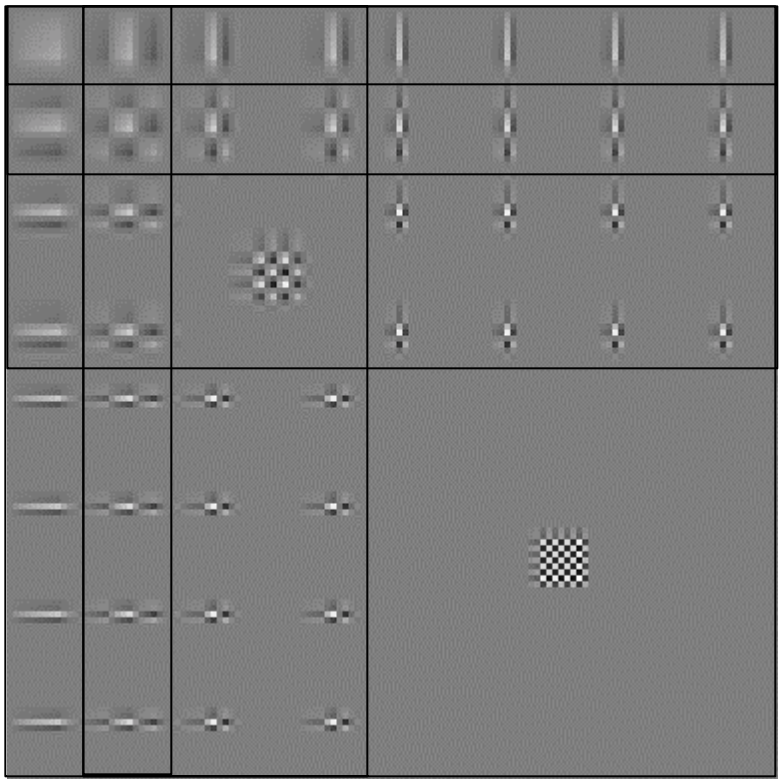


Figure 5.3.2 The types of filters produced by 3 recursive applications of the anisotropic 4th order Daubechies Fast Discrete Wavelet Transform.

The sliding window technique described above was applied to the CIE Y component of the 512x512 urban image and the greyscale 512x512 image of farmland. A slightly modified version of the standard 4th order Daubechies wavelet was used in which wrap around effects at image boundaries were eliminated using Gram-Schmidt orthogonalization of the ends of the matricies as described in [23]. To further reduce the dimensionality of the vectors for each of the blocks, each filter bank in the wavelet hierarchy was summed to create 16 coefficients. For larger data sets this additional step is not necessary. Data was modeled using the graphical model developed earlier for pixel level measurements but now generalized for the block level features and the relationships between these block level features. The results of these investigations are illustrated in Figure 5.3.3 to Figure 5.3.5.

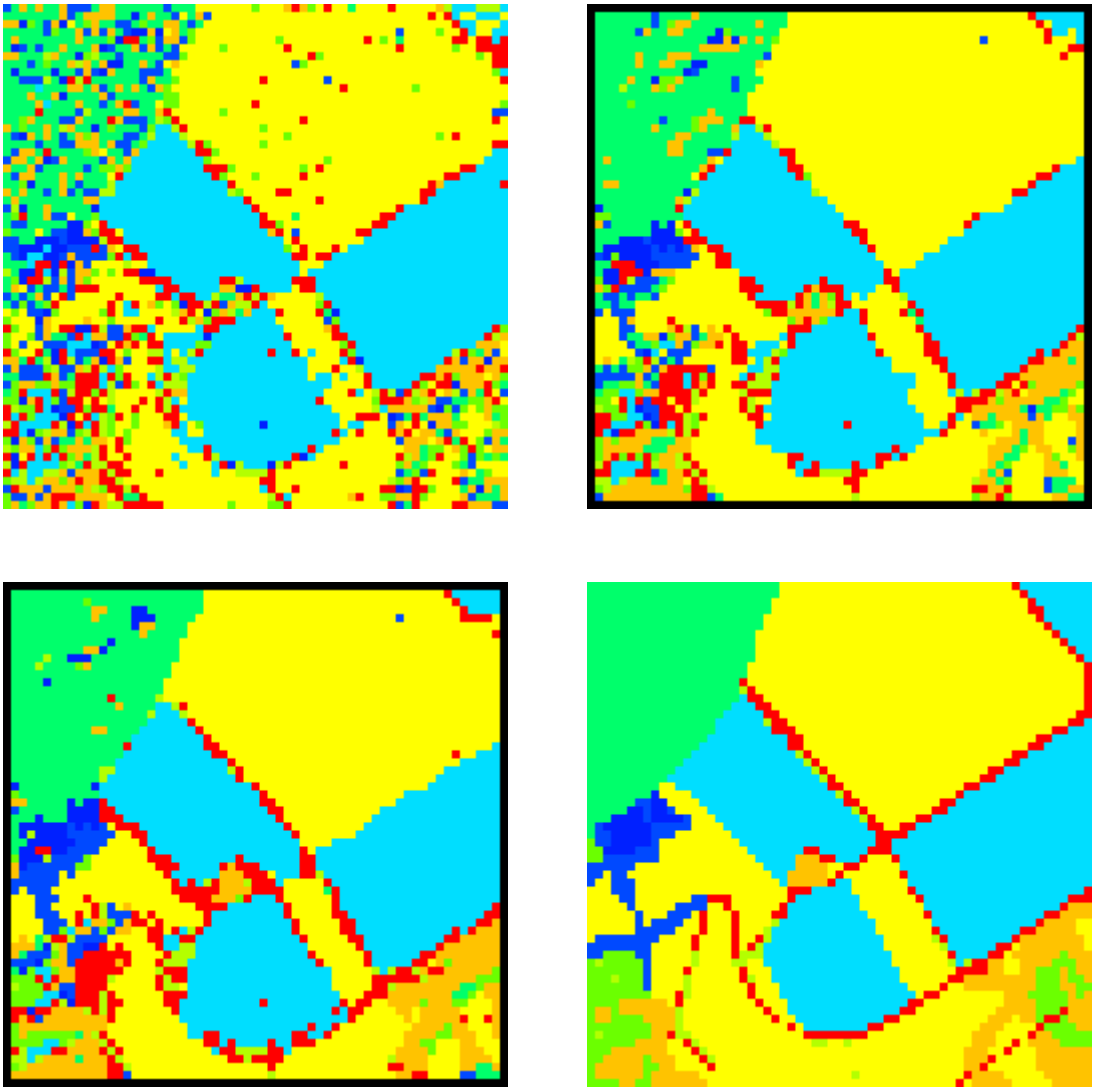


Figure 5.3.3 Upper Left: Classification of the farmland image based on wavelet coefficients from a sliding window only. Upper Right: A single iteration of probability propagation in the local context model. Lower Left: Three iterations of probability propagation in the local context model. Lower Left: The original hand labeled segmentation when re-sampled to a 64x64 image.

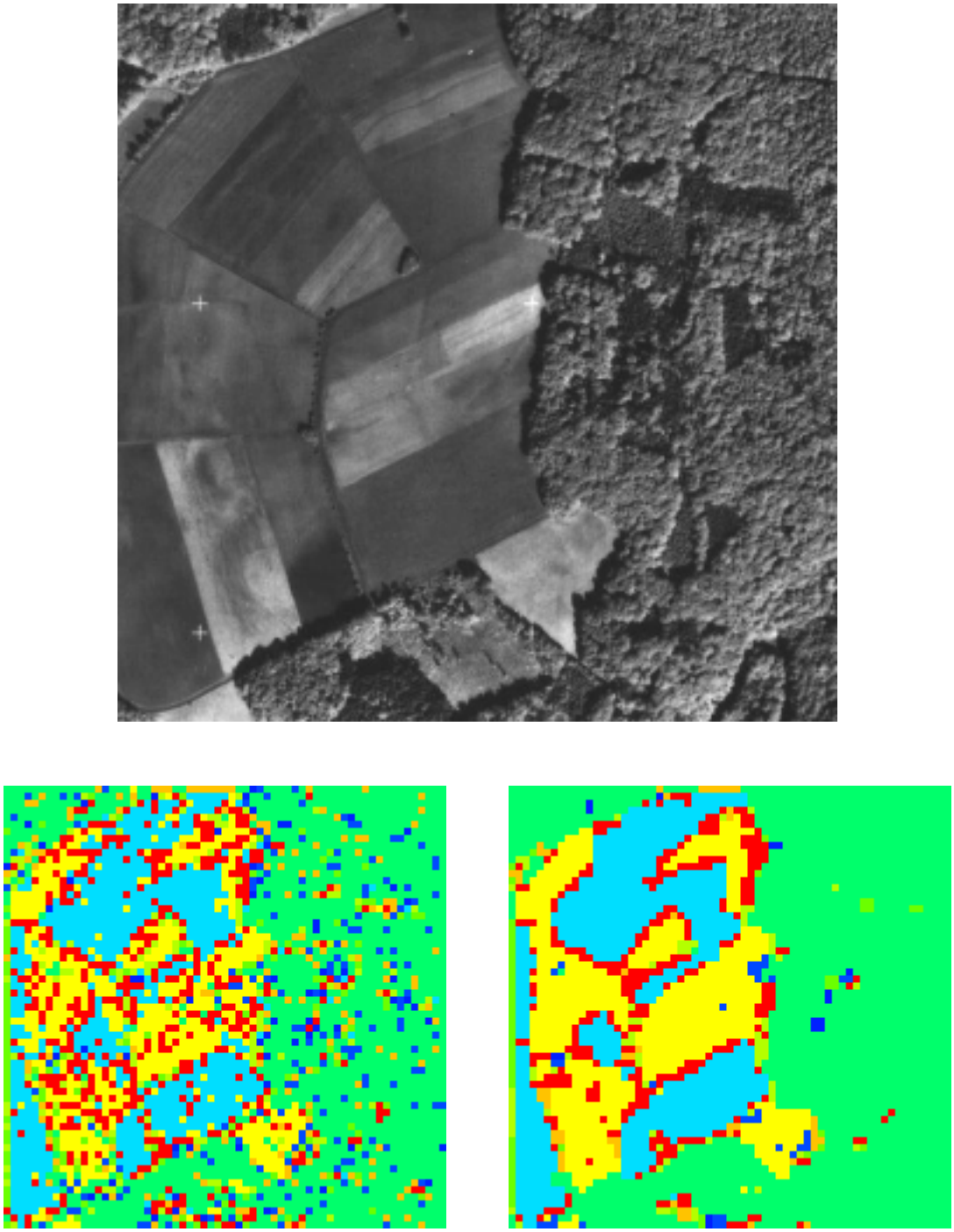


Figure 5.3.4 Top: Another image of farmland. Bottom Left: The wavelet only classification. Bottom Right: The wavelets with context. All probabilities for this classification and segmentation were "learned" from the previous farmland image and its associated labelling.

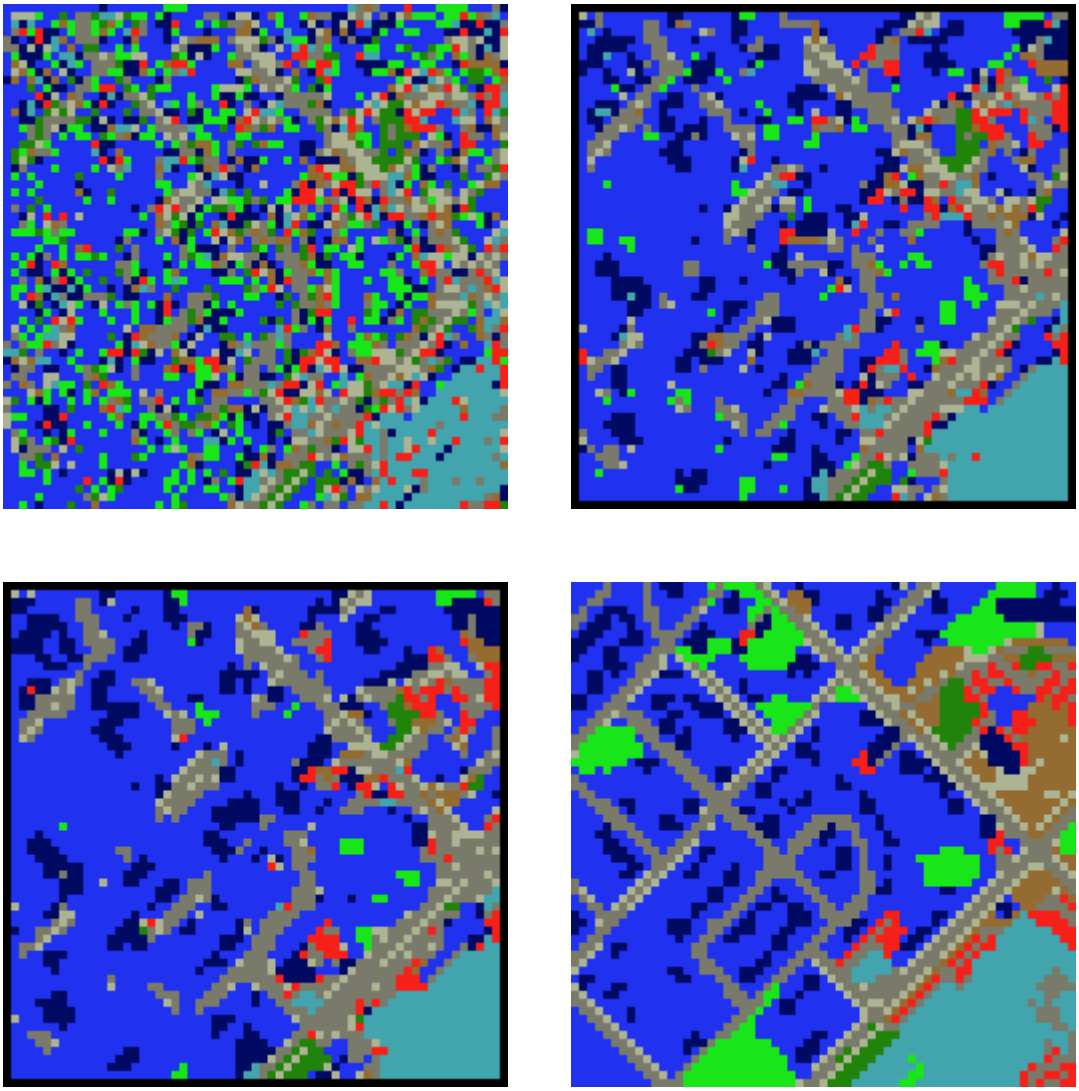


Figure 5.3.5 Upper Left: Classification of the window blocks for the image of an urban area based on wavelet coefficients alone. Upper Right: A single iteration of probability propagation in the local context model. Lower Left: Three iterations of probability propagation in the local context model. Lower Left: The original hand labeled segmentation when re-sampled to a 64x64 image.

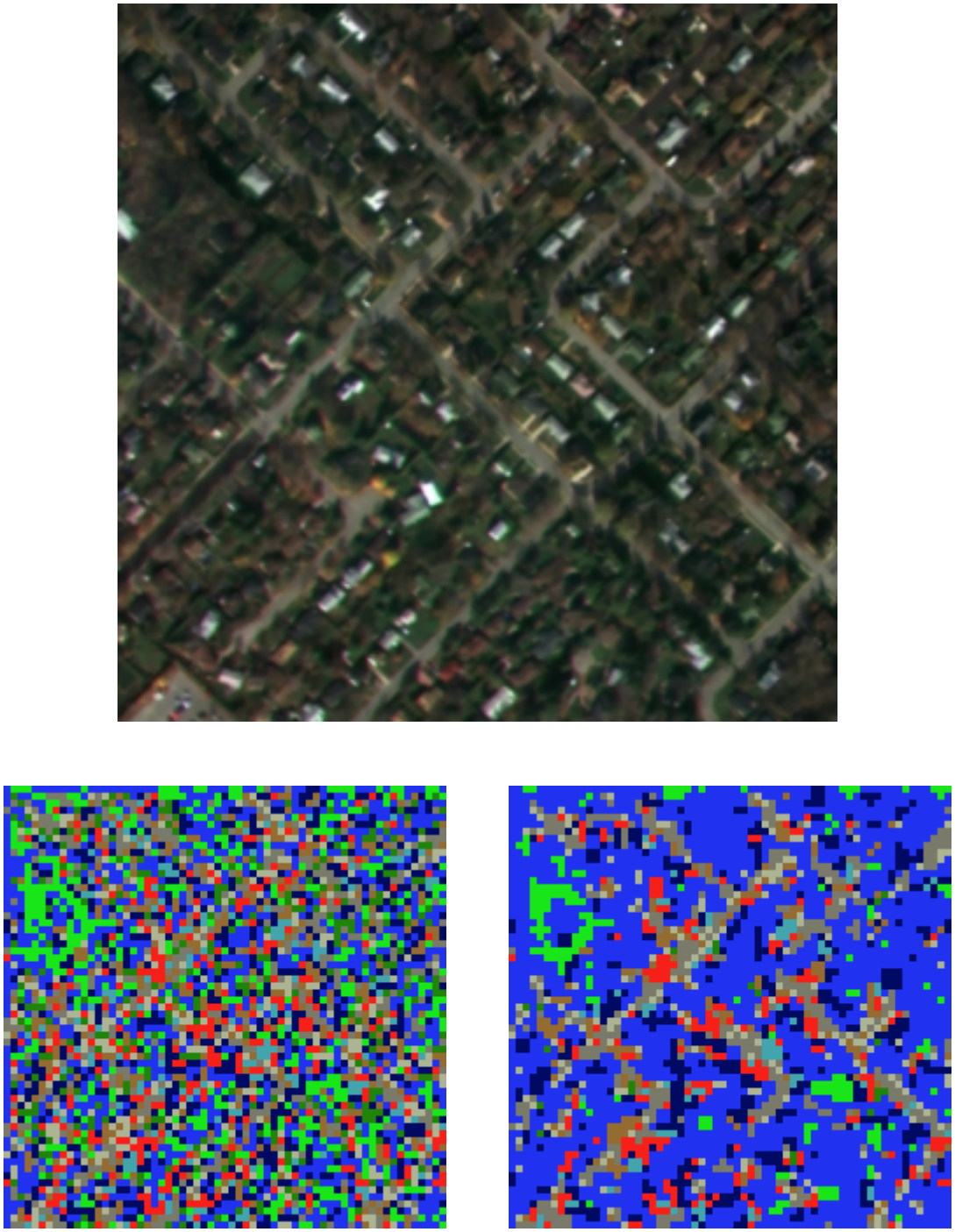


Figure 5.3.6 Top: Another image of an urban area. Bottom Left: The wavelet only classification. Bottom Right: The wavelets with context. All probabilities for this classification and segmentation were "learned" from the previous image of an urban area and its associated labelling.

5.4 Incorporating Prior Knowledge From a Coarser Scale Analysis

The coarse scale image segmentation and classification for the farm image is quite good. However, the coarse scale segmentation and classification for the urban image is quite bad. This observation emphasizes the importance of scale for segmenting images. For the segmentation of extremely large images of the earth, choosing an appropriate scale for each feature of interest is very useful. For this reason, recent research in the image processing community has looked at the use of the Multi-scale Markov Random Field [6]. In this technique the image is broken up into resolution levels. At each level of resolution is associated a random variable characterizing the content of the image. Starting with a single variable for the whole image, each successive level of resolution in the image is made into a grid of slightly higher resolution variables. Successive resolution levels only depend on a variable from the resolution level above. Finally at the lowest level, variables represent the image pixels. The stochastic model takes the form of a tree and for this reason very efficient statistically optimal algorithms exist for fitting the model to data [1]. However because of the tree structure, image segmentations can appear “blocky” and for this reason a number of techniques have been developed to overcome this limitation while still retaining the tree structure [33]. Such a model for a quad-tree is illustrated in the Figure 5.4.1 where on the left is illustrated the area of the image that each variable on the right characterizes.

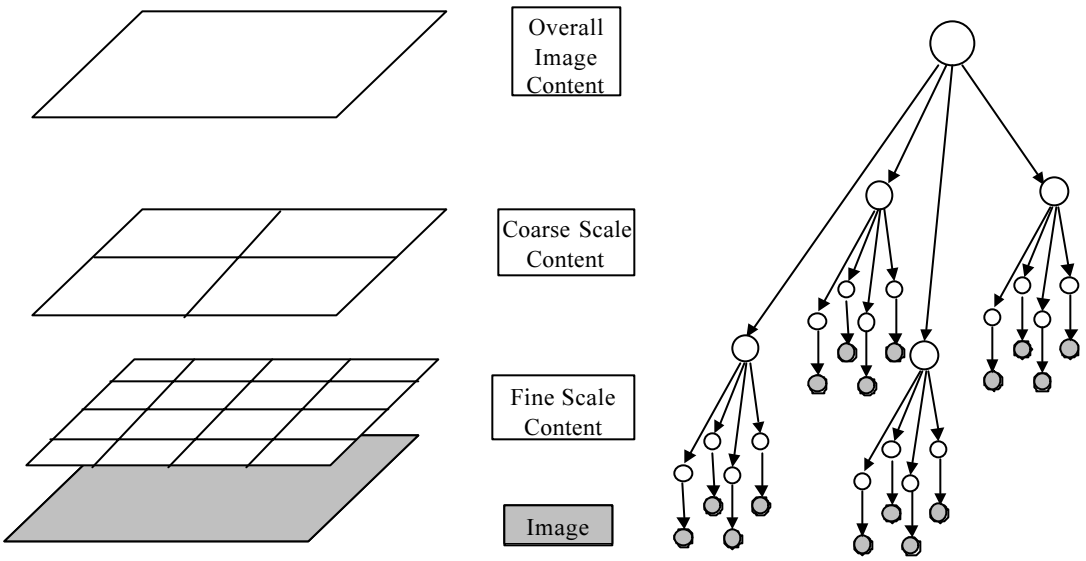


Figure 5.4.1 An illustration of a multiscale tree model as a Bayesian Network.

The tree model differs from the GMRF model developed earlier in that the relationships between pixel sites are now related to one another by a hidden variable and thus conditional probability distributions need to be specified and not clique potentials or joint probability distributions. Note also that the hidden variables in the tree can be transformed to observed variables if labels for these coarser scale interpretations are specified. Such is the case for our image of the urban area.

With the formalism of the graphical probability model it is possible to expand the previously developed probability model in which pixel interpretations interact locally with one another to incorporate probabilistic information from a coarser scale analysis. As alluded to earlier, our task of computing the appropriate probabilities in the model is simplified as we have labels for the multiple scale interpretations. Thus, consider that for each sliding window in the previous model we could introduce a joint probability table on the relationships between that sliding window and the pixel level variables that are primarily within that window (i.e.

just the middle pixels of the square window). Such a model is not a tree; in fact it contains many cycles even for a simple expanded quad-tree model. Figure 5.4.2 illustrates how such a model for a quad-tree with local interactions could be expressed using directed arrows for relationships encoded with conditional probability tables and undirected arrows for relationships with joint probability tables.

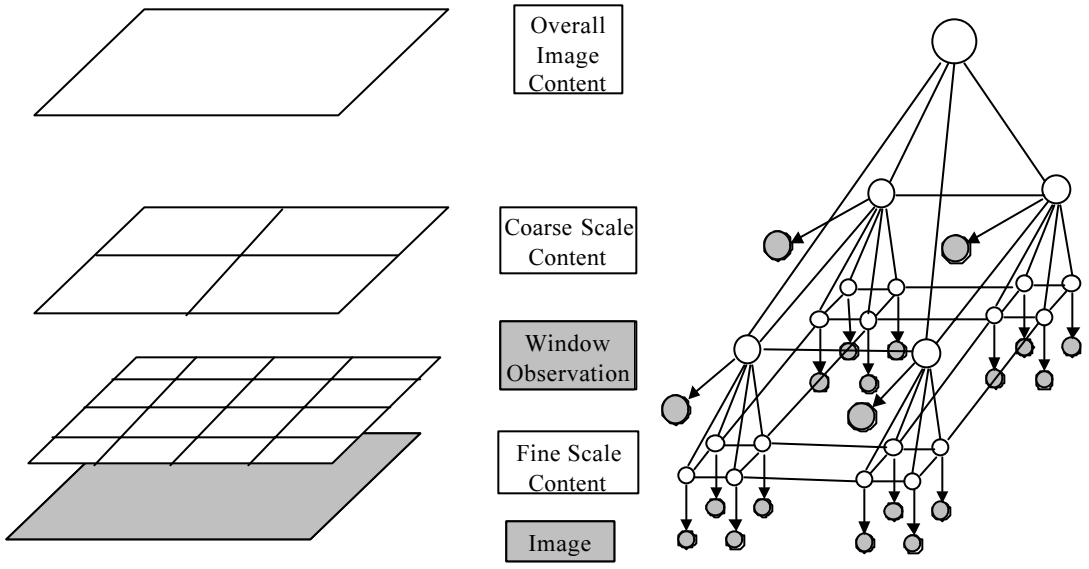


Figure 5.4.2 A Multiscale model with local interaction effects and observations from windowed multi-pixel regions and pixel level measurements.

Recall that the window observations are on greyscale values only, while the Image pixel level observations contain color. The cycles in such a model are problematic in that an exact solution to the inference problem in such a graph would involve, for example applying the junction tree algorithm to construct cliques in the graph. However solving the inference problem using such an approach is NP-hard as demonstrated by Cooper [13] causing potentially extremely slow exact solutions. A similar problem is found in decoding algorithms as discussed earlier. However probability propagation comes to our rescue for a

means of performing inference in such a model. The inference will be approximate, however it will be fast. If one looks down through the coarse scale model to the fine scale model below, such a model graphically takes the form of Figure 5.4.3. Only 3x3 patches of pixels are illustrated in the lower level for clarity. For a sliding window size of 8x8-pixels moving at four pixel increments a 4x4 patch of pixels would be produced as the primary pixels for each sliding window.

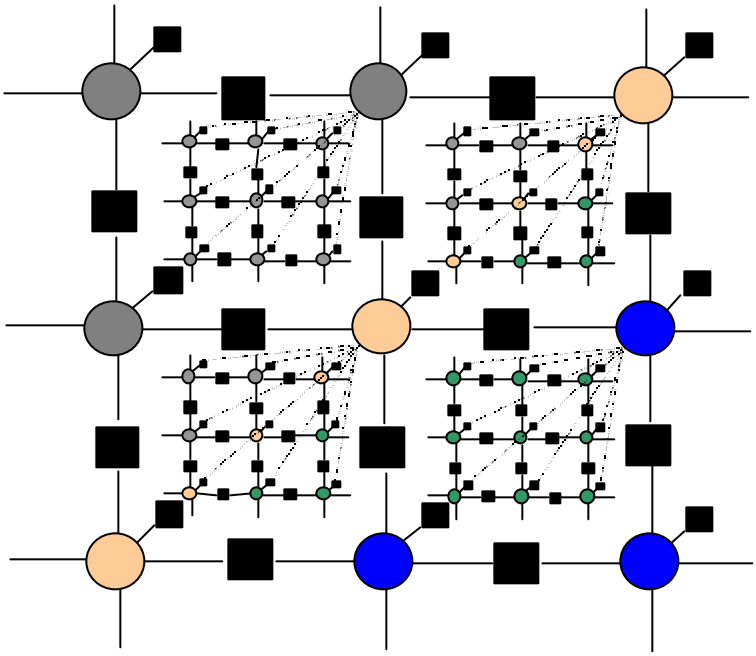


Figure 5.4.3 A hierarchical model for incorporating knowledge from the coarser scale analysis with pixel level analysis illustrated as a factor graph.

This model was applied to the urban image. Three iterations of message passing were applied to the coarse scale plane, one message was sent down from the higher level variables and then three iterations of message passing were performed among the lower level variables. For the investigation, each of the relationships between higher level window variables and lower level pixel variables were set to be the same within each window. The result is that, the

probabilistic messages sent from by the higher-level analysis produce what might be considered as a "square prior" over pixels in the lower level image. However, as one can observe from the following image, the "square prior" is relatively quickly "eroded" by the other probabilistic messages.

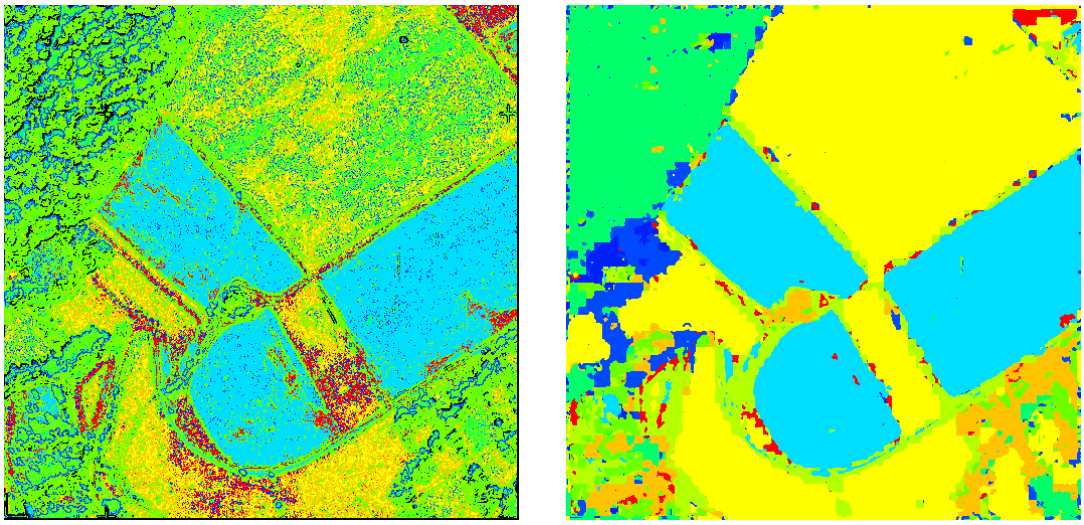


Figure 5.4.4 Left: Local message passing was used to segment the farmland image without the context from higher level analysis. Right: Local message passing in the farmland image with the context from the higher level analysis.

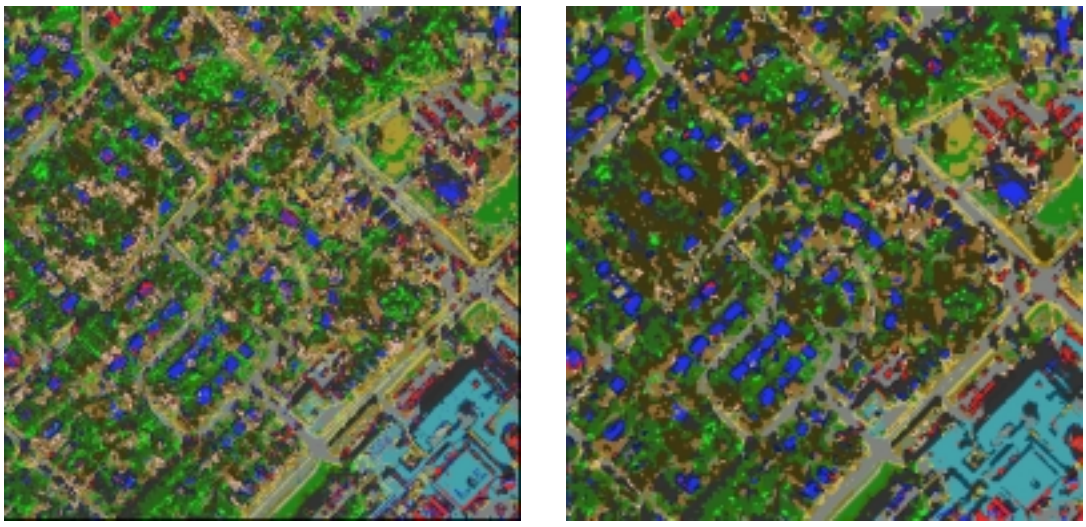


Figure 5.4.5 Left: Segmentation with no higher level context. Right: Segmentation with higher level context.

Clearly in the greyscale farmland image of Figure 5.4.4 the higher level information was essential for finding the classes specified in the labeled image. Figure 5.4.5 illustrates how many of the clearly inaccurate misclassifications such as house roofing being found on roads are "corrected" by the addition of the higher level analysis. However, the task of producing an extremely (conceptually) high level segmentation such as in the hand segmentation for the urban image introduced in Figure 1.1.1 is an extremely difficult task. In particular the extraction of roads amidst a high degree of clutter relies on high level knowledge concerning the nature and connectivity of roads. In fact, for highly cluttered scenes constructing a complete road network is extremely subjective and the existence or non-existence of a road can be completely ambiguous. However, the algorithms presented here have a chance at capturing some of these high level relationships in a probabilistic manner. With a slightly expanded neighborhood system for local interactions, concepts such as connectivity could be more accurately contained within the probabilistic model. Further, the tests shown in the previous section used fairly small amounts of data. With far larger training sets these techniques would be much more powerful for segmenting complex scenes.

However, robust image understanding systems can be quite large, incorporating a great deal of high level knowledge into "models of objects", using low-level features have been extracted as hypothesis "seeds". These systems can get extremely problem specific. However, most of these systems have one thing in common: the use of geometric relationships between curves. The next section discusses the common approach of extracting curves and lines from images and it is shown how these techniques can be used with the segmentation information obtained above.

6 Edge Enhancement and the Search for Salient Curves

6.1 Key Issues

The detection of edges within images is generally regarded as an important operation [72] and is very often the first step in the image processing approach to pattern recognition. Experiments now considered classic by Hubel & Wiesel on neurons in visual cortex provide further biological evidence for the importance of extracting edges early in the analysis of an image [31]. More recent work at the intersection of biology, statistics and information theory has begun to build a convincing argument for why we have edge detectors from an information theoretic point of view in terms of sparse, distributed and statistically independent feature detection [2]. The detection of edges is often coupled with a secondary step, linking single pixel scale edges together to form curves. First, edges are usually detected initially by taking spatial derivatives of an image [10]. Second, essentially some sort of search is made for a curve path through the edge-enhanced image. There exists an extremely large amount of literature on the subject of low level edge extraction or enhancement. Frequently, edge detection and the linking of edge segments into curves is performed on greyscale images only. However, evaluation of many of these purely low-level techniques by human observers in terms of their perceived accuracy at extracting edges from greyscale images has shown relatively little difference between techniques [68]. However, in this thesis, it is illustrated how edges can also be found in the "probability planes" of the various materials modeled in the previous section using color and essentially "texture" information. Edge segments in the probability images are then linking into curves based on

the algorithms developed in [73] and [11] respectively. The next few sections describe these algorithms.

6.2 Edge Enhancement

As discussed earlier, recent research into the multi-resolution analysis of images has tended to focused on computationally efficient ways to combine information from multiple scales using an initial linear transformation such as a Fourier Transform, a windowed Fourier Transform (or Gabor filter) or Wavelet transforms etc… For the purposes of extracting edges from a probability plane or greyscale image for subsequent linkage into curves, each of these transformations have significant drawbacks. For the detection of edges, the global nature of Fourier filtering is clearly inappropriate. Gabor filters or related windowed wavelet filters can be used to estimate edge magnitude and orientation, however these types of filters are biased to a particular direction. Thus, a separate filter must be computed for each edge direction. Further, oriented edge filters are generally rather large in diameter. Thus, this family of filters is relatively computationally expensive for the utility they provide with respect to edge orientation. Invertible wavelet transforms in two dimensions generally act as spatially localized frequency analyzers and do not respond in an unbiased manner to edges. However, spatially localized features such as edges can be easily extracted over multiple resolutions using a simple low-pass pyramid (e.g. a Gaussian smoothing filter) and a simple edge filter.

Gaussian smoothing pyramid filters have been used extensively, however the discretized Gaussian kernel commonly used in Laplacian of Gaussian pyramids [9] has been shown to

have a slight deviation from circularity [73] with respect to angular frequency response. Thus if one wishes to estimate edge orientation in the smoothed image, this fact will introduce an orientation bias. For this reason, in [73] the *frequency sampling technique* was used for the design of a rotationally symmetric smoothing filter. The 5x5 filter produced with this technique is illustrated in Figure 6.2.1.

.0006	.0178	.0290	.0178	.0006
.0178	.0670	.0880	.0670	.0178
.0290	.0880	.1194	.0880	.0290
.0178	.0670	.0880	.0670	.0178
.0006	.0178	.0290	.0178	.0006

Figure 6.2.1 (5x5) Smoothing filter kernel.

Most simple *edge segment* filters are essentially gradient estimators along an oriented line segment. But, not all simple edge filters are unbiased to a particular direction. However, the edge filters designed by Wilson and Bhalerao in [73] produce relatively unbiased orientation estimates, are of small radius (i.e. convolutions of 3x3 and 4x4 pixels) and subsequently are of minimal computational demand. In [73], a pair of convolutional kernels was designed so that each kernel gave an ideal response to one-dimensional image *features* with frequency spectrum concentrated along a line at orientation θ_i. Such that $\theta_1 = \pi/4$ and $\theta_2 = 3\pi/4$. The results of this filter design for 3x3 and 4x4 kernels are shown in Figure 6.2.2. Note that each kernel has a corresponding pair with its coefficients rotated by $\pi/2$.

-.0741	-.0955	0
-.0955	0	.0955
0	.0955	.0741

-.0107	-.0496	-.0277	0
-.0496	-.1292	0	.0277
-.0277	0	.1292	.0496
0	.0277	.0496	.0107

Figure 6.2.2 Wilson's (3x3) and (4x4) edge filter kernels.

Compare these filters with the more standard and simplistic Pewitt and Sobel edge detection filters illustrated in Figure 6.2.3.

-1		1		-1	-1	-1
-1		1				
-1		1		1	1	1

-1		1		-1	-2	-1
-2		2				
-1		1		1	2	1

Figure 6.2.3 Pewitt and Sobel edge detection masks, top and bottom respectively.

Convolution of an image with a pair of kernels produces two images that can be subsequently combined to produce something similar to a magnitude or flux image and a phase or edge segment orientation image. The resulting images could be considered as edge segment direction and magnitude measurements. One can compare the magnitude of the frequency response of the filter pairs to see clearly the symmetry of the new filters. These two image planes form the basis of input to subsequent analysis described in the next section.

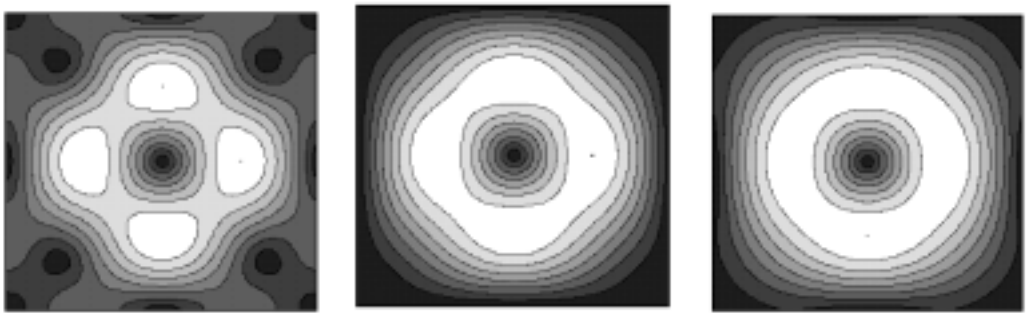


Figure 6.2.4 Frequency response of Pewitt, Sobel and Wilson 3x3 edge mask pairs.

6.3 The Search for Salient Curves

The task of finding perceptually relevant contours within an image has been approached in an extremely large number of different ways in the past. However, the purpose of extracting salient curves in the context of this thesis is to provide an extraction of the ridges in the probabilistic estimate of various classes for pixels or windows of pixels. As there are a large number of potential shapes that could be taken for the regions defined by these classes many existing contour modeling algorithms have significant shortcomings in this respect. One of the most popular techniques involves the use of active contour models or "snakes" [38]. In this scheme, curves are represented as a parametric equation and the equation is optimized under a number of constraints. Photometric constraints are responsible for *image forces* that pull the snake to features in the image. While, geometric constraints give rise to *internal forces*, controlling the shape of the snake. These constraints enforce geometric qualities such as smoothness. During optimization, the contour or snake moves around the image until a solution is found at an energy minimum. The energy function takes the following form:

$$E(v) = -\int_0^1 P(v(s,t))ds + \frac{1}{2}\int_0^1 \left[\alpha(s)\left|\frac{\partial v(s,t)}{\partial s}\right|^2 + \beta(s)\left|\frac{\partial^2 v(s,t)}{\partial s^2}\right|^2 \right] ds$$

Eq. 6.3.1

The first term of Eq. 6.3.1 represents the image energy and the second term represents the internal constraint energy. The functions $\alpha(s)$ and $\beta(s)$ are the functions that determine the geometric constraints. However, because there is no direct link between the internal and external energy terms, the discovery of the "correct" energy functional has been criticized as being an error prone process [14]. Another popular technique for finding contours is known as Shape-plus-texture modeling. These techniques have been used to vectorize faces images

[3]. However, they use pre-defined relationships between contour locations and expected greyscale values at specified offsets to the contours. There do exist a number of algorithms for identifying salient contours that do not rely on context specific information. Simple edge segment linking algorithms have been developed based on the connectivity of edge segments. However they tend to be rather susceptible to noise leaving a large number of broken edge segments. Other more complicated algorithms have been successful at extracting salient contours from noisy images [77] but rely on fairly heuristic methods [76]. Current trends in this area have led to the use of relatively simple gradient type information over multiple scales of resolution to locate contours by integrating this information [26]. This usually involves finding possible contours within the original image and also a coarser scale version of the image. The algorithm selected for this investigation is attractive in that it uses edge features extracted at multiple resolutions (making the algorithm less susceptible to noise) and then utilizes probability theory to fuse the information.

The algorithm selected for precursor curve extraction for this investigation is described in [21] and takes the following form. A sequential search through an edge-enhanced image is performed using the Zigangirov-Jelinek (Z-J) or stack algorithm [36], however the authors emphasize that any tree searching algorithm such as the A* algorithm [54] commonly taught in courses on Artificial Intelligence could be used. For all of these types of search strategies, a measure of goodness must be constructed. In [21] the authors take an approach to derive the measure of goodness or the path metric that is similar to an approach used in the past to decode convolutional codes [45]. In this approach, “random tails” are employed to ensure that the goodness statistic is not biased by length when two paths are compared. The path

metric consists of a likelihood ratio of the probability that the pixels along the path are an edge vs. the probability that they are not on an edge. The second term comes from the probabilities of a Markov Random Chain (MRC). The metric takes the following form

$$\Gamma(x_n, y_n) = \sum_{i=1}^{n} \left[ln \frac{P_{edge}\left(I(x_i, y_i), I(x_{j=1\ldots i-1}, y_{j=1\ldots i-1})\right)}{P_{non-edge}\left(I(x_i, y_i), I(x_{j=1\ldots i-1}, y_{j=1\ldots i-1})\right)} + ln\, P(s_{n+1} \mid s_n) \right]$$

Eq. 6.3.2

Where $I(x,y)$ = (magnitude, phase) at location (x,y). The path of an edge is modeled as an Autoregressive Moving Average ARMA process. The innovations in the linear filter of the ARMA are then used in the path metric as opposed to the original data. The filter can be computed recursively, simplifying the summation over the first term in Eq. 6.3.2. The ARMA assumption is used to both reduce the search space and to account for the discretization in the possible pixel path directions and the measured phase of the underlying edge segment from the edge filter. A Gaussian distribution for the each of the edge and non-edge classes is then used to model the filter output. The parameters of the Gaussian distributions and MRC are initially set heuristically. Once the algorithm has run on the lowest resolution image, there exists a hypothesized path through the higher resolution image. Potential pixel values can be gathered along this uncertain path and the EM algorithm is then used to find the parameters of a Hidden Markov Model with two hidden states, edge or not an edge and two conditional distributions over pixel values. In this way the two Gaussian distributions are updated to estimate the edge and non-edge conditional density functions. The original envisioned usage of this algorithm was on edge enhanced greyscale images. Thus initially, the transition probabilities for the MRC are set to being equally probable. However, as the algorithm could also be used on probability images for material classes as developed in the previous section in this investigation, contour characteristics can

be encoded using a prior distribution over the MRC for well known classes (e.g. roads). As the curve path is searched in the coarsest level of analysis, the MRC probabilities are updated. Knowledge of the edge path from coarser scale analysis is then passed through the MRC for each curve to the next, higher level of resolution analysis. At the next analysis level the chain is again updated as the curve is searched. These two measures are then combined and used for the path metric of the search algorithm. In this way, a very small number of *prior* parameters are needed [12] however, the potential to allow fairly strong priors with respect to different types of curves still exists. Figure 6.3.1 contrasts the algorithm's performance on the greyscale edge enhanced urban image and on a subset of materials from the previous probabilistic segmentation.

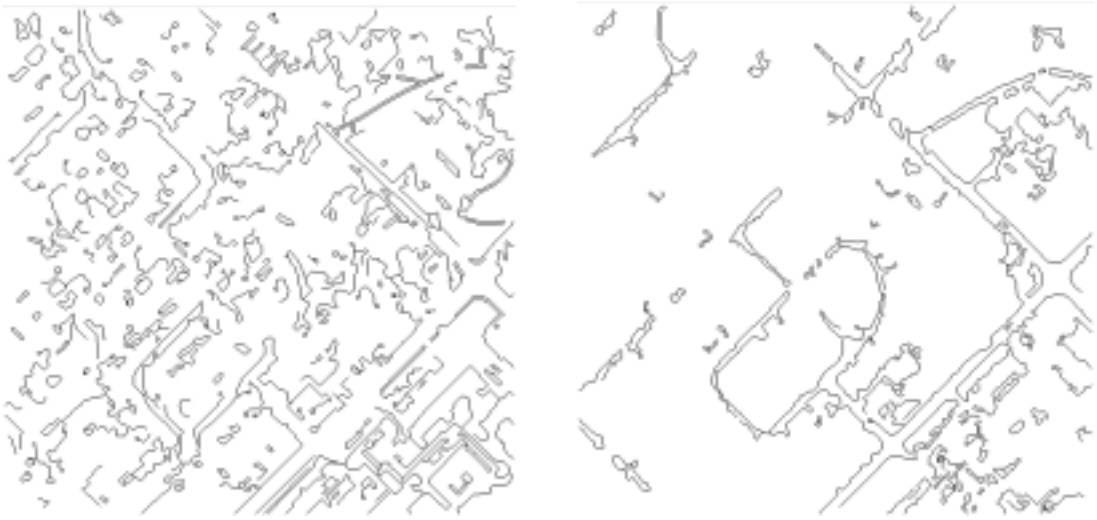


Figure 6.3.1 Left: Contours found from the edges of the greyscale urban image. Right: Contours from the probability planes for all road or car materials.

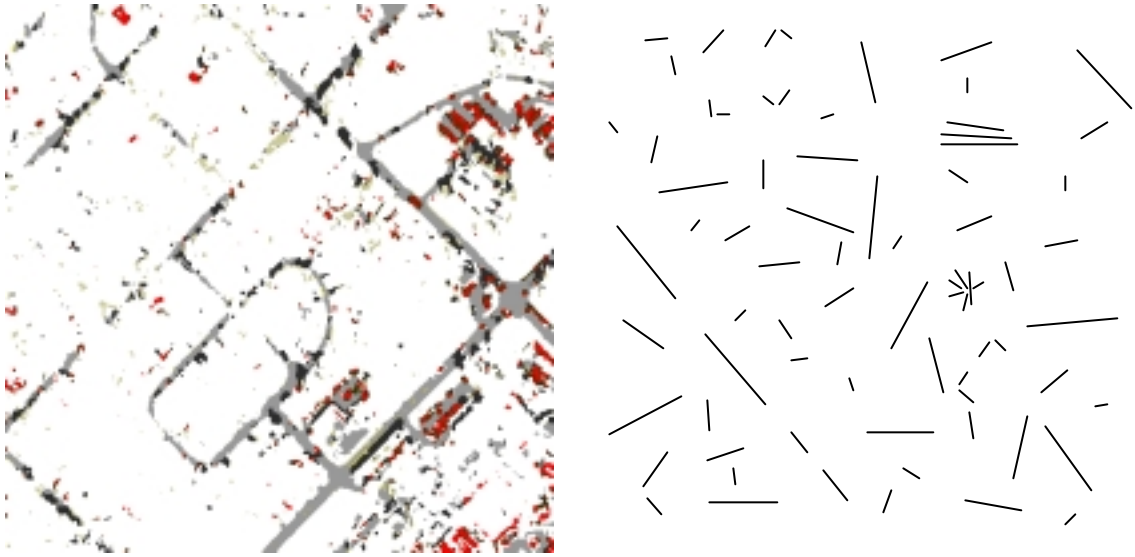


Figure 6.3.2 Left: Road and car materials found as the most likely material, from the previous segmentation procedure. Right: Randomly placed lines?

Humans segment highly cluttered scenes for practical purposes quite easily. Consider the figure on the right of Figure 6.3.2. Are there any patterns in this cluttered image of randomly placed lines? Contrast this with the segmentation of the urban area on the left.

From these two illustrations, one can see that there are a number of perceptual principles that human use to find structure in images. The right hand figure illustrates that these perceptual principles are extracted so readily by humans that they are found somewhat independent of the subject matter of the image. Consider the image on the left of Figure 6.3.2, most people will spontaneously find at least three groups in this image. These perceptually relevant features commonly known as parallelism, co-linearity, intersection and proximity can be extremely useful for reasoning about complex, cluttered images. Practically, in this situation they can be used to determine where the obscured road segments are located. Further, such features could be helpful for further removal of any false positives not eliminated by the high

level knowledge used in the previous segmentation procedure. The following Chapter discusses the role of perceptual organization in image analysis and the computation of perceptual features.

7 The Perceptual Organization of Curves

7.1 Key Issues

The grouping of low-level features and curves in particular is a well-known intermediate vision task. The task of grouping has often been justified as imparting efficiency to the computationally intensive process of high-level model identification. Much of the early work in this area was done by Lowe [44]. The problem of extracting structure from simple visual features was a key question investigated by the German school of psychology known as the Gestaltists [64]. This discussion will focus on the perceptual organization of curves and will draw from the results of the Gestaltists. To further simplify the computational demands of computing perceptual features, curves can be further broken down into straight, *line-segments.* In this context it is possible to focus on the following perceptual groups: proximity, parallelism, co-linearity, intersection and continuity. Although there are many shapes in the world, these features are extremely applicable to a large number of objects. Specifically, the computation of these types of Gestalt features is quite common in the area of aerial image analysis [24], but computation is relatively heuristic in most cases. The next sections discuss the computation of perceptual grouping relationships between straight line-segments. First, an appropriate system of measurement for these relationships is developed. Then, a graphical probability model is proposed to evaluate the saliency of perceptual line relationships. The model is general in that it does not rely on any detailed prior knowledge of the structure of the underlying objects in the image and thus it can be used to detect perceptually relevant features for a broad range of applications.

7.2 Extracting Lines from Curves

The extraction of salient curves is an extremely useful feature. However, extracting further information concerning the relationships between curves is an essential part of the task of image interpretation. This task can be somewhat complicated by the fact that curves can have a large degree of variability. In other words, the dimensionality of any system measuring the relationships between two arbitrary curves is extremely high. However, the task of perceptual grouping can be greatly simplified by using linked straight line-segment approximations of curves. For this investigation a recursive algorithm for segmenting 2-D curves into straight line-segments first proposed by Lowe [44] and refined by West and Rosin [65] was used. In this technique, a recursive subdivision of a curve is performed to obtain a binary tree that is traversed to select the "best" possible representation according to a "saliency metric". Such a procedure invites the use of a more rigorous probabilistic definition, however such an investigation is beyond the scope of this thesis.

The algorithm begins with lists of pixels provided by the curve segmentation algorithm and splits them into two at the point of maximum deviation between the approximation and actual data. The process continues until the sub-lists are of the size of pixels. This process produces a segmentation tree with increasingly finer segments and better approximations. The ratio of the length of the line-segment, over the maximum deviation of the curve from the line-segment is the calculated as a *significance rating*. The higher the value of this measure, the more significant the primitive is said to be. The tree is traversed up from the leaves and if a feature is more significant than all its children, it replaces them. The result is a segmentation of a curve into perceptually significant line segments. The process is illustrated Figure 7.2.1.

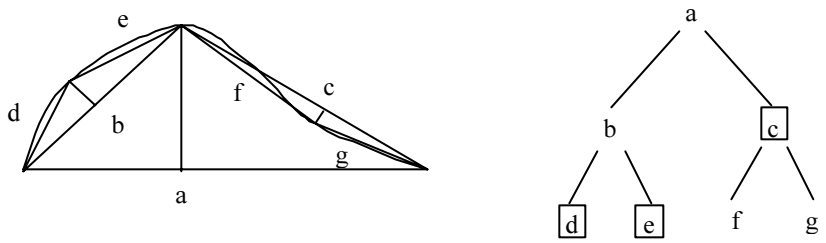


Figure 7.2.1 The segmentation of a curve and the segmentation tree with selected line-segments.

Figure 7.2.2 illustrates this line segmentation procedure on the curves extracted from the urban image.

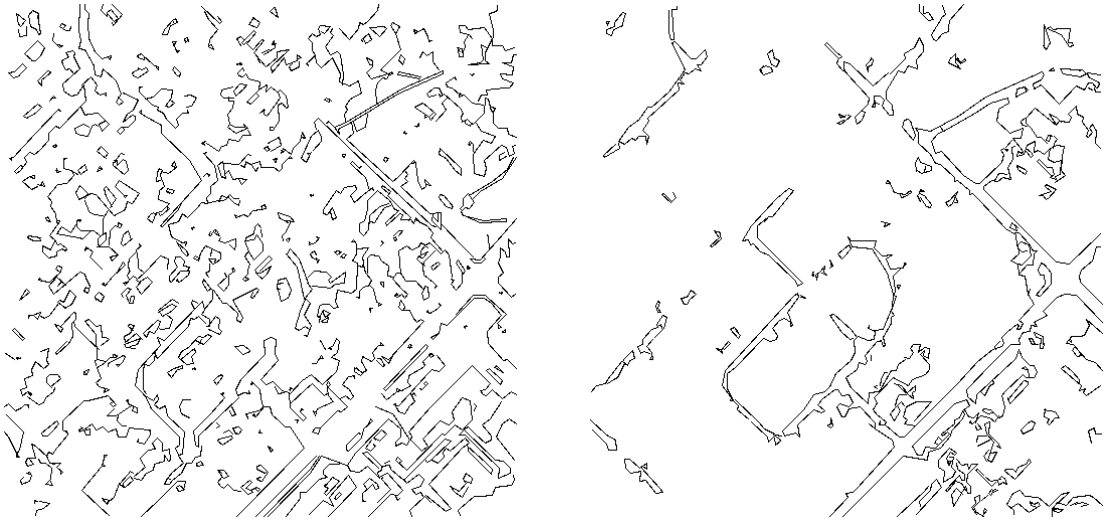


Figure 7.2.2 The line segmentation procedure applied to the curves extracted from greyscale edges (left) and from the color and texture segmentation image (right).

7.3 The Perceptual Organization of Lines, Prior Art

Once curves have been segmented into straight line-segments the task of computing the perceptually relevant relationships between lines is simplified. A number of approaches have

been taken in the past to calculate Gestalt relationships between lines. However, most of them are relatively heuristic in nature. Lowe has proposed a number of formulae for evaluating the perceptual relevance of a given relation relative to the global density of lines [44]. In this reasoning the formulae for the saliency of a relationship is relative to the chance of the relationship occurring by an *accident of viewpoint* and line density is used as an estimator of the likelihood of this sort of accident. Lowe takes into account the length of lines and their relative separation to form measures independent of absolute length. Lowe also uses some the simplifying assumption that all perceptual features are dependent on proximity to reduce the $O(N^2)$ search space of all possible lines. In contrast to Lowe's formulae ranking perceptual saliency based on expected random occurrences, other researchers have used voting methods [26] [67] to rank perceptual groups. In the work of Sarkar and Boyer [67], Gestalt relations are defined as logical statements. Position and orientation measurements for a given line are discretized and each line then applies a logical relation to the entire feature space and tags or "votes" for the discretized bins that satisfy the relationship. If points from two lines vote for the same location, the are deemed as being related. In the more specialized area of aerial image interpretation various relatively heuristic techniques are also common. In [24], various statistics regarding means and standard deviations of length are combined somewhat informally to rank various groupings based on the local image properties. Fuzzy sets have been used to model and manipulate Gestalt relationships between lines. In one example, the task was to group fragmented pixel wide road curve segments into a larger road network [70].

The use of formal uncertainty modeling for perceptual grouping has been the subject of more recent investigation [66] [8]. In [66] a Bayesian network was constructed for the Gestalt relationships of: parallelism, L-junction, T-junction, co-linearity and "none". However fuzzy logic like triangle and box class conditional probability distributions were used. Scale invariant features were defined and the parameters for the probability distributions for the network were learned from a labeled data set using *learning automata* [52]. In the following section it is shown how a Gaussian mixture model can be used for representing the conditional probabilities over relationship measurements within a probability network when an appropriate coordinate system is selected for representing the relationships between lines. Further, using the standard EM algorithm for learning the mixture model from a labeled data set allows substructure or perceptual features not explicitly modeled by the labeled data set to be discovered. However, as discussed earlier, first an appropriate coordinate system must be defined.

7.4 A System of Measurement for Relationships Between Line Pairs

Consider that a curve-segmented image has been segmented into straight line-segments. The word "line" will be used in place of line-segment to indicate lines of finite extent in the following discussion. The task of measuring the perceptual relevance of various Gestalt relationships can be greatly reduced if an appropriate coordinate system is used for measurements. For the purposes of extracting general perceptual relationships between pairs of lines, the coordinate system should satisfy a number of invariance conditions. Namely, measurements should be rotationally, transitionally and scale invariant. Appendix II

describes the simple but important task of defining a coordinate system that satisfies the desired invariance properties.

7.5 A Probability Model for Perceptual Relations Between Lines

A user interface was constructed using OpenGL [53] to allow a data set to be generated for pairs of lines extracted from a greyscale segmentation of a large collection of urban and scenes similar to the one described earlier. People were given access to the underlying original image and told to classify *pairs of lines* into five classes of relationships: Parallelism, Co-linearity, L-junction, Unspecified but relevant and Accident of viewpoint. Each of the first four perceptual relationships are implicitly dependent upon proximity relationships [44]. Thus, the proximity relationship is modeled by the intersection of these perceptual classes. From the generated data set, a Gaussian mixture model was fit for each perceptual class using the EM algorithm with an MDL penalty term to control model complexity. The accident of viewpoint class was modeled as a uniform distribution over the domain defined by 90% of probability mass of the other perceptually relevant features. In this way the complete coordinate systems domain was thus restricted based on the proximity relationship. The mixture models for each class were then combined into a class-subclass model as discussed earlier. Although the classes in the data are not completely Gaussian as illustrated in Figure 7.5.1, the power of the mixture model for expressing arbitrary distributions is illustrated in Figure 7.5.2.

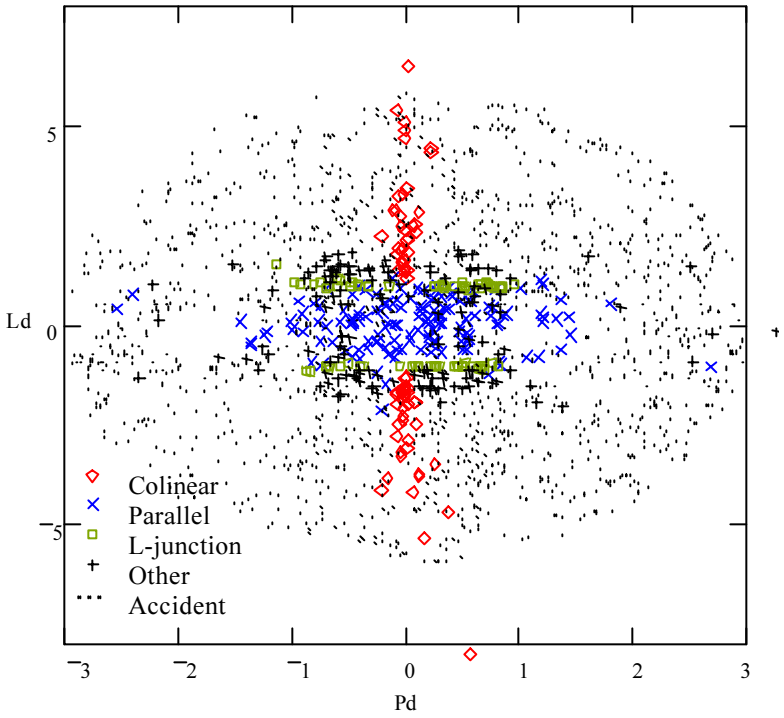


Figure 7.5.1 A plot of line relationships for the first two coordinates of the system developed in Appendix II.

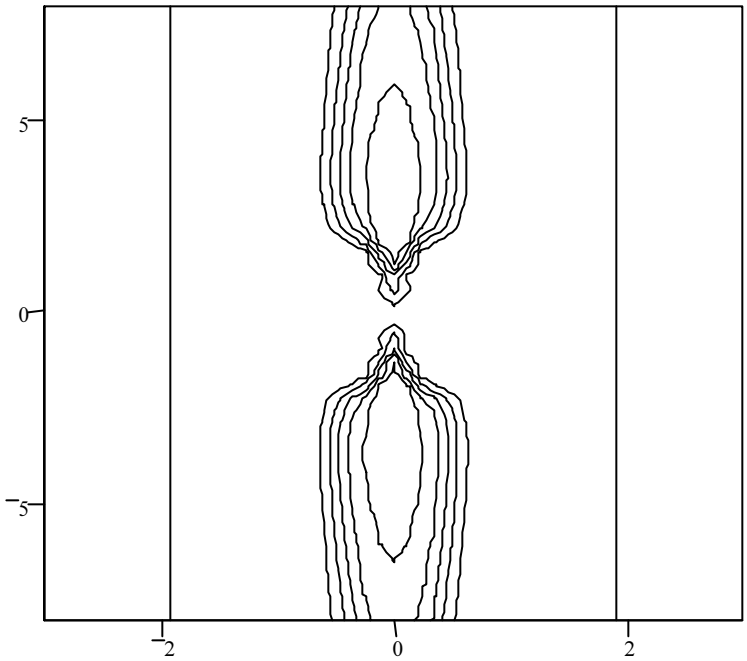


Figure 7.5.2 A contour plot of the posterior probability for co-linear lines.

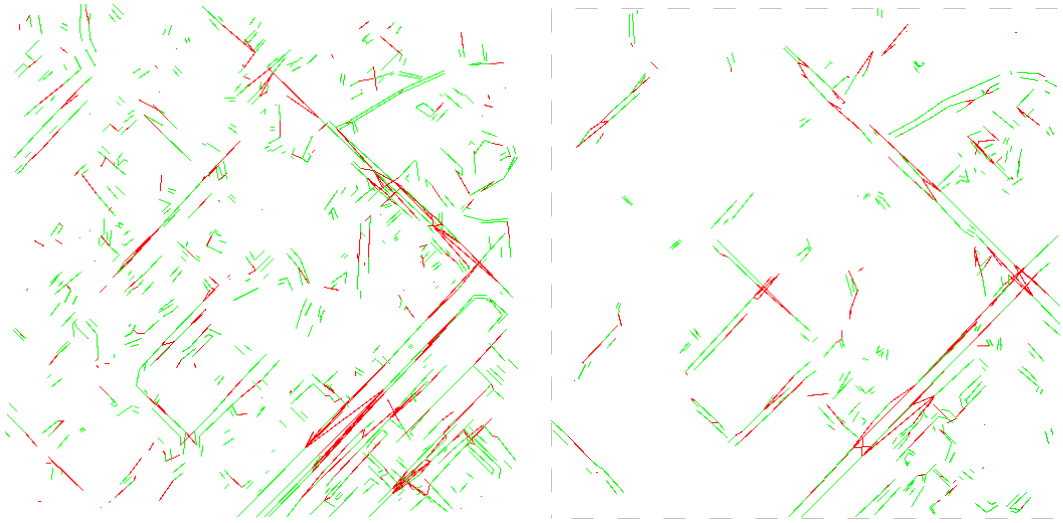


Figure 7.5.3 Left: The greyscale segmentation. Right: The color and texture segmentation. Green: Parallel lines found. Red: The completion of co-linear lines.

From Figure 7.5.1 one can see that the complexity of the line features has been greatly reduced in the color and texture segmented image. It is relatively common to apply these algorithms twice once on the initial line image and then again on the image consisting of co-linear lines "filled in" to create larger lines. One can imagine how such procedure could be quite useful.

These types of line images can be used to seed a much more highly specialized "model based" algorithms. The key issue here is that the use of the color and texture segmentation as the starting point for curve extraction imparts a great deal of computational efficiency to any further model identification procedure. For completely unrestricted images, searching for lines in greyscale edge images may provide a useful starting point for the model identification procedure. However, for the somewhat restricted domain of aerial image

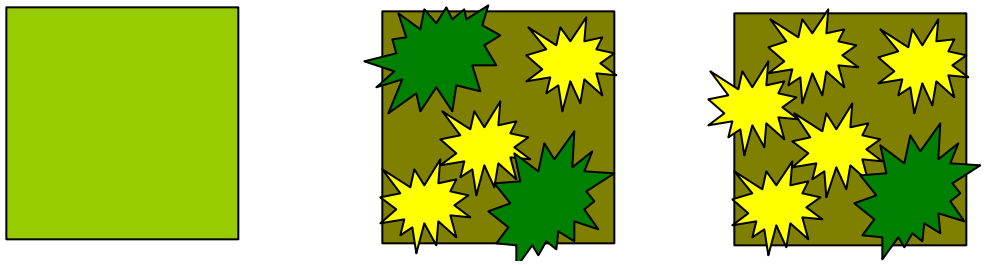
interpretation, the addition of the “higher-level” texture information is quite useful to speed up the process of extracting relevant linear features.

8 Sub-pixel Inference

Recall the earlier when color clustering was discussed it was shown that there is a pixel mixing effect that can be used to model some of the variation in the color of pixels. This fact is extremely important in satellite imagery where the resolutions are slightly coarser. For this reason it is common in the Remote Sensing community [63] [7] [32] to employ what is know as a "linear mixture model" in which each pixel is modeled as a linear combination of a number of "pure materials" or "end-members". In many applications the actual spectral values of these materials are found through fairly extensive experimentation. In one application a satellite sensor was mounted on a bucket truck and taken to areas where relatively pure materials of interest were located in order to capture an extremely accurate spectral "signature" of the material [63]. However, in many cases such extensive calibration is not feasible. In these cases finding the pure materials must be done from the data directly. In these cases, the clustering model using derivative information described previously can be quite useful for inferring pixels that are likely suffer from less mixing effects. The availability of high-resolution data helps to mitigate this difficulty for certain classes of problems. However, higher resolution data also opens the possibility for finer fidelity sub-pixel analysis to be performed. The following sections describe the linear mixing model and a discussion of practical uses of the model is given. Then a method for solving the mixed pixel problem using a linear program is proposed as a formalization of the commonly used heuristic procedures.

8.1 The Linear Pixel-mixing Model

The linear pixel-mixing model is conceptually simple. Each pixel is modeled as consisting of a linear combination of pure reflecting substances. Figure 8.1.1 illustrates this notion for a pixel that is modeled as a mixture of soil, grass and weeds.



Observed Pixel Color | Sub-pixel Contents | More weeds in there?

Figure 8.1.1 Modeling pixel color with a linear mixing model.

Mathematically, for a given pixel the model is expressed as:

$$\vec{\mathbf{x}}_{obs} = \sum_{i=1}^{N} f_i \vec{\mathbf{m}}_i + \vec{\varepsilon}$$

Eq. 8.1.1

The observed color is modeled as the fraction of each material and an error term. It is common to fit this model by minimizing the error term. However, clearly a number of situations will occur where there will be multiple solutions with an error term of zero. It is common to heuristically try sets of materials that seem appropriate for the particular location in the image. In some cases such a minimization procedure will result in fractions that are unrealistic (i.e. less than zero or greater than one). In some schemes, the corresponding set of "pure materials" is then eliminated as a potential "explanation". If no error term is used and the number of materials is equal to the number of spectral bands, then a simple matrix inversion can be used to generate a potential explanation. Here the same problem exists with

unrealistic fractions possibly resulting from the procedure. While additionally, this approach is limited in that the number of spectral bands determines the number of materials for each possible model of the observed pixel color.

Conceptually, what we could do to avoid the problems involved with these common approaches involves "ranking" the pure materials based on our prior knowledge of the materials that we think should be in the given pixel. Then, starting with the most likely pure material try to explain the observed color using less and less likely materials until we are forced to resort to using the error term or possibly some other extremely unlikely material (there may be an old car in the field…). In the next section it is shown that this procedure can be stated more formally and then expressed as a linear program if a few reasonable assumptions are made.

8.2 Model Derivation

Consider that if we look at the observed pixel color and we know the context of the pixel in the image we can assign a probability that we believe any given material to be within that pixel. For example, these probabilities could be obtained from our previous analysis in the following way. If one inspects the subclass variables for the Gaussian mixture model for our material classes that have been found using the EM algorithm, then each subclass can be given a label. Based on the means of the edge magnitudes of these clusters one can determine if a cluster is a mixture class or a pure class. If highly calibrated spectral data for the known pure materials in the image is available, then the clusters found with the clustering algorithm

could be matched to this calibrated data. If such data is not available then the means of the color components of the Gaussians can be used as the corresponding pure material. Now, recall that each subclass cluster will have a probability associated with that cluster. This probability is an estimate of the quantity we need, namely the probability that the given "pure material" was responsible for the observed color.

Consider now that we could break up that pixel into N sub-pixels responsible for the observed pixel color. One thus has a compound sub-pixel event leading to the observed pixel value. Figure 8.2.1 illustrates the subdivision process into four sub-pixels and illustrates one potential sub-pixel configuration.

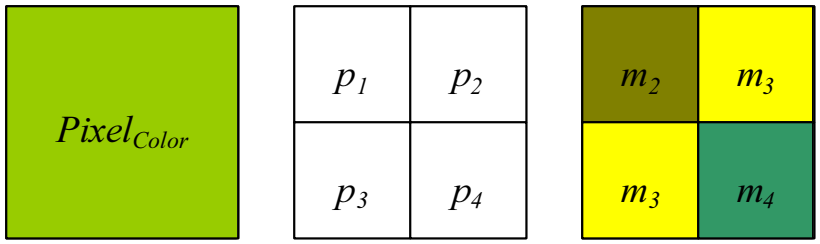


Figure 8.2.1 A given pixel color $Pixel_{Color} = \vec{\mathbf{x}}_{obs}$ (Left) can be decomposed into sub pixels (Centre), each of which could contain any of the pure material m_i selected from the set of known materials (Right).

If we assume the sub-pixel events are conditionally independent, then the probability of this event can be expressed as:

$$\begin{aligned} P(\vec{\mathbf{x}}_{conf}) &= P(m_2 \wedge m_2 \wedge m_3 \wedge m_4) \\ &= P(\vec{\mathbf{m}}_2)P(\vec{\mathbf{m}}_2)P(\vec{\mathbf{m}}_3)P(\vec{\mathbf{m}}_4) \end{aligned}$$

Eq. 8.2.1

Here events have been illustrated with the vector notation to emphasize that they are events with vector values. This calculation can be generalized to an arbitrary number of sub-pixels. If one defines a material as $\vec{\mathbf{m}}_i$ (a material vector $i = 1 \ldots M$), the set of all possible materials as $\{\vec{\mathbf{M}}\}$ (a set of vectors) and the number of sub-pixels belonging to material $\vec{\mathbf{m}}_i$ as p_i (where $\sum_i p_i = n$, the total number of sub-pixels) then the probability of a given configuration can be expressed as:

$$P(\vec{\mathbf{x}}_{conf}) = \prod_{\vec{\mathbf{m}}_i \in \vec{\mathbf{M}}} P(\vec{\mathbf{m}}_i)^{p_i}$$

Eq. 8.2.2

Further, the color of the pixel constrains the solution space of sub-pixels so that the vector addition of sub-pixels scaled by the fractional amount of the sub-pixels ($f_i = \frac{p_i}{n}$) must produce the observed pixel color. Formally, this can be written as follows:

$$\vec{\mathbf{x}}_{obs} = \sum_{i=1}^{m} f_i \vec{\mathbf{m}}_i$$

Eq. 8.2.3

The task of finding the most likely, consistent sub-pixel configuration given a set of potential material vectors can thus be expressed as:

$$\underset{p_i}{\text{argmax}} \left[P(\vec{\mathbf{x}}_{conf}) \right] = \underset{p_i}{\text{argmax}} \left[\prod_{\vec{\mathbf{m}}_i \in \vec{\mathbf{M}}} P(\vec{\mathbf{m}}_i)^{p_i} \right]$$

Eq. 8.2.4

Subject to the constraint of the previous equation. However, one equation involves fractions and the other equation involves sub-pixels. But, Eq. 8.2.2 can then be manipulated in two main ways. Firstly, one can take the logarithm of the probability function. Secondly, one can

divide the function by n and take the limit as $n \to \infty$ to find the most likely, consistent fraction f_i of a pixel for each potential material in the following manner.

$$P(\vec{\mathbf{x}}_{conf}) = \prod_{\vec{\mathbf{m}}_i \in \vec{\mathbf{M}}} P(\vec{\mathbf{m}}_i)^{p_i}$$

$$log\{P(\vec{\mathbf{x}}_{conf})\} = log\left\{\prod_{\vec{\mathbf{m}}_i \in \vec{\mathbf{M}}} P(\vec{\mathbf{m}}_i)^{p_i}\right\}$$

$$log\{P(\vec{\mathbf{x}}_{conf})\} = \sum_{\vec{\mathbf{m}}_i \in \vec{\mathbf{M}}} log\left\{P(\vec{\mathbf{m}}_i)^{p_i}\right\}$$

$$log\{P(\vec{\mathbf{x}}_{conf})\} = \sum_{\vec{\mathbf{m}}_i \in \vec{\mathbf{M}}} p_i \, log\{P(\vec{\mathbf{m}}_i)\}$$

$$log\{P(\vec{\mathbf{x}}_{conf})\} = \sum_{i=1}^{M} p_i \, log\{P(\vec{\mathbf{m}}_i)\}$$

$$\lim_{n\to\infty}\left[\frac{log\{P(\vec{\mathbf{x}}_{conf})\}}{n}\right] = \lim_{n\to\infty}\left[\sum_{i=1}^{M}\frac{p_i}{n} log\{P(\vec{\mathbf{m}}_i)\}\right]$$

$$\lim_{n\to\infty}\frac{log\{P(\vec{\mathbf{x}}_{conf})\}}{n} = \sum_{i=1}^{M} f_i \, log\{P(\vec{\mathbf{m}}_i)\}$$

$$\underset{f_i}{\text{argmax}}\left\{\lim_{n\to\infty}\frac{log\{P(\vec{\mathbf{x}}_{conf})\}}{n}\right\} = \underset{f_i}{\text{argmax}}\left\{\sum_{i=1}^{M} f_i \, log\{P(\vec{\mathbf{m}}_i)\}\right\}$$

Eq. 8.2.5

8.3 Sub-pixel Inference as a Constraint Satisfaction Problem

More specifically, the approach to finding a consistent explanation for the observed pixel color described in the previous section is equivalent to a standard linear optimization problem of the following form: For our M independent fractional variables $f_i,\ldots,f_M$, maximize the function z in which $a_{0i} = log\{P(\vec{\mathbf{m}}_i)\}$ and

$$z = a_{01} f_1 + a_{02} f_2 + \ldots + a_{0M} f_M$$

Eq. 8.3.1

subject to the primary constraints

$$f_1 \geq 0, f_2 \geq 0, \ldots, f_M \geq 0$$

Eq. 8.3.2

simultaneously subject to $N = n_1+n_2$ *additional constraints*, $n_1 = M$ of them of the form

$$f_i \leq 1, f_2 \leq 1, \ldots, f_M \leq 1$$

Eq. 8.3.3

(these constraints can be removed in practice as they are implicitly specified by the following constraint) and n_2=1+b, (where b is the number of spectral bands) such that the first equation is of the form

$$f_1 + f_2 + \ldots + f_M = 1$$

Eq. 8.3.4

and the remaining b constraints take the form

$$\begin{aligned}
&\vec{\mathbf{m}}_{11} f_1 + \vec{\mathbf{m}}_{21} f_2 + \ldots + \vec{\mathbf{m}}_{M1} = \vec{\mathbf{x}}_{obs_1} \\
&\vec{\mathbf{m}}_{12} f_1 + \vec{\mathbf{m}}_{22} f_2 + \ldots + \vec{\mathbf{m}}_{M2} = \vec{\mathbf{x}}_{obs_2} \\
&\vdots \\
&\vec{\mathbf{m}}_{1b} f_1 + \vec{\mathbf{m}}_{2b} f_2 + \ldots + \vec{\mathbf{m}}_{Mb} = \vec{\mathbf{x}}_{obs_b}
\end{aligned}$$

Eq. 8.3.5

In such problems, a set of values for the fractions $f_1 \ldots f_M$ that satisfy the constraints of (Eq. 8.3.2) – (Eq. 8.3.5) is referred to as a *feasible vector*. The function we are trying to maximize (i.e. Eq. 8.2.5) is referred to as the *objective function*. A feasible vector that maximizes the objective function is referred to as the *optimal feasible vector* [59]. Solutions to linear programs of this form are well known and a technique known as the *simplex method* first published by Dantzig [17] provides a good and very commonly used solution [59].

Thus, using our contextual knowledge the problem of finding consistent explanations for a given pixel can be framed as a simple linear program. This formulation provides a very

straightforward technique for finding consistent explanations allowing arbitrary combinations of sub-pixel materials and it not limited by the number of spectral bands.

9 Conclusions

This investigation has taken a probabilistic approach to image analysis. For each type of feature investigated in this thesis an appropriate linear transformation is first developed and then probability theory is used to classify these features. For color the CIE xyY system is used. For texture, the 4^{th} order Daubechies wavelet transformation is used. For geometric line relationships, an appropriate coordinate system is derived. An algorithm for image segmentation based on probabilistic message passing has been proposed and some results of its operation have been presented. The algorithm incorporates color, the local context of pixels, texture and the local context of textured areas. This algorithm allows homogeneous regions in the image to be further segmented into curves and lines. Perceptually relevant relationships between lines useful for further analysis are then evaluated using a probabilistic framework. Sub-pixel inference is then posed as a problem that can be solved with a linear program. The importance of integrating these features to generate a realistic image interpretation is demonstrated. Aspects of human visual perception are formulated mathematically and then used to solve real problems within the interpretation process. While, the formal methods for specifying the intermediate steps in the interpretation procedure within this investigation are grounded heavily in statistical decision theory. Finally, concrete results illustrating the utility of this approach have been presented.

The construction of very large software systems that use probability theory extensively is a relatively new endeavor. This thesis advocates the use of probability calculus for the management of uncertain reasoning procedures within relatively large software systems.

Many other technologies exist for managing uncertainty that were not discussed in detail within this thesis (e.g. Fuzzy logic, Artificial Neural Networks, etc…). However, probability theory is attractive in that it provides a principled formalism with a strong theoretical foundation. Further, the notion of probabilistic message passing is extremely useful in that it provides an intuitive visualization of how uncertain information is managed in a large system. Clearly, the construction of large software systems that must manage uncertainty deals with issues that have been investigated by both statisticians and software engineers. Thus, if probability theory is to be used to manage uncertain information within a large system, a developer of such a system must have a strong knowledge of statistical theory in addition to a good understanding of different types of statistical models and algorithms. Further, such a developer must be aware of many of the issues that have in the past been associated with software engineering. Many of the frontier applications associated with Artificial Intelligence (e.g. speech recognition, natural language understanding, computer vision and automated image interpretation, computational biology applications etc…) involve the construction of rather large systems that must function under uncertain conditions. For this reason, further research into the ways in which statistical decision theory can be incorporated into an "intelligent system" have great potential to allow such applications to be created in the future. This thesis has focused on an image interpretation application that has direct application to the monitoring and protection of the environment. It is likely that future progress in our ability to construct other such useful "AI" applications will benefit greatly from further research into the algorithms, modeling techniques and software engineering issues involved with integrating statistically sound reasoning procedures within a larger software system.

Bibliography

[1] Basseville, M. et al. (1992) Modeling and estimation of multiresolution stochastic processes. *IEEE Transactions on Inform. Theory*, vol. 38, pp. 766-784.

[2] Bell, A.J., Sejnowski, T.J. (1997) Edges are the 'Independent Components' of Natural Scenes. *Proc. Advances in NIPS 1996*, vol. 9, pp. 831.

[3] Beymer, D., (1995) Vectorizing Face Images by Interleaving Shape and Texture Computations. *MIT AI Memo* 1537.

[4] Bishop, C.M. (1995) *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

[5] Bouman, C. A. (1995) *CLUSTER: An unsupervised algorithm for modeling Gaussian mixtures*. Technical Report. Purdue University. http://www.ece.purdue.edu/~bouman/software/cluster/manual.pdf

[6] Bouman, C. A. and Shapiro, M. (1994) A Multiscale random field model for Bayesian Image segmentation. *IEEE Transactions on Image Processing*, vol. 3 pp. 162-177, December.

[7] Bouzidi, S., Berroir J.P. and Herlin, I.L. (1997) Subpixel mixture modelling applied for vegetation monitoring. *Environmental Software Systems*, vol. 2, pp. 41-48.

[8] Brunn, A., Gulch, E., Lang, F. and Forstner, W. (1998) A hybrid concept for 3D building acquisition. *ISPRS Journal of Photogrammetry & Remote Sensing* vol. 53, pp. 119-129.

[9] Burt, P. J. and Adelson, E. H. (1983) The Laplacian pyramid as a compact image code. *IEEE Tans. Comput*., vol. COM-31, pp. 532-540.

[10] Canny, J. A (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698.

[11] Cook, G. W. and Delp, E. J. (1995) Multiresolution Sequential Edge Linking. *Proc. International Conference on Image Processing.*

[12] Cook, G.W and Delp, E. J. (1995) Multiresolution Sequential Edge Linking. *International Conference on Image Processing.*

[13] Cooper, G.F. (1990) The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence.* vol. 42, 393-405.

[14] Covell, M. and Bregler, C., Eigen-Points (1996) *Proc. 3rd IEEE International Conference on Image Processing*. Lausanne, Switzerland 16-19, 1996. Pp. 471-4 vol. 3.

[15] Cross, G.R. and Jain, A.K. (1983) Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 1, January.

[16] Danielsson, P.E. (1980) Rotationally-invariant linear operators with directional response. *Proc 5th International Conference on Pattern Recognition* (Miami), pp. 1171-1176

[17] Dantzig, G. B. (1963), *Linear Programming and Extensions*. Princeton University Press: Princeton, New Jersey.

[18] Das, S. and Bhanu, B. (1998) A System for Model-Based Object Recognition In Perspective Aerial Images. *Pattern Recognition*, vol. 31. No. 4 pp. 465-491.

[19] Dempster, A. P., Laird, M., and Rubin D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society*, vol. B-39, pp. 1-38, 1977.

[20] Deng, Li. (1999) Personal Communication.

[21] Eichel, P. and Delp, E. (1985) Sequential Edge Detection In Correlated Random Fields. *Proc. IEEE Computer Vision and Pattern Recognition*. pp. 14-21

[22] Forgy, C. L. (1982) Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence* vol. 19, pp. 17-37.

[23] Freedman, M.H. and Press, W.H. (1996) Truncation of Wavelet Matrices: Edge Effects and the Reduction of Topological Control. *Linear Algebra and Its Applications.* vol. 234 pp. 1-19.

[24] Gamba, P. and Fabio, C. (1998) GIS and Image Understanding for Near-Real-Time Earthquake Damage Assessment. *Photogrammetric Engineering and Remote Sensing,* vol. 64, no. 10, pp. 987-994, October.

[25] Geman, S. and Geman, D. (1984) Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, November.

[26] Guy, G., and Medioni, G. (1993) Inferring global perceptual contours from local features. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 786-787.

[27] Harvey, W.A. and Tambe, M. (1993) Experiments in Knowledge Refinement for a Large Rule-Based System. *Technical Report CMU-CS-93-195*. School of Computer Science Carnegie Mellon University Pittsburg, PA 15213.

[28] Hayes-Roth, F. (1985) Rule-based systems, *Communications of the ACM*, vol. 28, pp. 921-932, September.

[29] Heckerman, D. Horvitz, E. and Nathwani, B. (1992) Towards normative expert systems. 1: The PATHFINDER project. *Methods of Information in Medicine*, vol. 31, pp. 90-105.

[30] Huang, L. T. (1996) A Visual Expert System for Atmospheric Pollution Modelling Using Causal Probabilistic Networks. M.Sc. Thesis, University of Guelph.

[31] Hubel, D.H. and Wiesel, T.N. (1968) Receptive fields and functional architecture of monkey striate cortex, *J. Physiol.,* 195, pp. 215-244.

[32] Huguenin, R., L., Karaska, M., A., Donald, V.B., Jensen, J.R. (1997) *Subpixel Classification of Bald Cypress and Tupelo Gum Trees in Thematic Mapper Imagery.* Vol. 63, no. 6, pp.717-725.

[33] Irving, W.W., Feiguth, P.W. and Willsky, A.S. (1997) An Overlapping Tree Approach to Multiscale Stochastic Modeling and Estimation. *IEEE Transactions on Image Processing*, vol. 6 no. 11 November.

[34] Ising, E. *Zeitschrift Physik*, vol. 31, p 253, 1925.

[35] ITU-R Recommendation BT. 709, (1990) Basic Parameter Values for the HDTV Standard for the Studio and for International Programme Exchange, [formerly CCIR Rec. 709] (Geneva: ITU, 1990)

[36] Jelinek, F. (1969) A fast sequential decoding algorithm using a stack. *IBM J. Res. and Dev*., vol. 13, pp. 675-685, November.

[37] Jensen, F. V. (1993) *Introduction to Bayesian Networks.* Hugin Expert A/S, Denmark.

[38] Kass, M., Witkin, A., Terzopoulous, D. (1987) Snakes, Active Contour Models. *Proc. International Conference on Computer Vision.*

[39] Knutsson, H., Wilson, R. and Granlund, G. H. (1983) Estimating the local orientation of anisotropic 2-d signals. Proc. *IEEE ASSP Workshop Spect. Est.* (Tampa) pp. 234-239.

[40] Kschischang, F. R. and Frey, B. J. (1998) Iterative Decoding of Compound Codes by Probability Propagation in Graphical Models. *IEEE Journal on Selected Areas in Communication* vol. 16, no. 1 January.

[41] Kschischang, F., Frey, B., and Loelinger H. Factor Graphs and the Sum-Product Algorithm. Submitted to: *IEEE Transactions on Information Theory*, (July 1998).

[42] Lakshmanan, S. and Derin, H. (1989) Simultaneous Parameter Estimation and Segmentation of Gibbs Random Fields Using Simulated Annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, August.

[43] Lauritzen, F. V. and Spiegelhalter, D. J. (1988) Local computations with probabilities on graphical structures and their applications to expert systems (with discussion). *Journal of the Royal Statistical Society series B*, **50**, 157-224. [Reprinted in Shafer and Pearl (1990)]. 2, 11, 15

[44] Lowe, D.G. (1987)Three – dimensional object recognition from single two – dimensional images, *Artificial Intelligence*, vol. 31, pp. 355-395.

[45] Massey, J. (1972) Variable length codes and the Fano metric. *IEEE Transactions on Information Theory*, vol. IT-18. pp. 196-198, January.

[46] McKeown, D. M., Harvey, W. A. and McDermott J. (1985) Rule based interpretation of aerial imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5):570-585, September.

[47] McKeown, D. M., Harvey, W. A. and Wixson, L. (1989) Automated knowledge acquisition for aerial image interpretation *Computer Vision, Graphics and Image Processing*, 46(1):37-81, April.

[48] McKeown, D.M. (1996) Top Ten Lessons Learned In Automated Cartography. *Proceedings of the 1996 ARPA Image Understanding Workshop*, February 12-15, Palm Springs CA.

[49] McKeown, D.M., Harvey, W.A., Polis, M.F., Bulwinkle, G. E., McGlone, C., Cochran, S.D., McMahill, J. and Shufelt, J. (1998) Research in Image Understanding and Automated Cartography 1997-1998. *Proceedings of the DARPA Image Understanding Workshop,* Monterey California, 20-23 November.

[50] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953) Equations of state calculations by fast computing machines. *J. Chem. Phys.,* vol. 21, pp. 1087-1091.

[51] Meyer, P., Staenz, K., Itten, K.I. (1996) Semi-automated procedures for tree species identification in high spatial resolution data from digitized color infrared-aerial photography. *ISPR Journal of Photogrammetry & Remote Sensing* 51, 5-16.

[52] Narendra, K.S. and Thatcher, M.L. (1989) *Learning Automata: An introduction.* Prentice Hall.

[53] Neider, J., Davis, T. and Woo, M. (1996) *OpenGL Programming Guide, The Official Guide to Learning OpenGL, Release 1*. Addison-Wesley Publishing Company.

[54] Nilsson, N. (1971) *Problem-Solving Methods in Artificial Intelligence*, N.Y.: McGraw-Hill.

[55] Pal, C. (1998) A Technique for Illustrating Dynamic Component Level Interactions Within a Software Architecture. *Proceedings of CASCON 1998*, The IBM Centre for Advanced Studies Conference, pp. 134-146.

[56] Pal, C., Slaney, M., Adams, R. L. (1999) *Sound Based Event Control Using Timbral Analysis*. United States Patent, Notice of Acceptance Received 1999.

[57] Pearl, J. (1986) A constraint-propagation approach to probabilistic reasoning. *Uncertainty in Artificial Intelligence*, editors, L. M. Kanal and J. Lemmer pp. 357-370. North-Holland, Amsterdam.

[58] Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Series in Representation and Reasoning.* Morgan Kaufmann.

[59] Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. (1992) *Numerical Recipes in C*. Second edition. pp. 431. Cambridge: Cambridge University Press. Reprint with corrections (1995).

[60] Publication CIE No. 15.2, (1986), *Colorimetry, Second Edition* (Vienna, Austria: Central Bureau of the Commission Internationale de L'Eclairage)

[61] Quian, R.J. and Huang, T. S. (1997) Object Detection Using Hierarchical MRF and MAP Estimation. *Proc. of IEEE Computer Vision and Pattern Recognition CVPR*, pp. 186-192.

[62] Ripley, B.D. (1996) *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press.

[63] Roberts, D.A, Gardner, M., Church, R., Ustin, S.L. and Green, R.O. (1997) Optimum Strategies for Mapping Vegetation using Multiple Endmember Spectral Mixture Models. *SPIE Conference on Imaging Spectrometry III*, vol. 3118, pp. 12, San Diego, CA July 27-Aug 1.

[64] Rock, I., Palmer, S., (1990) The legacy of Gestalt psychology, *Scientific American*, pp. 84-90 December.

[65] Rosin, P., West, G. A.W. (1995) Nonparametric Segmentation of Curves into Various Representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 17, no. 12.

[66] Sarkar, S. (1999) Fast Detection of Large Salient Groupings of Extended Low Level Image Features: Graph Spectral Partitioning and Learning Automata. *Manuscript in Progess.*

[67] Sarkar, S. and Boyer, K. (1994) A Computational Structure for Preattentive Perceptual Organization: Graphical Enumeration and Voting Methods. *IEEE Transactions on Systems Man, and Cybernetics*, vol. 24, no. 2, February.

[68] Sarkar, S. and Boyer, K. (1999) Personal Communication.

[69] Shachter, R.D. Evaluating influence diagrams. *Operations Research*, vol. 34, pp. 871-882.

[70] Steger, C., Mayer, H. and Radig, B. (1997) The Role of Grouping for Road Extraction. In: *Automatic Extraction of Man-Made Objects in Aerial and Space Images*, Birkhauser Verlag, Basel, Switzerland, pp. 265-274.

[71] Thiesson, B., Meek, C., Chickering, D.M., and Heckerman, D., Learning Mixtures of DAG Models. *Proc. AAAI 1999*, pp. 504-513.

[72] Torre, V. and Poggio, T. A. (1986) On edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2. pp. 147-163.

[73] Wilson, R. and Bhalerao, A. H. (1992) Kernel Designs for Efficient Multiresolution Edge Detection and Orientation Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, no. 3, pp. 384-389.

[74] Wong, A.K.C. and You, M. (1985) Entropy and Distance of Random Graphs with Applications to Structural Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. PAMI-7, no. 5 pp. 599-609 September.

[75] Wyszecki, G., and Styles, W.S., (1982) *Color Science: Concepts and Methods, Quantitative Data and Formulae,* Second Edition. New York: John Wiley & Sons.

[76] Yen, Shih-Chen, Leif, H.F., (1997) Identification of Salient Contours in Cluttered Images. *Proc. IEEE Computer Vision and Pattern Recognition CVPR.* pp. 273-278.

[77] Zucker, S. W., David, C., Dobbins, A., and Iverson, L. (1989) The organization of curve detection: coarse tangent fields and fine spline covering. *International Conference on Computer Vision*, 2:1-10.

Appendix I. Markov Random Fields and Probability Propagation

This appendix discusses the relationship between MRFs and Gibbs-MRFs. It also illustrates the probability propagation scheme used in this thesis and shows how the factor graph notation can be used to illustrate the commonly used MRF neighborhood systems.

A strictly positive Markov random field is specified with a potential representation consisting of positive functions ϕ_q such that

$$P(X_1, X_2, \ldots, X_n) \propto \prod_{q \in Q} \phi_q(X_q)$$

Eq. 8.3.6

Where X_i represents the variables in the graph (e.g. for an image these are the pixel sites) and X_q represents the variables in each of the cliques Q in the graph. To construct a valid distribution for the joint probability a normalization constant Z also known as the partition function is used such that

$$P(X_1, X_1, \ldots, X_n) = \frac{\prod_{q \in Q} \phi_q(X_q)}{\sum_{\text{all m}} \prod_{q \in Q} \phi_q(X_q)}$$

Eq. 8.3.7

To simplify our notation, let X_G represent all the variables in the graph. Now, a MRF can be specified with a Gibbs distribution and takes the following form:

$$P(X_1, X_1, \ldots, X_n) = \frac{1}{Z} exp\left(-\frac{1}{T} U(X_G)\right)$$
$$= \frac{exp\left(-\frac{1}{T}\sum_{q \in Q} V_q(X_q)\right)}{\sum_{\text{allm}} \left\{ exp\left(-\frac{1}{T}\sum_{q \in Q} V_q(X_q)\right)\right\}}$$
$$= \frac{\prod_{q \in Q} exp\left(-\frac{1}{T} V_q(X_q)\right)}{\sum_{\text{allm}} \left\{ \prod_{q \in Q} exp\left(-\frac{1}{T} V_q(X_q)\right)\right\}}$$

Eq. 8.3.8

One can thus see that the potential functions used in this investigation are related to the potentials that must be specified in a Gibbs-MRF in the following way

$$V_q(X_q) = -T\, log(\phi_q(X_q))$$

Eq. 8.3.9

Probabilistic inference in a MRF is commonly performed with sampling procedures based on stochastic relaxation. However, in contrast a scheme based on probabilistic message passing can be used based on the procedures described in Chapter 2. Using these techniques, the equation form of the computation for updating the probability $P^*(X_{x,y})$ of a pixel, window location or lattice site at location x,y being in each of its possible states (e.g. material classes $M_{x,y}$), given messages from the surrounding cliques in the graph can be written as follows.

$$
\begin{aligned}
P^*(M_{x,y}) &= \frac{\text{prior} \times \text{likelihood}}{\text{normalizat ion}} \\
&= \frac{\text{prior} \times \text{likelihood messages}}{\text{normalizat ion}} \\
&= \frac{P(M_{x,y}) \prod_{q \in Q,\, \text{s.t.} M_{x,y} \in M_q} \left[\sum_{M_{i,j} \in M_q,\, \text{s.t.}(i,j) \neq (x,y)} \phi_q^*(M_q) \right]}{\sum_m \left\{ P(M_{x,y}) \prod_{q \in Q,\, \text{s.t.} M_{x,y} \in M_q} \left[\sum_{M_{i,j} \in M_q,\, \text{s.t.}(i,j) \neq (x,y)} \phi_q^*(M_q) \right] \right\}} \\
&= \frac{P(M_{x,y}) \prod_{q \in Q,\, \text{s.t.} M_{x,y} \in M_q} \left[\sum_{M_{i,j} \in M_q,\, \text{s.t.}(i,j) \neq (x,y)} \left\{ \phi_q(M_q) \prod_{M_{i,j} \in M_q,\, \text{s.t.}(i,j) \neq (x,y)} \frac{P^*(M_{i,j})}{P(M_{i,j})} \right\} \right]}{\sum_m \left\{ P(M_{x,y}) \prod_{q \in Q,\, \text{s.t.} M_{x,y} \in M_q} \left[\sum_{M_{i,j} \in M_q,\, \text{s.t.}(i,j) \neq (x,y)} \left\{ \phi_q(M_q) \prod_{M_{i,j} \in M_q,\, \text{s.t.}(i,j) \neq (x,y)} \frac{P^*(M_{i,j})}{P(M_{i,j})} \right\} \right] \right\}}
\end{aligned}
$$

Eq. 8.3.10

Finally, the neighborhood systems used in the Remote Sensing community for GMRF models could thus be written as factor graphs in the following form.

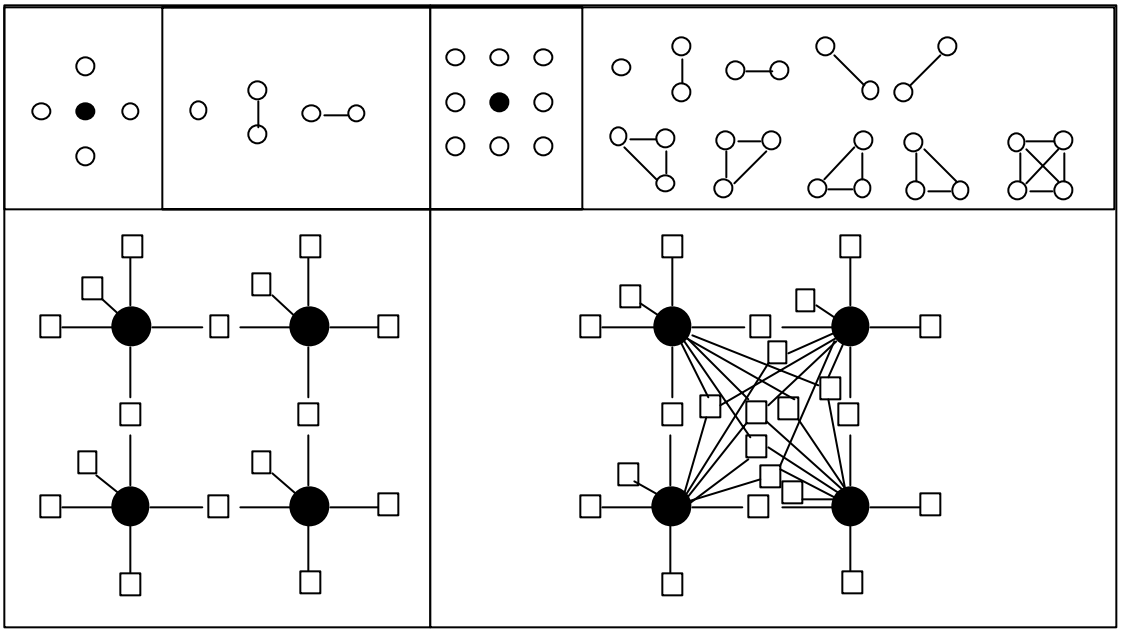


Figure 8.3.1 Top: MRF cliques for 1st and 2nd order systems Bottom: MRF lattices illustrated using the factor graph notation with variables as circles and functions as rectangles.

Appendix II. A Coordinate System for Line Relationships

Consider two lines l_s and l_o. Denote the length of *one half* the longer line as L_o and the length of *one half* of the shorter line as L_s. Place a Cartesian coordinate system at the center of each line and calculate the absolute orientation of the line as being the angle made by the half of the line extending in the positive y-quadrants. Use the angle in the range $[0,\pi)$ counter clockwise (i.e. a right handed coordinate system) from the positive x-axis of the Cartesian plane. In this convention, lines are considered as *oriented up* with respect to their midpoints. Perfectly horizontal lines have an absolute orientation angle of zero. Determine the longer line and the shorter line, selecting randomly when they are of equal length. Define the *relative orientation of the smaller line with respect to the larger line* θ_{rel} , as the angle from the shorter line to the longer line in the positive y-quadrants i.e. $\theta_{rel} = \theta_o - \theta_s$. This measurement can thus take on a positive or negative values. A calculation is illustrated for a negative θ_{rel} in Figure 8.3.2.

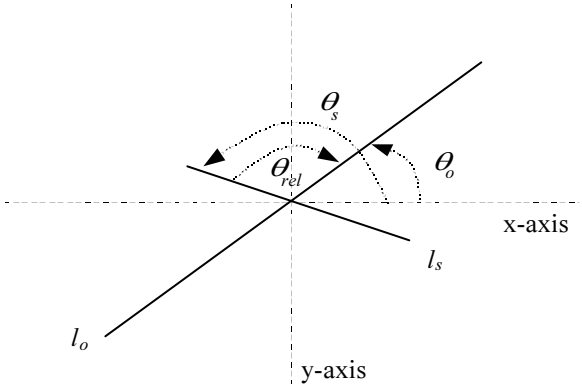


Figure 8.3.2 A line orientation calculation.

Draw a *vector* $\vec{\mathbf{V}}_m$ from the midpoint of the longer line to the midpoint of the smaller line and call the *length* of this vector L_m. Calculate the absolute orientation of this vector θ_m from

the positive x-axis (note: it shall lie in the range $[0,2\pi]$ because it is a vector). Find the vector projection $\vec{\mathbf{V}}_{\text{proj}}$ of the midpoint vector onto the (possibly extended) longer line. For this projection use the angle from the positive y-quadrant half of l_o to the midpoint vector $\vec{\mathbf{V}}_{\text{m}}$. Measure this angle in the usual way in that counter-clockwise is positive (i.e. use a right-handed coordinate system). Angles in the range $[-\pi,\pi)$ shall be used. Call this angle θ_{proj}. In this scheme, $\theta_{proj} = \theta_m - \theta_o$. The angles involved in the projection are illustrated in Figure 8.3.3 and note the notion of lines being oriented up determines how θ_{proj} is determined and thus the sign of θ_{proj}.

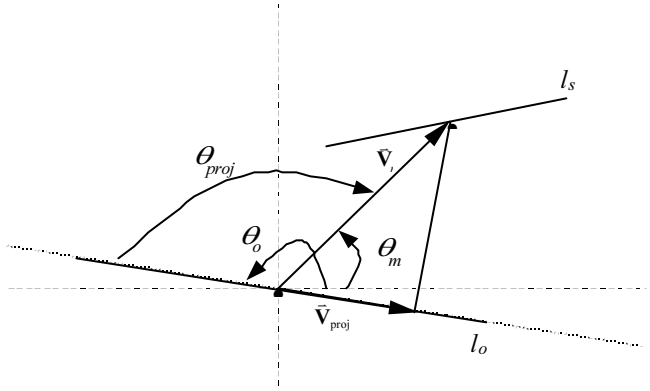


Figure 8.3.3 The projection of the midpoint vector onto the longer line.

In such a system, *positive distance along the longer line* represents going up from the midpoint. Define the length along the vector projection from the midpoint of the longer line as the *Linear distance* L_d (it can be positive or negative). Define the *Perpendicular distance* P_d as the distance from the end of the projection on the longer line to the midpoint of the smaller line. Keep in mind that this measurement can be positive or negative because under the current definitions, measuring θ_{proj} from the oriented l_o effectively defines a coordinate system with respect to l_o. The coordinate system looks like the standard Cartesian plane when

the longer line is horizontal and then rotates as the line rotates. *Pd* and *Ld* are thus found from the following simple trigonometric calculations.

$$P_d = \left|\vec{\mathbf{V}}_{proj}\right| sin(\theta_{rel})$$
$$L_d = \left|\vec{\mathbf{V}}_{proj}\right| cos(\theta_{rel})$$

Eq. 8.3.11

Figure 8.3.4 illustrates how the lines of Figure 8.3.3 would result in negative values for both P_d and L_d. Intuitively, the coordinate system defined by l_o has been rotated almost upside down from the standard orientation as θ_o, (the absolute orientation of the longer line) is almost π.

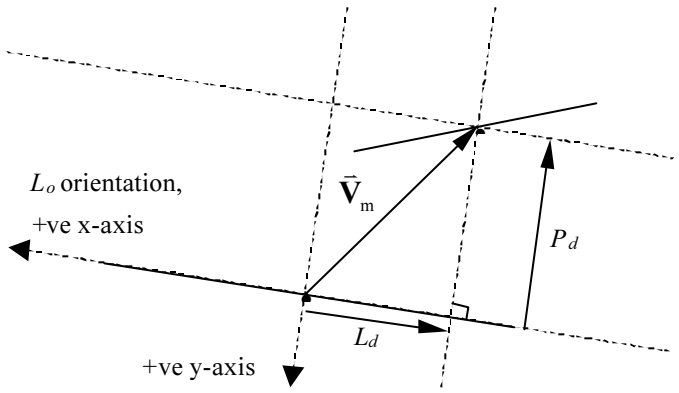


Figure 8.3.4 An example calculation of P_d and L_d.

Finally, normalize all the length measurements by one half the length of the longer line i.e. L_o. All the relationships between two lines, up to a translation, rotation or shift of scale can be specified using the following four measurements: P_d, L_d, L_s and θ_{rel}. This measurement system thus encodes all of the four relevant degrees of freedom.