COMPRESSION AND SEGMENTATION OF IMAGES USING AN INTER-SUBBAND WAVELET PROBABILITY MODEL

ROBERT W BUCCIGROSSI

A DISSERTATION

in

COMPUTER AND INFORMATION SCIENCE

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy.

1999

Supervisor of Dissertation

Graduate Group Chairperson

ABSTRACT

COMPRESSION AND SEGMENTATION OF IMAGES USING AN INTER-SUBBAND WAVELET PROBABILITY MODEL

Robert W Buccigrossi

Eero P Simoncelli

Study of the statistical qualities of images can lead to elegant solutions for many classical image processing and computer vision problems. Image compression, restoration, interpolation, and texture segmentation can be written in terms of probabilistic decision making, where the prior model is the set of "natural images." Recently, we introduced a statistical characterization of natural images in the wavelet transform domain. This characterization describes the joint statistics between pairs of subband coefficient magnitudes at adjacent spatial locations, orientations, and scales. This paper describes our work in image compression and unsupervised segmentation that utilizes inter-subband dependencies. In order to support unsupervised segmentation with a large feature set, this paper presents a novel unsupervised algorithm to reduce feature dimensionality called "Unsupervised Linear Discriminant Analysis" (ULDA) based upon Fisher's Linear Discriminant Analysis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

What is an image? For most people, this question must seem rhetorical. Since we were born, our eyes have been bombarded with images, and through that experience, (and potentially some evolutionary programming) we just *know* what images are. For example, people intuitively know that a picture is noisy, even though previously they may never have seen it. Figure 1.1 displays the same picture with different levels of additive independent Gaussian noise. One would not believe that the "clean image" is actually the image at the right of the figure.

Even with such intuitive knowledge, a definition of "natural images" that encompasses its statistical properties has been elusive. Why would such a definition be useful? Many image processing and early computer vision tasks like compression, restoration, interpolation, and texture segmentation can be described as statistical decision making problems with the distribution of natural images playing an important role. Therefore, finding inherent properties of images can lead to better solutions of these problems.

Recently, we introduced a novel probability model relating the coefficients between and among subbands of a wavelet decomposition of an image [BS97, SB97]. This probability model has been successfully applied in image compression, restoration, and texture synthesis [Sim97]. Work presented here displays that this model is also applicable to the computer vision task of texture segmentation.

This introduction continues with a brief review of image models and statistical properties within the literature. Later sections describe our image model, a new compression

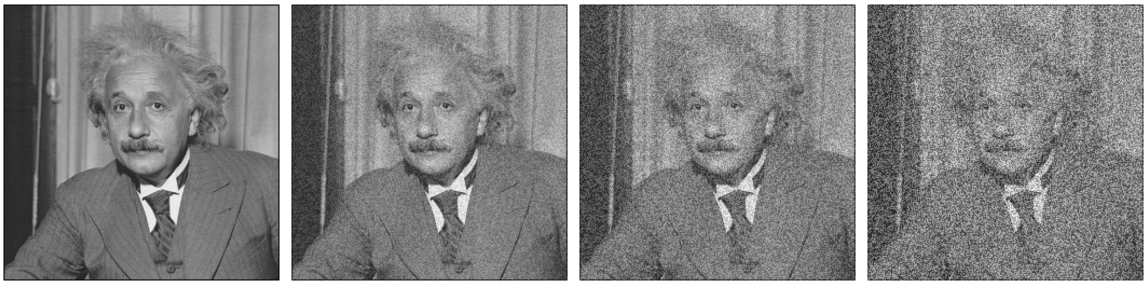


Figure 1.1: Images with progressively stronger added noise. People can easily differentiate between the noise and the image, though previously they may never have seen the original image.

algorithm and a new texture segmentation algorithm based upon the model. In order to support unsupervised segmentation with a large feature set, this paper also presents a novel unsupervised algorithm to reduce feature dimensionality called "Unsupervised Linear Discriminant Analysis" (ULDA) based upon Fisher's Linear Discriminant Analysis.

1.1 Physical Models of Images

Models of images can generally be separated into two types: physical and statistical. Physical models are developed through analysis of the physical interaction of light and objects and the final projection of the 3-D scene onto a 2-D plane. Statistical models are the result of analysis and generalization of the statistical properties found in a set of training images, typically ignoring the actual imaging process. A review of statistical models is the subject of section 1.2.

An image is formed when light from a three dimensional world is projected upon a two dimensional plane, either through the lens of an eye upon the retina, or the lens of a camera onto film or some other photo-sensitive material. The simplest perspective model is the "pinhole camera" where light must pass through an infinitely small hole[1] (Figure 1.2(a)). The projection of a point (x, y, z) in the world to the image plane (x', y') is given by:

$$\begin{aligned} x' &= \frac{fx}{z} \\ y' &= \frac{fy}{z} \end{aligned}$$

[1] Other perspective models, such as the "thin lens" and "thick lens" models, take into account the physics of the focusing instrument which cause blurring of objects away from the focal plane.

where f is the distance between the pinhole and the imaging plane. This implies that light originating from the same direction relative to the pinhole is projected onto the same point on the image plane.

Light usually is not directly created by objects in a scene. Randomly, some photons bounce from a light source, between the objects of a scene, and eventually through the lens of an eye or a camera. When a photon hits an object's surface, it is either (Figure 1.2(b)):

- absorbed (causing the color of an object)
- passed through the object (transparency)
- scattered in a random direction by the microscopic structure of the surface (Lambercian reflection)
- or reflected across the surface normal like a mirror (specularity).

Shadows are formed when an object blocks the light from hitting other objects. Light can reflect from one object surface to another causing mirror reflections or color bleeding. These complex interactions between photons and objects create a myriad of effects, from the dancing rainbows and reflections of a diamond to the dull blackness of a clean chalk board.

Researchers in computer graphics have successfully modeled the imaging process by representing a scene as a tree hierarchy of objects (Figure 1.3) [FvDFH96]. The objects' surfaces are then broken further into texture-mapped polygons (usually triangles use with graphics hardware). Each polygon is given reflectance properties defining its color, transparency, and specularity. A combination of ambient light, and point or polygon light sources are used to simulate scene illumination.

To create an image, a pinhole camera is placed into the scene. Depending upon the complexity of the graphics algorithm, the interaction of light from the sources and each surface, and from one surface to the next (radiosity) is calculated. Finally, light reflecting from the objects' surfaces are projected through the pinhole camera onto a 2-D surface, resulting in a photo-realistic image.

In computer vision, Hinton *et al.* use a very simplified version of this physical model for handwritten number recognition [HRD95]. Each pixel of an image is assumed to originate

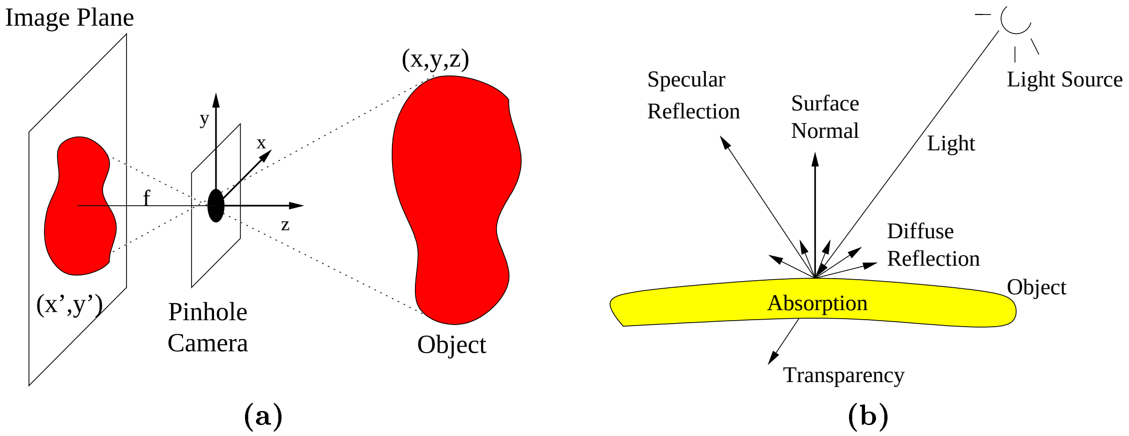


Figure 1.2: A model of image formation. **(a)** Light from objects are collected through a pinhole camera onto the image plane. **(b)** Light impacting upon the surface of an object can be absorbed, reflected like a mirror, reflected in a random direction, or can pass through the object.

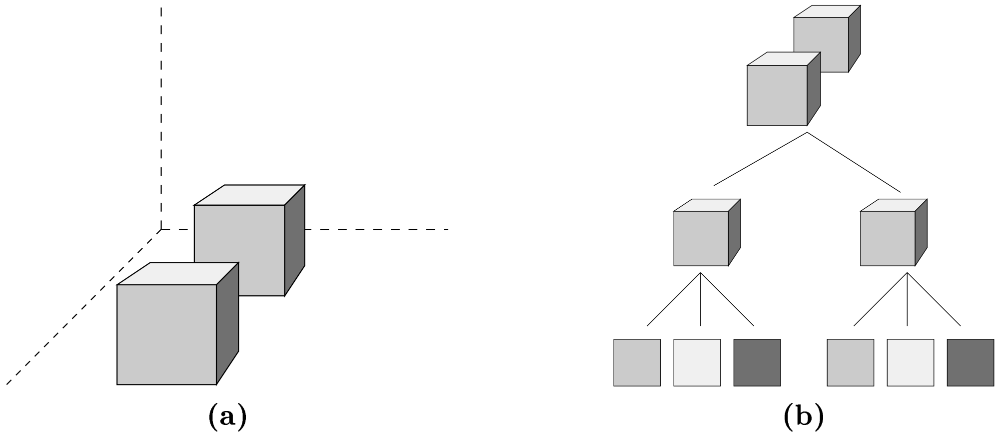


Figure 1.3: Tree hierarchy for a scene. **(a)** Simple scene consisting of two objects. **(b)** A tree representing the objects of the scene in (a). The original scene is recursively decomposed into smaller objects.

from either the background or one of ten classes of objects (the digits "0" through "9"). For each digit, Hinton creates a mini neural network, called an "autoencoder," to recognize the particular digit. The "correct" interpretation of the numbers is the one that gives the best reconstruction of the data.

Using the physical model is possible in number recognition because of the limited structure of the problem. There are only ten possible objects which are not capable of occluding each other. Light is assumed to be ambient, and color is ignored. The overwhelming complexity of general scenes has made it difficult to apply the physical model for three dimensional vision and image processing applications.

1.2 Statistical Models of Images

Because the statistical properties of images are difficult to discern from the image making process, statistical models have been developed through direct analysis of images. When compared to physical models, statistical models are less satisfactory in terms of defining a "natural image." In most cases, statistical models are conjectures which cannot be proven to exist. Even with these difficulties, statistical models are important since they can be applied directly in many image processing and computer vision applications.

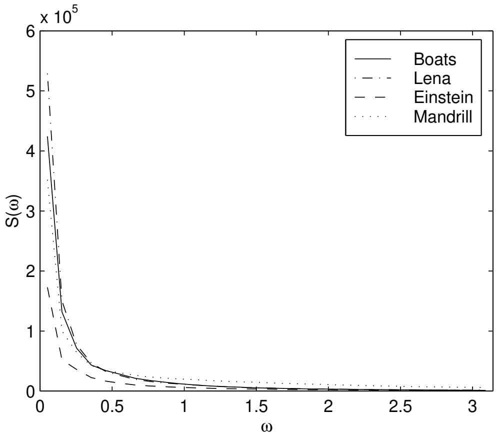
1.2.1 Scale Invariance in the Power Spectrum

One property, first mentioned by Deriugin [Der56], is that the energy in the frequency domain $S(\omega)$ (averaged over all orientations) tends to decay with the frequency:

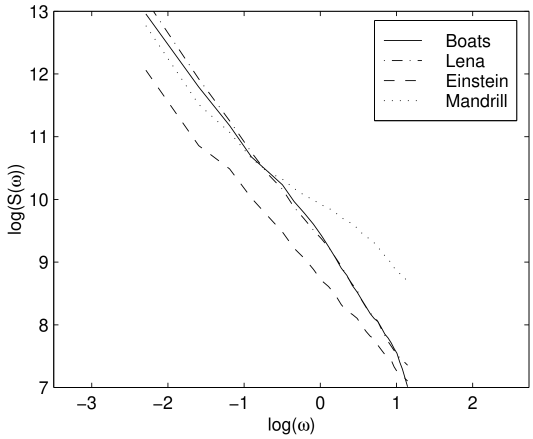
$$S(\omega) \propto \omega^{-2+\eta} \tag{1.1}$$

where η is small [Rud94]. Figure 1.4(a) portrays the relationship between $S(\omega)$ (approximated through the Discrete Fourier Transform) and ω for four images. Figure 1.4(b) contains the same information in the log/log domain, resulting in approximately straight lines with a slope of -2.

Intuitively, because images are 2-D projections of a 3-D world, the set of natural images should be closed under expansion or contraction, since this roughly corresponds to moving along the z axis of the 3-D world. This intuition leads to the claim of *scale invariance*:



(a)



(b)

Figure 1.4: $1/\omega^2$ power spectrum of images. **(a)** The squared magnitude of the Discrete Fourier Transform vs. the frequency. **(b)** The same plot as (a) in the log/log domain. These are approximately straight lines with -2 slope, demonstrating the $1/\omega^2$ relationship.

that image statistics do not change with scale. The $1/\omega^2$ frequency distribution does obey scale invariance because expansion or contraction of an image results in a scaling of the frequency and power, which maintains the $1/\omega^2$ shape[2].

1.2.2 Local Pixel Correspondence

In the physical model, image scenes are formed by an overlapping of objects, each further decomposed into smaller parts. Local points in the resulting image generally come from the same object. Therefore, the intensity of a pixel is greatly correlated with that of its surrounding neighbors. Figure 1.5(a) depicts the conditional distribution of a pixel given the intensity of its left neighbor collected over four images. A column of the graph represents the distribution of the pixel intensity for a given intensity of the neighbor. Notice that most of the mass of the density lies along the diagonal, implying that the pixel and its neighbor predominantly agree. Figure 1.5(b) exemplifies this relationship by showing the normalized histogram of the difference in intensities between a pixel and its left neighbor for the same images. The histogram is greatly peaked at 0, again implying that the neighboring pixel intensities predominantly agree.

[2]It should be noted that any frequency distribution of the form $1/\omega^n$ would obey scale invariance.

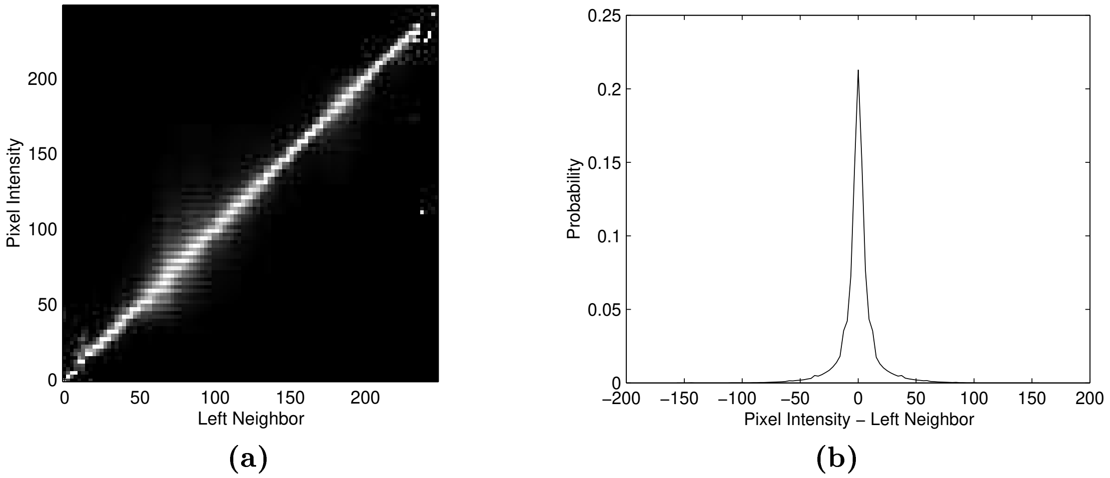
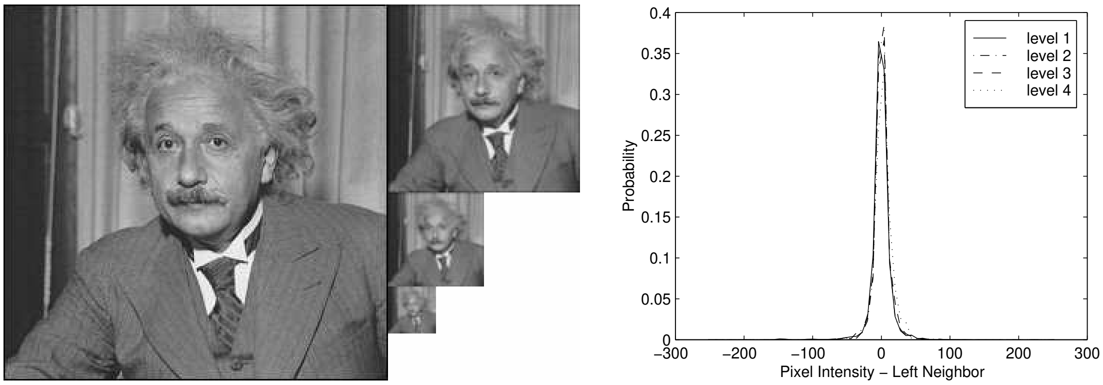


Figure 1.5: Correspondence of local pixel intensities. **(a)** The conditional histogram of a pixel intensity given its left neighbor collected over four images. **(b)** The normalized histogram of the difference of a pixel and its left neighbor. Since the distribution is peaked at 0, there is a high correlation between local pixel intensities.

This correspondence between local pixels led to the use of Markov random field (MRF) models in the image domain [CJ83]. In an MRF model, the distribution of a random variable X given a neighbor set Q_X is independent of all other random variables Y (which are neither functions of X nor of Q_X):

$$\mathcal{P}\left(X \mid Q_X, Y\right) = \mathcal{P}\left(X \mid Q_X\right).$$

In addition, for an MRF model, the relative position of the neighbor set Q_X with X is *strict-sense stationary*, meaning that the probably distributions are the same for any location of X in the image. Another common assumption is that these distributions are *ergodic*, implying that the moments of N samples of the random variable X converge to the moments of the underlying probability distribution of X as the number of samples N approaches infinity. With the combination of ergodicity and strict-sense stationarity, the statistics of the Markov distribution can be estimated from the histogram of the image itself. Cross and Jain achieve simple texture synthesis using an MRF model where $\mathcal{P}\left(X \mid Q_X\right)$ is a binomial distribution (in which X and Q_X are discrete random variables) whose parameter depends on a linear combination $l(Q_X)$ of the local pixel intensities [CJ83].



(a) **(b)**

Figure 1.6: Multi-scale correspondence of local pixel intensities. **(a)** A four level Gaussian pyramid of a picture of Albert Einstein **(b)** The normalized histograms of the difference of a pixel and its left neighbor for the four levels of the pyramid. Notice that the distribution is peaked at 0 for the different scales of the image.

Another popular model is the Gaussian Markov random field (GMRF) given by:

$$\mathcal{P}\left(X = x \mid Q_X = \vec{b}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[-\frac{(x-l(\vec{b}))^2}{2\sigma^2}\right]} \tag{1.2}$$

where $l(\vec{b})$ is a linear predictor formed from neighbor values and σ is the variance of $(X - l(Q_X))$ [KC97]. It should be noted from the sharp peak in Figure 1.5(b) that a Gaussian does not properly model the local correspondences, but even so, the GMRF is widely used because of its mathematical tractability.

1.2.3 Multi-Scale Local Pixel Correspondence

The GMRF model of Equation (1.2) does not encompass the scale invariance of images. If images are truly scale invariant, then there should be local pixel correspondences in contracted versions of images. Figure 1.6(a) shows a four level Gaussian pyramid created from a picture of Albert Einstein[3]. Visually, higher (smaller) levels of the pyramid correspond to contractions of the original image. Figure 1.6(b) shows the local intensity differences in each of the four levels of the pyramid from a pixel to its left neighbor. As can be seen from the figure, the local intensity correspondence is preserved across scale.

To utilize multi-scale pixel correspondences in texture segmentation, Krishnamachari

[3]The Gaussian pyramid is described in appendix section A.1

and Chellappa [KC97] used a multi-scale Gaussian Markov random field model. In a multi-scale GMRF, a GMRF is created for every level of a Gaussian pyramid. It is known that the Markov property is not preserved under convolution and subsampling, therefore the GMRF's at higher levels are approximations of the model given by the GMRF at the first level of the pyramid.

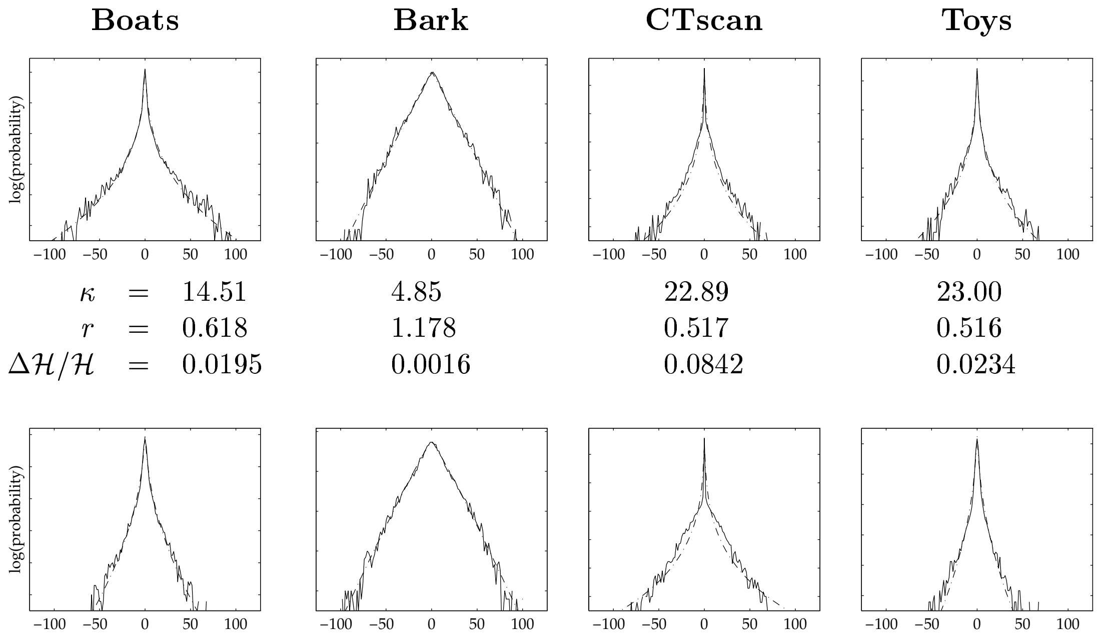
Another way to utilize the multi-scale pixel correspondence is with a wavelet pyramid[4] [Vet84]. The coefficients of the non-lowpass subbands can be interpreted as local differences taken from contracted versions of the original image, so it is not surprising that their point histograms are highly peaked at zero. A number of authors have observed that wavelet subband coefficients have highly non-Gaussian statistics [Mal89, Fie94, OF96, SA96]. The intuitive explanation for this is that images typically have spatial structure consisting of smooth areas interspersed with occasional edges or other abrupt transitions. The smooth regions lead to near-zero coefficients, and the structures give occasional large-amplitude coefficients.

Histograms[5] for subbands of separable wavelet decompositions of several images are plotted in figure 1.7. Compared to a Gaussian, these densities are more sharply peaked at zero, with more extensive tails. To quantify this, we give the sample Pearson Kurtosis κ below each histogram. The Pearson Kurtosis (fourth moment divided by squared second moment) is a measure of how "peaked" a distribution is [AS72]. The kurtosis of a Gaussian is 3, so if a distribution has a kurtosis greater than 3 (therefore having a sharper peak), it is called "leptokurtic." The estimated kurtoses of all of the subband densities are significantly greater than 3 and demonstrate that they are leptokurtic. These examples were computed for subbands of an orthonormal separable wavelet decomposition (see Chapter 3 for details), but they would be similar for any octave-bandwidth subband.

These non-Gaussian densities should be contrasted with statistics of frequency-based decompositions which are approximately Gaussian. Since the Gaussian is the maximal-entropy distribution for a given variance [CT91], wavelet-based coders are able to achieve higher degrees of compression than frequency-based coders such as JPEG. The non-Gaussianity of wavelet marginals (being leptokurtic) may be taken as an indication that the wavelet

[4] The wavelet pyramid is described in appendix section A.2

[5] By considering these as representative of the underlying coefficient densities, we are making implicit assumptions of strict-sense stationarity and ergodicity.



Top row (vertical orientation):

	Boats	Bark	CTscan	Toys
κ =	14.51	4.85	22.89	23.00
r =	0.618	1.178	0.517	0.516
$\Delta\mathcal{H}/\mathcal{H}$ =	0.0195	0.0016	0.0842	0.0234

Bottom row (diagonal orientation):

	Boats	Bark	CTscan	Toys
κ =	17.97	4.71	13.99	41.49
r =	0.566	1.207	0.628	0.426
$\Delta\mathcal{H}/\mathcal{H}$ =	0.0142	0.0013	0.0717	0.0327

Figure 1.7: Examples of 256-bin subband coefficient histograms (solid lines) fitted with the density of equation (1.3) (dotted lines), plotted in the log domain. Subbands correspond to four different images (a landscape, a texture, a medical image, and a synthetic graphics image) and two different orientations (top row: vertical, bottom row: diagonal), at the second highest frequency scale. Below each graph is the sample kurtosis κ (fourth moment divided by squared variance), the exponent r used in the model, and the ratio of the relative entropy $\Delta\mathcal{H}$ (Kullback-Leibler divergence) between the histogram and the model and the histogram entropy $\mathcal{H}$. The Bark image had the most Gaussian marginals in our image set, and the CTscan image was the worst fit (in terms of relative entropy).

basis is more appropriate for image representation than either pixel or Fourier representations.

These coefficient statistics have been previously modeled [Mal89, JF95, SA96, LRO97, JJK$^+$97] using a two-parameter "generalized Laplacian" (or "stretched exponential") density function of the form:[6]

$$f_{s,r}(c) = \frac{e^{-|c/s|^r}}{N(s,r)}, \tag{1.3}$$

[6] The lowpass subband coefficients are modeled using a uniform distribution.

where $N(s,r) = 2s\Gamma(1/r)/r$, and $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\ dt$, the Gamma function. The parameters $\{s,r\}$ are directly related to the second and fourth moments. Specifically:

$$\sigma^2 = \frac{s^2\Gamma(\frac{3}{r})}{\Gamma(\frac{1}{r})}, \qquad \kappa = \frac{\Gamma(\frac{1}{r})\Gamma(\frac{5}{r})}{\Gamma^2(\frac{3}{r})}, \tag{1.4}$$

where σ^2 is the distribution variance, and κ is the kurtosis.

We solve (numerically) for the parameters $\{s,r\}$ by minimizing the relative entropy (also known as the "Kullback-Leibler divergence") between a discretized model distribution and the 256-bin coefficient histogram [BS97]:

$$E(s,r) = -\sum_{n=1}^{256} h_n \log_2 \frac{\bar{f}_{s,r}(c_n)}{h_n},$$

where c_n represents the center of the domain of histogram bin n, h_n is the normalized histogram count (frequency) for histogram bin n, and $\bar{f}_{s,r}(c_n)$ is the integral of the density given in equation (1.3) over histogram bin n. The measure $E(s,r)$ corresponds to the cost (in bits) of encoding the data with an entropy coder that assumes the distribution $f_{s,r}(c)$ [CJ83]. For the images in our sample set, σ^2 is roughly proportional to β^l (where l indicates the scale or "pyramid level"), with $\beta \in [3.5, 10]$. The exponent r is typically in the range $[0.5, 1.0]$, corresponding to kurtosis values in the range $[6, 25.2]$.

We make no claim of optimality or uniqueness for this model: Other authors [ZWM96] have used alternative density functions to describe these distributions. Nevertheless, the fits are quite good. Figure 1.7 shows the log-domain plots of the sample histograms together with plots of the fitted density function of equation (1.3). We have included both the best and worst cases from the set of images in our test set (figure 3.2). Below each figure is the relative entropy between the histograms and fitted densities.

Figure 1.8 shows a scatterplot comparing the encoding cost using the model of equation (1.3), and the encoding cost assuming a Gaussian density vs. the encoding cost assuming accurate knowledge of a 256-bin histogram (calculated by the ideal first-order entropy[7]). The Gaussian examples were computed with the distribution variance matched to the sample variance. Note that the relative entropy of the generalized Laplacian model is less than 0.25 bits/coefficient for our sample images, as compared with the Gaussian density model which often has a relative entropy greater than 1.0 bit/coefficient.

[7] Entropy is described in appendix section A.4

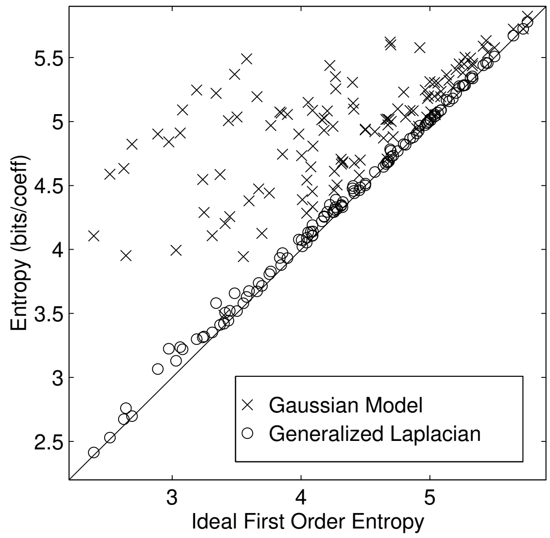


Figure 1.8: Comparison of encoding costs. Plotted are encoding cost assuming the generalized Laplacian density of equation (1.3) (O's), and the encoding cost assuming a Gaussian density (X's), versus the encoding cost using a 256-bin histogram. Points are plotted for 9 bands (3 scales, 3 orientations) of the 13 images in the sample set of figure 3.2. The average relative entropy (K-L distance) of the Gaussian models given the ideal is 0.592 bits/coefficient, while the average relative entropy of the generalized Laplacian models given the ideal is 0.035 bits/coefficient.

1.2.4 Inter-Subband Pixel Correspondence

Wavelet decompositions efficiently capture the local pixel correspondences across scale, but they also reveal other interesting properties of natural images. Figure 1.9 shows the magnitudes of a three level decomposition of "Einstein." Shapiro noted that regions of low-magnitude coefficients (shown in white) tend to occur at areas associated with the same spatial location of the original image, at the same orientation in the pyramid, but at different scales. With this he developed the Embedded Zero-Tree Wavelet (EZW) encoder which represents large sections of zeros in the pyramid with special flags [Sha93]. Through efficient ordering of the pyramid coefficients, Said and Pearlman [SP96] refined EZW into an encoder called SPIHT, which is among the state of the art in image coding.

Inter-subband correspondences have been used indirectly by other researchers. Pentland *et al.* developed a fractal based image coder that used a vector quantizer to predict fine scale coefficients from coarser scales [PS92, Ph93]. Rinaldo *et al.* and Schwartz *et al.* used adaptive arithmetic coding to capture conditional statistics between magnitudes of different subbands [RC95, SZB95]. Wu and Chen [WC97] extended the EZW encoder with

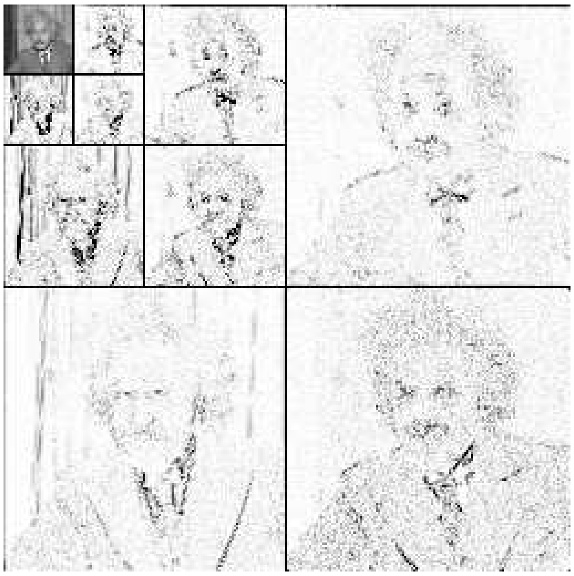


Figure 1.9: Magnitudes of a Wavelet Pyramid. Shown are the subband coefficients magnitudes of a 3-level wavelet pyramid. Higher magnitudes are shown in black for easier viewing. Regions of low or high-magnitude coefficients tend to be located at areas associated to the same spatial positions of the original image at adjacent scales.

an adaptive arithmetic encoder using local coefficient "contexts." De Bonet and Viola used the wavelet pyramid of a texture as a look-up table to generate new textures by directly sampling vectors of coefficients in the same spatial location [Bon97]. Note, however, that these techniques do not directly characterize the statistical properties of images. This paper presents a statistical model for the inter-subband relationships that leads to exciting improvements in image compression and unsupervised texture segmentation.

Chapter 2

Multi-Scale Subband Image Model

2.1 Joint Magnitude Statistics

In the previous section, we explored the marginal distributions of wavelet pyramid subbands. In addition, we found that there appear to be correspondences between the magnitudes of local coefficient neighbors. An image coder using only the marginal distributions assumes independence between local magnitudes, which is the highest entropy joint distribution for a given set of marginals [CT91]. Improvement in coding can be achieved by analyzing the true statistical relationships between wavelet coefficient magnitudes.

Consider two coefficients representing information at adjacent scales, but the same orientation (e.g, horizontal) and spatial location. As in the previous section, we will assume strict-sense stationarity and ergodicity, which allows us to consider the joint histogram of this pair of coefficients, gathered over the spatial extent of the image, as representative of the underlying statistics. Figure 2.1A shows the conditional histogram $\mathcal{H}\left(C|P\right)$ of a "child" coefficient conditioned on a coarser-scale "parent" coefficient. The histogram illustrates several important aspects of the relationship between the two coefficients. First, they are second-order decorrelated, since the expected value of C is approximately zero for all values of P. Second, the variance of the conditional histogram of C clearly depends on the value of P. Thus, although C and P are uncorrelated, *they are still statistically dependent.* Furthermore, this dependency cannot be eliminated through further linear transformation.

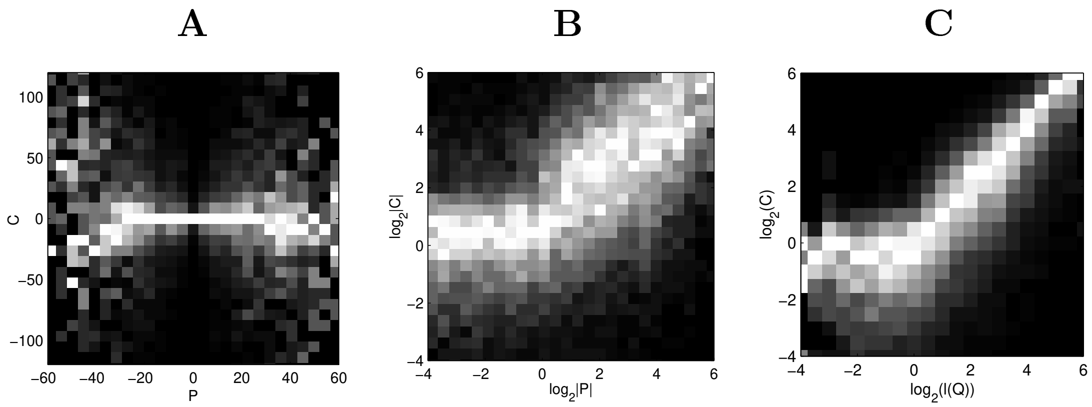


Figure 2.1: Conditional histograms for a fine scale horizontal coefficient. Brightness corresponds to probability, except that each column has been independently rescaled to fill the full range of display intensities. **A:** Conditioned on the parent (same location and orientation, coarser scale) coefficient. Data are for the Boats image. **B:** Same as **A**, but in the log domain. **C** Conditioned on a linear combination of neighboring coefficient magnitudes. Data are for the same subband of the Boats image as **A**.

The structure of the relationship between C and P becomes more apparent upon transforming the histogram into the log domain. Figure 2.1B shows the conditional histogram $\mathcal{H}(\log_2(C')|\log_2(Q))$, where $Q = |P|$ and $C' = |C|$. The right side of the distribution is unimodal and concentrated along a unit-slope line. This suggests that in this region, the conditional expectation, $\mathcal{E}(C'|Q)$, is approximately proportional to Q. Furthermore, vertical cross sections (i.e., conditional histogram for a fixed value of Q) have approximately the same shape for different values of Q. Finally, the left side of the distribution is concentrated about a horizontal line, suggesting that C' is independent of Q in this region. We suspect these low-amplitude coefficients are dominated by quantization and other noise sources.

The intuition for the right side of the distribution is that typical localized image structures such as edges tend to have substantial power across many scales. These structures will be represented in the wavelet domain via a superposition of basis functions at these scales. The signs and relative magnitudes of the coefficients associated with these basis functions will depend on the precise location, orientation and scale of the structure. But measurement of a large coefficient at one scale means that large coefficients at adjacent scales are more likely.

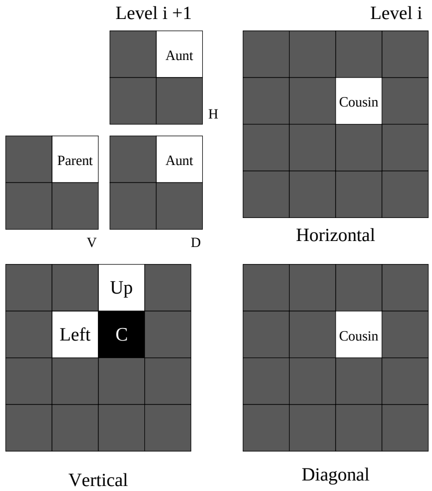


Figure 2.2: Subset of wavelet coefficients surrounding a given coefficient (C), that are potentially suitable for conditioning.

The form of the histograms shown in figure 2.1 is surprisingly robust across a wide range of images. Furthermore, the qualitative form of these statistical relationships also holds for pairs of coefficients at adjacent spatial locations (which we call "siblings"), adjacent orientations ("cousins"), and adjacent orientations at a coarser scale ("aunts"). This set of potential conditioning coefficients (we refer to these as "neighbors") is illustrated in figure 2.2.

2.2 Linear Magnitude Predictor

Given the piece-wise linear relationship between the magnitudes of large-amplitude coefficients, and the difficulty of characterizing the full multi-dimensional density, we chose to examine a linear predictor for coefficient magnitude:

$$l(\vec{Q}) \equiv \vec{w} \cdot \vec{Q} = \sum_k w_k Q_k, \tag{2.1}$$

where the coefficient magnitude set $\{Q_k\}$ corresponds to a subset of the potential conditioning neighbors, as depicted in figure 2.2. For a single subband, the weights w_k used to compute the predictor are chosen to minimize the expected squared error. That is:

$$\vec{w} = \mathcal{E}(\vec{Q}\vec{Q}^T)^{-1} \cdot \mathcal{E}(C' \cdot \vec{Q}), \tag{2.2}$$

where $\mathcal{E}(\cdot)$ indicates the expected value of a random variable, C' corresponds to the coefficient magnitude being estimated, and $\vec{Q}$ is a vector containing the magnitudes of the conditioning neighbors of C'. In practice, the expectation is computed by summing spatially over the subband (therefore assuming stationarity and ergodicity).

Figure 2.1C shows a conditional histogram, $\mathcal{H}\left(\log_2(C') | \log_2(l(\vec{Q}))\right)$ based on magnitudes of eight adjacent coefficients in the same subband, two cousin coefficients, and one parent coefficient (interpolated to the correct position using bilinear interpolation). Note that the distribution has a similar appearance to the single-parent distribution of figure 2.1A. But the linear region is extended, and the conditional variance is greatly reduced.

In order to determine which coefficients to include in the conditioning set $\{Q_k\}$, we calculated the mutual information between C and $l(\vec{Q})$ for a variety of choices of inter-band and intra-band coefficients in $\{Q_k\}$. The mutual information gives the theoretical coding gain (in bits per coefficient) obtained when encoding C using the conditional histogram $\mathcal{H}\left(C \mid l(\vec{Q})\right)$ (i.e., assuming $l(\vec{Q})$ is known to the receiver) compared with encoding C using only the marginal histogram $\mathcal{H}(C)$. Rather than exhaustively explore all possible neighbor subsets, we used a greedy algorithm. Specifically, the set is constructed incrementally: at each step, we incorporate the remaining neighbor whose inclusion maximizes the mutual information. Table 2.1 shows the greedy optimal neighbor subset for the three oriented subbands. Using this analysis, and imposing causality (assuming a standard scan-line ordering of the bits), we decided to include neighbors corresponding to the first four table columns when coding the horizontal and vertical bands, and the first five columns for the diagonal bands.

2.3 Conditional Probability Model

Based on the linear magnitude predictor described in the previous section, we wish to characterize the joint densities of the coefficients. We were surprised to observe that the conditional distribution in the log domain, when normalized for mean and variance, is highly consistent across subbands of an image, and even across a wide range of images. Figure 2.3 shows a comparison of these conditional distributions for four subbands from the "Boats" image, and also a comparison of a single band across four different images. We included only the right portion of each conditional histogram (i.e., the region in which

Child subband	Last neighbor included in predictor					
Horizontal	Left	Up	Parent	DiagCousin	LeftLeft	DiagAunt
	0.3322	0.4308	0.4635	0.4804	0.4903	0.4939
Vertical	Up	Left	Parent	DiagCousin	UpUp	DiagAunt
	0.3513	0.4356	0.4675	0.4865	0.4929	0.4987
Diagonal	Up	Left	Parent	Horiz Cousin	Vert Cousin	Left Left
	0.2175	0.2792	0.3134	0.3235	0.3294	0.3356

Table 2.1: Cumulative mutual information between coefficient magnitude C and a linear estimator $l(\vec{Q})$ composed of a subset of neighbors. Each entry gives the mutual information for a subset containing the neighbors indicated at the top of that column and all columns to the left. Notice that the local neighbors within the subband (Left and Up), the Parent, and the Cousins contribute most to the mutual information. Values are averaged over two scales (levels 1 and 2) of three training images (Lena, Boats, Baboon) using a 256 by 256 dimensional histogram (stationarity is assumed).

C' is proportional to $l(\vec{Q})$).

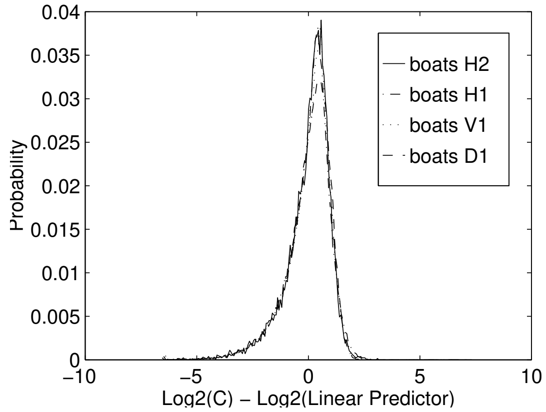
The fact that the conditional histograms seem to have a constant shape that shifts linearly with the predictor in the log domain suggests a model of multiplicative uncertainty. In particular, we use the following generative model for the conditional density:

$$C = M \cdot l(\vec{Q}) + N, \tag{2.3}$$

where $l(\vec{Q})$ is the linear magnitude predictor described previously, and M and N are two mutually independent zero-mean random variables. This implies that the variance of C is a linear function of $l^2(\vec{Q})$. We also assume explicitly that the coefficient is conditionally independent of the neighbors given the linear predictor: $\mathcal{P}\left(C \mid \{Q_n\}\right) = \mathcal{P}\left(C \mid l(\vec{Q})\right)$. This dependency only upon local neighbors implies that this model is a Markov Random Field as defined in section 1.2.2.

To model the distribution of M, we used a discretized lookup table obtained by averaging the mean- and variance-normalized conditional histograms (as shown in figure 2.1) of three training images (Lena, Boats, Baboon), at two scales (levels 2 and 3) and all three orientations. We assume N is independent of M, and Gaussian-distributed. Given these distributional assumptions, the model described by equation (2.3) is characterized by the linear weights $\{w_k\}$, and the variance N, σ^2.

In order to fit the model to a given set of data, the linear weights are chosen via equation (2.2) to be least-squares optimal. The variance is then computed by minimizing



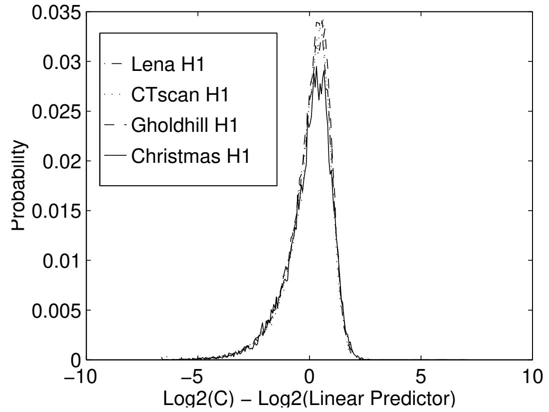


Figure 2.3: Comparison of conditional distributions in the log domain of different subbands and images. Distributions were normalized (in the log domain) to have mean 0 and variance 1. **Left:** Comparison of distributions for different subbands of the image 'boats'. **Right:** Comparison of distributions for different images (Lena, Goldhill, CTscan, Christmas).

the relative entropy between the joint model density and the joint histogram. Figure 2.4 shows comparisons of joint histograms of the second-level horizontal subband of four different images, with plots of the fitted density function generated by equation (2.3). The estimated densities are a reasonable fit, although several of the actual histograms show a narrowing of the conditional density for large predictor values. We believe this is due to small amounts of residual linear correlation between coefficients.

An entropy calculation shows the value and quality of the model. Figure 2.5 shows a scatterplot comparing encoding cost based on the joint probability model of equation (2.3) vs. the encoding cost assuming precise knowledge of a 256×256-bin histogram. Also included is a comparison to the first-order histogram entropies. The conditional model falls short of the ideal by less than 0.7 bit. In these situations, the conditional histogram (from which the ideal values in the table are computed) for large-magnitude predictors is sparse and has high variance. The predictive models, however, are *continuous* high-variance densities. Thus, the ideal values are deceptively low due to detailed knowledge of the coefficient values for this specific subband.[1] Nevertheless, the linear-predictive model is substantially better than the first-order model, consistent with the mutual entropy estimates of table 2.1.

Finally, we should consider the signs of the coefficients. As mentioned in the previous

[1]For example, at the extreme, a single sample taken from any joint distribution will yield a conditional histogram with a relative entropy of 0.

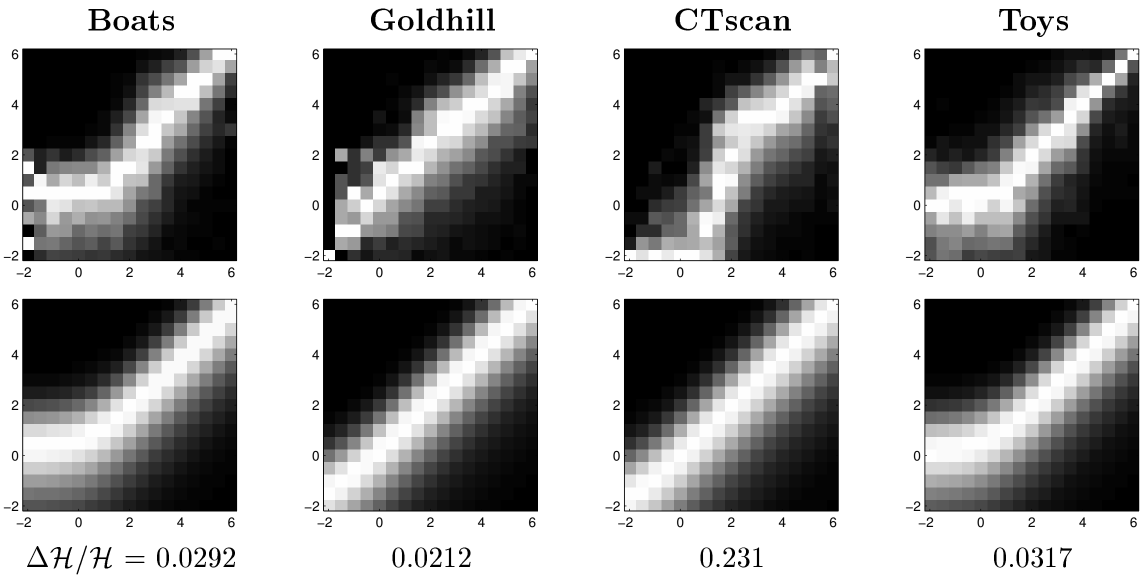


Figure 2.4: Top: Examples of log-domain conditional histograms for the second-level horizontal subband of different images, conditioned on an optimal linear combination of coefficient magnitudes from adjacent spatial positions, orientations, and scales. **Bottom**: Model of equation (2.3) fitted to the conditional histograms in the left column. Intensity corresponds to probability, except that each column has been independently rescaled to fill the full range of intensities. Below the model is the ratio of the relative conditional entropy $\Delta\mathcal{H}$ between the histogram and the model and the conditional histogram entropy $\mathcal{H}$

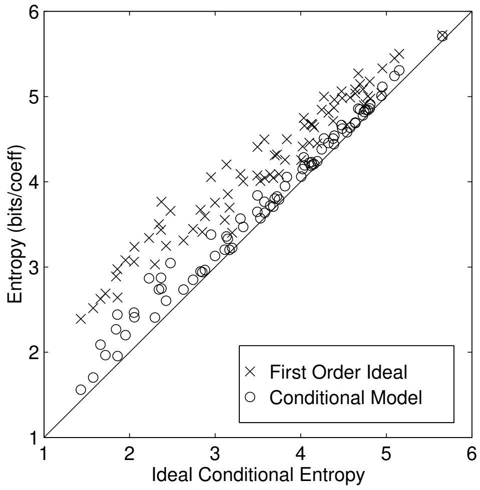


Figure 2.5: Comparison of encoding cost using the conditional probability model of equation (2.3), and the encoding cost using the first-order histogram as a function of the encoding cost using a 256×256-bin joint histogram. Points are plotted for 6 bands (2 scales, 3 orientations) of the 13 images in our sample set. The average relative entropy (K-L distance) of the ideal point statistics given the ideal joint statistics is 0.548 bits/coefficient, while the average relative entropy of the conditional model given the ideal joint statistics is 0.129 bits/coefficient.

section, we model the lowpass coefficient distribution as a uniform density. The values are almost entirely positive: Over our sample set of 13 images, only 2.4% of the coefficients are negative. For the bandpass subbands, the probability of positive and negative coefficients is equal. They are, however, not spatially independent. In the horizontal bands, for example, the probability of the "Up" neighbor having the same sign is 36%. We utilize this simple single-neighbor conditioning in our coder. There are also more complex relationships between sign bits in neighboring subbands, but we do not attempt to characterize those in this paper.

Chapter 3

Image Compression

In this chapter we describe the implementation of our novel compression algorithm, the Embedded Predictive Wavelet Image Coder (EPWIC), based on the conditional probability model developed in Section 2.3. Our implementation is simple and reasonably efficient, but comes quite close to the theoretical entropy associated with the probability model. In addition, it compares favorably with the current best coders in the image processing literature.

3.1 Separable Wavelet Decomposition

We utilize a recursive pyramid decomposition based on the separable 9/7 tap biorthogonal filter set of [ABMD92]. This set, which satisfies the one-dimensional system diagram shown in figure 3.1, is composed of four filters: the two analysis filters L_9, a 9 tap lowpass filter, and H_7, a 7 tap highpass filter, and the two synthesis filters L_7, a 7 tap lowpass filter, and H_9, a 9 tap highpass filter. The synthesis filters are related to the analysis filters via the expressions:

$$\begin{aligned} H_9(\omega) &= e^{j\omega} L_9(\omega + \pi) \\ H_7(\omega) &= e^{j\omega} L_7(\omega + \pi) \end{aligned}$$

The system diagram displays that these filters can split a one-dimensional signal into a lowpass and a highpass subband (using L_9 and H_7 respectively), each made half the size

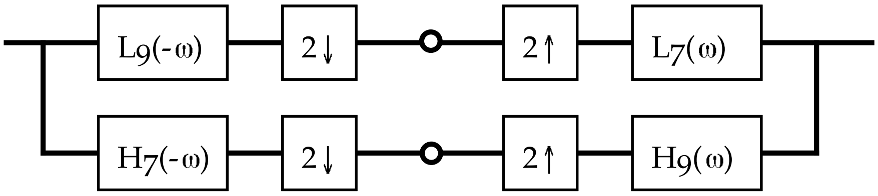


Figure 3.1: One-dimensional analysis/synthesis diagram for the dyadic biorthogonal wavelet decomposition used in EPWIC.

of the original signal by removing every other sample. In addition, the original signal can be reconstructed by inserting zeros at every other sample of the lowpass and highpass subband, applying time-reversed copies of L_7 and H_9 respectively, and summing the results. Intuitively, the lowpass subband contains the low frequencies of the original signal and therefore represents a smaller version of the signal while the highpass subband contains the fine detail of the signal. These filters have become quite popular in the compression literature because they are symmetric and have perfect reconstruction (to floating point errors).

In order to generate a single level of the wavelet pyramid decomposition of an image, the two analysis kernels, L_9 and H_7, are applied separably along the axes of the image sampling lattice. This results in lowpass, vertical, horizontal and diagonal subbands. Subsequent pyramid levels (i.e., subbands at different scales) are created by applying this 4-band splitting procedure recursively to the lowpass subband. Convolution boundaries are handled by symmetric reflection of the image about the edge pixels, as described in [Sim88].

We denote the basis functions in the separable wavelet transform as $w_s(x-n, y-m)$, where s indicates the subband (determined by the orientation and scale) and (n, m) indicates the spatial location of the basis function. These basis functions are normalized to have unity L_2-norm. The wavelet representation consists of the set of coefficients $\{c_s(n, m)\}$.

3.2 Coefficient Bitplane Encoding

In order to have maximal control over the ordering of image information, we map each subband coefficient to a 16-bit binary integer (including sign bit).[1] That is:

$$c_s(n,m) = a_s \rho_s(n,m) \sum_k 2^k b_s(n,m,k),$$

where a_s is a scalar multiplier for each subband (used to rescale the values for 16-bit integer representation), $\rho_s(n,m)$ is the sign (± 1), and $b_s(n,m,k)$ corresponds to the kth bit of the coefficient $c_s(n,m)$.

The wavelet decomposition describes the image $I(x,y)$ as a linear combination of the basis functions:

$$\begin{aligned} I(x,y) &= \sum_{s,n,m} a_s\ \rho_s(n,m)\ \sum_k 2^k\ b_s(n,m,k)\ w_s(x-n,y-m) \\ &= \sum_{s,n,m,k} b_s(n,m,k)\ \left[a_s \rho_s(n,m) 2^k w_s(x-n,y-m)\right]. \end{aligned}$$

The second expression suggests that we may view this as a type of representation in which the coefficients are restricted to the set $\{0,1\}$ (i.e., they are single-bit quantities). The basis functions of this representation are $w'_s(n,m,k) = a_s \rho_s(n,m) 2^k w_s(x-n,y-m)$, which are related to each other by translation, dilation, Fourier modulation, negation, and *scalar multiplication by powers of two.*

A goal of the progressive transmission of an image is to transmit coefficient bits in order of their significance so that the recipient may decode the best quality image possible with only the initial section of the transmission. Progressive transmission is also called "embedded encoding" since higher compressed versions of the image are therefore embedded into the bit stream. For our encoder, we define bit significance to be the ratio of the decrease in mean squared error represented by the bit divided by the encoding cost for the bit. Ideally, at each point we wish to select the bit which produces the most steeply descending rate-distortion curve. However, the overhead of perfectly ordering bits would be phenomenal since the location of each bit would need to be transmitted to the recipient.

In order to keep the complexity of the coder down, we assume that all bits at a given significance level k of a subband will be sent consecutively, in raster order. We refer to this

[1] A 16-bit coefficient representation was adequate for perfect reconstruction of a 5-level pyramid with our test images.

collection of bits as a *bitplane*. We also assume that the bitplanes of a given subband will be sent in order from most to least significant. Since most of the coefficient values are close to zero, the sign bit of each coefficient is sent only when needed, immediately after the first non-zero magnitude bit, as in [SZB95]. For example, if a coefficient has a quantized value of -14 (which in binary is -1110_2), since we are using a 15 bit representation for the magnitudes, the first 11 significant bits sent will be 0. When the 12th bitplane is sent, a 1 will be encoded to represent the most significant bit followed by a 1 to denote that the value is negative. The rest of the bits: 1, 1, and 0 will follow when the 13th, 14th and 15th bitplanes are encoded.

In general, the ordering of bitplanes across subbands should take into account both the encoded size of the bitplane and the improvement in decoded image quality resulting from the incorporation of that bitplane. We use a greedy algorithm (which we refer to as a "bang-for-the-buck" algorithm), in which we select at each step the remaining bitplane with the maximum ratio of MSE reduction per encoded bit. In other words, we choose the bitplane which produces the most steeply descending rate-distortion curve. Wang and Kuo [WK97] use a similar technique designed specifically for successive approximation quantization, in which they related the steepest rate-distortion curve with the highest quantization threshold. Our "bang-for-the-buck" algorithm is a generalization of this concept.

The individual magnitude bits are encoded with a static arithmetic encoder whose probabilities are determined directly from our image model. The arithmetic encoder is similar to the algorithm found in [PTVF92], which encodes a data stream using a probability distribution that is adaptively computed and stored in a histogram. Instead of computing such a histogram, our encoder uses the distribution specified by our statistical model. Since the "symbols" of our input stream are single bits, the probability that a bit is non-zero is all that is needed to construct the arithmetic code.

3.3 Calculation of Bit Probabilities

Our encoding technique makes direct use of the model joint probability density described earlier. In particular, both encoder and receiver must use this distribution to compute the conditional mean estimate for each coefficient, given the bits that have been sent/received thus far. In addition, as described above, the arithmetic coder and decoder must know the probability that a given bit will be non-zero.

Consider the arithmetic encoding of the kth bit of a particular subband coefficient, C_k. The encoder must calculate the probability that the bit is nonzero, given the set of all coefficient bits that have already been received. The set of bits received constrain the magnitude of the coefficient of interest (C'), and the magnitudes of each of the conditioning coefficients $\{Q_n\}$ to lie in particular ranges:

$$\begin{aligned} C' &\in [l_0, h_0] \\ Q_n &\in [l_n, h_n], \qquad n = 1, 2, \ldots, N. \end{aligned} \tag{3.1}$$

The probability that we wish to calculate is:

$$\begin{aligned} &\mathcal{P}\left(C_k = 1 \,|\, \text{bits received thus far}\right) \\ &\qquad = \mathcal{P}\left(m_0 < C' < h_0 \,|\, l_0 < C' < h_0, l_n < Q_n < h_n,\ \forall n\right) \\ &\qquad = \frac{\mathcal{P}\left(m_0 < C' < h_0 \,|\, l_n < Q_n < h_n,\ \forall n\right)}{\mathcal{P}\left(l_0 < C' < h_0 \,|\, l_n < Q_n < h_n,\ \forall n\right)} \\ &\qquad = \frac{\int dp\, \mathcal{P}\left(m_0 < C' < h_0 \,|\, l(\vec{Q}) = p\right)\, \mathcal{P}\left(l(\vec{Q}) = p \,|\, l_n < Q_n < h_n,\ \forall n\right)}{\int dp\, \mathcal{P}\left(l_0 < C' < h_0 \,|\, l(\vec{Q}) = p\right)\, \mathcal{P}\left(l(\vec{Q}) = p \,|\, l_n < Q_n < h_n,\ \forall n\right)}, \end{aligned} \tag{3.2}$$

where $m_0 = (l_0 + h_0)/2$. The last expression uses the assumption of our joint probability model: given $l(\vec{Q})$, the individual components of $\vec{Q}$ do not provide additional information about C':

$$\mathcal{P}\left(l_0 < C' < h_0 \,|\, l(\vec{Q}) = p, l_n < Q_n < h_n,\ \forall n\right) \approx \mathcal{P}\left(l_0 < C' < h_0 \,|\, l(\vec{Q}) = p\right).$$

In order to avoid the computationally expensive integration over p, we use two approximations in the conditional probability of C' that allow us to perform a simple one-dimensional calculation. First, we assume that the density of $log_2(C')$ given the magnitude predictor $l(\vec{Q})$ (see equation (2.3)) when normalized for mean and variance has a specific functional shape independent of the value of $l(\vec{Q})$, the subband, and the image (see figure 2.3). In this case, we may divide C' by its standard deviation, $\sqrt{l^2(\vec{Q}) + \sigma^2}$, to get a scalar random variable that is independent of $l(\vec{Q})$. We compute a lookup table, $G(\cdot)$, containing the average of the mean- and variance-normalized log-domain cumulative histograms of this quantity for three training images (Lena, Boats, Baboon) at two scales (levels 2 and 3) and all three orientations. We can then use $G(\cdot)$ as a parameterized model for the conditional cumulative:

$$\mathcal{P}\left(C' < c \,|\, l(\vec{Q}) = p\right) \approx G\left(\frac{\log_2(c/\sqrt{p^2+\sigma^2})}{\alpha}\right), \tag{3.3}$$

where σ^2 is the variance of N, and α^2 is the second moment of $\log_2(|M|)$.

Second, we eliminate the need for integration by replacing the integration variable p by its conditional mean. Specifically, we rewrite the model of equation (2.3) in terms of the current estimate of the magnitude predictor ($\hat{p}$), and the error of that estimate (e_p):

$$C = M \cdot (\hat{p} + e_p) + N.$$

Assuming e_p is independent of M and N, the variance of this expression is $\hat{p}^2 + \sigma_{\hat{p}}^2 + \sigma^2$, where $\sigma_{\hat{p}}^2$ is the variance of the error between p and $\hat{p}$, and the integral becomes:

$$\begin{aligned}
&\int dp\, \mathcal{P}\left(m_0 < C' < h_0 | l(\vec{Q}) = p\right) W(p) \\
&\quad = \int dp \left[\mathcal{P}\left(C' < h_0 | l(\vec{Q}) = p\right) - \mathcal{P}\left(C' < m_0 | l(\vec{Q}) = p\right)\right] W(p) \\
&\quad \approx \left[G\left(\tfrac{\log_2(h_0) - \log_2(\sqrt{\hat{p}^2+\sigma_{\hat{p}}^2+\sigma^2})}{\alpha}\right) - G\left(\tfrac{\log_2(m_0) - \log_2(\sqrt{\hat{p}^2+\sigma_{\hat{p}}^2+\sigma^2})}{\alpha}\right)\right] \int dp\, W(p),
\end{aligned}$$

where

$$W(p) = \mathcal{P}\left(l(\vec{Q}) = p \,|\, l_n < Q_n < h_n,\ \forall n\right).$$

The current estimate of the predictor $\hat{p}$ is defined from the current estimates of the neighbor magnitudes $\hat{c}_n$ (calculated in equation (3.5)):

$$\hat{p} \;=\; \sum_n w_n \hat{c}_n,$$

and the estimate error $\sigma_{\hat{p}}^2$ is defined by the magnitude error estimates $\sigma_{\hat{c}_n}^2$ (calculated in equation (3.6)), assuming the neighbors' error estimates are uncorrelated:

$$\sigma_{\hat{p}}^2 \;=\; \sum_n w_n^2 \sigma_{\hat{c}_n}^2$$

Finally, substituting this approximation into equation (3.3), and eliminating common factors gives:

$$\mathcal{P}\left(C_k = 1 | \text{bits received thus far}\right)$$
$$\approx \frac{G\left(\frac{\log_2(h_0)-\log_2(\sqrt{\hat{p}^2+\sigma_{\hat{p}}^2+\sigma^2})}{\alpha}\right) - G\left(\frac{\log_2(m_0)-\log_2(\sqrt{\hat{p}^2+\sigma_{\hat{p}}^2+\sigma^2})}{\alpha}\right)}{G\left(\frac{\log_2(h_0)-\log_2(\sqrt{\hat{p}^2+\sigma_{\hat{p}}^2+\sigma^2})}{\alpha}\right) - G\left(\frac{\log_2(l_0)-\log_2(\sqrt{\hat{p}^2+\sigma_{\hat{p}}^2+\sigma^2})}{\alpha}\right)}. \quad (3.4)$$

This is the expression used for the calculation of bit probabilities in the coder.

After a bit is sent or received for a coefficient magnitude, the conditional mean estimate $\hat{c}$ given its new range (l_1, h_1) is calculated from the joint probability density. As before, we avoid computationally expensive integration over p through the use of its estimate $\hat{p}$:

$$\hat{c} = \int_{l_1}^{h_1} dc\ \mathcal{P}\left(C' = c \,|\, l(\vec{Q}) = \hat{p}\right)\ c. \quad (3.5)$$

Similarly, the variance of the error between $\hat{c}$ and the actual magnitude is:

$$\sigma_{\hat{c}}{}^2 = \int_{l_1}^{h_1} dc\ \mathcal{P}\left(C' = c \,|\, l(\vec{Q}) = \hat{p}\right)\ (\hat{c} - c)^2. \quad (3.6)$$

In the coder, the integrals in equations (3.5) and (3.6) are approximated by summation over a uniform partitioning of the range (l_1, h_1) and calculating the probability of each bin through the cumulative density given in equation (3.3). The effects of accumulated errors in these estimates are analyzed in section 3.5.

3.4 Summary and Complexity Analysis of EPWIC

To encode an image $I(x, y)$:

1. Choose a terminal MSE (we typically use the variance associated with the quantization of the original image pixels).

2. Calculate the coefficients $c_s(n, m)$ of the wavelet decomposition.

3. For each subband in the decomposition, quantize coefficients (to 16 bits) and retain the quantization binsize, $1/a_s$.

4. Characterize the statistics for each subband:

(a) Calculate least-squares optimal weights $\vec{w}$ of the linear predictor $l(\vec{Q})$ (equation (2.2)).

(b) Calculate σ^2 (the variance of N) and α^2 (the variance of M) that minimize the relative entropy between the joint coefficient histogram and the probability density described by equation (2.3).

5. Transmit an EPWIC identification tag (16 bits), the number of levels in the pyramid (3 bits), and width and height of the image (5 bits each), and σ^2 (8 bits, representing $[2^{-5}, 2^5]$).

6. While the decoded MSE is greater than the goal MSE:

(a) Determine which of the set of candidate bitplanes (i.e., the most significant remaining bitplane of each subband) should be encoded next, by comparing the "bang-for-the-buck" (MSE reduction / encoding size). For each candidate bitplane:

- Compute the conditional means of all coefficients in the subband assuming the bitplane is sent (using equation (3.5)). Compute the reduction in MSE that results when the image is reconstructed from the resulting wavelet pyramid.
- For each bit in the bitplane, calculate the probability of the bit being nonzero (using equation (3.4)), and construct a code stream using the arithmetic coder.

(b) Update the conditional variances of the subband to be transmitted (using equation (3.6)).

(c) Transmit a tag identifying the subband to which the bitplane belongs ($\log_2$(number of subbands) bits).

(d) If this is the first encoded bitplane of this subband, transmit:

- The quantization binsize a_s (7 bits, representing the interval $[2^{-15}, 2^{16}]$),
- $\{w_k\}$ (8 bits each, representing $[-0.1, 1.1]$).
- and α^2 (8 bits, representing $[2.7, 4]$).

(e) Transmit the encoded data.

The memory overhead of the algorithm is quite reasonable. The wavelet pyramid requires as much space as the original image, as do the current estimates of coefficient conditional mean and variance. In addition, a look-up table is used for the conditional cumulative function $G(.)$ (153 floating point numbers).

For our implementation of EPWIC running in MatLab on a 300 megahertz Pentium workstation, encoding a 512×512 image to 64Kbytes using a 5-level pyramid takes approximately 5 minutes. Roughly 7 seconds of this time is used to estimate the parameters, 16 seconds for arithmetic encoding of the bitplanes, and 4.5 minutes are used to calculate the "bang-for-the-buck" for the next candidate bitplanes. In order to accommodate the testing of non-orthonormal filters, the MSE was calculated by reconstructing the pyramid twice, with and without the candidate bitplane. Thus, the calculation cost could be significantly reduced through use of orthonormal filters, for which the MSE improvement can be estimated directly from the coefficients. Decoding the image takes approximately 16 seconds, 8 of which are used in the calculation of the magnitude estimators and the conditional probabilities.

3.5 Results

In order to demonstrate the performance of EPWIC, we encoded the set of 13 images shown in figure 3.2. Each image was decomposed into a discrete wavelet pyramid containing 5 scales. For comparison purposes, we considered four other image coders:

1. EPWIC-1: we implemented a progressive encoder utilizing the marginal (generalized Laplacian) density of equation (1.3) as a model of the first order distribution. The coder is otherwise similar to the conditional implementation (EPWIC-2), in that it uses the same greedy algorithm for ordering of bitplanes, and uses the same arithmetic coding scheme.

2. JPEG: we used version 5b of CJPEG, a standard non-progressive JPEG image coder from the Independent JPEG Group.

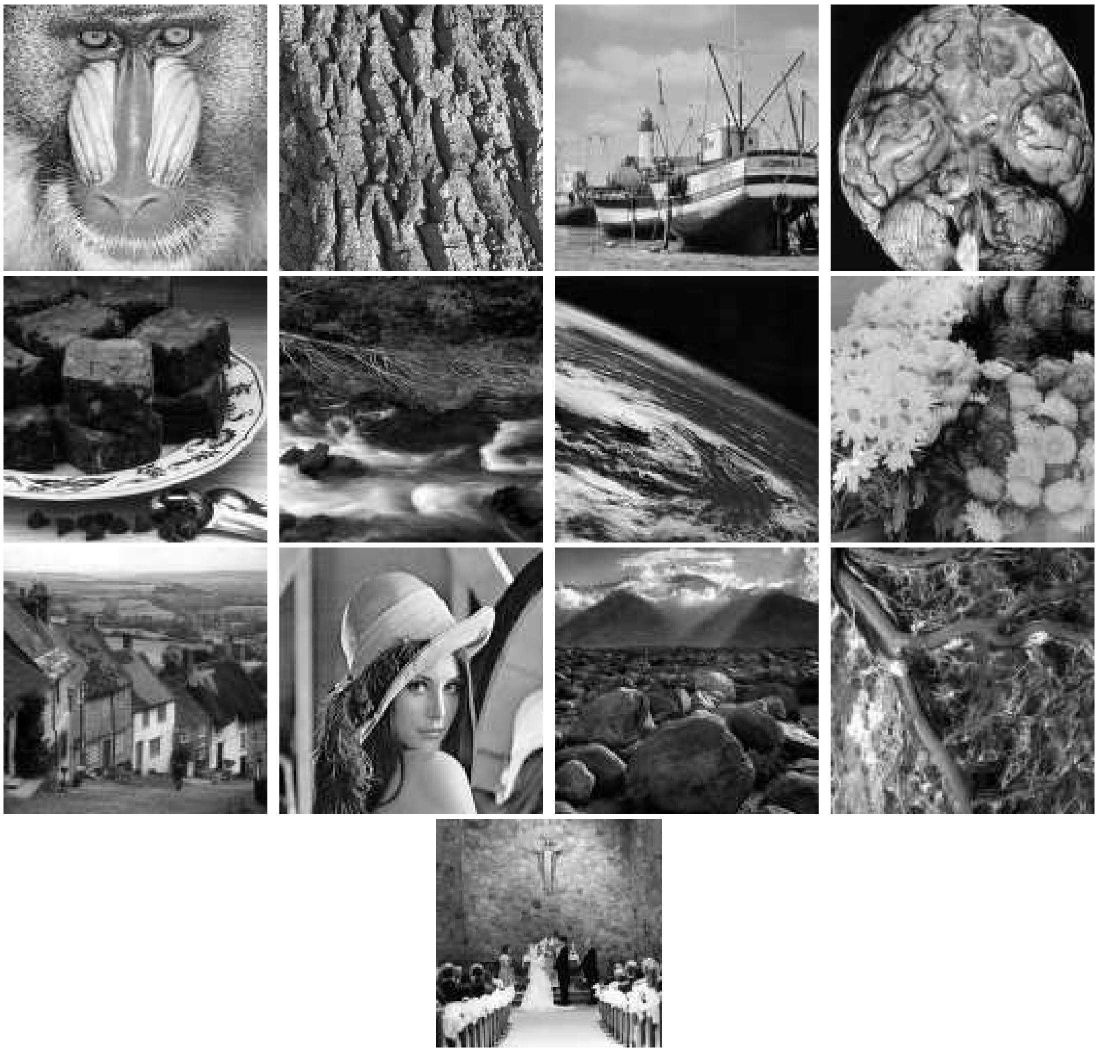


Figure 3.2: Full set of images used in our experiments. Left to right, top to bottom: Baboon, Bark, Boats, Brain, Brownie, Cantrell, Earth, Flowers, Goldhill, Lena, MTWill, Vein, and Wedding. Images are 8-bit/pixel, and of size 512×512.

3. EZW[2]: as described in [Sha93].

4. SPIHT: as described in [SP96].

Table 3.1 lists the PSNR values for EPWIC-2, EPWIC-1, EZW, SPIHT, and JPEG for five of the images. It should be noted that these PSNR values were calculated directly from the decompressed images, and the bitrates indicate actual encoding sizes, not entropy estimates. We were surprised to find that EPWIC-1 surpasses EZW for most images, since the model for this coder incorporates no joint statistical information, while EZW

[2]We thank the David Sarnoff Research Center for their assistance in the EZW comparisons.

Image	Coder	Bits/Pixel								
		0.008	0.0016	0.031	0.063	0.125	0.25	0.5	1.0	2.0
Boats	EPWIC-2	21.67	23.36	25.21	27.24	29.72	32.97	37.04	41.69	46.96
	SPIHT	22.28	23.75	25.46	27.46	29.85	33.07	37.16	41.73	46.81
	EPWIC-1	21.55	23.30	24.98	26.92	29.09	31.95	35.80	39.97	45.11
	EZW	21.34	22.83	24.81	26.86	28.87	31.69	35.58	40.00	45.82
	JPEG	NA	NA	18.29	21.83	27.76	30.88	34.63	39.10	43.54
Goldhill	EPWIC-2	22.00	23.54	24.93	26.59	28.23	30.08	32.83	36.25	41.78
	SPIHT	22.56	23.89	25.26	26.70	28.39	30.45	32.99	36.44	41.99
	EPWIC-1	22.12	23.63	24.95	26.52	28.07	29.92	32.38	35.73	40.98
	EZW	21.66	23.41	24.66	26.10	27.88	29.73	31.92	35.09	39.82
	JPEG	NA	NA	17.92	23.96	26.85	29.18	31.59	34.46	38.46
Lena	EPWIC-2	21.68	23.54	25.70	28.03	30.85	33.78	37.15	40.34	45.06
	SPIHT	22.07	23.95	25.95	28.36	31.10	34.12	37.23	40.43	45.11
	EPWIC-1	21.26	23.35	25.41	27.61	30.18	33.04	35.99	39.27	44.04
	EZW	21.03	22.99	25.01	27.46	30.26	33.32	36.46	39.79	44.64
	JPEG	NA	NA	17.95	21.92	28.24	31.42	34.84	37.95	41.62
Flowers	EPWIC-2	21.12	22.66	24.10	25.68	27.66	29.91	32.77	36.84	42.77
	SPIHT	21.68	23.01	24.42	25.95	27.88	30.25	33.23	37.22	43.00
	EPWIC-1	21.43	22.96	24.34	25.90	27.75	29.92	32.58	36.16	41.56
	EZW	20.96	22.45	23.93	25.66	27.16	29.29	32.16	35.89	40.97
	JPEG	NA	NA	16.89	22.04	25.89	28.86	31.51	34.54	38.80
Wedding	EPWIC-2	19.77	21.46	23.36	25.50	27.85	30.35	33.75	37.81	44.00
	SPIHT	20.42	21.88	23.86	25.83	28.10	30.70	33.86	38.04	44.07
	EPWIC-1	19.50	21.49	23.30	25.44	27.67	30.32	33.34	37.21	43.16
	EZW	19.34	21.01	23.08	25.35	27.27	29.68	32.65	36.41	42.10
	JPEG	NA	NA	17.07	21.92	26.19	29.03	32.11	35.66	40.48

Table 3.1: PSNR values $10 \cdot \log_{10}(255^2/\text{MSE})$ at different compression ratios for EPWIC-2, EPWIC-1, EZW, SPIHT, and JPEG. Images are shown in figure 3.2.

exploits some of the joint conditional relationships between coefficients at different scales. EPWIC-2 surpasses EZW at nearly all compression levels, and approaches the encoding capability of SPIHT at the higher compression rates. Figure 3.3 summarizes these results, by showing the PSNR of each coder (relative to EZW), averaged over the 13 images in our set. EPWIC-1 outperforms EZW for most compression ratios by about 0.3dB, and EPWIC-2 outperforms EZW by 0.5dB at 1 Kbyte, and nearly 1.5dB at 16 Kbytes and above.

Also shown in figure 3.3 is the encoding size (relative to that of EZW) as a function of target SNR. This gives a sense of how long one would wait during progressive transmission for a result of a given quality. For example, EZW would have a transmission time roughly 15% higher than EPWIC-2 for an image quality of 26dB.

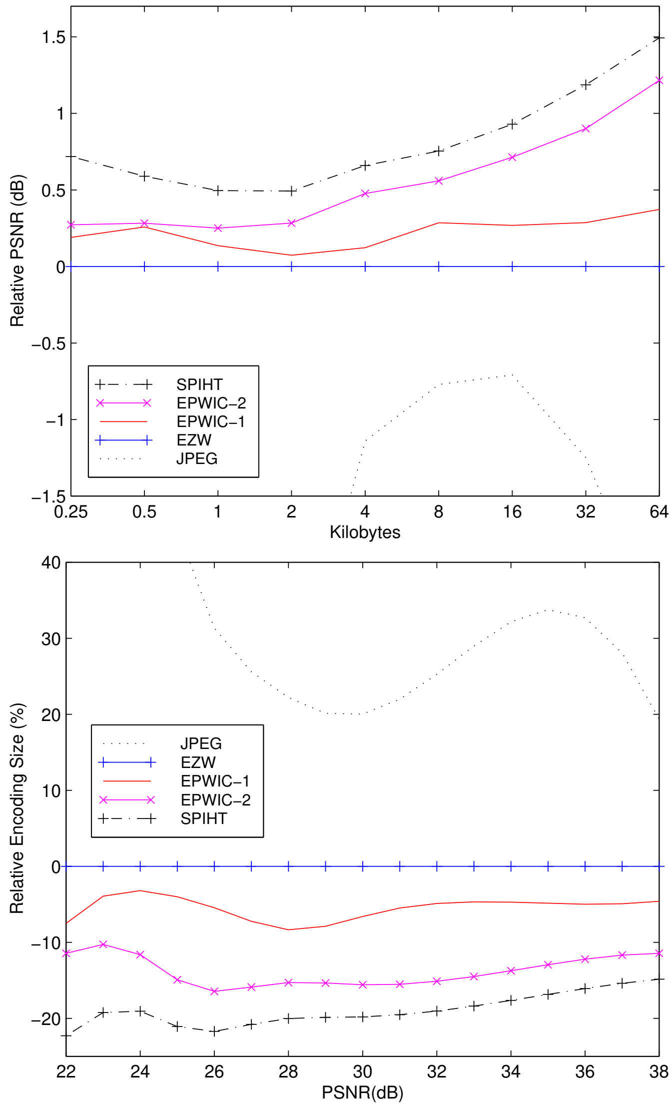


Figure 3.3: Relative rate-distortion tradeoff for five image coders (JPEG, EZW, SPIHT, EPWIC-1, and EPWIC-2). **Top**: PSNR values (in dB), relative to EZW (horizontal line), as a function of the number of encoded bytes. **Bottom**: Number of bytes necessary to achieve a given PSNR, relative to EZW (horizontal line). All curves are averages over the set of 13 images in our collection (shown in figure 3.2).

Figure 3.4: A progressive series of the Boats image, encoded using EPWIC-2. Data transmission amounts are given above each image.

In figure 3.4, an EPWIC-2 progressive transmission series is given for the Boats image. Wavelet aliasing artifacts are quite noticeable in the early stages of the transmission: these are a consequence of using a critically sampled subband representation. At 16 Kbytes (compared with an original image size of 256 Kbytes), the reconstructed image is remarkably close to the original. We find that EPWIC-2 compressed images are visually indistinguishable from SPIHT compressed images at the same bitrate.

One would to know how much encoding performance is being lost in the integration approximations of equation (3.4), and how much is due to the overhead of sending the model parameters. In order to understand the cost of the approximations, we developed a non-progressive version of the coder called "EPWIC-2 NP". Instead of encoding the pyramid bitplane by bitplane, EPWIC-2 NP encodes the subbands simultaneously, at a fixed quantization level. In order to understand the cost of the model parameters, the

PSNR for EPWIC-2 NP were re-calculated for image sizes not including the overhead of encoding the parameters.

Figure 3.5 shows the results of this analysis. The non-progressive coder gives an improvement of roughly 0.1 dB over progressive EPWIC-2. Thus, we conclude that the integration approximation in EPWIC-2 does not greatly penalize the progressive encoding process. Without the overhead of the model parameters (which at low bitrates result in as much as 0.4 dB improvement) EPWIC-2 NP becomes comparable to SPIHT for encoding sizes up to 2K. As the encoding size increases the relative overhead of the model parameters diminishes.

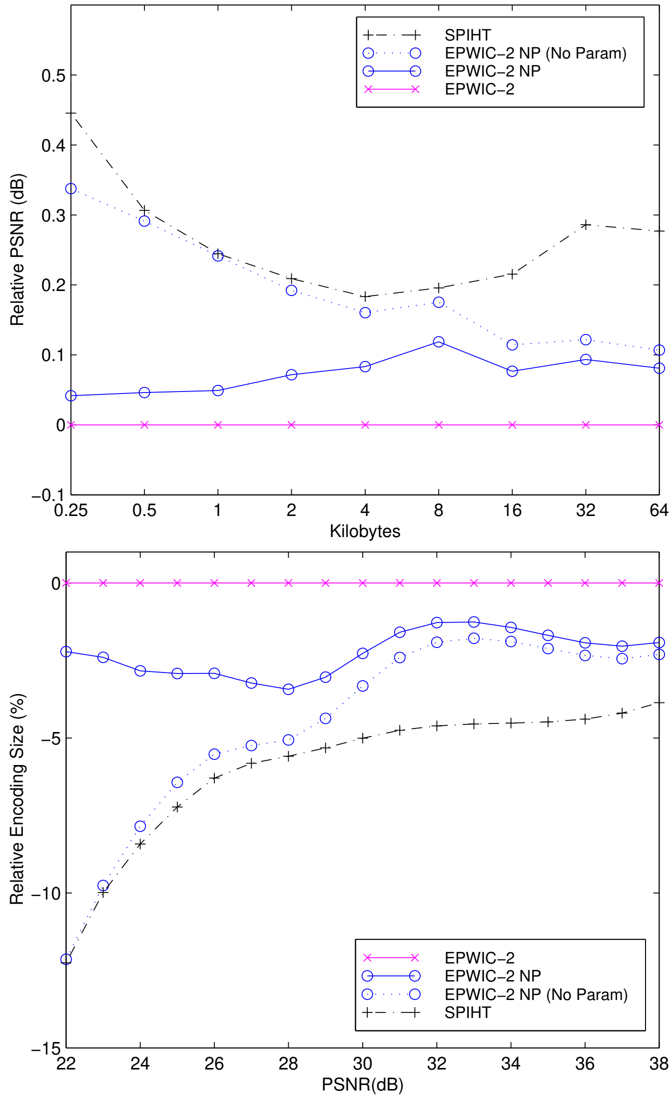


Figure 3.5: Overhead of model parameters and integration estimation. EPWIC-2 NP is a non-progressive version of EPWIC-2. Also shown in these graphs are the PSNR values excluding the overhead of encoding the model parameters for EPWIC-2 NP. **Left**: PSNR values (in dB), relative to EPWIC-2 (horizontal line), as a function of the number of encoded bytes. **Right**: Number of bytes necessary to achieve a given PSNR, relative to EPWIC-2 (horizontal line). All curves are averages over the set of 13 images in our collection (shown in figure 3.2).

Chapter 4

Texture Segmentation

From a simplification of the physical model in section 1.1, an image can be described as a superposition of many **textured** objects. Therefore, a potential first step in the problem of image interpretation and understanding can be the segmentation of an image by texture. The general task of "blind texture segmentation" classifies homogeneous regions of images without prior knowledge of the textures that are present.

Unfortunately, the definition of a **texture** is as elusive as the definition of an image itself. Over time, characterizations of textures have converged to a "statistical" definition, in which textures are viewed as being sampled from a probability density. This seems to contrast with a traditional view of textures as "structural," or repeating patterns of microtextures [Har79]. As can be seen in figure 4.1, textures can incorporate both qualities: microtextures randomly modified in terms of orientation, location, and intensity. However, it has been shown that structural elements can be reproduced purely with statistical methods [ZWM96].

In section 4.1, we review and uncover new statistical qualities of textures which may be useful as features in segmentation. Section 4.3 presents a novel variation of LDA, called "unsupervised LDA" (ULDA), which can be used to project features onto a subspace manageable by segmentation algorithms. In section 4.4, we briefly describe the K-means and EM algorithms used to segment the feature sets. Finally, segmentation results displaying the improvement in segmentation with ULDA and the new texture features are presented in section 4.6.

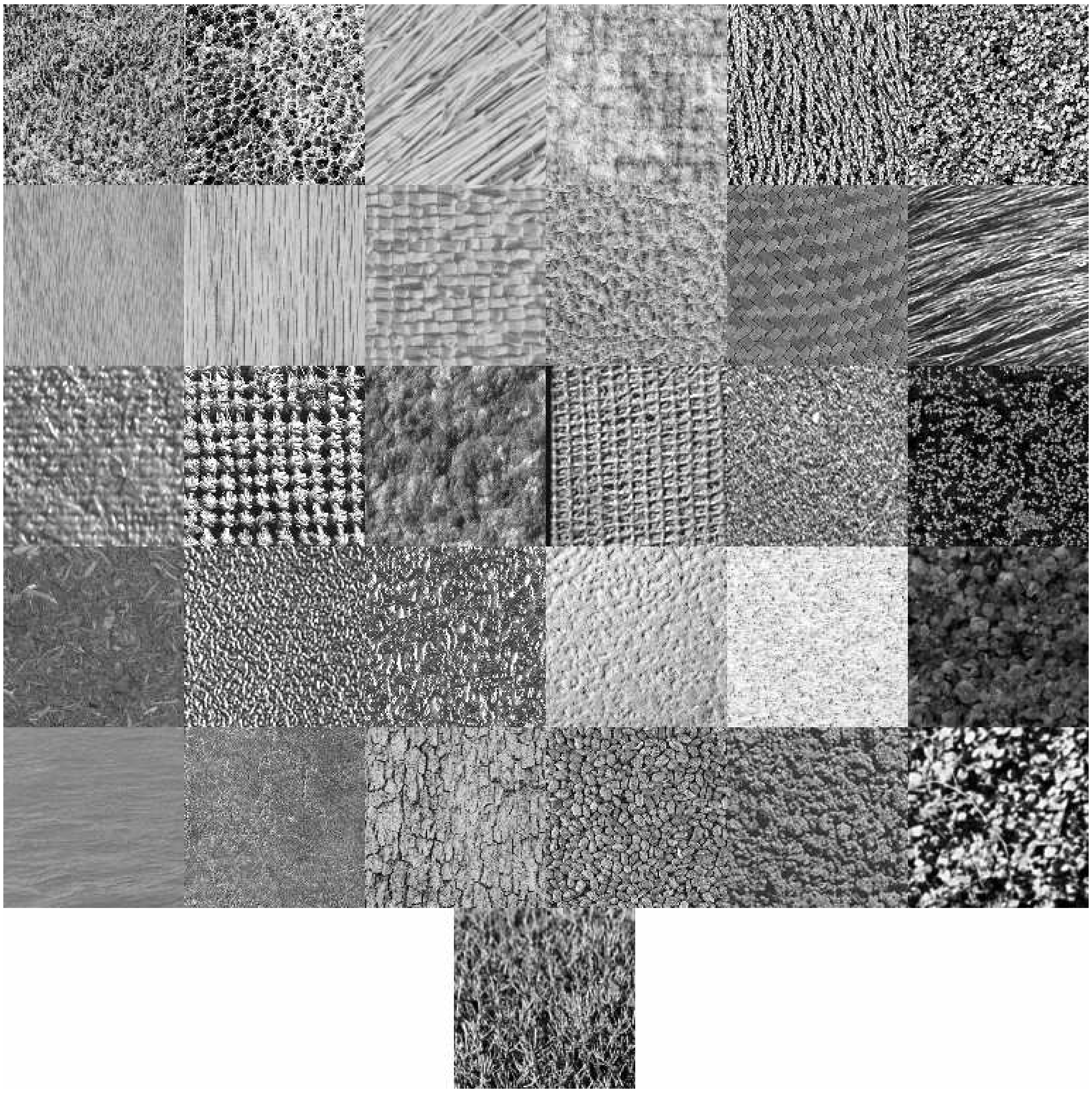


Figure 4.1: Set of textures used for segmentation testing [Bro66]. The textures were normalized for mean and variance in pixel intensity before being used within the texture mixtures.

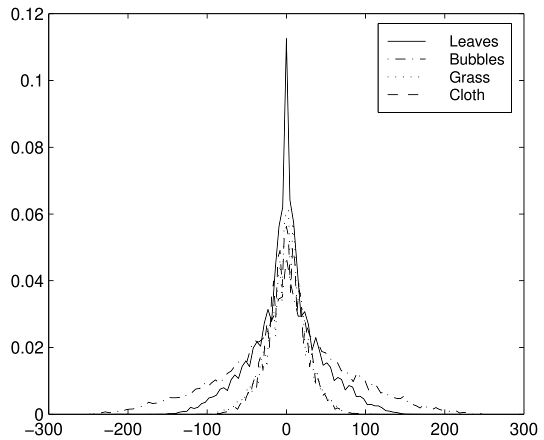


Figure 4.2: Correspondence of local pixel intensities in textures. Shown are the normalized histograms of the difference of a pixel and its left neighbor for four different textures.

4.1 Statistical Qualities of Textures

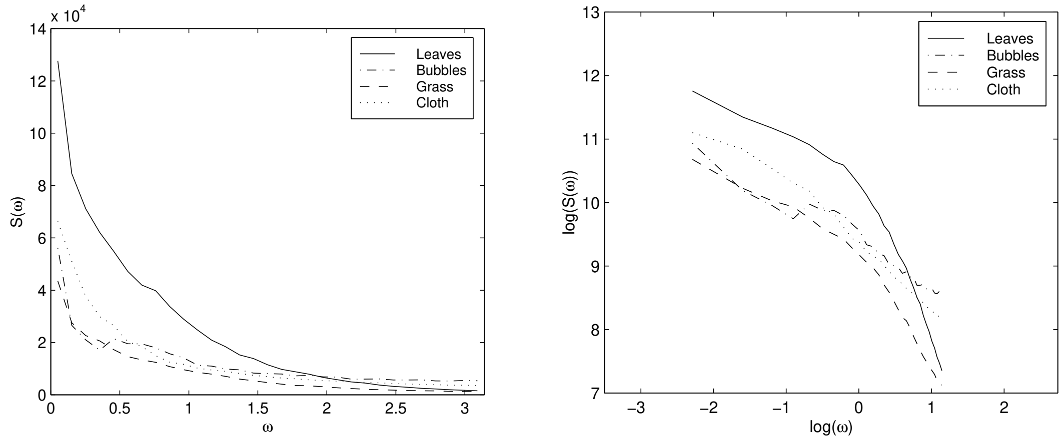
Similar to natural images, textures typically have local correspondence between neighboring pixel intensities. Figure 4.2 depicts the normalized histograms of the difference between the intensities a pixel and its left neighbor for four different textures. The distributions are peaked at zero, displaying a strong agreement between local values.

Beginning with the pioneering work of Julesz, Markov processes have been used to capture the periodicity and correlation between local texture pixels [Jul75, HS80, CJ83]. Furthermore, Julesz hypothesized that there exists some N for which Nth order joint empirical densities of neighbors can be used to partition textures into classes that are indistinguishable to a human.

4.1.1 Oriented Energy of Textures

Though not truly periodic, most textures have an almost periodic formation of similar microtextures. Due to this semi-periodic structure, dominant frequencies and orientations are formed, causing the squared magnitudes of the Discrete Fourier Transform to deviate from the $1/\omega^2$ relationship typical for natural images (see section 1.2.1). Figure 4.3 portrays the relationship between $S(\omega)$ (the energy averaged over all orientations) and ω for four textures. In the log domain, they result in distinct curves, suggesting that frequency energy may be a quality that can be used to differentiate textures.

When the energy is averaged over a region of orientations, differences between textures



(a) **(b)**

Figure 4.3: Power spectrum of textures. **(a)** The squared magnitude of the Discrete Fourier Transform vs. the frequency. **(b)** The same plot as (a) in the log/log domain. Due to periodicity in textures, these are not straight lines with -2 slope like typical natural images (see figure 1.4).

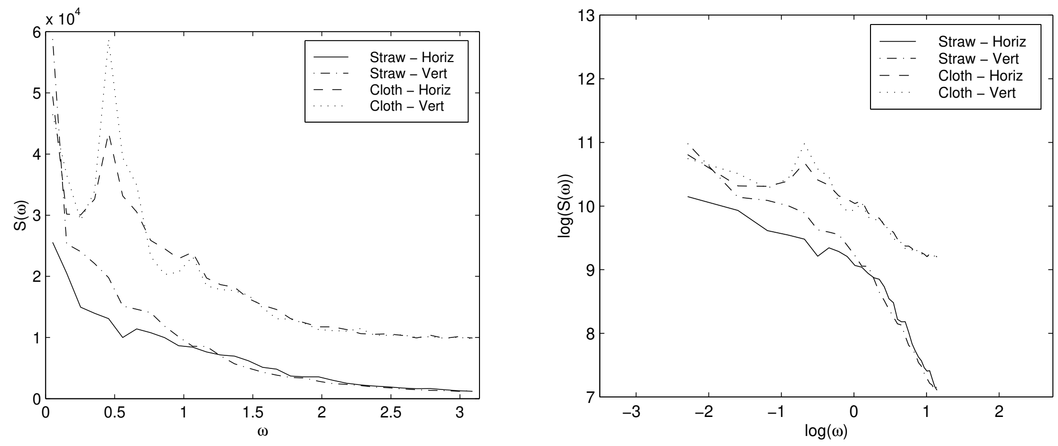
are further accentuated. In figure 4.4, $S(\omega)$ is averaged about the horizontal and vertical orientations for two textures. Differences in oriented energy suggest its potential use as a feature to discriminate between textures. Indeed, numerous researchers have implemented texture classification and segmentation techniques based upon the energy or variance of spatial-frequency decompositions such as Gabor functions [BCG90, TPH95, HPB98], difference of offset Gaussians [MP90][1], wavelet pyramids [Uns95], and steerable pyramids[2] [GBGP94].

4.1.2 Inter-Subband Pixel Correspondence

For natural images, subband coefficients in a wavelet pyramid decomposition are generally decorrelated (see section 2.1). However, for over-complete representations like the steerable pyramid, the coefficients are restricted to a linear subspace. In addition, semi-periodic features (peaks in the frequency domain) in textures may cause strong correlations between signed coefficients. Figure 4.5(a) shows the conditional histogram of a coefficient given its local down neighbor for a horizontal subband in the decomposition of a cloth texture.

[1] To model the outputs of V1 simple cells, Malik and Perona use half-wave rectification followed by local inhibition upon the local filter responses.

[2] Please see appendix section A.3 for more information about steerable pyramids.



(a) **(b)**

Figure 4.4: Oriented energy of textures. **(a)** The squared magnitude of the Discrete Fourier Transform for two orientations vs. the frequency. **(b)** The same plot as (a) in the log/log domain. For textures, different orientations may have significantly different amounts of energy.

There is a clear conditional relationship between the signed coefficients. The existence and strength of these local correlations have led researchers to use auto-correlation of subbands to categorize and generate textures [BCG90, RW90, BS94].

Similar to natural images (see section 2.1), inter-subband and intra-subband magnitude correlations may exist within a pyramid decomposition of textures. Since a steerable pyramid linearly projects an image onto a higher dimensional space, there will be some correlation between magnitudes. However, particular textures exibit an unusually strong correlation between magnitudes within and among subbands due to the almost periodic repetition of microtextures. Figure 4.5(b) displays a histogram for a horizontal subband magnitude given a diagonal subband magnitude for the decomposition of a cloth texture. This histogram do not follow the multiplicative model of equation (2.1), but does suggest that auto-correlations of subband magnitudes and intra-subband correlations of magnitudes may be used to differentiate textures. Portilla and Simoncelli have used these correlations very successfully in texture synthesis [PS99].

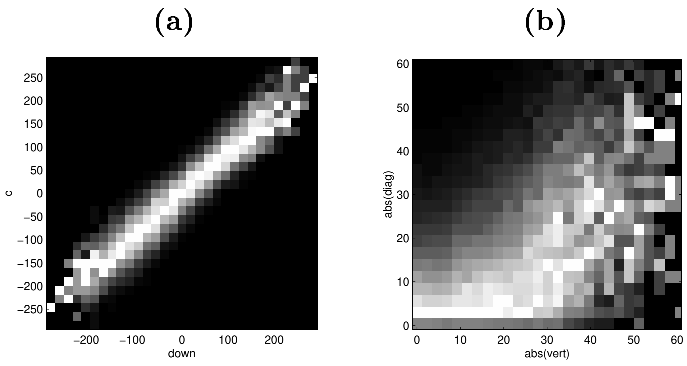


Figure 4.5: Conditional histograms of texture decompositions. Brightness corresponds to probability, except that each column has been independently rescaled to fill the full range of display intensities. **(a):** Coefficient conditioned on a down neighbor for a horizontal subband of a cloth texture. **(b):** Horizontal magnitudes conditioned on magnitudes in the same location but in a diagonal subband.

4.2 Dimensionality of Feature Vectors

The majority of texture segmentation algorithms operate on *feature vectors* associated with each pixel. Typical features are pixel intensity [SBLM98], color, and localized spatial-frequency filter responses (as described in section 4.1). Generally, since a given set of features may be inadequate to distinguish specific classes, the discriminating power of a feature vector grows with the dimensionality of the vector.

Standard clustering algorithms such as K-means and Expectation-Maximization (EM) are iterative techniques which converge to local minima of an objective segmentation function. The running time of each iteration is proportional to the number of features[3]. In addition, the chances of getting stuck in local minima grow with the dimensionality of the feature vector. Thus, there is a natural tradeoff involved in choosing the size of the feature vector.

In addition to subband magnitudes, section 4.1 suggests coefficient auto-correlations and magnitude correlations within and among subbands may serve as potential features. Including correlations result in hundreds of features: far too many to be directly used in segmentation algorithms.

An attractive solution involves choosing a rich feature set, and adaptively reducing the

[3]For EM, this assumes a simple Gaussian model as described in section 4.4.2.

dimensionality of this set for each individual segmentation problem. A classical approach to reducing the feature dimensionality is Principal Components Analysis (PCA). In PCA, features are projected onto a subspace which captures most of the variance of the feature data. As demonstrated in section 4.3.1, however, scatter within classes can conceal the scatter between classes, causing class distinctions to be lost in the projected features.

A method of Linear Discriminant Analysis (LDA) due to Fisher [Fis36] overcomes this drawback of PCA by finding the subspace which maximizes the ratio of inter-class to intra-class scatter. The formulation relies on having a training set of classified data. Belhumeur *et al.* [BHK97] applied LDA to the problem of face recognition in which training sets were provided. The resulting feature axes (called "Fisherfaces") effectively categorized faces under a broad range of lighting conditions and facial expressions. Unfortunately, LDA cannot be applied directly to blind texture segmentation, since a labeled training set is not available.

Recent advances in segmentation algorithms permit multiple feature sources to be simultaneously considered without complex statistical modeling. Shi *et al.* [SBLM98] present a graph theoretic approach to maximize both inter-class differences and intra-class similarity through efficient approximation of *normalized cuts*. Hofmann *et al.* [HPB98] directly define a global objective function to maximize inter-segment similarities and intra-segment differences which has the quality of *shift invariance*, removing the potential negative effects of multiple feature sources.

In the following section, we present a novel variation of LDA, called "unsupervised LDA" (ULDA), that projects features onto a linear subspace that simultaneously maximizes inter-class differences and intra-class similarities, without the use of a texture training set. This is accomplished by approximating the underlying texture statistics with local averages of image features. The resulting reduced feature subspace can then be segmented using K-means or EM (expectation-maximization) analysis.

4.3 Projecting onto a Feature Subspace

The problem of unsupervised texture segmentation relates to the general problem of classification: Given a set of N items with n-dimensional quantitative *feature vectors*

$\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, is there a function mapping each item to one of c classes[4] such that visually "similar" items are assigned to the same class? In texture segmentation, the N items are typically the individual pixel locations and the classes are the different textures to which pixels are assigned. A common approach is to first reduce the dimensionality of the feature space through a linear projection W onto an m-dimensional subspace (where $m < n$) which attempts to maintain the differences between classes. The resulting reduced-dimensional features $\mathbf{y}_k \in \mathbb{R}^m$ are defined as:

$$\mathbf{y}_k = W^T \mathbf{x}_k \qquad k \in \{1, 2, ..., N\}$$

where $W \in \mathbb{R}^{n \times m}$ has orthonormal columns.

4.3.1 PCA

The total scatter matrix S_T of the features is defined as

$$S_T = \sum_{k=1}^{N} (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T$$

where μ is a vector containing the mean of the features $\{\mathbf{x}_k\}$. The scatter of the reduced-dimensional features $\{\mathbf{y}_k\}$ is therefore given by $W^T S_T W$. The goal of principal component analysis (PCA) is to calculate a projection W_{PCA} which maximizes the scatter of the projected features:

$$\begin{aligned} W_{PCA} &= \arg\max_{W} \left| W^T S_T W \right| \\ &= [\mathbf{w}_1 \mathbf{w}_2 ... \mathbf{w}_m] \end{aligned} \tag{4.1}$$

where $||$ denotes the matrix determinant and $\{\mathbf{w}_1, ..., \mathbf{w}_m\}$ are the m eigenvectors of S_T with the greatest eigenvalues.

Unfortunately, the maximization of the scatter in equation (4.2) includes intra-class scatter along with inter-class scatter. Figure 4.6 presents a two-dimensional example where PCA projects features onto a one-dimensional space which hopelessly mixes the samples from the two classes.

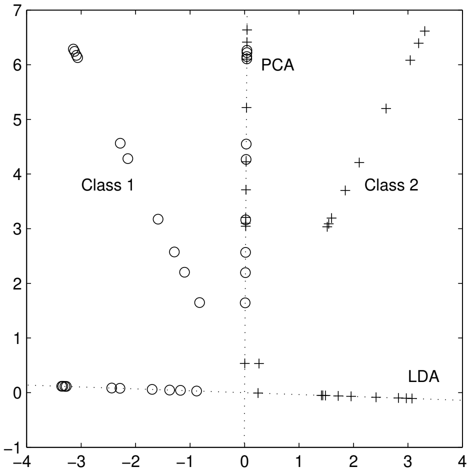


Figure 4.6: Comparison of PCA and LDA. In this example, the intra-class scatter of the '+' and 'o' classes causes PCA to incorrectly project them onto a vertical axis, resulting in impossible classification. LDA correctly maximizes the inter-class scatter resulting in a better projection.

4.3.2 LDA

Linear discriminant analysis (LDA) can be used where the example feature sets X_j for each class are known. Fisher's Linear Discriminant seeks to maximize the ratio of inter-class to intra-class scatter in the projected subspace. Inter-class scatter is defined as:

$$S_B = \sum_{j=1}^{c} N_j(\mu_j - \mu)(\mu_j - \mu)^T$$

where N_j is the number of samples in X_j. The intra-class scatter is defined as:

$$S_W = \sum_{j=1}^{c} \sum_{k \in X_j} (\mathbf{x}_k - \mu_j)(\mathbf{x}_k - \mu_j)^T.$$

LDA calculates a projection W_{LDA} which maximizes the ratio of the inter-class scatter to the intra-class scatter:

$$\begin{aligned} W_{LDA} &= \arg\max_{W} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \\ &= [\mathbf{w}_1 \mathbf{w}_2 ... \mathbf{w}_m] \end{aligned} \tag{4.2}$$

where $\{\mathbf{w}_1, ..., \mathbf{w}_m\}$ are the m eigenvectors with the greatest eigenvalues of the generalized eigenvalue problem:

$$S_B W = S_W W D \tag{4.3}$$

[4]In many cases the number of classes c is also unknown. We do not address this issue in this paper and instead assume that c is known.

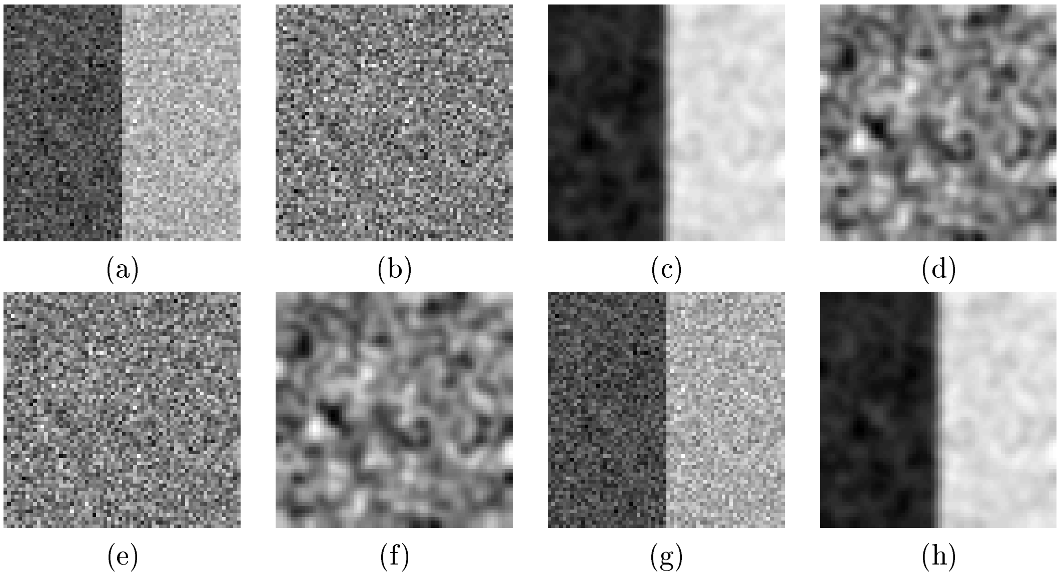


Figure 4.7: Comparison of PCA and ULDA. **(a)**, **(b)**: A two-dimensional feature set $\mathbf{x}$ similar to that in Figure (4.6) arranged into images. **(c)**, **(d)**: the blurred features ($\mathbf{x}_{avg}$). **(e)**, **(f)**: $\mathbf{y}$ calculated through PCA on the original and blurred features respectively. **(g)**, **(h)**: $W\mathbf{x}$ and $W\mathbf{x_{avg}}$ calculated through ULDA. Note that unlike PCA, ULDA successfully maintains the class separation without prior knowledge of the feature statistics.

in which D is a diagonal matrix.

In figure 4.6, LDA successfully separates two classes whose intra-class scatter dominates the PCA projection.

4.3.3 Unsupervised LDA

In unsupervised texture segmentation, class samples or training sets are not available. However, assume an image to be segmented has the following properties:

1. each pixel k belongs to exactly one texture class $X_{t(k)}$,
2. texture segments form contiguous regions in the image,
3. the feature vector $\mathbf{x}_k$ of each pixel k is the mean feature for its class plus additive zero-mean noise:

$$\mathbf{x}_k = \mu_{t(k)} + \mathbf{N}_k,$$

4. and the additive noise $\mathbf{N}_k$ is decorrelated between pixels and parametrized by the

class $X_{t(k)}$. Different features may have different zero-mean noise models, but class membership uniquely determines its noise distribution.

Property (1) is a general assumption in the problem of texture segmentation. Property (2) is used to justify the use of Markov Models or spatial dependence on class assignment [KC97]. Property (3) is necessary for most clustering techniques (PCA, LDA, K-means, and many general optimization functions). Algorithms that use averaging of features to estimate the underlying texture statistics (K-means and EM) require property (4).

Assume that pixel k and all pixels in a local neighborhood η_k belong to the same texture class (as a result of properties (1) and (2)). Provided $w : \eta_k \rightarrow \mathbb{R}$ is a non-negative weighting function summing to one, by properties (3) and (4), the local weighted average of features $\mathbf{x}_{avg}(w, k)$ defined as:

$$\mathbf{x}_{avg}(w, k) = \sum_{b \in \eta_k} w(b)\mathbf{x}_b$$

has a mean of $\mu_{t(k)}$ and a variance less than or equal to the variance of $\mathbf{N}_{t(k)}$ (equal if and only if w is a delta function). Therefore, $\mathbf{x}_{avg}(w, k)$ is a better estimate of $\mu_{t(k)}$ than $\mathbf{x}_k$.

Using local averages, we can define a new unsupervised classification technique called "unsupervised LDA" (ULDA), which maximizes the ratio of inter- to intra-class scatter. Inter-class scatter is estimated as the scatter of local averages:

$$S'_B = \sum_{k=1}^{N} (\mathbf{x}_{avg}(w, k) - \mu)(\mathbf{x}_{avg}(w, k) - \mu)^T$$

where μ is the mean of $\mathbf{x}_{avg}$, and intra-class scatter is estimated by the difference of feature vectors and the local averages:

$$S'_W = \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{x}_{avg}(w, k))(\mathbf{x}_k - \mathbf{x}_{avg}(w, k))^T.$$

ULDA determines a projection W_{ULDA} through the standard LDA equations (4.2) and (4.3) but using the unsupervised calculations of the inter- and intra-class scatter.

Figure 4.7 presents a two feature image set (with distributions similar to those given in figure 4.6) and the single dimension projections resulting from PCA and ULDA. The first feature, shown in figure 4.7(a), is a unit step edge with independent additive Gaussian noise with a variance of 1/4. The second feature, shown in figure 4.7(b), is independent random Gaussian noise with a variance of 1. Even without prior knowledge of the feature statistics, ULDA is able to maintain class separation while PCA projects the two classes together.

4.4 Segmentation

In order to use ULDA for blind texture segmentation, a classification algorithm must be applied to the resulting reduced-dimensional feature set. In this section we briefly describe the K-means algorithm and an expectation-maximization (EM) algorithm with a simple Gaussian model.

4.4.1 K-means

Given the number of classes c, feature vectors $\{\mathbf{x}_k\}$, and an initial guess of the class feature means $\{\mu'_j\}$, the K-means algorithm alternates between the following two steps:

1. Assign each pixel k to the feature set X'_j whose mean μ'_j is the closest in Euclidean distance to the feature vector $\mathbf{x}_k$

2. Reestimate the feature means (where N'_j denotes the number of pixels in X'_j):

$$\mu'_j = \frac{1}{N'_j} \sum_{k \in X'_j} \mathbf{x}_k \qquad j \in 1, ..., c.$$

These two steps are iterated until convergence.

4.4.2 EM

The expectation-maximization (EM) algorithm is an iterative technique to calculate the maximum likelihood of unknown data given observable data and a joint model relating the two data sets [DLR77]. Specifically, EM searches for statistical parameters θ' from a possible parameter set $\mathbf{\Theta}$ (i.e. the class feature means) which satisfy:

$$p(\mathbf{o} = \mathbf{o}'|\theta') \quad = \quad \arg \underset{\theta \in \mathbf{\Theta}}{max} \sum_{\mathbf{u}' \in \mathbf{U}} p(\mathbf{o} = \mathbf{o}', \mathbf{u} = \mathbf{u}'|\theta)$$

where $\mathbf{o}$ represents observed data (i.e. the feature vectors for the image being segmented), $\mathbf{u}$ represents unobserved data (i.e. the assignment of pixels to texture classes), and $\mathbf{U}$ is the set of all possible unobserved data values.

Let us assume that:

1. the class assignments $\mathbf{u}$ are independent of the texture means θ,

2. the feature vectors of $\mathbf{o}$ are conditionally independent given the class assignments $\mathbf{u}$:

$$
\begin{aligned}
& p(\mathbf{o} = \mathbf{o}' | \mathbf{u} = \mathbf{u}') \\
& = \prod_k p(\mathbf{o}_k = \mathbf{x}_k | \mathbf{u}_k = \mathbf{u}'_k)
\end{aligned}
$$

3. each element of a feature vector is independent and has a Gaussian distribution given its texture membership and class mean:

$$
\begin{aligned}
& p(\mathbf{o}_k = \mathbf{x}_k | \mathbf{u}_k = j, \theta_j = \mu_j) \\
& = \prod_i Gauss(\mathbf{x}_{ki}, \mu_{ji}, \sigma^2_{ji})
\end{aligned}
$$

   where $Gauss(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/\sigma^2}$,

4. the texture feature variances σ^2_k are known or well approximated,

5. and that the distribution of texture assignments is uniform.

It can be shown that EM then reduces to a version of K-means with "soft class assignments:"

- (E step) For each pixel, calculate the conditional probability that the pixel is in each texture class:

$$
\begin{aligned}
f(j|k) &= p(\mathbf{u}_k = j | \mathbf{o}_k = \mathbf{x}_k, \theta_j = \mu_j) \\
&\propto p(\mathbf{o}_k = \mathbf{x}_k | \mathbf{u}_k = j, \theta_j = \mu_j) \\
&\propto \prod_i Gauss(\mathbf{x}_{ki}, \mu_{ji}, \sigma^2_{ji})
\end{aligned}
$$

- (M step) For each texture, recalculate the expected mean feature vector given the conditional probabilities:

$$
\mu'_j = \frac{\sum_{k=1}^{N} \mathbf{x}_k \, f(j|k)}{\sum_{k=1}^{N} f(j|k)}
$$

Unlike K-means, a judicious choice of σ_j^2 may lead to automatic determination of the number of classes in EM [WA96].

4.5 Implementation

4.5.1 Choice of Features

Images are decomposed using a complex steerable pyramid [SP98]. From section 4.1 and similar to the those used in [PS99] for synthesis, we use the following features for segmentation:

- Subband magnitude point statistics: The magnitudes of the complex coefficient ($L*O$ features, where L is the number of pyramid levels and O is the number of orientations). By itself, this set is representative of the commonly used localized spatial-frequency response features.

- Real valued coefficient correlations: The product of the real part of each subband magnitude with B local neighbors within the subband ($L*O*B$ features).

- The square root of intra-subband coefficient magnitude correlations: The square root of the product of each subband magnitude with B local neighbor magnitudes ($L*O*B$ features).

- The square root of inter-subband coefficient magnitude correlations: The square root of the product of each subband magnitude with its parent ($(L-1)*O$ features) and the square root of the product of each subband magnitude with magnitudes at the same level but different orientations ($L*O*(O-1)/2$ features).

4.5.2 Empirical Determination of Variables

To determine practical values for the number of levels L, the number of orientations O and the blur filter w, a test set of 50 images containing 4 randomly chosen square texture patches from the set of 31 textures (shown in figure 4.1) were segmented. Textures were normalized for mean and variance to reduce the ability to segment them using only point statistics of pixel intensity. Figure 4.8 shows a sample 4 patch image. A search was

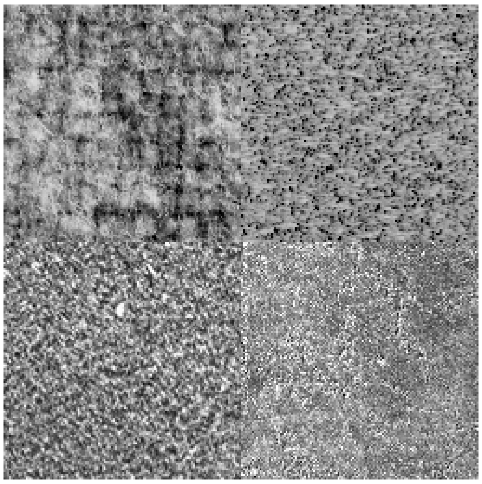


Figure 4.8: Example of a 4-texture mixture. The textures were normalized for mean and variance in pixel intensity before being used within the texture mixtures. Actual segmentation results for this mixture are shown in figure 4.10.

conducted to minimize the segmentation error of the test set using K-means and EM with only the blurred magnitudes of the complex subbands as features. The blur filter w was chosen to be Gaussian so only the variance needed to be determined. In EM, the variance ${\sigma^2}_{ji}$ for each model was set to the sample variance of the i^{th} element of the feature vector over the whole image. Figure 4.9 displays error results holding all but one variable fixed.

Without the use of feature dimension reduction techniques, there is a tradeoff between the number of features and the running time of the segmentation algorithm. Also, a larger blurring variance increases the chance of averaging features over texture boundaries. Therefore, for experiments using only complex subband magnitudes as features, the number of levels L was chosen to be 2, the number of orientations O was chosen to be 8 (resulting in 16 features), and the variance of the blur filter w was chosen to be 10.

For tests using inter-subband and intra-subband correlations as features, the number of orientations was reduced to 4, and only 4 local neighbors (up, up-left, left, and down-left) were used for intra-subband correlations, resulting in 90 features.[5]

[5]Using 8 orientations with inter-subband and intra-subband correlations (resulting in 208 features) did not yield a significant decrease in segmentation error.

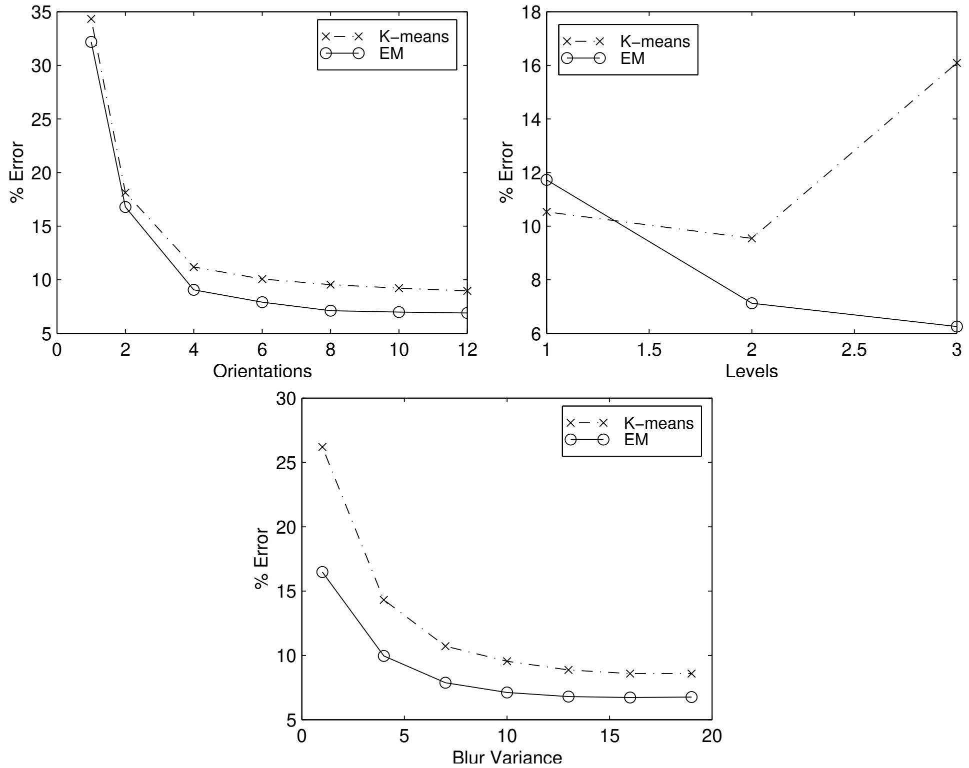


Figure 4.9: Segmentation error vs. varying feature parameters. For these graphs, two parameters were held constant while one was allowed to vary. **Top Left**: Varying number of orientations. **Top Right**: Varying number of pyramid levels. **Bottom**: Different variances of the blur filter.

4.6 Results

4.6.1 Comparison of Feature Reduction Techniques

To test the efficacy of ULDA, we generated mixtures from a library of 31 textures shown in figure 4.1 (12 Brodatz textures and 19 other textures including water, fabric, stone, trees, food, and grass). As in the previous section, textures were normalized for mean and variance to reduce the ability to segment them using only point statistics of pixel intensity. Two tests were conducted: 115 segmentations of images containing 4 randomly chosen square texture patches, and 115 segmentations of images containing 2 textures arranged in randomly generated irregular and disjoint regions (formed by thresholding a blurred fractal image). The resulting images are 256x256 pixels in size.

In these tests, only the magnitudes of 16 complex steerable pyramid subbands were used as features (2 scales, 8 orientations). These features are projected onto a $(c-1)$-dimensional space (where c is the number of textures in the mixture, assumed known) using PCA, LDA, and ULDA as described in section 4.3. For LDA, the ideal segmentation was used to calculate the inter-class and intra-class scatter matrices. Therefore, LDA represents an ideal linear feature dimension reduction transform. For ULDA, features were blurred using a 21 tap Gaussian filter with a variance of 10. The resulting feature subspace was then used as input to both K-means and EM algorithms, as described in section 4.4. In EM, the variance ${\sigma^2}_{j_i}$ for each model was set to the sample variance of the i^{th} element of the feature vector over the whole image.

To initialize both EM and K-means algorithms, feature mean guesses for 20 classes were uniformly generated within the range of the image features. Duplicated or unused classes were removed after every 4 iterations until c classes were reached. The algorithms were allowed to iterate until convergence was reached (typically 10-20 iterations).

Table 4.1 presents the percentage of incorrect classifications for each technique over 115 segmentation trials. ULDA outperformed both PCA and the original full feature set using either EM or K-means. EM outperformed K-means for the segmentation on both full and reduced feature sets.

When compared to supervised LDA, which in this case relied on prior knowledge of the ideal segmentation to calculate the scatter matrices, ULDA performs well, but clearly

	4 Square Regions			
	Full Set	PCA	LDA	ULDA
K-means	9.10	11.04	6.41	6.78
EM	6.73	13.65	3.92	6.41

	2 Irregular/Disjoint Regions			
	Full Set	PCA	LDA	ULDA
K-means	9.38	10.54	6.45	7.82
EM	7.82	10.22	5.51	6.81

Table 4.1: Errors of different feature projection and segmentation techniques. Shown are percent incorrect classifications for two sets of 115 tests. Textures were randomly drawn from a library of 31. LDA uses the ideal segmentation to determine the scatter matrices, and therefore represents ideal linear feature dimension reduction. The ULDA feature set also has a large improvement in processing time (please see the text).

does not find the ideal subspace in which to project the features. Because of a lack of prior classification knowledge, ULDA uses local averages which are calculated across texture boundaries. Therefore, a potential two step unsupervised segmentation algorithm is possible: first ULDA can be used for an initial segmentation, and the result could to calculate LDA scatter matrices, which are then used for a final segmentation. This two step algorithm is left for future research.

When compared to the full feature set, the ULDA algorithm provides a substantial improvement in efficiency. The time cost per iteration of K-means and EM is roughly linear with the number of features. On a 200 MHz Pentium machine running Matlab, for a 4-texture segmentation with the full set of 16 features, 100 iterations of EM took 150.7 seconds and 86.2 seconds for K-means, while for the 3 feature ULDA set 100 iterations of EM took 39.4 seconds and 25.0 seconds for K-means. Surprisingly, the number of iterations needed for convergence with the full feature set and the ULDA-reduced feature set were the same: an average of 16 iterations for EM and 20 for K-means for 4-texture segmentation. Therefore, the total time requirements for segmentation are roughly linear with the number of features.

Figure 4.10-4.13 displays examples of the 2-texture and 4-texture segmentations with error percentages for each. Also shown are images whose pixels are the projections of the

	4 Square Regions	2 Irregular Regions
Mag	6.41	6.74
Intra-Real	6.26	7.13
Intra-Mag	8.39	7.01
Inter-Mag	5.83	6.56
Intra-Real, Intra-Mag, Inter-Mag	3.67	6.04
Mag, Intra-Real	4.22	6.06
Mag, Intra-Real, Intra-Mag	4.48	6.09
Mag, Intra-Real, Inter-Mag	3.56	6.00
Mag, Intra-Real, Intra-Mag, Inter-Mag	3.37	6.01

Table 4.2: Errors of different feature sets. Shown are percent incorrect classifications for two sets of 115 tests. "Mag" represents the complex subband magnitudes, "Intra-Real" represents intra-subband real correlations, "Intra-Mag" represents intra-subband magnitude correlations, and "Inter-Mag" represents inter-subband magnitude correlations. For the four patch segmentation, the added features reduce segmentation errors almost by half over the standard feature set of subband magnitudes alone.

feature vectors onto the first few axes of the reduced feature subspace. In most segmentations, the ULDA axes significantly improve the ability to discriminate textures than the PCA axes.

4.6.2 Comparison of Feature Sets

Tests were also conducted using different combinations of the four feature sets described in section 4.5.1. As in the previous section, 115 images each of 4-square patch and 2-fractal patch images were segmented. In this case, ULDA was used to project the feature vector onto a $(c-1)$-dimensional space followed by EM segmentation. Table 4.2 presents the percentage of incorrect classifications for each set of features. For a fair comparison, when complex magnitudes were used alone, a full 8 orientations were used (resulting in 16 features). When other feature sets were present, 4 orientations were used as described in section 4.5.2.

For the 4-patch segmentation, the inclusion of the inter-subband and intra-subband features reduces the segmentation errors almost by a half when compared to the standard feature set of subband magnitudes alone. In general, adding elements to the set increased the performance. Though adding intra-subband magnitude correlations to the set of subband magnitudes and intra-subband real correlations slightly increased errors, its presence in the full set of features aided segmentation. For the 2-patch segmentation, the improvement was not as significant (only about 10%). An explanation for this is that the magnitudes alone were adequate to differentiate two textures, and most of the segmentation errors were caused by the irregular shape of the regions.

In Figures 4.10-4.13 the bottom row displays the results of segmenting images using the full collection of features.

4.6.3 Segmentation of Natural Images

Figures 4.14-4.16 displays the successful segmentations of 3 natural images using the ULDA/EM algorithm. Note that the segmented regions shown in the "Cheetah" and "Zebra" examples capture the pixels corresponding to the animals, including their limbs.

4.7 Discussion

We have presented ULDA, an unsupervised algorithm based on LDA for feature dimension reduction. When used in conjunction with EM or K-means for texture segmentation, ULDA resulted in up to 50% less segmentation errors than the full 16-dimension feature set or a PCA reduced feature set. In addition, ULDA reduced segmentation time to almost 25% of the time required for segmentation with the full feature set.

When compared to LDA trained with the ideal segmentation, ULDA produced almost twice the segmentation errors in our 4-patch experiments. This can be explained by ULDA's use of averages calculated across texture boundaries. A potential improvement would be the use of a two step unsupervised algorithm: first applying ULDA to segment the image, and then using LDA trained with the ULDA results to segment the image again.

In addition, we have presented an expanded feature set which uses correlations between subband coefficients and magnitudes. These new features reduced the segmentation errors by half when compared to the standard feature set of subband energies. Subband correlations, however, result in hundreds of features, and therefore their use requires a feature dimension reduction technique like ULDA.

As can be seen in figures 4.10-4.13, the predominant errors in segmentation occur at texture boundaries. This is explained by the averaging over texture boundaries used by

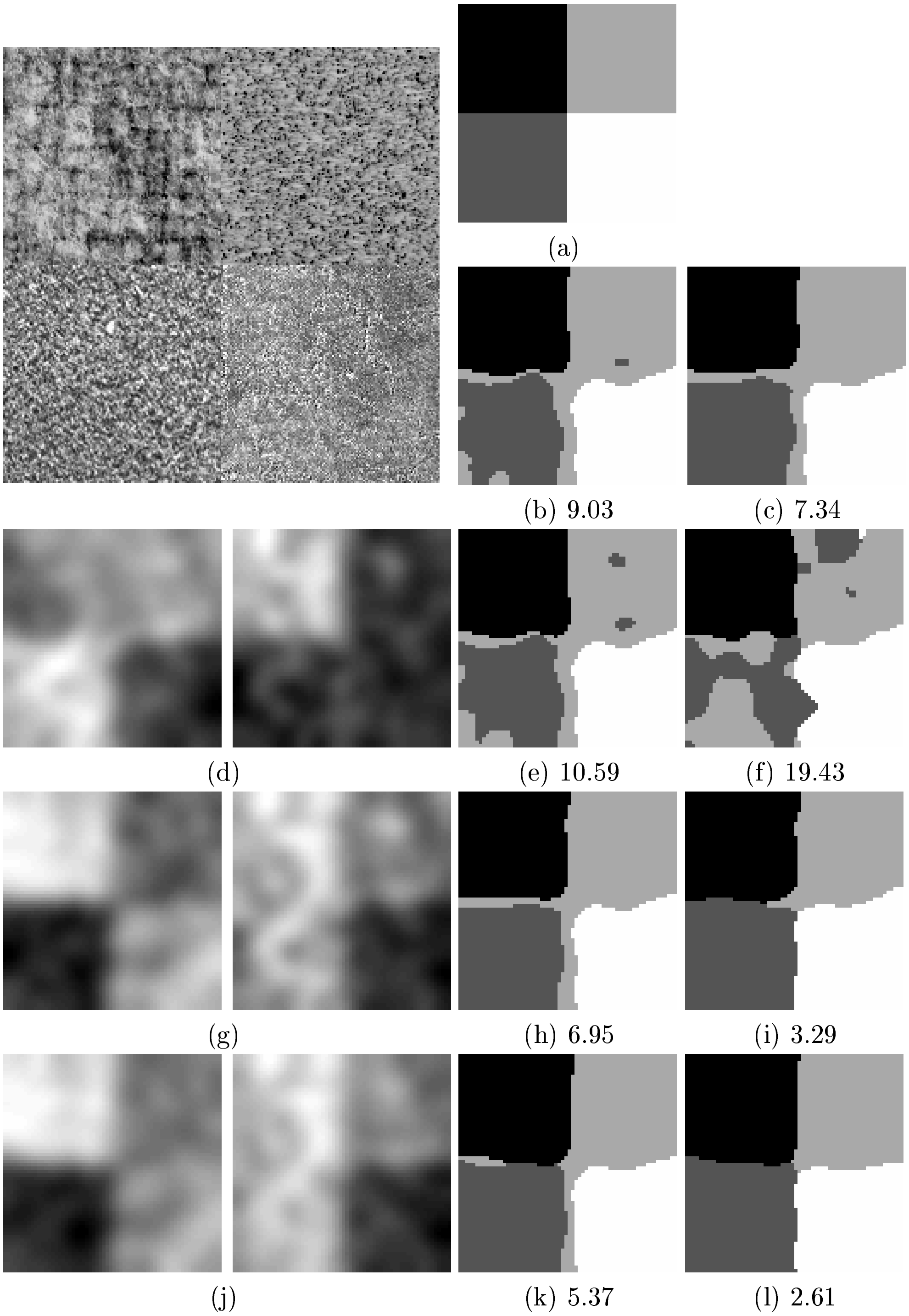
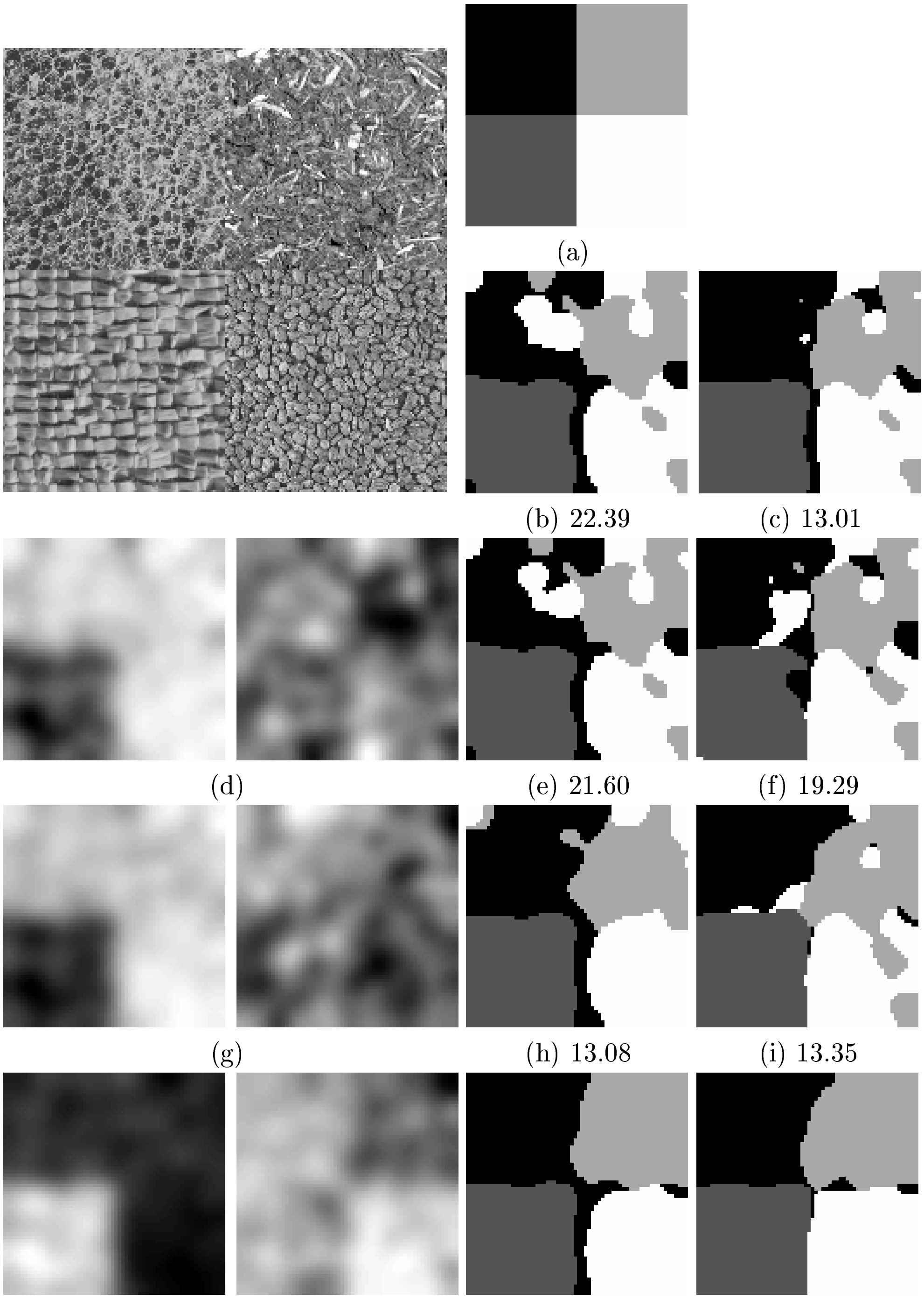


Figure 4.10: Example of blind 4-texture segmentation. Under the segmentations are the percentage of error. Items (b) through (i) were done with subband magnitudes as features. Items (j) through (l) were done with the full set of magnitudes and correlation features. **(a)** Ideal segmentation. **Raw magnitude features: (b)** K-means, **(c)** EM. **PCA-reduced magnitude features: (d)** most significant eigenvalue PCA features, **(e)** K-means, **(f)** EM. **ULDA-reduced magnitude features: (g)** most significant eigenvalue ULDA features, **(h)** K-means, **(i)** EM. **ULDA-reduced magnitude and correlation features: (j)** most significant eigenvalue ULDA features, **(k)** K-means, **(l)** EM.



(a)

(b) 22.39 (c) 13.01

(d) (e) 21.60 (f) 19.29

(g) (h) 13.08 (i) 13.35

(j) (k) 6.62 (l) 4.30

Figure 4.11: Another example of blind 4-texture segmentation. Under the segmentations are the percentage of error. Please see figure 4.10 for definitions of labels.

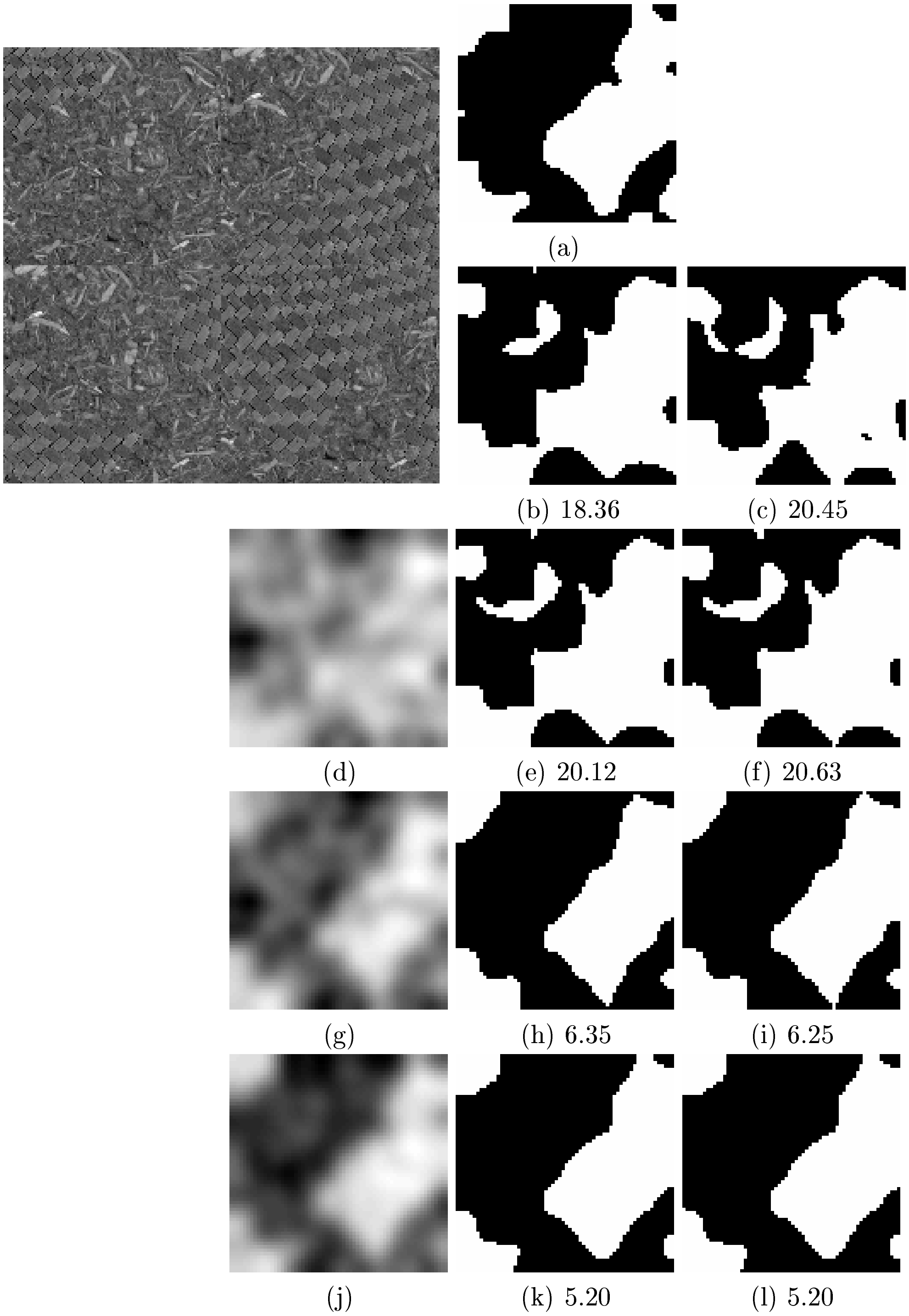


Figure 4.12: Example of blind 2-texture segmentation. Under the segmentations are the percentage of error. Please see figure 4.10 for definitions of labels.

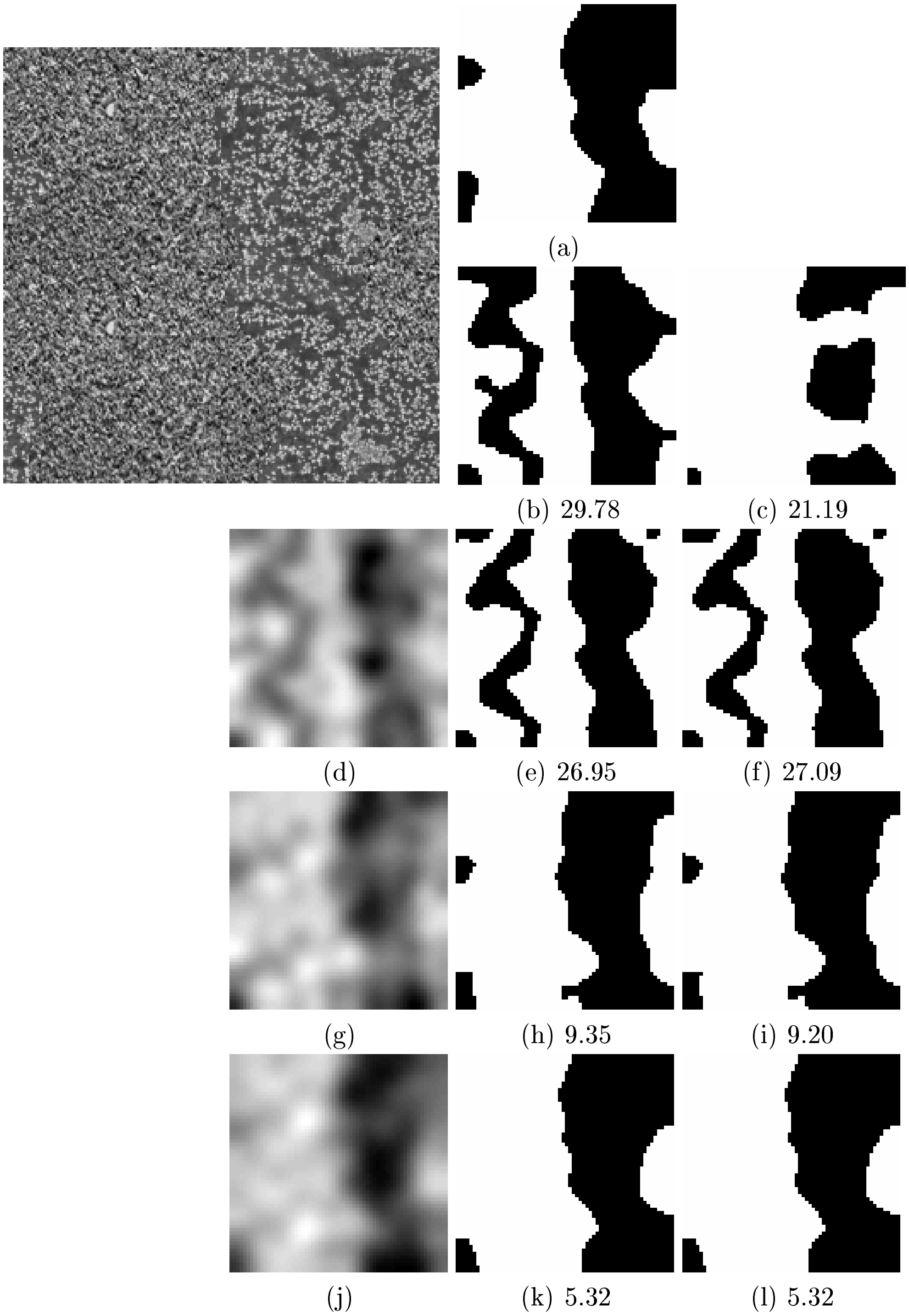


Figure 4.13: Another example of blind 2-texture segmentation. Under the segmentations are the percentage of error. Please see figure 4.10 for definitions of labels.

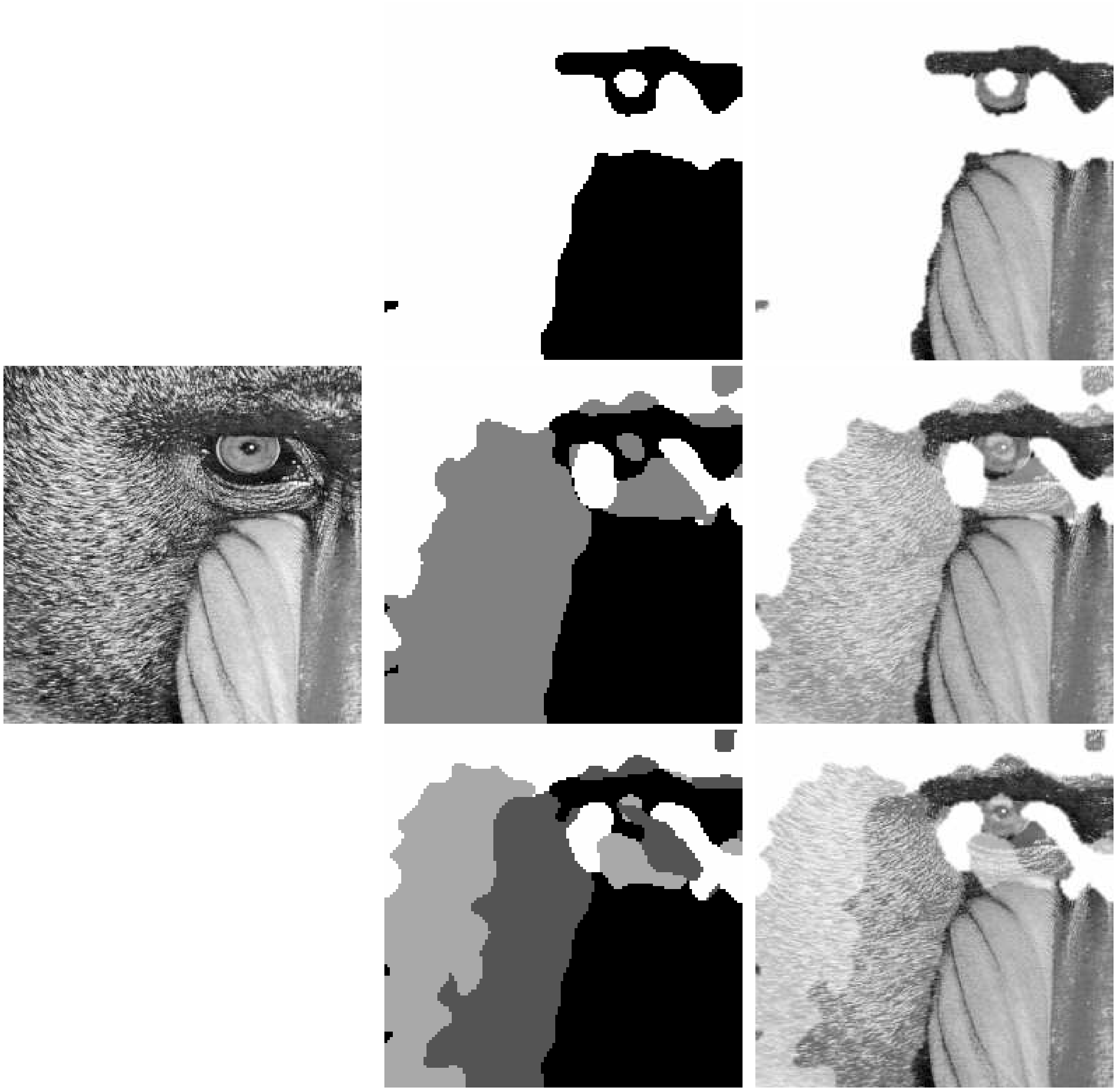


Figure 4.14: Blind segmentation of natural images using EM on a ULDA-reduced feature set. Image is a 256x256 section of "Mandrill". **Left column:** original image. **Middle column:** segmentations, with increasing number of classes. **Right column:** same, with original image projected onto the classes.

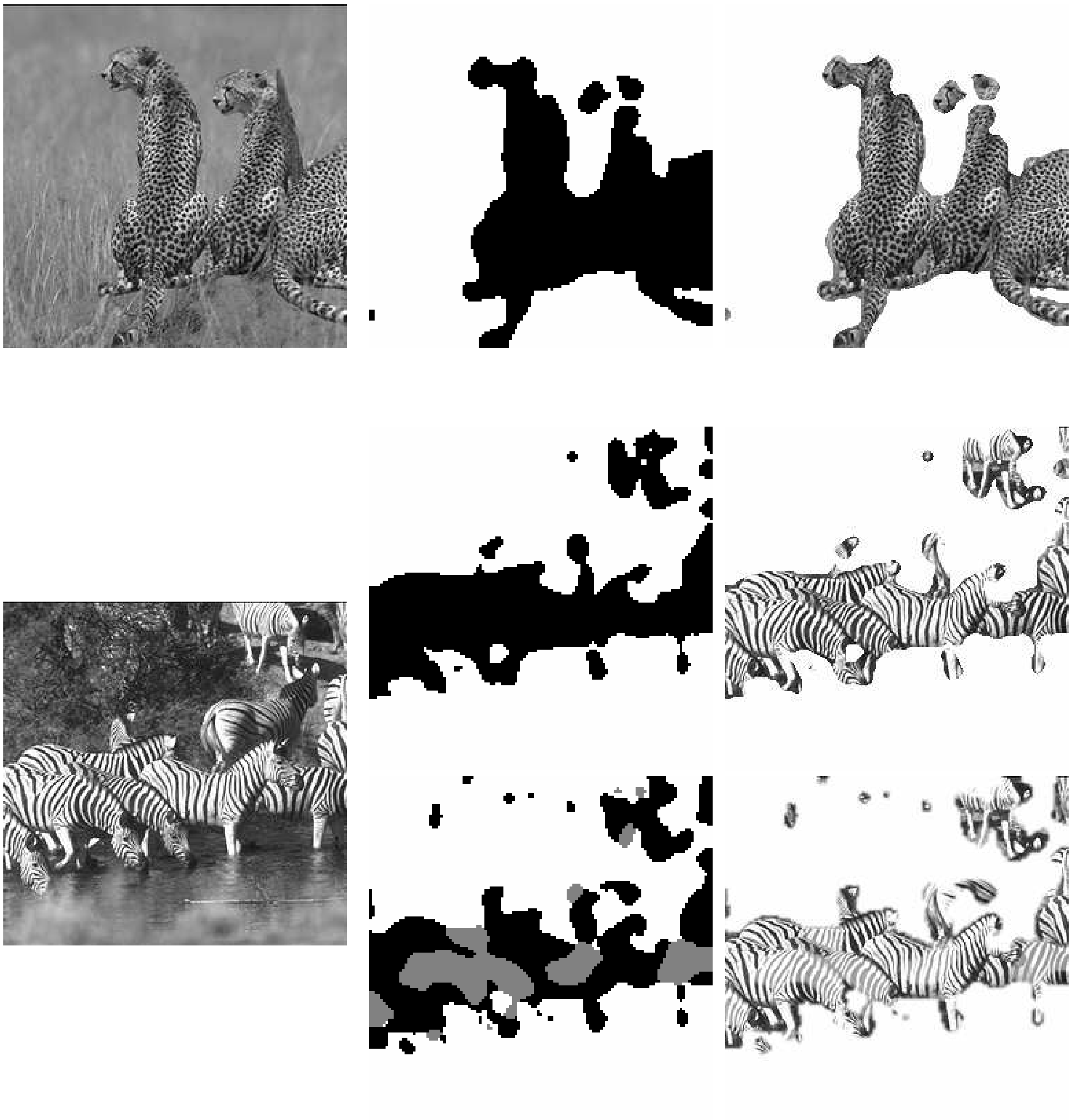


Figure 4.15: Blind segmentation of natural images using EM on a ULDA-reduced feature set. Images are 256x256 sections of "Cheetah" and "Zebra". **Left column:** original images. **Middle column:** segmentations, with increasing number of classes. **Right column:** same, with original image projected onto the classes. Thanks to Jianbo Shi for providing the Cheetah and Zebra images.

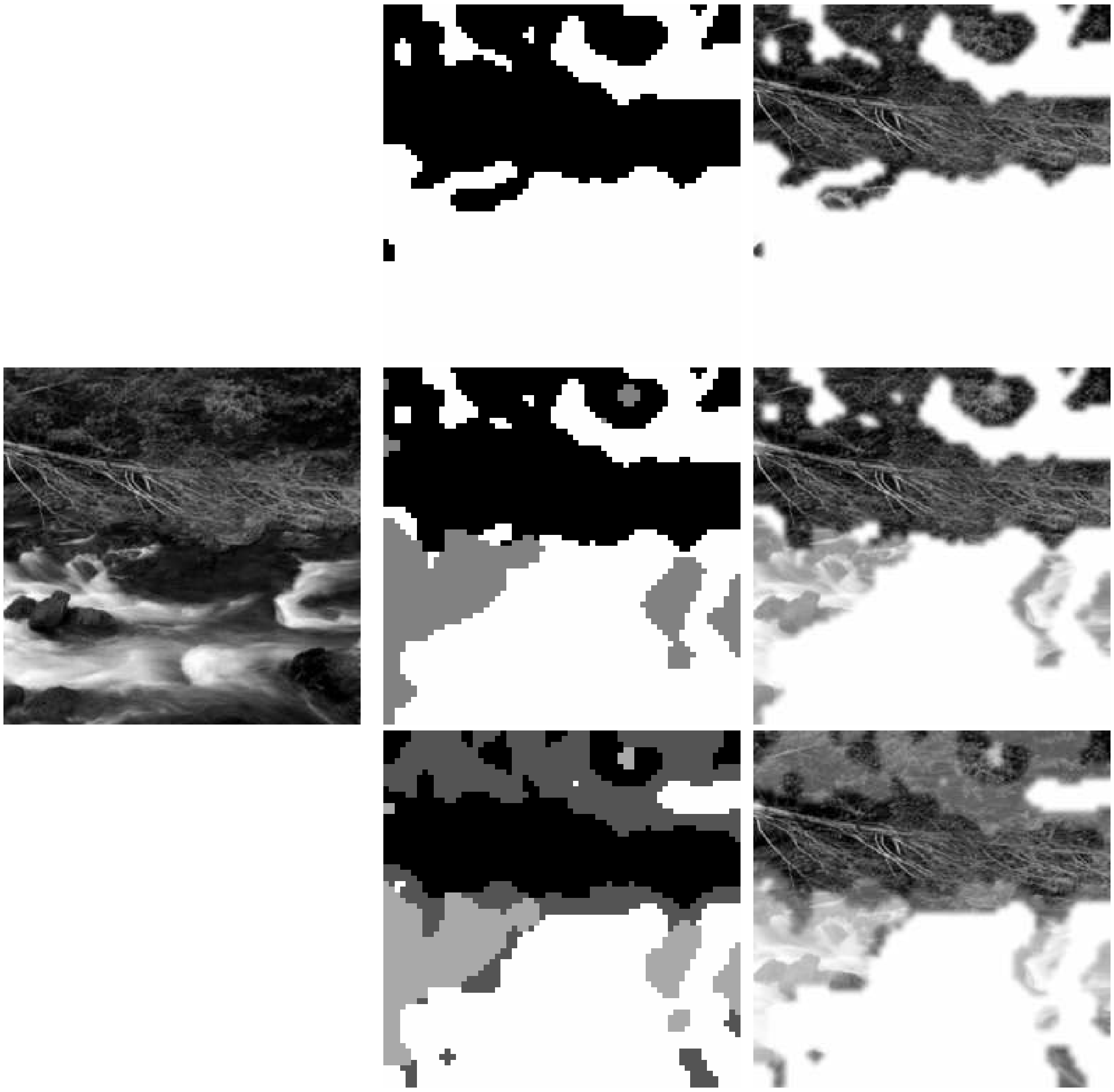


Figure 4.16: Blind segmentation of natural images using EM on a ULDA-reduced feature set. Image is a 256x256 section of "Cantrell". **Left column:** original image. **Middle column:** segmentations, with increasing number of classes. **Right column:** same, with original image projected onto the classes.

ULDA in scatter matrix estimation. Again, these errors could potentially be reduced by using the ULDA segmentation as a training set for LDA feature reduction.

However, as can be seen in figures 4.14, a gradual change in texture qualities forces EM or K-means to generate artificial segmentation boundaries. The gradual shift in the orientation of the baboon's hair is segmented by orientation due to the rigid definition of the segmentation classes. This could potentially be avoided through an algorithm which analyzes discontinuities in features instead of direct classification. In addition, as can be seen in figures 4.14-4.16 the EM and K-means segmentation creates boundaries at shadows and other strong intensity changes. A potential solution would be to normalize correlation features by the local variance, forming a normalized correlation potentially immune to such intensity variations.

Chapter 5

Conclusion

We have presented a statistical model that successfully characterizes the conditional relationships between the magnitudes of a wavelet pyramid decomposition. In this model, a linear predictor for a subband coefficient magnitude is formed by a weighted sum of its neighboring magnitudes within and among subbands local in orientation and scale. We have found that the log difference between a magnitude and its predictor has a characteristic functional form shared among all of our test images which can be stored in a look-up table.

The use of a linear magnitude predictor and the resulting model are contributions to image processing due to their direct characterization of inter- and intra-subband relationships, yielding a better intuition for the properties of natural images, and their immediate application to the fields of compression and image restoration. Indeed, when we used the model to develop EPWIC-II, the progressive image coder, the results are comparable to the state of the art. Unlike SPIHT and similar image coders, which use searching or other adaptive techniques to capture statistical properties, EPWIC-II directly calculates bit probabilities from the magnitude predictor model. EPWIC-II would not be appropriate for a commercial product since other encoders are more efficient and achieve slightly better compression, but EPWIC-II clearly demonstrates the efficacy of the magnitude predictor model.

Work can still be done to improve the model. Currently, the linear weights are calculated to minimize the mean squared error of the magnitude estimate. Assuming a specific distribution for the estimate errors, optimal weights could be found through maximum likelihood. Another technique suitable for image compression would be to calculate the weights to minimize the entropy of the prediction errors. In addition, it would be interesting to determine how this model can be extended to color images.

We have also presented a novel unsupervised feature dimension reduction technique suitable for general blind segmentation problems. Unsupervised Linear Discriminant Analysis (ULDA) allows hundreds of features to be used efficiently without the need for prior training or known classes. Since ULDA can be used wherever local averages of features closely approximate class means, it is a strong contribution to the general segmentation community. When used in conjunction with EM or K-means for texture segmentation, ULDA resulted in up to 50less segmentation errors than the full 16-dimension feature set or a PCA reduced feature set. In addition, ULDA reduced segmentation time to almost 25% of the time required for segmentation with the full feature set.

Future research in ULDA can be done to improve its use for image segmentation. Since most segmentation errors occur due to feature averaging across texture boundaries, a two step segmentation algorithm is possible. The results of an initial segmentation with ULDA could be used to calculate the scatter matrices with Fisher's Linear Discriminant Analysis. The resulting reduced feature set will be less susceptible to errors across texture boundaries, since averages will be taken over whole regions selected by the ULDA segmentation.

The successful use of inter- and intra-subband magnitude relationships in our linear magnitude model suggest their potential use as features in texture segmentation. However, magnitude correlations can result in hundreds of features, requiring the use of a feature reduction technique like ULDA. When we included magnitude correlations as features for texture segmentation, there was up to a 50% reduction in segmentation errors when compared to using subband variances alone.

The use of magnitude correlations for texture analysis is a contribution to the computer vision community. Their use in synthesis is difficult due to the daunting task of simultaneously matching intensity statistics, subband variance statistics, and subband correlation statistics with the statistics of a sample texture. However, this has been recently solved by Javier Portilla and Eero Simoncelli and used to successfully generate complex periodic and aperiodic textures [Sim97].

There is still a gap between physical models of natural images and current understanding of their statistical properties. Though intra-subband and inter-subband relationships capture some of the qualities of object edges and features, their characterization is only a small step in analyzing how light portrays the world around us. In the end, continuing statistical analysis may lead to a definitive answer of the question: "What is an image?"

Appendix A

Signal Processing and Information Theory Tools

This section describes the multi-scale signal processing and information theoretic tools used in this paper.

A.1 The Gaussian Pyramid

The Gaussian pyramid is one of the simplest tools that can be used for multi-scale analysis of images [BA83]. The first level of the pyramid is defined to be the original image: $S_1 = I$. Successive levels of the pyramid are defined by recursively applying a lowpass filter (typically a 2-D Gaussian) and subsampling by 2 in both the x and y directions. The system diagram for one step of this process is given in Figure A.1. An example of a four level Gaussian pyramid is given in Figure 1.6(a). Note that each level of the pyramid is a contracted version of the previous level.

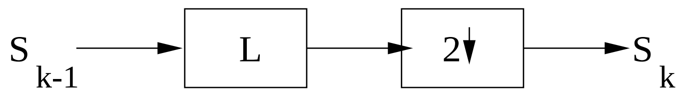


Figure A.1: A step in the creation of a Gaussian pyramid. The first level (S_1) is defined to be the original image. Successive levels are created by recursively applying a lowpass filter L and subsampling the result by 2 in both the x and y directions.

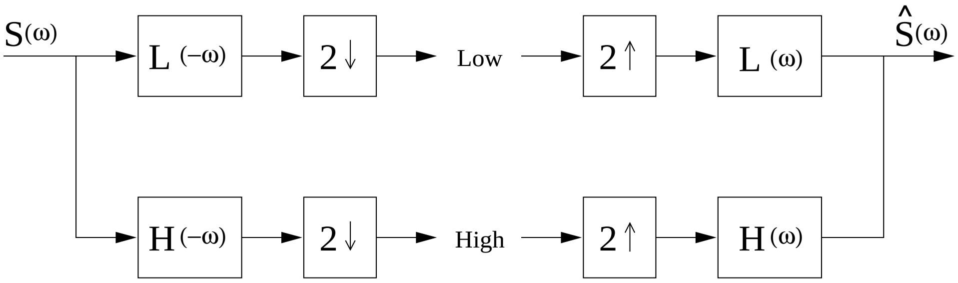


Figure A.2: One-dimensional system diagram for one-stage orthonormal subband decomposition.

A.2 The Wavelet Pyramid

The wavelet pyramid is another linear multi-scale decomposition with properties conducive for signal compression. Central to the one dimensional orthogonal wavelet pyramid is a pair of filters L and H (lowpass and highpass respectively) that satisfy:

$$||H(\omega)||^2 + ||L(\omega)||^2 = 2$$

For one dimensional signals, a single level of the pyramid is constructed through application of the system diagram given in Figure A.2.[1] The signal *low* is a lowpass version of the original signal with half the number of samples. Similarly *high* is a highpass version with half the number of samples. Therefore, the total number of samples in *low* and *high* (called "subbands") is equal the the number of samples in the original signal S. Since the original signal can be reconstructed from its subbands, this decomposition provides an alternative representation of the signal potentially useful in compression.

For images, a two dimensional wavelet decomposition can be performed by separably applying the four combinations of L and H in the x and y directions of the image. The result has four subbands, each one quarter the size of the original image: lowpass, horizontal, vertical, and diagonal. A pyramid is then formed by recursively repeating the decomposition on the lowpass subband. An example of a two level pyramid is given in Figure A.3. As in the one dimensional decomposition, the total number of samples in the subbands is equal to the number of samples in the original signal and the subbands can be used to reconstruct the original signal.

[1] A similar decomposition is the biorthogonal filter set which satisfies the one-dimensional system diagram shown in figure 3.1.

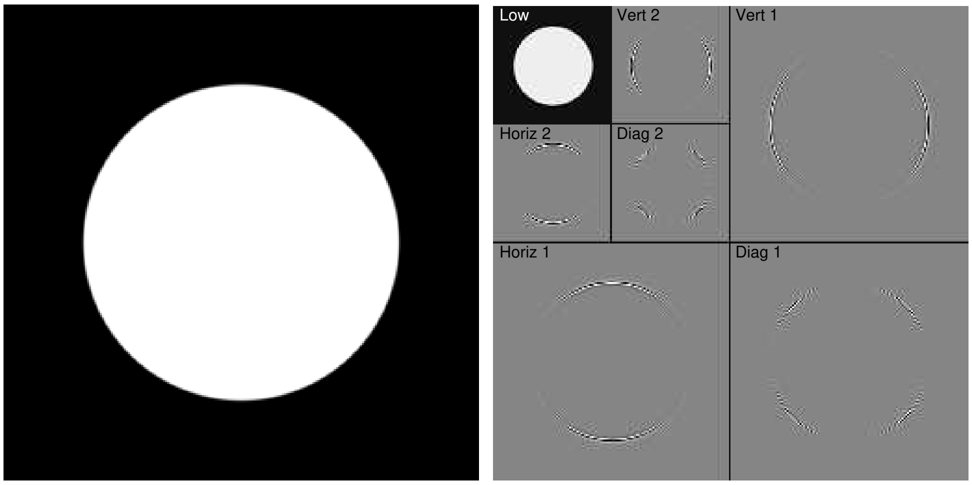


Figure A.3: Two level wavelet pyramid. Left: The original image of a white circle on a black background. **Right:** A two level wavelet pyramid formed form the image on the left. The larger three subbands belong to the first level and correspond to the highest frequencies. The lowpass of the first level is further decomposed to form another level of the pyramid.

A.3 The Steerable Pyramid

A steerable pyramid is a linear image decomposition, in which the image is subdivided into an collection of subbands localized in both scale and orientation [SFAH92, SF95]. The scale tuning of the filters is constrained by a recursive system diagram (described below). The orientation tuning is constrained by the property of steerability [FA91]. A set of filters form a steerable basis if (1) they are rotated copies of each other, and (2) a copy of the filter at any orientation may be computed as a *linear combination* of the basis filters. The simplest example of a steerable basis is a set of $N + 1$ Nth-order directional derivatives. Since steerable filters represent the response at all orientations with a small number of samples, they are useful in oriented image processing tasks [FA91]. Since the decomposition is *over-complete*, meaning that the number of coefficients in the decomposition is greater than the number of pixels in the original image, steerable pyramids are not used in image compression.

The steerable pyramid is built using the system diagram shown in Figure A.5 [SF95]. The highpass filter $H(r)$ and two lowpass filters $L_1(r)$ and $L_2(r)$ (functions of only the

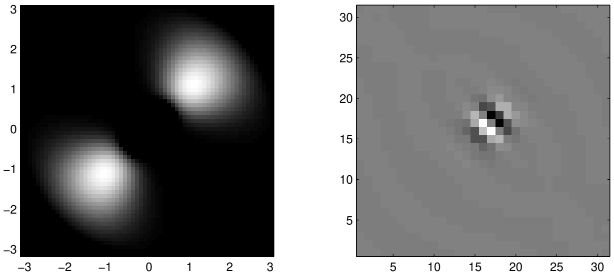


Figure A.4: One orientation of a steerable filter requiring 8 orientations for steering. **Left:** Filter in the Fourier domain **Right:** Impulse response.

radius in the Fourier domain) are constrained such that for all r and ω:

$$|H(r)|^2 + |L_0(r)|^2 \left[\sum_{n=0}^{N-1} |F_n(r,\omega)|^2 + |L_1(r)|^2 \right] = 1.$$

The oriented filters, F_n, are of the form:

$$F_n(r,\omega) = kB(r)(-i)^{N-1} cos\left(\frac{\omega n \pi}{N}\right)^{N-1} \tag{A.1}$$

where N is the number of sampled orientations, $n \in [0, 1, ..., (N-1)]$ is the filter number, $B(r)$ is a bandpass filter constrained by the pyramid recursion relationship, and k is a constant depending on N such that the squares of the raised cosine terms sum to 1 for all ω. An example is shown in Figure A.4. The filter L_1 must also bandlimit below the Nyquist rate, to avoid spatial aliasing in the subsampling operation.

A pyramid is formed by subsampling the response of L_1 by 2 and repeating the application of the steerable filters F_n and the lowpass filter L_1 to the result (in Figure A.5 the repeated operation is enclosed in a dashed box). The number of iterations determines the number of levels in the pyramid. The overall transform is a tight frame: it is inverted using a set of convolution kernels that are space-reversed copies of those use to build the pyramid.

A.4 Entropy and the Kullback-Leibler Distance

Given a discrete random variable X whose range is the set $\mathcal{X}$, it can be shown that in order to minimize expected encoding size, the number of bits required to encode the event $\{X = x\}$ is $-\log_2(\mathcal{P}(X = x))$ [CT91]. The entropy $\mathcal{H}(X)$ of X is the expected number

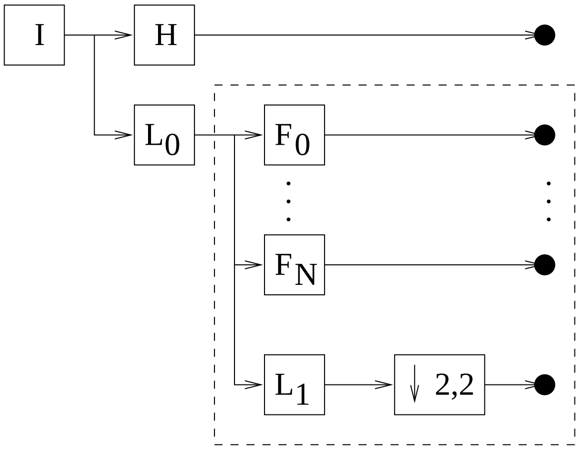


Figure A.5: Construction of a steerable pyramid. A pyramid is created by repeating the contents of the dashed box upon the downsampled result of L_1. The image can be reconstructed by reversing the system diagram with the time reversed filters.

of bits used to encode the state of X:

$$\begin{aligned}\mathcal{H}(X) &= \mathcal{E}_X\left[-\log_2(\mathcal{P}\left(X=x\right))\right] \\ &= -\sum_{x\in\mathcal{X}} \mathcal{P}\left(X=x\right)\log_2(\mathcal{P}\left(X=x\right)).\end{aligned}$$

For an ideal encoder, if the event $\{X = x\}$ is assumed to have probability q, the number of bits used to encode the event is $-\log_2(q(x))$. Therefore, the cost $\mathcal{H}(q||p)$ of encoding X using a probability model $q(x)$, when the actual distribution is $p(x)$ is the expectation of $\log_2(q(x))$ is:

$$\begin{aligned}\mathcal{H}(q||p) &= \mathcal{E}_X\left[-\log_2(q(x))\right] \\ &= -\sum_{x\in\mathcal{X}} p(x)\log_2(q(x)).\end{aligned}$$

Note that since p is the real distribution of X, $\mathcal{H}(X) = \mathcal{H}(p||p)$.

The Kullback-Leibler distance $\Delta\mathcal{H}(q||p)$ is the overhead of encoding a discrete random variable X with the assumed distribution q when the real distribution is p:

$$\begin{aligned}\Delta\mathcal{H}(q||p) &= \mathcal{H}(q||p) - \mathcal{H}(p||p) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log_2\left(\frac{q(x)}{p(x)}\right).\end{aligned}$$

Note that $\Delta\mathcal{H}(q||p) = 0$ if and only if $p(x) = q(x)$ for all $x \in \mathcal{X}$.

If Y is another discrete random variable, the conditional entropy $\mathcal{H}(X|Y)$ is the cost of encoding the state of X knowing the state of Y:

$$\begin{aligned}
\mathcal{H}(X|Y) &= \mathcal{E}_Y\left[\mathcal{E}_X\left[-\log_2(\mathcal{P}\left(X=x|Y=y\right))\right]\right] \\
&= -\sum_{y\in\mathcal{Y}} \mathcal{P}\left(Y=y\right) \sum_{x\in\mathcal{X}} \mathcal{P}\left(X=x|Y=y\right) \log_2(\mathcal{P}\left(X=x|Y=y\right)) \\
&= -\sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} \mathcal{P}\left(X=x, Y=y\right) \log_2(\mathcal{P}\left(X=x|Y=y\right)).
\end{aligned}$$

Notice that if X and Y are independent, $\mathcal{H}(X|Y) = \mathcal{H}(X)$.

Mutual information $\mathbf{I}(X,Y)$ is a measure of the information shared by random variables X and Y. It is defined as the difference between the entropy of X and the conditional entropy of X given Y:

$$\begin{aligned}
\mathbf{I}(X,Y) &= \mathcal{H}(X) - \mathcal{H}(X|Y) \\
&= \mathcal{H}(Y) - \mathcal{H}(Y|X) \\
&= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \mathcal{P}\left(X=x, Y=y\right) \log_2\left(\frac{\mathcal{P}\left(X=x, Y=y\right)}{\mathcal{P}\left(X=x\right)\mathcal{P}\left(Y=y\right)}\right).
\end{aligned}$$

Bibliography

[ABMD92] M Antonini, M Barlaud, P Mathieu, and I Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Proc*, 1:105–220, April 1992.

[AS72] M Abramowitz and C A Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, 9th printing edition, 1972.

[BA83] P J Burt and E H Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Comm.*, COM-31(4):532–540, April 1983.

[BCG90] A C Bovik, M Clark, and W Geisler. Multichannel texture anaysis using localized spatial filters. *IEEE Trans. PAMI*, 12(1):55–73, 1990.

[BHK97] P Belhumeur, J Hespanha, and D Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. PAMI*, 19(7):711–720, July 1997.

[Bon97] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Computer Graphics*, pages 361–368. ACM SIGGRAPH, 1997.

[Bro66] P Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, 1966.

[BS94] C A Bouman and M Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Trans. Image Proc*, 3(2), 1994.

[BS97] R W Buccigrossi and E P Simoncelli. Image compression via joint statistical characterization in the wavelet domain. Technical Report 414, GRASP Laboratory, University of Pennsylvania, May 1997.

[CJ83] G R Cross and A K Jain. Markov random field texture models. *IEEE Trans. PAMI*, 5(1):25–39, January 1983.

[CT91] T Cover and J Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[Der56] N G Deriugin. The power spectrum and the correlation function of the television signal. *Telecommunications*, 1(7):1–12, 1956.

[DLR77] A Dempster, N Laird, and D Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc B*, 39:1–38, 1977.

[FA91] W T Freeman and E H Adelson. The design and use of steerable filters. *IEEE Trans. PAMI*, 13(9):891–906, 1991.

[Fie94] D J Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[Fis36] R A Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.

[FvDFH96] J Foley, A van Dam, S Feiner, and J Hughes. *Computer Graphics: Principles and Practice*. Addison-Welsey Publishing, second edition, 1996.

[GBGP94] H Greenspan, S Belongie, R Goodman, and P Perona. Rotation invariant texture recognition using a steerable pyramid. In *ICPR*, volume B, pages 162–167, 1994.

[Har79] R M Haralick. Statistical and structural approach to texture. *Proc. IEEE*, 67:786–804, May 1979.

[HPB98] T Hofmann, J Puzicha, and J Buhmann. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Trans. PAMI*, 20(8):803–818, August 1998.

[HRD95] G E Hinton, M Revow, and P Dayan. *Advances in Neural Information Processing Systems 7*, chapter Recognizing Handwritten Digits Using Mixtures of Linear Models, pages 1015–1022. MIT Press, Cambrige MA, 1995.

[HS80] M Hassner and J Sklansky. The use of Markov random fields as models of texture. *Comp. Graphics Image Proc.*, 12:357–370, 1980.

[JF95] R L Joshi and T R Fischer. Image subband coding using arithmetic and trellis coded quantization. *IEEE Trans. Circuits and Systems for Video Technology*, 5(6):515–523, 1995.

[JJK+97] R L Joshi, H Jafarkhani, J H Kasner, T R Fischer, N Farvardin, M W Marcellin, and R H Bamberger. Comparison of different methods of classification in subband coding of images. *IEEE Trans. Image Proc*, 6(11):1473–1486, 1997.

[Jul75] B Julesz. Experiments in the visual perception of texture. *Scientific American*, 232(4):34–43, 1975.

[KC97] S Krishnamachari and R Chellappa. Multiresolution Gauss-Markov random field models for texture segmentation. *IEEE Trans Image Processing*, 6(2):251–267, February 1997.

[LRO97] S M LoPresto, K Ramchandran, and M T Orchard. Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization frameqork. In *Data Compression Conference*, March 1997.

[Mal89] Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. PAMI*, 11:674–693, July 1989.

[MP90] J Malik and P Perona. Preattentive texture discrimination with early vision mechanisms. *J. Optical Society of America*, 7(2):923–932, May 1990.

[OF96] B A Olshausen and D J Field. Natural image statistics and efficient coding. *Network: Computation in Neural Systems*, 7:333–339, 1996.

[Ph93] A P Pentland and B horowitz. *Digital Images and Human Vision*, chapter A Practical Approach to Fractal-Based Image Compression. MIT Press, 1993.

[PS92] A P Pentland and E P Simoncelli. Fractal-based image compression and interpolation, 1992. U.S. Patent Number 5,148,497, filed 14 Feb 1990, issued 15 Sep 1992.

[PS99] J Portilla and E P Simoncelli. Texture representation and synthesis using correlation of complex wavelet coefficient magnitudes. Technical Report 54, Consejo Superior de Investigaciones Cientificas (CSIC), Madrid, March 1999.

[PTVF92] W H Press, S A Teukolsky, W T Vetterling, and B P Flannery. *Numerical Recipes in C - Second Edition.* Cambridge University Press, Cambridge, MA, 1992.

[RC95] R Rinaldo and G Calvagno. Image coding by block prediction of multiresolution subimages. *IEEE Trans Image Processing*, July 1995.

[Rud94] D L Ruderman. The statistics of natural image. *Network: Computation in Neural Systems*, 5:517–548, 1994.

[RW90] T R Reed and H Wechsler. Segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations. *IEEE Trans. PAMI*, 12(1), 1990.

[SA96] E P Simoncelli and E H Adelson. Noise removal via Bayesian wavelet coring. In *Third Int'l Conf on Image Proc*, volume I, pages 379–382, Lausanne, September 1996. IEEE Sig Proc Society.

[SB97] E P Simoncelli and R W Buccigrossi. Embedded wavelet image compression based on a joint probability model. In *Fourth Int'l Conf on Image Proc*, volume I, pages 640–643, Santa Barbara, October 1997. IEEE Sig Proc Society.

[SBLM98] J Shi, S Belongie, T Leung, and J Malik. Image and video segmentation: The normalized cut framework. In *IEEE Int'l Conf on Image Processing*, Chicago, Illinois, 1998.

[SF95] E P Simoncelli and W T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Second Int'l Conf on Image Processing*, Washington, DC, October 1995.

[SFAH92] E P Simoncelli, W T Freeman, E H Adelson, and D J Heeger. Shiftable multiscale transforms. *IEEE Trans. Information Theory*, 38(2):587–607, March 1992. Special Issue on Wavelets.

[Sha93] Jerome Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans Signal Processing*, 41(12):3445–3462, December 1993.

[Sim88] E P Simoncelli. Orthogonal sub-band image transforms. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA, May 1988. Also available as MIT Media Laboratory Vision and Modeling Technical Report #100.

[Sim97] E P Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *31st Asilomar Conf on Signals, Systems and Computers*, Pacific Grove, CA, November 1997. IEEE Sig Proc Society. Invited presentation.

[SP96] A Said and W A Pearlman. An image multiresolution representation for lossless and lossy compression. *IEEE Trans. Image Proc*, 5(9), September 1996.

[SP98] E Simoncelli and J Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *IEEE Int'l Conf on Image Processing*, Chicago, IL, October 1998.

[SZB95] E Schwartz, A Zandi, and M Boliek. Implementation of compression with reversible embedded wavelets. In *Proc SPIE*, 1995.

[TPH95] A Teuner, O Pichler, and B J Hosticka. Unsupervised texture segmentation of images using tuned matched gabor filters. *IEEE Trans. IP*, 4(6):863–870, 1995.

[Uns95] M Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Processing*, 4(11):1549–1560, November 1995.

[Vet84] M Vetterli. Multidimensional subband coding: some theory and algorithms. *Signal Processing*, 6(2):97–112, 1984.

[WA96] Y Weiss and E H Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *IEEE CVPR*, pages 321–326, 1996.

[WC97] X Wu and J Chen. Context modeling and entropy coding of wavelet coefficients for image compression. In *ICASSP*, Munich, April 1997.

[WK97] H Wang and C J Kuo. A multi-threshold wavelet coder (MTWC) for high fidelity image compression. In *IEEE ICIP*, pages 652–655, 1997.

[ZWM96] S Zhu, Y Wu, and D Mumford. Filters, random fields and maximum entropy (FRAME) – towards the unified theory for texture modeling. In *IEEE Conf Computer Vision and Pattern Recognition*, June 1996.