

STATISTICAL MODELING AND SEGMENTATION  
OF SKY/GROUND IMAGES

By

SINISA TODOROVIC

A THESIS PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2002

## ACKNOWLEDGMENTS

Herewith, I would like to express my sincere gratitude to Dr. Michael Nechyba for his wise guidance of my research for this thesis. As my advisor, Dr. Nechyba has been directing but on no account confining my interests. Therefore, I have had the impression of complete freedom, which, in my opinion, is a prerequisite for fruitful scientific research. I especially appreciate his readiness and expertise to help me solve numerous implementation issues. Most importantly, I am thankful for the friendship that we have developed collaborating on this work.

Also, I thank Dr. Antonio Arroyo, whose brilliant lectures on machine intelligence have inspired me to endeavor research in the field of robotics. As the director of the Machine Intelligence Lab (MIL), Dr. Arroyo has created a warm, friendly, and hard working atmosphere among the “MIL-ers.” Thanks to him, I have decided to join the MIL, which has proved on numerous occasions to be the right decision. Therefore, I thank Dr. Arroyo and all the members of the MIL for helping me assimilate faster to the new environment.

I thank Dr. Peter Ifju and his team of students at the Aerospace Engineering Department for enabling me to participate in their micro air vehicle (MAV) project. My research has been initially inspired by the mutual work of Dr. Nechyba and Dr. Ifju. Certainly, the theory developed in the thesis would have been futile, had there not been the perfect application—MAV.

Finally, special thanks go to Ashish Jain for innumerable interventions in cases in which the Linux operating system has been my top adversary.

## TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS . . . . .	ii
LIST OF TABLES . . . . .	v
LIST OF FIGURES . . . . .	vi
KEY TO ABBREVIATIONS . . . . .	vii
KEY TO SYMBOLS . . . . .	viii
ABSTRACT . . . . .	x
1 INTRODUCTION . . . . .	1
1.1 Overview of the Proposed Computer Vision System . . . . .	2
1.2 Overview of the Thesis . . . . .	3
2 FEATURE SPACE . . . . .	5
2.1 Color . . . . .	5
2.2 Texture . . . . .	5
2.2.1 Wavelet Transform . . . . .	6
2.2.2 Wavelet Properties . . . . .	6
2.2.3 Complex Wavelet Transform . . . . .	9
2.2.4 Color Multi-resolution Representation . . . . .	12
3 STATISTICAL MODELING . . . . .	13
3.1 Marginal Density . . . . .	13
3.2 Graph Models . . . . .	14
3.3 HMT Model . . . . .	15
3.4 HMT Properties . . . . .	16
4 TRAINING OF THE HMT MODEL . . . . .	18
4.1 Implementation of the EM Algorithm . . . . .	18
4.1.1 E Step . . . . .	20
4.1.2 Up Step . . . . .	20
4.1.3 Down Step . . . . .	21
4.1.4 M Step . . . . .	21
4.2 Implementation Issues . . . . .	22

	4.2.1 Scaled Up Step . . . . .	22
	4.2.2 Scaled Down Step . . . . .	24
5	MULTISCALE SEGMENTATION . . . . .	26
6	EXPERIMENTAL RESULTS . . . . .	30
	6.1 Real-Time Realization Problems . . . . .	30
	6.2 Classification of the First Type of Test Images . . . . .	32
	6.3 Segmentation of the Second Type of Test Images . . . . .	33
7	CONCLUSION . . . . .	36
	REFERENCES . . . . .	38
	BIOGRAPHICAL SKETCH . . . . .	40

## LIST OF TABLES

<u>Table</u>		<u>page</u>
2.1	Coefficients of the filters used in the Q-shift DTCWT. . . . .	10
6.1	The performance of the RGB-based classifier for ground test images. .	32
6.2	The performance of the RGB-based classifier for sky test images. . . .	32
6.3	Performance of the wavelet-based classifier. . . . .	33

## LIST OF FIGURES

<u>Figure</u>		<u>page</u>
1.1	A computer vision system . . . . .	2
2.1	Two levels of the DWT of a two-dimensional signal. . . . .	7
2.2	The original image (left) and its two-scale dyadic DWT (right). . . . .	7
2.3	The Q-shift Dual-Tree CWT. . . . .	9
2.4	The CWT is strongly oriented at angles $\pm 15^\circ, \pm 45^\circ, \pm 75^\circ$ . . . . .	11
3.1	The HMT of the three-level CWT. . . . .	14
5.1	The context vector. . . . .	28
6.1	Training images of the sky and ground. . . . .	31
6.2	The classification error using RGB and HSI color spaces. . . . .	33
6.3	A MAV's flight: favorable sky/ground patterns. . . . .	34
6.4	Water surface with ice patches similar to clouds. . . . .	34
6.5	A mountain view with fuzzy horizon line. . . . .	34
6.6	Segmentation of cut and uncut grass regions. . . . .	35

## KEY TO ABBREVIATIONS

*B*: The blue feature of the RGB color space

CWT: Complex Wavelet Transform

DWT: Discrete Wavelet Transform

*G*: The green feature of the RGB color space

*H*: The hue feature of the HSI color space

HMT: Hidden Markov Tree

HSI: The color space that consists of hue, saturation and intensity features

*I*: The intensity feature of the HSI color space

MAP: Maximum A Posteriori

MAV: Micro Air Vehicle

pdf: Probability density function

*R*: The red feature of the RGB color space

RGB: The color space that consists of red, green and blue features

*S*: The saturation feature of the HSI color space

## KEY TO SYMBOLS

The list shown below gives a brief description of the major mathematical symbols defined in this work. For each symbol, the page number corresponds to the place where the symbol is first used.

$\alpha_{i,\rho(i)}^J(m, n)$ , The transition probability that the node $i$ at the scale $J$ is in a state $m$ , given that its parent node is in a state $n$ . . . . .	16
$\alpha_i^J(m)$ , A joint probability of the tree under the root node 0 with the removed sub-tree under the node $i$ at the scale $J$ and that the node $i$ is in the state $m$ . . . . .	18
$\hat{\alpha}_i^J(m)$ , The dynamically scaled $\alpha_i^J(m)$ value. . . . .	23
$\beta_i^J(m)$ , A probability of the whole tree under the node $i$ at the scale $J$ , given that the node $i$ is in a state $m$ . . . . .	18
$\hat{\beta}_i^J(m)$ , The dynamically scaled $\beta_i^J(m)$ value. . . . .	23
$\beta_{\rho(i),i}^J(m)$ , A probability of the whole tree under the node $i$ at the scale $J$ , given that the parent node $\rho(i)$ is in a state $m$ . . . . .	18
$\beta_{\rho(i)\setminus i}^{J+1}(m)$ , A probability of the cut tree under the parent node $\rho(i)$ at the scale $J + 1$ with all the children nodes removed, given that the node $\rho(i)$ is in a state $m$ . . . . .	18
$\chi(i)$ , Children nodes at the finer scale $J - 1$ of the node $i$ at the scale $J$ . . . . .	15
$\mathbf{c}_i^J$ , The context vector of the coefficient $i$ at the scale $J$ . . . . .	28
$g(\mathbf{W}^J, \mathbf{\Omega}^J)$ , The discriminant function. . . . .	27
$J$ , The number of the scale. . . . .	15
$M$ , The number of states in the Gaussian mixture model. . . . .	13
$\mu_i^J$ , The mean value of the random variable $\mathbf{w}_i^J$ . . . . .	16
$\omega_i^J$ , The class label of the coefficient $i$ at the scale $J$ . . . . .	27
$\mathbf{\Omega}^J$ , The collection of all class labels at the scale $J$ . . . . .	27



$\rho(i)$ , A parent node at the coarser scale $J + 1$ of the node $i$ at the scale $J$ . . . . .	15
$\mathbf{S}_{\chi^{(i)}}^{J-1}$ , A random variable representing the state of a child node at the scale $J - 1$ . . . . .	16
$\sigma_i^J(m)$ , The standard variation of the random variable $\mathbf{w}_i^J$ . . . . .	16
$\mathbf{S}_i^J$ , A random variable representing the state of a coefficient at the position $i$ and at the scale $J$ . . . . .	16
$\mathbf{S}_{\rho^{(i)}}^{J+1}$ , A random variable representing the state of a parent node at the scale $J + 1$ . . . . .	16
$\Theta$ , A random vector that consists of all parameters, which determine the HMT model. . . . .	16
$\Theta^g$ , All parameters of the statistical model for the ground. . . . .	26
$\Theta^s$ , All parameters of the statistical model for the sky. . . . .	26
$W_{HH}$ , The set of wavelet coefficients of the DWT obtained filtering rows with highpass and columns with highpass filters. . . . .	6
$W_{HL}$ , The set of wavelet coefficients of the DWT obtained filtering rows with highpass and columns with lowpass filters. . . . .	6
$W_{LH}$ , The set of wavelet coefficients of the DWT obtained filtering rows with lowpass and columns with highpass filters. . . . .	6
$\mathbf{w}_i^J$ , A random variable representing the magnitude of a coefficient of the CWT or HSI at the position $i$ and at the scale $J$ . . . . .	16
$\mathbf{W}^J$ , The collection of all random variables $\mathbf{w}_i^J$ at the scale $J$ . . . . .	15
$x_{max}$ , The maximum number of columns of an image. . . . .	15
$y_{max}$ , The maximum number of rows of an image. . . . .	15

Abstract of Thesis Presented to the Graduate School  
of the University of Florida in Partial Fulfillment of the  
Requirements for the Degree of Master of Science

STATISTICAL MODELING AND SEGMENTATION  
OF SKY/GROUND IMAGES

By

Sinisa Todorovic

December 2002

Chair: Michael C. Nechyba

Major Department: Electrical and Computer Engineering

In this thesis, a novel computer vision algorithm for statistical image modeling is proposed. The necessity for such an algorithm initially came as a requirement for vision-based stability and control of micro air vehicles (MAV). The MAV flight autonomy can be achieved by real-time horizon tracking, using the on-board vision system. Occasionally, this algorithm fails in scenarios where the underlying Gaussian assumption for the sky and ground appearances is not appropriate. Therefore, we propose a statistical modeling framework to build prior models of the sky and ground. Once trained, these models can be incorporated into the existing horizon-tracking algorithm for MAVs. Since the appearances of the sky and ground vary enormously, no single feature is sufficient for accurate modeling. As such, we rely both on color and texture as critical features in our modeling framework. Specifically, we choose the hue, saturation and intensity (HSI) color system for color representation, and the complex wavelet transform (CWT) for

texture representation. We then use the Hidden Markov Tree (HMT) model, as the underlying statistical model over our feature space. With this approach, we have achieved reliable and robust image segmentation of flight images from on-board our MAVs as well as on more difficult-to-classify sky/ground images. Furthermore, we demonstrate the generality of our modeling framework through another segmentation task—classifying the cut/uncut grass regions on images taken by an on-board camera of a lawn mower.

## CHAPTER 1 INTRODUCTION

Recently, the researchers in the Department of Mechanical and Aerospace Engineering and in the Department of Electrical and Computer Engineering, at the University of Florida, Gainesville, have successfully implemented a vision-based horizon tracking algorithm for flight stability and autonomy of Micro Air Vehicles (MAVs) [1]. The horizon tracking algorithm works well, especially when the sky and ground distributions are relatively coherent. Occasionally, however, horizon detection fails in scenarios where the underlying Gaussian assumption for the sky and ground appearances is not appropriate. Moreover, the horizon detection algorithm is bootstrapped by assuming that initially the sky occupies the upper part of the image. For complex mission scenarios, this may be an incorrect assumption with potentially fatal consequences to the flight vehicle. For example, in the case of deploying MAVs on munitions for post-impact bomb damage assessment, a MAV would separate from the munition prior to impact, and an upright attitude with respect to the ground cannot be guaranteed. Correct identification of sky and ground, therefore, becomes imperative.

While modeling the appearance of sky and ground regions in images may seem intuitively easy, it is, in fact, a very challenging task. Depending on lighting, weather, landscape, etc., the appearance of the sky and ground can vary enormously. Moreover, in some cases, even human perception fails, tricked by “unusual” patterns of sky and ground. For instance, the images of water surface reflecting the sky or mountain tops covered with snow represent the extreme cases, in which human detection of sky and ground is unreliable.

Consequently, given the complex variations in our two image classes (i.e., sky and ground), careful consideration must be given to selecting sufficiently discriminating features and a sufficiently expressive modeling framework. Next we present the overview of our approach.

### 1.1 Overview of the Proposed Computer Vision System

In this thesis, we propose a computer vision algorithm for image segmentation. The proposed vision system represents, in fact, a Bayes classifier, as illustrated in Figure 1.1. A Bayes classifier assigns a pixel to a particular class if a posteriori probability of the class given the pixel value is maximum. Designing a Bayes classifier guarantees the best performance, subject to the correctness of available statistical models [2]. Therefore, it is necessary to statistically model interdependencies among pixels as accurately as possible. This is achieved in the computer vision system, whose components are explained further in the text.

The first module is a preprocessing block, which subsamples the input image and thus reduces the overall computation cost.

In the following block, the key features of an image are extracted. Having experimented with color and texture features separately, we conclude that only the feature set that includes both color and texture clues enables accurate statistical modeling for our application. Previous experiments also suggest that it is important to represent both local as well as regional interdependencies in the feature space. As such, we first employ the hue, intensity and saturation (HSI) color space to represent color. Also, we resort to wavelet-based multi-resolution analysis in

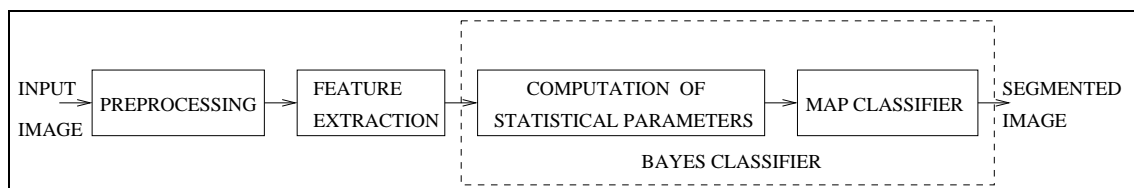


Figure 1.1 A computer vision system

the form of the Complex Wavelet Transform (CWT). Thus, from original complex representation of image classes, we obtain a finite set of features.

Given our feature selection, we then choose the Hidden Markov Tree (HMT) model as our underlying statistical model. Since the true distribution of the chosen feature set is not known, it is necessary to statistically model the interdependencies among the features. Therefore in the following block, we estimate statistical parameters which completely characterize the underlying distribution of the features.

Once, the model parameters are computed from the available training data, it is possible to find the likelihood of image classes for each pixel. Employing the maximum a posteriori (MAP) criterion, we assign a class label to each pixel, and, thus, perform segmentation.

In conclusion, the mutual dependence of all modules in the system should also be emphasized. The system performs poorly overall if only one of the components is badly designed. For instance, a perfect set of features and a sophisticated statistical model could yield horrible results if only the segmentation block is bad. Therefore, one should always have in mind that modularity, as depicted in Figure 1.1, serves only for presentation purposes.

## 1.2 Overview of the Thesis

In the following chapters we discuss the main components of our computer vision system.

First, in Chapter 2, we explain our choice of the feature space for statistical modeling and review fundamental theory, necessary for better understanding of our work. Specifically, we give an overview of the most important aspects of the HSI color system and the CWT. Next, in Chapter 3 we introduce the properties of the Hidden Markov Model, and in Chapter 4 we develop the algorithm for training HMT models, pointing out implementation issues. Further, in Chapter 5, we

explain the procedure for multiscale image segmentation. In Chapter 6, we present experimental results. We illustrate the reliability and robustness of our approach on examples of flight images from on-board our MAVs as well as on more difficult-to-classify sky/ground images. Furthermore, we demonstrate the generality of our modeling framework through another segmentation task—classifying the cut/uncut grass regions on images taken by an on-board camera of a lawn mower. Finally, we discuss the experimental results in Chapter 7.

## CHAPTER 2 FEATURE SPACE

For our statistical models, we seek to identify features that lead to improved segmentation performance without unnecessarily increasing computational complexity. Color or texture clues by themselves yield poor segmentation results. Therefore, we consider a feature space that spans both color and texture domains.

### 2.1 Color

The color information in a video signal is usually encoded in the RGB color space. Unfortunately, the  $R$ ,  $G$  and  $B$  color channels are highly correlated; therefore, we choose the HSI space as a more appropriate color representation [3]. Though, HSI exhibits numerical instability at low saturation, this problem can be easily overcome by assigning some application-dependent, predefined values to the hue ( $H$ ), intensity ( $I$ ) and saturation ( $S$ ) features. Thus, for sky/ground segmentation, we assume that low  $S$  and/or  $I$  values most likely appear for the ground class, which then sets  $H$  value appropriately.

### 2.2 Texture

For the choice of texture-based features, we have considered several filtering, model-based and statistical methods for texture feature extraction. Our conclusion complies with the comparative study of Randen and Husoy [4] that for problems with many textures with subtle spectral differences, as in the case of our complex classes, it is reasonable to assume that the spectral decomposition by a filter bank yields consistently superior results over other texture analysis methods. Our experimental results also suggest that it is crucial to analyze both local as well as



regional properties of texture. As such, we employ the wavelet transform, due to its inherent representation of texture at different scales and locations.

### 2.2.1 Wavelet Transform

Wavelet atom functions, being well localized both in space and frequency, retrieve texture information quite successfully [5]. The conventional discrete wavelet transform (DWT) may be regarded as equivalent to filtering the input signal with a bank of bandpass filters, whose impulse responses are all given by scaled versions of a mother wavelet. The scaling factor between adjacent filters is 2:1, leading to octave bandwidths and center frequencies that are one octave apart. The octave-band DWT is most efficiently implemented by the dyadic wavelet decomposition tree of Mallat [6], where wavelet coefficients of an image are obtained convolving every row and column with impulse responses of lowpass and highpass filters, as shown in Figure 2.1. Practically, coefficients of one scale are obtained convolving every second row and column from the previous finer scale. Thus, the filter output is a wavelet subimage that has four times less coefficients than the one at the previous scale. The lowpass filter is denoted with  $H_0$  and the highpass filter with  $H_1$ . The wavelet coefficients  $W$  have in index  $L$  denoting lowpass output and  $H$  for highpass output.

Separable filtering of rows and columns produces four subimages at each level, which can be arranged as shown in Figure 2.2. The same figure also illustrates well the directional selectivity of the DWT, because  $W_{LH}$ ,  $W_{HL}$ , and  $W_{HH}$ , bandpass subimages can select horizontal, vertical and diagonal edges, respectively.

### 2.2.2 Wavelet Properties

The following properties of the DWT have made wavelet-based image processing very attractive in recent years [5, 7, 8]:

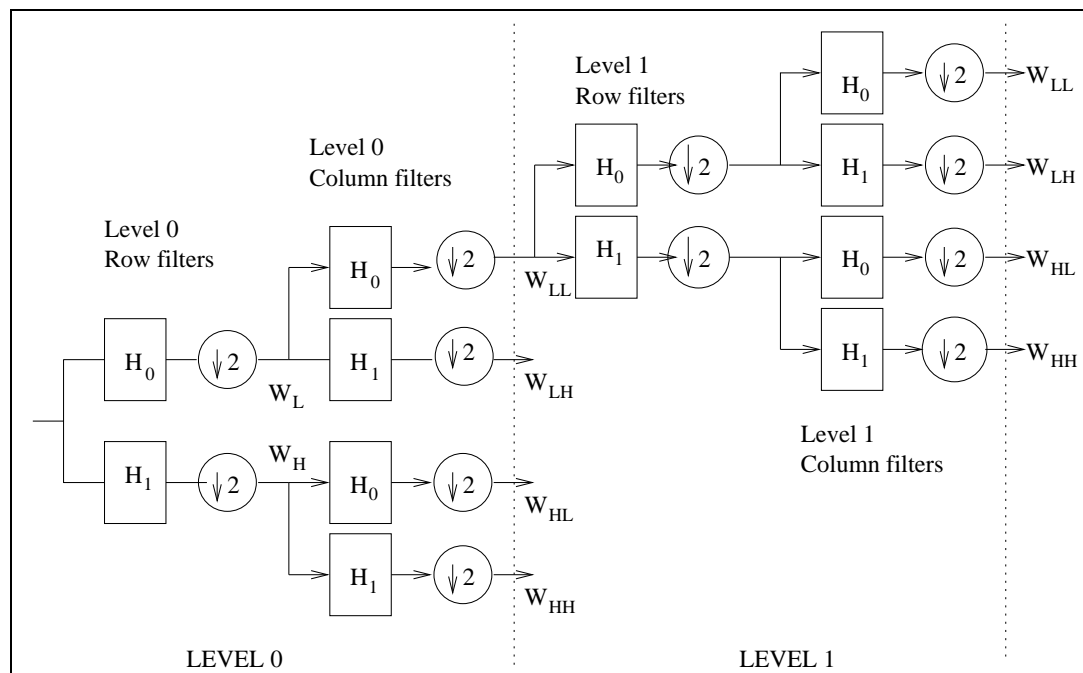


Figure 2.1 Two levels of the DWT of a two-dimensional signal.

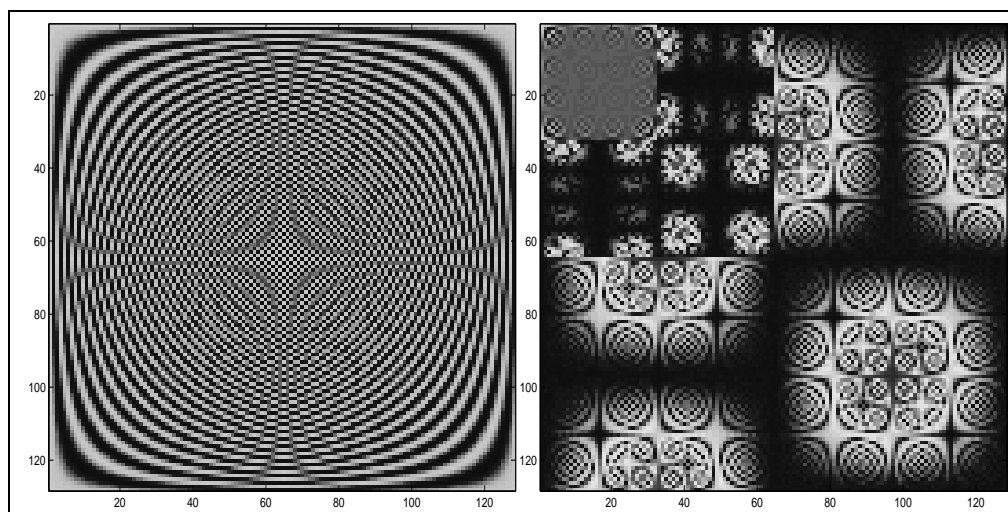


Figure 2.2 The original image (left) and its two-scale dyadic DWT (right).

- a. locality: each wavelet coefficient represents local image content in space and frequency, because wavelets are well localized simultaneously in space and frequency
- b. multi-resolution: DWT represents an image at different scales of resolution in space domain (i.e., in frequency domain); regions of analysis at one scale are divided up into four smaller regions at the next finer scale (Fig. 2.2)
- c. edge detector: edges of an image are represented by large wavelet coefficients at the corresponding locations
- d. energy compression: wavelet coefficients are large only if edges are present within the support of the wavelet, which means that the majority of wavelet coefficients have small values
- e. decorrelation: wavelet coefficients are approximately decorrelated, since the scaled and shifted wavelets form orthonormal basis; dependencies among wavelet coefficients are predominantly local
- f. clustering: if a particular wavelet coefficient is large/small, then the adjacent coefficients are very likely to also be large/small
- g. persistence: large/small values of wavelet coefficients tend to propagate through scales
- h. non-Gaussian marginal pdf: wavelet coefficients have peaky and long-tailed marginal distributions; due to the energy compression property only a few wavelet coefficients have large values, therefore a Gaussian pdf for an individual coefficient is a poor statistical model

It is also important to introduce shortcomings of the DWT. Discrete wavelet decompositions suffer from two main problems, which hamper their use for many applications, as follows [9]:

- a. lack of shift invariance: small shifts in the input signal can cause major variations in the energy distribution of wavelet coefficients
- b. poor directional selectivity: for some applications horizontal, vertical and diagonal selectivity is insufficient

When we analyze the Fourier spectrum of a signal, we expect the energy in each frequency bin to be invariant to any shifts of the input. Unfortunately, the DWT has a significant drawback that the energy distribution between various wavelet scales depends critically on the position of key features of the input signal,

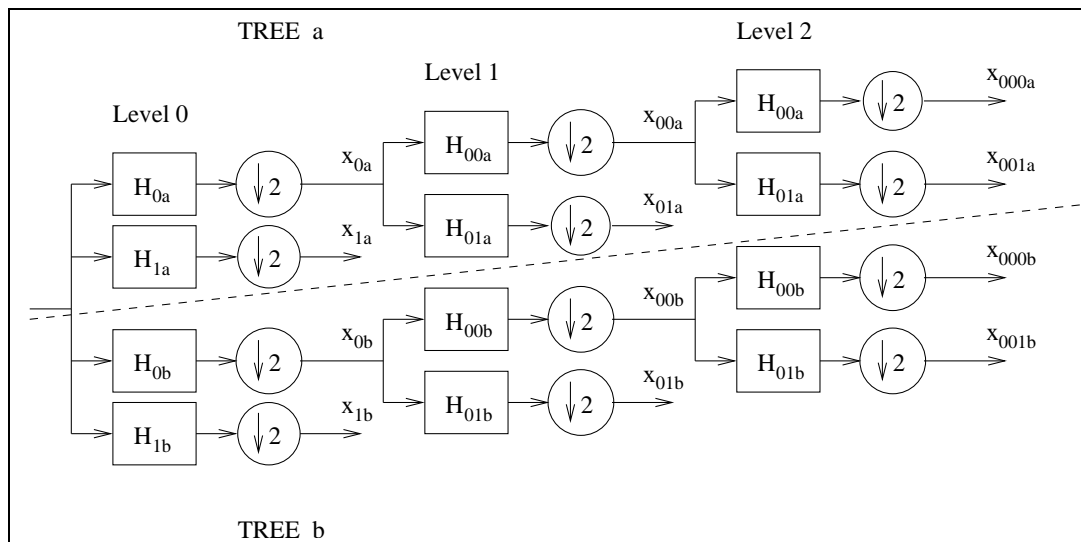


Figure 2.3 The Q-shift Dual-Tree CWT.

whereas ideally dependence is on just the features themselves. Therefore, the real DWT is unlikely to give consistent results when used in texture analysis.

In literature, there are several approaches proposed to overcome this problem (e.g., Discrete Wavelet Frames [5, 10]), all increasing computational load with inevitable redundancy in the wavelet domain. In our opinion, the Complex Wavelet Transform (CWT) offers the best solution providing additional advantages, described in the following subsection.

### 2.2.3 Complex Wavelet Transform

The structure of the CWT is the same as in Figure 2.1, except that the CWT filters have complex coefficients and generate complex output. The output sampling rates are unchanged from the DWT, but each wavelet coefficient contains a real and imaginary part, thus a redundancy of 2:1 for one-dimensional signals is introduced. In our case, for two-dimensional signals, the redundancy becomes 4:1, because two adjacent quadrants of the spectrum are required to represent fully a real two-dimensional signal, adding an extra 2:1 factor. This is achieved by additional filtering with complex conjugates of either the row or column filters [9, 11].

Table 2.1 Coefficients of the filters used in the Q-shift DTCWT.

$H_{13}$ (symmetric)	$H_{19}$ (symmetric)	$H_6$
-0.0017581	-0.0000706	0.03616384
0	0	0
0.0222656	0.0013419	-0.08832942
-0.0468750	-0.0018834	0.23389032
-0.0482422	-0.0071568	0.76027237
0.2968750	0.0238560	0.58751830
0.5554688	0.0556431	0
0.2968750	-0.0516881	-0.11430184
-0.0482422	-0.2997576	0
$\vdots$	0.5594308	0
	-0.2997576	
	$\vdots$	

Despite its higher computational cost, we prefer the CWT over the DWT because of the CWT's following attractive properties. The CWT is shown to possess almost shift and rotational invariance, given suitably designed biorthogonal or orthogonal wavelet filters. We implement the Q-shift Dual-Tree CWT scheme, proposed by Kingsbury [12], as depicted in Figure 2.3. The figure shows the CWT of only one-dimensional signal  $x$ , for clarity. The output of the trees  $a$  and  $b$  can be viewed as real and imaginary parts of complex wavelet coefficients, respectively. Thus, to compute the CWT, we implement two real DWT's (see Fig. 2.1), obtaining a wavelet frame with redundancy two. As for the DWT, here, lowpass and highpass filters are denoted with 0 and 1 in index, respectively. The level 0 comprises odd-length filters  $H_{0a}(z) = H_{0b}(z) = H_{13}(z)$  (13 taps) and  $H_{1a}(z) = H_{1b}(z) = H_{19}(z)$  (19 taps). Levels above the level 0 consist of even-length filters  $H_{00a}(z) = z^{-1}H_6(z^{-1})$ ,  $H_{01a}(z) = H_6(-z)$ ,  $H_{00b}(z) = H_6(z)$ ,  $H_{01b}(z) = z^{-1}H_6(-z^{-1})$ , where the impulse response of the filters  $H_{13}$ ,  $H_{19}$  and  $H_6$  is given in the table 2.1.

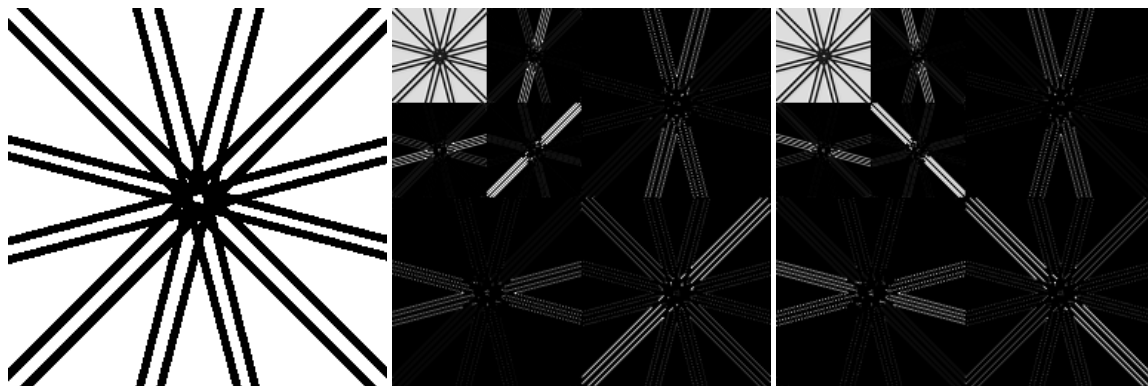


Figure 2.4 The CWT is strongly oriented at angles  $\pm 15^\circ$ ,  $\pm 45^\circ$ ,  $\pm 75^\circ$ .

Aside from being shift invariant, the CWT is superior to the DWT in terms of directional selectivity, too. A two-dimensional CWT produces six bandpass subimages (analogous to the three subimages in the DWT) of complex coefficients at each level, which are strongly oriented at angles of  $\pm 15^\circ$ ,  $\pm 45^\circ$ ,  $\pm 75^\circ$ , as illustrated in Figure 2.4.

Another advantageous property of the CWT exerts in the presence of noise. The phase and magnitude of the complex wavelet coefficients collaborate in a non trivial way to describe data [11]. The phase encodes the coherent (in space and scale) structure of an image, which is resilient to noise, and the magnitude captures the strength of local information that could be very susceptible to noise corruption. Hence, the phase of complex wavelet coefficients might be used as a principal clue for image denoising. However, our experimental results have shown that phase is not a good feature choice for sky/ground modeling. Therefore, we consider only magnitudes.

To conclude, we consider the CWT a suitable tool to analyze image texture. The CWT inherited attractive properties of the DWT, such as: locality, multi-resolution, energy compression, etc. Its shift invariance and increased directional selectivity give additional power to statistical modeling of image classes. Therefore,

we choose, beside the color features HSI, the CWT to complement our feature space.

#### **2.2.4 Color Multi-resolution Representation**

In order to unify the color and texture features into a unique modeling framework, we also form multi-resolution representation for the  $H$ ,  $S$  and  $I$  features, analogous to the CWT structure. The  $H$ ,  $S$  and  $I$  values at coarser scales are computed as the mean of the corresponding four values at the next higher-resolution scale. Hence, the  $H$ ,  $S$  and  $I$  features also exhibit the clustering and persistence properties to some extent.

Naturally, the choice of the feature space determines the statistical image model. In the next chapter, we describe the HMT model as an appropriate statistical framework for modeling our complex classes in the chosen feature space.

## CHAPTER 3 STATISTICAL MODELING

In order to compute the joint pdf of the chosen features, which fully characterizes a class for our Bayes classifier (see Fig 1.1), we must account for the key dependencies. In our case, we need to examine thoroughly all the properties of the CWT and HSI features, and to best incorporate this knowledge into our statistical model.

### 3.1 Marginal Density

We model the true marginal pdf of a feature as a mixture of Gaussian pdf's. Let's refer to the magnitude of a CWT coefficient and/or the  $H$ ,  $S$  and  $I$  color values, simply as coefficient. Then, a coefficient can be in one of the  $M$  states, which results in an M-state Gaussian mixture model. It means that we associate to each coefficient  $\mathbf{w}_i$ , at location  $(x, y)$ , a state random variable  $\mathbf{S}_i$ <sup>1</sup>. In general,  $\mathbf{S}_i$  can take on values  $m \in \{0 \dots M - 1\}$ . The mixture model is completely parameterized with the probability measure function (pmf) of the state random variable  $P(\mathbf{S}_i = m)$  and with the means  $\mu_i(m)$  and the variances  $\sigma_i^2(m)$  of each state. By increasing the number of states, we can make our model fit arbitrarily close to real densities with a finite number of discontinuities [7].

Thus, the marginal pdf of a wavelet coefficient  $\mathbf{w}_i$  is given by

$$p(\mathbf{w}_i) = \sum_{m=0}^{M-1} P(\mathbf{S}_i = m)p(\mathbf{w}_i|\mathbf{S}_i = m), \quad (3.1)$$

---

<sup>1</sup> Note that boldface letters represent random variables and their particular realizations are not boldface.



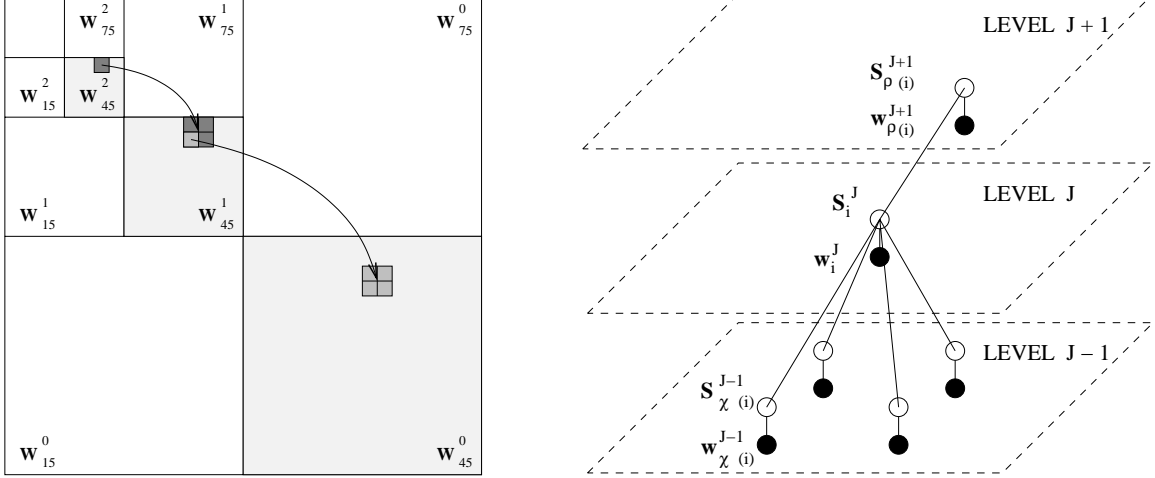


Figure 3.1 The HMT of the three-level CWT.

where

$$p(\mathbf{w}_i | \mathbf{S}_i = m) = \frac{1}{\sqrt{2\pi} \sigma_i(m)} e^{-\frac{(w_i - \mu_i(m))^2}{2\sigma_i^2(m)}}. \quad (3.2)$$

Although each coefficient  $\mathbf{w}_i$  is conditionally Gaussian, given its state random variable  $\mathbf{S}_i$ , the overall density is non-Gaussian due to the randomness of  $\mathbf{S}_i$ .

### 3.2 Graph Models

A probabilistic graph associates each random variable with a node in the graph [13]. Correlation between pairs of variables is represented by links connecting the corresponding nodes. The chosen feature space can be modeled by a probabilistic graph in the following way. A node is assigned to each coefficient and an edge is placed between two mutually dependent nodes. All coefficients that belong to the same scale are placed in the corresponding horizontal plane. The coefficients of the finer scales form planes under the current one, and the coefficients that belong to the coarser scales form planes above the current one, as illustrated in Figure 3.1. Four adjacent coefficients at one scale have a unique parent belonging to the upper coarser scale, only within the same feature (shaded area). States  $\mathbf{S}_i^J$  are depicted as white nodes and coefficient values  $\mathbf{w}_i^J$  as black nodes.

Thus, we obtain a pyramid structure, because the coarser the scale the fewer nodes there are. We can develop graphs that can capture arbitrary interdependencies, but the computational complexity increases substantially for graphs more complicated than trees. If we keep only horizontal edges, which connect adjacent nodes of one scale, we obtain the Hidden Markov Chain Model (HMC). By connecting only the adjacent nodes vertically across scale, we obtain the Hidden Markov Tree Model (HMT).

### 3.3 HMT Model

The HMT models both the clustering and persistence properties. A tree structure is formed connecting every parent coefficient<sup>2</sup> from the coarser scale with its children at the finer scale. There are four direct children from one parent node, and every child has only one parent node, as depicted in Figure 3.1. Leaf nodes have no children and they are formed from the coefficients at the finest resolution. Although we do not include horizontal edges among the nodes of one scale in our model, the clustering property is still well modeled, because the adjacent nodes at one scale have the same parent. Therefore the local correlation is indirectly modeled by interscale dependencies [14].

In order to discuss HMT properties, first we need to introduce the following notation. The random field of all wavelet coefficients at a scale  $J$  is denoted with  $\mathbf{W}^J = \{\mathbf{w}_i^J\}$ . A node  $i$  has a parent node  $\rho(i)$  and a set of children nodes  $\chi(i)$ . This implies that  $J(\chi(i)) = J(i) - 1$  and  $J(\rho(i)) = J(i) + 1$ . The scales are ordered from the finest  $J = 0$  to the coarsest  $J = L - 1$ , where  $L = \log_2 x_{max}$ . In this project, the dimensions  $x_{max}$  and  $y_{max}$  of an image are assumed to be equal to facilitate computation of the CWT.

---

<sup>2</sup> Here, coefficient refers to the magnitude of a CWT coefficient and/or the  $H$ ,  $S$  and  $I$  color values.

### 3.4 HMT Properties

The HMT imposes that  $\mathbf{w}_i^J$  is conditionally independent of all other random variables given its associated state  $\mathbf{S}_i^J$ , [15]. Secondly, given a parent state  $\mathbf{S}_{\rho(i)}^{J+1}$ ,  $\mathbf{w}_i^J$  is independent of the  $\rho(i)$ 's ancestors. Conversely, given a child state  $\mathbf{S}_{\chi(i)}^{J-1}$ ,  $\mathbf{w}_i^J$  is independent of the child's descendants. Combining these properties shows that  $\mathbf{w}_i^J$  is conditionally independent of the entire tree, given only the parent state  $\mathbf{S}_{\rho(i)}^{J+1}$  and the children states  $\mathbf{S}_{\chi(i)}^{J-1}$ . Here follows a mathematical formulation of these statements:

$$p(\mathbf{w}_i^J \mid \{\mathbf{w}_k^J\}_{k \neq i}, \{\mathbf{S}_k^J\}_{k \neq i}, \mathbf{S}_i^J = m) = p(\mathbf{w}_i^J \mid \mathbf{S}_i^J = m), \quad (3.3)$$

$$p(\mathbf{w}_i^J \mid \{\mathbf{w}_k^J\}^{J(k) > J(\rho(i))}, \mathbf{w}_{\rho(i)}^{J+1}) = p(\mathbf{w}_i^J \mid \mathbf{w}_{\rho(i)}^{J+1}), \quad (3.4)$$

$$p(\mathbf{w}_i^J \mid \{\mathbf{w}_k^J\}^{J(k) < J(\chi(i))}, \mathbf{w}_{\chi(i)}^{J-1}) = p(\mathbf{w}_i^J \mid \mathbf{w}_{\chi(i)}^{J-1}). \quad (3.5)$$

It is important to note that the Markov structure is related to the states of coefficients and not to the coefficient values. Further, since the states are never known exactly, the Markov model is hidden with real continuous outcomes. Also, unlike the usual HMM treatment in literature [15], here transitions between states are not related to time. In fact, the HMT deals with transitions between scales (i.e., horizontal planes) of the pyramidal graph structure.

Using an M-state Gaussian mixture model for each wavelet coefficient  $\mathbf{w}_i$ , the parameters for the HMT model are

- i.  $P(\mathbf{S}_0^{L-1} = m)$ ,  $m \in [0, M - 1]$ , the pmf of the root node  $\mathbf{S}_0^{L-1}$
- ii.  $a_{i,\rho(i)}^J(m, n) = P(\mathbf{S}_i^J = m \mid \mathbf{S}_{\rho(i)}^{J+1} = n)$ , the transition probability that  $\mathbf{S}_i^J$  is in a state  $m$  given that  $\mathbf{S}_{\rho(i)}^{J+1}$  is in a state  $n$ , where  $m \in [0, M - 1]$  and  $n \in [0, M - 1]$
- iii.  $\mu_i^J(m)$ , and  $(\sigma_i^J(m))^2$ , the mean and variance, respectively, of the wavelet coefficient  $\mathbf{w}_i^J$ , given  $\mathbf{S}_i^J$  is in a state  $m$ , where  $m \in [0, M - 1]$

These parameters can be grouped into a model parameter random vector  $\Theta$ .

For training the HMT, we employ the expectation maximization (EM) algorithm [16, 17], which iteratively estimates model parameters  $\Theta$ . Under mild conditions, the iteration converges to a local maximum of the function  $p(\mathbf{W}|\Theta)$ . The training of the HMT is explained in the following chapter.

CHAPTER 4  
TRAINING OF THE HMT MODEL

Our aim is to determine the parameter vector  $\Theta$ , given the pyramidal structure of observed coefficients  $\mathbf{W} = W$ . Here, the usual EM algorithm for training HMM's [15], is modified for the continuous outcome HMT.

First, let's introduce the following notation. We define  $\mathcal{T}_i^J$  to be the subtree of observed coefficients with a root at node  $\mathbf{w}_i^J$ , at the level  $J$ .  $\mathcal{T}_i^J$  contains the coefficient  $\mathbf{w}_i^J$  and all its descendants. If  $\mathcal{T}_k^{\{J(k)<J\}}$  is a subtree of  $\mathcal{T}_i^J$  (i.e.,  $\mathbf{w}_k^{\{J(k)<J\}}$  and all its descendants are members of  $\mathcal{T}_i^J$ ), then we define  $\mathcal{T}_{i \setminus k}^J$  to be the set of coefficients obtained by removing the subtree  $\mathcal{T}_k^{\{J(k)<J\}}$  from  $\mathcal{T}_i^J$ . Without loss of generality, we order  $\mathbf{W}$  so that  $\mathbf{w}_0^{L-1}$  is at the root of the entire tree. Thus,  $\mathcal{T}_0^{L-1}$  is the entire tree of observed coefficients and we interchange the symbols  $\mathcal{T}_0^{L-1}$  and  $\mathbf{W}$  when convenient.

### 4.1 Implementation of the EM Algorithm

Customary for the EM algorithm [15], for each subtree  $\mathcal{T}_i^J$ , we define the following conditional densities:

$$\beta_i^J(m) = p(\mathcal{T}_i^J \mid \mathbf{S}_i^J = m, \Theta), \quad (4.1)$$

$$\beta_{\rho(i),i}^J(m) = p(\mathcal{T}_i^J \mid \mathbf{S}_{\rho(i)}^{J+1} = m, \Theta), \quad (4.2)$$

$$\beta_{\rho(i) \setminus i}^{J+1}(m) = p(\mathcal{T}_{\rho(i) \setminus i}^{J+1} \mid \mathbf{S}_{\rho(i)}^{J+1} = m, \Theta), \quad (4.3)$$

and the joint probability

$$\alpha_i^J(m) = P(\mathbf{S}_i^J = m, \mathcal{T}_{0 \setminus i}^{L-1} \mid \Theta), \quad (4.4)$$

with  $\mathbf{S}_i^J$  taking discrete values and  $\mathcal{T}_{0 \setminus i}^{L-1}$  taking continuous values.

To determine the HMT parameter vector  $\Theta$ , it is necessary first to compute the state and transition probabilities for all the observed coefficients  $\mathbf{W}$ ,  $P(\mathbf{S}_i^J = m \mid \mathbf{W}, \Theta)$  and  $P(\mathbf{S}_i^J = m \mid \mathbf{S}_{\rho(i)}^{J+1} = n, \mathbf{W}, \Theta)$ . From the HMT properties (3.3), (3.4) and (3.5) it follows that the trees  $\mathcal{T}_i^J$  and  $\mathcal{T}_{0 \setminus i}^{L-1}$  are independent, given the state random variable  $\mathbf{S}_i^J$ . This fact, together with the Markov chain rule, leads to the desired expressions in terms of the defined  $\alpha$  and  $\beta$ . First, we obtain

$$P(\mathbf{S}_i^J = m, \mathcal{T}_0^{L-1} \mid \Theta) = \alpha_i^J(m) \beta_i^J(m), \quad (4.5)$$

$$P(\mathbf{S}_i^J = m, \mathbf{S}_{\rho(i)}^{J+1} = n, \mathcal{T}_0^{L-1} \mid \Theta) = \beta_i^J(m) a_{i, \rho(i)}^J(m, n) \alpha_{\rho(i)}^{J+1}(m) \beta_{\rho(i) \setminus i}^{J+1}(n). \quad (4.6)$$

Then, the joint pdf of  $\mathbf{W}$  can be expressed as

$$\begin{aligned} p(\mathbf{W} \mid \Theta) &= p(\mathcal{T}_0^{L-1} \mid \Theta) = \sum_{m=0}^{M-1} P(\mathbf{S}_i^J = m, \mathcal{T}_0^{L-1} \mid \Theta), \\ &= \sum_{m=0}^{M-1} \alpha_i^J(m) \beta_i^J(m). \end{aligned} \quad (4.7)$$

Applying the Bayes rule to (4.5)-(4.7) results in the desired conditional probabilities

$$P(\mathbf{S}_i^J = m \mid \mathbf{W}, \Theta) = \frac{\alpha_i^J(m) \beta_i^J(m)}{\sum_{n=0}^{M-1} \alpha_i^J(n) \beta_i^J(n)}, \quad (4.8)$$

$$P(\mathbf{S}_i^J = m, \mathbf{S}_{\rho(i)}^{J+1} = n \mid \mathbf{W}, \Theta) = \frac{\beta_i^J(m) a_{i, \rho(i)}^J(m, n) \alpha_{\rho(i)}^{J+1}(n) \beta_{\rho(i) \setminus i}^{J+1}(n)}{\sum_{n=0}^{M-1} \alpha_i^J(n) \beta_i^J(n)}. \quad (4.9)$$

From (4.8), (4.9) and (4.3) follows the expression for the transition probability

$$\begin{aligned} P(\mathbf{S}_i^J = m \mid \mathbf{S}_{\rho(i)}^{J+1} = n, \mathbf{W}, \Theta) &= \frac{P(\mathbf{S}_i^J = m, \mathbf{S}_{\rho(i)}^{J+1} = n \mid \mathbf{W}, \Theta)}{P(\mathbf{S}_{\rho(i)}^{J+1} = n \mid \mathbf{W}, \Theta)}, \\ &= \frac{\beta_i^J(m) a_{i, \rho(i)}^J(m, n) \alpha_{\rho(i)}^{J+1}(n) \beta_{\rho(i) \setminus i}^{J+1}(n)}{\alpha_{\rho(i)}^{J+1}(n) \beta_{\rho(i)}^{J+1}(n)}, \\ &= \frac{\beta_i^J(m) a_{i, \rho(i)}^J(m, n)}{\beta_{\rho(i), i}^J(n)}. \end{aligned} \quad (4.10)$$

Apparently, the expressions (4.8) and (4.10) guarantee that the computed probabilities are less or equal to 1. From the state and transition probabilities all other components of the random vector  $\Theta$  can be computed.

#### 4.1.1 E Step

In determining probabilities for the state variables, the state information is propagated throughout the tree by means of upward-downward<sup>1</sup> algorithm. The up step calculates  $\beta$ 's by transmitting information about the fine-scale wavelet/color coefficients up to coarser scales. The down step computes  $\alpha$ 's by propagating information about the coarse-scale wavelet/color coefficients down to finer scales.

In the following subsections we present the up and down steps for the  $l$ -th iteration of the EM algorithm.

#### 4.1.2 Up Step

The up part of the EM algorithm consists of the following steps:

1.  $\forall \mathbf{S}_i^0 = m$ , where  $m \in \{0, \dots, M-1\}$ , calculate

$$\beta_i^0(m) = N(w_i^0, \mu_i^0(m), \sigma_i^0(m)), \quad (4.11)$$

2.  $\forall \mathbf{S}_i^J = m$ , where  $m \in \{0, \dots, M-1\}$ , calculate

$$\beta_{\rho(i),i}^J(n) = \sum_{m=0}^{M-1} a_{i,\rho(i)}^J(m, n) \beta_i^J(m) \quad (4.12)$$

$$\beta_{\rho(i)}^{J+1}(m) = N(w_{\rho(i)}^{J+1}, \mu_{\rho(i)}^{J+1}(m), \sigma_{\rho(i)}^{J+1}(m)) \prod_{k \in \chi(\rho(i))} \beta_{\rho(k),k}^J(m), \quad (4.13)$$

$$\beta_{\rho(i)\setminus i}^{J+1}(m) = \frac{\beta_{\rho(i)}^{J+1}(m)}{\beta_{\rho(i),i}^J(m)}. \quad (4.14)$$

---

<sup>1</sup> E step of the EM algorithm is often referred to as the forward-backward algorithm in the HMM literature and as the upward-downward algorithm in the AI literature. Since the tree structure imposes moves up and down in the E step computation, here we use the latter term.

3. Move up the tree, setting  $J = J + 1$ .
4. If  $J = L$ , then stop; else return to step 2.

### 4.1.3 Down Step

The down part of the EM algorithm consists of the following steps:

1.  $\forall \mathbf{S}_0^{L-1} = m$ , where  $m \in \{0, \dots, M-1\}$ , set

$$\alpha_0^{L-1}(m) = P(\mathbf{S}_0^{L-1} = m \mid \mathbf{W}, \Theta = \Theta_l) . \quad (4.15)$$

2. Move down the tree, setting  $J = J - 1$ .
3.  $\forall \mathbf{S}_i^J = m$ , where  $m \in \{0, \dots, M-1\}$ , calculate

$$\alpha_i^J(m) = \sum_{n=0}^{M-1} a_{i,\rho(i)}^J(m, n) \alpha_{\rho(i)}^{J+1}(n) \beta_{\rho(i)\setminus i}^{J+1}(n) . \quad (4.16)$$

4. If  $J = 0$ , then stop; else return to step 2.

### 4.1.4 M Step

Once  $\alpha$  and  $\beta$  values are computed for all nodes, the maximization step of the EM algorithm follows straightforward. The state and transition probabilities could be computed applying (4.8) and (4.10). We could then find  $\mu_i^J(m)$  and  $\sigma_i^J(m)$  for all coefficients. However, all that will make our HMT model too complex.

In order to simplify and to avoid the risk of overfitting the HMT model, we assume that statistic parameters are equal for all coefficients belonging to the same scale. It means that all nodes at the scale  $J$  are characterized by the following unique parameters:  $P(\mathbf{S}^J = m \mid \mathbf{W}, \Theta)$ ,  $\mu^J(m)$ ,  $\sigma^J(m)$ , and  $P(\mathbf{S}^J = m \mid \mathbf{S}^{J+1} = n, \mathbf{W}, \Theta)$ . Here,  $\mathbf{S}^J$  denotes the state random variable for all nodes  $i$  at the same scale  $J$ .



Applying the assumption, in the M step, results in the following expressions for all K nodes at the scale  $J$ , for the  $l$ -th iteration:

$$\begin{aligned} P(\mathbf{S}^J = m \mid \mathbf{W}, \Theta) &= \frac{1}{K} \sum_{i=0}^{K-1} P(\mathbf{S}_i^J = m \mid \mathbf{W}, \Theta = \Theta_l), \\ &= \frac{1}{K} \sum_{i=0}^{K-1} \frac{\alpha_i^J(m) \beta_i^J(m)}{\sum_{n=0}^{M-1} \alpha_i^J(n) \beta_i^J(n)}, \end{aligned} \quad (4.17)$$

$$\begin{aligned} P(\mathbf{S}^J = m \mid \mathbf{S}^{J+1} = n, \mathbf{W}, \Theta) &= \frac{1}{K} \sum_{i=0}^{K-1} P(\mathbf{S}_i^J = m \mid \mathbf{S}_{\rho(i)}^{J+1} = n, \mathbf{W}, \Theta = \Theta_l), \\ &= \frac{1}{K} \sum_{i=0}^{K-1} \frac{\beta_i^J(m) a_{i,\rho(i)}^J(m, n)}{\beta_{\rho(i),i}^J(n)}, \end{aligned} \quad (4.18)$$

$$\mu^J(m) = \frac{\sum_{i=0}^{K-1} w_i^J P(\mathbf{S}_i^J = m \mid \mathbf{W}, \Theta)}{K P(\mathbf{S}^J = m \mid \mathbf{W}, \Theta)}, \quad (4.19)$$

$$(\sigma^J(m))^2 = \frac{\sum_{i=0}^{K-1} (w_i^J - \mu^J(m))^2 P(\mathbf{S}_i^J = m \mid \mathbf{W}, \Theta)}{K P(\mathbf{S}^J = m \mid \mathbf{W}, \Theta)}. \quad (4.20)$$

## 4.2 Implementation Issues

The computer implementation of the upward-downward algorithm, if the equations (4.11) - (4.16) are directly used, suffers from numerical underflow. In order to make the algorithm stable, we need to introduce adaptive scaling of  $\beta$  values. The dynamic scaling is performed  $\beta$  values only. Since  $\beta$ 's are computed only in the up step, we need only to rewrite the equations (4.11) - (4.14). The scaled values are denoted with  $\hat{\beta}_i$ . In the following subsections we present the scaled up step and down step algorithms.

### 4.2.1 Scaled Up Step

The scaled up part of the EM algorithm consists of the following steps:

1.  $\forall i$  at the scale  $J = 0$  compute

$$\begin{aligned}\beta_i^0(m) &= N(w_i^0, \mu_i^0(m), \sigma_i^0(m)) , \\ s_i^0 &= \sum_{m=0}^{M-1} \beta_i^0(m) \quad , (\text{scaling factor for } \beta' \text{ s}) , \\ \hat{\beta}_i^0(m) &= \frac{1}{s_i^0} \beta_i^0(m) .\end{aligned}\tag{4.21}$$

2.  $\forall i$  at a scale  $J$  compute

$$\begin{aligned}\hat{\beta}_{\rho(i),i}^J(n) &= \sum_{m=0}^{M-1} a_{i,\rho(i)}^J(m, n) \hat{\beta}_i^J(m) = \frac{1}{s_i^J} \sum_{m=0}^{M-1} a_{i,\rho(i)}^J(m, n) \beta_i^J(m) , \\ &= \frac{1}{s_i^J} \beta_{\rho(i),i}^J(n) ,\end{aligned}\tag{4.22}$$

$$\begin{aligned}\tilde{\beta}_{\rho(i)}^{J+1}(m) &= N(w_{\rho(i)}^{J+1}, \mu_{\rho(i)}^{J+1}(m), \sigma_{\rho(i)}^{J+1}(m)) \prod_{k \in \chi(\rho(i))} \hat{\beta}_{\rho(k),k}^J(m) , \\ &= \frac{1}{\prod_{k \in \chi(\rho(i))} s_k^J} \beta_{\rho(i)}^{J+1}(m) ,\end{aligned}\tag{4.23}$$

$$\tilde{s}_{\rho(i)}^{J+1} = \sum_{m=0}^{M-1} \tilde{\beta}_{\rho(i)}^{J+1}(m) = \frac{1}{\prod_{k \in \chi(\rho(i))} s_k^J} \sum_{m=0}^{M-1} \beta_{\rho(i)}^{J+1}(m) ,\tag{4.24}$$

$$\hat{\beta}_{\rho(i)}^{J+1}(m) = \frac{\tilde{\beta}_{\rho(i)}^{J+1}(m)}{\tilde{s}_{\rho(i)}^{J+1}} = \frac{\beta_{\rho(i)}^{J+1}}{\sum_{m=0}^{M-1} \beta_{\rho(i)}^{J+1}(m)} = \frac{\beta_{\rho(i)}^{J+1}(m)}{s_{\rho(i)}^{J+1}} ,\tag{4.25}$$

$$\hat{\beta}_{\rho(i)\setminus i}^{J+1}(m) = \frac{\hat{\beta}_{\rho(i)}^{J+1}(m)}{\hat{\beta}_{\rho(i),i}^J(m)} = \frac{\frac{\beta_{\rho(i)}^{J+1}(m)}{s_{\rho(i)}^{J+1}}}{\frac{\beta_{\rho(i),i}^J(m)}{s_i^J}} = \frac{s_i^J}{s_{\rho(i)}^{J+1}} \beta_{\rho(i)\setminus i}^J(m) .\tag{4.26}$$

3.  $J = J + 1$ .

4. If  $J = L$ , then stop; else return to step 2.

From (4.21) and (4.25) it is obvious that scaling at each step of the up algorithm limits the dynamics of  $\hat{\beta}$  values to the interval  $[0, 1]$ . If we carefully observe the equation (4.16) in the down-step algorithm,  $\alpha$  values need not be scaled, because limiting  $\hat{\beta}_{\rho(i)\setminus i}^{J+1}(m)$  stabilizes  $\hat{\alpha}$ 's in (4.16), as well.

### 4.2.2 Scaled Down Step

The scaled down part of the EM algorithm consists of the following steps:

1. For  $J = L - 1$  compute

$$\begin{aligned} t_{\rho(0)}^L &= 1, (\text{scaling factor for } \alpha_0^{L-1}), \\ \hat{\alpha}_0^{L-1}(m) &= \frac{1}{t_{\rho(0)}^L} \alpha_0^{L-1}. \end{aligned} \quad (4.27)$$

2.  $J = J - 1$ .
3.  $\forall i, J(i) = J$ ,

$$\begin{aligned} \hat{\alpha}_i^J(m) &= \sum_{n=0}^{M-1} a_{i,\rho(i)}^J(m, n) \hat{\alpha}_{\rho(i)}^{J+1}(n) \hat{\beta}_{\rho(i)\setminus i}^{J+1}(n), \\ &= \frac{s_i^J}{s_{\rho(i)}^{J+1} t_{\rho(i)}^{J+1}} \sum_{n=0}^{M-1} a_{i,\rho(i)}^J(m, n) \alpha_{\rho(i)}^{J+1}(n) \beta_{\rho(i)\setminus i}^{J+1}(n), \\ &= \frac{s_i^J}{s_{\rho(i)}^{J+1} t_{\rho(i)}^{J+1}} \alpha_i^J(m). \end{aligned} \quad (4.28)$$

4. If  $J = 0$ , then stop; else return to step 2.

Finally, it is necessary to check the expressions for the state and transition probabilities to account for the scaled  $\hat{\alpha}$ 's and  $\hat{\beta}$ 's. From the equations (4.17), (4.25) and (4.28) it follows:

$$\begin{aligned} \hat{P}(\mathbf{S}^J = m \mid \mathbf{W}, \Theta) &= \frac{1}{K} \sum_{i=0}^{K-1} \frac{\hat{\alpha}_i^J(m) \hat{\beta}_i^J(m)}{\sum_{n=0}^{M-1} \hat{\alpha}_i^J(n) \hat{\beta}_i^J(n)}, \\ &= \frac{1}{K} \sum_{i=0}^{K-1} \frac{\frac{s_i^J}{s_{\rho(i)}^{J+1} t_{\rho(i)}^{J+1}} \alpha_i^J(m) \frac{1}{s_i^J} \beta_i^J(m)}{\sum_{n=0}^{M-1} \frac{s_i^J}{s_{\rho(i)}^{J+1} t_{\rho(i)}^{J+1}} \alpha_i^J(n) \frac{1}{s_i^J} \beta_i^J(n)}, \\ &= P(\mathbf{S}^J = m \mid \mathbf{W}, \Theta), \end{aligned} \quad (4.29)$$

and from the equations (4.18), (4.22) and (4.25)

$$\begin{aligned}
\hat{P}(\mathbf{S}^J = m \mid \mathbf{S}^{J+1} = n, \mathbf{W}, \Theta) &= \frac{1}{K} \sum_{i=0}^{K-1} \frac{\hat{\beta}_i^J(m) a_{i,\rho(i)}^J(m, n)}{\hat{\beta}_{\rho(i),i}^J(n)}, \\
&= \frac{1}{K} \sum_{i=0}^{K-1} \frac{\frac{1}{s_i^J} \beta_i^J(m) a_{i,\rho(i)}^J(m, n)}{\frac{1}{s_i^J} \beta_{\rho(i),i}^J(n)}, \\
&= P(\mathbf{S}^J = m \mid \mathbf{S}^{J+1} = n, \mathbf{W}, \Theta). \quad (4.30)
\end{aligned}$$

Clearly, the equations (4.29) and (4.30) show that it is not necessary to perform additional computation due to scaling. Thus, the substitution of the scaled  $\hat{\alpha}$ 's and  $\hat{\beta}$ 's in the M step does not change the results of the expressions (4.17)-(4.20).

For the next  $(l + 1) - th$  iteration step of the EM algorithm, we use the computed  $\Theta = \Theta_l$  of the  $l - th$  step. In this way the EM converges to a local maximum of the joint pdf  $f_{\mathbf{W}|\Theta}(W|\Theta)$ . There are various criteria when to stop the iteration. In order to simplify the computation, we compare the values of the components of  $\Theta_l$  and  $\Theta_{l+1}$ . If they differ less than some preset value, then the iteration is stopped.

After training the HMT, we are able to compute the joint pdf of all pixels and to classify images using a Bayes classifier. In the next chapter we explain how to obtain the joint pdf of an image and how to classify it.

## CHAPTER 5 MULTISCALE SEGMENTATION

The overall statistical model consists of Hidden Markov Trees  $\mathcal{T}_f$  that assign a node  $i$  to each coefficient<sup>1</sup>  $\mathbf{w}_{i,\mathcal{T}_f}^J$ , at a scale  $J$ , of the same feature  $f$ . We assume that the features are mutually independent. In other words, connecting coefficients that belong only to the same feature  $f$ , we obtain nine mutually independent probability trees  $\mathcal{T}_f$ ,  $f \in \mathcal{F} = \{\pm 15^\circ, \pm 45^\circ, \pm 75^\circ, H, S, I\}$ . Each tree  $\mathcal{T}_f$  represents, in fact, a distinct HMT model, which can be trained separately from the other trees, as explained in the previous chapter. After training, we perform Bayesian classification, where the independent trees  $\mathcal{T}_f$  are unified into a unique statistical model.

As a result of training the HMT for sky and ground images, we obtain the model parameter vectors  $\Theta^s$  and  $\Theta^g$ , respectively. It follows, for each coefficient  $\mathbf{w}_{i,\mathcal{T}_f}^J$  of the tree  $\mathcal{T}_f$ , it is possible to compute the likelihood for the class sky as

$$\begin{aligned} p(\mathbf{w}_{i,\mathcal{T}_f}^J | \Theta_f^s) &= \sum_m p(\mathbf{w}_{i,\mathcal{T}_f}^J, \mathbf{S}_{\mathcal{T}_f}^J = m | \Theta_f^s), \\ &= \sum_m p(\mathbf{w}_{i,\mathcal{T}_f}^J | \mathbf{S}_{\mathcal{T}_f}^J, \Theta_f^s) P(\mathbf{S}_{\mathcal{T}_f}^J | \Theta_f^s), \\ &= \sum_m p(\mathbf{w}_{i,\mathcal{T}_f}^J | \mathbf{S}_{\mathcal{T}_f}^J, \Theta_f^s) P(\mathbf{S}_{\mathcal{T}_f}^{L-1} = m | \Theta_f^s) \prod_J P(\mathbf{S}_{\mathcal{T}_f}^J = m | \mathbf{S}_{\mathcal{T}_f}^{J+1} = n), \end{aligned} \quad (5.1)$$

where  $p(\mathbf{w}_{i,\mathcal{T}_f}^J | \mathbf{S}_{\mathcal{T}_f}^J, \Theta_f^s)$  represents the Gaussian distribution, fully characterized by  $(\mu_f^J)^s$  and  $(\sigma_f^J)^s$ , which are the components of the known  $\Theta^s$ . Clearly, from the assumption that the trees  $\mathcal{T}_f$  are independent, for all  $f \in \mathcal{F}$ , the likelihood of any

---

<sup>1</sup> Here, coefficient refers to the magnitude of a CWT coefficient and/or the  $H$ ,  $S$  and  $I$  color values.

coefficient  $\mathbf{w}_i^J$ , at a scale  $J$ , given a class, say sky, can be computed as

$$p(\mathbf{w}_i^J | \Theta_f^s) = \prod_{f \in \mathcal{F}} p(\mathbf{w}_{i, \mathcal{T}_f}^J | \Theta_f^s) \quad (5.2)$$

Consequently, we are able to perform Bayesian classification at all scales, and combining these results to perform segmentation of an image.

Under successful image segmentation, we imply both accurate classification of large, homogeneous regions, and distinct detection of their boundaries. To achieve this goal, both large and small scale neighborhoods should be analyzed. Therefore, we implement a multiscale segmentation algorithm, based on the algorithm proposed by Choi and Baraniuk [18].

Denoting with  $\mathbf{W}^J$  all coefficients from all features at a scale  $J$ ,  $\mathbf{W}^J = \{\mathbf{w}_i^J\}$ , and with  $\Omega^J$  the collection of all class labels at the scale  $J$ ,  $\Omega^J = \{\omega_i^J\}$ , where  $\omega_i^J \in \{s, g\}$ , the classification is performed according to the MAP rule, as follows:

$$\hat{\Omega}^J = \arg \max_{\Omega^J} \{P(\Omega^J | \mathbf{W}^J)\} \quad , \quad (5.3)$$

$$= \arg \max_{\Omega^J} \{p(\mathbf{W}^J | \Omega^J) P(\Omega^J)\} \quad . \quad (5.4)$$

Thus, our goal is to compute the discriminant function  $g(\mathbf{W}^J, \Omega^J) = p(\mathbf{W}^J | \Omega^J) P(\Omega^J)$ , in (5.4), which we present further in the text.

Assuming that a class label  $\omega_i^J$  completely determines the distribution of the corresponding coefficient  $\mathbf{w}_i^J$ , it follows that all coefficients are mutually independent, given their class labels:

$$p(\mathbf{W}^J | \Omega^J) = \prod_{i \in J} p(\mathbf{w}_i^J | \omega_i^J) \quad , \quad (5.5)$$

where  $p(\mathbf{w}_i^J | \omega_i^J)$  is already known from (5.2).

To compute the joint probability  $P(\Omega^J)$ , we assume that the distribution of  $\Omega^J$  is completely determined by  $\Omega^{J+1}$  at the coarser  $J + 1$  scale. Here, we once again introduce a new Markov tree, where class labels  $\omega_i^J$  play a role analogous to

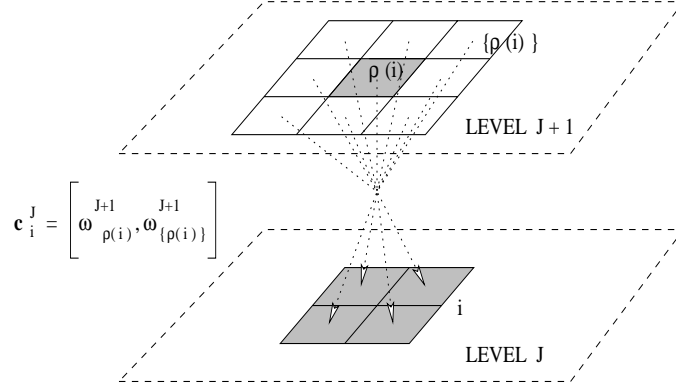


Figure 5.1 The context vector.

the hidden states  $\mathbf{S}_i^J$  in the HMT. Combined with the Markov property that, given  $\omega_{\rho(i)}^{J+1}$ ,  $\omega_i^J$  is independent of all  $\omega_k^J$ ,  $k \neq i$ , the Markov chain rule reads:

$$P(\Omega^J) = P(\Omega^L) \prod_{i \in J} \prod_{j=J}^{L-1} P(\omega_i^j | \Omega^{j+1}) , \quad (5.6)$$

where  $L$  denotes the coarsest level, which is still able to provide statistically reliable segmentations  $\Omega^L$ .

The conditional probability  $P(\omega_i^j | \Omega^{j+1})$  in (5.6), being unknown in general, must be estimated using a prohibitive amount of data. In order to surmount this problem, we introduce contexts [18]. To each coefficient  $\mathbf{w}_i^J$ , with the hidden class label  $\omega_i^J$ , we assign the context  $\mathbf{c}_i^J$ , which represents the information on  $\Omega^{J+1}$ . The choice of  $\mathbf{c}_i^J$ , poised between complexity and accuracy, in our case, is a vector that consists of the two components:

1. the class label  $\omega_{\rho(i)}^{J+1}$ , of the corresponding direct parent (as in the HMT),
2. the majority vote of the class labels  $\omega_{\{\rho(i)\}}^{J+1}$ , where  $\{\rho(i)\}$  denotes the eight neighboring coefficients of the direct parent  $\omega_{\rho(i)}^{J+1}$ .

Recall that in the HMT a coefficient  $\mathbf{w}_i^J$  has exactly one parent  $\mathbf{w}_{\rho(i)}^{J+1}$  and four children  $\mathbf{w}_{\chi(i)}^{J-1}$ . It follows that all four neighboring coefficients at a scale  $J$ , have the same context vector  $\mathbf{c}_i^J = [\omega_{\rho(i)}^{J+1}, \omega_{\{\rho(i)\}}^{J+1}]$ , as illustrated in Figure 5.1.

We assume that  $\mathbf{c}_i^J$  represents a reliable source of information on the distribution of all class labels at  $J + 1$  level. Thus, we rewrite the equation (5.6) as

$$P(\Omega^J) = P(\Omega^L) \prod_{i \in J} \prod_{j=J}^{L-1} P(\omega_i^j | \mathbf{c}_i^j) . \quad (5.7)$$

Finally, we derive a more convenient expression for the discriminant function from (5.4) as

$$g(\mathbf{W}^J, \Omega^J) = P(\Omega^L) \prod_{i \in J} p(\mathbf{w}_i^J | \omega_i^J) \prod_{j=J}^{L-1} P(\omega_i^j | \mathbf{c}_i^j). \quad (5.8)$$

The unknown transition probabilities  $P(\omega_i^J | \mathbf{c}_i^J)$  can be obtained, maximizing the discriminant function (5.8) in cascades, employing the EM algorithm at each scale  $J$  [18]. Using the already known likelihoods  $p(\mathbf{w}_i^J | \omega_i^J)$  from training, the  $P(\omega_i^J | \mathbf{c}_i^J)$  is estimated in the ML sense by averaging over the entire scale  $J$ , because it is reasonable to assume that the transition probabilities are equal for all coefficients at the same scale.

The initial parameters for the EM at the level  $L$  are  $p(\mathbf{w}_i^L | \omega_i^L)$ , obtained in training (5.2). Also, for the coarsest level  $L$ , we assume that  $P(\omega_i^L | \mathbf{c}_i^L) = 1$  for all coefficients. Then, the EM is performed in cascades towards the finer scales, estimating  $P(\omega_i^J | \mathbf{c}_i^J)$  and  $P(\omega_i^J)$ , until the finest level  $J = 0$  is reached. Experimental results show that the algorithm converges rapidly if the initial parameters are set to the values estimated at the previous coarser scale.

Once the transition probabilities  $P(\omega_i^J | \mathbf{c}_i^J)$  are estimated for all class labels and context values, it is possible to perform image segmentation, using the MAP rule as follows:

$$\hat{\omega}_i^0 = \arg \max_{\omega_i^0 \in \{s, g\}} P(\omega_i^0) P(\omega_i^0 | \mathbf{c}_i^0) p(\mathbf{w}_i^0 | \omega_i^0) . \quad (5.9)$$

In the next chapter we present experimental results for sky/ground image segmentation, using the proposed computer vision system.



## CHAPTER 6 EXPERIMENTAL RESULTS

In our experiments, first, the HMT model is trained using a data base of training images. We recorded two sets of 500 images. One set represents the sky and the other ground, only. Images are carefully chosen to account for great variability within classes, as illustrated in Figure 6.1.

Then, the trained HMT is tested on another set of test images. There are two types of testing images. The first testing set contains images similar to the training data base. The second type of testing images consists of flight images of our MAV's, where both the sky and ground appear. In the following sections we present the experimental set-up and results.

### 6.1 Real-Time Realization Problems

In order to implement the HMT-based Bayes classifier in real-time, which is our ultimate goal, the amount of computation must be as small as possible. There are several avenues to decrease the amount of computation. The greatest computational load comes from the EM algorithm. It is known that given  $M$  training images of  $N$  pixels each, the total computational cost per EM iteration is  $O(MN)$  [7].

Thus, first, we reduce  $N$ , the number of pixels to be analyzed. Since the fast CWT algorithm requires image dimensions be a power of 2, we subsample each  $640 \times 480$  image in the data base to dimensions  $128 \times 128$ . Hence, the maximum number of levels in the HMT is  $L = \log_2 127 = 7$ . Segmentation results show that the  $128 \times 128$  image resolution is sufficient for reliable classification.

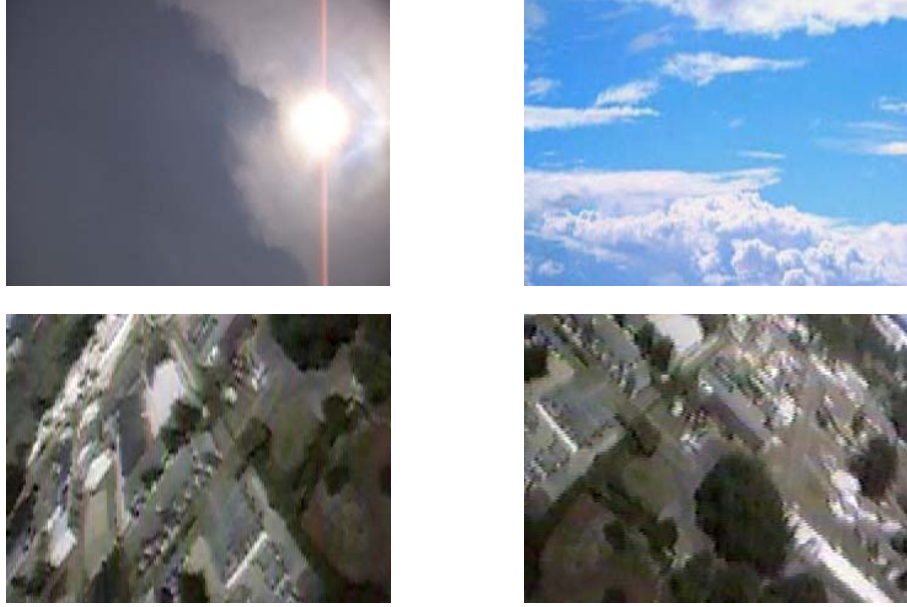


Figure 6.1 Training images of the sky and ground.

Second, with reducing the number of HMT parameters, we lower the number  $M$  of images necessary for training the HMT. Recall that we compute only the components of  $\Theta$  for all coefficients at a scale  $J$ —not for each particular coefficient  $\mathbf{w}_i^J$ . This reduces the number of iteration steps for the EM to converge. In our experiments, the convergence criterion compares values of the components of  $\Theta_{l+1}$  with the values from the previous iteration step in  $\Theta_l$ . The EM algorithm stops if

$$|\Theta_{l+1} - \Theta_l| < \varepsilon ,$$

where  $\varepsilon$  should not be an arbitrarily small value, because it would lead to overfitting the HMT model. If  $\varepsilon = 10^{-4}$ , the convergence is achieved in 5 iteration steps, in the worst case.

Finally, intelligent choice of initial values also reduces the number of iteration steps. Appropriate setting of all the parameters and the code optimization made the execution of training relatively fast. Training on 900 images (450 of each class), on an Athlon 900 MHz PC, is less than 3 minutes, and it takes less than 1 minute, for testing 100 images (50 of each class).

## 6.2 Classification of the First Type of Test Images

Having trained the HMT on the training set, we obtained parameters  $\Theta$  for all features  $f \in \{\pm 15^\circ, \pm 45^\circ, \pm 75^\circ, H, S, I\}$ . Then, we tested the performance of our computer vision system on the first type of test images, where either the sky or ground is present. Here, the goal is to recognize the whole image either as the sky or as the ground.

First, we show that the choice of H, S, I color features is justified. As we have already pointed out, the R, G, B color features exhibit significant statistical dependence. Here, we present the classification error of our algorithm for both R, G, B features and H, S, I features in Table 6.1 and Table 6.2. Better results are achieved when the number of possible color states is increased. However, a better model is computationally more expensive.

Table 6.1 The performance of the RGB-based classifier for ground test images.

class	number of states	error HSI	error RGB
ground	2	4 %	6 %
	3	2 %	4 %
	4	2 %	4 %
	5	2 %	4 %
	6	2 %	4 %

Table 6.2 The performance of the RGB-based classifier for sky test images.

class	number of states	error HSI	error RGB
sky	2	12 %	26 %
	3	10 %	18 %
	4	6 %	16 %
	5	6 %	12 %
	6	6 %	10 %

Comparing the classification results for HSI-based and RGB-based classifier, presented in Figure 6.2, we conclude that the HSI color feature space is superior to the RGB space.

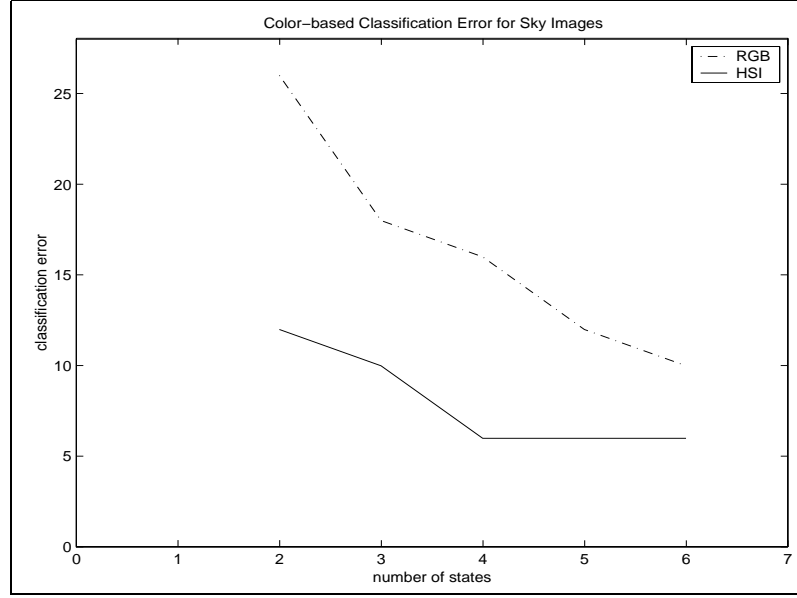


Figure 6.2 The classification error using RGB and HSI color spaces.

The performance of the wavelet-based classifier, the case when only the CWT features are used ( $f \in \{\pm 15^\circ, \pm 45^\circ, \pm 75^\circ\}$ ), is presented in Table 6.3.

Table 6.3 Performance of the wavelet-based classifier.

class	error CWT
ground	2 %
sky	6 %

From Tables 6.2, 6.1, and 6.3, we conclude that even when the number of states is increased the color-based classifier performs worse than the wavelet-based classifier. Therefore, for reliable classification, both color and texture features have to be incorporated.

### 6.3 Segmentation of the Second Type of Test Images

Having trained the HMT for all features  $f \in \{\pm 15^\circ, \pm 45^\circ, \pm 75^\circ, H, S, I\}$ , we tested the performance of our computer vision system on the second type of test images. This set of images consists of flight images of our MAV's, as well as other difficult-to-segment sky/ground images. Here, the goal is to segment an image into sky and ground regions. The image segmentation is presented in Figures 6.3-6.5.

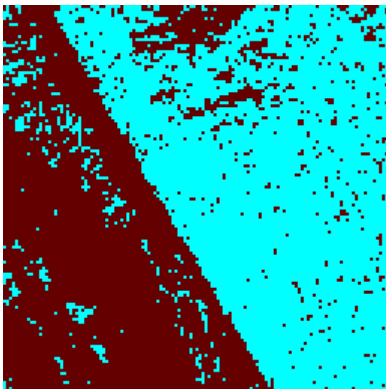


Figure 6.3 A MAV's flight: favorable sky/ground patterns.

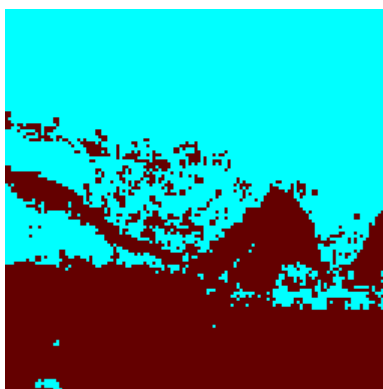


Figure 6.4 Water surface with ice patches similar to clouds.

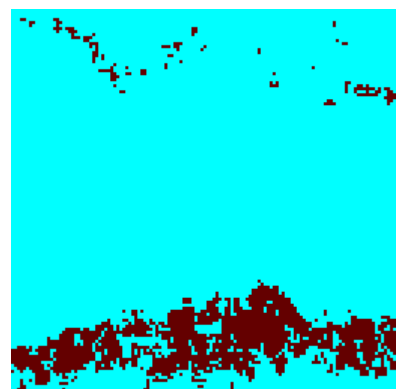


Figure 6.5 A mountain view with fuzzy horizon line.

Finally, we illustrate the generality of our algorithm in segmentation of cut/uncut grass regions (see Figure 6.6). We present the original image (left), the subsampled image (center), and segmentation of the subsampled image (right). These images<sup>1</sup> show a grass lawn from the perspective of a camera mounted on an autonomous lawn mower. Even at image resolutions as low as  $64 \times 64$ , we achieve satisfactory results at very fast processing speeds.

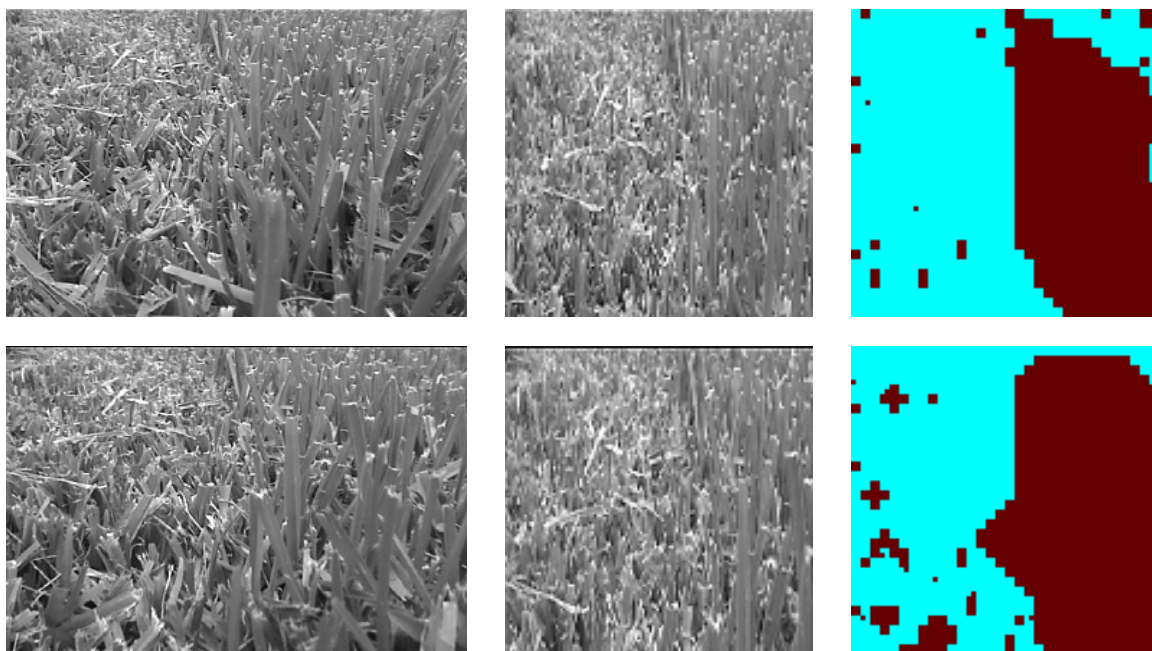


Figure 6.6 Segmentation of cut and uncut grass regions.

---

<sup>1</sup> We thank Rand Chandler for giving us access to his image database.

## CHAPTER 7 CONCLUSION

Segmentation of complex image classes, such as sky and ground, demand an elaborate consideration of class properties. Clearly, in some cases, color provides sufficient information for sky and ground detection. However, due to video noise and/or unfavorable class patterns, both color and texture clues are necessary for successful recognition.

In this thesis, first, we presented our choice of features, consisting of H and I values from the HSI color space, and CWT coefficients. In our experiments, the HSI color space proved to be superior to the RGB space. Also, in the early stage of considering the method for texture analysis, experimental results showed that the DWT (realized with Daubechies wavelets) is inferior to the CWT.

Second, we showed the implementation of the HMT model and the training steps for obtaining its parameters. The contribution of this thesis reflects in the derivation of the formulae, which must be used in the EM algorithm to surmount the implementation issues; specifically the numerical underflow. We proposed the method, where dynamic scaling is performed for  $\beta$  values only, whereas in literature [7, 11, 15] other approaches are used.

Further, we described how the learned parameter set could be used for computing likelihoods of all nodes at all scales of our HMT. We then developed multiscale Bayesian classification for our application.

The most important contribution of this thesis reflects in successful implementation of color features into the HMT framework. In relevant literature [7, 11, 18], the HMT model is related only to wavelets. Incorporating color features resulted

in more reliable image segmentation, which is illustrated for diverse sky/ground images and for cut/uncut grass images.

We incorporated in our design results from the available literature, modifying the original algorithms for our purposes where appropriate. We designed our computer vision system with real-time constraints in mind. Therefore, some aspects are suboptimal with respect to the classification error.



## REFERENCES

- [1] Scott M. Ettinger, Michael C. Nechyba, Peter G. Ifju, and Martin Waszak, “Vision-guided flight stability and control for Micro Air Vehicles,” in *Proc. IEEE Int. Conference on Intelligent Robots and Systems*, Lausanne, October 2002.
- [2] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, John Wiley & Sons, New York, 2nd edition, 2000.
- [3] Heng-Da Cheng, Xihua Jiang, Ying Sun, and Jingli Wang, “Color image segmentation: advances and prospects,” *Pattern Recognition*, vol. 34, pp. 2259–2281, 2001.
- [4] Trygve Randen and Hakon Husoy, “Filtering for texture classification: A comparative study,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–310, 1999.
- [5] Stephane Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, 2nd edition, 2001.
- [6] Stephane G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
- [7] Matthew S. Crouse, Robert D. Nowak, and Richard G. Baraniuk, “Wavelet-based statistical signal processing using Hidden Markov Models,” *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, 1998.
- [8] Jerome M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.
- [9] Nick Kingsbury, “Image processing with complex wavelets,” *Philosophical Transactions Royal Society London*, vol. 357, no. 1760, pp. 2543–2560, 1999.
- [10] Michael Unser, “Texture classification and segmentation using wavelet frames,” *IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1549–1560, 1995.
- [11] Diego Clonda, Jean-Marc Lina, and Bernard Goulard, “Mixed memory model for image processing and modeling with complex Daubechies wavelets,” in *Proc. SPIE’s International Symposium on Optical Science and Technology*, San Diego, August 2000.

- [12] Nick Kingsbury, “Complex wavelets for shift invariant analysis and filtering of signals,” *Journal of Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234–253, 2001.
- [13] Judea Pearl, *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, 1st edition, 1988.
- [14] Justin K. Romberg, Hyeokho Choi, and Richard G. Baraniuk, “Bayesian tree-structured image modeling using wavelet-domain Hidden Markov Models,” *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1056–1068, 2001.
- [15] Lawrence R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [16] Geoffrey J. McLachlan and Krishnan T. Thriyambakam, *The EM Algorithm and Extensions*, John Wiley & Sons, New York, 1st edition, 1996.
- [17] Helmut Lucke, “Which stochastic models allow Baum-Welch training?,” *IEEE Transactions on Signal Processing*, vol. 44, no. 11, pp. 2746–2756, 1996.
- [18] Hyeokho Choi and Richard G. Baraniuk, “Multiscale image segmentation using wavelet-domain Hidden Markov Models,” *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1309–1321, 2001.

## BIOGRAPHICAL SKETCH

In 1994, Sinisa Todorovic graduated at the School of Electrical Engineering, the University of Belgrade, Yugoslavia, with a major in electronics and telecommunications. From 1994 until 2001, he worked as a software engineer in communications industry. Specifically, from 1998 to 2001, as an employee of Siemens, he worked on a huge network installation project in Russia. In fall 2001, Sinisa Todorovic was admitted to the master's degree program at the Department of Electrical and Computer Engineering, University of Florida, Gainesville. His main field of interest, throughout his graduate studies, was computer vision and pattern recognition.