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).

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:

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

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_∞ 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 χ^2 test. Because model histogram is measured with repect to itself, no bin is empty, so the denominator in the χ^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 χ^2 difference measure.

The χ^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.

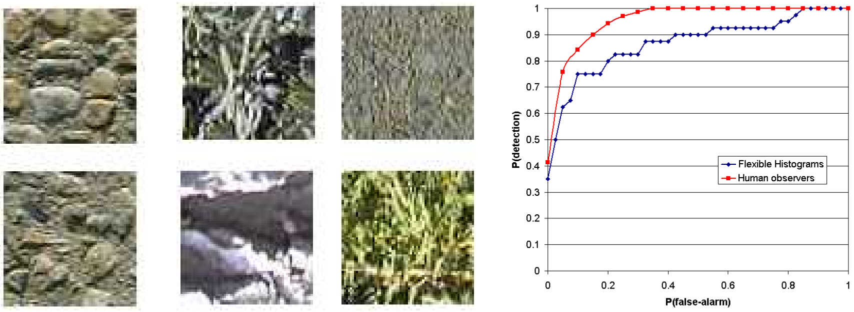
A binary decision is made by comparing the likelihood measure of each image to a threshold η_{model}. Using standard χ^2 tables a value for η_{model} for any desired level of certainty can be determined. However, we do not assume that the flexible histogram completely describes the model texture and 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 η_{model} is empirically we maximize the percentages of true-positives while guaranteeing an expected level of false-positives. The curve generated by varying η_{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



(a) (b) (c)

Figure 1: (a) Two images which contain the same texture. (b) Images which contain different textures. (c) ROC classification curves for human observers (top) and for the flexible histogram model (bottom).

bins, and to smooth the frequency counts in the bins from the other model images.

5 Experiments

5.1 Standardized tests

On "easy" data sets, such as the the MeasTex `Brodatz` texture test suite, performance is slightly higher than other techniques, our approach achieved 100% correct classification compared to 97% achieved by a gaussian MRF approach (Chellappa and Chatterjee, 1985). On the MeasTex `lattice` test suite, our approach achieved 98% while the best alternate method, in this case Gabor Convolution Energy method (Fogel and Sagi, 1989) achieved 89% (gaussian MRF's achieve only 79%.)

5.2 Natural textures

To measure performance on far more difficult textures, we collected a set of 20 types of natural texture and compared the classification power of this model to that of human observers (humans discriminate textures extremely accurately.) Three examples of each texture were acquired by extraction of image patches from a single image of a roughly texturally homogeneous natural scene.

An example of two images of the same texture are shown in Figure 1 (a), and of several textures are shown in (b). In this dataset, some images in the same class look very different, and some in different classes look similar.

To get a notion of the level of performance which is feasible we measured the receiver operator characteristics achieved by human observers. 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`

Using one of the three examples of a given texture as a model, we measured the ROC curve for discrimination among a set made up of the remaining 40 images. the top curve in The flexible histogram achieved the lower ROC curve and the averaged results for 7 observers generated the upper curve in Figure 1. The maximum likelihood decision rule yields an average of 75% percent accuracy for the flexible histogram model, while human observers achieve about 83% (while chance = 5%.)

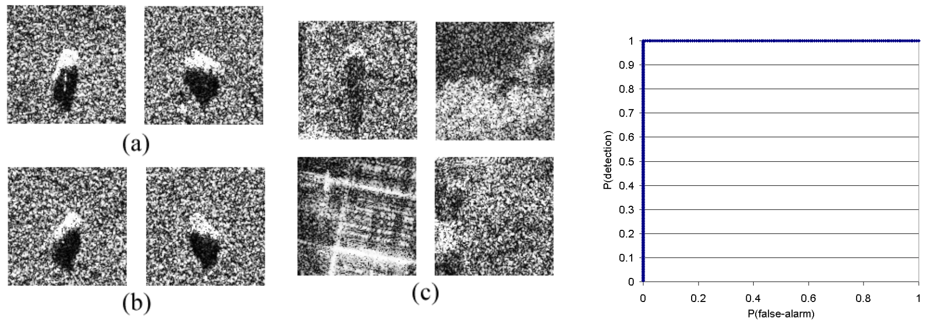


Figure 2: LEFT: 128×128 SAR images of (a) T72 tanks, (b) BMP2 personnel carriers, and (c) clutter images which contain no vehicles. RIGHT: ROC curve for discriminating full resolution (128×128) T72, or BMP2 vehicles from clutter images.

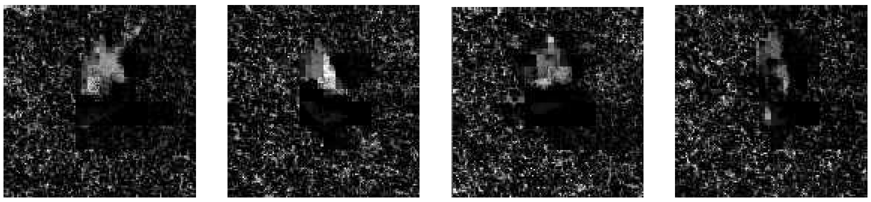


Figure 3: 2D flexible histograms for an image containing a target with respect to each of the four model images.

5.3 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 (Model Based Vision Lab,). In each class there were 140 images; two examples of each are shown in Figure 2. The target vehicle was a T72 tank (a), distractor images consisted of images of a BMP2 personnel carrier (b), and clutter images (c).

Using this data set we ask two questions: how well can we discriminate between images containing clutter and those containing a vehicle; and how well can we discriminate between vehicle types, given only models of one, or both of them.

The ROC plot on the right of Figure 2 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 real applications the the maximum acceptable P(false-alarm), (Neyman-Pearson criterion) is below the $\pm \sim 1\%$ precision of these experiments. In Figure 3 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,

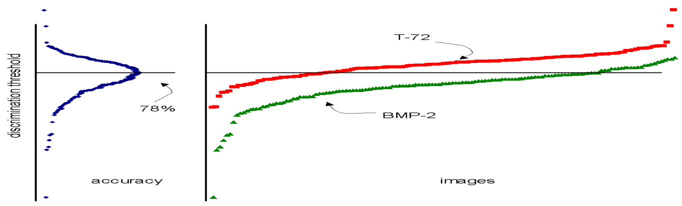


Figure 4: By searching over $\eta_{classify}$ the maximum accuracy achievable by the system can be found.

defined by the parent vectors 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 χ^2 calculation.

5.4 SAR vehicle classification system

We now turn to the question of discriminating between the T72 and BMP2 vehicles. Using just examples of one vehicle type, only 67% are correctly classified. By combining the information provided by both pairs of model images performance can be improved.

Given some image, test whether 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 from comparison of flexible histograms based on each model type, we can obtain a new measure: $L_{classify} = \chi^2_{model_1} - \chi^2_{model_2}$ To remove biases we compare $L_{classify}$ to a threshold $\eta_{classify}$ using a binary decision rule:

$$L_{classify} \underset{\text{class 2}}{\overset{\text{class 1}}{\gtrless}} \eta_{classify} \tag{4}$$

The accuracy of the classification system is given by: $P\left(H = h_1 | I \in \mathcal{I}_1, \eta_{classify}\right) \times P\left(H = h_2 | I \in \mathcal{I}_2, \eta_{classify}\right)$ where $P(H = h_k | I \in \mathcal{I}_k, \eta_{classify})$ is the probability that the classification hypothesizes an image of type k given that the image is really in the class $\mathcal{I}_k$, 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 4 is the accuracy achieved by the system as a function of the threshold. In the right, the output of the classification system is shown for the two types of images. The top curve are the sorted responses to the T72, and the bottom to the BMP2[3] 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.

6 Discussion

Because flexible histogram bins are determined directly from the model images, as difference measures increase, the differentiation between similarity measures de-

[3]When performing classification, of course, the system does not have access to these labels, as finding them is the objective!

creases. 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.

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. SAR classification rates shown here are only preliminary, and further analysis, including experiments on larger data sets, and on data which includes images with targets and clutter in close proximity, are required to further refine the system and obtain a better estimate of its overall potential.

References

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

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

Chellappa, R. and Chatterjee, S. (1985). Classification of textures using gaussian markov random fields. In *Proceedings of the International Joint Conference on Acoustics, Speech and Signal Processing*, volume 33, pages 959–963.

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

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

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

Fogel, I. and Sagi, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–113.

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

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

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

Model Based Vision Lab. Mbvlab. URL: http://www.mbvlab.wpafb.af.mil.htm.

QBIC. The ibm qbic project. http://wwwqbic.almaden.ibm.com/.

Virage. The virage project. URL: http://www.virage.com.

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