Novel Statistical Multiresolution Techniques for Image Synthesis, Discrimination, and Recognition

by

Jeremy S. De Bonet

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

By treating images as samples from probabilistic distributions, the fundamental problems in vision – image similarity and object recognition – can be posed as statistical questions. Within this framework, the crux of visual understanding is to accurately characterize the underlying distribution from which each image was generated. Developing good approximations to such distributions is a difficult, and in the general case, unsolved problem.

A series of novel techniques is discussed for modeling images by attempting to approximate such distributions directly. These techniques provide the foundations for texture synthesis, texture discrimination, and general image classification systems.

Thesis Supervisor: Paul Viola
Title: Assistant Professor of Electrical Engineering and Computer Science

Acknowledgments

Greg Galperin. Through defending these ideas during conversations with him, both the justification and underlying rational became clear. At least to me.

Paul Viola. I would like to thank him for jointly playing both the role of advisor and and of co-investigator. Many of the ideas in this thesis are attempts to answer questions raised during our conversations.

The Microsoft Advanced Technology Vision Research Group.

iconoclast *lit. image destroyer*
one who attacks established beliefs or institutions

Contents

Chapter 1

Introduction

This thesis consists primarily of the description, discussion and analysis of three techniques for the approximation of statistical distributions over images.

Chapter 2 outlines a technique for treating input texture images as probability density estimators from which new textures, with similar appearance and structural properties, can be sampled. In a two-phase process, the input texture is first analyzed by measuring the joint occurrence of texture discrimination features at multiple resolutions. In the second phase, a new texture is synthesized by sampling successive spatial frequency bands from the input texture, conditioned on the similar joint occurrence of features at lower spatial frequencies. Textures synthesized with this method more successfully capture the characteristics of input textures than do previous techniques.

Chapter 3 describes a technique for measuring texture similarity based on a comparing distributions similar to those of Chapter 2, which captures the texture characteristics within images using an image representation which measures the joint occurrence of features across spatial resolutions. An image classification system is described which measures the likelihood that the distribution derived from one image could have generated another. Classification of natural textures indicates a high level of specificity, and recent results on target detection in SAR imagery are encouraging.

In Chapter 4 a new method is presented for producing a measure of the similarity between images using its visual structure. This method is used to build an image database system based on a ”query by image example” paradigm. A typical query consists of a small set of images, submitted by the user, which are representative of those for which he is searching. A characteristic signature is computed for the query images. These characteristic signatures consist of the responses of an extensive set of filter networks which form a textures-of-textures representation, and captures the visual structure within the images. By comparing the query image signatures to the recomputed signature for each image in the database, a similarity ranking is determined for each of the images in the database.

Chapter 5 describes the results of a series of experiments which were designed to measure the performance of the image database system presented in Chapter 4, and compare its performance to several other basic methods which are at the heart of other image retrieval techniques. In each experiment we measure the retrieval rates for a set of pictures which are truly similar, and vary only along an individual visual dimension. The target image sets consist of two classes of variation: sets which contain images generated from a single image which has been altered to varying degrees by one of several image manipulation routines; and sets of images of the same subject taken under some physical variation.

In the experiments in Chapter 5, the image database system presented in Chapter 4 significantly outperformed the other techniques with which it was compared. To attain an understanding of which components of the characteristic signatures used by the textures-of-textures / filter network tree retrieval method, are critical for achieving this performance level, we repeat the experiments performed in Chapter 5, for different textures-of-textures configurations. Experiments where performance successes were achieved with smaller configurations indicate which components of the characteristic signature are critical for attaining invariance to each of the types of visual variation in the target sets.

From the experiments in Chapter 5 and Chapter 6, we discover exactly which components of the characteristic signature are critical for achieving the observed retrieval performance levels. Further, where performance was less than satisfactory, additional improvments and extensions are suggested. In Chapter 7 we integrate all of these changes into the image database system, and examine its performance on the controlled-variation sets of target images, and on more general image database queries.

Chapter 2

Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images

Synthetic texture generation has been an increasingly active research area in computer graphics. The primary approach has been to develop specialized procedural models which emulate the generative process of the texture they are trying to mimic. For example, models based on reaction-diffusion interactions have been developed to simulate seashells [75] or animal skins [67].

More recently work has been done which considers textures as samples from probabilistic distributions. By determining the form of these distributions and sampling from them, new textures that are similar to the originals can, in principle, be generated. The success of these methods is dependent upon the structure of the probability density estimator used in the sampling procedure. Recently several attempts at developing such estimators have been successful in limited domains. Most notably Heeger and Bergen [32] iteratively resample random noise to coerce it into having particular multiresolution oriented energy histograms. Using a similar distribution, and a more rigorous resampling method Zhu *et al.* [77] have also achieved some success. In work by Luettgen, *et al.* [40] multiresolution Markov random fields are used to model relationships between spatial frequencies within texture images.

In human visual psychophysics research, much of texture perception studies has been focused on developing physiologically plausible models of texture discrimination. These models involve determining to which measurements of textural variations humans are most sensitive. Typically based on the responses of oriented filter banks, such models are capable of detecting variations between patches perceived by humans to be different textures ([4, 5, 6, 7, 12, 29, 33], for example.) The approach presented here uses these resulting psychophysical models to provide constraints on a statistical sampling procedure.

In a two-phase process, an input texture is first analyzed by computing the joint occurrence, across multiple resolutions, of several of the features used in psychophysical models. In the second phase, a new texture is synthesized by sampling successive spatial frequency bands from the input texture.

If this is done in a random way, then the visual structures which are characteristic of the input texture are lost. By conditioning on the similar joint occurrence of features at all lower spatial frequencies, we attempt to retain the important visual characteristics during synthesis.

The sampling methodology is based on the following hypothesis: if at some resolutions two regions in the input texture are indistinguishable, they can be rearranged at that resolution without changing the visual characteristics of the texture. In the procedure outlined here, we find such indistinguishable regions using multiresolution feature detectors. By rearranging textural components at locations and resolutions where the discriminability is below threshold, new texture samples are generated which have similar visual characteristics.

2.1 Motivation

The goal of probabilistic texture synthesis can be stated as follows: to generate a new image from an example texture, such that the new image is sufficiently different from the original yet still appears as though it was generated by the same underlying stochastic process as was the original texture.

If successful, the new image will differ from the original, yet have perceptually identical texture characteristics. This can be measured psychophysically in texture discrimination tests. To satisfy both criteria, a synthesized image should differ from the original in the same way as the original differs from itself.

From an input texture patch, such as that shown in Figure 2-1, there are infinitely many possible distributions which could be inferred as the generative process. The prior beliefs we have about the form of such generative processes will shape the distributions from which new textures will be sampled. Depending on the accuracy of these priors, the resulting textures may or may not, satisfy the above criteria for "good" synthesis.

One possible prior over the distribution of images is that the original texture is the only sample in the distribution, and that no other images are texturally similar. From this assumption simple tiling results as shown in Figure 2-2. Clearly this fails the "sufficiently different" criteria stated above.

Another feasible – though also clearly inadequate – prior is to assume that the pixels in the input texture are independently sampled from some distribution. Textures generated with this model do not capture the non-random structure within the original. The result of such an operation is shown in Figure 2-3. As expected it fails to capture the character of the original and is perceptually different. This is evidenced by the ease with which the original can be located when superimposed on the synthesized texture. This effect, commonly known as "popout," occurs because the textures are perceptually different and do not appear to have been generated by the same process ([6, 29, 33],e.g.).

The goal of texture synthesis is to generate a texture, such as that shown in Figure 2-5, which is both random, and indiscriminable from the original texture. Figure 2-5 satisfies these criteria in that it differs significantly from the original yet appears to have been generated by the same physical process. Because of the perceptual similarity between this texture, which was synthesized by the procedure in this paper, and the input texture (generated by some other process) it is difficult to locate the region which contains the superimposed

Figure 2-1: An example texture image for input to a texture synthesis process.

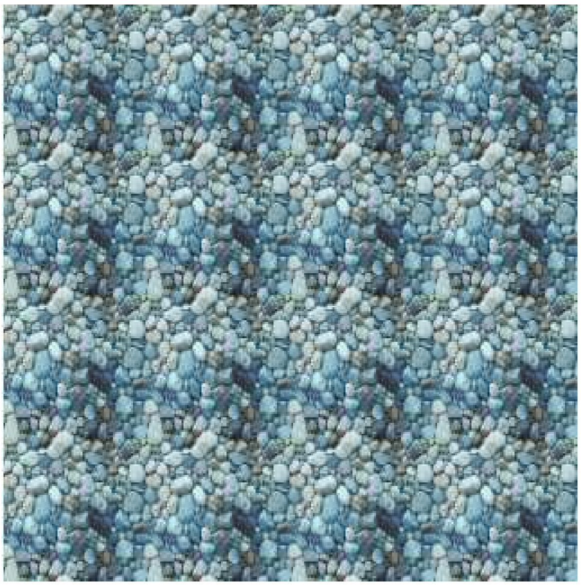


Figure 2-2: Simple repetition of the image does not result in a texture which appears to have come from the same stochastic distribution as the original.

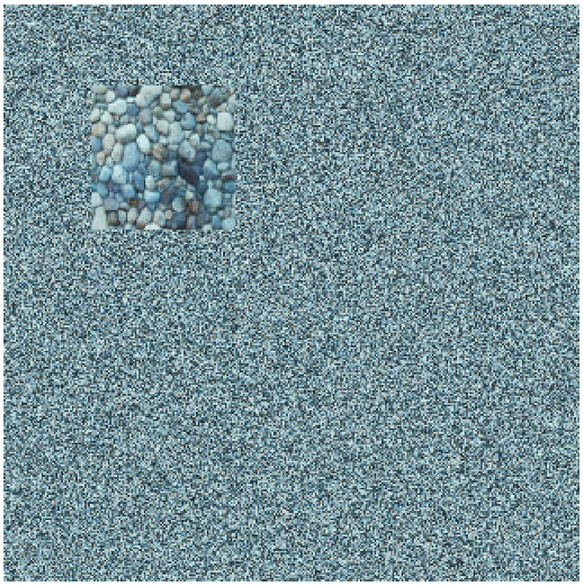


Figure 2-3: Textures that contain randomness not present in the original are perceptually different textures. This texture was generated by uniformly sampling the pixel values of the original. The original texture superimposed on the synthetic one is easily identified.

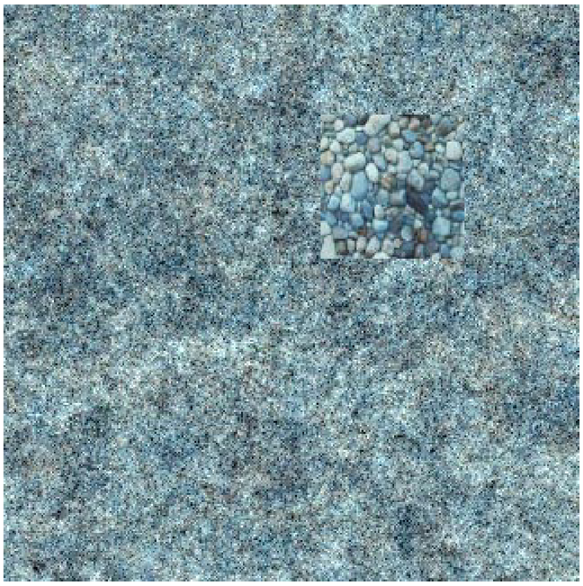


Figure 2-4: Sampling each spatial frequency band from the corresponding band in the original does not capture the detail which is characteristic of the input texture, indicating that relationships between frequencies is critical. The synthesized texture is different from the superimposed original texture, which is clearly discriminable.

Figure 2-5: The objective is to generate a patch such as the one above which is different from the original yet appears as though it could have been generated by the same underlying stochastic process. This texture, which was synthesized using the technique described in this paper, is perceptually very similar to the original, and the superimposed original is not readily located.

original.

2.2 Functional synthesis framework

Mathematically, the goal of texture synthesis is to develop a function, F, which takes a texture image, I_{input}, to a new texture sample, I_{synth}, such that the difference between I_{input} and I_{synth} is above some measure of visual difference from the original, yet is texturally similar. Formally,

$$F(I_{\text{input}}) = I_{\text{synth}} \tag{2.1}$$

subject to the constraints that

$$D^* \left(I_{\text{input}}, I_{\text{synth}}\right) < T_{\text{max disc}} \tag{2.2}$$

and

$$V^* \left(I_{\text{input}}, I_{\text{synth}}\right) > T_{\text{min diff}} \tag{2.3}$$

where D^* is a perceptual measure of the perceived difference of textural characteristics, and V^* a measure of the perceived visual difference between the input and synthesized images. The two functions D^* and V^* are hypothetical functions which represent the perceptual response of a human observer to input textures. In making D^* and V^* separate functions, we are explicitly modeling texture similarity as a different phenomenon from visual similarity. This corresponds to the notion that we can perceive two patches as the same texture, yet simultaneously be aware of their visual differences.

For a synthesized image to be acceptable, the perceived difference in textural characteristics must fall below a maximum texture discriminability threshold $T_{\text{max disc}}$, and the perceived visual difference must be above a minimum visual difference threshold, $T_{\text{min diff}}$. The success of a synthesis technique is measured by its ability to minimize D^* while maximizing V^*.

Human perception of texture differences, indicated by the hypothetical function D^*, depends on our prior beliefs about how textures should vary. These beliefs incorporate much of human visual experience; therefore, determining a computable metric, D, to approximate D^*, is a complex and often ill-defined task. Devising a good approximation for V^* is an even more difficult task. For texture synthesis purposes however, V^* can be roughly approximated by direct correlation.

The difficulty of determining a function D, to approximate D^*, depends on the structure and textual complexity of the two images. Many psychophysically based approximations have been proposed (for example, [7, 12, 69, 37] to list a few.)

Clearly, more complex textures can be represented in larger images; therefore, determining a discrimination function, say D_{small}, between images which have few pixels is less difficult than determining a similar function D_{large} over larger images.

Using a multiresolution approach, this work approximates D^* with a process which begins from low resolution images. By decomposing the function F into a set of functions F_i which each generate a single spatial frequency band of the new texture, I_{synth}. The domain of the each function F_i is a subset of the domain of F, as F_i's need only be a

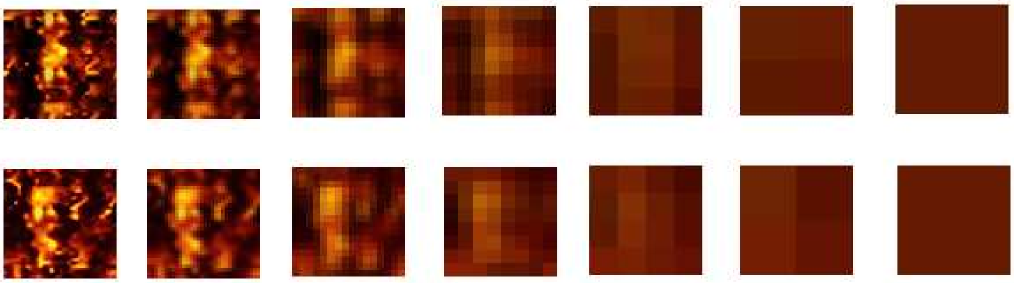


Figure 2-6: If two textures are similar (in a D^* sense), then downsampled versions of them look similar as well.

function of the information contained in the low spatial frequency bands of I_{input}.

This is intuitively shown by the following reasoning. If two textures are similar (in a D^* sense), then downsampled versions of them look similar as well. This is shown in the series of images in Figure 2-6; as resolution decreases, the textures continue to look similar. It would be very strange if this were not the case, as that would imply that two textures which are dissimilar could be made similar by adding high frequency information.

Consider taking an image, I_{input}, and generating from it I'_{input} by removing its high frequencies (by low pass filtering with a Gaussian kernel.) Given just I'_{input}, and without knowledge of the additional information in I_{input}, we could consider generating a new image I'_{synth} which is similar in textural appearance to I'_{input}. Attempting to achieve this is the same as our original problem of trying to synthesize I_{synth} from I_{input}, except with lower resolution images. Therefore, generating I'_{synth} which is similar to I'_{input} is independent of the highest frequency band of I_{input}. This argument can be repeated to show that I''_{synth} can be generated from I''_{input} without knowledge of I'_{input}, and so on. Thus to synthesize a low resolution version of a texture, the function F_i is dependent only on the spatial frequencies of the input at or below i:

$$L_i\left(I_{\text{synth}}\right) = F_i\left[L_i\left(I_{\text{input}}\right), L_{i+1}\left(I_{\text{input}}\right), \cdots, L_n\left(I_{\text{input}}\right)\right] \tag{2.4}$$

where $L_i\left(I_{\text{synth}}\right)$ is the i^{th} spatial frequency octave (or equivalently the i^{th} level of the Laplacian pyramid decomposition.)

The original function, F, in equation (2.1) is then constructed by combining the spatial frequency bands generated by F_0 through F_N. In doing this, we have decomposed the problem of synthesizing an image into a set of steps where we successively generate each higher resolution. The method presented here simplifies the difficulty of minimizing (approximate) D^* difference by initially synthesizing textures which are similar at low spatial frequencies, and then maintaining that similarity as it progresses to higher frequencies. A new texture is synthesized by generating each of its spatial frequency bands so that as higher frequency information is added textural similarity is preserved.

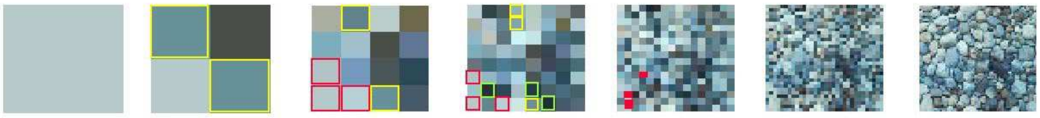


Figure 2-7: The synthesis procedure is based upon the hypothesis that at lower resolutions there are regions which are below some threshold of discriminability and that the randomness within a texture is in the locations of these regions.

2.3 Texture generation procedure

2.3.1 Hypothesis of texture structure

The sampling procedure used by this method is dependent upon the accuracy of the following hypothesis. Images perceived as textures contain regions which differ by less than some discrimination threshold, and randomization of these regions does not change the perceived characteristics of the texture. In other words, at some low resolution texture images contain regions whose difference measured by D^* is small, and reorganizing these low frequency regions, while retaining their high frequency detail will not change its textural (D^*) characteristics yet will increase its visual (V^*) difference.

In Figure 2-7, at each resolution examples of potentially interchangeable regions are highlighted. Rearranging the image at these resolutions and locations, while retaining their high resolution structure, corresponds to moving whole textural units (which in Figure 2-7 are individual pebbles.)

2.3.2 Analysis and Synthesis Pyramids

A new texture is synthesized by generating each of its spatial frequency bands so that as higher frequency information is added textural similarity is preserved. Each synthesized band is generated by sampling from the corresponding band in the input texture, constrained by the presence of local features. The general flow of this process is outlined in Figure 2-8.

In this first phase the input image is decomposed into an analysis pyramid, which exposes its multiresolution band-pass and feature response information.

To compute oriented filter responses at multiple resolutions, we first decompose the input image into a Gaussian pyramid, each level of which contains successively lower pass spatial frequency information in the input image. The original image forms the lowest level of the Gaussian pyramid; i.e. $G_0 = I$.

Each successive level of the pyramid is produced by convolution with a Gaussian kernel followed by downsampling by a factor of two:

$$G_{n+1} = 2\!\downarrow\!(G_n \otimes K_{Gaussian}) \qquad (2.5)$$

where $2\!\downarrow\!(\cdot)$ is the two-times downsampling operation and G_n is the n^{th} level of the pyramid, which is $1/2^n$ the size of the original in each dimension.

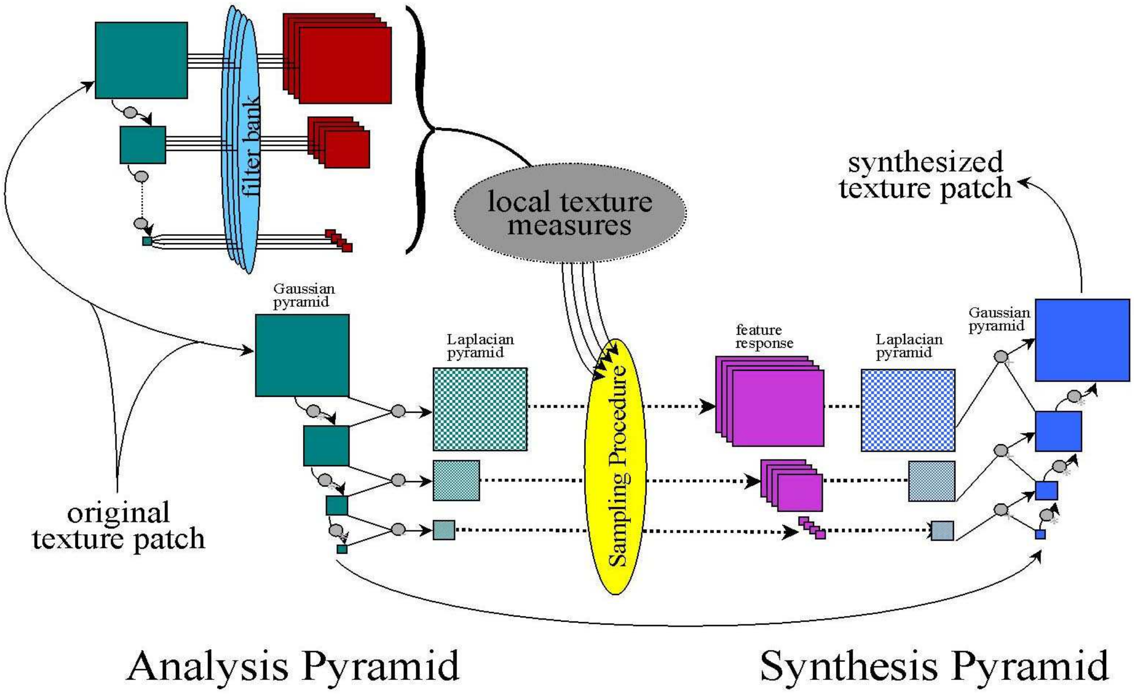


Figure 2-8: A schematic of the multiresolution sampling procedure. An input texture is decomposed into an analysis pyramid which contains both the information in the separate spatial frequency bands and the response of a filter bank at each resolution. By sampling from successively higher resolutions of this analysis pyramid, using the corresponding filter bank responses as constraints, a synthesis pyramid is generated. Combining the spatial frequency information from the synthesis pyramid yields the synthesized texture.

From every pair of levels G_i and G_{i+1} in the Gaussian pyramid, the spatial frequency information which is only in G_i and not in G_{i+1} is:

$$L_i(I) = (G_i(I) - 2\!\uparrow\![G_{i+1}(I)]) \tag{2.6}$$

where $2\!\uparrow\![\cdot]$ is a $2\times$ up-sampling operation.

In this way the spatial frequency information in the source image can be separated into separate bands L_i. This structure is commonly known as a Laplacian pyramid. Each level of the Laplacian pyramid contains the information from a one octave spatial frequency band of the input image. Because of this, the pyramid can be inverted [1] and the original image recovered. For a complete discussion of Laplacian and Gaussian pyramids, the reader is referred to the article by Burt and Adelson, [10], in which the Laplacian pyramid was introduced.

From each level of the analysis pyramid a corresponding level of a new pyramid is sampled. If this sampling is done independently at each resolution, as shown in Figure 2-4, the synthesized image fails to capture the visual organization characteristic of the original, indicating that the values chosen for a particular spatial frequency should depend on the values chosen at other spatial frequencies. We also infer that these values only depend on values at that and at lower spatial frequencies.

However, using only the Laplacian information in the lower frequency bands to constrain selection is also insufficient. Such a procedure which samples from a distribution conditioned exclusively on lower resolutions only loosely constrains the relationship between the 'child' nodes of different 'parents.' Sampling from such a distribution can result in high frequency artifacts which are not present in the intended distribution.

The one dimensional analog of this situation is shown in Figure 2-9. In this case the child can be slightly redder or bluer than its parent, allowing for children of the same parent to differ by at most some small hue shift. Because the distribution for values is conditioned only on the values of the parent, it is possible for elements which do not share a parent, but which are neighbors nevertheless, to differ by an arbitrarily large hue shift.

To prevent this, constraints must be propagated across children of different parents; however, constraint propagation on a two dimensional network results in dependency cycles. Sampling from distributions which contain such cycles, requires iterative procedures such as Gibbs sampling, or rejection sampling, which are not, in general, guaranteed to converge in finite time. In [77] Zhu *et al.* apply this direct approach, and even with convergence times on the order a day many synthesized textures fail to capture the critical visual characteristics. The current technique constrains the selection process within a spatial frequency band without creating cycles by using image features to constrain sampling.

In the one dimensional schematic these features, which measure the relationships between locations within a level, are indicated in Figure 2-10 by the diamonds which cover children of different parents.

Because the objective is to synthesize textures that contain the same textural characteristics as the original, yet vary from it in global form, it is assumed that global structure within the input texture is coincidental and should not constrain synthesis. Given this assumption

[1] To invert the Laplacian pyramid, the mean level of the image, G_n is also required.

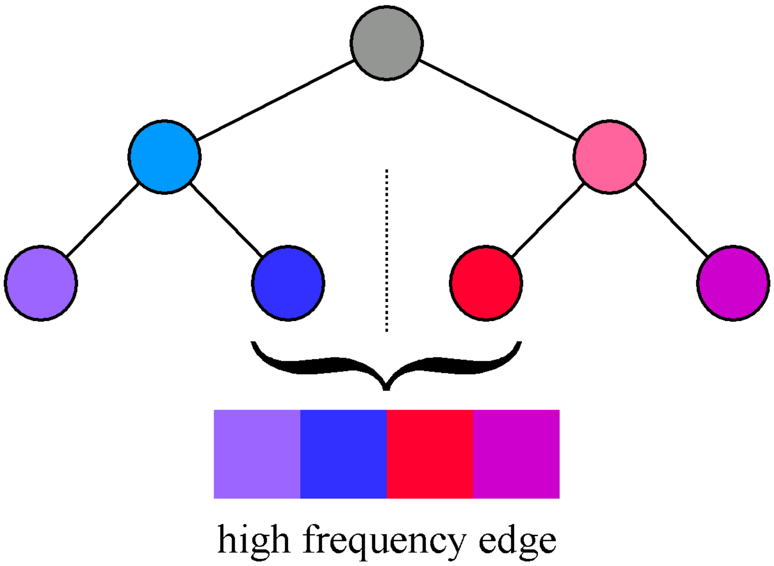


Figure 2-9: Conditioning sampling based upon only lower frequency information ignores dependencies between elements within a level, possibly resulting in the addition of high frequency edges not present in the original.

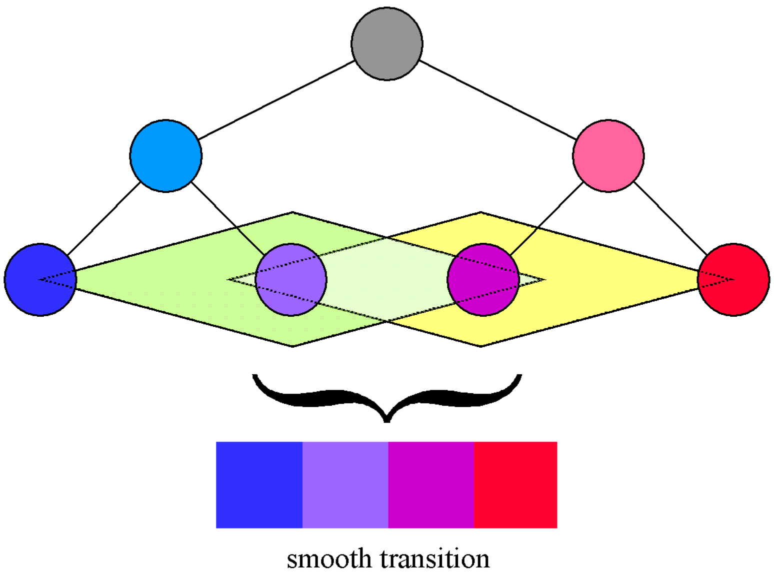


Figure 2-10: Using local features which measure relative responses between neighboring regions, allows the current model to propagate constraints across ”familial” boundaries. When sampling, these features provide constraints for the distribution of candidate values.

it is sufficient to use the responses of a set of *local* texture measures as features which provide the basis for an approximation to the human perceptual texture-discriminability function D^*. A filter bank of oriented first and second Gaussian derivatives – simple edge and line filters – were used in addition to Laplacian response. At each location (x, y) in the Gaussian part of the analysis pyramid at level i, the response of each feature j, is computed. When, at the lowest resolutions, the pyramid layers are too small, the features cannot be computed, and a constant value is used. These feature responses are used to further constrain the sampling procedure.

$$F_i^j(I, x, y) = \begin{cases} (G_i(I) \otimes f_j)(x, y) & \text{if size of } G_i(I) \geq f_j \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

At each location in the original image we construct a vector of the responses of each filter at each resolution. We call this the the *parent structure* of the region.

The constraints provided by this entire parent structure is stronger than that from just the "parent" Laplacian value, because they capture some of the relationships between pixels within a local neighborhood.

2.3.3 Sampling procedure

A "synthesis pyramid" is generated by sampling from the analysis pyramid conditioned on the joint occurrence of similar feature response values at multiple resolutions. When the synthesized pyramid has been completely generated, the band-pass information is combined to form the final synthesized texture.

Initially the top level – lowest resolution – of the analysis pyramid, which is a single pixel, is copied directly into the synthesis pyramid. When synthesizing a texture larger than the original, the top level of the synthesis pyramid is larger that in the analysis pyramid; in this case the analysis level is simply repeated to fill the synthesis level.

Subsequent levels of the synthesis pyramid are sampled from the corresponding level of the analysis pyramid. At each location in the synthesis pyramid, the local parent structure is used to constrain sampling. The parent structure, $\vec{S}_i$, of a location, (x, y), in image I, at resolution i, is a vector which contains the local response for features 1 through M, at every lower resolution from $i + 1$ to N:

$$S(x, y) = \begin{array}{l} \left\{ F_0^0(x, y), F_0^1(x, y), \cdots, F_0^M(x, y), \right. \\ F_1^0\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right), F_1^1\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right), \cdots, F_1^M\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right), \\ \cdots, \\ \left. F_N^0\left(\left\lfloor \frac{x}{2^N} \right\rfloor, \left\lfloor \frac{y}{2^N} \right\rfloor\right), F_N^1\left(\left\lfloor \frac{x}{2^N} \right\rfloor, \left\lfloor \frac{y}{2^N} \right\rfloor\right), \cdots, F_N^M\left(\left\lfloor \frac{x}{2^N} \right\rfloor, \left\lfloor \frac{y}{2^N} \right\rfloor\right) \right\} \end{array} \tag{2.8}$$

The parent structure of a location in a synthesis pyramid is depicted in Figure 2-11; in this schematic, each cell represents the set of local feature responses.

Two locations are considered indistinguishable if the square difference between every component of their parent structures is below some threshold. For a given location (x', y') in the synthesis image, I_{synth}, the set of all such locations in the input image can be com-

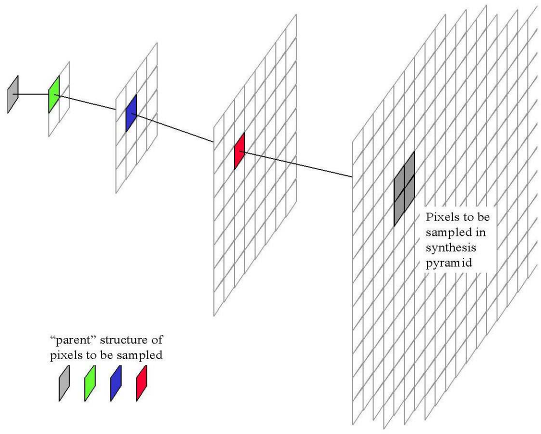


Figure 2-11: The distribution from which pixels in the synthesis pyramid are sampled is conditioned on the "parent" structure of those pixels. Each element of the parent structure contains a vector of the feature measurements at that location and scale.

puted:

$$\mathcal{C}_i\left(x', y'\right) = \left\{ (x, y) \middle| D \left(\begin{array}{c} \vec{S}_i\left(I_{\text{synth}}, x', y'\right), \\ \vec{S}_i\left(I_{\text{input}}, x, y\right) \end{array} \right) \leq \vec{T}_i \right\} \tag{2.9}$$

Where the distance function D, between two parent structures u and v, is given by:

$$D\left[u, v\right] = \frac{\left(u - v\right)^T \left(u - v\right)}{Z} \tag{2.10}$$

where Z is a normalization constant which eliminates the effect of contrast, equal to $\sum_{x,y} \vec{S}_i\left(I_{\text{input}}, x, y\right)$.

To be a member of set $C_i\left(x', y'\right)$ the distance between each component of the parent structures must be less than the corresponding component in a vector of thresholds for each resolution and feature:

$$\begin{array}{rl} \vec{T}_i = & [\ \ T_{i+1}^0\, T_{i+1}^1\ \cdots\ T_{i+1}^M \\ & \ \ \ \ T_{i+2}^0\, T_{i+2}^1\ \cdots\ T_{i+2}^M \\ & \ \ \ \ \cdots \\ & \ \ \ \ T_N^0\, T_N^1\ \cdots\ T_N^M]^T \end{array} \tag{2.11}$$

Where each element T_i^j is a threshold for the j^{th} filter response at the i^{th} resolution.

The values for new locations in the synthesis pyramid are sampled uniformly from among all regions in the analysis pyramid that have a parent structure which satisfies equation (2.9). This yields a probability distribution, over setting a new location in the synthesis pyramid to each location in the analysis pyramid which satisfies the conditions based on the joint occurrence of features at lower spatial frequencies:

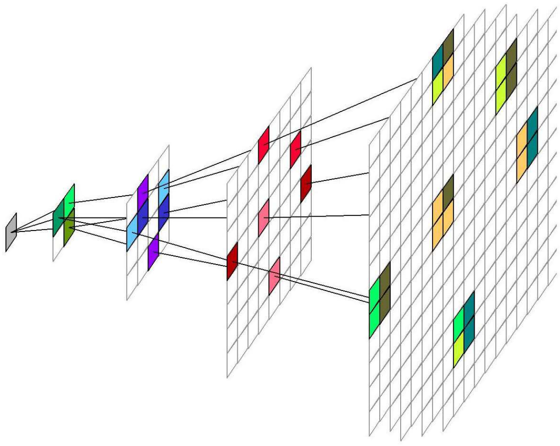


Figure 2-12: An input texture is decomposed to form an analysis pyramid, from which a new synthesis pyramid is sampled, conditioned on local features within the pyramids. A filter bank of local texture measures, based on psychophysical models, are used as features.

$$P\left(L_i\left(I_{\text{synth}}, x', y'\right) \Rightarrow L_i\left(I_{\text{input}}, x, y\right) \Big| (x, y) \in \mathcal{C}_i\left(x', y'\right)\right) \\ = \quad 1 \,/\, \|\mathcal{C}_i\left(x', y'\right)\| \tag{2.12}$$

Variations between the analysis and synthesis pyramids occur when multiple regions in the analysis pyramid satisfy the above criterion. The parent structure of such a group of candidate locations is depicted in Figure 2-12. As the thresholds increase, the number of candidates from which the values in the synthesis pyramid will be sampled, increases. The levels of the thresholds, T_i^j, mediate the rearrangement of spatial frequency information within the synthesized texture, and encapsulate a prior belief about the degree of randomness in the true distribution from which the input texture was generated.

Algorithmically, this sampling procedure can be described with the pseudo-code:

```
SynthesizePyramid
Loop i from top_level-1 downto 0
  Loop (x',y') over Pyr_synth[level i]
    C = ∅
    Loop (x,y) over Pyr_analysis[level i]
      C = C ∪ {(x,y)}
      Loop v from top_level downto i+1
        Loop j for each feature
          if D ( Pyr_analysis[v][j](x/2^(v-i), y/2^(v-i)),
                 Pyr_synth[v][j](x'/2^(v-i), y'/2^(v-i)) )
            < threshold[level v][feature j]
          then
            C = C - {(x,y)}
            break to next (x,y)

    selection = UniformRandom[0, ||C||]
    (x,y) = C[selection]
    Pyr_synth[v](x',y') = Pyr_analysis[v](x,y)
```

With more complex code, additional efficiency can be obtained by skipping whole regions which share a parent structure element that is above threshold difference.

Upon the completion of this sampling process for each level of the synthesis pyramid, the Laplacian portion of this pyramid is inverted to form the new texture by expanding each level to full size and summing. [2]

Though each band is sampled directly from the input image, the image which results from the recombination of each of these synthesized layers contains pixel values (i.e. RGB colors) not present in the original, because non-zero thresholds allow synthesized spatial frequency hierarchies which differ from those in the original.

Similarly, during this procedure it is possible to synthesize pyramids which correspond to images which cannot be displayed as the values are outside of the $[0, 255]$ displayable range. When this occurs, the pyramid is collapsed using full precision images, then values above or below the range of legal pixel values are "clipped" by replacing them with the closest legal value. Alternative solutions are to scale the color range of the entire image to fit within the display gamut, which decrease the overall contrast of the synthesized image, or to combine the two approaches using a saturating non-linearity (such as a sigmoid.) In practice however, the clipped regions do not detract from the overall appearance of the synthesized images.

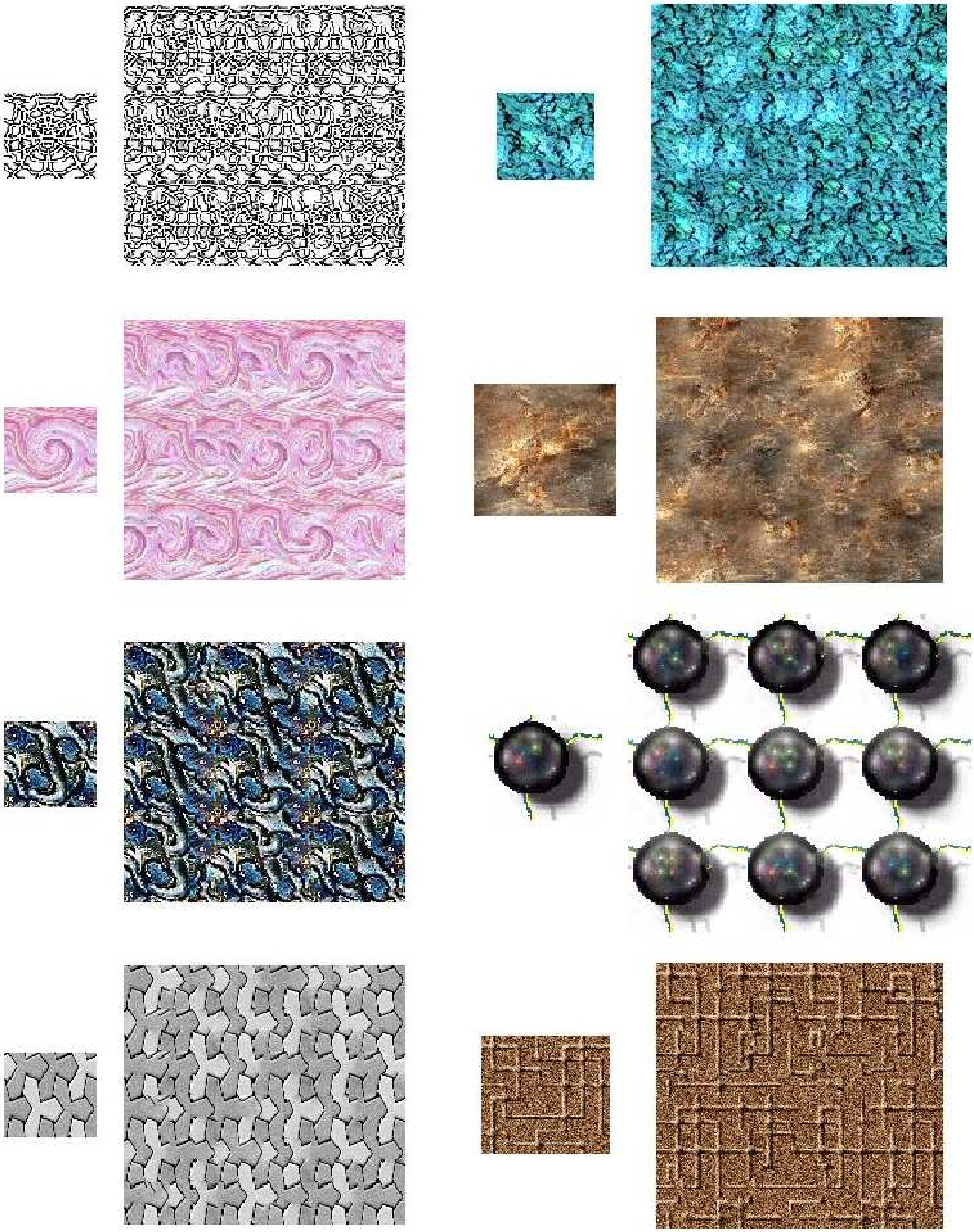


Figure 2-13: Texture synthesis results. The smaller patches are the input textures, and to their right are synthesized images which are 4 or 9 times larger.

2.4 Examples of texture synthesis

For 800 full color input textures, we synthesized new textures, each four times larger than the original. Some typical results are shown in Figure 2-13. The results from these examples are indicative of the synthesis performance on the entire set and were chosen only because they reproduce well on paper. The results of all 800 textures are available on the world wide web via the URL:

`http://www.ai.mit.edu/~jsd/Research/TextureSynthesis`

In the synthesis examples through out this paper thresholds of the form:

$$T_i^j = \alpha / i^\beta \tag{2.13}$$

were used with $\alpha \in [0, 0.4]$ and $\beta \in \{0, 1\}$. The parameter α establishes the prior belief about the sensitivity of D^*, the threshold $T_{\text{max disc}}$ in equation (2.2); larger β incorporates the belief that the 'true' distribution which generated the input texture is spatially homogeneous, and that the low frequency structure within the input image should not be an influential factor in region discrimination.

Shown in Figure 2-14 are a series of synthesized textures for $\beta = 0$ and $\alpha \in \{0.05, 0.10, \ldots, 0.30\}$. As the threshold increases, progressively more locations in the original become indistinguishable, and the amount of variation from the original increases. For this texture, the synthesized image which balances sufficient difference from the original with perceptual similarity, lies somewhere between $\alpha = .15$ and $\alpha = .20$ (images d-e.) For different images, the ideal threshold is different, reflecting our prior belief about the randomness implied by the original. Another synthesis series for a different input image is shown in Figure 2-15. In this case $\beta = 1$, a varies over the same range, and the ideal threshold is somewhere around $\alpha = 0.25$ (image f.)

2.5 Comparison to Heeger and Bergen model (1995)

In [32] Heeger and Bergen propose an iterative texture synthesis procedure which uses histogram matching to coerce a random image into texture similar to the input.

The fundamental approach taken by this technique is to match the histograms from each subbands of the random image to the corresponding input texture histogram. The subbands which they use are derived from the steerable pyramid decomposition of the images [60], and serve the essentially the same function as does the filter bank used in this work. Steerable pyramids have an additional steerability property, which allow the response of a filter at a non-basis orientation to be computed from the response of several basis-orientation filters. However, this property is not used by the Heeger and Bergen model.

By iteratively applying the matching procedure to each of the subband histograms, then to the pixel histogram, this technique gradually shifts the energy distribution in the random

[2] As discussed in [10], Laplacian pyramid inversion can be done more efficiently by a recursive procedure of expanding the lowest level and adding it to the next.

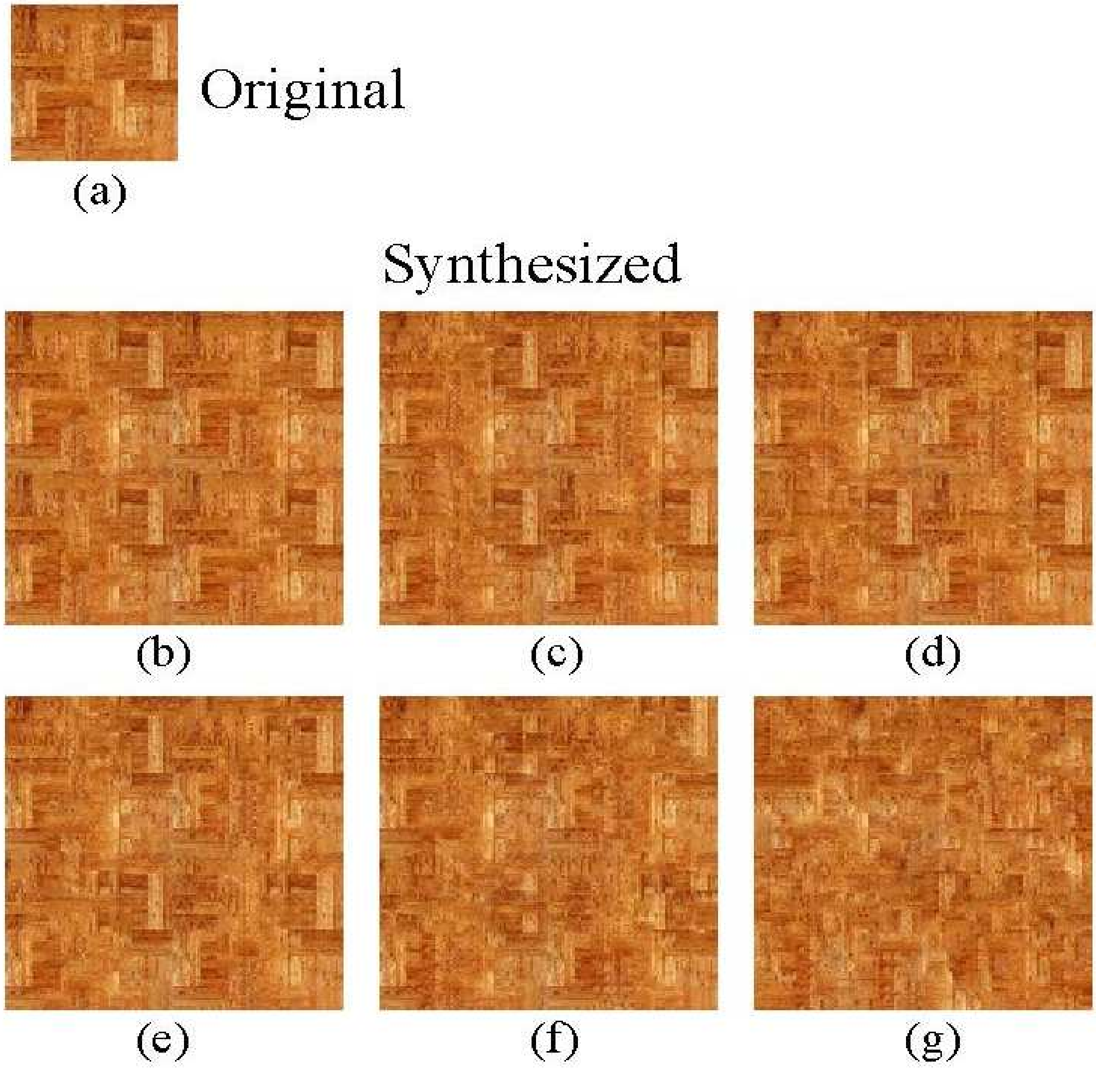


Figure 2-14: This series of 6 images (b-g) was generated from the original (a). For each a single threshold is used for all features and resolutions. Thresholds increase from 0.05 to 0.3 from (b) to (g).

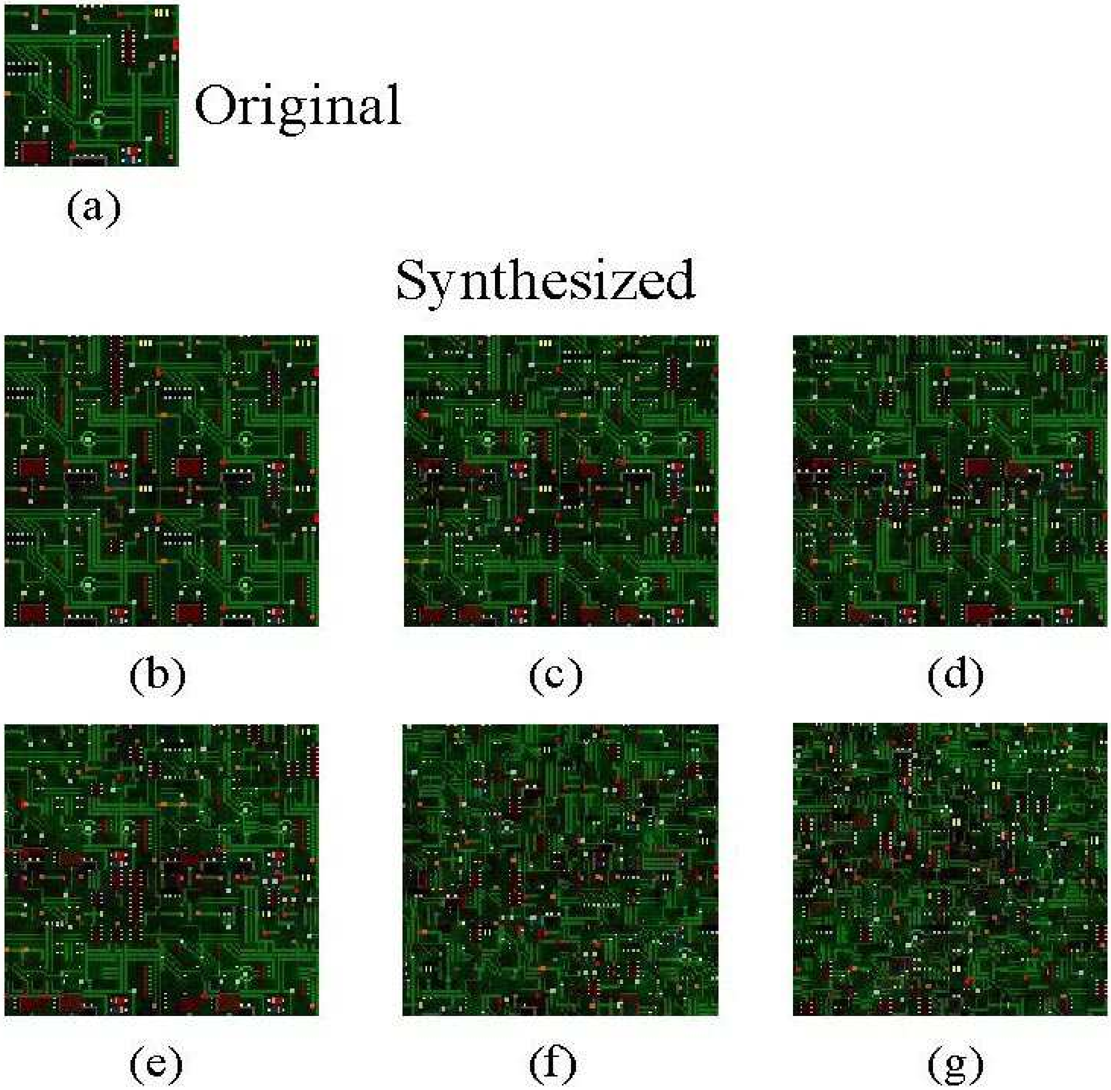


Figure 2-15: A series of synthesized textures for which the thresholds are inversely proportional to the spatial frequency and proportional to 0.05 in (b) to 0.3 in (g).

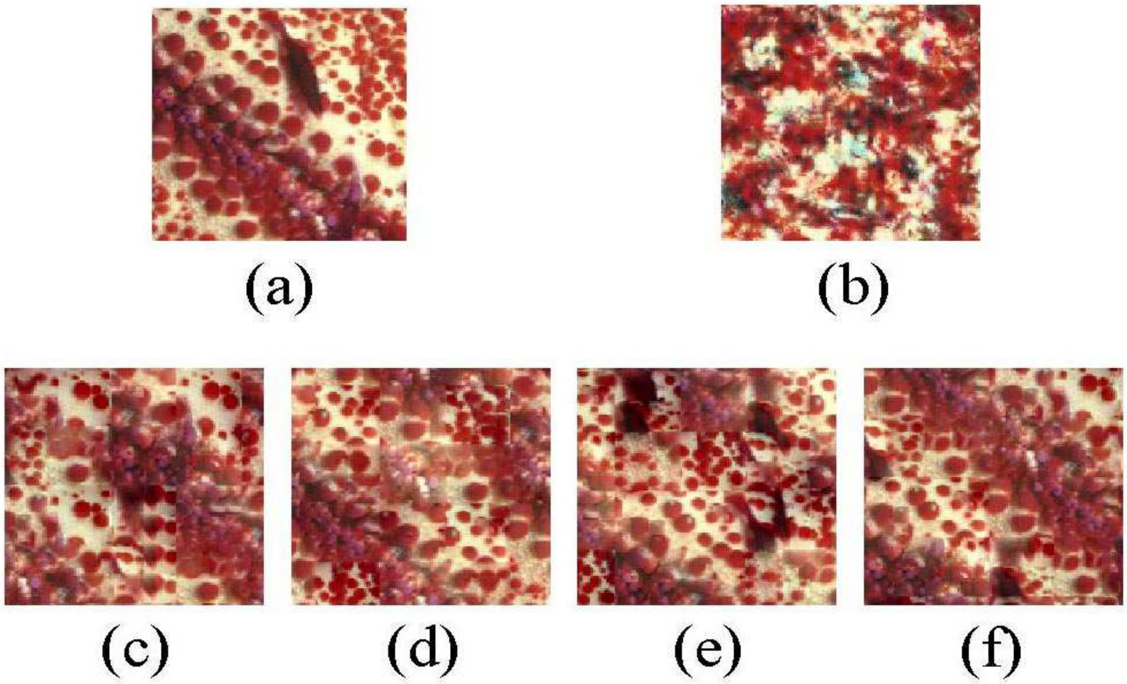


Figure 2-16: An input texture (a) which is beyond the limitations of Heeger and Bergen (1995) model (b), can be used successfully by this techniques to synthesize many new images. Four such synthesized images, using the same set of thresholds, are shown in (c) - (f).

image's histograms toward those of the input texture. After about five iterations, the operation tends to converge; however, they state that with additional iterations, artifacts due to reconstruction error are introduced.

In Figure 2-16(a) an input texture which is beyond the limitations of Heeger and Bergen model is shown. Synthesis from this input image using their technique, results in textures such as that shown in (b), clearly this image does not have the same visual characteristics as the original. Using the technique presented here, images (c) through (f) are obtained.

Even though the present model and the Heeger and Bergen model use effectively the same feature set, the present model can capture visual characteristics which are beyond the limitations of the other.

In performing the match-histogram operation, Heeger and Bergen force the energy distribution in each subband of the random image to match that of the input image. However, in doing so they do not explicitly constrain the energy distribution *across features or resolutions.* Even though two images may have the same subband histograms, they could have very different appearances because the joint occurrence of this energy does not match the original. This effect can be seen trivially via the following thought experiment. Given an input image which contains texture which is spatially homogeneous it is easy to see that the match-histogram procedure will accept an output image which is spatially inhomogeneous. Suppose we divided the energy in each subband into four groups, and forced each quadrant of the random image to match the energy in one of these groups. The complete histograms for each subband will match the histograms of the original however, clearly the reconstructed texture will be spatially inhomogeneous.

To overcome this effect Heeger and Bergen iterate over matching the subband histograms and matching the pixel-value histograms. However, doing this only loosely places constraints across features.

In the present model, the energy distribution across features or resolutions is explicitly required to match, by enforcing the satisfaction of equation (2.9). Because of requirement, regions in the synthesis pyramid are required to have similar energies for all features at all determined resolutions. Incorporation of these joint constraints captures visual characteristics which are critical in the overall perceived appearance of the synthesized texture.

In Figures 2-17 through 2-19 the three textures synthesized with the Heeger and Bergen model are shown. These images were chosen because they have textures which have randomness at different scales. The texture in Figure 2-17 is a standard Brodatz[3] texture [8]. The textures in Figures 2-18 and 2-19 are examples from Heeger and Bergen (1995), which they considered to be textures that can be successfully synthesized with their technique.

In each figure two examples of matched subband histogram pairs are also shown. In each pair the left panel shows the histogram for the subband in the input image, and the right the corresponding subband in the synthesized image. Though only two histograms are shown they are indicative of the quality of the match in all subbands at all resolutions. For every texture the subband histograms match quite well, yet the D^* difference between the synthesized and original images images is high — the textures look different. The fact that the results can be poor when the constraints are fully satisfied indicates that the constraints provided by the histogram matching function are insufficient. Or equivalently, these constraints provide a poor estimate of D^* texture discriminability. Because of this, when the procedure converges to a minimum histogram-match error it does necessarily reach the minimum D^* difference.

In Figure 2-20 textures synthesized using the present method from the same input images as Figures 2-17 through 2-19. Across each row one input texture is used to synthesize new textures using thresholds of $\alpha = \{0.1, 0.2, 0.4\}$. Because this model incorporates joint feature occurrence constraints, it is better able to capture the characteristics of the original image.

For the top image, there are many image locations which are very similar to one another, therefore at the lowest threshold, the synthesized image achieves a high V^* difference; furthermore it simultaneously achieves a low D^* difference. Because of this, the results of this synthesis are better than those in Figure 2-17. With higher threshold levels V^* difference continues to increase, but D^* difference also seems to grow as well.

The middle image is less self-similar, and as a result, the synthesized image achieves the best high V^* / low D^* difference at the second threshold level. With lower thresholds, the V^* difference is too low, and with higher thresholds the D^* is too high. All three synthesized images, but especially the middle one, are more successful than those in Figure 2-18 at replicating the texture.

[3]Textures taken from the book *Textures: a Photographic Album for Artists and Designers*, photographed by Philip Brodatz have become a de facto standard in the texture processing literature

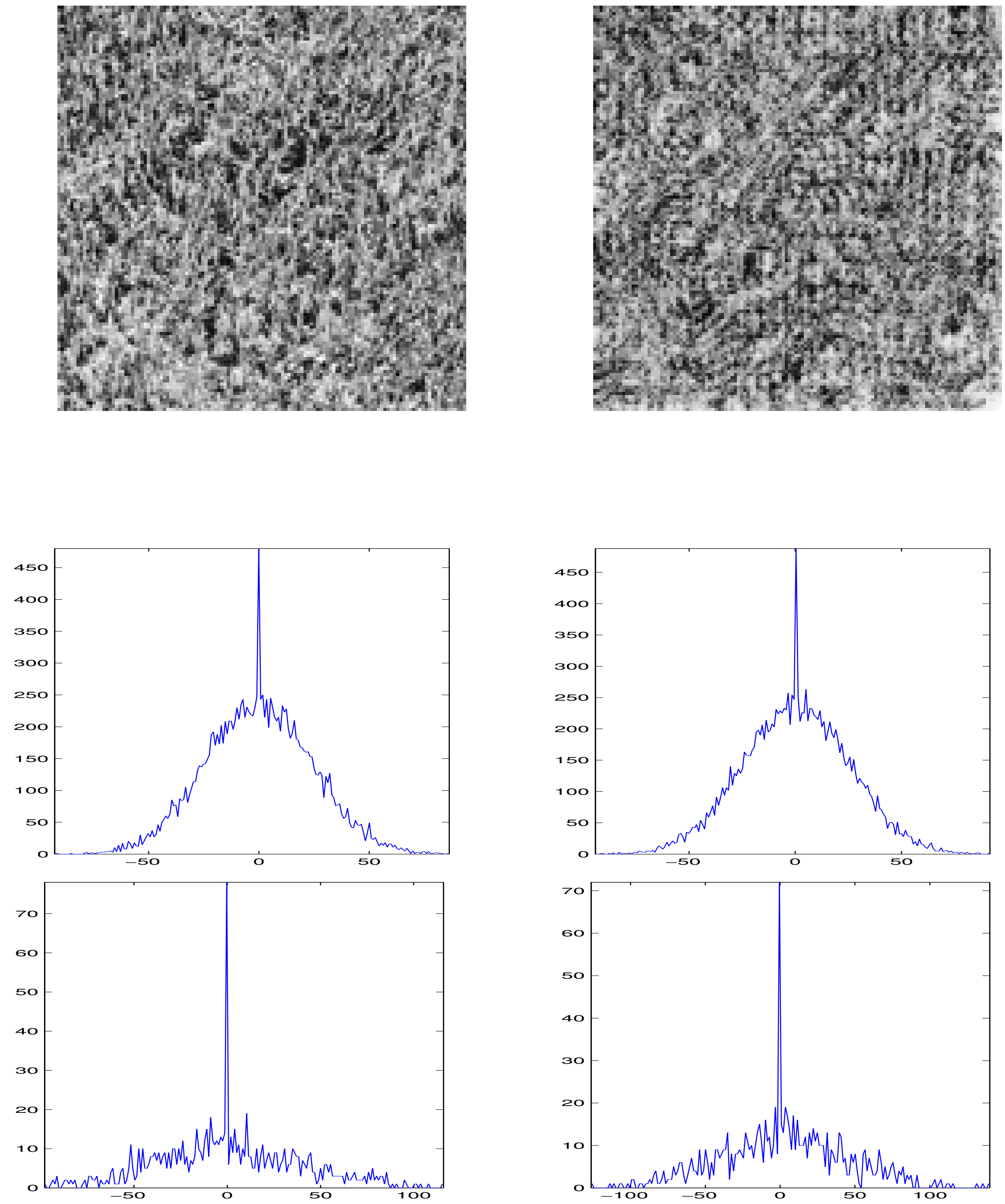
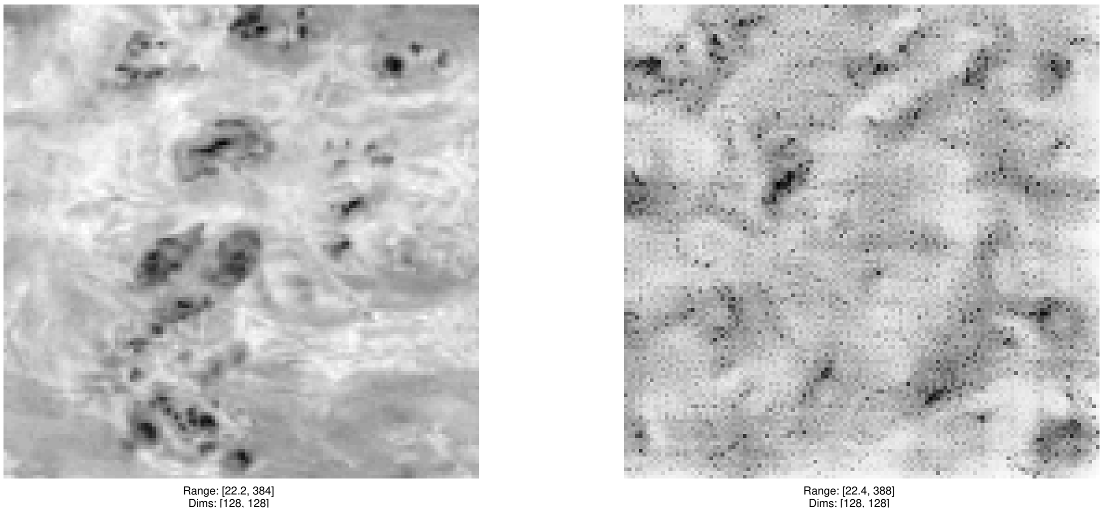


Figure 2-17:
LEFT IMAGE: A Brodatz [8] texture which has been used as input.
RIGHT IMAGE: The synthesized image using the Heeger and Bergen procedure.
LEFT GRAPHSS: The histogram for two example subbands in the input image.
RIGHT GRAPHS: The corresponding subband histogram in the synthesized image.



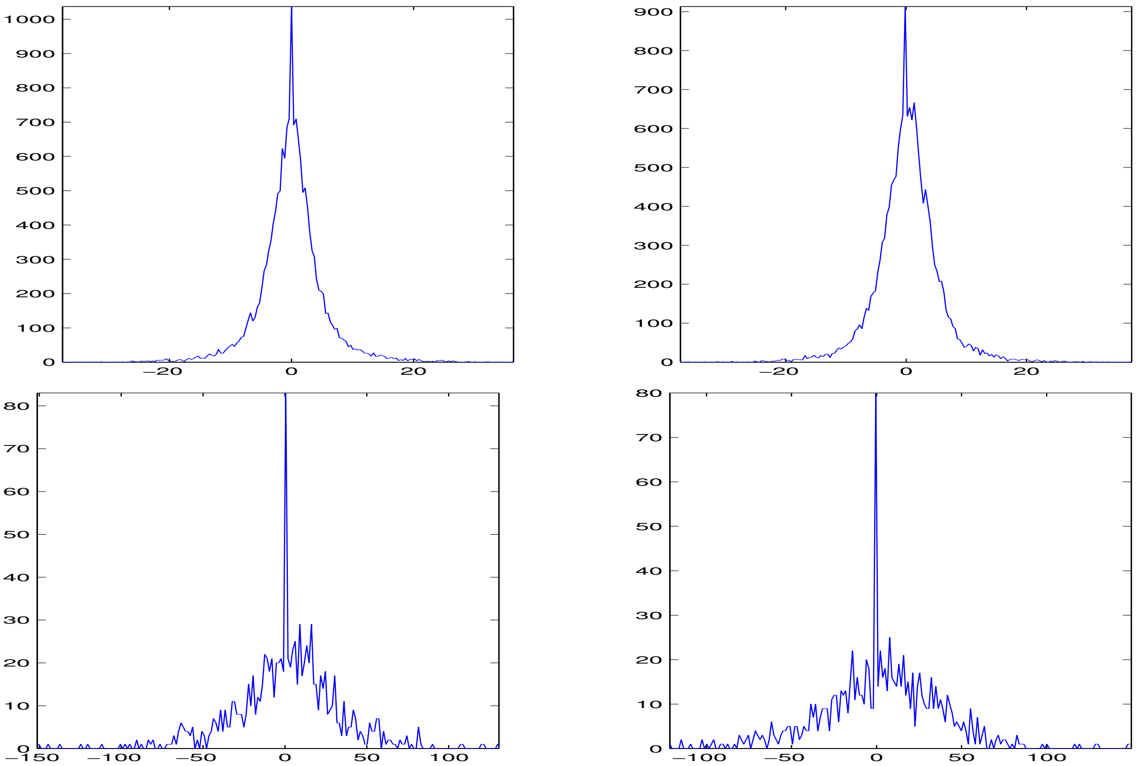


Figure 2-18:
LEFT IMAGE: A texture from Heeger and Bergen (1995) which they claim can be successfully used for synthesis with their model.
RIGHT IMAGE: The synthesized image using their procedure.
LEFT GRAPHSS: The histogram for two example subbands in the input image.
RIGHT GRAPHS: The corresponding subband histogram in the synthesized image.

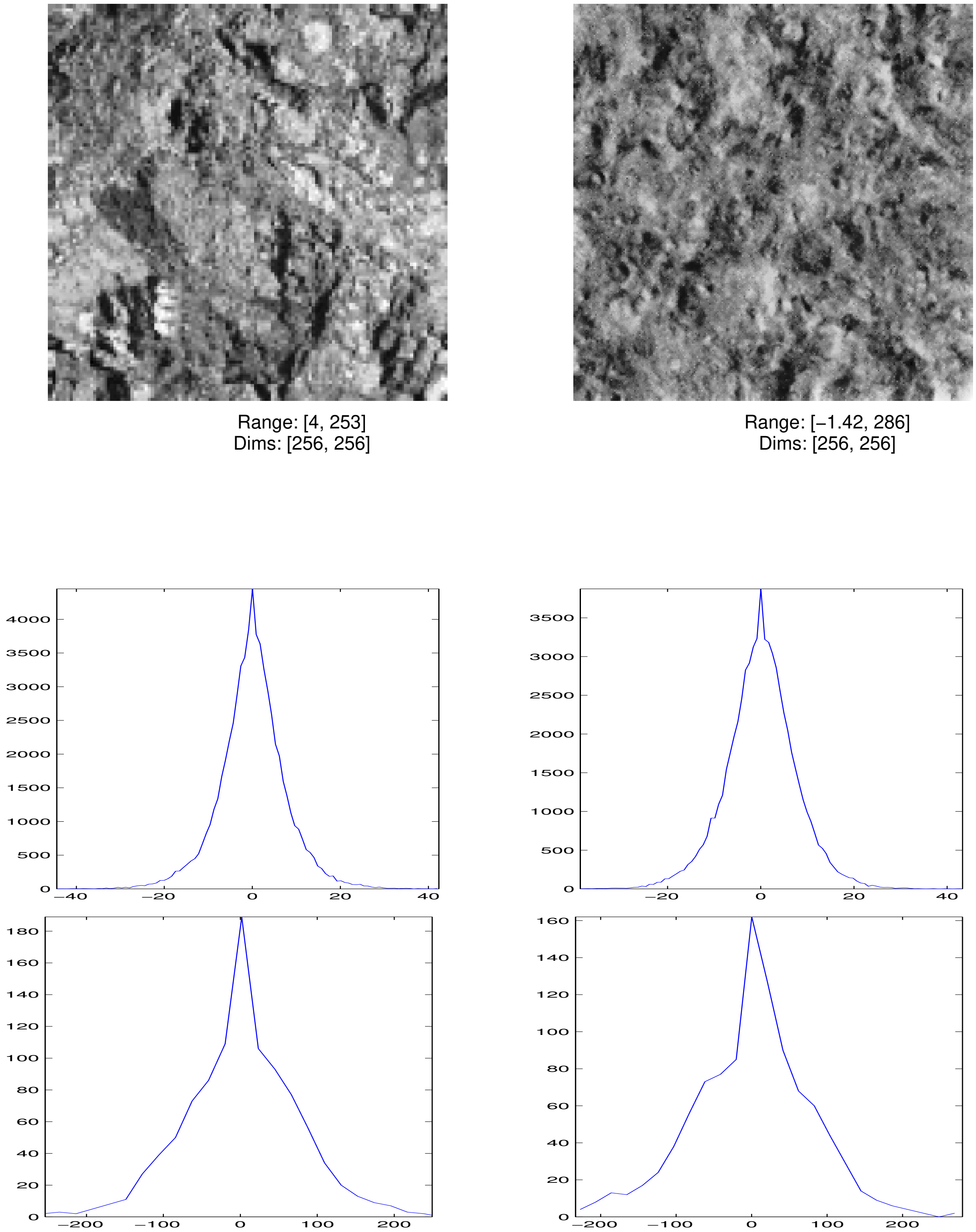


Figure 2-19:
LEFT IMAGE: A second texture from Heeger and Bergen (1995) which they claim can be successfully used for synthesis with their model.
RIGHT IMAGE: The synthesized image using their procedure.
LEFT GRAPHSS: The histogram for two example subbands in the input image.
RIGHT GRAPHS: The corresponding subband histogram in the synthesized image.

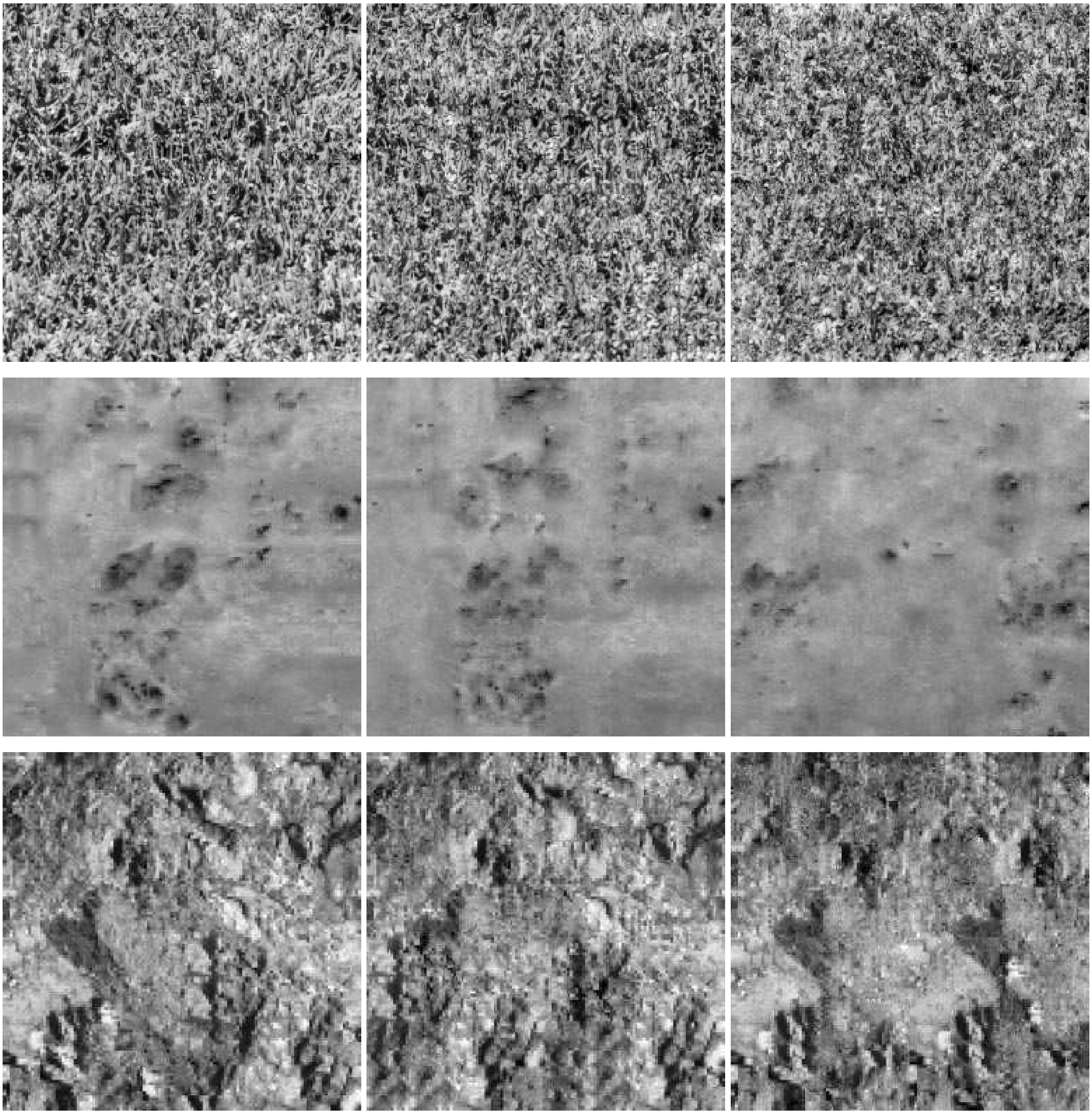


Figure 2-20: Textures synthesized using the present method which incorporates joint feature occurrence constraints. Across each row one input texture is used to synthesize new textures using thresholds of 0.075, 0.15 and 0.225.

The bottom image image is far less self-similar than the ones above it. As a result, the best high V^* / low D^* differences occur at the highest threshold; with lower thresholds, the V^* difference is too low. Because regions within the image are not inherently indistinguishable (to the feature set used), the effect of raising the threshold to enforce randomness is to cause the rearrangement of structures at a much coarser level than in the previous two synthesis. This occurs because the low frequency information is more self-similar than is the high frequency; therefore, low frequency rearrangements occur at lower thresholds. The high frequency detail is mostly rearranged due to the low frequency rearrangements. As a result, large textural units [4] are rearranged to synthesize the texture. These large units can be identified in both the input and synthesized images. Again, this result is more successful than those in Figure 2-19 at replicating the texture.

2.6 Limitations

The model presented here for estimating a generative probability density uses constraints based on joint occurrence of feature responses at multiple resolutions to capture the characteristics of the original image. However, this set of constraints does not capture all visual characteristics.

Because the constraints are local, the estimator presented here cannot model texture images with complex visual structures. Such structures include: reflective and rotational symmetry; progressive variations in size, color, orientation, etc.; and visual elements with internal semantic meaning (such as symbols) or which have meaning in their relative positions (such as letters.)

Simply adding additional complex features to attempt to capture these sorts of visual structures over conditions the sampling procedure, and simple tiling results. If appropriate thresholds could be determined through additional analysis of the input image, the effects of complex features could be mediated, and they might provide useful constraints.

Because it samples exclusively from the input image, this model assumes that the 'true' distributions from which each spatial frequency band in the input was generated, can be accurately approximated by only those values present in that image. If there were a model for the probability of values not present in the original, synthesized textures could possibly be generated which contain additional variation from the original which does not increase texture (D^*) difference yet increases the visual (V^*) difference.

2.7 Conclusion

We have presented a method for synthesis of a novel image from an input texture by generating and sampling from a distribution. This multiresolution technique is capable of capturing much of the important visual structure in the perceptual characteristics of many texture images; including artificial (man-made) textures and more natural ones, as shown in Figure 2-21. The input texture is treated as probability density estimator by using the

[4] This is very related to the notion of "texton" [33], which is the fundamental unit of a texture.

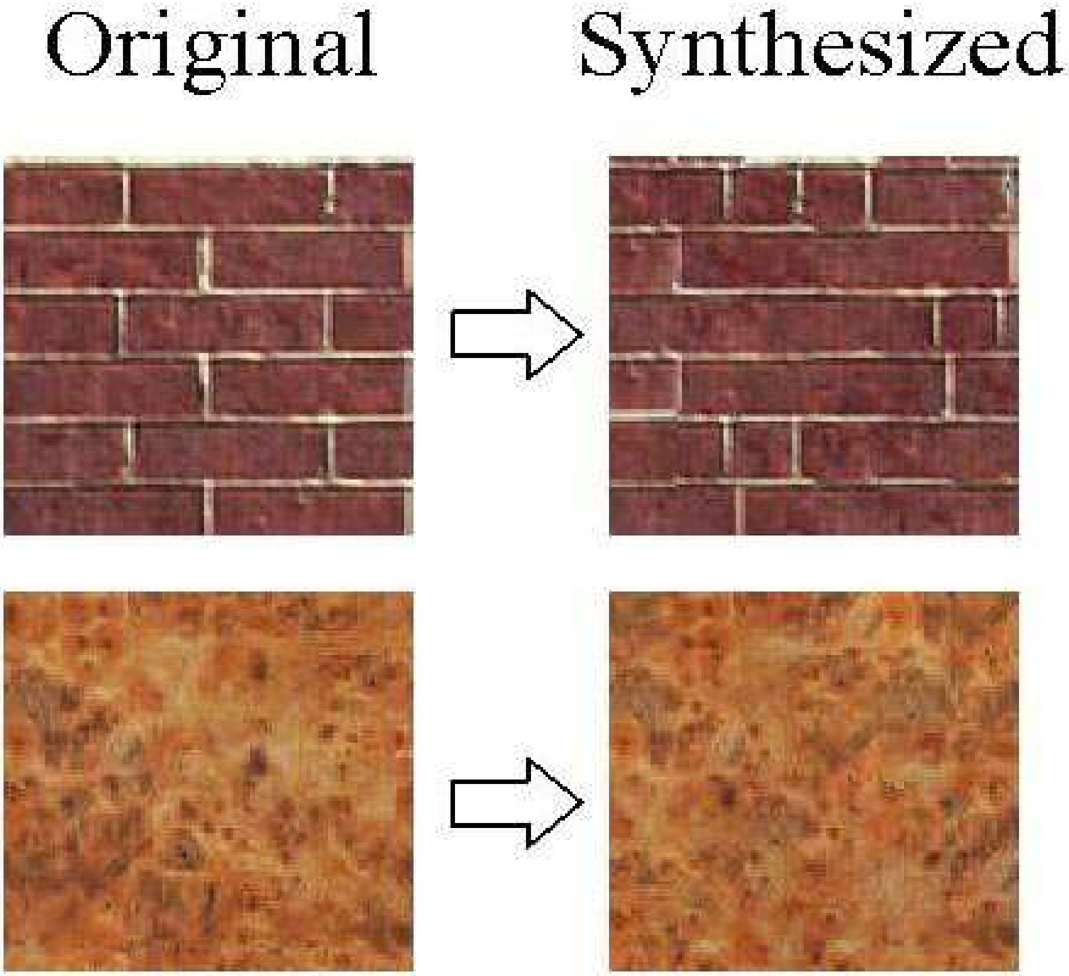


Figure 2-21: The characteristics of both man-made and natural textures can be captured and replicated with this process.

joint occurrence of features across multiple resolutions to constrain sampling. Prior beliefs about the 'true' randomness in the input are incorporated into the model through the settings of thresholds which control the level of constraint provided by each feature. Many of the textures generated by sampling from this estimator can simultaneously satisfy two the two criteria of successful texture synthesis: the synthesized textures are sufficiently different from the original, and appear to have been created by the same underlying generative process. These textures can be synthesized from more intricate input examples, and produce textures which appear more akin to the originals, than those produced by earlier techniques (e.g. Figure 2-16.)

Chapter 3

Flexible Histograms: Multiresolution Texture Discrimination Model

3.1 Overview

The fundamental problem in computer vision, image recognition, requires the integration of information from visual cues which discriminate between different objects and scenes. In this chapter we describe a technique for using the joint occurrence of local features at multiple resolutions to build a model to measure the likelihood that images contain the same textures.

In human visual psychophysics there is a long and rich history of models which attempt to replicate the ability of human observes to discriminate between patches of texture with often subtly different visual characteristics. Beginning with the work of Julesz, [33], and subsequently formalized in later works (e.g. [5, 7, 12],) these perceptually based texture discrimination models typically compare the outputs of local oriented filters. For example in [7], Bergen and Landy lay out a framework for segregating images based upon the opponent energy in the textures in different regions. Opponent (H-V) energy was computed by taking the difference between horizontal- or vertical-edge sensitive units [1] An image region which contains a different texture is found by by comparing the maximum difference in opponent energy to a template of region.

In image database retrieval research, variations of these models have been incorporated into systems designed to retrieve images which contain regions consisting of particular textures [53, 74, 18]. Recently, similar techniques have been applied in the reverse problem of synthesizing texture images which share the same visual appearance as a given input texture [32, 77, 17]. In Chapter 2 we presented a texture model which sampled a new texture directly from the spatial frequency bands of the original, constrained by the joint occurrence of features at each spatial frequency. Incorporation of this multiresolution constraint resulted in the ability to synthesize textures which faithfully mimic the texture characteristics from more intricate examples. In this work we reverse this procedure and use it as the basis for a texture discrimination model.

[1] Right-tilted versus left-tilted energy was computes as well.

3.2 Discrimination By Inverting The Synthesis Procedure

To synthesize the textures in the previous chapter we rearranged the spatial frequency components of the original texture by selecting from among only those regions which would generate visual structures which would be *indistinguishable* from structures in the original. We measured whether or not two regions are "indistinguishable" by comparing the joint response of filter-banks at multiple resolutions in the images.

Similar techniques have been used by other authors ([33, 5, 12] for example.) However, the technique used here is different from others in that it explicitly uses constraints which require that all feature responses at all resolutions are *jointly* within threshold. Based on this technique we can build a texture discrimination system which measures the similarity between the textures within two images. The performance of this system provides a measure of the utility of this measure of texture difference.

Given the synthesis model presented in the preceding section, we can ask what is the probability that one image could have been the model for synthesizing another. However, because the synthesis technique samples each spatial frequency band directly from the source image, an image which contains even a single spatial frequency component not present in the source will have zero probability of having been synthesized with this technique. In the case of real texture images, there is an extremely low probability that *all* of the values in each spatial frequency band of a test image are present in the model (source) image.

If we had an explicit model of how the spatial frequency content of a single type of parent structure can vary in nature — how a particular parent structure can vary in several images of the same texture — then that could be used to measure the likelihood that some new parent structure could be a different instantiation of the original. However, we have no principled way of building such a model, and furthermore, such a model is probably highly dependent on the type of textures considered.

The approach we use here, is to hypothesize that we do not know the complete distribution which has generated the model image. We then view the presence of similar features as positive evidence that the two textures are similar, in this way a parent structure in the target image which is not present in the model *only decreases the likelihood* that the two images contain the same textures — as opposed to enforcing a likelihood of zero. We consider such a model here.

3.3 Analysis Pyramid and Flexible Bin labels

Given an example image, which contains the texture of interest, features are extracted at each image location at multiple resolutions. For each location in the image, we collect the responses of a set of local oriented filters. This collection forms the *parent structure* of each location, and will be used as a center of a bin in the flexible histogram.

To compute oriented filter responses at multiple resolutions, we first decompose the input image into a Gaussian pyramid, each level of which contains successively lower pass spatial frequency information in the input image. The original image forms the lowest level of the Gaussian pyramid; i.e. $G_0 = I$.

Each successive level of the pyramid is produced by convolution with a Gaussian kernel followed by downsampling by a factor of two:

$$G_{n+1} = 2{\downarrow}(G_n \otimes K_{Gaussian}) \tag{3.1}$$

where $2{\downarrow}(\cdot)$ is the two-times downsampling operation and G_n is the n^{th} level of the pyramid, which is $1/2^n$ the size of the original in each dimension. At each level of the pyramid the response to a set of filters is computed. This can be thought of as passing each low-pass image through a filter bank producing a set of filter response images F_n^i, for filter i at resolution n:

$$F_n^i = G_n \otimes K_i \tag{3.2}$$

At each location (x, y) in the original image we construct a vector of the responses of each of M filters at a location (x, y) at all N levels of the Gaussian pyramid. We call this the the *parent structure*, S, of the region (x, y):

$$S(x,y) = \begin{array}{l} \left\{F_0^0\left(x,y\right), F_0^1\left(x,y\right), \cdots, F_0^M\left(x,y\right),\right. \\ F_1^0\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right), F_1^1\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right), \cdots, F_1^M\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right), \\ \cdots, \\ \left. F_N^0\left(\left\lfloor \frac{x}{2^N} \right\rfloor, \left\lfloor \frac{y}{2^N} \right\rfloor\right), F_N^1\left(\left\lfloor \frac{x}{2^N} \right\rfloor, \left\lfloor \frac{y}{2^N} \right\rfloor\right), \cdots, F_N^M\left(\left\lfloor \frac{x}{2^N} \right\rfloor, \left\lfloor \frac{y}{2^N} \right\rfloor\right) \right\} \end{array} \tag{3.3}$$

Given an example image, which contains the texture of interest, we compute the parent structure for each location as described in section 2.3.2.

From equation (2.8), we see that the parent structures of two locations share values only when they have similar visual structure. Because of their proximity nearby regions have similar low frequency appearance and share values in their parent structures. (In fact, neighboring pixels share the same values for all but the highest resolution.)

The requirement that each feature at each resolution is within threshold is made explicit in the candidate set membership test in equation (2.9). We use the candidate set associated with each location in the model image as a bin in the the flexible histogram.

We can reformulate the candidate set membership condition by scaling each dimension (i.e. each feature at each resolution) so that the threshold to which we are comparing it is the same. Then, the L_∞ norm of in this dimension-scaled space can be compared to a single threshold, T yielding an equivalent condition:

$$\left\| D\left[\vec{S}\left(I,x,y\right) - \vec{S}\left(I',x',y'\right)\right]\left(\vec{z}I\right)^{-1}\right\|_\infty < T \tag{3.4}$$

where each component z_i of the normalization vector $\vec{z}$ scales the corresponding feature response so that $F_i < T_i \rightarrow \left(\vec{S}_i\right)(z_i) < T$.

The bin described by the inequality in equation (3.4) defines a $N \times M$ dimensional hypercube in resolution and feature space.

From equation (3.4) it is clear that all the regions in an image I' which are considered "indiscriminable" from $I\left(x,y\right)$ must fall within a hyper-rectangle (or hyper-cube, if all the dimensions are scaled equally) in the feature-times-resolution dimensional space. The use of the L_∞ norm, which causes the hard edges of the hypercube, establishes the criterion that each feature at each resolution must jointly be within threshold. Using another norm, for example the L_2 norm, would define a hyper-ellipse in this space and would allow similarity in one dimension to compensate for difference in another. By preventing this "trade off" effect, two texture regions are only close under this measure, when *none* of the feature differences is above threshold. We are viewing each of the dimensions as the response of an independent texture discriminator; when any discriminator can detect an above threshold difference, the the parent structure is not accepted into the candidate set. This loosely agrees with our intuitions about human perception of image difference. For example, a picture of a familiar face will look improper if its eyes are enlarged, regardless of the fact that the skin-color may be exact.

3.4 Flexible Histograms

For every candidate set in the model image we define a histogram bin which measures the frequency of regions in a test image which are within threshold (defined by equation (3.4).) In this way generate a histogram for the test image, taken with respect to the model image. Because the bin labels used to generate these histograms are dependent on the model image, we term them "flexible." Thus when searching for different types of textures (i.e. when using different models) the bins used in the histogramming process will be specialized for each texture.

Because the parent structures of many locations in the model image are similar, the bins in the flexible histogram are non-exclusive. Thus, a parent structure can fall into multiple bins. This has the effect of increasing the relative importance of model image parent structures which occur often.

To compare a test image to a model, we compare the histogram generated for the test image, with respect to the bins defined by the model image, to the histogram generated by the model image, with respect to the same set of bins — i.e. with respect to itself.

In the initial stage of model processing, the parent structure for every pixel in the model image is computed. Using those parent structures, a histogram is computed for the model image with respect to itself. For each parent structure, the number of other parent structures in the model image which are within threshold distance are accumulated and stored in a bin corresponding to the location of the pixel whose parent structure was used. Thus for an $I \times J$ image, IJ bins are computed, which can be thought of as a 2D histogram, or can be rolled out into a conventional 1D histogram. This is outlined in the schematic in Figure 3-1.

Using a similar process a histogram for a test image is computed by accumulating the number of parent structures in the test image which are within threshold distance of each model image parent structure. The model and test histogram are each computed with respect to the bins defined by the model image. Therefore, corresponding bins in each histogram are counts of parent structures in each image which are within threshold of *the same* parent structure in the model.

We want to ask the question: "are the two textures different?" By considering the histograms as approximations to an underlying generative distribution, we can ask "are the two texture distributions different?" Since the two flexible histograms have the same

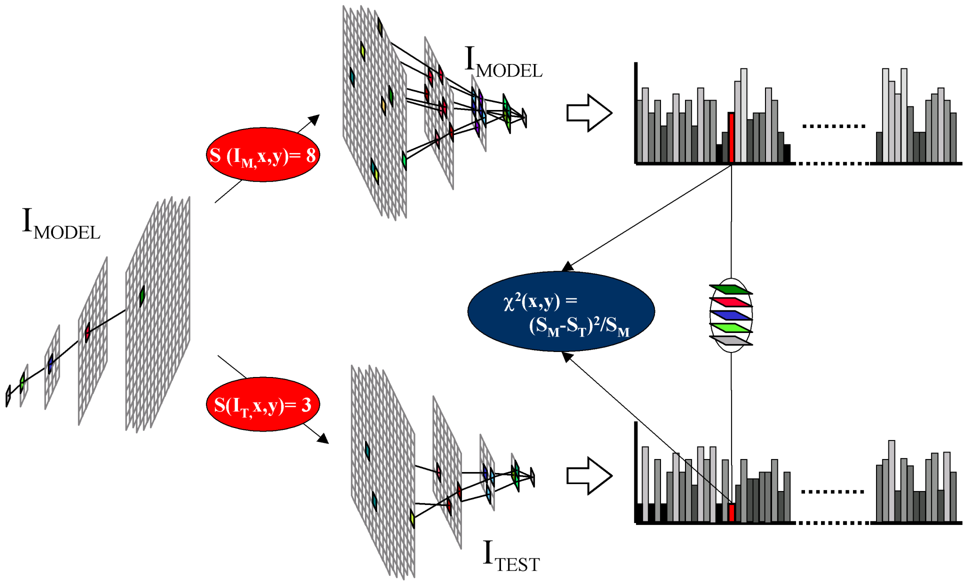


Figure 3-1: By measuring the frequency with which locations with similar parent structures occur, a *flexible histogram* is extracted. The bins of this histogram are determined by the parent structures in successive locations in the model.

bin labels, we can ignore the fact that the labels are flexible, and compare them using the standard chi-square test:

$$\chi^2 = \sum_i \frac{\left(B_i^{\text{model}} - B_i^{\text{test}}\right)^2}{B_i^{\text{model}}} \tag{3.5}$$

Where B_i^{model} is the count of parent structures in the model which fall into bin i, and B_i^{test} is the corresponding count for the test image.

Because we measure the χ^2 (un)likelihood that the test is a sample from the distribution described by the model, the denominator in equation (3.5) is greater than zero for all bins. Thus, no image results in a $\chi^2 = \infty$, achieving the desired criterion that though perhaps very unlikely, no two images should be measured to contain different textures with complete certainty.

The χ^2 value yields a measure of the difference between two textures and its negation yields a similarity likelihood. Given this measure, we can compute the likelihood that a set of images in a collection contain a particular type of texture.

If a binary decision is needed — "is the texture the same or different?" — a texture-detector can be built by comparing the likelihood measure of each image to a threshold η_{model}. Using standard chi-square tables we could determine a value for η_{model} for any desired level of certainty that the two histograms represent the same distributions. However, we do not presume that the flexible histogram for a given model completely describes the texture, and choosing a fixed likelihood could eliminate true-positives which fall below the threshold, but far above false-positives. What is important is that η_{model} is chosen empirically to maximize the percentages of true-positives while guaranteeing an expected level of false-positives. The curve generated by varying η_{model}, is known as the receiver operating characteristics curve, and will be shown for several sets of data in the experiments section below.

3.5 Incorporating multiple model images

Suppose we have access to not just a single image of the desired texture, but several images. Under these circumstances one would hope that a discrimination system would be able to perform better, because it has more information about the target texture. With slight modification the current model can incorporate information from multiple examples of the target texture to improve its performance.

Multiple model images could be used in several ways. Each could be used as a separate model, and then their likelihood measurements for a given test could be combined with some function, such as taking the average, or maximum, or some function in between, i.e. top-k-average. A better method however, if we believe that the models are truly examples of the same texture, is to use the additional information to improve a single estimate of the underlying generative distribution.

We can do this in two ways, first by simply adding the additional parent-structures images into the flexible histogram taken with respect to only a single image. This has the effect of smoothing the the frequency-counts in the bins, but does not affect the number of bins present. This improves the texture model by simply incorporating the relative frequencies of parent structures in the additional model images, which (if they are truly examples from the same underlying generative distribution) will increase the accuracy of the histogram's approximation to the distribution.

The size normalized χ^2 likelihood, which corrects for the model and test image histograms that generated over different numbers of parent structures, is given by:

$$\chi^2 = \sum_i \frac{\left(\sqrt{\frac{\sum_k |I_{model_k}|}{|I_{test}|}} B_i^{model} - \sqrt{\frac{|I_{test}|}{\sum_k |I_{model_k}|}} B_i^{test}\right)^2}{B_i^{model}} \tag{3.6}$$

Alternatively, if we are willing to incur the additional computational cost of comparing more bins, the additional model images can be used to both generate additional bins and to smooth the frequency counts in the bins from the other model images. In this case equation (3.6) is still used, however, the number of bins over which i ranges is increased.

3.6 Specifics of the current instantiation

There are two sets of parameters which must be specified to fully describe the procedure which was implemented and used to perform the experiments presented here. The exact form of the filters, and their associated maximum difference thresholds required for histogram bin membership.

3.6.1 Filters

In the current system five features were used. Though informal experiments indicate that their exact nature is not critical in performance (assuming thresholds are adjusted appropriately), the exact form is described here for completeness.

Feature $F_n^0(I_n)$ is a Laplacian, which captures the non-oriented bandpass frequency information present in the image, I at resolution n. Specifically $F_0(I)$ contains the spatial frequency information in I which is not present in its Gaussian $2\times$ downsampled version, I_{n+1}:

$$\begin{aligned} F_n^0(I_n) &= I_n - 2{\uparrow}[2{\downarrow}(I_n)] && (3.7)\\ &= i_n - \frac{1}{8}\left\{\begin{array}{ccc} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{array}\right\} \otimes I_n && (3.8) \end{aligned}$$

Features $F_n^1(\cdot)$ through $F_n^5(\cdot)$ are the response of 3×3 oriented filters which are sensitive to horizontal and vertical edges and bars:

$$\begin{aligned} F_n^1(I_n) &= \frac{1}{8}\left\{\begin{array}{ccc} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{array}\right\} \otimes I_n && (3.9)\\ F_n^2(I_n) &= \frac{1}{8}\left\{\begin{array}{ccc} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{array}\right\} \otimes I_n && (3.10)\\ F_n^3(I_n) &= \frac{1}{4}\left\{\begin{array}{ccc} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{array}\right\} \otimes I_n && (3.11)\\ F_n^4(I_n) &= \frac{1}{4}\left\{\begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array}\right\} \otimes I_n && (3.12) \end{aligned}$$

Convolution at the edges were handled by reflection about the borders of the image.

It is sufficient to use filters with a relatively small support (i.e. 3×3) because at each resolution we are concerned only with the features which are present *only in that resolution*. By applying this filter bank to each resolution and considering the joint response across resolution, it captures the same longer range information, as do larger-support kernels.

We can see this effect in a one dimensional example. The optimal discrete approximation to a Gaussian is given by the row of Pascal's triangle corresponding to the desired support:

$$
\begin{array}{ccccccccc}
 & & & & 1 & & & & \\
 & & & 1 & & 1 & & & \\
 & & 1 & & 2 & & 1 & & \\
 & 1 & & 3 & & 3 & & 1 & \\
1 & & 4 & & 6 & & 4 & & 1 \\
 & & & & \vdots & & & &
\end{array} \tag{3.13}
$$

Using a discrete approximation with a support of 3 (i.e. $\{1\ 2\ 1\}$) does not provide as accurate an approximation as a would a larger support, *when we consider convolution only at a single resolution*. However, two applications of the size 3 kernel is equivalent to a single application of the size 5 approximation:

$$\{1\ 2\ 1\} \otimes \{1\ 2\ 1\} = \{1\ 4\ 6\ 4\ 1\} \tag{3.14}$$

Further, since convolution is a linear operation:

$$
\begin{aligned}
\{1\ 4\ 6\ 4\ 1\} \otimes S &= (\{1\ 2\ 1\} \otimes \{1\ 2\ 1\}) \otimes \Psi & (3.15)\\
&= \{1\ 2\ 1\} \otimes (\{1\ 2\ 1\} \otimes \Psi) & (3.16)
\end{aligned}
$$

Where Ψ is a one dimensional signal.

Thus, application of $F^1_{n+1}(I_{n+1})$, for example, is equivalent to application of a larger vertical edge filer on I_n.

3.6.2 Bin membership threshold

To be considered similar enough to be counted in a particular flexible histogram bin, each component-wise difference must be below some resolution and feature dependent threshold, T^i_n in equation (3.4).

The level of these thresholds are critical in establishing the distribution of bins into which a particular parent structure will fall. A threshold which is too low will result in loss of generality, essentially causing the histogram bins to become overly specific feature detectors which are unlikely to respond to even true-positives. Conversely a threshold which is too large will result in bins which accept large variation in parent-structures, weakening the overall specificity of the combined model.

To establish appropriate levels for these thresholds, we measured the differences between image regions which are believed to contain the same texture. We did this for a set of 800 textures from the MIT AI Lab Learning & Vision Group Texture database [31]. Specifically we computed the approximate average difference between the components of pairs of parent structures taken from the *same texture image*. Each texture was 64×64 pixels. A full computation of the $4096^2/2 \approx 16$M possible pairwise differences for all 800 images is impractical, and was approximated by computing the differences for $10,000$ pairs of parent structures chosen at random for each image.

The approximate average differences are shown in Figure 3-2, Average difference is plotted against resolution, and each curve represents the measurements for a separate fea-

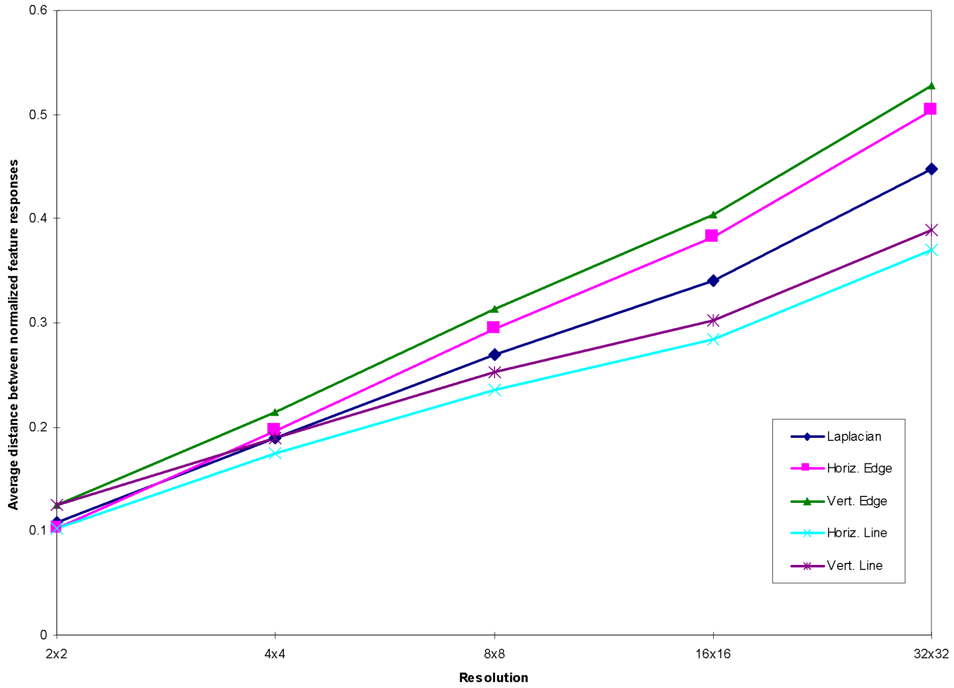


Figure 3-2: The approximate average parent structure differences for a set of 800 textures.

ture. The average difference between feature responses increases roughly proportionally to the log of the resolution, with a slope which varies across features.

In marked contrast to this data, we collected the same measurements for a set of 800 natural images taken from the Corel image database [13]. The data for the natural images is shown in Figure 3-3. roughly independent of the resolution at which they are computed. For natural images, the levels of each curve are roughly constant with respect to resolution, indicating that the average difference between feature responses is roughly independent of the resolution at which they are computed. Additionally the average difference is roughly constant across feature type.

From the image data we extract appropriate levels for the thresholds for equation (3.4). Since the measurements in Figure 3-3 are differences in parent structure components, for parent structures which are present in the *different images*, they provide a reasonable upper bound on the acceptable differences between parent structures from images which contain we consider to be *different textures*. In the current implementation we use $T_n^i = 0.2$ for all resolutions and features.

3.7 Experiments

3.7.1 Natural textures

To measure the performance of this system we measured its receiver operating characteristics on a set consisting of three 64×64 examples of 20 different textures. The three examples of each texture were acquired by extraction of image patches from a single image

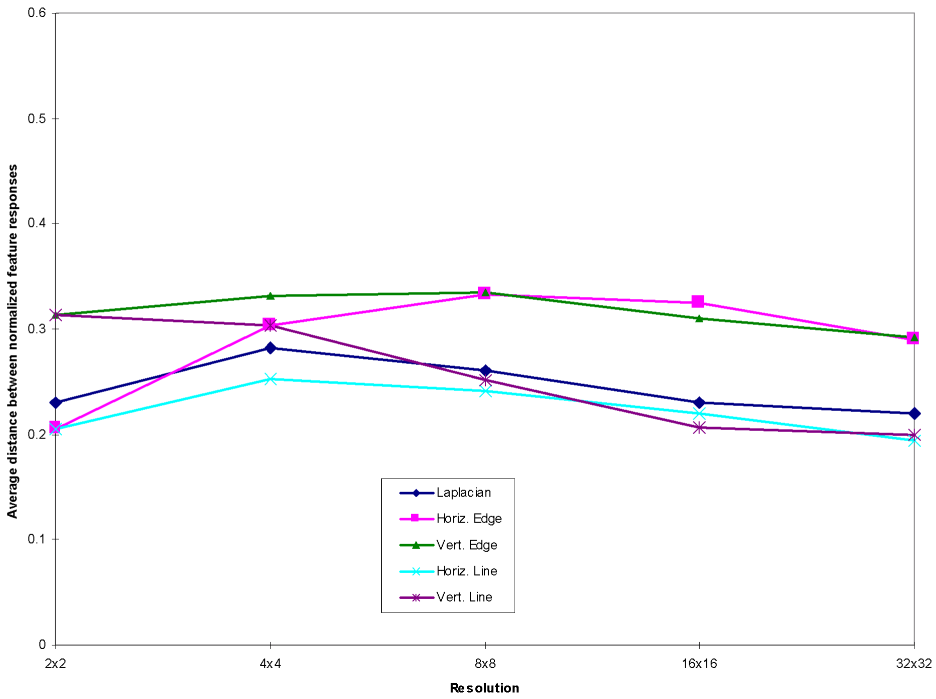


Figure 3-3: The approximate average parent structure differences for a set of 800 non-texture images.

Figure 3-4: Two images which contain the same texture.

Figure 3-5: Images which contain different textures.

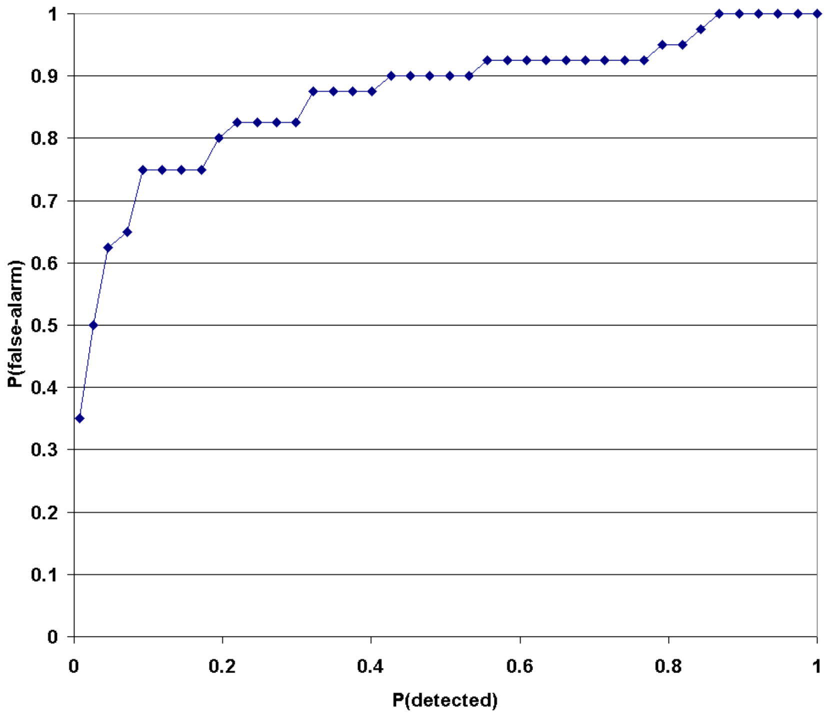


Figure 3-6: Receiver operating characteristic (ROC) curve for discrimination of natural textures with this technique.

of a roughly texturally homogeneous natural scene. Examples of two images from the same texture are shown in Figure 3-4, and examples from several different types of textures are shown in Figure 3-5. The images cover a large range of textural characteristics, however, as is clear in Figure 3-5 some examples of different textures (i.e. from different sources) do have a similar appearance. As a result, we expect that even human observers will be unable to perfectly identify examples of a target texture. In section 3.7.2 we compare the performance of this system to that of several human observers.

Using one of the three examples of a given texture as a model, we measured the receiver operating characteristics curve for discrimination among a set made up of 40 images, two examples from each texture. In this way, we acquired 20 ROC curves, one for each model, which were averaged to generate the curve shown in Figure 3-6. In this graph, percent correctly detected (true-positives) is plotted versus percent incorrectly determined to be the target texture (false-positives), as a function of increasing threshold η. On this graph a diagonal line from $(0\%, 0\%)$ to $(100\%, 100\%)$ represents chance; the ROC of a detector whose response is random, i.e. is completely independent of the data, would fall on this line.

From this curve the optimal Bayes decision rule can be determined. Given some belief about the cost of making an error in detection, and of the prior probabilities of each texture occurring, the Bayes optimal rule is given by:

$$\frac{P(d)}{P(fa)} \underset{reject}{\overset{accept}{\gtrless}} \frac{C_{fa} - C_d}{C_{fn} - C_n} \times \frac{P(\neg T)}{P(T)} \equiv \eta_B \tag{3.17}$$

where C_d, C_{fn}, C_n, and C_{fa} are the costs associated with detection (true-positive), false-negative, true-negative and false-alarm (false-positive) respectively. $P(T)$ and $P(\neg T)$ are the prior probabilities of the presence or absence of the target texture.

In the "toy" problem of the natural textures, we have no prior beliefs about the costs associated with correct and incorrect responses. Thus we can treat left-hand side of the binary test in equation (3.17) as a tunable parameter η.

If for example, we set η to 1 (which is equivalent to setting the expected cost of a random answer to zero) we are left with the maximum likelihood decision rule:

$$\frac{P(d)}{P(fa)} \underset{accept}{\overset{reject}{\gtrless}} 1 \tag{3.18}$$

On the natural textures used in this experiment, the maximum likelihood decision rule yields an average of 75% percent accuracy. Compared to 5% accuracy obtained by chance, this is a good performance level; however, for this data many techniques may be able to achieve such performance. For example simple color histograms (i.e. those in [65]) could easily discriminate between many of the textures.

To get a notion of the level of performance which is feasible by a system which is capable of integrating visual cues from multiple local and global characteristics available in the the texture images, we measure the receiver operator characteristics achieved by human observers.

3.7.2 Comparison to human performance of discrimination of natural textures

To measure the relative similarity between test images perceived by human observers we need to acquire a ranking for each image which contains the target texture versus those which do not.

However, this cannot be done directly because human observers do not produce reliable rankings when presented with large sets of options; thus, presenting the observer with all 40 test images and requesting a full ranking would produce poor measurements. Because of this, psychophysical measurements of this form are typically done using an k-alternative forced choice (kAFC) paradigm in which the observer is presented with some number, k, images from which they must choose one [30]. In this case the observer indicates which image is perceived to be most similar (in texture composition) to the target

From each response a partial ranking, of the chosen image over the $k-1$ other presented images, is acquired. To acquire a full ranking in this way requires $\sum_{i=1}^{\log_k(tests)} N/k(i)$ k-AFC questions per target texture. Clearly for small k, this results in infeasibly long testing times.

However, to generate an ROC curve we require the rankings of *only the true-positive images*. Thus, by using a procedure which identifies the rankings of most similar images first, the testing for a give target can be terminated after the the rankings of the true-positives

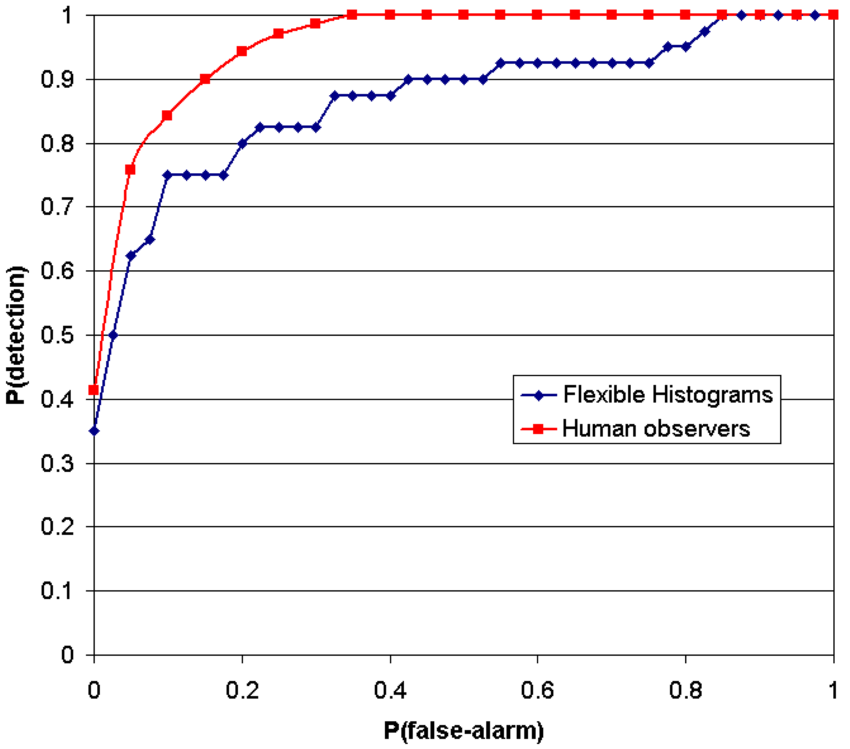


Figure 3-7: ROC curve averaged over 7 observers for 5 target textures.

have been determined. Further, since we expect human observers to rank the true-positives near the top, much of the cost accrued in determining a full ranking can be avoided.

The procedure we use is a series of $k_i - AFC$ questions, where $k_i = \{40, 39, \ldots\}$. In the first question the top-most similarly perceived image is found. Once determined this image is removed from the set of alternatives, and those remaining are presented in the next question. At worst this technique could require 40 questions per target; in practice, however, the true-positives are identified within far fewer.

The procedure has been implemented through a world wide web interface and can be viewed from the URL:

`http://www.ai.mit.edu/~jsd/Research/Discrimination/Human`

As a side-effect of terminating after both target-images had been identified, we found that observers began to recognize images which had been correct responses to other targets, when presented with the same set of distractors (i.e. non-target images).

To prevent the positive feedback, received by terminating after the correct images had been identified, we extended each trial using the following strategy:

- If both target-images are not identified continue.
- If both target-images are identified continue with probability 0.1.

This hampers the ability of the observers to build and use multiple models. In addition, each observer was only presented with 5 of the 20 possible target textures, to prevent these effects of over-exposure to the test set.

The averaged results for 7 observers across all target textures are shown in the red (top) curve in Figure 3-7. The performance of human observers is slightly better than that of the flexible histogram technique (shown in blue); the maximum likelihood probability for detection for a human observer is about 83%, and for flexible histograms about 75%.

In the next experiment we measure the performance of the present texture model on discrimination between different classes of images generated from synthetic aperture radar (SAR) data.

3.8 Vehicle detection in SAR data

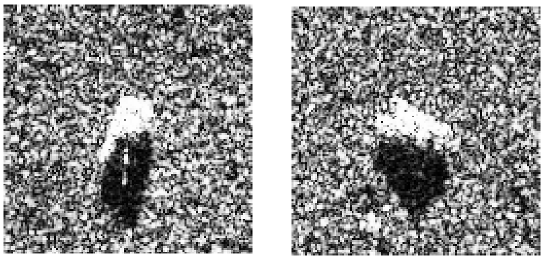
In this section we present the preliminary results of using this model for vehicle detection in synthetic aperture radar images. Using a model constructed from four SAR images of a particular vehicle type, we classify a data set consisting of three types of images: images of the target vehicle, images of a second vehicle, and clutter images which contain no vehicles. The data was acquired from the Model Based Vision Lab MSTAR project [43]. [2]

In each class there were 140 images; two examples of each are shown in Figure 3-8. The target vehicle was a T72 tank (a), distractor images consisted of images of a BMP2 personnel carrier (b), and clutter images (c) taken from a farmland region in Huntsville, Al. All of the SAR data was collected with a (measured) depression of approximately 15 degrees; the magnitude of the complex SAR images was used to generate sets of non-overlapping 128×128 gray scale images. Contrast equalization was used to maximize the dynamic range of each image, and to eliminate any potential contrast or brightness cues in the original data which could be used to perform discrimination.

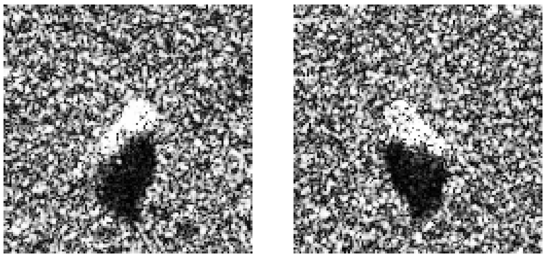
Using this dataset we ask two questions: with four model images of the T72 vehicle, how well can we discriminate between images containing clutter and those containing a vehicle; and given a model of either type of vehicle how well can we discriminate between it an the other vehicle class. From the examples in Figure 3-8 it is clear that the clutter images (c) have a globally different appearance from those containing vehicles (a and b). Therefore we anticipate better performance for any technique in discriminating between images of class (c) and those of (a) or (b) than between images of (a) versus those of (b).

The ROC plot in Figure 3-9 shows the performance of the current technique at discriminating between clutter images and images containing *either* vehicle type, given two model images of *only* the T72. The point 100% detected versus 0% false-alarm is reached indicating that at some threshold, perfect performance, over this limited data set, is achieved. However, this measure is only preliminary; in military applications the Neyman-Pearson criterion, i.e. the maximum acceptable P(false-alarm), is below the $\pm \sim 1\%$ precision of these experiments.

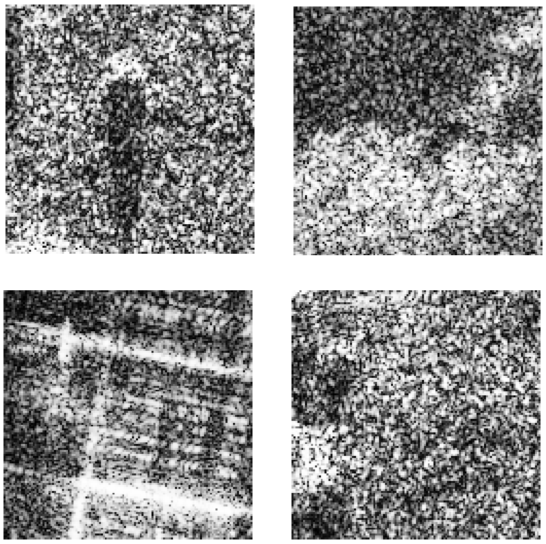
[2]"This data set was collected in September of 1995 at the Redstone Arsenal, Huntsville, AL by the Sandia National Laboratory (SNL) SAR sensor platform. The collection was jointly sponsored by DARPA and Wright Laboratory as part of the Moving and Stationary Target Acquisition and Recognition (MSTAR) program. SNL used an X-band SAR sensor in one foot resolution spotlight mode. Strip map mode was used to collect the clutter data. This subset of data from the September 1995 collection has been identified by DARPA and Wright Laboratory for public release." [66]



(a)



(b)



(c)

Figure 3-8: 128×128 SAR images of (a) T72 tanks, (b) BMP2 personnel carriers, and (c) clutter images which contain no vehicles.

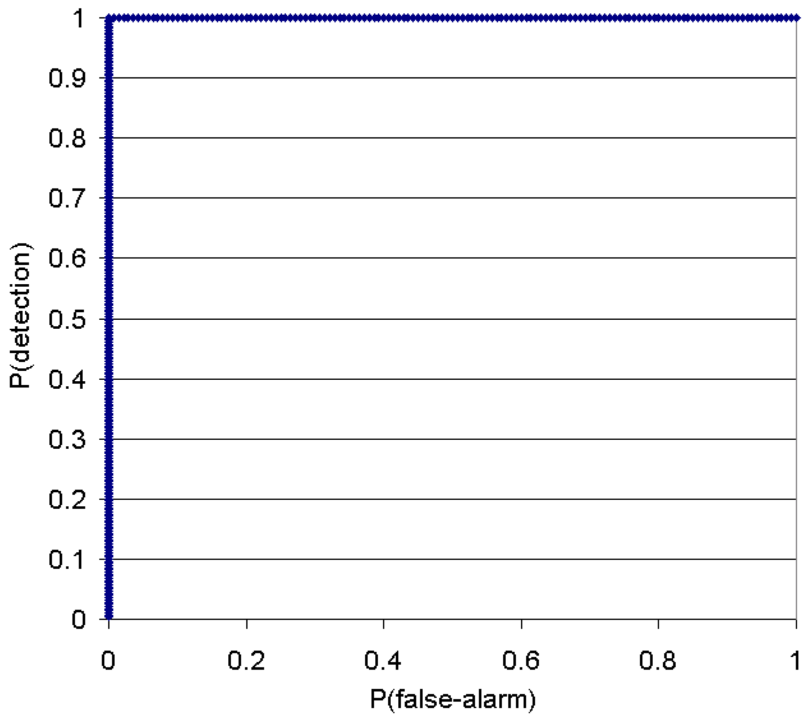


Figure 3-9: ROC curve for discriminating full resolution (128×128) T72, or BMP2 vehicles from clutter images.

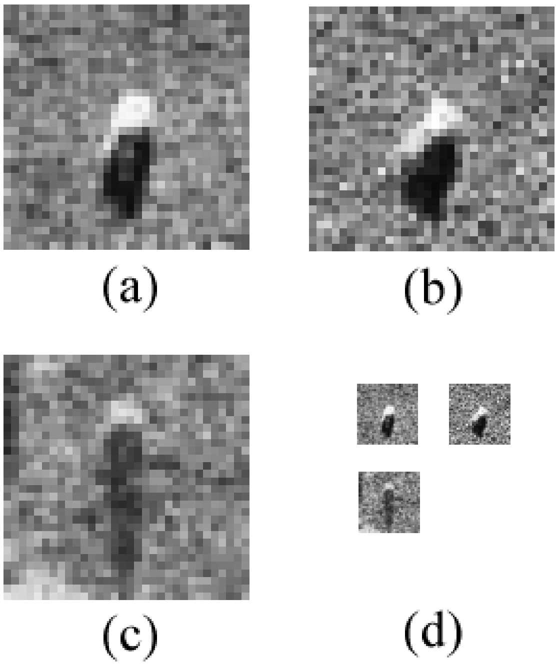


Figure 3-10: $4\times$ downsampled (a) T72, (b) BMP2, (c) clutter images blown up for visibility. (d) images at true size.

In real SAR applications, the volume of input images is such that the $O\left(pixels^2\right)$ operations required to fill the flexible histograms is prohibitive. An important issue is the development of *prescreening* algorithms which can reliably guarantee 100% probability detection with a minimal probability of false-alarm. The output of such a system can then be fed into a more precise, though more computationally intensive, second stage algorithm. To decrease the run-time of the flexible histogram method, we can consider decreasing the size of the input images by low-pass filtering and subsampling.

Figure 3-10 shows three of the images in Figure 3-8 down-filtered by a factor of $4\times$ in each dimension. The effect of down sampling is that it removes the highest frequency information in the input images. However, a significant computational advantage is gained because the operations required by flexible histogram technique decreases by a factor[3] of 256.

Figure 3-11 shows the performance using versions of the images, which have been downsampled $2\times$, $4\times$, and $8\times$ in each dimensions; as resolution decreases, computation time speeds up by factors of 16, 256 and 4096 respectively. Using lower resolutions causes however, causes dramatic decrease in discrimination performance as well.

As resolution decreases, progressively more of the vehicle-containing images which are least similar (under the flexible histogram χ^2 measure) to the model, are "confused" with the most model-similar clutter images. This effect can be visualized by examining the flexible histograms of true positive images as resolution changes changes.

[3] 4 times down sampling in each dimension yields 4^2 times fewer pixels, which yields a speed-up factor of $4^{2^2} = 256$.

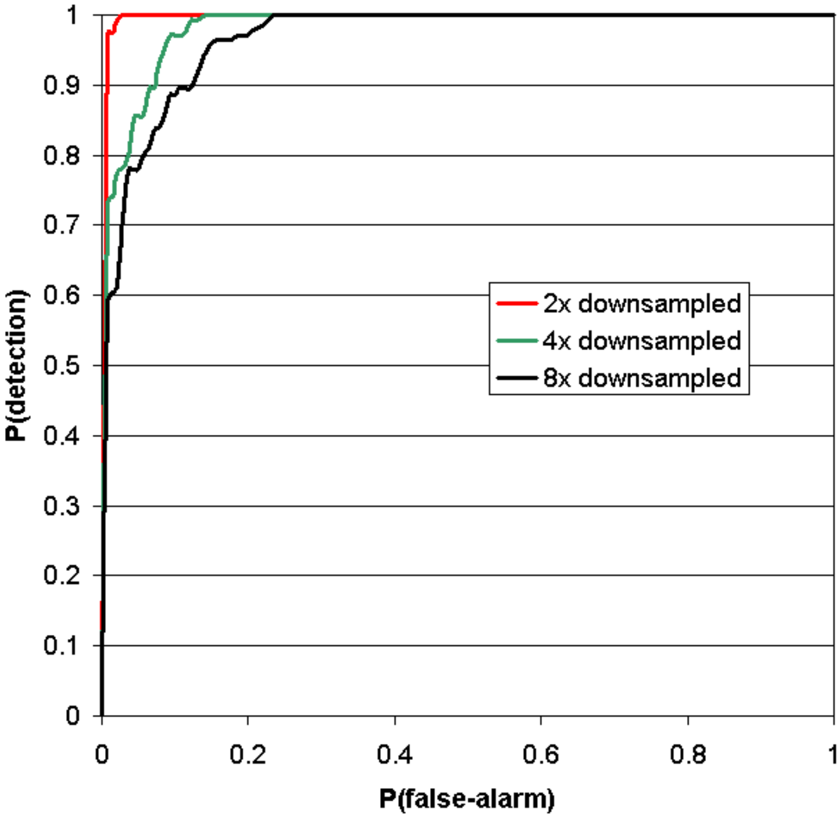


Figure 3-11: ROC curve for detecting vehicles using downsampled images.

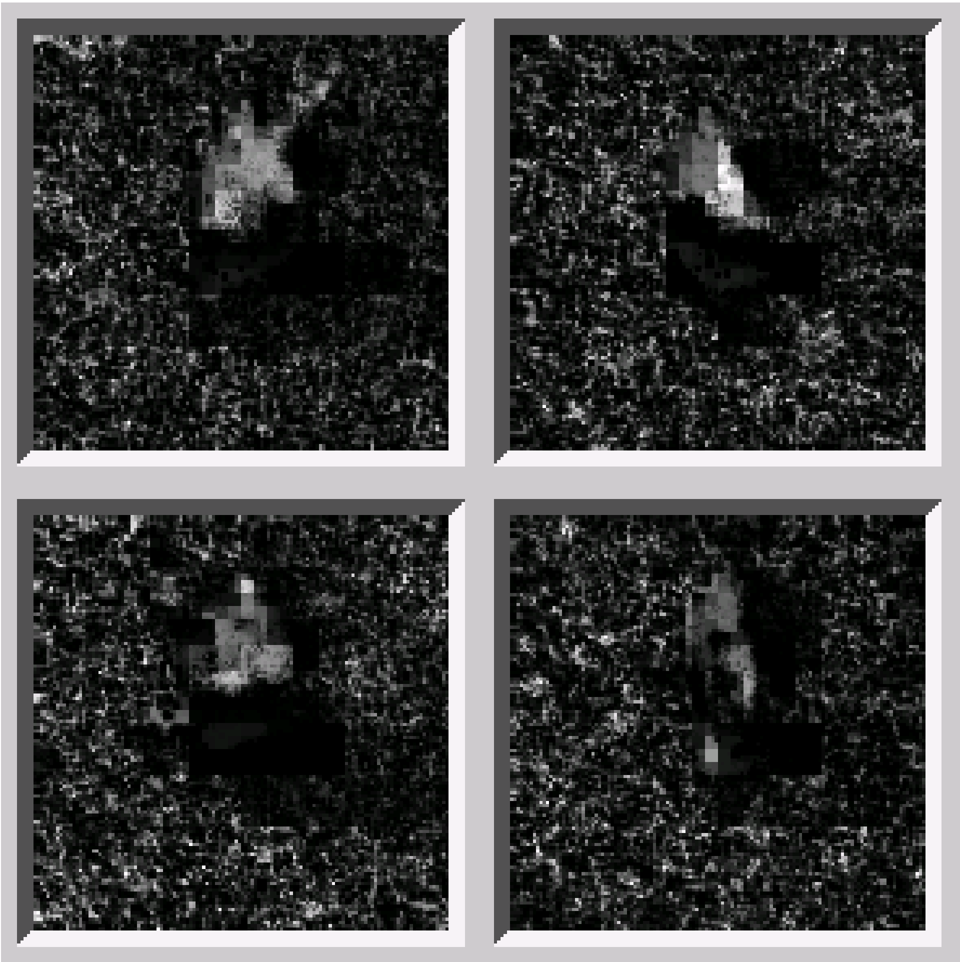


Figure 3-12: 2D flexible histograms of a full resolution image containing a target. Each histogram is with respect to one of the four model images.

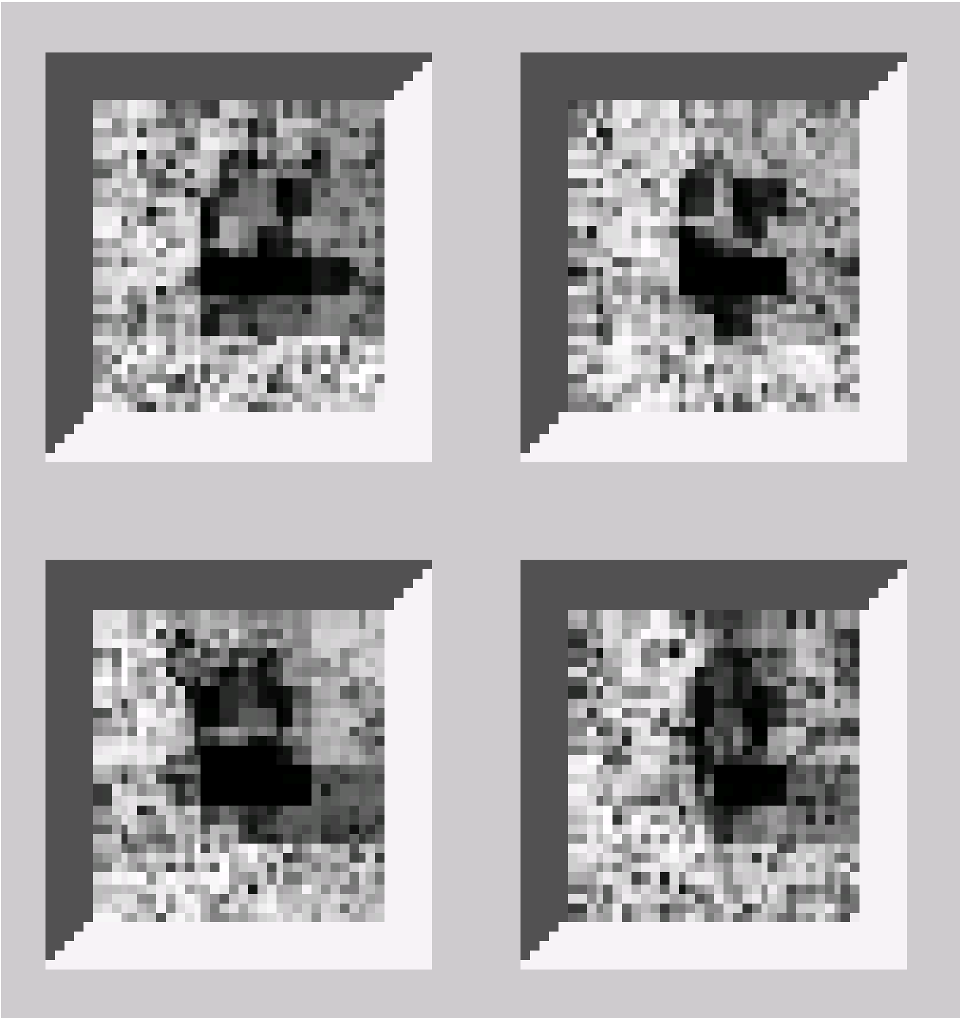


Figure 3-13: 2D flexible histogram of $4\times$ downsampled image containing targets.

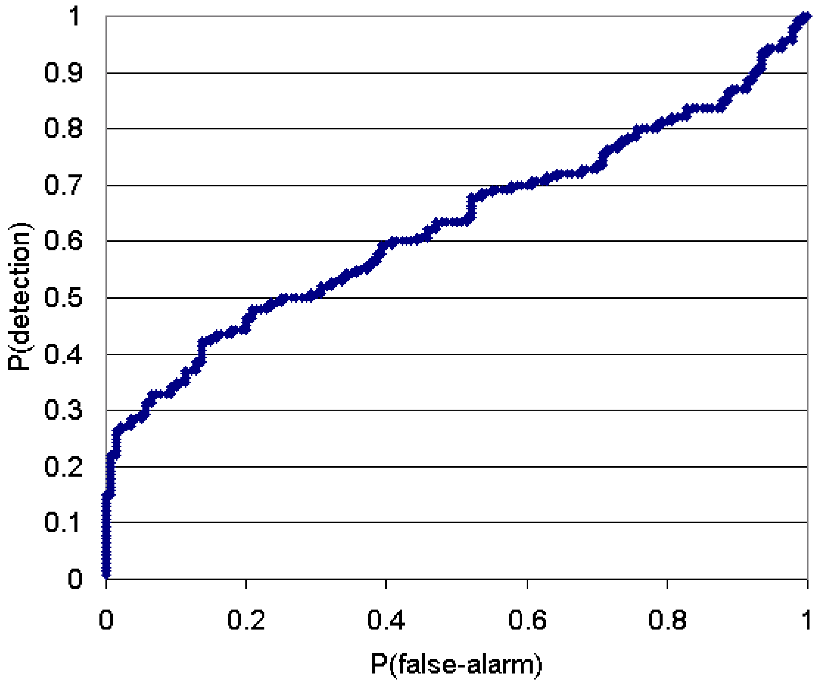


Figure 3-14: ROC curve for discriminating between T72 and BMP2 vehicles using T72 based model.

In Figure 3-12 the two-dimensional flexible histograms for a full resolution true positive image, with respect to the four model images are shown. The bright areas indicate which bins, defined by the parent structures in the model images, are most present in the target image. In these images the bright regions are located in the regions of the model images which contain the target vehicle, indicating that it is this region which is going to dominate the chi-square calculation (equation (3.5).)

In Figure 3-13 the two dimensional flexible histograms for a low resolution version of the same true positive images are shown. When the images are downsized, the regions of the model images most similar to the parent structures in the target image — those regions which will dominate the chi-square measure — are the clutter regions surrounding the target vehicle. As a result, with decreased resolution, the flexible histogram becomes less discriminating between target and clutter images.

Nevertheless, these lower resolution methods can be used as a prefiltering stage which can “screen-out” a large number of the clutter images while utilizing a relatively small amount of processing power. Equation (3.17) can be used to obtain a threshold which guarantees 100% detection, with arbitrary certainty, and a minimal number of false-alarms. All of the images detected by this prefilter system — both true positives and false-alarms — can then be fed into a higher resolution version to eliminate more of the false-alarms.

We now turn to the question of discriminating between the T72 and BMP2 vehicles. Here we only consider the performance of on full resolution images, which yields the best performance of the current method. Using a two images of the T72 tank the ROC curve

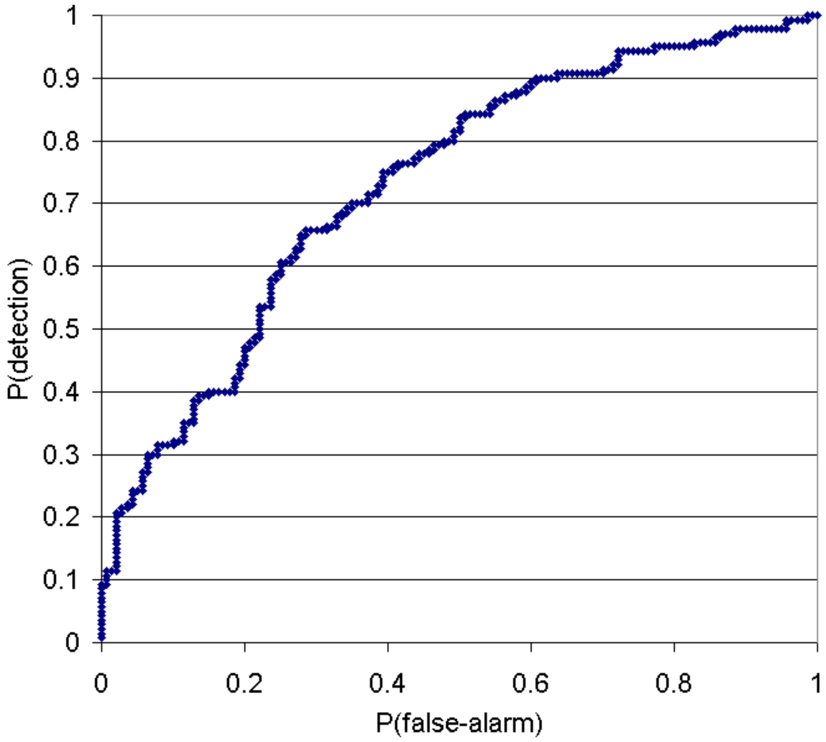


Figure 3-15: ROC curve for discriminating between T72 and BMP2 vehicles using BMP2 based model.

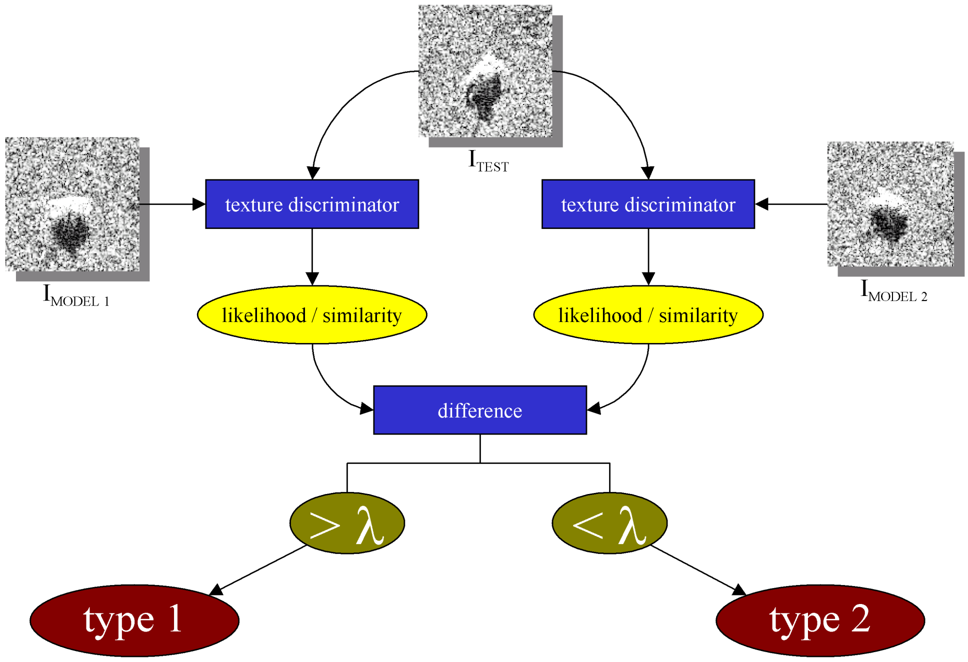


Figure 3-16: Schematic for classification system which combines multiple discrimination models.

shown in Figure 3-14 is obtained. Using a two images of the BMP2 tank the ROC curve shown in Figure 3-15 is obtained. Each curve represents performance which, though better than chance, can only correctly classify 67% (using maximum likelihood classification.)

Using both models simultaneously, however, a classification system was built which improved the overall performance.

3.9 Multi-model classification system

By combining the information provided by both pairs of model images we can design a classification system which yields a ROC curve which is better than either Figure 3-14 or Figure 3-15.

Given some image, we can ask if it is more likely to have been generated by one model than by the other. By taking the difference of the χ^2 measures obtained from comparison of flexible histograms based on each model type, we can obtain a new measure:

$$L_{classify} = \chi^2_{model_1} - \chi^2_{model_2} \tag{3.19}$$

The resulting multi-model system can be described by the diagram in Figure 3-16. Since χ^2 measures can only be directly compared when they are generated with respect to the same model, simply comparing $L_{classify}$ to zero will not take into account any biases in the combined model classification system. To remove biases we compare $L_{classify}$ to a threshold $\eta_{classify}$ using a binary decision rule:

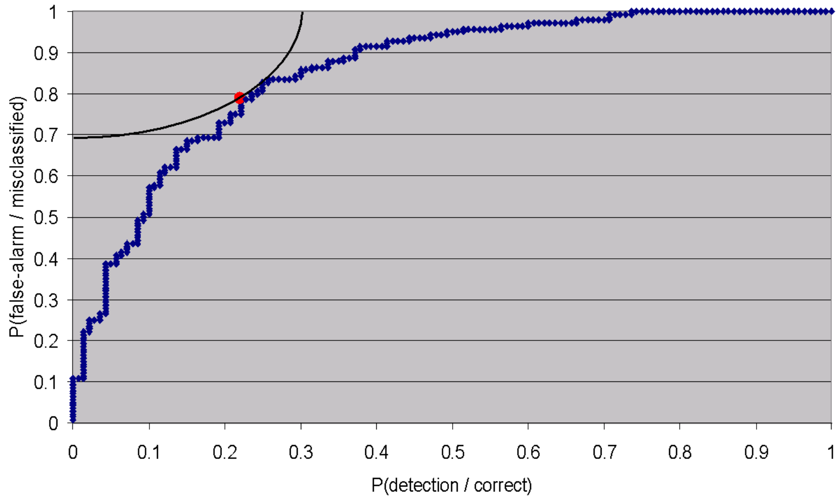


Figure 3-17: ROC curve for classification of T72 versus. BMP2 using models of each

$$L_{classify} \underset{accept}{\overset{reject}{\gtrless}} \eta_{classify} \tag{3.20}$$

By varying the threshold $\eta_{classify}$ an ROC curve is constructed, from which the optimal threshold can be determined.

Figure 3-17 shows the ROC obtained curve obtained by varying $\eta_{classify}$.

The accuracy of the classification system is given by:

$$\begin{aligned} Accuracy &= [P(H = h_1 | I = i_1, \eta_{classify})] \times \\ &\quad [P(H = h_2 | I = i_2, \eta_{classify})] \end{aligned} \tag{3.21}$$

where $P(H = h_k | I = i_k, \eta_{classify})$ is the probability that the classification hypothesizes an image of type k given that the image is really type k, and some decision threshold $\eta_{classify}$. By varying the value chosen for $\eta_{classify}$ we can maximize the accuracy obtained by the system.

The left plot of Figure 3-18 is the accuracy achieved by the system as a function of the threshold. In the right plot, the output of the classification system, $L_{classify}$, is shown for the two types of images. The top (red) curve are the responses to the T72, and the bottom to the BMP2. When performing classification of course, the system does not have access to these labels (as finding them is the objective.) From the peak on the left plot, we see that the maximum accuracy obtained is 78%. At this point, we find the maximum-accuracy threshold value $\eta_{classify}$, which produces a linear discriminator shown by the horizontal line; on the right plot, all images which fall below the line are classified as BMP2, and those above as T72.

Alternatively we can think of finding the maximum accuracy threshold by examination of the ROC curve in Figure 3-17. Optimal classification is obtained at the point on the

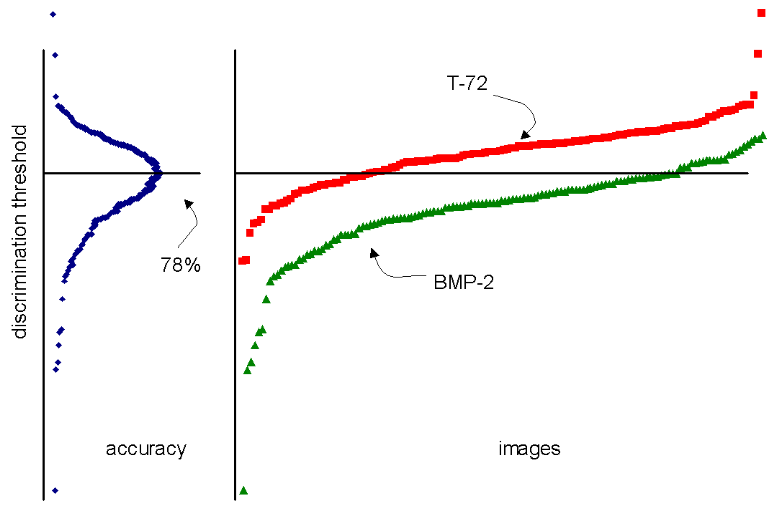


Figure 3-18: By searching over $\eta_{classify}$ the maximum accuracy achievable by the system can be found.

ROC curve which is at the intersection of the ROC curve with the circle of minimum radius centered at $(100\%, 0\%)$. (A circle is used because we assume equal prior probabilities, and cost ratios, which simplifies the Bayes decision rule in equation (3.17), to the maximum likelihood decision rule in equation (3.18). With different priors we would consider the intersection of the ROC curve with the minimum radius ellipse, whose axis ratio is set by those priors.) This intersection falls at 78% P(detection / correct classification), which agrees with the maximization method shown in Figure 3-18.

3.9.1 Flexible histogram difference of synthesized images

To compare the flexible histogram discrimination method to the synthesis procedure on which it was based, we synthesized a set of textures, and measured their distance to to original. We synthesized 5,000 textures at each of two thresholds, from a single example, using the procedure described in Chapter 2. Thresholds of $T = 500$ and $T = 1000$ were used. The original image is shown in Figure 3-19, and examples the images synthesized with the lower threshold are shown in Figure 3-20, and with the higher threshold in Figure 3-21. Flexible histogram difference was then measured between each of the 10,000 images and the original. In Figure 3-22 we plot a histogram of the frequency of images at each difference level.

The two sets of synthesized images cluster into separate groups. The means of each cluster, however, are relatively low compared to the difference of a very different textures which fall off to the right, beyond the realm where the flexible histogram model can accurately measure differences. For the 800 textures in the Learning & Vision Group Texture database [31] all completely saturate or nearly saturate the distance measure by yielding nearly empty flexible histograms.

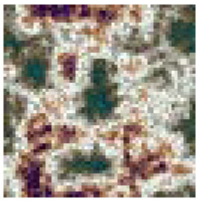


Figure 3-19: The base image from which 10,000 images were synthesized, and compared using the flexible histogram model.

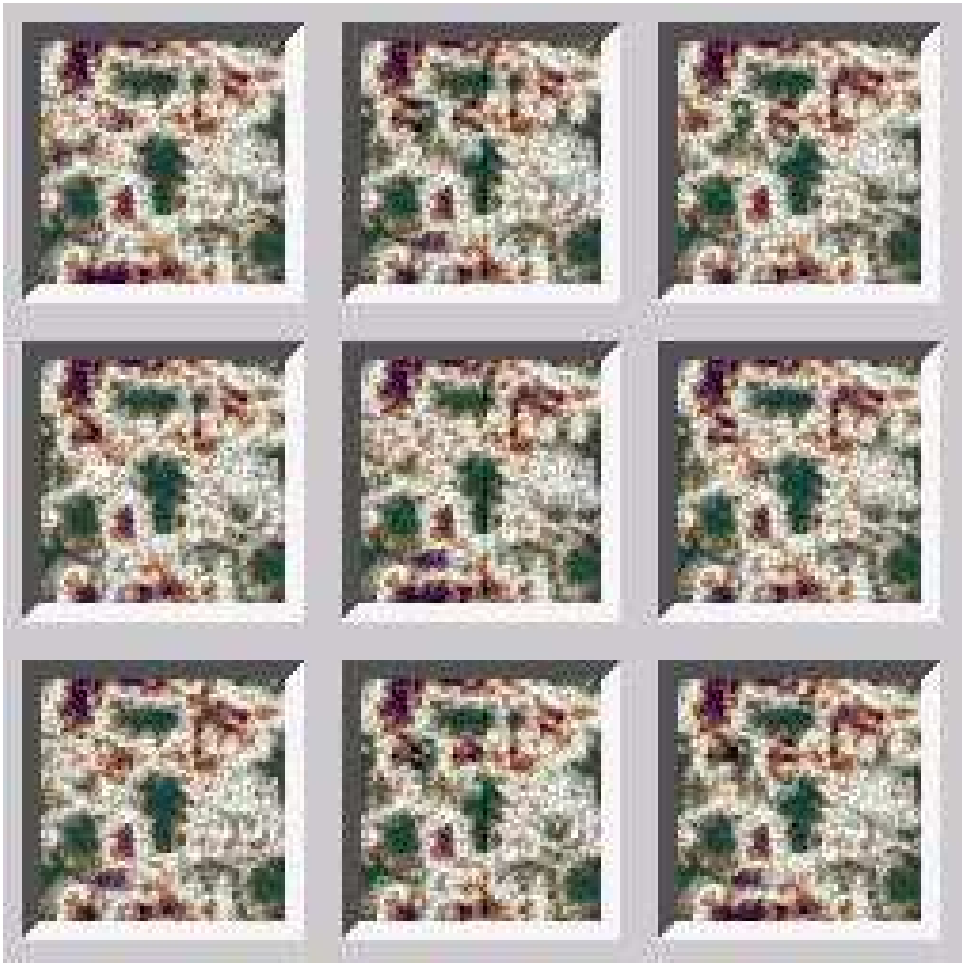


Figure 3-20: Ten example of the 5,000 images synthesized with a low threshold, from the texture in Figure 3-19.

Figure 3-21: Ten example of the 5,000 images synthesized with a higher threshold, from the texture in Figure 3-19.

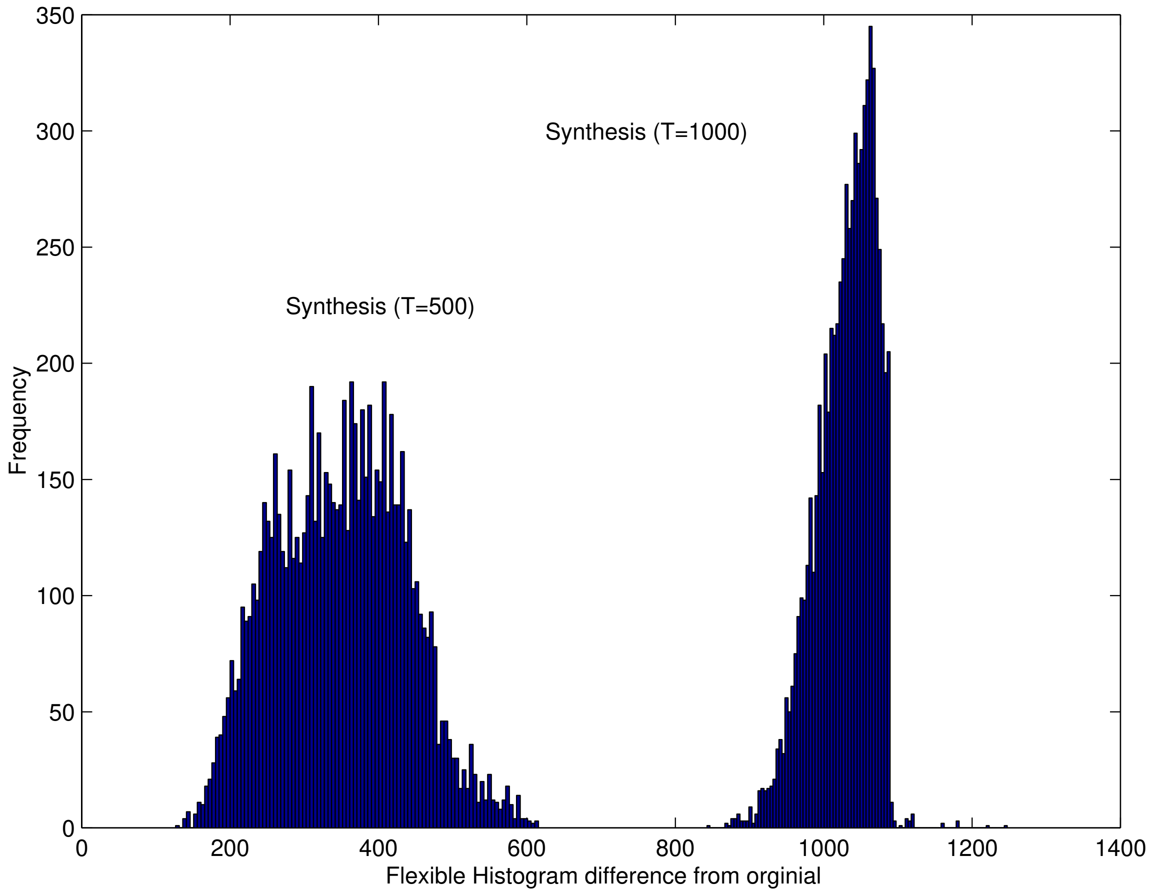


Figure 3-22: Images synthesized with two different thresholds cluster into two groups in flexible histogram similarity

Figure 3-23: Control image, that is very different from the base image in Figure 3-19, for comparison to similarity measures of synthesized images

Though it is possible at any threshold level to synthesize images which are very similar (indeed, even identical) to the input image, with higher thresholds, it becomes increasingly unlikely. In fact, in the 5,000 examples synthesized with the higher threshold $T = 1000$, none fall below the flexible histogram difference level of even the most different image synthesized with a threshold of $T = 500$. Thus to achieve a larger V^* difference, the synthesis technique must sacrifice D^* distance (as approximated by the flexible histogram model).) However, from the perceived textural similarity between the images in Figure 3-21 and the original texture in Figure 3-19, it is clear that this level of discriminability is below the $T_{\text{max disc}}$ threshold.

The smaller variance in difference in the set of synthesized images with the higher threshold and the saturation of the model when very different textures are tested, is symptomatic of the fact that the flexible histogram model does not have an explicit model of variations in parent structure values. As a result, as difference measures increase, the resolution (i.e. differentiation between difference measures) decreases. This is evidence that the flexible histogram texture difference measure is most useful when discriminating between similar images, and is less useful when discriminating between images which are all very different from the model image. In some sense this corresponds to the "comparing between apples and oranges," syndrome: when images are very different from the model, it becomes difficult to to identify which is *more* different, and all that can be said is that they a both *very* different. In many applications however, such as in the case of SAR imagery, it is between very similar images which we need to precisely discriminate.

For general image databases, where images vary greatly in content and appearance, and even images which are considered similar have very different local properties, such an approach decreases in viability. To deal with the larger variety in such image databases, we consider a more sophisticated model in the following chapters.

Potential future research directions for the current texture model include analysis of performance in other domains where only structural information is present, including texture based image segmentation; character and symbol recognition; stereo image correspondence determination; and integration of this texture discrimination approach into systems which use cues such as color or shape for classification and recognition.

3.10 Discussion

The flexible histogram multiresolution texture discrimination approach described here makes explicit the requirement that to be considered similar, textures must contain similar distributions of *joint* feature responses over multiple resolutions. Results on discrimination between natural images indicate performance which, though less than that achievable by human observers, is far greater than chance. Further, for the natural textures used, many other approaches could potentially be successfully applied. Therefore, the main significance of these results is to show that this approach, which is based only on the multiresolution texture organization within images, has significant discrimination power.

Because it extracts information from the visual structure in an image, it is particularly applicable to classification of synthetic aperture radar data, where only impoverished noisy, grey-scale information is available and where color based techniques will fail.

Preliminary experiments indicate that this approach may have sufficient discrimination power to provide the basis of a target detection system. Further analysis, including experiments on larger data sets, and on data which includes images with target vehicles and clutter in close proximity, are required to further refine the system and obtain a better estimate of its overall potential.

Chapter 4

Textures-of-Textures: Toward robust Image Database Retrieval

In this section a new algorithm is presented which approximates the perceived visual similarity between images. The images are initially transformed into a feature space which captures visual structure, texture and color using a tree of filter networks. Similarity is then measured as the distance in this *perceptual feature space*. Using this algorithm we have constructed an image database system, dubbed by some as "Rosetta," [18] because of its ability to (approximately) "translate" between the raw image representation in pixel space into perceptual space in which direct linear discrimination techniques can be applied [20, 1]

4.1 Textures of textures overview

The Rosetta system performs example based retrieval on large image databases. A typical query consists of a small set of images which are representative of a broader class (e.g. images of automobiles or images of city skylines). From the example images a characteristic signature in feature space is computed and is compared to the features of each image in the database. The closest database images are returned.

Performance in this area is notoriously difficult to quantify. We have acquired a set of 2900 images which have been divided into 29 classes based on visual and semantic similarity. In the first set of experiments we use a small set of randomly selected images from each class as a query and measure the reliability with which we can return other images from that class and reject images from other classes.

In a second set of experiments we generated special sets of target images. These target sets contained images of canonical scene which which differed from one another along a single visual dimension. The set contained images generated from a single image which had been altered to varying degrees with an image manipulation function; or contained different images which have been taken with progressive variation of some physical condition. Retrieval rates of the present system are compared to several other techniques which are the basis of many other image retrieval systems [53, 74, 36, 21].

4.2 Expanding the representational power of the flexible histogram model

Though the flexible histogram model of Chapter 3 has the ability to differentiate between different textures, the the distinct curves in Figure 3-2 and Figure 3-3 gives strong indication that textures and natural images have very different parent structure distributions. The flexible histogram model is based on the fundamental assumption that the target images contain a spatially homogeneous texture distribution. For natural images this assumption does not hold.

Furthermore, the "comparing apples and oranges" effect described in section 3.9.1, indicates that if the images in the target set are somewhat dissimilar from the query images, the flexible histogram will be unable to compute an accurate rank even if they are more similar to the query images than the clutter images.

A comparison of the performance of the flexible histogram model to the textures-of-textures model can be found in section 5.3. The poor results of the flexible histogram model in these experiments are a result of this.

To improve the representational power of this model, one could consider adding additional features. Without carefully determined thresholds however, the effect of this would be to *narrow* the space of images which are similar to the given model. This effect is seen directly when the texture synthesis procedure is performed with added features (with arbitrary thresholds): almost all the candidate sets contain only a single location, and simple tiling results.

The flexible histogram model differs from previous models (e.g. [4, 5, 33, 6, 12]) in its explicit measurement of the joint occurrence of feature responses across multiple resolutions. By requiring that each element of the parent structures jointly satisfy the condition in equation (3.4) the model achieves its discriminative power.

By increasing the number of constraints described by joint occurrence of features, we can potentially increase the range of visual structures to which the model is sensitive.

An additional problem with the flexible histogram model is that the representation changes from query to query. This is precisely what makes the model "flexible." However, because of this, the computations must be done at query time, excluding the possibility of developing an off-line preprocessing stage to reduce the required online computations. In a large scale image database application performing the flexible histogram computation for every image is prohibitive. Currently, a single query of the 2900 image database takes on the order of half an hour.

In the following sections we describe a model which subsumes many of the joint occurrence constraints exploited by the flexible histogram model. Using an architecture which measures "textures of textures," this new model measures the joint occurrence of features at the same location (as does the flexible histogram model) as well as the joint occurrence of features in neighboring regions. By computing a fixed set of features which are independent of the particular query being performed, this system will allow us to move most of the required computations into an off-line precomputation stage.

4.3 Image database retrieval

Without supplementary information, there exists no way to directly measure the similarity between the content of images. In general, one cannot answer a question of the form: ”is image A more like image B or image C?” without specification in some form of what criteria are to be used to make such a comparison. People perform such tasks by inferring a criterion, based on their visual experience or by complex reasoning about the situations depicted in the images, to use as a measure of similarity between the images. Though they can perform searches for complex or loosely defined images – for example, images depicting “pride” – people typically must examine all, or a large portion, of the database. As the prevalence and size of multimedia databases increases, however, automated techniques will become critical in the successful retrieval of relevant information. Such techniques must be able to measure the similarity between the visual content of natural images.

A digitized image can be interpreted as a single very high dimensional point in pixel space. From this point of view, it is not unreasonable to consider the distance between images in pixel space as a measure of the visual similarity between images. Clearly if two images are very near in pixel space they look similar. Unfortunately images which are far apart in pixel space are often very similar in visual content. What is needed is some sort of “Rosetta stone” which can translate images into another representation which would allow us to interpret and compare them based on their content and visual structure.

Many algorithms have been proposed for image database retrieval. For the most part these techniques compute a feature vector from an image which is made up of a handful of image measurements. Visual or semantic distance is then equated with feature distance. Examples include color histograms, texture histograms, shape boundary descriptors, eigen-images, and hybrid schemes [53, 46, 74, 34, 49, 50, 59].

A query to such a hybrid system typically consists of specifying two types of parameters: the target values of each of the measurements, usually by submitting a query image; and a set of weights, which determine the relative importance of deviations from the target in each measurement dimension.

To deal with the infinity of possible queries a user could want to make of a non-homogeneous image database, these techniques typically rely on a small set of general features which which capture some non-specific properties of images. To allow the user to set the relative importance of each of these properties, they are typically restricted to those characteristics perceived by humans as salient. As a result of their generality however, many images which are actually very different in content, generate the same feature responses and cannot be discriminated. For example features based on color histograms would easily confuse a photo of white paper with a photo taken outdoors in the snow.

For illustration, the features, descriptions and accompanying instructions used by the by QBIC and Virage search engines are shown in Tables 4.1 and 4.2. These two systems, though not the most sophisticated, are currently the most commercially viable image retrieval systems. From these descriptions it is clear that the techniques they use for measuring similarity are based on global and localized color histograms, and simple texture measures. What is also interesting to note, is that the query process is expected to be iterative, as the user varies the relative weights of each technique to try to retrieve the images in which they are interested.

Color Percentages	This method finds images that have similar color amounts to what you specify. Example: If you click on a beach scene with blues and whites, images with approximately the same amounts of the same blues and whites will be returned.
Color Layout	If you click on an image, other images with similar colors in the similar locations will be retrieved. The location of the color DOES matter in this search.
Texture	If you click on an image, other images with similar textures will be retrieved. In the current version of QBIC, texture is computed by measuring the coarseness, contrast, and presence/absence of directionality of each image.

Table 4.1: The features used by the QBIC search engine. Source: Fine Arts Museums of San Francisco (http://206.14.230.208/cgi-bin/QbicStable)

In marked contrast to this generality, another very different approach has been taken in the related problem of image classification. In typical classification tasks, such as character recognition (e.g. [14]) or face recognition (e.g. [51]) the goal is to develop a fully automatic technique which can precisely classify images from very small domain. To accomplish this, various networking schemes are used to derive an extremely sensitive set of features which are fine-tuned to make precise discriminations between different visual structures. The detectors which result after, fine-tuning with training sets containing hundreds to several thousands of examples, for features which do not correspond to notions of visual saliency perceived by human observers. They do however, pick out those features which are best for discriminating between images in the input domain. For example specialized detectors arise to distinguish between a '1' and '7' [14, 56]. In this way these techniques employ a large set of very specialized feature detectors to achieve the accuracy and precision required to discriminate between visually similar image classes. However, the specificity of these features eliminate their usefulness in the more general task of querying over non-homogeneous databases. A '1' versus '7' discriminator, though useful for digit discrimination, has little value when attempting to find pictures of cars, for example.

In this work we attempt to synthesize these two concepts. To achieve the specificity of the classification techniques, features that are highly specialized must be measured; as using very specific features increases the chance that there exists subset of these features are present mostly in the target image class. To simultaneously achieve the robustness required of generalized image database, it must be insured that the set of features is large enough; as it must incorporate all the characteristics of an image that could potentially be a needed criterion for satisfying a query;

This approach is designed to take an image, which in pixel space lies close to many images - only some of which are of the same class - and transform it into a higher dimensional space, so that in this new space, images which are not visually similar lie far apart from one another. Once done, the tasks of finding discriminators between image classes, and of measuring visual similarity, becomes far simpler.

To generate a large set of features which can capture the critical visual characteristics

Color	Virage's VIR Image Engine evaluates the hue, saturation, and tint of an image to determine a generalized color value. It evaluates both dominant color and color variation. Setting Color high generally yields the most expected results. Note, however, that a grayscale image and a color image from the same source might be more closely ranked with a low Color setting.
Composition	Virage's VIR Image Engine evaluates the relative locations of colored areas in an image to determine an overall composition value. Setting Composition high yields results in which images with the same colors, the same amounts of colors, and colors in the same relative areas are ranked together. Since composition often keys more off of how a picture is framed than off of the subject of the picture, you might want to start your query with a low to medium Composition value and gradually increase it to hone in on the image you want.
Texture	Virage's VIR Image Engine evaluates pattern variations within narrow sample regions to determine a texture value. It evaluates granularity, roughness, repetitiveness, and so on. Pictures with strong textural attributes – a sandstone background for example – tend to be hard to catalog with keywords. A visual search is the best way to locate images of these types. For best results, set Texture high when your query image is a rough or grainy background image and low if your query image has a central subject in sharp focus or can be classified as animation or clip-art.
Structure	Virage's VIR Image Engine evaluates the boundary characteristics of distinct shapes to determine a structure value. It evaluates information from both organic (photographic) and vector sources (animation and clip art) and can extrapolate shapes partially obscured. Polka dots, for example, have a strong structural element. For best results, set Structure high when the objects in your query image have clearly defined edges and low if your query image contains fuzzy shapes that gradually blend from one to another.

Table 4.2: The features used by Virage's VIR search engine. Source: Virage Technology Demo (http://www.virage.com/online/help.htm)

for any query, we pay attention both to local texture and global structure. Further, we hypothesize that there is in fact no clear distinction between them: that they are simply two ends of a continuum. Our algorithm represents images at many levels of resolution: measuring color, edge orientation, and other local properties at each resolution. The visual properties captured by these local operations changes at different scales. A horizontal color edge at a high resolution might be related to the leaves of a tree, while a horizontal color edge at a much lower resolution might be caused by a blue sky above a green field. This sort of multi-scale feature analysis is of critical importance. It has been used successfully in the context of object recognition [55, 73].

Our system differs from others because it detects not only first order relationships, such as the edges described above, but also measures how these first order relationships are related to one another. Thus, by finding spatial relationships between image regions with particular local structural organization, more complex – and therefore more discriminating – features can be extracted. Some research efforts have attempted to model a few of these structural organizations explicitly [39]. The Rosetta system does this by building textures of textures. For example, at the highest level of resolution, vertical edge detectors will respond both to skyscrapers and picket fences. At this resolution the two images are not distinguished by the presence of vertical texture. If we examine the spatial organization of the vertical texture we find that picket fences yield horizontal bars of vertical energy. It is the non-linear conjunction of texture and spatial organization that allows our system to distinguish a variety of complex images.

There are tens of potentially useful color and texture features which occur in local regions of natural images. There are hundreds of conjunctive features that can be formed by computing a feature at one resolution and then measuring its structural organization at another. This analysis can be repeated many times: in effect yielding measures of higher order textural organization. There are literally thousands of these multiply conjoined features. Taken together such a representation is called the *characteristic signature* of an image.

No human utilizing such a system can be expected to determine desired values or weights for so many conjunctive features. Instead, a user retrieves images from the Rosetta system by presenting a set of query images. The system computes the desired feature values and weights from this set. Thus, this paradigm can be described as "query by image example." Variations in the feature vectors of the query images are used to determine the relative importance of each image feature in the query. Those features which have consistent values across all the query images receive the largest weights. Weighting in this way causes those features which are consistent within a class to be most important in determining class membership. For example, in one query chromatic-content may be the primary measure, while in another, spatial-arrangement may be dominant.

This system is diagramed in Figure 4-1. Images from from some source are used to generate a database and to choose set of query images, which are indicative of the target images desired. Currently we have performed retrieval over databases and query sets from the top four sources (indicated in red.)

For each query image, a characteristic signature is computed. Statistics over the signatures are computed, and from them a target location and normalization factor in signature space is extracted. The distance of the characteristic signature of each image in the database

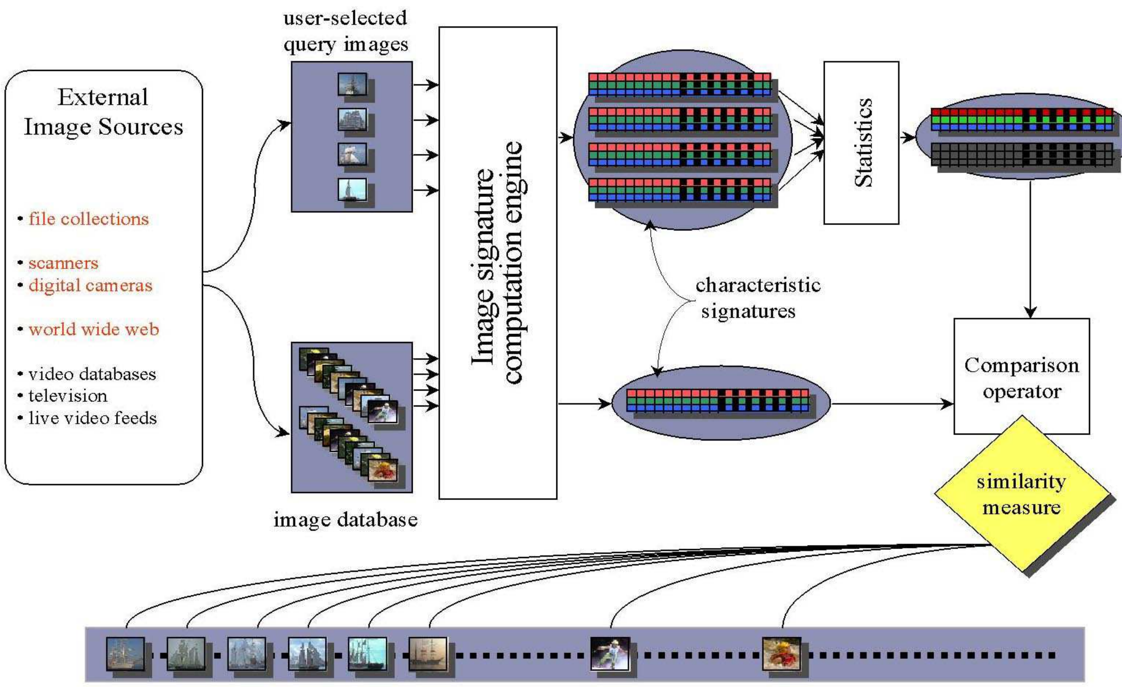


Figure 4-1: A schematic of the image database system presented here.

to this target location (scaled by the normalization factor) is then measured. This measure is then used to rank the images in the database.

4.4 Computing the Characteristic Signature

The textures-of-textures measurements used by the Rosetta system are based on the outputs of a tree of non-linear filter-networks. Each path through the tree creates a particular filter network, which responds to certain structural organization in the image. Measuring the appropriately weighted difference between the signatures of images in the database and the set of query-images, produces a similarity measure which can be used to rank and sort the images in the database. A schematic of this tree of networks is given in Figure 4-2, the leaves of this tree form a *characteristic signature* for an image. Measuring the appropriately weighted difference between the signatures of images in the database and the set of query-images, produces a similarity measure which is used to rank and sort the images in the database.

The computation of the characteristic signature is straightforward. At the highest level of resolution the image is convolved with a set of local linear features. In the experiments in this chapter there are 25 local features including oriented edges and bars. The results of these convolutions are 25 feature response images. These images are then rectified by squaring, which extracts the *texture energy* in the image, and then downsampled by a factor of two. Thus at each level of the tree a 25-way branching occurs, as shown in Figure 4-3. Along each branch the presence of particular visual structure is measured.

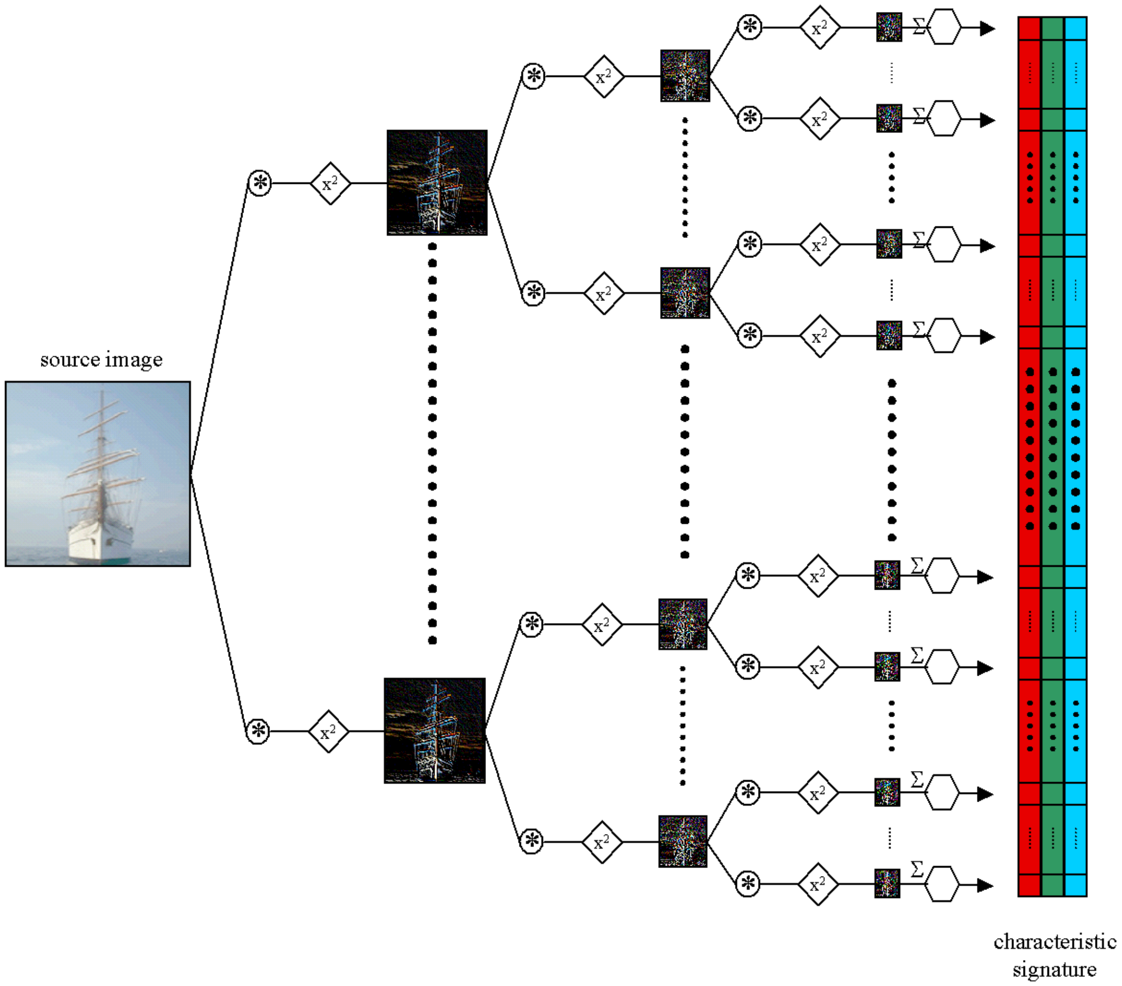


Figure 4-2: A tree of filter-networks is used to compute the characteristic signature for an image

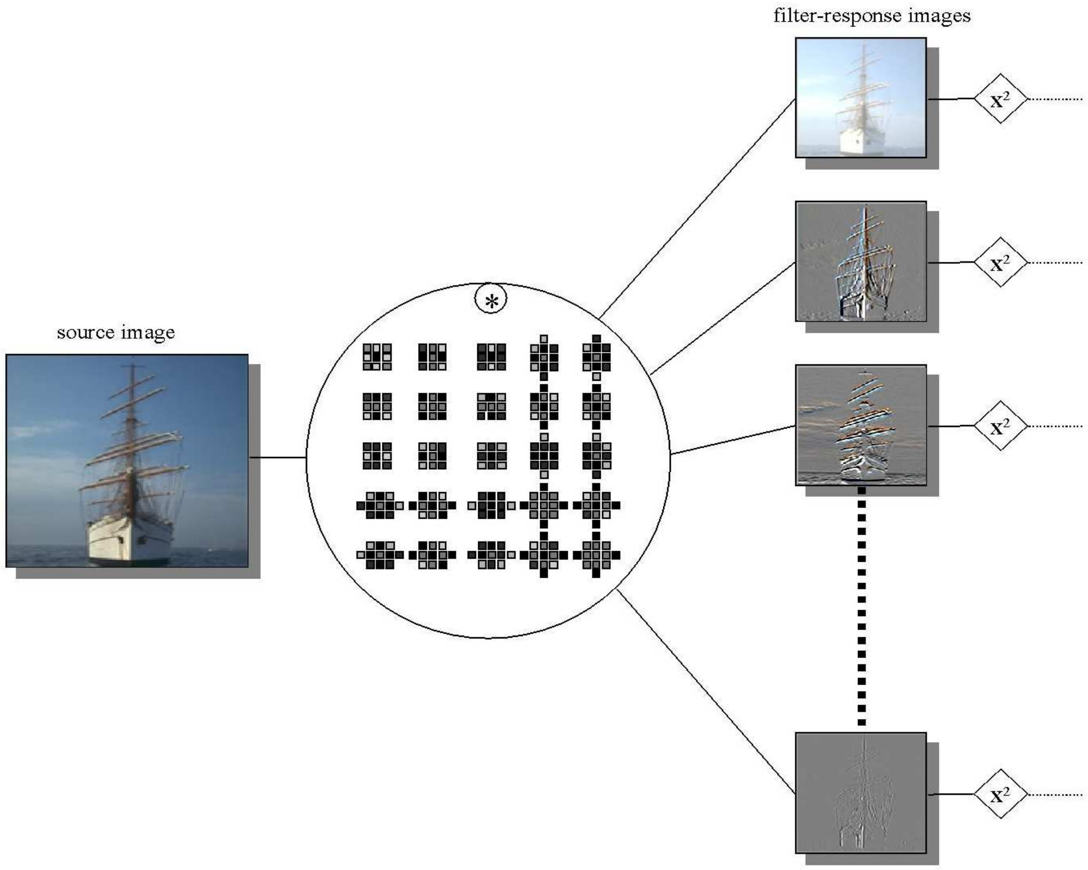


Figure 4-3: Thus at each level of the filter-network tree a 25-way branching occurs, identifying the presence of particular visual structure.

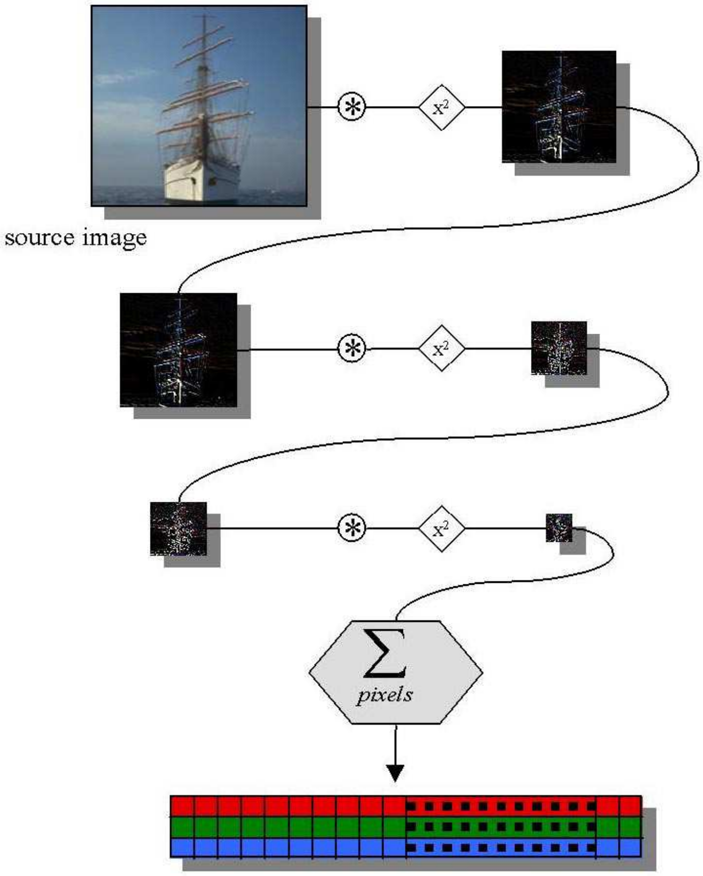


Figure 4-4: A single branch down this tree consists of three levels of convolution, rectification and downsampling, followed by summation.

After a single level the output images consist of texture energy in the image, for example one branch would be sensitive to vertical edges, and would respond to both skyscrapers and picket fences.

This convolution, rectification and downsampling is then repeated on each of these 25 half resolution images producing 525 quarter scale textures-of-textures energy images. With the application of a second layer, the network becomes more specific. When applied to the output of the first layer, the convolution and rectification operation responds to regions where the texture specified in the first layer has some spatial arrangement. For example if the second convolution were sensitive to horizontal bars, the network would be sensitive to horizontal arrangements of short vertical lines, and the network would no longer respond well to the skyscrapers, but would to the fences.

With additional layers additional specificity is achieved; and repeating this procedure a third time yields 15,625 meta-texture feature images at eighth scale.

The aggregate values in each of these images provides one element in the characteristic signature.

A single branch down this tree consists of three levels of convolution, rectification and downsampling followed by summation, and is depicted by Figure 4-4. This process is done independently for each color channel in the input image, creating a signature which contains 46,875 such measurements.

More formally the characteristic signature of an image is given by:

$$S_{i,j,k,c}(I) = \sum_{pixels} E''_{i,j,k}(I_c) \tag{4.1}$$

where I is the image, i, j and k index over the different types of linear filters, and I_c are the different color channels of the image. The definition of E is:

$$\begin{aligned} E_{i,j,k}(I) &= 2{\downarrow}[(F_i \otimes I)^2] && (4.2)\\ E'_{i,j,k}(I) &= 2{\downarrow}[(F_j \otimes E_i(I))^2] && (4.3)\\ E''_{i,j,k}(I) &= 2{\downarrow}[(F_k \otimes E_{i,j}(I))^2]\,. && (4.4) \end{aligned}$$

where F_i is the $i^t h$ filter and $2{\downarrow}(\cdot)$ is the downsampling operation.

4.5 Feature computation

At each resolution in the characteristic signature computation, one of 25 filters are applied. The filtered image is then rectified and down sampled and passed on to the next level of the computation.

Informal observations indicate that the exact form the filters used is not critical in the qualitative performance of the system. For completeness, however, the filters used are described here.

Each of the 25 filters are separable into a horizontal convolution, followed by a vertical convolution. Using the cross application of five horizontal and five vertical filters generates the entire set of 25 filters. The kernels of the five horizontal filters used are:

$$\begin{aligned} h_0 &= \frac{1}{4}[1\ 2\ 1] && (4.5)\\ h_1 &= \frac{1}{2}[1\ -1] && (4.6)\\ h_2 &= \frac{1}{4}[1\ -2\ 1] && (4.7)\\ h_3 &= \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} && (4.8)\\ h_4 &= \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} && (4.9) \end{aligned}$$

The kernels used for the five vertical filters are the transpose of the corresponding horizontal kernel, i.e. $v_i = h_i^T$.

Thus the 25 filters used are described by:

$$F_{5*i+j}\,(I) = v_i \otimes h_j \otimes I \qquad i, j \; \in \; [0, 4] \tag{4.10}$$

Horizontal (vertical) kernel h_0 (v_0) is a discrete Gaussian approximation; kernel h_1 is most sensitive to horizontal edges; h_2 most sensitive to horizontal bars; kernels h_3 and h_4 are sensitive to diagonal bars at 45 and 135 degrees.

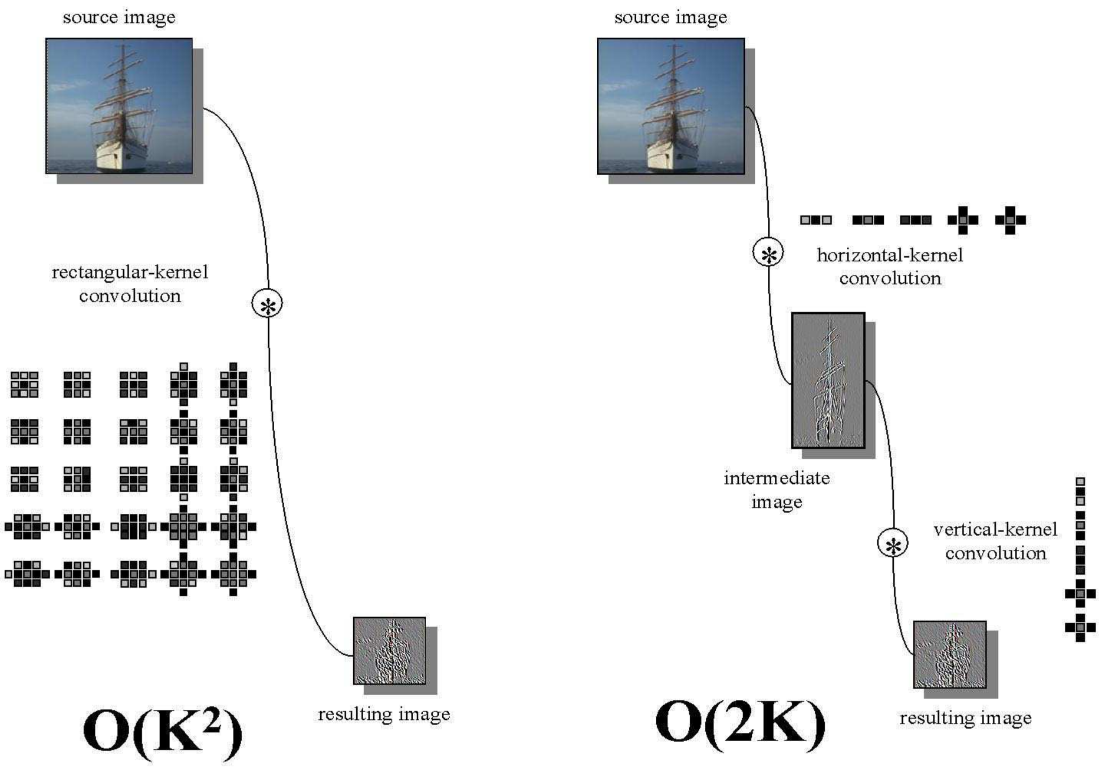


Figure 4-5: Because the 25 convolution kernels were chosen to be separable,each can more efficiently computed by two-half level convolutions

4.5.1 Computational feasibility of characteristic signatures

Computing all the networks independently requires on the order of 600 billion operations per 256x256 image. which clearly unacceptable for use with a several thousand image database.

However, the computation of such a signature becomes feasible when we note that:

$$E_{i,j,k}\left(I\right) = E_{i,\alpha,\beta}\left(I\right) \qquad \forall \alpha, \beta\ \in K \tag{4.11}$$

and

$$E'_{i,j,k}\left(I\right) = E_{i,j,\beta}\left(I\right) \qquad \forall \beta\ \in K \tag{4.12}$$

Which allows for the reuse of intermediate calculations. Geometrically, this corresponds to viewing the complete set of filter-networks as a tree, as depicted in Figure 4-2. By making use of the intermediate computations at each level, the number of computations is reduced to 11 billion per image,

Because the 25 convolution kernels were chosen to be separable, into one of 5 vertical convolutions followed by one of 5 horizontal convolutions, each can more efficiently computed by two-half level convolutions as shown in Figure 4-5. This changes the “big O” cost of each convolution from $O\left(K^2\right)$ to $O\left(2K\right)$, where K is the kernel size. In the present case we use small kernels ($K = 5$) and as a result only obtain a $2.5\times$ speedup. However, we can

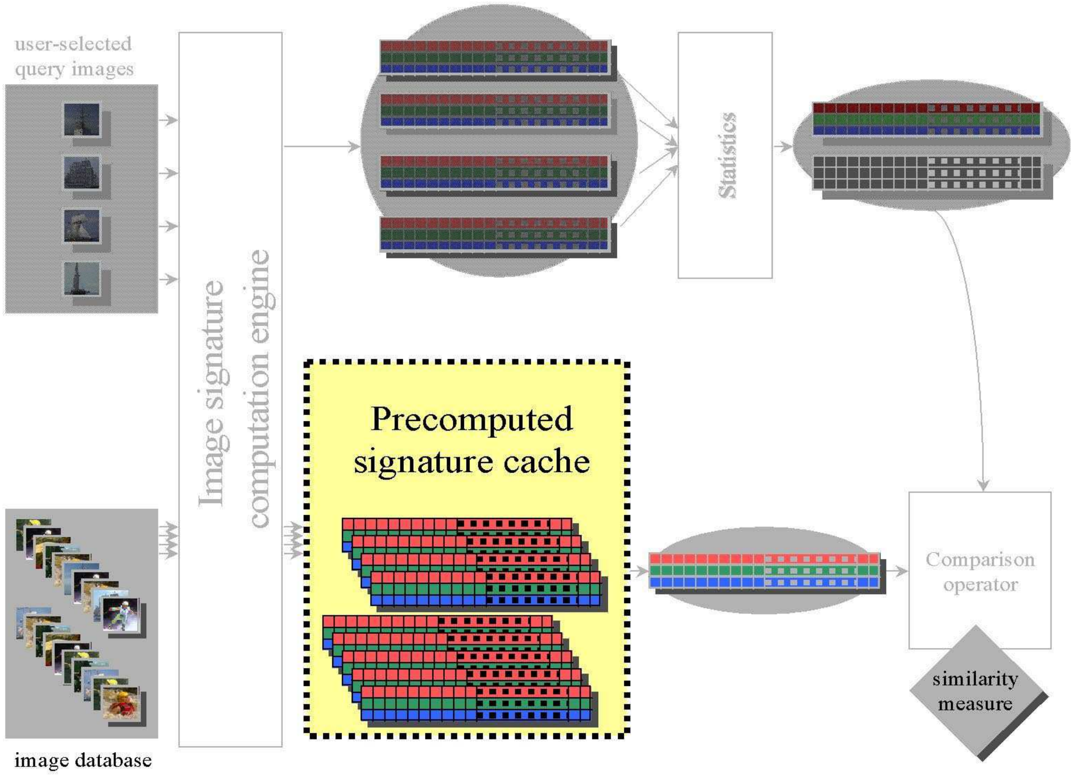


Figure 4-6: Because the characteristic signature for an image is independent of the query images, once computed the characteristic signatures can be cached and used for multiple queries

use the same trick as we did before, and reuse the intermediate results of each half-level convolution.

By exploiting the hierarchical construction of these measurements, the the number of computations required to the characteristic signature for a 128×128 thumbnail image is around 1.1 billion. On a current PC, can be done in the order of tens of seconds, making it a feasible computation to perform once for every image in a database.

Because the characteristic signature for an image is independent of the query images, once computed the characteristic signatures can be cached and used for multiple queries, as shown in Figure 4-6. This ability is a distinct advantage over the flexible histogram model discussed in Chapter 3, which requires that all computations be done at query time. In the current model, the bulk of the computation can be done off-line, leaving only a simple distance calculation to perform at query time. In the Rosetta implementation, such a cache is used. By quantizing the characteristic signatures to two bytes per entry the entire database can be stored in memory (requiring about 300 megabytes.)

4.6 Features captured by the characteristic signature

The effect of repeated application of the filters described in the the proceeding section is to generate a (very large) set of of feature detectors.

The "fence" and "skyscraper" detectors described above are not statistically independent, i.e. neither horizontally arranged vertical edges, nor vertically arranged vertical edges, can occur without the presence of vertical edges. However, the relationship between the responses of the two detectors is complex and non-linear. By measuring each explicitly, we generate feature detectors whose responses are *linearly independent*, and whose individual contributions are important when using a linear discriminant to compare the characteristic signatures of two images.

The overall effect then of the characteristic signature method is to greatly increase the dimensionality of the problem; as $46,625$ elements in the characteristic signature is far larger than the $120 \times 80 \times 3$ dimensions in the original pixel space of the images used in our database. However, in the characteristic signature space many of the complex non-linear interactions between pixels, which are critical in determining the visual characteristics of the image, are made explicit, by transforming them into different dimensions. The intended effect of this is to transform different clusters of images which may fall near of one another in pixel space, far apart in the characteristic signature space. Using linear discrimination methods in this new space, can take advantage of the responses of *any* of the dimensions of the characteristic signature, which make explicit the non-linear relationships in the pixels of the image.

This concept of dimensionality *increasing* is not new, and has been considered in other domains, [58, 62, 56, 57], it is also an implicit assumption in neural network architectures which contain more hidden layers than inputs ([14, 56, 57], for example.)

4.7 Subsumption of the flexible histogram representation

The flexible histogram model presented in Chapter 3 obtains its discriminative power by measuring the joint occurrence of feature responses at multiple resolutions. The current textures-of-textures model subsumes many of the same constraints.

The parent structure used to establish flexible histogram bins is described by equation (2.8) which can be written in an equivalent form using the filters in the textures-of-textures networks:

$$S(x,y) = \begin{bmatrix} F_{12}\left[I(x,y)\right], \\ F_1\left[I(x,y)\right], \\ F_5\left[I(x,y)\right], \\ F_2\left[I(x,y)\right], \\ F_{10}\left[I(x,y)\right], \\ F_{12}\left\{\downarrow F_0\left[I(x,y)\right]\right\}, \\ F_1\left\{\downarrow F_0\left[I(x,y)\right]\right\}, \\ F_5\left\{\downarrow F_0\left[I(x,y)\right]\right\}, \\ F_2\left\{\downarrow F_0\left[I(x,y)\right]\right\}, \\ F_10\left\{\downarrow F_0\left[I(x,y)\right]\right\}, \\ F_{12}\left(\downarrow F_0\left\{\downarrow F_0\left[I(x,y)\right]\right\}\right), \\ F_1\left(\downarrow F_0\left\{\downarrow F_0\left[I(x,y)\right]\right\}\right), \\ F_5\left(\downarrow F_0\left\{\downarrow F_0\left[I(x,y)\right]\right\}\right), \\ F_2\left(\downarrow F_0\left\{\downarrow F_0\left[I(x,y)\right]\right\}\right), \\ F_{10}\left(\downarrow F_0\left\{\downarrow F_0\left[I(x,y)\right]\right\}\right), \\ \vdots \end{bmatrix} \tag{4.13}$$

Where $F_0(\cdot)$ is the Gaussian convolution operation which when followed with the subsampling operation $\downarrow(\cdot)$, as is done in the computation of the characteristic signatures, is equivalent to the downsampling operation $2\!\downarrow\!(\cdot)$.

A subset of the characteristic signature contains feature responses of the form:

$$\left\{ \begin{array}{l} \sum_{pixels} F_{12}\left[I\right], \\ \sum_{pixels} F_1\left[I\right], \\ \sum_{pixels} F_5\left[I\right], \\ \sum_{pixels} F_2\left[I\right], \\ \sum_{pixels} F_10\left[I\right], \\ \sum_{pixels} F_{12}\left(\downarrow\left\{F_0\left[I\right]\right\}^2\right), \\ \sum_{pixels} F_1\left(\downarrow\left\{F_0\left[I\right]\right\}^2\right), \\ \sum_{pixels} F_5\left(\downarrow\left\{F_0\left[I\right]\right\}^2\right), \\ \sum_{pixels} F_2\left(\downarrow\left\{F_0\left[I\right]\right\}^2\right), \\ \sum_{pixels} F_10\left(\downarrow\left\{F_0\left[I\right]\right\}^2\right), \\ \sum_{pixels} F_{12}\left(\downarrow\left\{F_0\left\{\downarrow F_0\left[I\right]\right\}\right\}^2\right), \\ \sum_{pixels} F_1\left(\downarrow\left\{F_0\left\{\downarrow F_0\left[I\right]\right\}\right\}^2\right), \\ \sum_{pixels} F_5\left(\downarrow\left\{F_0\left\{\downarrow F_0\left[I\right]\right\}\right\}^2\right), \\ \sum_{pixels} F_2\left(\downarrow\left\{F_0\left\{\downarrow F_0\left[I\right]\right\}\right\}^2\right), \\ \sum_{pixels} F_10\left(\downarrow\left\{F_0\left\{\downarrow F_0\left[I\right]\right\}\right\}^2\right) \end{array} \right\} \subset \text{ characteristic signature} \tag{4.14}$$

Which extracts features similar to the top three resolutions in equation (4.13), except for the non-linear squaring operation which is applied after each level. The rectification provided by squaring extracts energy, and is necessary to prevent neighboring regions of the

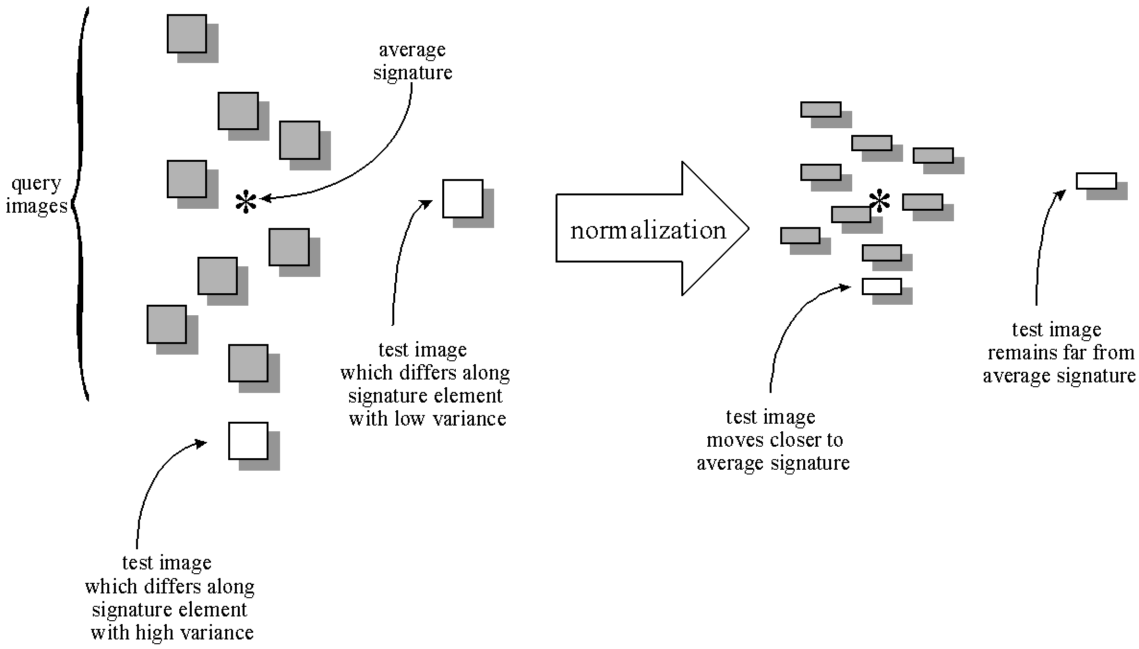


Figure 4-7: Normalization using inverse variance of that element across the query-image group allows elements which are salient within the group of query-images contribute more to the overall similarity measure.

same edge from "destructively interfering" with one another during the aggregation stage. By aggregating the responses of each filter, some of the specialization of the individual flexible histogram bins is lost, as only the mean value of each feature is retained; however, the memory required to keep the full feature response images is prohibitive.

4.8 Using Characteristic Signatures To Form Image Queries

In our image query paradigm, we describe similarity in terms of the difference between a database-image and a group of example query-images. This is done by comparing the characteristic signature of each image in the database to the mean signature of the query-images. The relative importance of each element of the characteristic signature in determining similarity is proportional to the inverse variance of that element across the example-image group. This is a diagonal approximation to the Mahalanobis distance [20]. The full Mahalanobis distance cannot be used because it requires the computation of $46,625^2$ numbers, which is beyond the memory limitations of the computers currently available. Using the diagonal approximation to the Mahalanobis distance has the effect of normalizing the vector-space defined by the characteristic signatures (independently along each of its dimensions), so that characteristic elements which are salient within the group of example-images contribute more to the overall similarity of an image. The effect of this normalization is shown in a two dimensional space in Figure 4-7. Notice scaling occurs only along a cardinal axis; using the full Mahalanobis distance would allow scaling along a diagonal.

The similarity between an image and the group of query-images is the negative of the

Sunsets and Sunrises	Mountains of America	WW II Planes
Christmas Celebration	Coasts	Wild Animals
Sailboats	Birds	Trees and Leaves
Air Shows	Patterns	Underwater Reefs
Flowers	The Arctic	China and Tibet
Rural Africa	Ireland	Western Canada
Arizona Desert	Spirit of Buddha	Auto Racing
Bridges	People	Churches
Food	Lakes and Rivers	Waterfalls
Fields	Exotic Cars	

Table 4.3: The classification labels of the 29 Corel photo CD's which comprise the image database.

sum squared difference between the average query-image signature and the database-image signature weighted by the variance of across the query-image signatures:

$$L = -\sum_i \sum_j \sum_k \sum_c \frac{\left[\overline{S_{i,j,k,c}(I_q)} - S_{i,j,k,c}(I_{test})\right]^2}{Var\left[S_{i,j,k,c}(I_q)\right]} \tag{4.15}$$

Given that we only have a few examples of images in the target class, we expect our estimation of the variances on each dimension to be somewhat inaccurate and *unstable*, as we are trying to approximate the variance over $46,625$ number with only 3 to 5 examples of each. Clearly, with more example images, we can obtain a better estimate of both the variances and means of the target region in characteristic signature space. However, in practice normalization improves similarity measurements, even with just a few query images.

4.9 Experiments

An image database query system must retrieve images which are similar to those for which the user is searching. Because the concept of *similarity* in the goal above is not well defined it is difficult to quantify query results.

We used a database of 2900 images from 29 Corel Photo CD (collections 1000-2900.) Each CD contains 100 images which have been categorized by theme. Examples of these themes include "Sunsets & Sunrises," and "Mountains of America," as well as less specific collections such as "Spirit of Buddha," or "Christmas Collection," and classes which contain images which are very similar, i.e. "Exotic Cars," and "Auto Racing." Each image has been placed exclusively into one category, however, some could reasonably belong in multiple categories. For example, consider categorizing an image depicting a sunrise over the Rockies, or a 1967 Porsche. A complete list of the 29 CD titles are found in Table 4.3. Because of this lack of mutual exclusion between true category membership, we would not expect any image query system – or human – to exactly select the same images for a category as did the original classifier.

Figures 4-8 through 4-11 show the results of typical user queries on this system. The

Figure 4-8: TOP: A sample query intended to return images of cars. BOTTOM: The top 30 responses found by the Rosetta system.

top windows in each Figure contain the query-images submitted by the user. The bottom windows show the thirty images found to be most similar; similarity decreases from upper left (most similar) to lower right. Though these examples provide an anecdotal indication that the system is generating similarity measures which roughly conform to human perception, it is difficult to ascertain from them a quantitative evaluation of the system. To better measure the performance of the system two experiments on this database were performed.

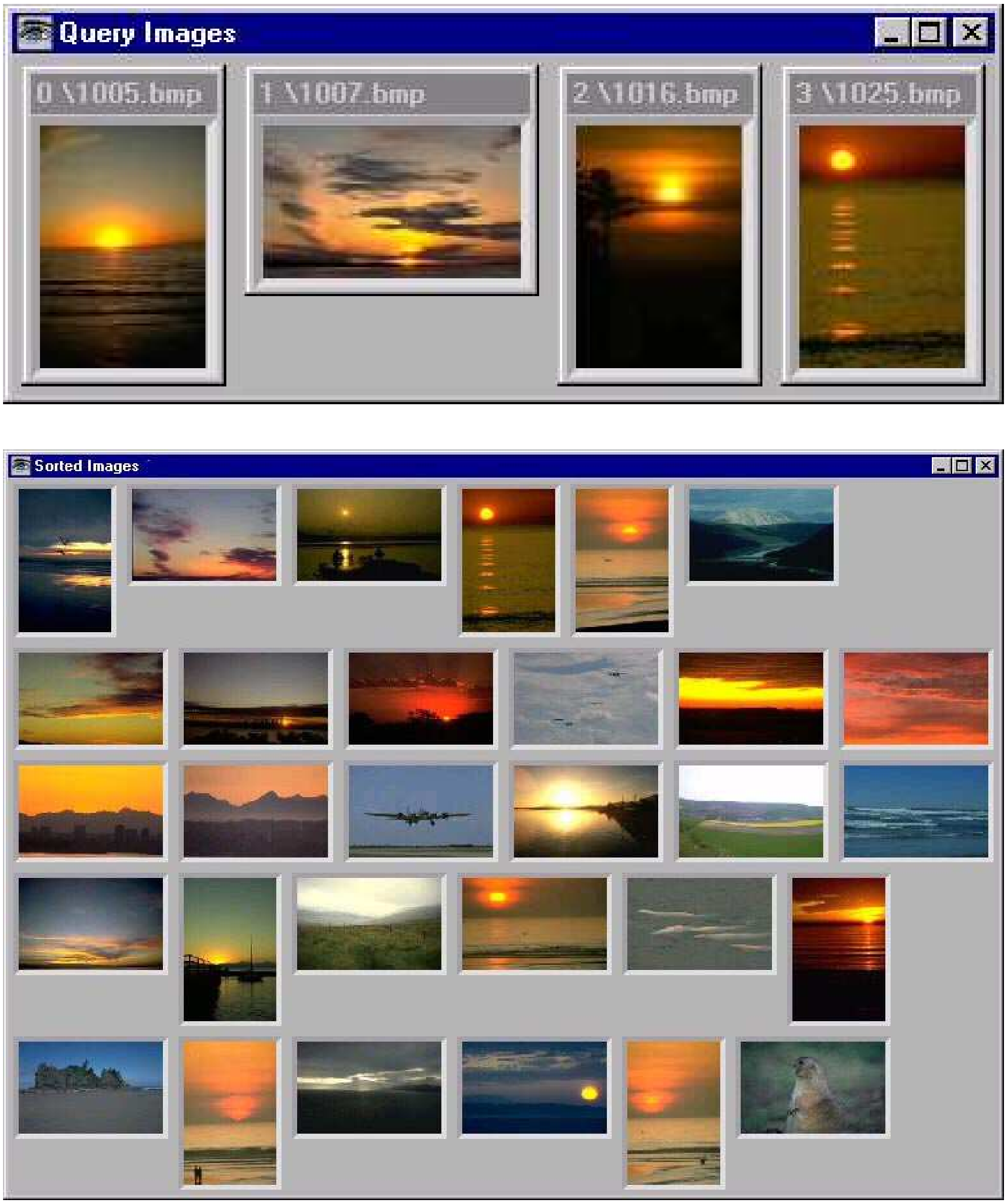


Figure 4-9: TOP: A sample query intended to return images of cars. BOTTOM: The top 30 responses found by the Rosetta system.

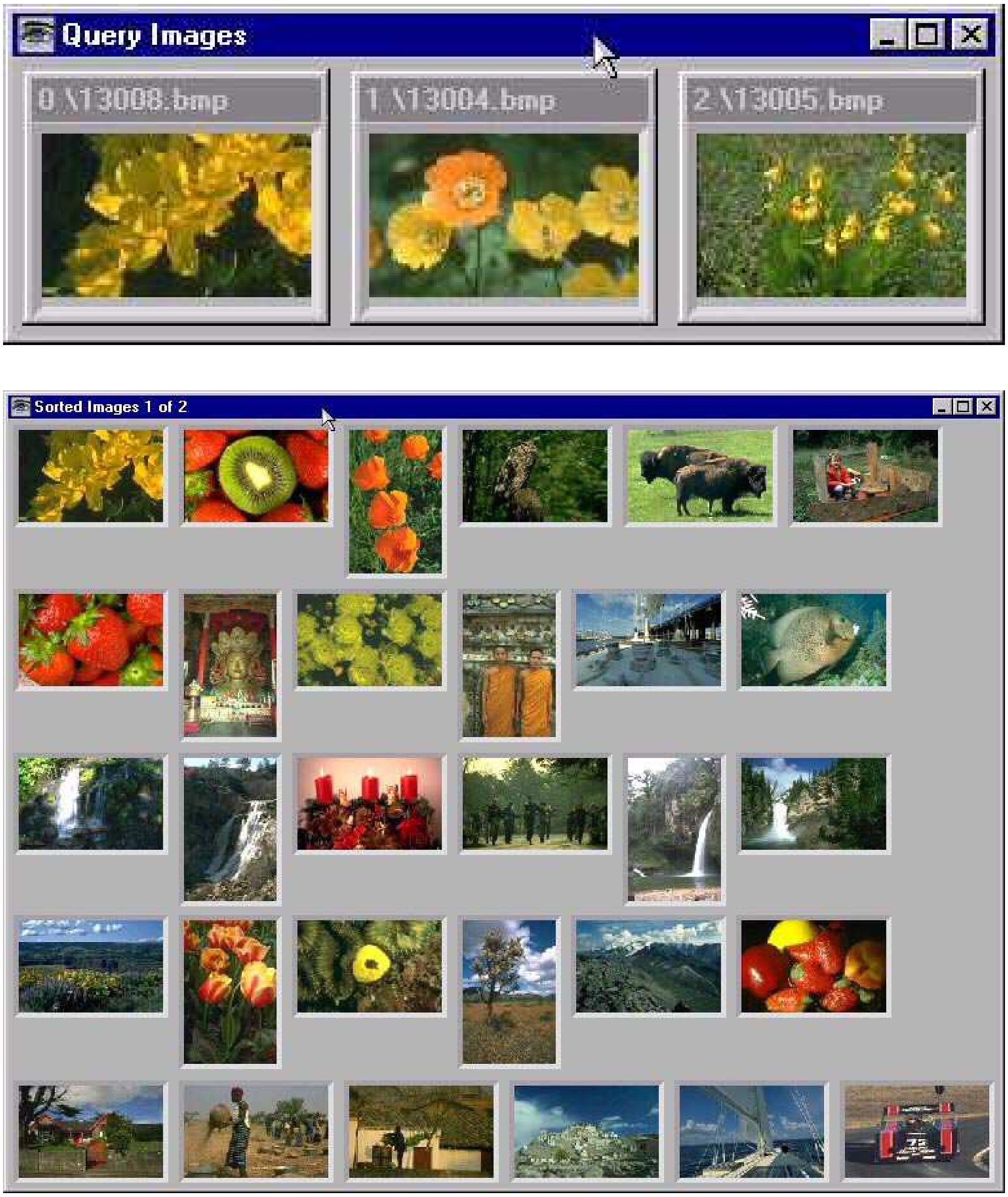


Figure 4-10: TOP: A sample query intended to return images of cars. BOTTOM: The top 30 responses found by the Rosetta system.

Figure 4-11: TOP: A sample query intended to return images of cars. BOTTOM: The top 30 responses found by the Rosetta system.

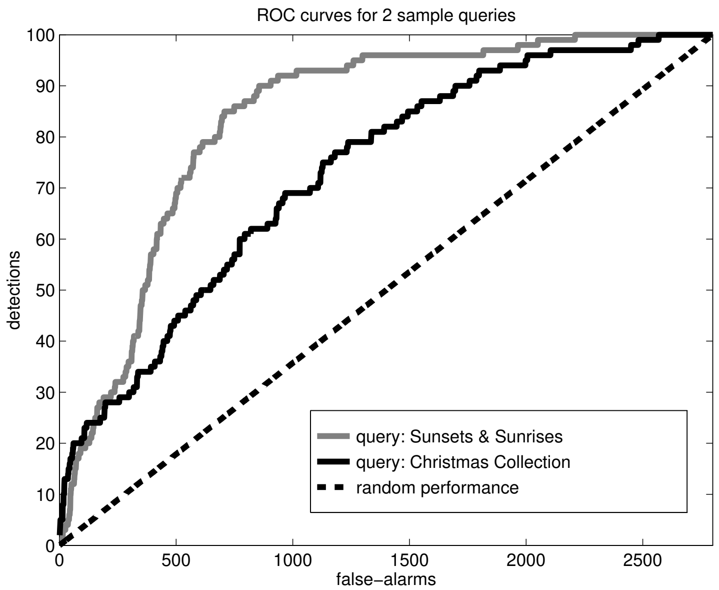


Figure 4-12: The ROC curve for two queries.

4.10 Experiment 1

The images in each collection were selected because they were determined by a human observer to be representative examples of the theme of the collection. Therefore it is reasonable to take a few images from a category and to hope to use them to retrieve other images from that category. The ranking of the true images in that category provide a measure of success.

The results of two queries are shown in the receiver operating characteristics curve in Figure 4-12.

For each query four images were randomly chosen from a single image collection. The similarity was then measured between this query set and all the images in the database. The number of images from the target collection is plotted against the number of other images as a function of image similarity.

The top (grey) curve was generated by a query for images from the "Sunsets & Sunrises" collection, which contains images which all share common visual characteristics. Two of the images from this collection are shown in Figure 4-13a. Though there is significant chromatic variation between the various images in this group they all share very similar structural characteristics.

The middle (black) curve in Figure 4-12, shows a query from the "Christmas collection," whose images contain far more visual variety. Two typical images from this collection are shown in Figure 4-13b. Because of this increased variety one would anticipate poorer performance for such a query. However, performance is still significantly better than chance, which is indicated by the diagonal dashed line. This is due to the fact that though there is significant variety, many of the images do still contain similar structures.

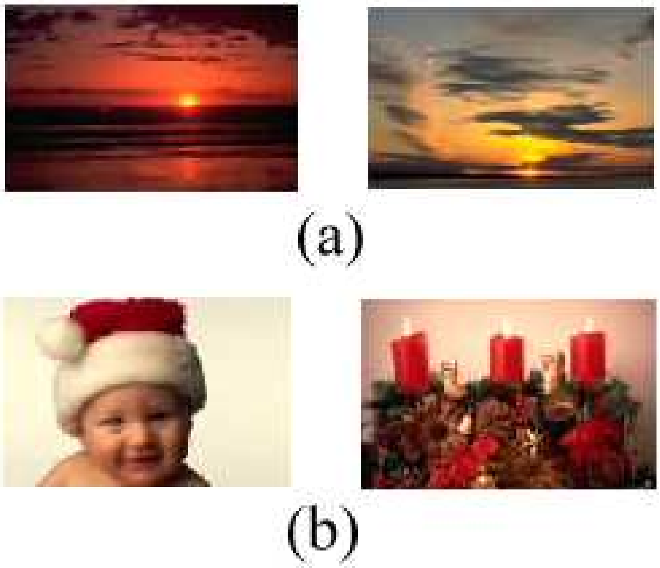


Figure 4-13: Example images from two categories in the database.

4.11 Image Classification

A second measure of the Rosetta system's performance is obtained by measuring how well the system can discriminate between two classes given a few examples from each.

Using examples from only the "Sunsets & Sunrises" collection, and attempting to classify images from both that collection and from the "Christmas Collection" the black curve in Figure 4-14 is obtained. On this plot, the number of correctly classified images (number of detections) is plotted against the number incorrectly classified (false-alarms.) From this curve we can see, for example, that if searching for "Sunsets & Sunrises", about of 32 out of the top 50 responses would be correctly classified yielding an accuracy of only 64 percent. Chance, represented by the dashed line is 50 percent. If however, we present the system with examples of each collection, the grey curve in Figure 4-14 is obtained. On this curve 43 out of the top 50 are correctly classified, yielding 86 percent accuracy. Because of the large variation within the image collections, as discussed above, performance drops off steadily as we consider retrieving successively larger portions of the target collection.

Figure 4-14 suggests that not only can iterative query refinement be applied to the current model, by indicating which of the retrieved images are positive examples, but also that we can build models of negative examples and use this classification paradigm to improve performance. The Rosetta system with the proposed extension of positive and negative iterative refinement is depicted in Figure 4-15, however this this extension has not yet been added to the current system.

4.12 Discussion

We have presented a technique for approximating perceived visual similarity, by measuring the structural content similarity between images. We have developed a system called

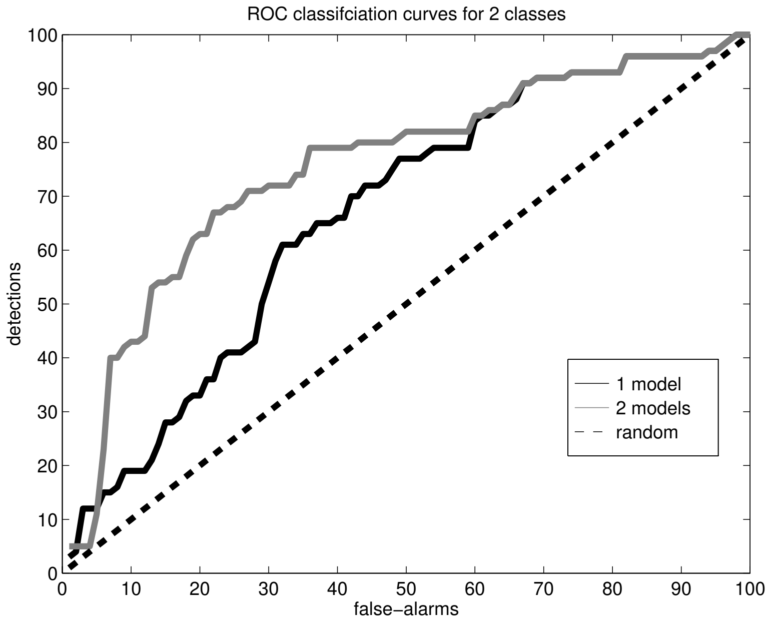


Figure 4-14: Using examples from multiple classes (grey curve) improves performance over examples from just one class (black curve.)

Rosetta which transforms images into a very high dimensional "characteristic signature" space which captures the visual structure in the image. Using this representation, the present system directly compares database-images to a set of query-images.

A word wide web version of this system has been created:

`http://www.ai.mit.edu/~jsd/Research/ImageDatabase/Demo`

Experiments indicate that the present system can retrieve images which share visual characteristics with the query-images, from a large non-homogeneous database. Because the characteristic signature space incorporates structural information, it can perform queries where simpler methods, such as color histogramming fail. Though the results of queries using the Rosetta system are encouraging, they are not perfect – as evidenced by the false alarms in Figures 4-8 through 4-11 – we believe that with additional research its performance will improve.

In experiment 2 we demonstrated retrieval performance can be improved by modeling distracting images as a separate class. This suggests a that not only can iterative reinforcement be applied to the current model by indicating which of the retrieved images are positive examples, but also that we can build models of negative examples and use this classification paradigm to improve performance.

Using this image representation we would like to build associations between words, or phrases and collections of images. Given groups of images which have all been labeled with the same phrase, the average characteristic signature (and variance normalization vector) can be computed, and associated with that label. Queries could then be performed using these combinations of these labels, by recalling and combining the associated statistics. Such an extension to this system is diagramed in Figure 4-16; however, establishing groups of images which are indicative of the visual representation of a large set query phrases is

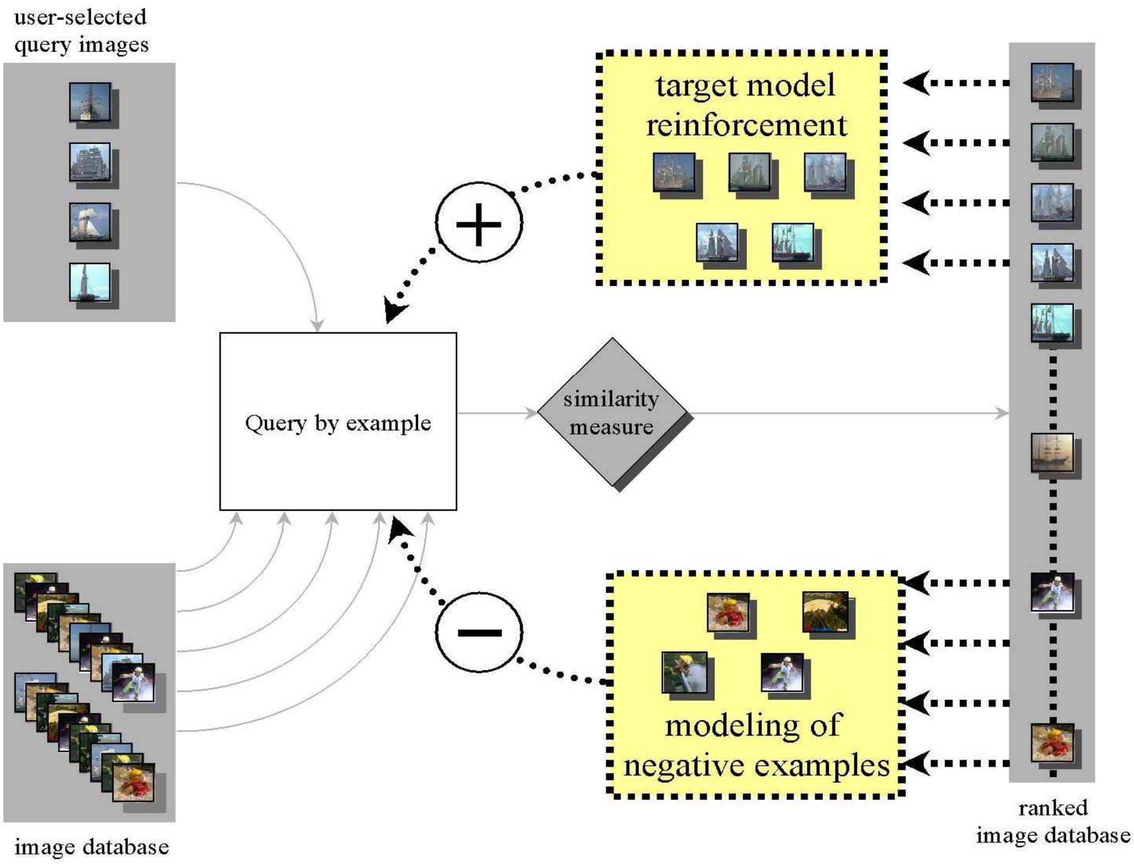


Figure 4-15: Iterative refinement can be applied to the current model by indicating which of the retrieved images are positive examples, or negative examples

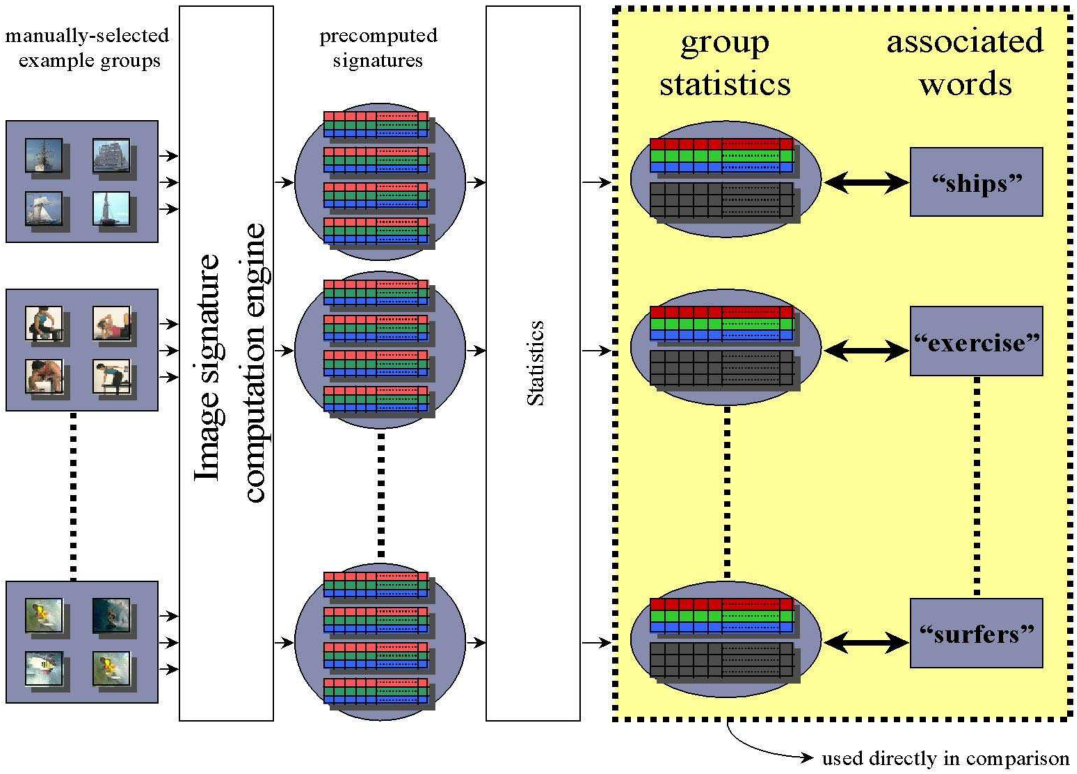


Figure 4-16: In future work we will build associations between words, or phrases and the mean characteristic signature and normalization factors for collections of images.

a formidable task. Ideally such associations could be learned from word and image co-occurrence in multimedia documents, such as sites on the world wide web.

Chapter 5

Retrieval Performance Evaluation Experiments

To a measure the performance of the textures-of-textures image database system a series of experiments was done. In each experiment, we searched for a set of target images given two examples from that set. We compare the performance of the present system to ten other techniques. Though these techniques are not as sophisticated as those used in systems developed by other researchers, they are indicative of the types of methods upon which they are based.

It is difficult to fully characterize the performance of the image retrieval technique. This is a fundamental problem of the domain. Images vary from each other in an astronomical number of ways, and similarity is perceived by human observers based upon complex interactions between recognition, cognition, and assumption. It seems unlikely that an absolute criterion for image similarity can ever be determined, or if one truly exists. However, if we can establish sets of images which we believe are visually similar, we can establish a basis for comparing algorithms.

In each experiment we measure the retrieval rates for a set of ten target images which we believe to be visually similar because they consist of images of a single scene, and differ because of single type of variation. There are two classes of variations which we examine. In the first class, a set of images is generated from a single image which has been altered to varying degrees by one of several image manipulation routines. The second class of image set consists of images of the same subject taken under a variety of physical variations. The base image for all of these variations is shown in Figure 5-1.

In each experiment we perform 45 database queries are made using every possible pair of images from the target set as query images. Retrieval performance is measured using ROC curves which are averaged across all queries. Each query is performed on the entire 2900 Corel image set used in Chapter 4 plus the images in the target set. Since each target set contains different manipulations of the same base image, obviously only one target set is included at a time.

In each query we compare retrieval rates for the following techniques:

1. **ToT-25ˆ3** The current textures-of-textures system using 25 filters at each of 3 levels in each filter network.

Figure 5-1: The base / canonical image used in generating target image sets

2. **RGB-216C** R,G,B color histograms using 216 bins by dividing each color dimension 6 parts. The target histogram is generated by combining the histograms from the two model images.

3. **RGB-512C** R,G,B color histograms using 512 bins with a combined histogram model. Histograms with different bin sizes are included to show the sensitivity of these techniques on bin size.

4. **HSV-512C** H,S,V color histograms using 512 bins with a combined histogram model. The hue, saturation and value (HSV) color space was designed to more accurately capture the differences in colors perceived by human observers [22].

5. **HSV-216C** H,S,V color histograms using 216 bins with a combined histogram model.

6. **RGB-216NN** R,G,B color histograms using 216 bins, in which similarity is measured by the minimum distance (nearest neighbor) between the test histogram and each of the histograms for the two model images.

7. **RGB-512NN** R,G,B color histograms using 512 bins and a nearest neighbor measure.

8. **HSV-512NN** H,S,V color histograms using 512 bins and a nearest neighbor measure.

9. **HSV-216NN** H,S,V color histograms using 216 bins and a nearest neighbor measure.

10. **COR-fullres** Full resolution image correlation, in which similarity is measured by the maximum correlation (nearest neighbor) to each of the model images separately.

11. **COR-fullres** $4\times$ downsampled image correlation, in which similarity is measured by the maximum correlation (nearest neighbor) to of each of the $4\times$ downsampled model images separately.

The details of the supplementary techniques are found in Appendix A

5.1 Image manipulation performance experiments

5.1.1 Performance Experiment: Brightness

In the first experiment we consider a target set of images constructed from a single image by variation of its brightness. The 10 images in the target image set are shown in Figure 5-2. Images increase in brightness from left to right and top to bottom. Clipping was used to handle brightness shifts which would result in pixels outside of the displayable gamut. Images in the data set were generated by varying each pixel with the following function:

$$I_\gamma\left(x,y\right)_{r,g,b} = \max\left\{\min\left[I\left(x,y\right)_{r,g,b} + \gamma, 255\right], 0\right\} xs \quad (5.1)$$

To generate the target set, γ is varied from -50 to $+50$ by steps of 10.

Receiver operating curves for each technique are shown in Figure 5-3. Percentage of target images found is plotted as a function of number of clutter images misclassified as target images. The number of clutter misclassifications is plotted on a log scale to emphasize the performance region about which we typically care most in retrieval applications. In typical applications the Neyman-Pearson criterion, which restricts the maximum number of misclassifications, is very low.

On these plots, chance performance is indicated by the dotted curve from $\left(P\left(0\right), P\left(0\right)\right)$ to $\left(P\left(100\right), P\left(100\right)\right)$. Each cure is restricted, by construction, to be (not-strictly) monotonically increasing. Perfect performance would be a curve which passes through $\left(P\left(100\right), P\left(0\right)\right)$ and travels along the top of the graph to $\left(P\left(100\right), P\left(100\right)\right)$. For further discussion of the general nature ROC curves see section 3.7.1.

The best performance was achieved by the present textures-of-textures model, which generated the top curve. For almost any number of retrieved images, the current model returns the highest percentage of target images. The performance of each system under some Neyman-Pearson criterion, which would state that we are only willing to consider N falsely retrieved images before quitting, can be read off of the intersection of its ROC curve with a vertical line at N. For typical large scale applications N tends to be small, because human interaction is required to tease out false-positives. For example, with a modest Neyman-Pearson criterion of $N = 50$ false-positives, the current system retrieves over 90% of the target images.

As we expected the color histogram techniques were unable to handle the variation caused by manipulation of the images' brightness. With 512 bins, varying all the pixels by more than 32 in red, green, or blue, will guarantee that each pixel falls into a different histogram bin. Similarly changing red, green, and blue, by more than 32 shifts the brightness (or *Value*) of the pixel by more than 32 moving it into a new H,S,V histogram bin. In each

Figure 5-2: The 10 target images used to measure performance with respect to variations in brightness

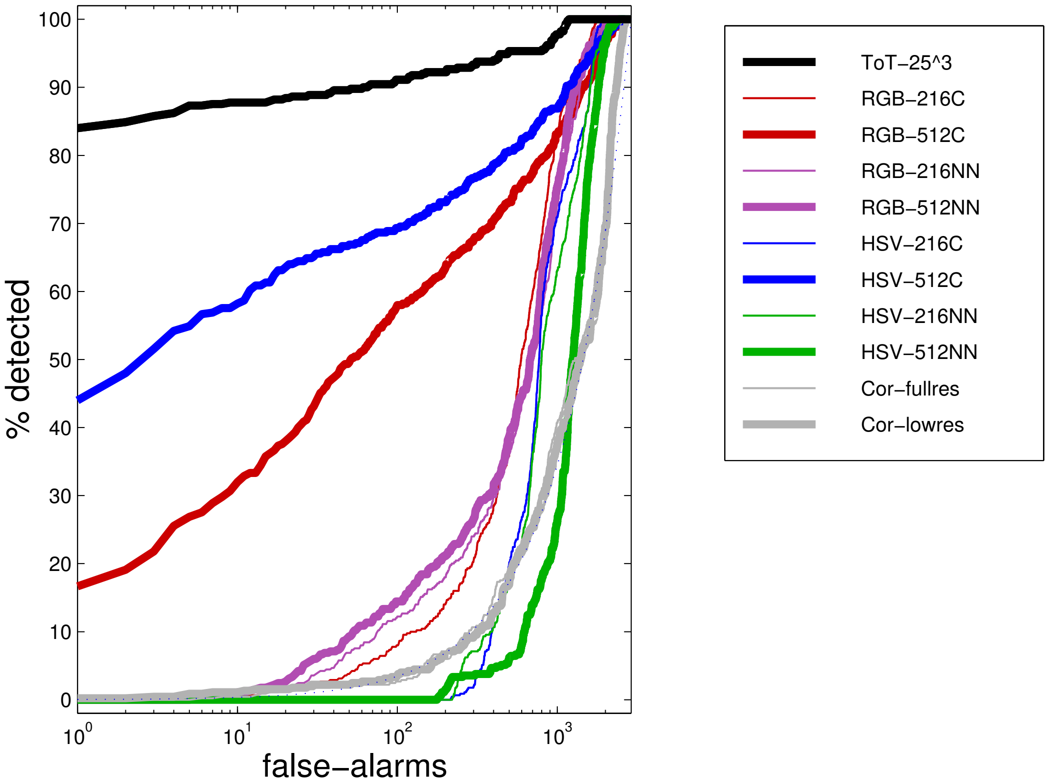


Figure 5-3: ROC curves for retrieval performance of each technique measured with respect to variations in brightness

case a change of $\Delta\gamma = 10$ will cause roughly one third of the pixels to shift into different bins.

As a result the histograms generated by each of the images in the target set are very different from one another.

However, because the frequency spectrum is of natural images is typically $1/f^2$, there is much more energy in low spatial frequencies. As a result pixel color variations will most often be low frequency, and will tend to vary smoothly. Neighboring pixels therefore, will tend to vary slowly in brightness from a given pixel [15].

As a result, when the brightness is changed and some pixels move out of a particular bin, some of its neighboring pixels are likely to replace them. Inasmuch as neighboring (in brightness) histogram bins in the original image tend to be near the same levels, brightness-modified images will generate similar histograms. With a Neyman-Pearson criterion of at most 50 false-positives, the HSV-512C histogram achieves 63% retrieval of the target images. At the same point, the RGB-512 histogram achieves only about 52% retrieval of the target images.

The color histograms with 216 bins ($\{$RGB, HSV$\}$-216$\{$C, NN$\}$) have larger bin sizes (42 in each dimension). This makes them more stable with respect to the variation caused by brightness shifting. However, the stability afforded by larger bins makes the histograms less discriminating, and thus more likely to be confused with the histograms generated from the clutter images. As a result their overall performance for each colorspace and model type achieve levels of performance similar to that of the 512 bin histograms. With a Neyman-Pearson criterion of at most 50 false-positives, most of the HSV and RGB histograms drop below 20% retrieval of the target images.

Two color techniques, RGB-216NN and HSV-512C stand out however, and perform better relative to the other methods, though also at levels well below the performance textures-of-textures model. However, the separation of these two histogram techniques from the others, is indicative of the sensitivity of such techniques to the specific bin choices. In effect, a type of *bin aliasing* occurs, causing wildly different response levels in a few isolated cases. The color distribution present in the canonical image seems to be near one of the chaotic regions in both the RGB-216NN and HSV-512C techniques. In several of the experiments below, their performance differs significantly from the other color histogram techniques.

Texture measures which are only sensitive to relative values of neighboring pixels, are almost completely unaffected by global (i.e. very low frequency) brightness shifts. Only saturation caused by the $\max$ and $\min$ terms in equation (5.1) cause variation in their response. As a result, the representation used by the present system is roughly invariant to changes in brightness, resulting in good retrieval performance.

Correlation based techniques are strongly affected, as a global brightness change of γ causes a $N \times M$ (γ^2) increase in the level of the (L_2) difference measure.

5.1.2 Performance Experiment: Contrast

In this experiment we use the same base image as in section 5.1.1 (shown in Figure 5-1) and vary its contrast. In Figure 5-4 (a) the lowest contrast image is shown, in (b) the highest

Figure 5-4: The 10 target images used to measure performance with respect to variations in contrast

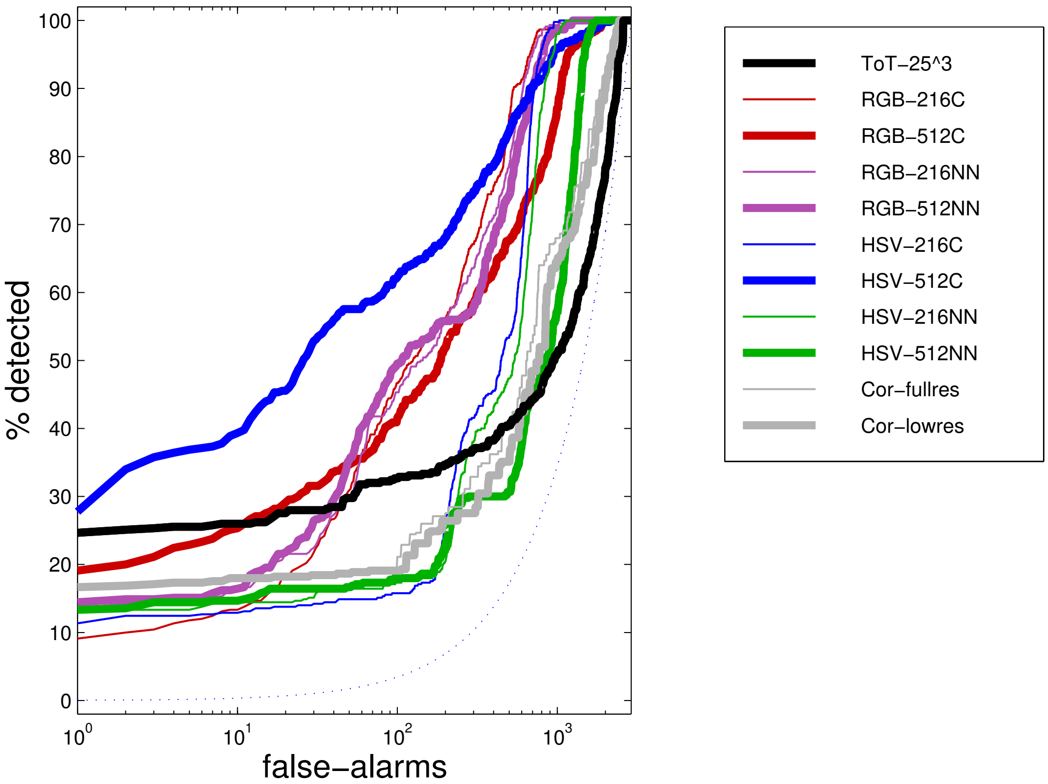


Figure 5-5: ROC curves for retrieval performance of each technique measured with respect to variations in contrast

contrast. Images in the data set were generated by varying each pixel with the following function:

$$I_\gamma\left(x,y\right)_{r,g,b} = \max\left(\min\left\{\left[I\left(x,y\right)_{r,g,b} - 127\right] \times \gamma + 127, 255\right\}, 0\right) \quad (5.2)$$

To generate the target set, we use $\gamma \in \{0.9^0, 0.9^1, \ldots, 0.9^{10}\}$. Receiver operating curves for each technique are shown in Figure 5-5.

The current system achieved its worst performance in this experiment, achieving little better than chance retrieval. This degradation in performance occurs because the first level of each filter network is highly sensitive to the oriented and unoriented energy present in the image. With a decrease in image contrast, each filter in the first level returns a weaker response to the structure within the image. With a Neyman-Pearson criterion of at most 50 false-positives, the ToT-25^3 curve returns only about 40% of the current images.

This shortcoming suggests several possible extensions to the model which could possibly fix this vulnerability. An initial stage which performed some sort of contrast equalization before computation of the characteristic signature could decrease this susceptibility; however, it is not clear how such equalization would negatively affect the discriminative power of the model.

An alternative solution could be to insert a thresholding non-linearity, such a sigmoid, after each filtering operation. This technique is used by LeCun *et al.* [14], and has the effect of roughly quantizing the response of each filter. After the application of a sigmoid, the response images would be roughly binary and simply indicate the presence or absence of an above threshold amount of oriented energy in the original. We anticipate that such an extension would improve the performance of system in lower Neyman-Pearson criterion measurements, and eventually catastrophically degrading when energy falls below the sigmoidal threshold.

A third solution is to normalize the total response of the characteristic signature . This would have the effect of forcing all the characteristic signatures to fall on the surface of a sphere. When comparing normalized characteristic signatures , only their orientation is significant. Because the characteristic signature generated for an image with reduced contrast is shorter than the characteristic signature for the original, but has the same orientation, normalizing their lengths will make them identical.[1] We consider this extension in Chapter 7.

For the 216 bin color histograms performance improves relative to its performance in the brightness variation experiment (section 5.1.1.) as the color variations caused by manipulation of contrast is less than that caused by brightness variation. Under this manipulation, color shifts are small enough that many of the pixels fall into the same bins after contrast manipulation. However, the shifts are sill large enough to cause the pixels to change bins in the 512 bin histograms. Further, contrast variation tends to cause "clumping" of most pixels into a smaller number of bins, thus decreasing the number of bins which can help uniquely identify images in the target set. As a result, performance of the 512 bin histograms degrades. (This effect is occurring in the 216 bin histograms as well, but is counteracted by the large performance boost they achieve due to the fact that the color shifts are smaller; as

[1] Except for roundoff error which introduces non-invertible effects.

a result, the 216 histogram ROC curves show an overall performance increase.)

Because the shifts are smaller, the correlation based techniques also achieve better performance relative to the brightness variation series. However, their performance is still little better than chance.

5.1.3 Performance Experiment: Noise

In this experiment we use the same base image as in section 5.1.1 (Figure 5-1) and add Gaussian white noise. Because adding noise of a particular variance can generate many possible modified images, we generate three target sets, for different noise variances σ.

The images in a set were generated by adding noise with variance σ to each pixel, separately to each RGB channel:

$$I_\sigma(x,y)_r = I(x,y)_r + 255\nu_r \quad (5.3)$$
$$I_\sigma(x,y)_g = I(x,y)_g + 255\nu_g \quad (5.4)$$
$$I_\sigma(x,y)_r = I(x,y)_b + 255\nu_b \quad (5.5)$$

where ν_r, ν_g, and ν_b are independently chosen for each pixel from a normal distribution, $N(0,\sigma)$.

We generated 3 sets for $\sigma = \{0.25, 0.5, 0.75\}$, corresponding to average signal to noise ratios of 48.07,3.38,and 0.94 [54]. These images are shown in shown in Figures 5-6 through 5-10. Corresponding receiver operating curves for retrieval of target set with added noise of each variance shown in Figures 5-7 through 5-11.

With all three levels of noise, the characteristic signature representation used by the textures-of-textures model is virtually unaffected, as evidenced by its achievement of perfect performance in each. From the exceptional performance of the downsampled correlation technique (COR-lowres) we see that the conglomeration of neighboring values effectively cancels out the effect of the Gaussian noise which is added independently to each pixel.

In the ToT-25ˆ3 model, each filtering operation similarly combines information from local neighborhoods of pixels. As a result the texture energies are largely unaffected. Further, at each successive level increases the size of the neighborhood over which information is integrated, further increasing its stability with respect to noise.

Under each successive amount of image degradation, performance decreases to varying degrees for each color histogram technique. The catastrophic behavior seen in the ROC curves for the color histogram retrieval methods is indicative of the sensitivity of these models to the precise values of each pixel. With the addition of Gaussian noise, the images in the target set are separated by fixed distances in histogram space. As a result queries using any these disparate points in histogram space as a model result in ten quantized similarity measurements (one for each image in the target set.) These ten levels are interleaved with the similarity measures of the clutter images, resulting in the discrete steps taken in the ROC curve.

Figure 5-6: The 10 target images used to measure performance with respect to small variations due to noise (SNR = 48.07)

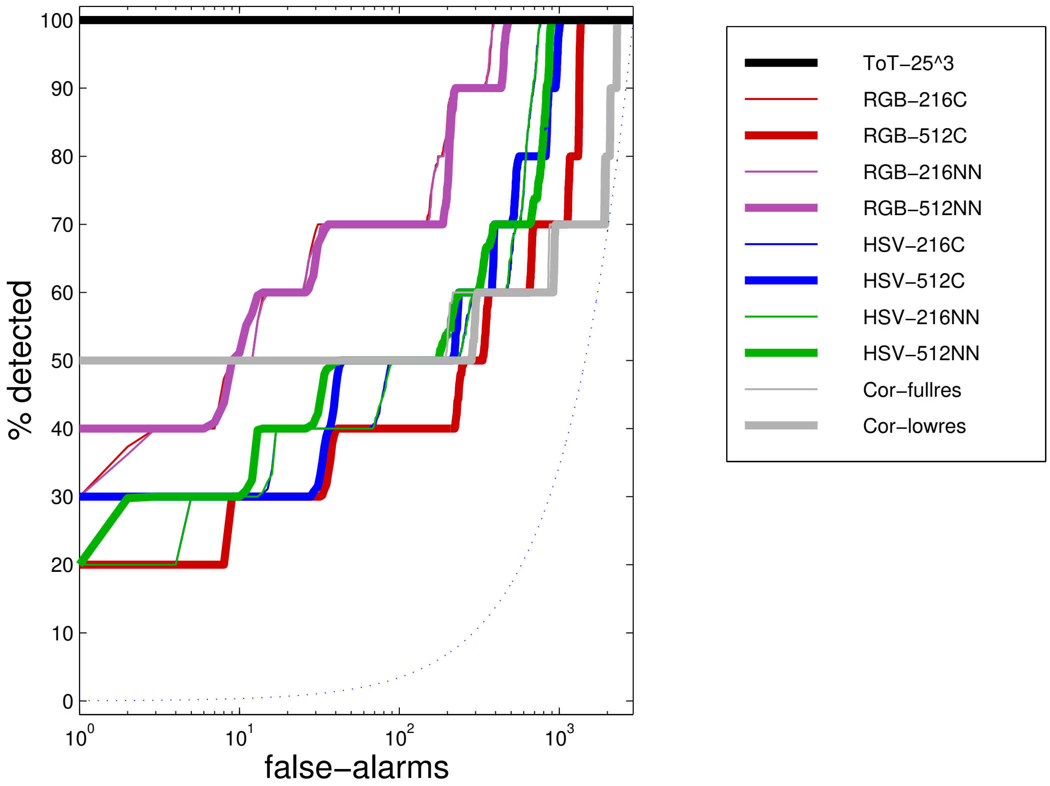


Figure 5-7: ROC curves for retrieval performance of each technique measured with respect to small variations due to noise (SNR = 48.07)

Figure 5-8: The 10 target images used to measure performance with respect to larger variations due to noise (SNR = 3.38)

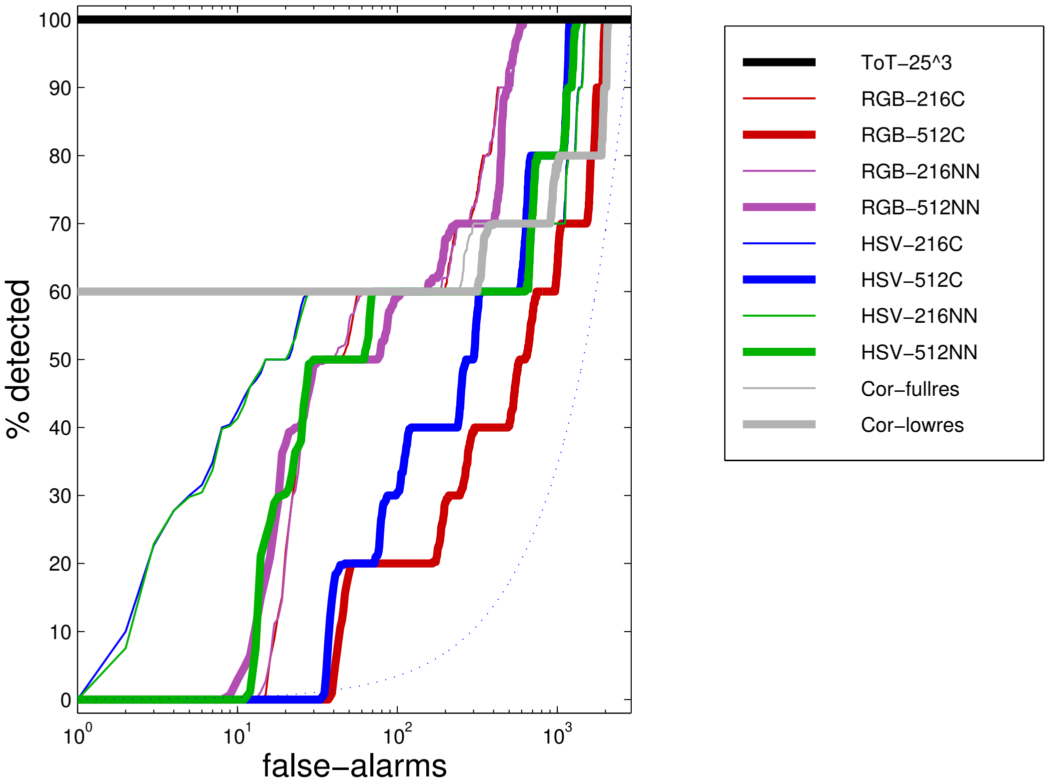


Figure 5-9: ROC curves for retrieval performance of each technique measured with respect to larger variations due to noise (SNR = 3.38)

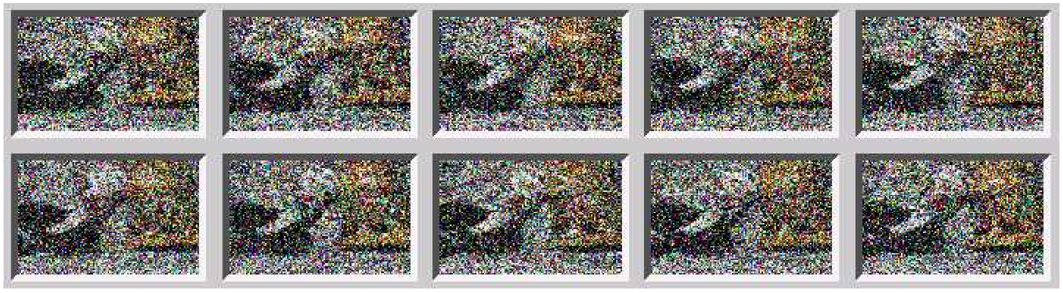


Figure 5-10: The 10 target images used to measure performance with respect to very large variations due to noise (SNR = 0.94)

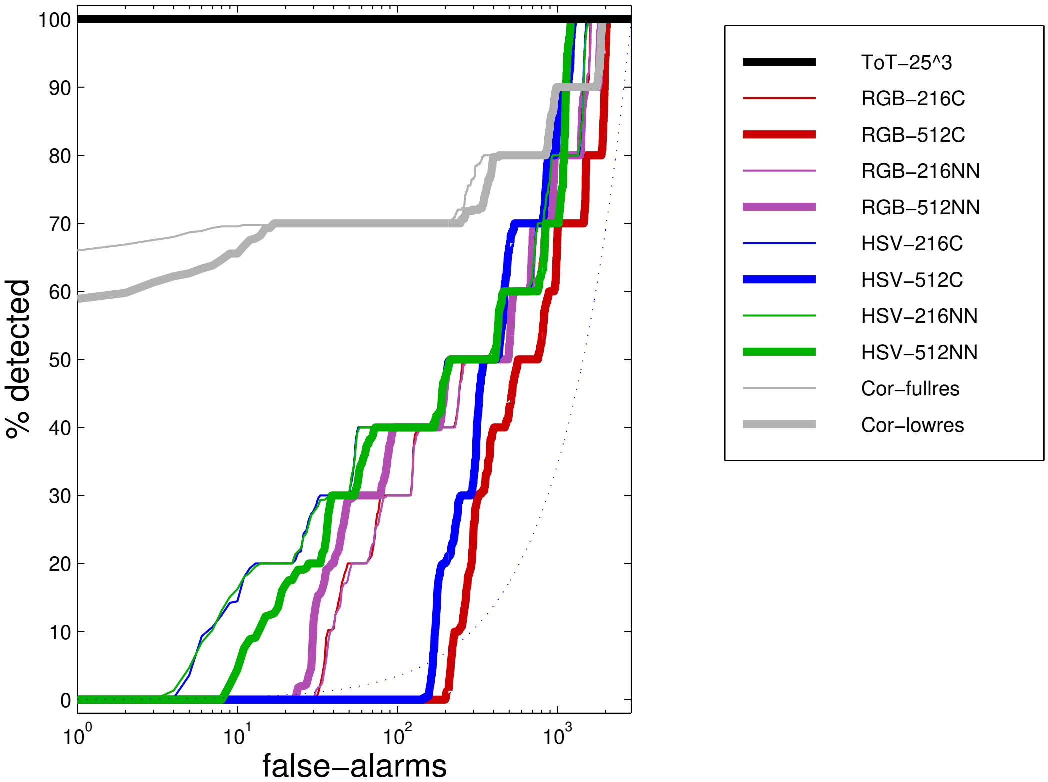


Figure 5-11: ROC curves for retrieval performance of each technique measured with respect to very large variations due to noise (SNR = 0.94)

5.1.4 Performance Experiment: Rotation

In this experiment we consider the effect of image plane rotation on the performance of each retrieval system. Each image in the set was generated by rotating the base image by varying angles about its center:

$$I_\gamma(x,y) = I\left[\begin{array}{c} (x - X_{center}) \times \cos(\gamma) + X_{center} - (y - Y_{center}) \times \sin(\gamma) + Y_{center}, \\ (x - X_{center}) \times \sin(\gamma) + X_{center} + (y - Y_{center}) \times \cos(\gamma) + Y_{center} \end{array}\right] \tag{5.6}$$

We generated two target sets, one with small rotations over $\pi/6$ radians, with steps of $\pi/60$; shown in Figure 5-12. In the second set we rotate by larger steps of $2\pi/10$, and a full rotation is contained within the target set. This set is shown in Figure 5-14.

Receiver operating curves for retrieval of target images in the small rotation set are shown in Figure 5-13. Because the rotational selectivity of the filters used in the generation of the characteristic signature is relatively low, the ToT-25ˆ3 model is able to achieve virtually perfect performance. Each of the oriented half-filters h_1, h_2, v_1 and v_2 have responses which are roughly proportional to $\cos\left(\frac{\pi}{2}\theta\right)$ times its primary orientation of $\theta = 0$. The oriented half-filters h_4, h_5, v_4 and v_5 have responses which are roughly proportional to $\cos\left(\frac{\pi}{6}\theta\right)$ times its primary orientation of $\theta = 0$. The non-oriented filters h_0 and v_0 are roughly invariant to rotations. As a result, combinations of these filters in the filter set F_0 through F_{25} are roughly unaffected for the small rotations of $\pi/60$ between images in the target set. Furthermore, any features which are sensitive to rotation will have a very high variance across any pair of images used as query images, and will therefore, have a lesser effect in the characteristic signature comparison.

Because of these invariances the textures-of-textures model achieves almost perfect performance. With a Neyman-Pearson criterion of at most 50 false-positives, the textures-of-textures technique retrieves 100% of the target image in the $\pi/6$-rotation set, and about 92% over the 2π-rotation set.

Because rotation does not affect the pixel values, the color histogram techniques perform reasonably well. Perfect performance is not achieved because the pixels at the edges of the images move in and out of the frame as the image is rotated, and are replaced with pixels which are only present in the full size image Figure 5-1. With at most 50 false-positives, the best of the color histogram techniques retrieves less 70% of the target image in the $\pi/6$-rotation set, and drops to about 60% over the 2π-rotation set. However, the RGB-216NN and HSV-512C show significantly worse performance than the other techniques by almost a factor of two.

Under small rotations, the correlation based techniques are reasonably stable because of neighboring pixels in natural images tend to vary only slightly, as the frequency spectrum is of natural images is typically $1/f^2$ [15]. Higher power in lower frequencies implies that as image regions move further apart, they are increasing likely to be different. With larger rotations, however, the image regions which are compared when correlation is performed are further apart. As a result, performance of correlation techniques drops substantially, by over 40% in the low false-positives regime, when rotation is over 2π radians.

Figure 5-12: The 10 target images used to measure performance with respect to small variations in rotation over 60 degrees

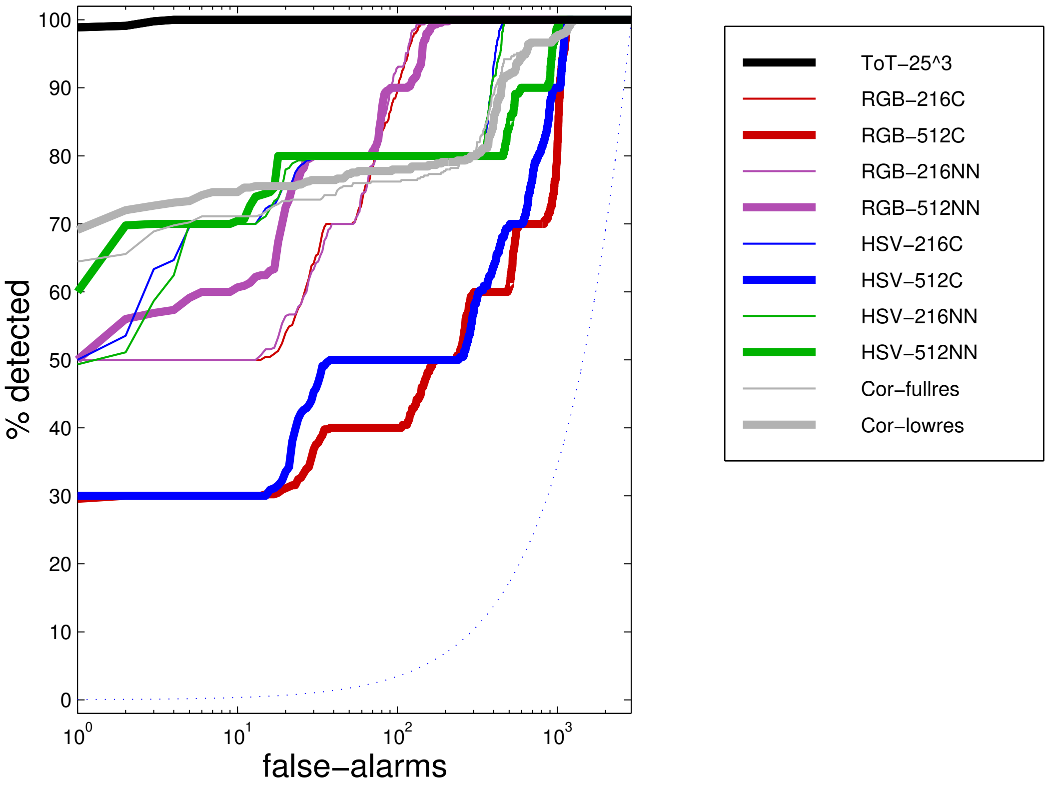


Figure 5-13: ROC curves for retrieval performance of each technique measured with respect to small variations in rotation over 60 degrees

Figure 5-14: The 10 target images used to measure performance with respect to larger variations in rotation over 360 degrees

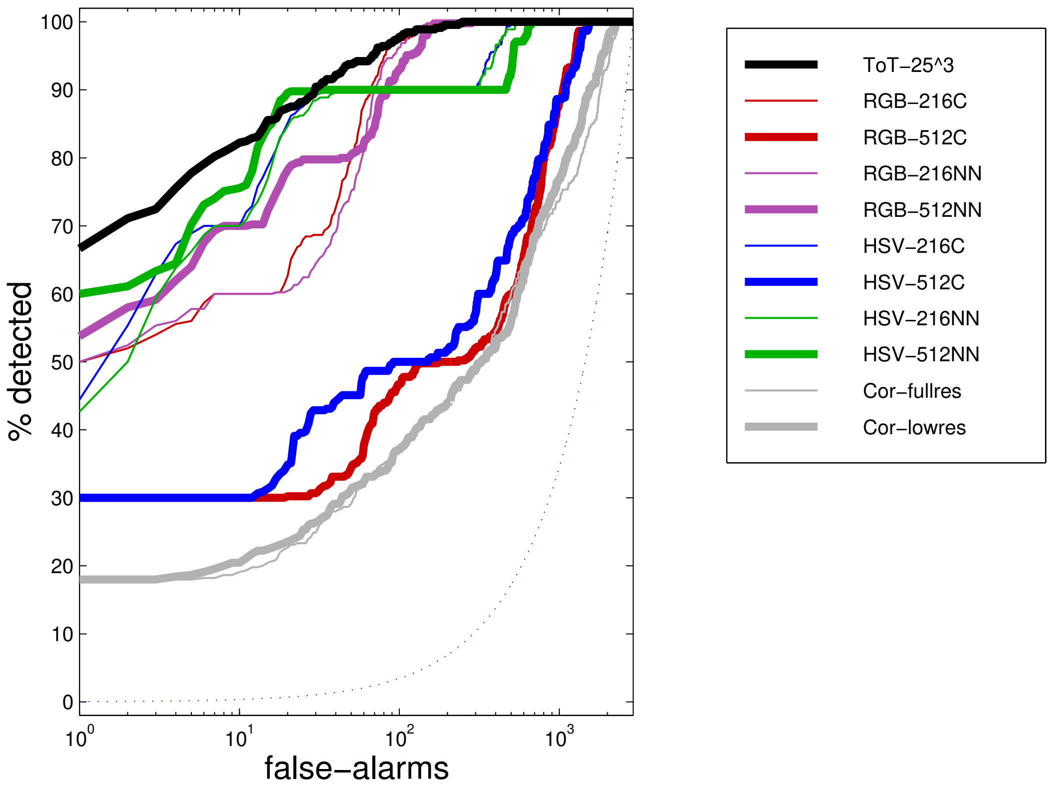


Figure 5-15: ROC curves for retrieval performance of each technique measured with respect to larger variations in rotation over 360 degrees

5.1.5 Performance Experiment: Translation

In this experiment we use the full image Figure 5-1 and extract from it a series of smaller images taken at varying translation across the middle section:

$$I_{\gamma}\left(x, y\right) = I\left(x + X_{center} - W/2 + \gamma, y + Y_{center} - H/2\right) \tag{5.7}$$

where $H = 80$ and $W = 120$ are the width and height of the canonical image.

To generate the target set, γ was varied from -50 to $+50$ pixels by 10. The target image set is shown in Figure 5-16. The width of the canonical image is 120 pixels, therefore there is at least a 20 pixel overlap between each of the images.

Receiver operating curves for each technique are shown in Figure 5-17. In the range over which we are most concerned, i.e. under low Neyman-Pearson criteria, the textures-of-textures technique outperforms the other methods. With at most 50 false-positives, 80% of the target images are retrieved.

With a larger number of false-positives, between 150 and 900, several of the color-based techniques retrieve a slightly larger percentage of the target images. This may be due to the uniformity in color distribution across the middle section of the canonical image (Figure 5-1) and although the image structures which move into the frame are very different in structure, the chromatic distribution is very similar. Again we see that RGB-216NN and HSV-512C performance is significantly below that of the other color histogram techniques (by about 30%).

Even with slight translations, correlation based techniques can be expected to fail. Downsampling increases the robustness to this, but simultaneously decreases the discrimination power of the technique.

5.1.6 Performance Experiment: Zoom

Using the high resolution full size image in Figure 5-1 as a base, from which a set of images at different zooms were generated using bilinear interpolation:

Figure 5-16: The 10 target images used to measure performance with respect to variations in translation

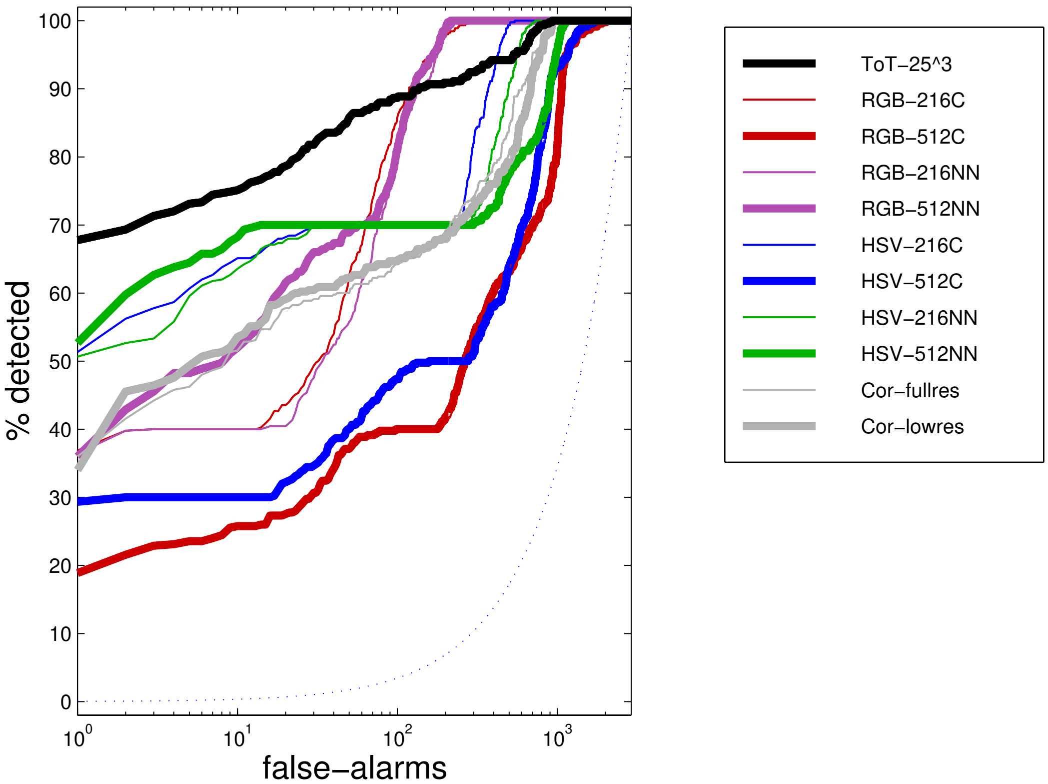


Figure 5-17: ROC curves for retrieval performance of each technique measured with respect to variations in translation

Figure 5-18: The 10 target images used to measure performance with respect to variations in zoom

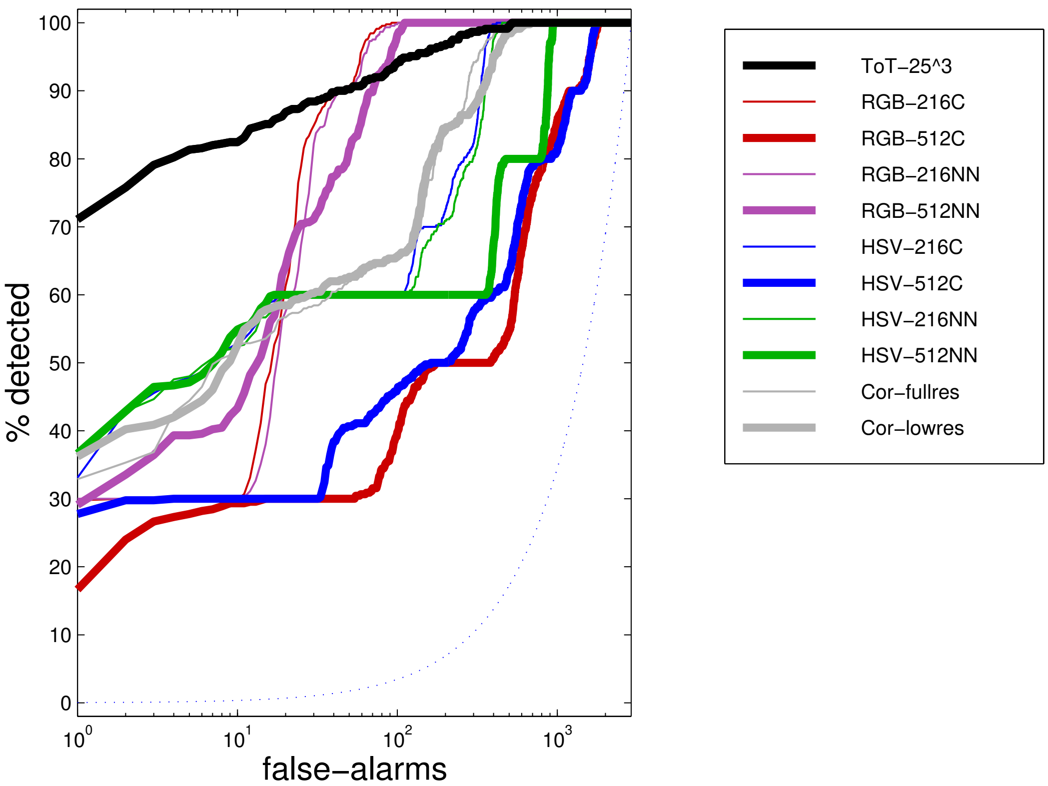


Figure 5-19: ROC curves for retrieval performance of each technique measured with respect to variations in zoom

$$
\begin{aligned}
I_\gamma(x,y) = \frac{1}{Z} \Bigg[& \left(\sqrt{\left[(x-X_{center})\,\gamma - \lfloor x-X_{center}\rfloor\,\gamma\right]^2 + \left[(y-Y_{center})\,\gamma - \lfloor y-Y_{center}\rfloor\,\gamma\right]^2}\right) \\
& \times \; I\left[\lfloor (x-X_{center})\,\gamma + X_{center}\rfloor, \lfloor (y-Y_{center})\,\gamma + Y_{center}\rfloor\right] \\
+ & \left(\sqrt{\left[1-(x-X_{center})\,\gamma + \lfloor x-X_{center}\rfloor\,\gamma\right]^2 + \left[(y-Y_{center})\,\gamma - \lfloor y-Y_{center}\rfloor\,\gamma\right]^2}\right) \\
& \times \; I\left[\lfloor (x+1-X_{center})\,\gamma + X_{center}\rfloor, \lfloor (y-Y_{center})\,\gamma + Y_{center}\rfloor\right] \\
+ & \left(\sqrt{\left[1-(x-X_{center})\,\gamma + \lfloor x-X_{center}\rfloor\,\gamma\right]^2 + \left[1-(y-Y_{center})\,\gamma + \lfloor y-Y_{center}\rfloor\,\gamma\right]^2}\right) \\
& \times \; I\left[\lfloor (x+1-X_{center})\,\gamma + X_{center}\rfloor, \lfloor (y+1-Y_{center})\,\gamma + Y_{center}\rfloor\right] \\
+ & \left(\sqrt{\left[(x-X_{center})\,\gamma - \lfloor x-X_{center}\rfloor\,\gamma\right]^2 + \left[1-(y-Y_{center})\,\gamma + \lfloor y-Y_{center}\rfloor\,\gamma\right]^2}\right) \\
& \times \; I\left[\lfloor (x-X_{center})\,\gamma + X_{center}\rfloor, \lfloor (y+1-Y_{center})\,\gamma + Y_{center}\rfloor\right] \Bigg]
\end{aligned}
\tag{5.8}
$$

where Z is a normalization factor equal to the sum of the weights on each of the four boundary pixels.

In Figure 5-18 the target set is shown. To generate the target set, γ was varied over $\{0.87^0, 0.87^1, \ldots, 0.87^9\}$. A geometric progression is used so that the relative zoom between two successive images in the target image set is constant.

The images in the target set are shown in Figure 5-18. Receiver operating curves for each technique are shown in Figure 5-19.

In the low Neyman-Pearson criterion range (in which we are most concerned) the textures-of-textures method greatly outperforms the other techniques. With at most 50 false-positives, 85% of the target images are retrieved.

With higher acceptable numbers of false-positives, between 130 and 600, several of the color-based techniques retrieve a slightly larger percentage of the target images. This occurs because although the the image structures which move into the frame at different zoom levels are very different, their chromatic distribution is very similar. Again we see the RGB-216NN and HSV-512C performance below that of the other color histogram techniques (by between 20 and 30%).

When a small number of false positives are acceptable, all of the competing techniques fall below the textures-of-textures model; and for very low false-positives, they fall below by more than 30%.

The filters used to generate the characteristic signature are roughly one-octave band pass. With zoom factors of less than $2\times$ the features to which each filter responds will still activate (or "excite") the same paths in the filter-network tree detectors, though to a degree which diminishes with increased zoom factors. As a result of of this stability the performing the textures-of-textures method is able to achieve 90% at a Neyman-Pearson criterion of 50.

Figure 5-20: The 10 target images used to measure performance with respect to variations in occlusion

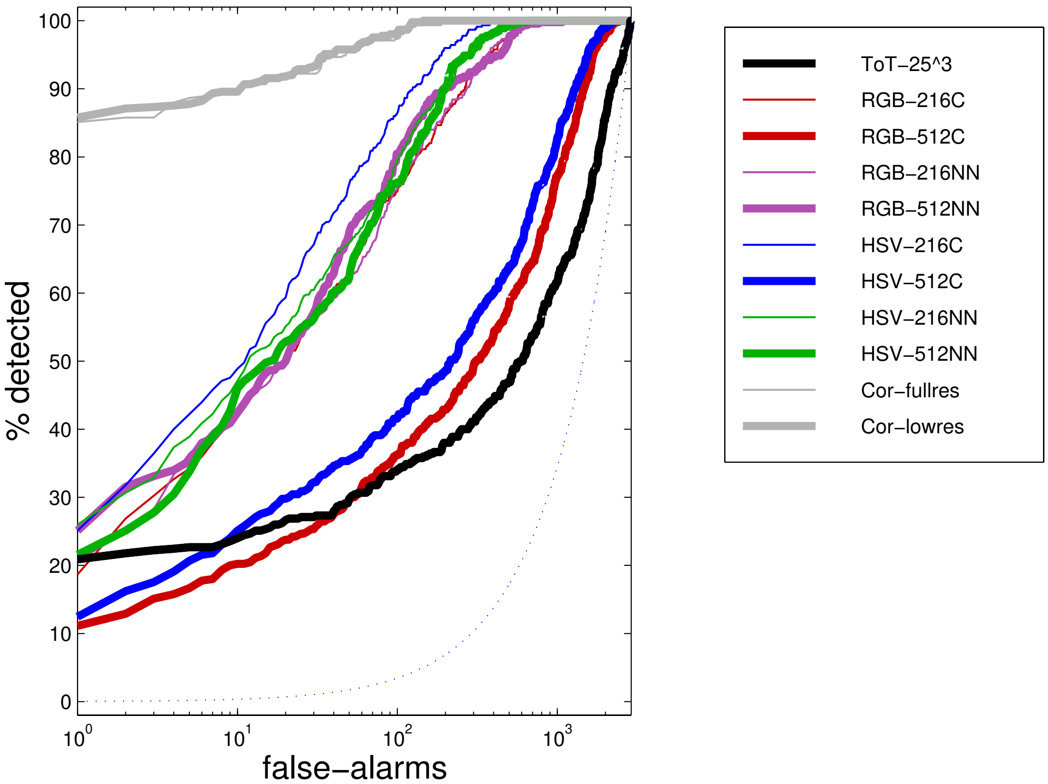


Figure 5-21: ROC curves for retrieval performance of each technique measured with respect to variations in occlusion

5.1.7 Performance Experiment: Occlusion

In this experiment we occlude different regions of the canonical image. To prevent techniques from identifying the target images by recognizing the occluded region, we occluded circularly symmetric regions, centered at random locations, with patches of random texture extracted from images in the MIT AI Learning & Vision Group texture database [31].

$$I_\gamma(x,y) = \begin{array}{ll} & \left\{1 - \exp\left[\frac{(x-X_r)^2+(y-Y_r)^2}{40^2}\right]\right\} I(x,y) \\ + & \left\{\exp\left[\frac{(x-X_r)^2+(y-Y_r)^2}{40^2}\right]\right\} I_{texture}(x,y) \end{array} \tag{5.9}$$

Where (X_r, Y_r) is a random point in the image, and $I_{texture}$ is an image from the texture database.

The target image set is shown in Figure 5-20. From the appearance of these images, it it evident that they all have very different global and local visual appearance. In fact the variation is so large that it is arguable whether or not we consider them visually similar.

Receiver operating curves for each technique are shown in Figure 5-21. Because the visual structure of the target images is completely different, the present system was unable to successfully retrieve a significant portion of the target images, and achieved a performance which far below the other techniques.

However, except for the super-imposition of occluding textures, the images were not perturbed in any way. Those regions which remained unoccluded were in perfect correlation. Thus the correlation based techniques achieved the best performance on this experiment. This indicates that though we may not consider robustness to this type of manipulation critical, it is possible.

The aggregation of feature responses at the last stage of characteristic signature computation makes the present method highly sensitive to the variations caused by occlusion. To overcome this, one could consider directly correlating the energy images from the leaf of each branch in the filter-network tree. However, storing all 46,875 images may be prohibitively expensive. Isolation of a small set of *critical features* for a given query and then recalculation of the associated energy images for direct correlation, is a feasible solution, and will be considered in future research. In the Chapter 6, we examine the performance of a method which isolates such a set critical features and uses their responses to measure similarity.

5.2 Physical manipulation performance experiments

In the next set of experiments, the target images were constructed by varying the physical environment in which the pictures were taken. These experiments may be more indicative of the performance of each technique in practice; as we expect similar images in a natural image database to vary because of such physical variations.

5.2.1 Performance Experiment: Camera position

In this experiment, the camera position has been varied over a series of images in which the objects have been undisturbed. The target images are shown in Figure 5-22. All images

Figure 5-22: The 10 target images used to measure performance with respect to variations in camera position

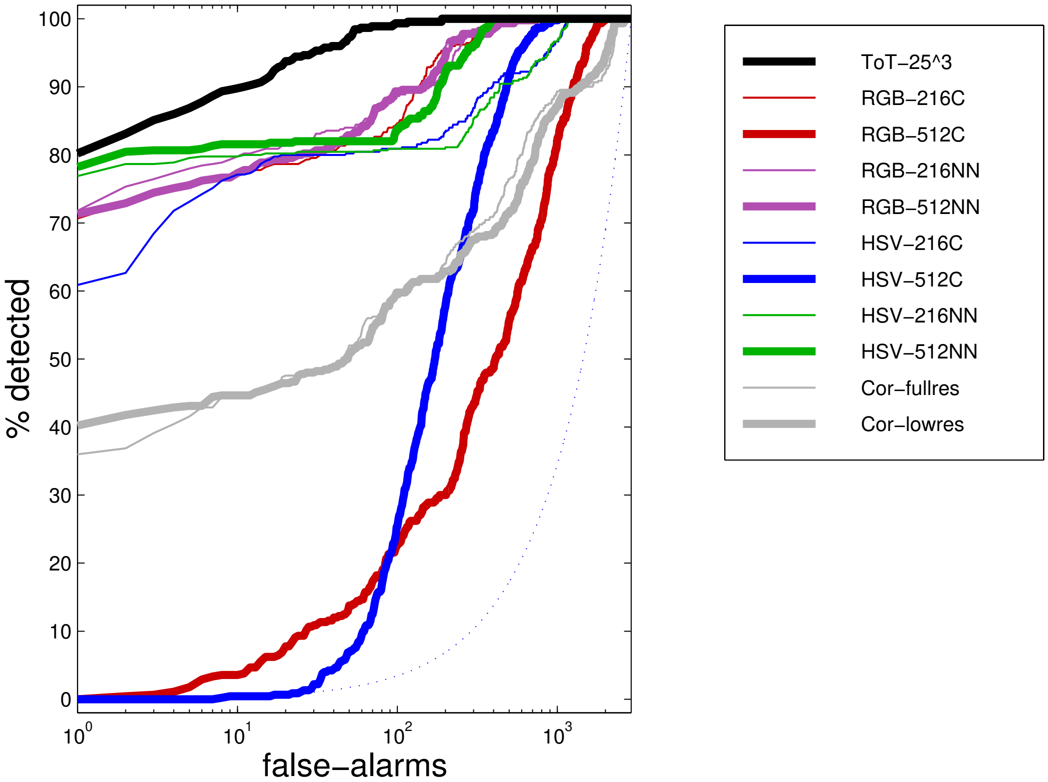


Figure 5-23: ROC curves for retrieval performance of each technique measured with respect to variations in camera position

were taken from roughly 4 feet from the Buddha figurine which was placed at the center of each image.

Pictures were taken at approximately $(\pm\pi/10, 0)$, $(\pm 2\pi/10, 0)$, $(\pm 3\pi/10, 0)$, $(\pm 4\pi/10, 0)$ and $(\pm\pi/4, \pi/6)$ measured in azimuth and elevation from the canonical full frontal view in Figure 5-1.

Receiver operating curves for each technique are shown in Figure 5-23. The best performance was achieved by the textures-of-textures model, which generated the top curve. For any number of retrieved images, the current model returned the highest percentage of target images. At a Neyman-Pearson criterion of 50, over 95% of the images are retrieved, while the best of the other techniques retrieve less than 85%.

Correlation based techniques did not work well, as changes in camera position cause large changes in every pixel value (in this case, especially pixels near the right and left edges.)

Two color techniques, RGB-216NN and HSV-512C, perform surprisingly poorly relative to similar methods. This however, is indicative of the instability of histogram techniques; in many cases variations in target images will cause only small shifts in generated histograms, but periodically similar variations will cause catastrophic performance degradation — due to *chromatic aliasing* with respect to the histogram bins. The behavior of a small change in input yielding a drastic change in output, known formally as *instability*, decreases the robustness and viability of any system which relies on histogram operations.

Techniques for combining multiple histograms with different bin granularities and for histograms with overlapping bins, do exist and have been shown to improve robustness in some cases [48, 16]

5.2.2 Performance Experiment: Light position

In this experiment, the position of the light source has been varied over a series of images in which the objects have been undisturbed. We generated two light position variation series.

In the first series, "soft shadows" were generated. The shadows were made soft by to factors: constant ambient light was present in addition to the moving spotlight; and the spotlight was varied over a relatively narrow cone, with a radius of $\pi/6$ in both azimuth and elevation centered around the focal axis along which the images were taken. The target images are shown in Figure 5-24.

Receiver operating curves for each technique are shown in Figure 5-25. The best performance was achieved by the textures-of-textures model, which generated the top curve. For any number of retrieved images, the current model returned the highest percentage of target images. At a Neyman-Pearson criterion of 50, over 98% of the images are retrieved, while the best of the other techniques retrieve less than 75%. Change in lighting position systematically changes almost all of the pixel values, and as a result correlation based techniques perform little better than chance.

The second series, consisted of a set of images containing "hard shadows." The shadows were made hard by reversing the two factors above: no ambient light was present; and the spotlight varied almost π radians in azimuth. (Only variation in elevation was used because the physical arrangement of the objects prevented either the placement of the spotlight, or the casting of shadows with large variations in the angle of elevation.) The target

Figure 5-24: The 10 target images used to measure performance with respect to small variations in light position, which cause soft shadows

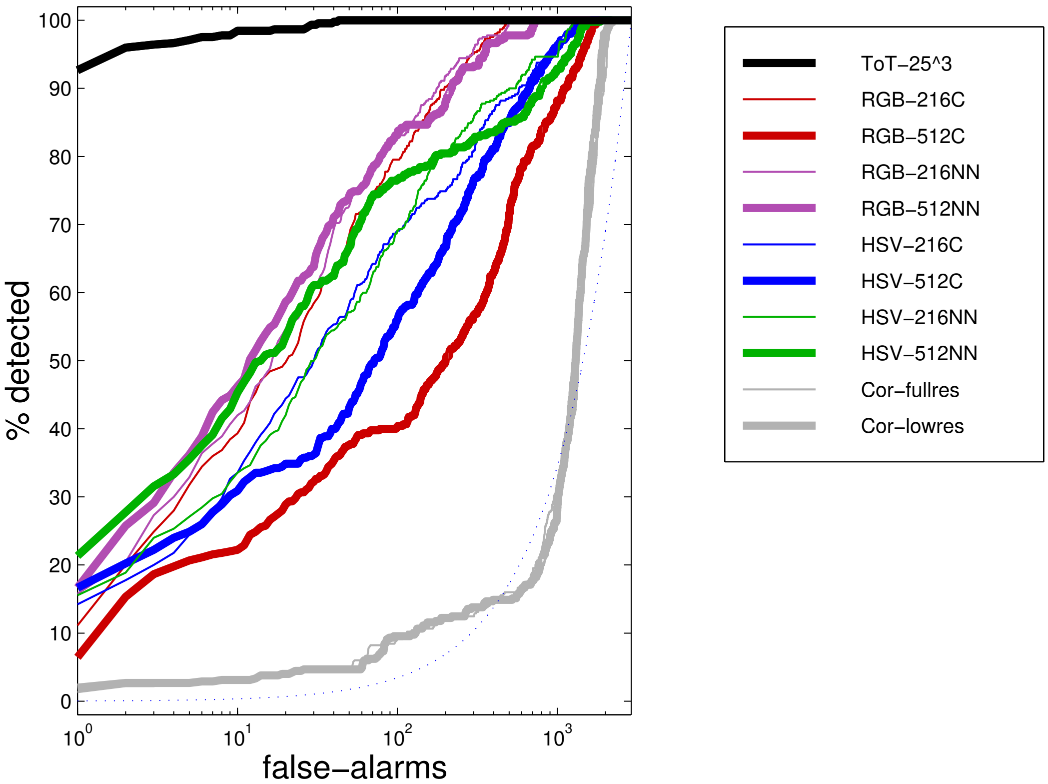


Figure 5-25: ROC curves for retrieval performance of each technique measured with respect to small variations in light position, which cause soft shadows

Figure 5-26: The 10 target images used to measure performance with respect to larger variations in light position, which cause hard shadows

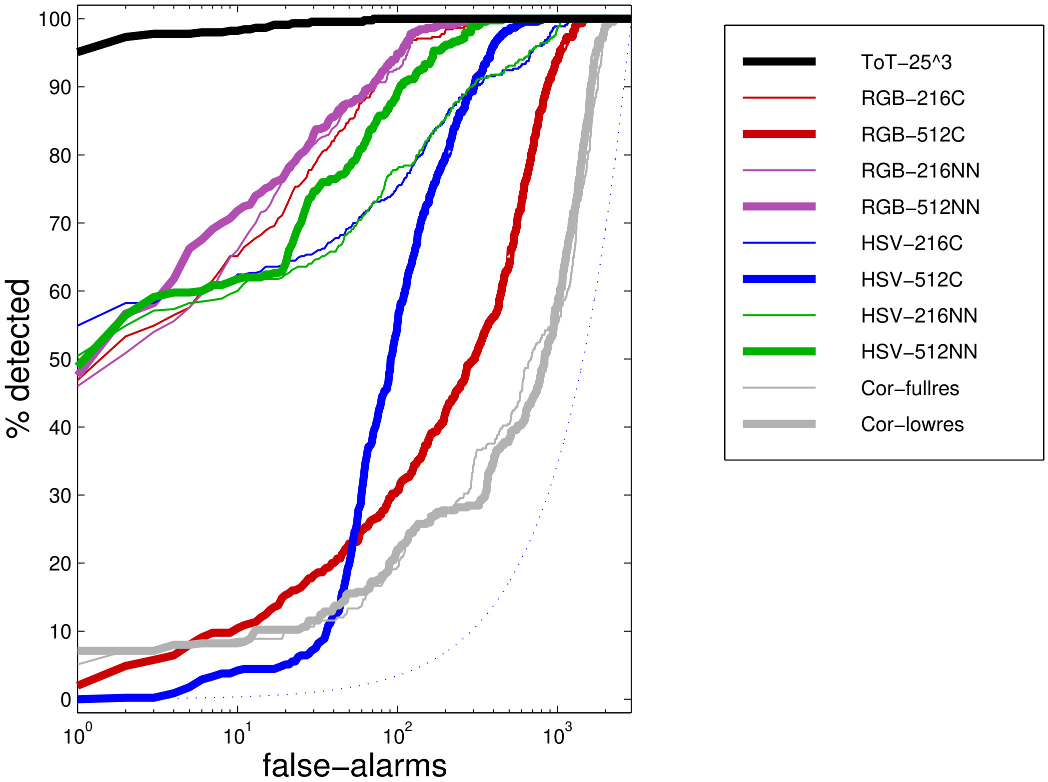


Figure 5-27: ROC curves for retrieval performance of each technique measured with respect to larger variations in light position, which cause hard shadows

images for this series are shown in Figure 5-26.

Receiver operating curves for each technique are shown in Figure 5-27.

Despite the fact that the images appear to be more different from one another in the hard shadow data set than in the soft shadow set, the performance of every technique improves.

The best performance was still achieved by the textures-of-textures model, which at a Neyman-Pearson criterion of 50, retrieves over 99% of the target images. The performances of the all the other techniques improve by more than 10%. Notably in the region below a Neyman-Pearson criterion of 30, the current model improves by 5% to 97%, and five of the color techniques (all, except for the RGB-216NN and HSV-512C) improve by roughly 30%.

At first it appears somewhat strange that performance should *improve* while the differences among the target images *increases*. However, this effect can be explained by examination of the clutter set of Corel images. Though these images are a sampling of photographs of natural scenes, they are constrained sampling in that each picture was perceived by a photographer to be esthetic in some way. As a result, a vast majority of the images contain well balanced lighting, which causes smooth lighting variations across the image, and tends to cause soft shadows. As a result, even though the target images are closer to one another in the soft shadow case, they are also closer to more of the clutter images, causing more clutter images to be confused with target image in the soft shadow case than in the hard shadow case.

In both the hard and soft shadow cases, however, the textures-of-textures technique performs markedly better than the competing techniques.

5.2.3 Performance Experiment: Object pose

In this experiment, both lighting and camera position were held constant in the arrangement used in the canonical image. In each image in the target set, the objects were physically rearranged. Both the poses and relative positions of the objects were varied. Variation included rotation of the objects in the plane orthogonal to the camera (which exposes formerly unseen views of the objects to the camera.)

In some real sense performance with respect to this sort of variation is a very important measure of a retrieval system's robustness. Object pose variations such as these are what would be anticipated when comparing pictures taken from a particular location as objects within the scene move over time. Many real-world applications of image retrieval consist solely, or in a great part, of databases of images which possess these sorts of variation. A notable class of such examples are retrieval systems designed to deal with images from video sequences: such as human and vehicle surveillance, scene-change detection, and (television) program identification, etc.

If we consider the objects in the target images to be roughly equivalent to objects which might move (or be moved) in a scene this experiment can be loosely considered a measure of the ability of each system to recognize the scene portrayed in a picture.

Receiver operating curves for each technique are shown in Figure 5-29. By far the best performance was achieved by the textures-of-textures model, which generated the top curve; for any number of retrieved images, it returned the highest percentage of target images. The other techniques roughly cluster in performance far below that of the current

Figure 5-28: The 10 target images used to measure performance with respect to variations in object pose

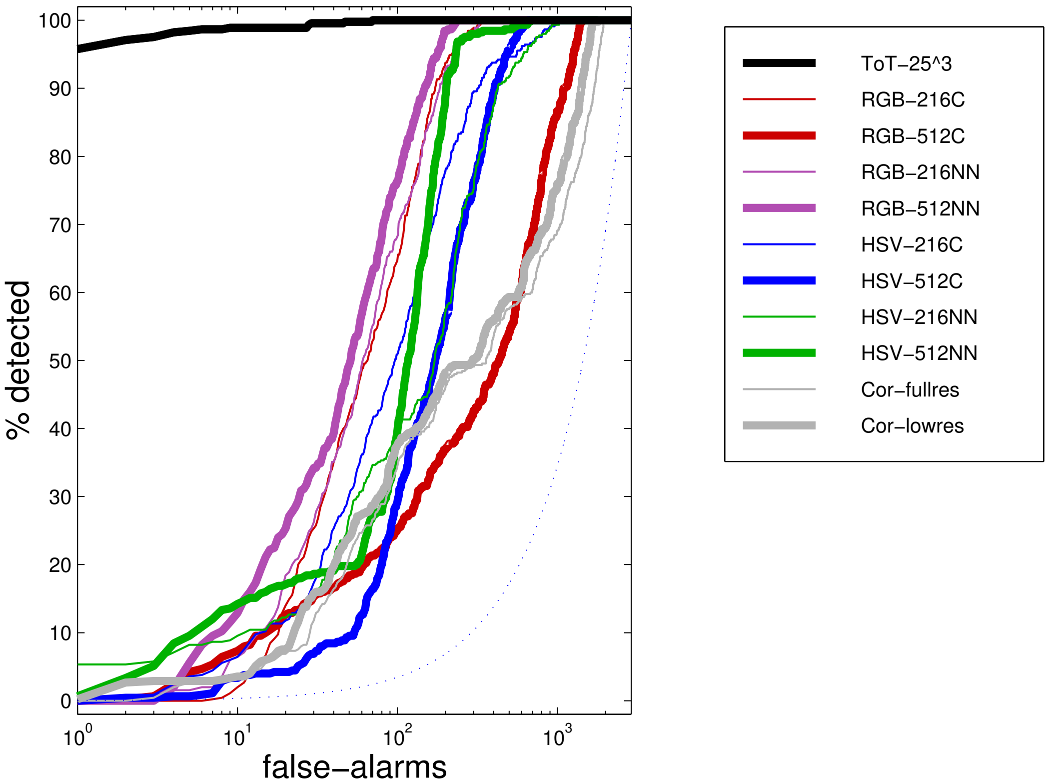


Figure 5-29: ROC curves for retrieval performance of each technique measured with respect to variations in object pose

system. At a Neyman-Pearson criterion of 50, over 98% of the images are retrieved, compared to the best of the other techniques which retrieve less than 50%.

This is strong evidence that the invariances which are incorporated into the system by the structure of the characteristic signature computation, and which have been illustrated in the proceeding experiments, tend to combine and enhance one another when considering complex real-world types of variation.

5.2.4 Performance Experiment: Object location

In the final performance experiment, all the physical parameters were simultaneously varied. The objects were photographed with a variety of lighting conditions, camera positions, relative positions and orientations, and most significantly, *physical surroundings*.

This experiment, complements the previous experiment section 5.2.3, in being a highly important measure of a retrieval system's robustness. By varying the location in which each picture is taken, only the objects in the image are consistent from image to image. Thus, this experiment is a measure of the general object recognition performance of each technique. Real-world applications which focus on object recognition, achieve their high levels of performance by specializing in a particular domain. For example face recognition [51, 64], or character recognition [14] typically require hundreds of thousands of training examples to build a representation which is specifically designed to recognize certain types of targets. Clearly a system developed and trained specifically to recognize written characters would not be expected to be able to do well recognizing faces.

In this situation however, we consider the ability of each technique to recognize the objects in an image without a large quantity of training data. In theory, if there were sufficient time and data, a method could be developed which which built separate models of each of the (potentially huge number of) different objects which it might encounter. Such a system is reminiscent of the notion of a "grandmother cell" which fires only when a (the person's, presumably) grandmother is in view [1]. However, to achieve the robustness and flexibility of recognition at the level of performance found in human observers, it would require an astronomical number of such specialized models, making the prospect of such a technique infeasible.

Essentially the only characteristic which is constant across all the images is the presence of the target objects. Thus, we are in this experiment measuring the abilities of each of these models to perform robust object recognition. To perform this task successfully, a method must generalize over the measure of visual similarity, with very few examples (in this case just two), *in the right way.*

It is possible to perform this task, humans can do it. Though we have not tested this formally, it is clear from our familiarity with the types of images in the database, that human observers can easily find all the target images. For example, the green Gatorade bottle alone is a sufficient cue to the identity of the targets, as it can be clearly recognized (by a human observer) in each of the images and is not present in any of the non-target distractor images.

Receiver operating curves for each technique are shown in Figure 5-31. The best performance was achieved by the present textures-of-textures model, which generated the top

Figure 5-30: The 10 target images used to measure performance with respect to variations in object location

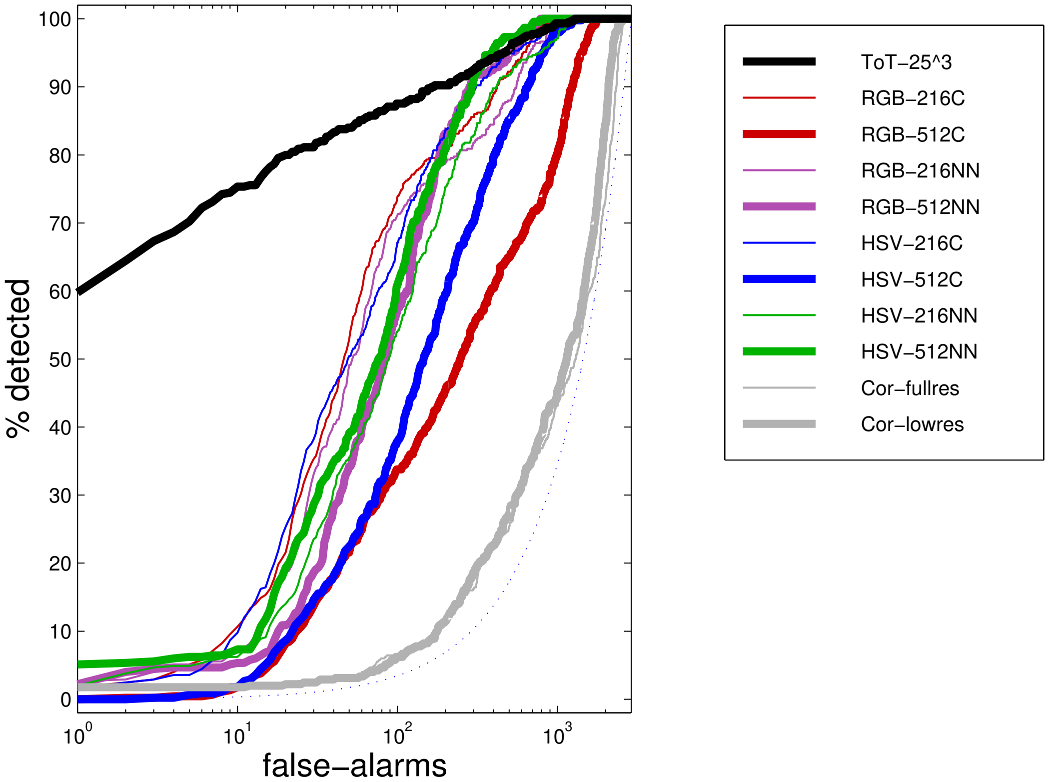


Figure 5-31: ROC curves for retrieval performance of each technique measured with respect to variations in object location

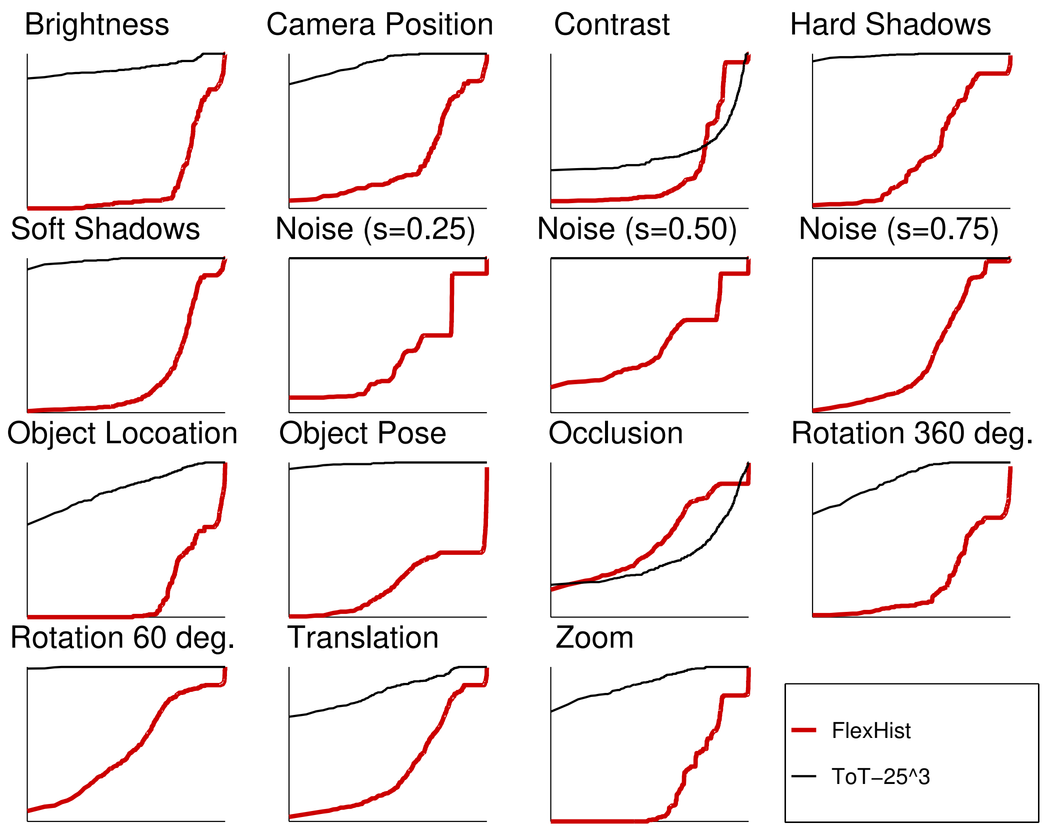


Figure 5-32: Receiver operating curves for the flexible histogram technique are compared to those for ToT-25ˆ3 model across all the experiments.

curve. The other techniques roughly cluster in performance far below that of the current system.

At a Neyman-Pearson criterion of 50, over 87% of the images are retrieved, compared to the best of the other techniques which retrieve less than 60%. Though the ToT-25ˆ3 curve for this experiment is somewhat lower than it achieved in the experiment in section 5.2.3, given the increased difficulty of the variations — as both pose and location are varied in this target set — its performance give further indication of the ability of this system to capture the visual characteristics in the images which are visually salient.

5.3 Comparison of the Texture-Of-Textures model to the Flexible Histogram model

In section 4.2 we conjectured that the flexible histogram model would do a poor job measuring the difference between natural images. Using the experiments described in this chapter, we can now validate that hypothesis.

We repeated each of the 15 experiments using the flexible histogram technique to measure image similarity. Because the flexible histogram computation is $O(N^2)$ (in the number of pixels), it took over two weeks to perform all 675 queries.[2] In Figure 5-32 receiver operating curves for the flexible histogram technique are compared to those for ToT-25ˆ3 model across all the experiments. Axis values for all graphs have been omitted, but are the same as those of the other ROC curves in this chapter. As we conjectured, in every experiment the flexible histogram technique performs extremely poorly, especially in the low Neyman-Pearson range.

This is due to the inherent assumption made by the flexible histogram technique, namely that images in the same class should be able to predict — synthesize — one another. For natural images classes, this is not true. The textures-of-textures model does not make this assumption, and as a result it cannot be used to synthesize images, but it does provide better generalization of an image class given only a few examples. For retrieval from image databases, it is this ability which is critical.

5.4 Discussion

In the preceding 15 experiments, the textures-of-textures model substantially outperforms all of the other techniques in 13 experiments. The difference in performance is especially large in the critical region of low Neyman-Pearson criterion.

To get a comparative sense of the overall performance of each technique, cross sections of the 15 experiments for 30,50 and 100 maximum acceptable false positives are shown in the bar graphs in Figures 5-33, 5-34, and 5-35, respectively.

The two experiments where the current system did not achieve the best performance were the contrast variation series, and the occlusion series.

Because of the sensitivity of the filtering operations, which compose the bulk of the characteristic signature computation, the vulnerability of the technique to variation of global contrast level is not surprising. Possible extensions to this model, which could help make it robust to contrast variations include the addition of an initial stage which equalizes the contrast before computation of the characteristic signature could decrease this susceptibility. It is not clear however, how such equalization would negatively affect the discriminative power of the model.

An alternative extension suggested by the work of LeCun, *et al.* [14], is the insertion of a sigmoidal non-linearity after each filtering operation. The thresholding effect of the added sigmoid would be to roughly quantize the response of each filter. Such an extension could potentially improve the performance of system in lower Neyman-Pearson criterion measurements, with which we are most concerned in typical applications. This improvement however, would not be without cost: when the energy at the end of each filter network falls below the sigmoidal threshold, a catastrophic degradation of the characteristic signature representation will occur, and retrieval rates will fall off accordingly. In Chapter 7 we consider a third option, which is to normalize the characteristic signature *after* it has been computed.

[2] 15 experiments, each with 45 queries.

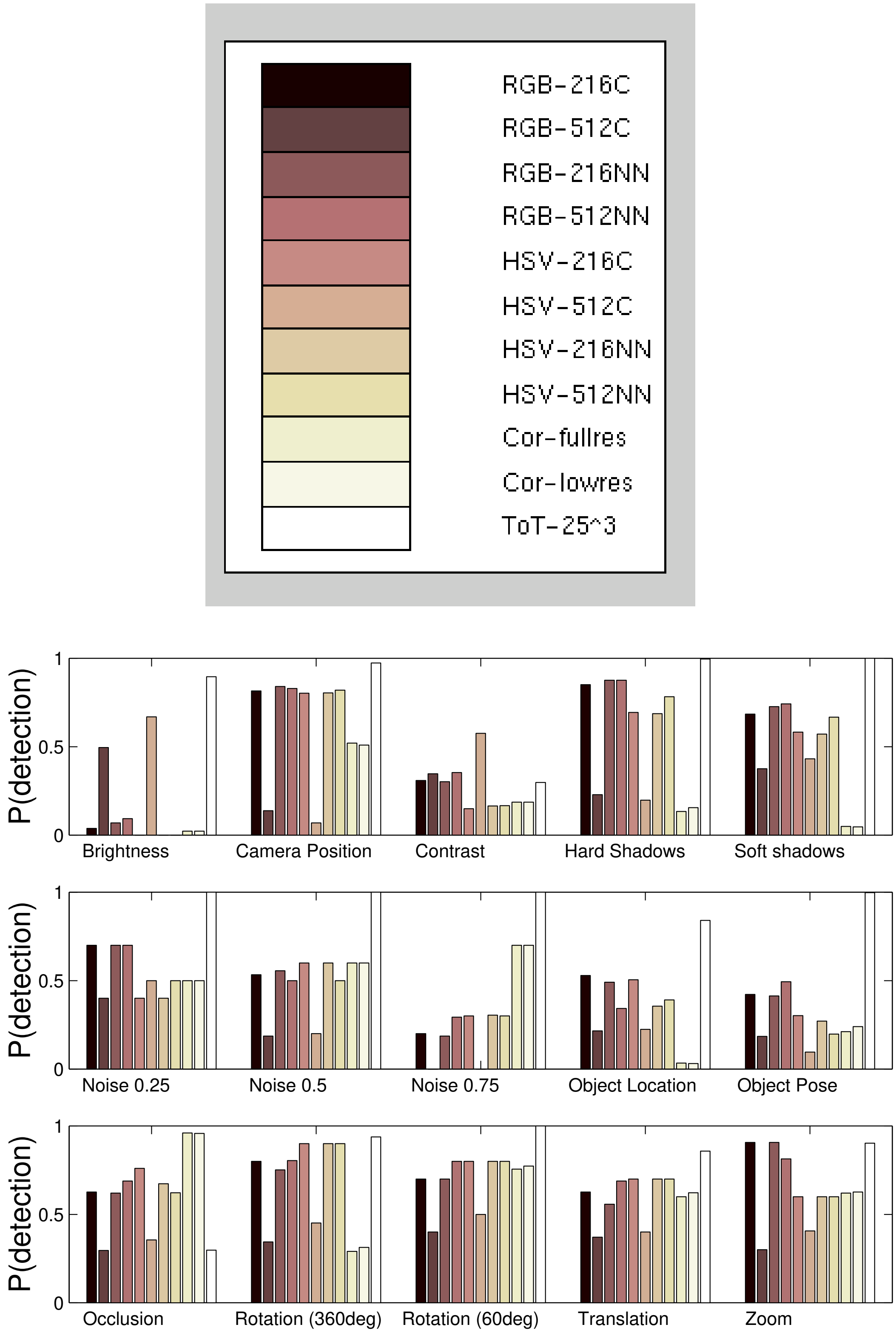


Figure 5-33: Cross sections of the 15 experiments under a Neyman-Pearson criterion of at most 30 false-positives.

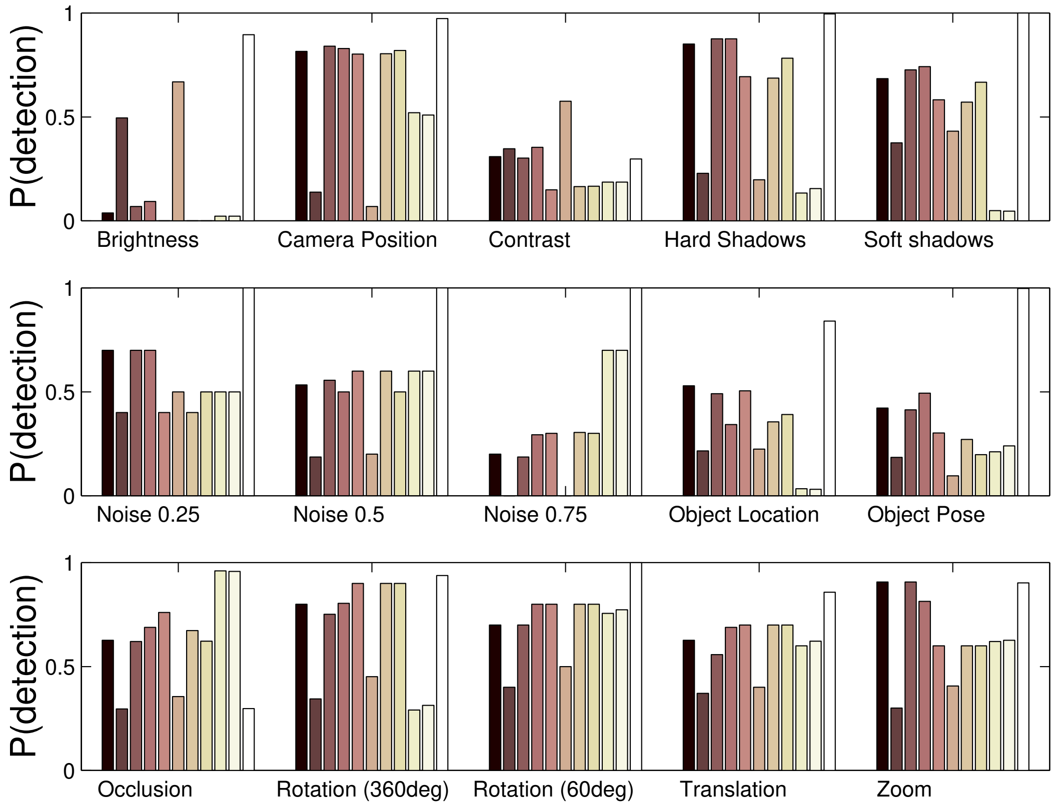


Figure 5-34: Cross sections of the 15 experiments under a Neyman-Pearson criterion of at most 50 false-positives.

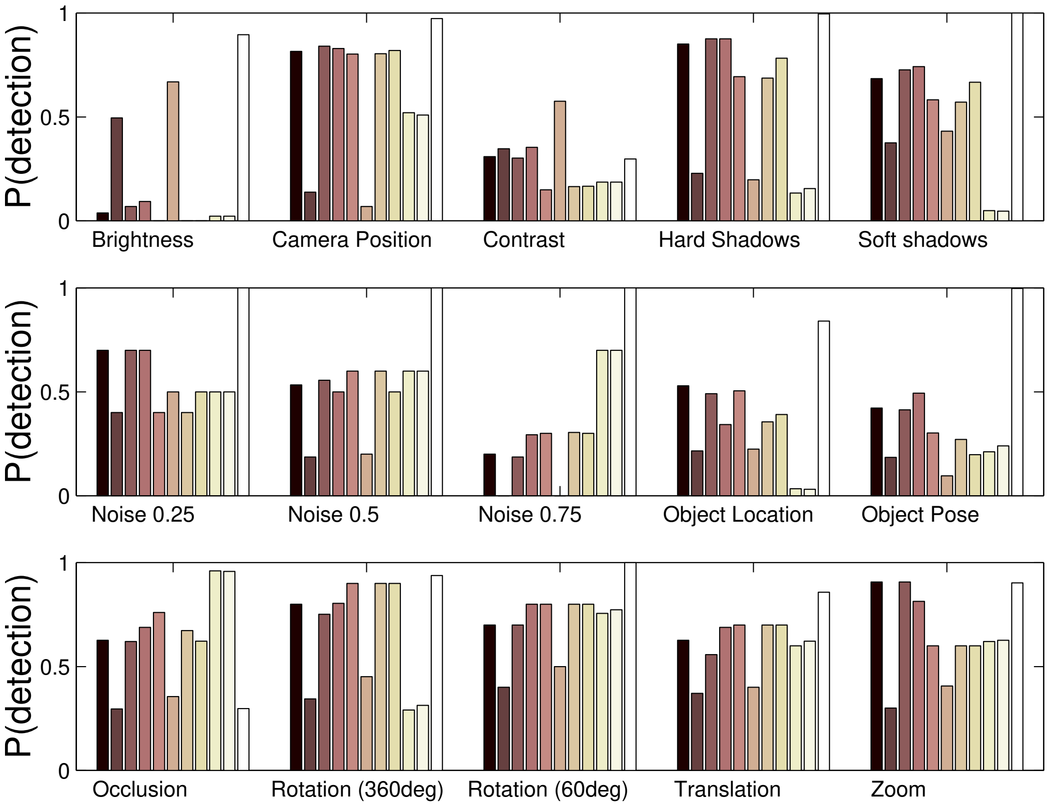


Figure 5-35: Cross sections of the 15 experiments under a Neyman-Pearson criterion of at most 100 false-positives.

A third solution is to normalize the total response of the characteristic signature . When comparing normalized characteristic signatures , only their orientation is significant. Because the characteristic signature generated for an image with reduced contrast is shorter than the characteristic signature for the original, but has the same orientation, normalizing their lengths will make them identical.[3] We consider this extension in Chapter 7.

The performance under low Neyman-Pearson criteria is in many applications, the most critical; therefore, investigations into contrast-compensation techniques will be the subject of future research efforts.

[3]Except for roundoff error which introduces non-invertible effects.

Chapter 6

Analysis of the Textures-of-Textures retrieval technique

To attain an understanding of which components of the textures-of-textures retrieval method we compare retrieval performance for filter network trees with different configurations. By restricting the tree in different ways we essentially "dissect" the model, and determine which elements of the characteristic signature are critical for its high performance level.

6.1 Different configurations of the filter-network tree

To dissect the textures-of-textures method, as described in Chapter 4, and determine which of its components are responsible for its successes, we repeat the experiments performed in Chapter 5, for different configurations of the network tree.

Performance successes with smaller configurations indicate which components of the characteristic signature are critical for attaining invariance to each of the types of visual variation in the target sets.

In each query we compare retrieval rates for the following configurations:

1. **ToT-25ˆ3** The textures-of-textures system, as described in Chapter 4; using 25 filters at each of 3 levels in each filter network.

2. **ToT-25ˆ2** A reduced filter network tree with two levels.

3. **ToT-25ˆ1** A reduced filter network tree with only a single level.

4. **ToT-9ˆ3** a filter network tree. with only the first three horizontal and vertical half-filters, resulting in a branching factor of 9, applied over three levels.

5. **ToT-9ˆ2** a filter network tree with 9 filters applied at two levels.

6. **ToT-9ˆ1** a filter network tree with 9 filters applied at only a single level.

7. **ToT-25ˆ3-Top5000** use of the full 25 filters at each of 3 levels, but comparison of only the 5,000 characteristic signature elements with the lowest variance across the set of query images.

8. **ToT-25^3-Top1975** comparison of only the 1,975 characteristic signature elements with the lowest variance from the the full 25 filters at each of 3 levels. (1,975 elements are used because this is the size of the complete ToT-25^2 characteristic signature.)

9. **ToT-25^3-Top500** comparison of only the 500 characteristic signature elements with the lowest variance from the the full 25 filters at each of 3 levels.

10. **ToT-25^3-Top75** comparison of only the 75 characteristic signature elements with the lowest variance from the the full 25 filters at each of 3 levels. (75 elements are used because this is the size of the complete ToT-25^1 characteristic signature.)

11. **ToT-25^3-Random5000** comparison using 5,000 randomly chosen characteristic signature elements with the lowest variance from the the full 25 filters at each of 3 levels.

The receiver operating characteristics for each configuration of the filter network tree, for retrieval on each set of target images, described in Chapter 5, are shown in Figures 6-1 through 6-5. In each set of plots, the axes on each graph are the same as those in Chapter 5 and are not printed to allow all plots to fit on a page.

6.1.1 Number of levels

By comparing curves for configurations ToT-25^1, ToT-25^2, and ToT-25^3, which is done in Figure 6-1, we see performance improves in almost every retrieval task — except for the contrast and occlusion series where the performance of all configurations was poor — with each additional level of of the network tree. Furthermore, the performance gains are quite significant, in the range of 25% to 30% over the low Neyman-Pearson criterion range. In all of the configurations tested, the number of levels used was the factor which had the greatest influence on performance. Adding additional levels, however, causes an exponential increase in the number of elements in the characteristic signature . With a branching factor of 25, adding an additional level, i.e. generating a *ToT-25^4* configuration, would result in about 1.2 million [1] elements in the characteristic signature . With the computing power which is currently readily accessible such a large signature is prohibitive; a 2,900 image database, such as the one used here, would require 6 gigabytes [2] of storage.

6.1.2 Dominating characteristic signature elements

Are the top 5,000 sufficient?

Because the contribution of each element in the characteristic signature is normalized by the its variance across the query set, the elements whose variance is smallest will tend to dominate the others. By examining the curves which just compare the elements with the lowest variances, we can determine if a comparing some number of elements fewer than

[1] Specifically, $25^4 \times 3 = 1,171,875$ elements.

[2] assuming 2-byte precision per characteristic signature element.

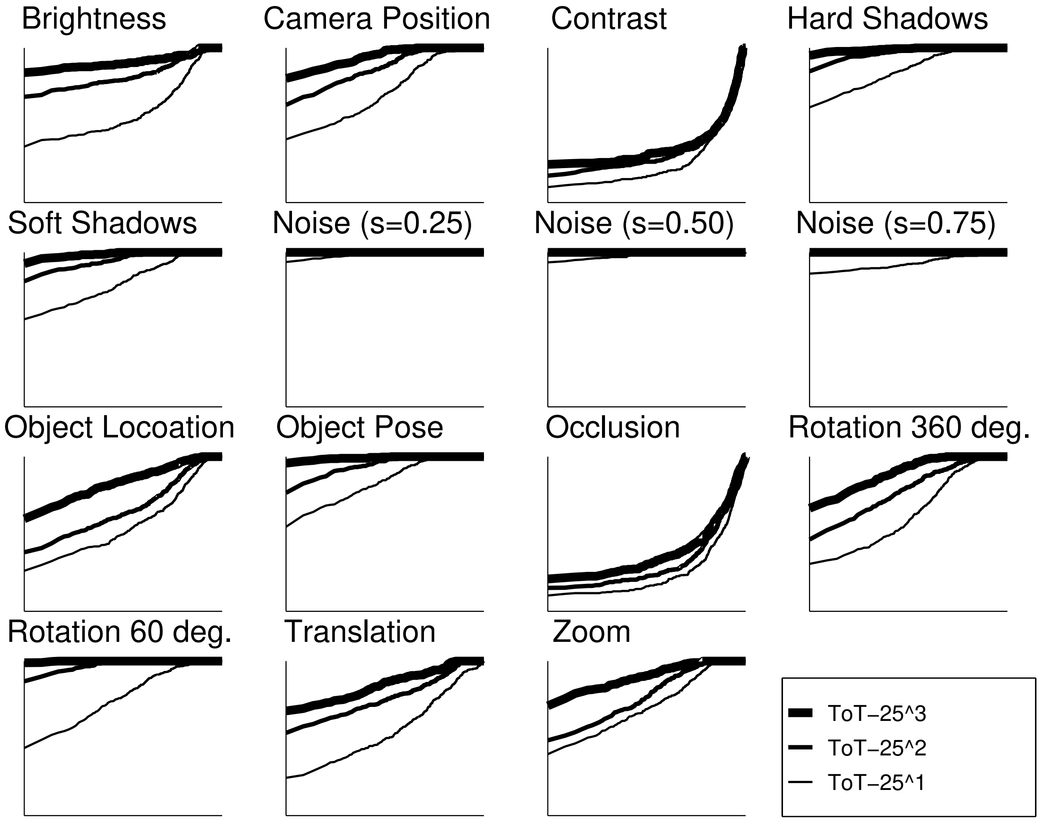


Figure 6-1: ROC curves for 1, 2, or 3 levels of the filter network tree.

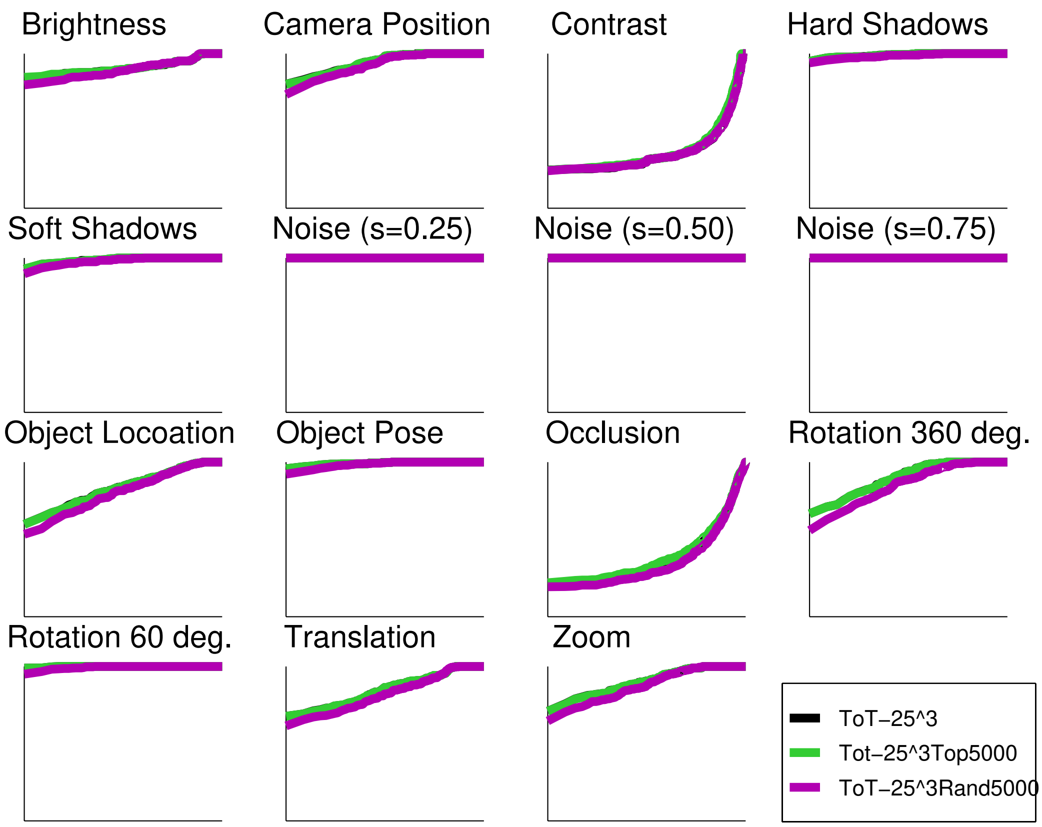


Figure 6-2: The retrieval performance of the full model compared to one in which only the 5,000 elements with the smallest variances are used. As a control we also include a model in which 5,000 randomly selected elements from the full characteristic signature are used.

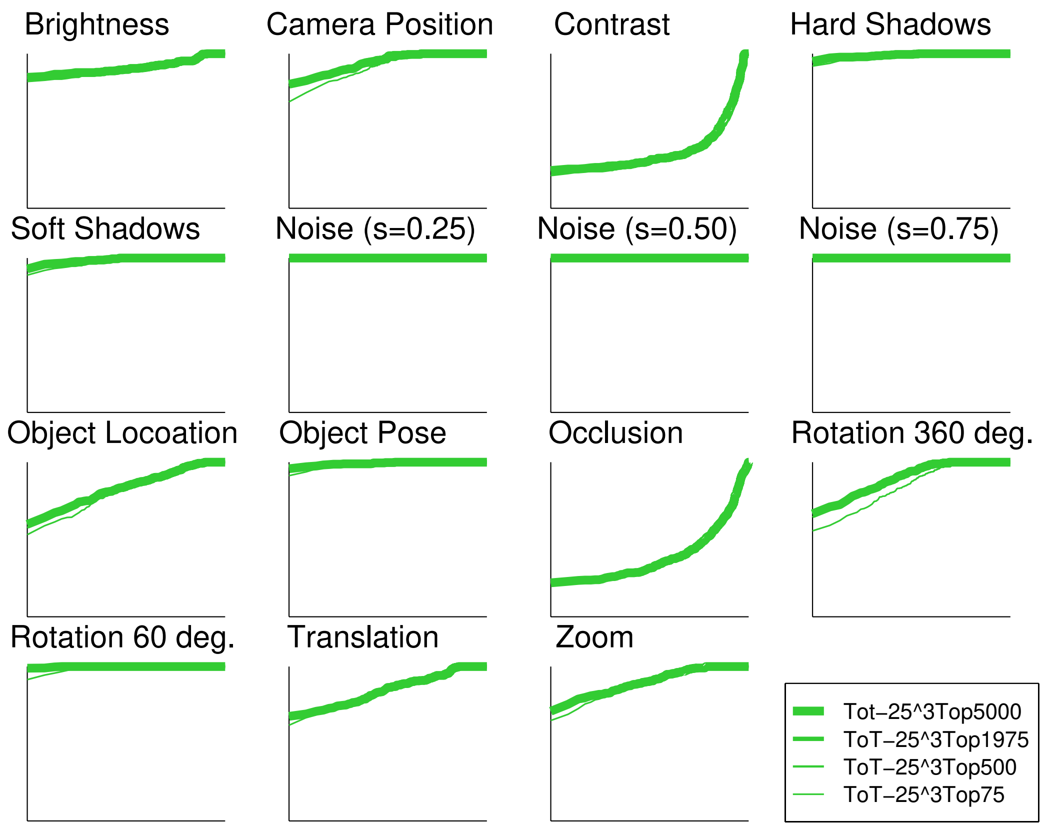


Figure 6-3: ROC curves for configurations which use successively smaller subsets of the top (lowest variance) elements of the characteristic signature .

the complete set can achieve performance comparable to that of the full set. In Figure 6-2 we compare the retrieval performance of the full ToT-25ˆ3 model to the ToT-25ˆ3-Top5000 configuration, in which only the 5,000 elements with the smallest variances are used. As a control we also include ToT-25ˆ3-Rand5000, in which 5,000 randomly selected elements from the full characteristic signature are used. (The same 5,000 elements are used for all queries.) Using 5,000 random elements reduces the performance by as much as 10% in many experiments Variations caused by large rotations, object location, translation, and camera position are most affected.

In every experiment, the comparisons based on the top 5,000 elements of the characteristic signature achieve the same level of performance as the comparison based on the full signature. This indicates that the full comparison is dominated by the elements with the smallest variances.

Are fewer sufficient?

From the plots in Figure 6-2 we see that some subset of the top 5,000 elements dominate the full comparison. In Figure 6-3 we compare configurations ToT-25ˆ3-Top5000, ToT-25ˆ3-Top1975, ToT-25ˆ3-Top500, and ToT-25ˆ3-Top75, which each use successively smaller subsets of the top (lowest variance) elements of the characteristic signature . Across almost every experiment the configurations which use 500 or more of the lowest variance elements perform as well as the full comparison using the entire characteristic signature . In some variations, zoom, rotation and object location, using just 500 elements results in *slightly* lower performance in the low Neyman-Pearson range; however, the performance decrease is extremely small and may not generalize over similar target sets generated from different base images.

When only using the top 75 elements, performance dropped slightly – by at most 10% to 12% (in the camera position variation and in rotation over 360 degrees.) The computational speedup attained however is a factor of 625 $(46,875/75)$ which is quite substantial [3]. This suggest schemes for speeding up queries by first performing a prefiltering query. By allowing large enough Neyman-Pearson criterion for this prefiltering query, we can with arbitrary certainty guarantee that only false-positives are removed in this filter, i.e. we can guarantee that there are no false-negatives, that all the true-positives pass through. Any degree of certainty can be guaranteed trivially with a null operation, as we can set the Neyman-Pearson criterion to be the size entire clutter image set. Clearly this would gain us nothing; however, from the ROC curves in Figure 6-3 we see evidence that in practice, the performance of the ToT-25ˆ3-Top75 and ToT-25ˆ3 techniques converge above an Neyman-Pearson criterion of 800. Thus using such a prefilter can generate about a $3\times$ speedup without much loss of performance.

6.1.3 Smaller branching factors

Partitioning the filter set

In Figure 6-4 we compare configurations which use a smaller set of filters at each level and thus have smaller branching factors. Configuration ToT-9ˆ3 consists of the use of the nine filters which are generated by the successive application of the first three horizontal kernels h_0, h_1, and h_2 followed by the first three vertical filters v_0, v_1, and v_2. This subset of the 25 filters used in ToT-25ˆ3 can be visualized:

$$\begin{array}{cc} & \begin{array}{ccccc} h_0 & h_1 & h_2 & & \end{array} \\ \begin{array}{c} v_0 \\ v_1 \\ v_2 \\ \\ \\ \end{array} & \begin{pmatrix} F_0 & F_1 & F_2 & . & . \\ F_5 & F_6 & F_7 & . & . \\ F_{10} & F_{11} & F_{12} & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{pmatrix} \end{array} \tag{6.1}$$

Configuration ToT-4ˆ3 consists of the use of the nine filters which are generated by the successive application of the last two horizontal kernels h_3 and h_4 followed by the first three vertical filters v_3 and v_4. Thus, consisting of the filters:

[3]There is also a one time cost $\propto O\,(46,875 \log 46,875)$ required for finding the top 75 elements, but this is amortized over the entire database and is insignificant.

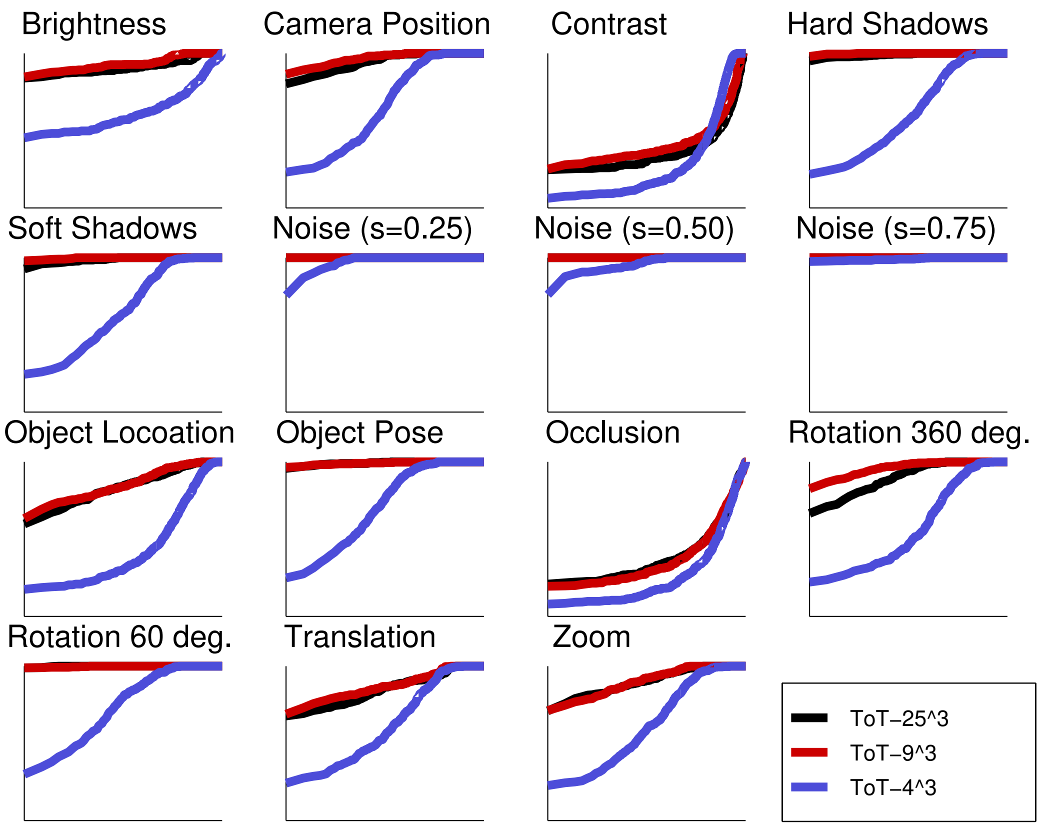


Figure 6-4: ROC curves for configurations which use a smaller set of filters at each level and thus have smaller branching factors.

$$\begin{array}{cc} & \begin{array}{ccccc} & & & h_3 & h_4 \end{array} \\ \begin{array}{c} \\ \\ v_3 \\ v_4 \end{array} & \begin{pmatrix} . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & F_{18} & F_{19} \\ . & . & . & F_{23} & F_{24} \end{pmatrix} \end{array} \tag{6.2}$$

In splitting the filter set in this way, we can examine how important each of the two subgroups of half-filters is in establishing the overall performance of the system.

In each experiment the ToT-4ˆ3 configuration performed significantly worse than the ToT-9ˆ3 and ToT-25ˆ3 configurations. In some experiments the difference in performance was as high as 60% in the low Neyman-Pearson criterion range.

The ToT-9ˆ3 configuration slightly out performed the larger ToT-25ˆ3 configuration in almost every experiment. This is surprising because the characteristic signature generated in the ToT-25ˆ3 configuration is a complete superset of the ToT-9ˆ3 characteristic signature . However, the difference is extremely small in all experiments except for two: large rotations and camera position. The better performance for retrieval of target images which have undergone large rotation is not surprising. The filters which were removed from the filter set are precisely those which have the most specific orientation response. Without them, the ToT-9ˆ3 configuration representation is less sensitive to changes due to orientation. Performance improves because the target images are similar enough that even without this orientation selectivity they are still clustered away from the clutter images, and without the additional orientation-selective dimensions they cluster closer together.

Level variation with smaller branching factor

Though the ToT-9 configuration out performs the ToT-25 model with three levels, the effect of additional levels is unclear. In Figure 6-5 we compare the performance of the ToT-25ˆ1, ToT-25ˆ2, and ToT-25ˆ3 configurations with the ToT-9ˆ1, ToT-9ˆ2, and ToT-9ˆ3 configurations.

In all experiments performance improves for all configurations with the addition of each successive level.

The performance of the ToT-25ˆ1, one level configuration is better than that of the ToT-9ˆ1 by between 5% and 15% in every experiment, except for the case of small rotations and occlusion where their performance was roughly identical.

With the addition of another level, the performance of the 9 branching model, ToT-9ˆ2 is closer to the ToT-25ˆ2 configuration's performance than it was with only a single level. In one case, the contrast variation series, the ToT-9ˆ2 model slightly outperforms the ToT-25ˆ2 model, though both have the exceptionally poor performance which was noted in section 5.1.2.

In every experiment the the ToT-9ˆ3 model is at the same level as, or slightly outperforms the ToT-25ˆ3 model as discussed in section 6.1.3.

As additional levels are added to each configuration, it is notable that the relative improvement in performance is about the same, or perhaps even slightly greater for the ToT-9 series. This suggests that with an additional level, the performance of a ToT-9ˆ4 configuration might be better than that of a ToT-25ˆ4 model. Further the branching factor of 9,

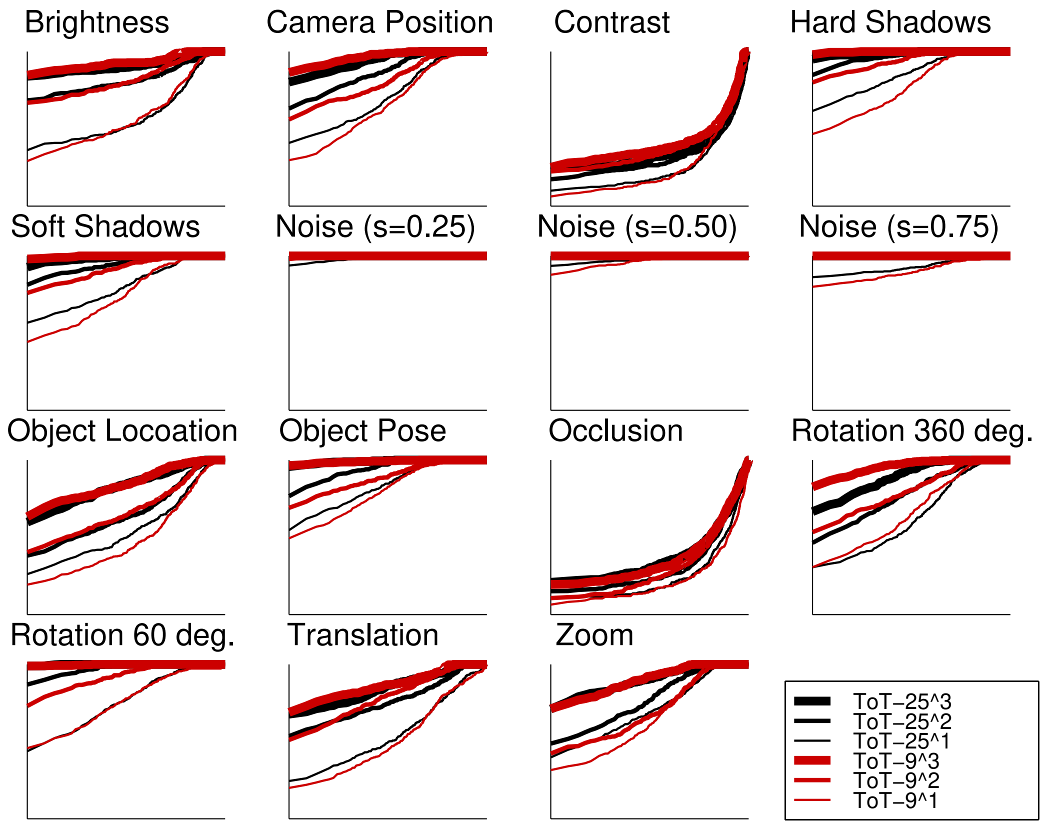


Figure 6-5: ROC curves for 1, 2, or 3 levels of the filter network trees with branching factors of 9 and 25.

as opposed to 25, makes such a computation feasible. A ToT-9ˆ4 configuration results in 19,683 characteristic signature elements, [4] which is less than half the size of the ToT-25ˆ3 characteristic signature . We consider this configuration in Chapter 7.

[4] A ToT-9ˆ4 configuration results in $19,683 = 9^4 \times 3$

Chapter 7

Improving The Texture-Of-Textures Model

The experiments in Chapter 5 measured how well the textures-of-textures model retrieves images which vary due to both physical and post processing variations, and indicated for what types of variations it is not robust. In Chapter 6 we illustrated which components of the filter network tree and the corresponding characteristic signature are important in establishing the observed retrieval rates.

7.1 Filter Network Tree Configurations With Additional Layers

From these two sets of experiments several possible extensions are suggested, which could improve the retrieval rates of the original model described in Chapter 4. Because of the progressive improvements with each level added to the filter network tree, the data in Figure 6-5 suggest that with even more layers performance might continue to improve. With a branching factor of 25, building such a large tree and storing the resulting characteristic signatures for an entire database is prohibitively expensive; however, in section 6.1.3 we showed that when decreasing the set of filters used at each level, performance was not hurt, and even improved slightly in some cases.

Combining this information suggests that a 4-layer filter network tree with a branching factor of 9, which does result in manageable characteristic signatures, could show a large performance gain over the 3-layer models.

In Figure 7-1 we plot the receiver operating curves for a 4-layer filter network tree with a branching factor of 9, ToT-9^4, and the corresponding curves for the equivalent 3-layer configuration ToT-9^3. The effects of adding the additional layer are mixed.

In three experiments, brightness and contrast and occlusion variations, the performance of the 4-layer tree is better than that of the 3-layer. However, the improvement in the low Neyman-Pearson range is really only a significant gain in the brightness experiment where it is a 12% improvement over already notably high retrieval rates of 80 to 85%. In the contrast and occlusion experiments an improvement of 12% on top of the lack-luster performance, around 40% for the 3-layer network, still leaves much to be desired.

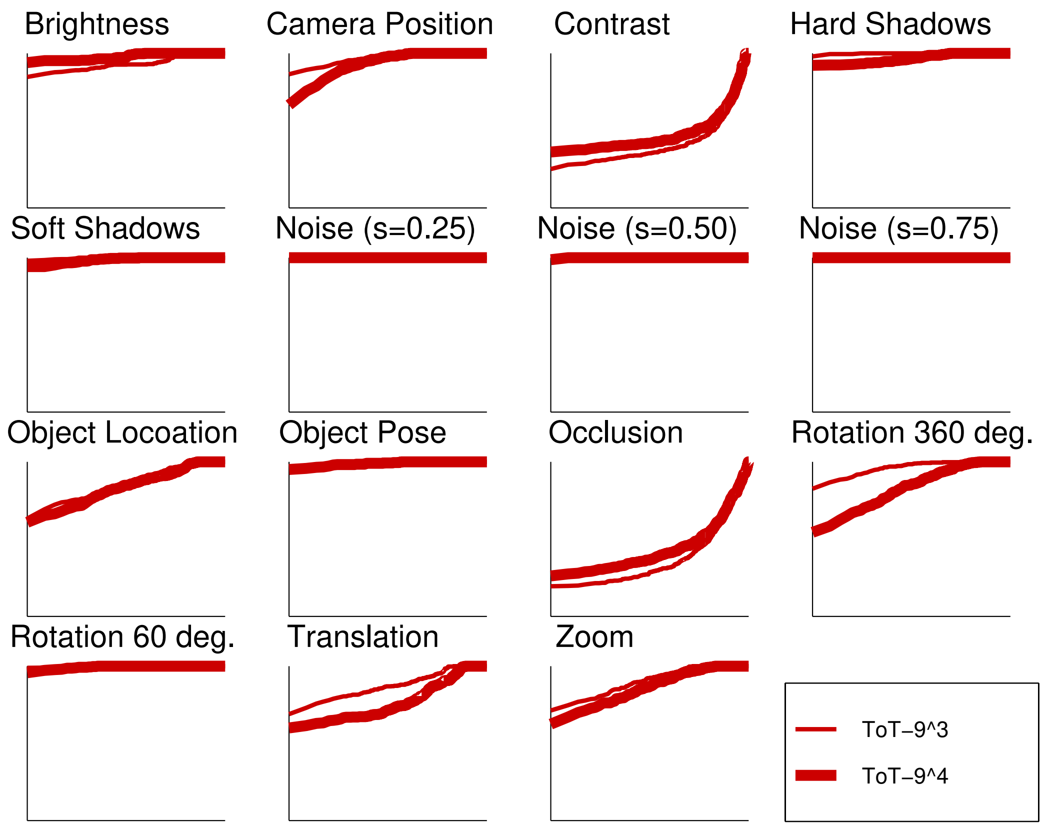


Figure 7-1: The receiver operating curves for a 4-layer filter network tree with a branching factor of 9 and the corresponding curves for the equivalent 3-layer configuration.

In three other experiments — variations of large rotations, translation and zoom — the 4-layer configuration showed significant decreases in performance compared to the 3-layer model. In each case performance dropped between 8 and 12% in the low Neyman-Pearson region. Further, in two additional experiments — camera position, and hard shadows — performance dropped significantly in just the very low Neyman-Pearson region (for camera position) or dropped slightly over the whole low Neyman-Pearson region (hard shadows.) However, These performance drops are small and only serve to indicate that the performance of the 4-layer model is *not better* than that of the 3-layer model in these experiments.

Some insight helps explain why it is on these experiments where the 4-layer filter network trees achieve a lower performance. In the experiments where the 4-layer configuration performs worse than the 3-layer configuration, the lower spatial frequencies across the target image sets are significantly modified by the variations exhibited by the sets. The 4-layer configuration incorporates constraints at a full octave lower in spatial frequency than does the 3-layer model, and as a result, the characteristic signature generated by the 4-layer results in a representation which is less stable with respect to lower frequency variations. The 3-layer model performs its aggregation of feature responses at a higher resolution than does the 4-layer model. In doing so, it is invariant to low frequency changes in the images. Such changes include the rearrangement of parts of the scene, or change of viewing position. Representational invariance to these low-frequency changes is crucial for successful image retrieval in the general case.

7.2 Counteracting The Effects Of Contrast: Characteristic Signature Normalization

In Figure 5-5 the deleterious effects of contrast manipulation on the performance of the textures-of-textures method was made apparent. In section 5.1.2 several strategies were suggested for compensating for this vulnerability. Each solution proposes overcoming the contrast problem by adding contrast invariance at different stages of the characteristic signature computation. The proposed solutions were to

- normalize the original image
- add a sigmoidal non-linearity which would effectively quantize the feature response images at each level
- normalize the characteristic signature

The purpose of removing the effects of contrast is to cause similar images, which differ in appearance because of contrast shifts, to appear more like each other and less like the clutter images in the database. Achieving this is easier in higher dimensional spaces, where clusters can differ because of any one of many dimensions.

Before characteristic signature computation, the input image is in N-dimensional pixel-space. During the first levels of characteristic signature computation, the representation is of lower dimensionality than N. At the final level however, the characteristic signature

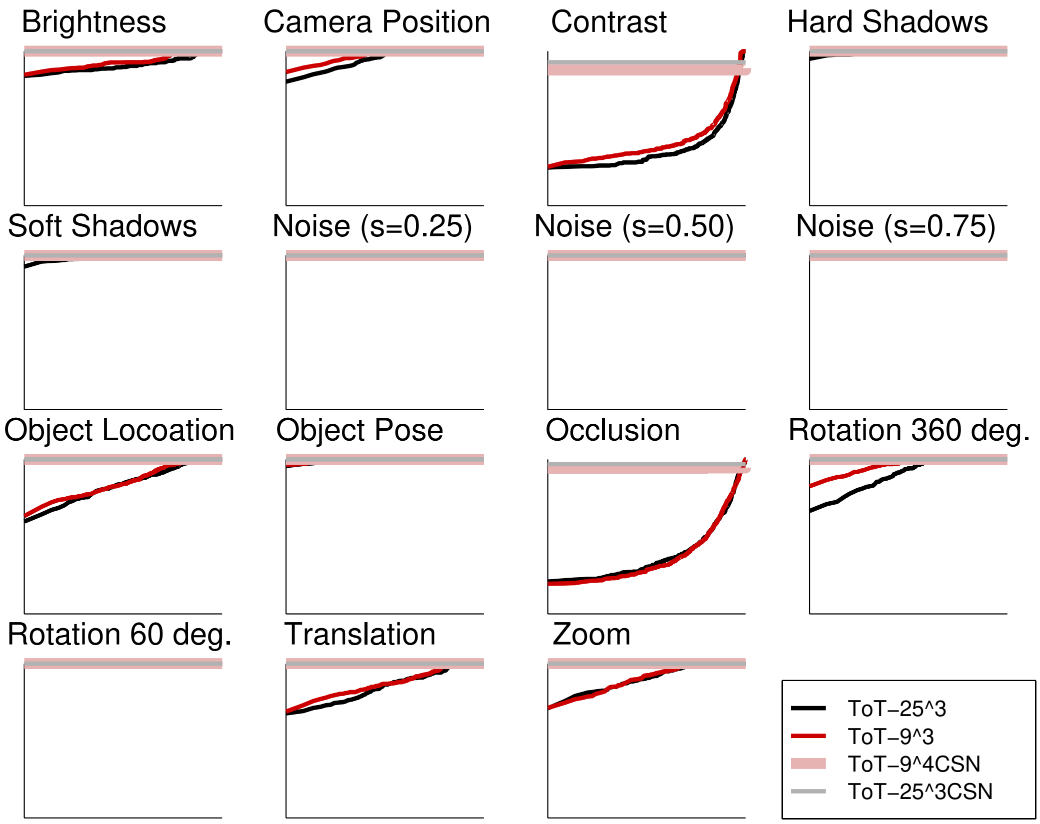


Figure 7-2: Comparison of performance of characteristic signature normalized and non-normalized models.

is of higher dimensionality than the original pixel representation. Therefore, by performing normalization at earlier stages, images which might be distinguishable in characteristic signature space, could be projected on top of one another (making them indistinguishable.) When full characteristic signature is computed before normalization is performed, the maximal advantage from increasing the representational dimensionality is achieved.

Characteristic signature normalization is performed in the obvious way:

$$S_{i,j,k,c}^{\text{Normalized}} = \frac{S_{i,j,k,c}}{\sum_{i',j',k',c'} S_{i',j',k',c'}} \tag{7.1}$$

We repeated the experiments from Chapter 5 using the ToT-25ˆ3CSN configuration, which is the ToT-25ˆ3 with added characteristic signature normalization, and the ToT-9ˆ4CSN configuration, which is ToT-9ˆ4 with characteristic signature normalization. The receiver operating characteristics of each curve are shown in Figure 7-2.

In almost every experiment, both characteristic-signature-normalized techniques achieved perfect performance. This result is rather astounding.

The two experiments where they did not achieve perfect retrieval were precisely those where the non-normalized versions were weakest: in contrast and occlusion variation.

Though they did not achieve perfect performance, they were able to achieve immediate retrieval rates of around 90% and 95% for contrast and occlusion variation respectively.

However, this success was at the cost of not retrieving the last few percent until the clutter images were exhausted. This quantized behavior is the result of the fact that in both contrast reduction and occlusion, information is lost. In occlusion the cause for this loss is clear, some pixels are completely covered, and the original values lost. For contrast, the cause is slightly subtler; when the contrast of the image is decreased and stored in integer valued pixels, the fractional portion is lost, making the process uninvertable. For the last few percent of the target images the information lost was critical in accurately determining similarity.

In both cases the ToT-25ˆ3CSN configuration performed slightly better than did ToT-9ˆ4CSN, achieving 93% compared to 88% for contrast variation and 97% versus 94% for occlusion. Though this difference is of only limited significance, the ability of each model to generalize when presented with query images which are less similar than those used in these experiments, is very different as evidenced by the quality of the images retrieved.

7.3 Qualitative Performance In Real World Queries

Both the ToT-25ˆ3CSN and ToT-9ˆ4CSN configurations achieve nearly perfect performance on the experiments in Chapter 5. However, in practice that the ToT-25ˆ3CSN results in more robust generalization when given query images which specify a target set which is much more loosely defined than those used in the Chapter 5 experiments.

To see these effects, we present several anecdotal query and response sets, which are indicative of the general performance of the systems when presented with realistic queries.

In Figures 7-3 and 7-4 we compare the image set retrieved by the characteristic-signature-normalized model, ToT-9ˆ4CSN, to the set retrieved by the ToT-25ˆ3 model, when presented with query images which contain two cars of different colors, in different surroundings. In this case, the target image class — i.e. those for which I was searching — were images of cars. Query results are shown for each model, using the world wide web interface, which is accessible via:

`http://www.ai.mit.edu/~jsd/Research/ImageDatabase/Demo`

In the left panel of Figure 7-3 the query images are shown, and in the right panel are the thirty top responses. Even though the ToT-9ˆ4CSN model achieved virtually perfect performance in the variation experiments, in a realistic query, we find that the model does not generalize well. In the set of thirty responses, only two cars are present.

In Figure 7-4 the same query images are used, and even though the ToT-25ˆ3 model did not perform as well on the variation experiments, in this realistic query it was better able to capture the general visual notion suggested by the query images. In the set of thirty responses, eleven cars are present.

The remaining question is: even though characteristic signature normalization improves the performance of the ToT-25ˆ3CSN technique on the Chapter 5 experiments, will that performance boost translate into more diverse queries. In Figure 7-5 we show the results of the same query using the ToT-25ˆ3CSN model. In the set of thirty responses, 17 cars are present. Furthermore, of the images which are not cars, several are airplanes which have

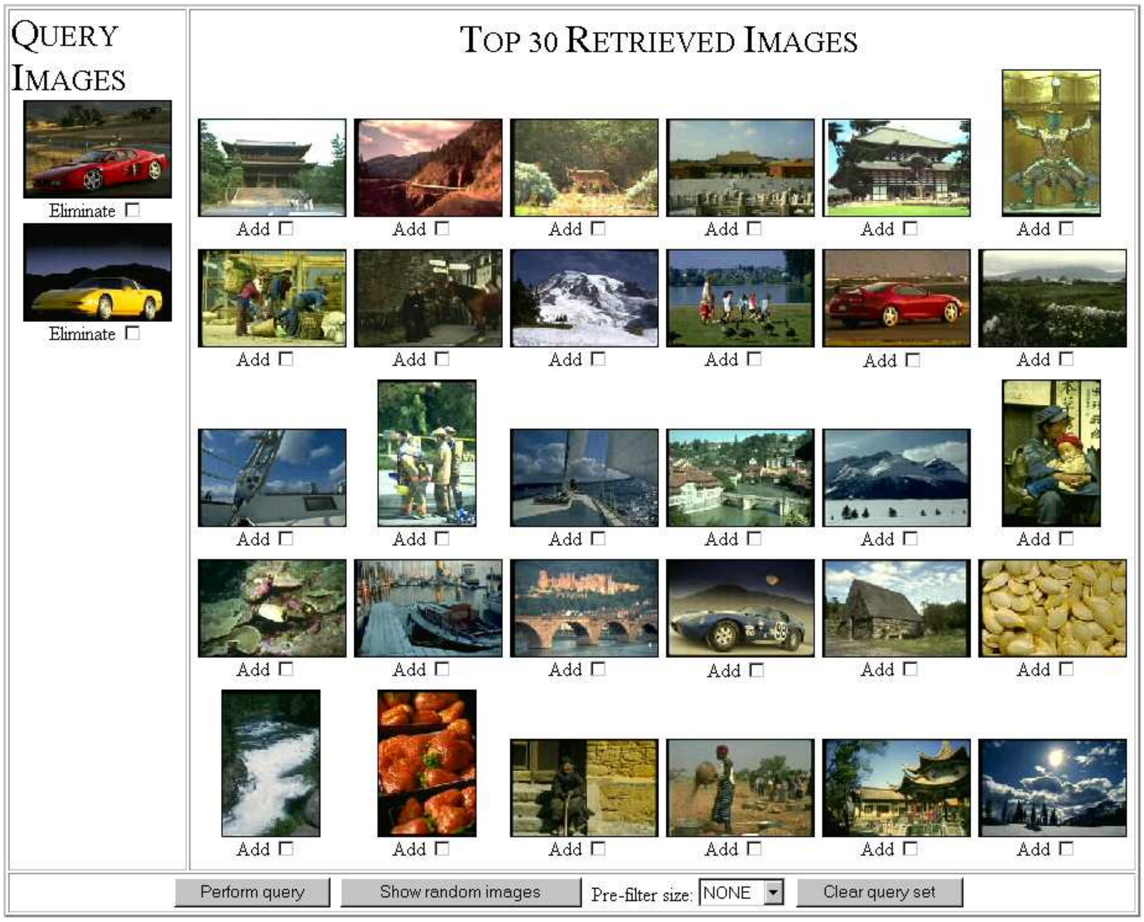


Figure 7-3: A query for cars using the WWW interface, and the retrieval results of the textures-of-textures filter network tree configuration with 4 levels, a branching factor of 9 and characteristic signature normalization. Only two cars are found.

Figure 7-4: The same query shown in Figure 7-3, using the filter network tree configuration described in Chapter 4, with 3 levels, a branching factor of 25 (no characteristic signature normalization.) Eleven of the 30 retrieved images contain cars.

Figure 7-5: The same query shown in Figure 7-3, using a filter network tree configuration with 3 levels, a branching factor of 25 and characteristic signature normalization. Seventeen of the 30 retrieved images contain cars.

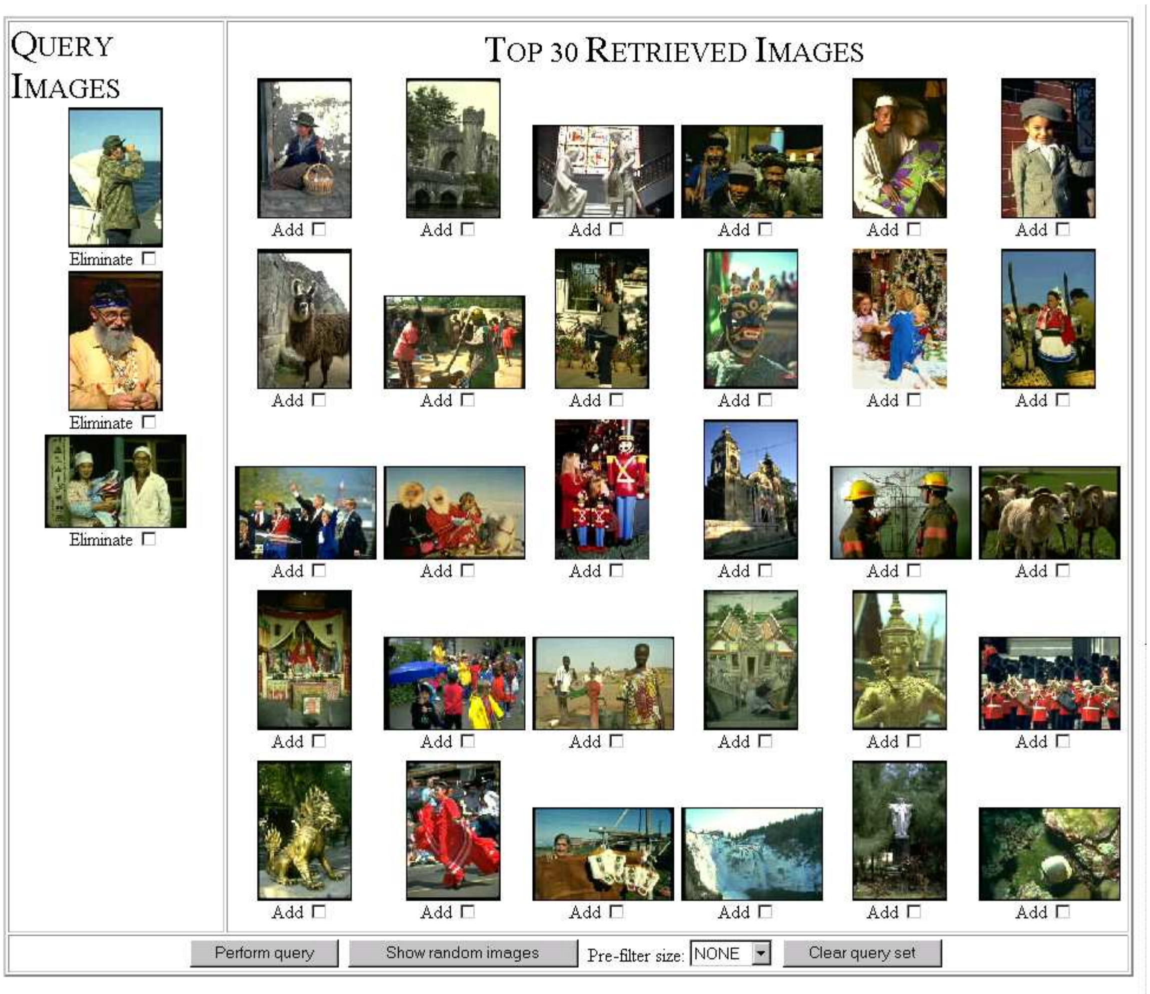


Figure 7-6: A difficult query aimed at getting images of "people wearing hats" and the images retrieved with using a filter network tree configuration with 3 levels, a branching factor of 25 and characteristic signature normalization. Eleven images contain people (or statues of people) wearing hats.

very similar appearances. Even though this evidence is only anecdotal, the performance of the ToT-25ˆ3CSN model seems to be better than the other models for all of the queries which we have compared in this way.

7.4 "So how good is it, really?"

Even with the performance experiments in the preceding chapters, it is still difficult to fully characterize the performance of the textures-of-textures retrieval technique. This is a fundamental problem of the domain. Images vary from each other in an astronomical number of ways, and similarity is perceived by human observers based upon complex interactions between recognition, cognition, and assumption. It seems unlikely that an absolute criterion for image similarity can ever be determined, or if one truly exists.

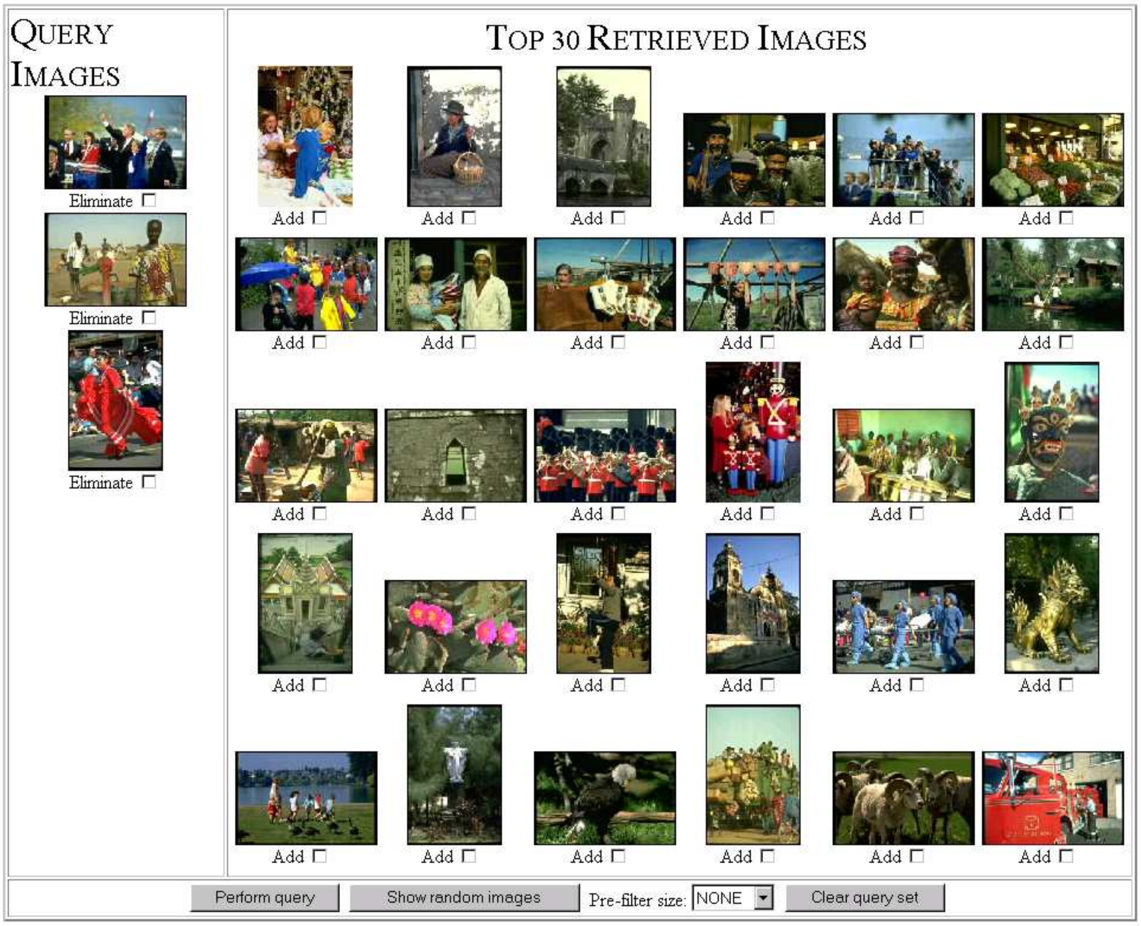


Figure 7-7: Another difficult query designed to return "people not wearing hats;" a slightly different target set than in Figure 7-6. Ten images contain people not wearing hats.

In Figures 7-6 and 7-7 two queries are shown designed to retrieve people with (in Figure 7-6) and without (in Figure 7-7.) In the retrieved images for the “with hats” query, 11 images contained people with hats and 9 contained images of people without. In the “no hats” query, 8 images contained people with hats and 10 contained images of people without. Is that significant? Probably not. But it is remarkable that in both cases image of people were returned. That in and of itself is a success.

However, this raises an additional point. When asked what he would retrieve from a database, when presented with the “no hats” query in Figure 7-7, a human observer replied that the objective was to find images of people with contained only their upper body. The problem is this is a valid interpretation of the query images. When presented with this query, how is a computer system to know what is the key element? As it happens, the system also returned 9 images which contain only the upper body.

So how good is it? This is truly difficult to quantify. The experiments performed in the preceding chapters give some indication that the system can recognize objects and scenes under a variety of conditions. However, the performance of the system is still no where near the ability of a human observer. When presented with and image, human observers can recall and incorporate vast amounts of learned knowledge. For example, when presented the query images Figure 7-7, which contain people, or those in Figures 7-3 through 7-5, which contain cars — both objects with which humans are intimately familiar — human observers can use that familiarity to infer visual features, and semantic meaning which is only suggested by the images themselves.

Even when presented with objects with which they are unfamiliar, the visual experience of a human observer, built up over the course of a lifetime, can be used to infer physical and visual properties.

So how good is it, really? Compared to the methods which form the basis of other techniques which have been used for image retrieval, it appears to perform extremely well. Compared to human observers, there is still a long way to go. The importance of this research, however, is that it suggests a new methodology for approaching the problem of image recognition — a new methodology which appears to be robust and extensible. With additional research, development and experimentation, the full potential of this approach will be explored and discovered.

Chapter 8

Concluding Remarks

We have presented three novel techniques: for synthesizing textures, texture discrimination, and image retrieval. Each technique uses explicit representations which approximate the statistical distributions over images. The distribution approximations used share the property that they capture visual structure by incorporating feature constraints at multiple resolutions. A key component in their successes is their requirement that each constraint is *jointly* satisfied — that the *joint* occurrence of the required features occurs at multiple resolutions.

The techniques presented here achieve remarkable levels of performance. The synthesized textures more successfully capture the characteristics of input textures than do previous techniques. Using the flexible histogram measure classification of natural textures indicates a high level of specificity, and recent results on target detection in SAR imagery are encouraging. The textures-of-textures approach to image recognition perform better on retrieval tasks than to more conventional approaches, and can be analyzed and extended to further improve its performance.

However, the true success and contribution of these techniques is that they present a novel solution. Each achieves reasonable levels of performance in their domain using new approaches — approaches which can be incorporated into existing systems, which present alternatives to current methods, and which can be be applied to new problems.

Appendix A

Specification competing retrieval techniques

In this appendix we include the details of the techniques compared to the textures-of-textures method in Chapter 5. Their implementations are straightforward, and *standard*. The details are included here for the sake completeness.

A.1 Color histogram bin determination

Each color histogram quantizes the three color axis – either red, green, blue or hue, saturation and value – into K mutually exclusive cells. The resulting quantized space forms a three dimensional array of cells, $C[a][b][c]$, into one which each color falls. Each cell forms one bin in the histogram, $B_{aK^2+bK+c} = C[a][b][c]$:

$$B\left[aK^2 + bK + c\right] = \sum_{p \in pixels} \left\{ \begin{array}{cc} & \delta\left[(a)\,\frac{256}{K} \leq p_r < (a+1)\,\frac{256}{K}\right] \\ \times & \delta\left[(b)\,\frac{256}{K} \leq p_g < (b+1)\,\frac{256}{K}\right] \\ \times & \delta\left[(c)\,\frac{256}{K} \leq p_b < (c+1)\,\frac{256}{K}\right] \end{array} \right\} \tag{A.1}$$

where $\delta(\cdot) = 1$ iff the condition in its argument is true. Thus each histogram contains K^3 bins. For the 216 histogram $K = 6$, and for the 512 bin histograms, $K = 8$.

The function used to convert from the RGB to the HSV colorspace was modified from the sample in from Folly *et al.* ([22]), and is shown in Table A.1

A.2 Color histogram comparison

Histograms were compared with a standard χ^2 measure [52]:

```
HSVPixel RGBToHSV(const RGBPixel &pxl)
{
    byte yH,yS,yV;

    byte yMax = FiplMax(pxl);
    byte yMin = FiplMin(pxl);

    yV=yMax;

    if (yMax!=0)
        yS=(yMax-yMin);
    else
        yS=0;

    if (yS==0)
        yH=0; //UNDEFINED
    else
    {
        byte yDelta = yMax-yMin;
        if (pxl.R()==yMax)
            yH=42*1+42*(pxl.G()-pxl.B())/yDelta;
        else if (pxl.G()==yMax)
            yH=42*3+42*(pxl.B()-pxl.R())/yDelta;
        else if (pxl.B()==yMax)
            yH=42*5+42*(pxl.R()-pxl.G())/yDelta;
    }

    return HSVPixel(yH,yS,yV);
}
```

Table A.1: RGB to HSV colorspace conversion routine, modified from [22].

$$\chi^2 = \sum_{i \in bins} \frac{\left(B_i^{\text{model}} - B_i^{\text{test}}\right)^2}{B_i^{\text{model}} + B_i^{\text{test}}} \tag{A.2}$$

For the combined histogram models, the χ^2 difference between the test image histogram and the sum of the histograms generated by the two query images, was used directly as the similarity measure used for ranking.

For the nearest neighbor models, the similarity measure used for ranking was:

$$\min\left[\chi^2\left(I_{query_1}\right), I_{test}, \chi^2\left(I_{query_2}\right), I_{test}\right] \tag{A.3}$$

A.3 Correlation comparison measure

The correlation function used for comparing a query image to the model image was the sum pixel difference measured with the L_2 norm:

$$\sum_{p \in pixels} \sqrt{\sum_{c \in colors} \left(p_c^{\text{query}} - p_c^{\text{test}}\right)^2} \tag{A.4}$$

The similarity measure used was the minimum correlation of the test image with either of the two query images.

Bibliography

[1] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In M. S. Landy and J. A. Movshon, editors, *Computational Models of Visual Perception*, pages 3–20. MIT Press, Cambridge MA, 1991.

[2] J. L. Arrowood and M. J. T. Smith. Exact reconstruction analysis/synthesis filter banks with time-varying filters. In *Proc. ICASSP '93*, pages III–233–236, 1993.

[3] A. J. Bell and T. J. Sejnowski. The 'independent components' of natural scenes are edge fliters. submitted to Vision Research.

[4] J. R. Bergen. Theories of visual texture perception. In D. Regan, editor, *Vision and Visual Dysfunction*, volume 10B, pages 114–134. Macmillian, New York, 1991.

[5] J. R. Bergen and E. H. Adelson. Early vision and texture perception. *Nature*, 333(6171):363–364, 1988.

[6] J. R. Bergen and B. Julesz. Rapid discrimination of visual patterns. *IEEE Transactions on Systems Man and Cybernetics*, 13:857–863, 1993.

[7] J. R. Bergen and M. S. Landy. Computational modeling of visual texture segregation. In M. S. Landy and J. A. Movshon, editors, *Computational Models of Visual Perception*, pages 253–271. MIT Press, Cambridge MA, 1991.

[8] Phillip Brodatz. *Textures: a Photographic Album for Artists and Designers*. Dover Publications, New York, 1966.

[9] P. Burt. Fast filter transforms for image processing. *Communications on Graphics and Image Processing*, 16:20–51, 1981.

[10] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[11] T. Chen and P. P. Vaidyanathan. Multidimensional multirate filters and filter banks derived from one dimensional filters. *Electronics Letters*, pages 225–228, January 1991.

[12] C. Chubb and M. S. Landy. Orthogonal distribution analysis: A new approach to the study of texture perception. In M. S. Landy and J. A. Movshon, editors, *Computational Models of Visual Perception*, pages 291–301. MIT Press, Cambridge MA, 1991.

[13] Corel. Corel stock photography. Web: http://commerce.corel.ca/.

[14] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropogation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[15] J. G. Daugman. Uncertainty relation for resolution in space spatial frequency and orientation optimized by two–dimensional visual cortical filters. *Journal of the Optical Society of America*, A 2:1160–1169, 1985.

[16] J. S. De Bonet. Flexible histograms: Multiresolution texture discrimination model. http://www.ai.mit.edu/people/jsd/Research/Publications, April 1997.

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

[18] J. S. De Bonet and P. Viola. Rosetta: An image database retrieval system. In *Proceedings from Image Understanding Workshop*, San Mateo, CA, 1997. Morgan Kaufmann.

[19] J. S. De Bonet and Q. Zaidi. Comparison between spatial interactions in perceived contrast and perceived brightness. Vision Research, (In submission), 1996.

[20] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[21] Excalibur. The excalibur project. Web: http://www.excalib.com/.

[22] J.and A. van Dam Foley, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, second edition, 1990.

[23] W. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[24] Benjamin Friedlander and Boaz Porat. Detection of transient signals by the gabor representation. *IEEE Trans. ASSP*, 37(2):169–180, February 1989.

[25] D. Gabor. Theory of communication. *J IEE*, 93:429–459, 1946.

[26] A. Gagalowicz. Texture modelling applications. *The Visual Computer*, 3:186–200, 1987.

[27] A. Gagalowicz and S. D. Ma. Model driven synthesis of natural textures for 3–D scenes. *Computers and Graphics*, 10:161–170, 1986.

[28] N. Graham, J. Beck, and A. Sutter. Nonlinear processes in spatial-frequency channel models of perceived texture segregation: Effects of sign and amount of contrast. *Vision Research*, 32:719–743, 1992.

[29] N. Graham, A. Sutter, and C. Venkatesan. Spatial-frequency and orientation-selectivity of simple and complex channels in region segregation. *Vision Research*, 33:1893–1911, 1993.

[30] D. M. Green and J. A Swets. *Signal Detection Theory and Psychophysics*. John Wiley and Sons, Huntington, N.Y., 1966.

[31] MIT LV Group. Massachusetts institute of technology learning & vision group texture database. Web: http://www.ai.mit.edu/projects/lv.

[32] D. J. Heeger and J. R. Bergen. Pyramid based texture analysis/synthesis. In *Computer Graphics*, pages 229–238. ACM SIGGRAPH, 1995.

[33] B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, IT–8:84–92, 1962.

[34] M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: the candid approach. *SPIE Vol. 2420 Storage and Retrieval for Image and Video Databases III*, pages 238–248, 1995.

[35] M. Kelly, T.M. Cannon, and D.R. Hush. Query by image example: the candid approach. *Storage and Retrieval for Image and Video Databases III*, 2420:238–248, 1995.

[36] MIT Media Lab. Massachusetts institute of technology media lab photobook image retrieval system. Web: http://vismod.www.media.mit.edu/vismod/demos/photobook/index.html.

[37] M. S. Landy and J. R. Bergen. Texture segregation and orientation gradient. *Vision Research*, 31:679–691, 1991.

[38] V. Larson, L. M. Novak, and C. Stewart. Joint spatial-polarimetric whitening filter to improve SAR target detection performance for spatially distributed targets. *SPIE Conf. on Alg. for SAR Imagery*, April 1994.

[39] P Lipson, E Grimson, and P Sinha. Configuration based scene classification and image indexing. In *Computer Vision and Pattern Recognition*, 1997.

[40] M. R. Luettgen, W. C. Karl, A. S. Willsky, and R. R. Tenney. Multiscale representations of markov random fields. *IEEE Trans. on Signal Processing*, 41(12):3377–3396, 1995.

[41] S. D. Ma and A. Gagalowicz. Determination of local coordinate systems for texture synthesis on 3-D surfaces. *Computers and Graphics*, 10:171–176, 1986.

[42] D. Marr. *Vision*. W. H. Freeman and Company, 1982.

[43] MBVLab. Model based vision lab. http://www.mbvlab.wpafb.af.mil.htm.

[44] P. Moulin. A wavelet regularization method for diffuse radar-target imaging and speckle-noise reduction. *Journal of Mathematical Imaging and Vision*, 3(1):123–134, January 1993.

[45] P. Moulin, J. A. O'Sullivan, and D. L. Snyder. A method of sieves for multiresolution spectrum estimation and radar imaging. *IEEE Trans. Inform. Theory*, 38(2):801–813, March 1992.

[46] V. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project: querying images by content using color, texture, and shape. *IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science & Technology*, 1908:173–187, 1993.

[47] L. M. Novak, G. J. Owirka, and C. M. Netishen. Performance of a high-resolution polarimetric SAR automatic target recognition system. *The Lincoln Laboratory J.*, 6(1):11–24, 1993.

[48] Greg Pass and Ramin Zabih. Histogram refinement for content-based image retrieval. *IEEE Workshop on Applications of Computer Vision*, 1996.

[49] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. Technical report, MIT Media Lab, 1995.

[50] R. W. Picard and T. Kabir. Finding similar patterns in large image databases. *ICASSP*, V:161–164, 1993.

[51] T. Poggio and D. Beymer. Learning networks for face analysis and synthesis. In Martin Bichsel, editor, *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, pages 160–165, Zurich, Switzerland, 1995.

[52] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipies In C: The Art and Science of Computing*. Cambridge University Press, New York, 1988.

[53] QBIC. The ibm qbic project. Web: http://wwwqbic.almaden.ibm.com/.

[54] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.

[55] R. P. N. Rao and D.H. Ballard. Object indexing using an iconic sparse distributed memory. Technical Report TR-559, University of Rochester, 1995.

[56] D. de Ridder. Shared weights neural networks in image analysis. Master's thesis, Delft University of Technology, March 1996.

[57] D. de Ridder and A.and Duin R. P. W. Hoekstra. Feature extraction in shared weights neural networks. In E.J.H. et al Kerckhoffs, editor, *Proc. of the second annual conference of the ASC*, pages 289–294, Lommel, Belgium, June 1996. ASCI.

[58] A. J. Robinson and F. Fallside. Static and dynamic error propagation networks with application to speech coding. In D. Z. Anderson, editor, *Neural Information Processing Systems*. American Institute of Physics, 1988.

[59] S Santini and R Jain. Gabor space and the development of preattentive similarity. In *Proceedings of ICPR 96*. International Conference on Pattern Recognition, Vienna, August 1996.

[60] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. *IEEE Second International Conference on Image Processing*, October 1995.

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

[62] Izzet Sirin and H. Altay Guvenir. An algorithm for classification by feature partitioning. Technical Report CIS-9301, Bilkent University, 1993. Also in Bilkent University.

[63] B. Spehar, J. S. De Bonet, and Q. Zaidi. Brightness induction from uniform and complex surrounds: A general model. *Vision Researc*, pages 1893–1906, 1996.

[64] K.-K. Sung and T. Poggio. .example-based learning for view-based human face detection. In *Proceedings from Image Understanding Workshop*, pages 843–850, San Mateo, CA, November 1994. Morgan Kaufmann.

[65] M.J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer VISION*, 7(1):11–32, November 1991.

[66] MStar target data. Model based vision lab. http://www.mbvlab.wpafb.af.mil/public/MBVDATA/mstrcd1.htm.

[67] G. Turk. Genereating textures on arbitrary surfaces using reaction-diffusion. In *Computer Graphics*, volume 25, pages 289–298. ACM SIGGRAPH, 1991.

[68] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[69] M. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82, 1986.

[70] M. Vetterli. Filter banks allowing perfect reconstruction. *Signal Processing*, 10(3):219–244, April 1986.

[71] M. Vetterli and C. Herley. Wavelets and filter banks: Relationships and new results. In *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, NM, 1990.

[72] P. Viola. Feature-based recognition of objects. In *AAAI Fall Symposium on Learning and Computer Vision*, 1993.

[73] Paul Viola. Complex feature recognition: A bayesian approach for learning to recognize objects. Technical Report 1591, MIT AI Lab, 1996.

[74] Virage. The virage project. Web: http://www.virage.com/.

[75] A. Witkin and M. Kass. Reaction–diffusion textures. In *Computer Graphics*, volume 25, pages 299–308. ACM SIGGRAPH, 1991.

[76] A.S. Willsky W.W. Irving and L.M. Novak. A multiresolution approach to discriminating targets from clutter in sar imagery. *Proc. SPIE*, 2487, 1995.

[77] S. C. Zhu, Y. Wu, and D. Mumford. Filters random fields and maximum entropy(frame): To a unified theory for texture modeling. *To appear in Int'l Journal of Computer Vision*, 1996.