# Flexible Histograms: A Multiresolution Texture Discrimination Model

**Jeremy S. De Bonet & Paul Viola**
Artificial Intelligence Laboratory
Learning & Vision Group
545 Technology Square Massachusetts Institute of Technology
Cambridge, MA 02139

EMAIL: jsd@ai.mit.edu & viola@ai.mit.edu
HOMEPAGE: http://www.ai.mit.edu/projects/lv

## Abstract

In this paper 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. Preliminary experiments indicate that this approach may have sufficient discrimination power to perform target detection is synthetic aperture radar (SAR).

Submitted to *NIPS 1997*. Section: Visual Processing. Preferred format: Talk.

## 1 Introduction

The notion of texture is difficult to capture formally. Textures are usually the output of some random physical process wherein local structure is repeated seemingly at random and there is a lack of global structure. So while the fur of a leopard is considered to be a texture, the entire leopard is not. This distinction is unavoidably arbitrary. While few would argue that the bark of a tree is a texture, there is real debate whether the global structure of a tree is texture or not. Another definition of texture might include some reference to the simplicity of the generation process. The branching of a tree is highly redundant and can be captured accurately with simple recursive rules. Viewed in this light the bark of the tree and its branches are not necessarily different. A similar argument might be made for a wall of bricks: the wall texture is a simple arrangement of brick texures. We will present an approach to texture recognition that is successful at modelling random repeating textures such as fur or bark. The approach naturally and automatically extends to structural patterns at many scales.

Previous approaches to texture have focussed on models of local structure[1]. For example in visual psychophysics many models of the human ability to discriminate subtly different texture have been proposed. Beginning with the work of Julesz, these texture discrimination models typically analyze a patch of texture by computing the outputs of local oriented filters such as the Gabor filters (Julesz, 1962; Bergen and Adelson, 1988; Bergen and Landy, 1991; Chubb and Landy, 1991). In related computational approaches, a single feature vector which contains the outputs Gabor-like filters at multiple scales is used to describe a patch of "texture". Two textures are said to be different if their Gabor features vectors are different — typically this is measured as the $L_2$ distance between the vectors. Motivation for the use of these Gabor-like filters comes in part from the parallel between Gabor filters and the early processing of the retina and visual cortex. In addition, the Gabor filters are "optimal" detectors for local spatial frequency. These filters tease apart local information about frequency and orientation[2].

While these Gabor-like models are both simple and effective, they make a serious limiting assumptions regarding the definition of texture. For example, it is assumed that the Gabor features for different patches of the same texture are the same. This may be true for a very fine texture like sand or fur, it is unlikely to be true for a texture with significant internal structure like a brick wall or a striped shirt. Typically such Gabor vector techniques take one of three approaches to this problem. Some techniques simply rule out such images as being outside the scope of texture models. Other techniques blur out the Gabor feature vectors which are computed at different points in a texture. This has the effect of computing the average Gabor vector for the texture. Still other techniques attempt to learn which types of variation are acceptable in a texture and which are not. The most complex of these attempt to learn a collection of Gabor feature vectors which describe a type of texture (**?**).

Recently, the notion that a texture can be described as a Gabor feature vector has been applied in the problem of *synthesizing* texture images (Heeger and Bergen, 1995; Zhu, Wu and Mumford, 1996). Interestingly these researchers have found that it is not sufficient to model a texture as a single, or even a few, Gabor feature vectors. Instead these approaches generate a new texture from an old one by matching the observed distributions of filter responses. Even for simple textures there will significant variation in the output of filters at different location in the image. In order for textures to match in appearance they must match in these observed distributions. Because of the computational complexity involved Heeger and Bergen chose to match the distributions of the filters independently. There approach generated very pleasing results in simple textures, but failed on more complex structural textures like "brick wall". In (De Bonet, 1997) we presented a related texture generation procedure which not only matched the the observed distribution of each filter independently, but also modelled the statistical dependence between filters. In this work we invert this generation procedure and use it as the basis for a texture discrimination model.

[1] There have been many models of texture discrimination which include pixel co-occurence matrices and Gaussian Markov random fields (**?**; Chellappa and Chatterjee, 1985). We will not discuss these here.

[2] "Gabor-like" filters can be computed in many ways. Many authors have used various wavlet pyramids instead of the direct implementation of Gabor filters(**?**). Wavlets have many additional valuable properties, including the fact that wavlet filter computation can be inverted. In this paper we refer to the different types of filters as Gabor-like and differentiate only when necessary.

## 2 Analysis Pyramid and Parent Vectors

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 vector* 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: $G_{n+1} = 2{\downarrow}(G_n \otimes K_{Gaussian})$ 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 $n$: $F_n^i = G_n \otimes K_i$

At each location $(x, y)$ in the original image we construct a parent vector, $V$, of the responses of each of $M$ filters at a location $(x, y)$ at each of the $N$ resolutions:

$$\begin{aligned}\vec{V}(x,y) = \; & \left[F_0^0(x,y), F_0^1(x,y), \cdots, F_0^M(x,y),\right. \\ & F_1^0\left(\left\lfloor\tfrac{x}{2}\right\rfloor, \left\lfloor\tfrac{y}{2}\right\rfloor\right), F_1^1\left(\left\lfloor\tfrac{x}{2}\right\rfloor, \left\lfloor\tfrac{y}{2}\right\rfloor\right), \cdots, F_1^M\left(\left\lfloor\tfrac{x}{2}\right\rfloor, \left\lfloor\tfrac{y}{2}\right\rfloor\right), \cdots, \\ & \left. F_N^0\left(\left\lfloor\tfrac{x}{2^N}\right\rfloor, \left\lfloor\tfrac{y}{2^N}\right\rfloor\right), F_N^1\left(\left\lfloor\tfrac{x}{2^N}\right\rfloor, \left\lfloor\tfrac{y}{2^N}\right\rfloor\right), \cdots, F_N^M\left(\left\lfloor\tfrac{x}{2^N}\right\rfloor, \left\lfloor\tfrac{y}{2^N}\right\rfloor\right)\right] \end{aligned} \quad (1)$$

## 3 Flexible Histograms

For every pixel in the model image, we define a histogram bin which is a measure of the frequency of regions in a test image which are within a threshold distance, $T$. Using the $L_\infty$ norm makes the requirement that each feature at each resolution explicit:

$$B_{x,y} = \frac{1}{Z} \sum_{x',y'} \Theta\left(T - \left\| \left[\vec{V}(I,x,y)\, \vec{V}(I',x',y')\right] (D)^{-1} \right\|_\infty\right) \quad (2)$$

where each component $d_{i,i}$ of diagonal matrix $D$ scales the corresponding feature response; and $\Theta(\cdot) = 1$ if its arument is greater than 0, and is 0 otherwise; $Z$ is a normalization factor. Thus, each bin provides a count of pixels in image $I'$ which fall within a $N \times M$ dimensional hypercube in resolution and feature space centered at $\vec{V}(I,x,y)$.

Because the bin labels used to generate these histograms are dependent on the model image, we term them "flexible." In this way, when searching for different types of textures (i.e. when using different model images) the bins used in the histogramming process are specialized.

Such a histogram can be viewed as a discrete approximation to a Parzen density estimator. We use discrete form because is allows for the use of a low-to-high resolution scheme which drastically reduces the computations required to compute the density estimate. From equation (1), we see that the parent vectors 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 vectors. Thus, this discrete approximation allows for whole image regions to be "pruned" when equation (2) is not satisfied at a low resolution.

As in a continuous Parzen density estimator the bins in the flexible histogram are non-exclusive therefore parent vector can fall into multiple bins. This has the effect of increasing the relative importance of model image parent vectors which occur more often. The normalization factor $Z$, compensates for this, allowing equation (2) to be a probability distribution. For making similarity measurements however, $Z$ factors out, and we therefore need not be concered with estimating it.

A histogram for a test image is computed by accumulating the number of parent vectors in the test image which are within threshold distance of each model image parent vector. Similarly a flexible histogram is computed for the model image with respect to *itself*.

Since corresponding bins in each histogram are counts of parent vectors in each image which are within threshold of *the same* parent vector in the model, we can compare them using the standard $\chi^2$ test. Because model histogram is measured with repect to itself, no bin is empty, so the denominator in the $\chi^2$ ratio is greater than zero for all bins, achieving the criterion that though perhaps very unlikely, no two images should be measured to contain different textures with complete certainty. A similarity likelihood is acquired from the inverse of $\chi^2$ difference measure.

The $\chi^2$ measure can be viewed as a comparison of Parzen density estimates:

$$\chi^2 = \frac{\left(B\left(x,y\right) - B'\left(x,y\right)\right)^2}{B\left(x,y\right)} \propto \sum_{x,y} \left\{ \begin{array}{rl} & \sum_{x',y'} R\left[\vec{V}\left(I,x,y\right) - \vec{V}\left(I,x',y'\right)\right] \\ - & \sum_{x',y'} R\left[\vec{V}\left(I,x,y\right) - \vec{V}\left(I',x',y'\right)\right] \end{array} \right\}^2 \quad (3)$$

where $R\left(\cdot\right)$ is the Parzen density distance function.

A binary decision is made by comparing the likelihood measure of each image to a threshold $\eta_{model}$. Using standard $\chi^2$ tables a value for $\eta_{model}$ for any desired level of certainty can be determined. However, we do not assume that the flexible histogram completely describes the model texture and choosing a fixed likelihood can eliminate true-positives which fall below the threshold, but could be easily decteded as it is far above the false-positives. By choosing $\eta_{model}$ is empirically we maximize the percentages of true-positives while guaranteeing an expected level of false-positives. The curve generated by varying $\eta_{model}$, known as the receiver operating characteristics (ROC) curve, will be shown for several sets of data in the experiments below.

## 4 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.

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 vectors in the additional model images, which (if they are truly examples from the same underlying generative distribution) will increase the accuracy of the histogram approximation to the distribution.

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