M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 292
Appears in the proceedings of: IEEE Conference on Pattern Recognition, Jerusalem, Oct. 1994

Texture Orientation for Sorting Photos "at a Glance"

Monika M. Gorkani

Machine Vision Group
IBM Almaden Research Center, K54
650 Harry Rd; San Jose, CA 95120
gorkani@almaden.ibm.com

Rosalind W. Picard

Vision and Modeling Group
MIT Media Laboratory
20 Ames St; Cambridge, MA 02139
picard@media.mit.edu

Abstract

We investigate a measure of "dominant perceived orientation" that has recently been developed to match the output of a human study involving 40 subjects. The results of this measure are compared with humans analyzing seven "teaser" images to test its effectiveness for finding perceptually dominant orientations. The use of low-level orientation is then applied to a "quick search" problem important in image database applications. Since both pigeons and humans are able to perform coarse classification of certain kinds of scenes, e.g., city from country, without taking time or brainpower to solve the image understanding problem, we conjecture that the collective behavior of low-level textural features such as orientation may be doing most of the work. We demonstrate a simple test of global multiscale orientation for quickly searching a database of vacation photos for likely "city/suburb" shots. The orientation features achieve agreement with human classification in 91 out of 98 of the scenes.

1 Introduction

The fact that orientation is an important feature for texture recognition and discrimination [1] has been recognized for some time, especially after the physiological experiments performed by Hubel and Wiesel [2] suggested the existence of orientation selective mechanisms in the human visual system. Oriented filters are now in use for a variety of texture problems such as multiscale texture analysis [3] [4] and analysis of flow textures [5] [6].

However, the use so far has been restricted to relatively "low-level" and pixel-level applications. In this paper we consider using "vision texture" or the global texture properties of the image as a quick way to make a first pass at higher-level problems, such as annotating or retrieving a particular set of your digitized vacation photos. Studies with pigeons [7] support the hypothesis for a mechanism that looks at collective low-level features for making comparatively high-level quick classifications. In this study, the textural features are dominant orientations, measured by an algorithm developed to approximate human perception of dominant orientation.

A number of researchers have proposed computational schemes for measuring local orientation at each pixel position using directional filters in the spatial domain [6] [5] [8] [9] and in the Fourier domain [10] [11]. However, to find global dominant orientation information in textures, further decision making is needed. For example, in Figure 1, an algorithm finding dominant global orientations would use the local estimated orientations of all the pixels to decide that Figure 1(a) has an overall dominant vertical orientation and Figure 1(b) has overall dominant orientations in the vertical and horizontal directions.

(a) (b)

Figure 1: Two textures quickly distinguished by their global orientation (from E. H. Adelson).

In [12] Picard and Gorkani introduced an algorithm to extract dominant orientation information from a texture in a way that closely approximated results from a large human visual study. The orientation finding algorithm was able to find at least one dominant orientation chosen by the human subjects in 95 of the 111 test images from the Brodatz Album [13]. Except for some small modifications, this is the algorithm used in this paper. After a brief overview of the algorithm in the next section, we apply it to seven "teaser" images which were designed to test the limitations of the method regarding filter size and some "higher-level" human visual processing. A human study is also run on these images to compare the computer and human orientation decisions. Finally we show an application to natural scenes, where the dominant textural orientations are used to quickly index through 98 digitized photos[1]. The algorithm picks out those scenes corresponding to a city or a suburb which have strongly oriented man-made structures such as buildings, cars and sign posts. The results of the algorithm were found to agree with human classifications in 91 of the 98 images.

This work was done while author Gorkani was at the MIT Media Lab and at INRIA Rocquencourt; it was supported in part by BT, PLC.

[1] Images available by ftp from the authors.

2 Background: finding "perceptual" orientations

The orientation-finding algorithm we use is similar to those of Kass and Witkin [5], Rao and Schunck [6] and Bigün and Granlund [8] in that it estimates the local orientation and its strength at each pixel of the image using a combination of the magnitudes of the outputs of a set of directional filters convolved with the image in the spatial domain. Unlike the works mentioned above, the implementation used here and in [12] extracts orientation information over multiple scales using a steerable pyramid [14], then combines the orientations from different scales and decides which are dominant perceptually, as determined by a human study.

The bottom level of the steerable pyramid (level0) is the original image, and each higher level is obtained by filtering and subsampling the previous level. At each level, a set of directional filters are used to estimate orientations. In the two studies presented here, we used four and three levels of the pyramid respectively (four levels were used in [12]). The number of pyramid levels is set to be the largest it can given the size of the image or subregion for which the orientation is being computed.

In [12], to find the dominant global orientations, we first accumulated the calculated orientation and its strength at each pixel position into a "strength histogram," H_s:

$$H_s(k) = \frac{N_\theta(k)}{\sum_{i=0}^{b-1} N_\theta(i)} \quad , \quad k = 0, 1, 2, \ldots, b-1 \tag{1}$$

where $N_\theta(k)$ is the sum of the strengths associated with pixel positions having an angle within the interval: $-90° + \frac{k180°}{b} \leq \theta \leq -90° + \frac{(k+1)180°}{b}$, and $b = 158$ is the number of bins in $180°$.

The algorithm in this paper uses a slightly different histogram. The $H_s(k)$ histogram was known to exaggerate the orientation in cases such as the Brodatz texture $D44$ shown in Figure 2, a blank image with high-contrast orientation in only a small area. If most of an image region has no orientations except in a small area, the normalization in (1) causes the peaks associated with the orientations in the area to be weighted highly. In the Brodatz study, use of $H_s(k)$ gave very good results. However in natural scenes, where a small area of orientation tends not to be perceptually salient, the results were better if we used a histogram with a different weighting which we call a "number histogram," H_n:

$$H_n(k) = \frac{N_{n\theta}(k)}{N_t} \quad , \quad k = 0, 1, 2, \ldots, b-1 \tag{2}$$

where $N_{n\theta}(k)$ is the sum of the number of pixels in the image associated with angles with strengths bigger than S within the same intervals as for $H_s(k)$, and N_t is the total number of pixels in the image.

Since the orientation at a pixel is calculated by taking a ratio of the magnitudes of the filter outputs, in the case of a blank image, the magnitudes will be zero and the ratio will be undefined. For H_s, the normalization is similarly undefined. This case never occurred for the Brodatz textures, but may occur in natural scenes. This problem is alleviated by use of H_n which only includes orientations having strength $> S$.

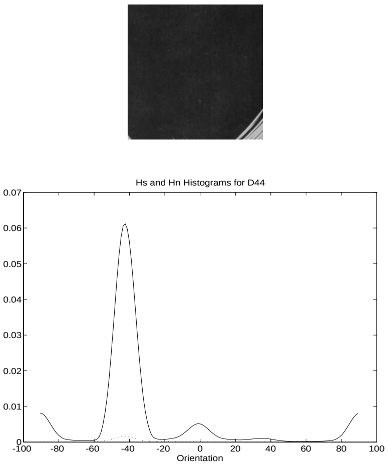

Figure 2: $D44$ with its prominent H_s and less prominent H_n histograms.

To determine S, the peaks of H_n were analyzed for several images with no oriented structures. It was found that choosing $S = 5$ resulted in H_n not being influenced by the undefined orientation problem described above. The division by the total number of pixels in the image ensures that if there is only a small area of oriented structures then it will be weighted less than a big area of oriented structures. The difference can be seen in the case of Figure 2 where the peak is much less prominent in H_n, represented by the dotted line, than in H_s, represented by the solid line.

To find the dominant global orientations for each image, the peaks of the orientation histograms for all the scales of the pyramid have to be analyzed to determine which of them, if any, correspond to a dominant orientation. This is done using a measure of "salience" of a peak, dependent on the height of the peak, its steepness and its width [12]. These salience measures are then thresholded (different thresholds for the salience measure are used for each level of the pyramid) to obtain decisions about orientations at each scale, and combined over scale. The thresholds used in [12] were found by iterative adjustment until they gave results which closely matched those found from the study of forty human subjects. The same thresholds were used in the first study below, and were used as a starting point for the second study, with only very slight changes needed to optimize their performance.

3 Study: Teaser images

We designed a set of 256×256 "teaser" images to investigate some of the limitations of filter size and "higher-level" human visual processing on the orientation-finding algorithm (same as [12] except that the H_n histogram described above was used and the contrast normalization step was omitted due to the fact that it is very computationally expensive). A human study involving 39 subjects was carried out on this data using the same conditions as [12]. None of the subjects were researchers in computer vision or pattern recognition.

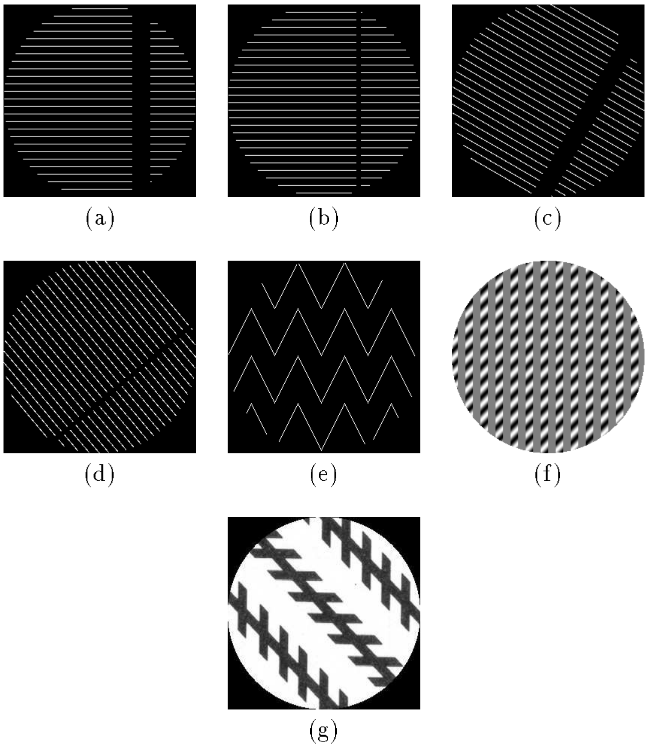

Figure 3: "Teaser" images shown to subjects.

The images used in this study are shown in Figure 3(a)-(g). Figure 3(a) shows horizontal lines with a vertical break between them. It was interesting for us to see whether the human subjects would consider the vertical break important or not. Results showed that 25 out of the 39 subjects who viewed this image considered the vertical orientation to be a dominant one. The computer did find both the vertical and horizontal orientations and gave the vertical orientation a smaller salience measure. The overall strength given to the vertical orientation by humans was also smaller than the horizontal orientation.

Figure 3(b) is the same as Figure 3(a) except that the distance between the breaks is closer. The goal here was to see whether the width of the break changes the strength of the perception of the vertically oriented line, and whether the current sized filters would pick this break at such a small width. Indeed the computer picked out the vertical orientation although the salience measure is slightly less than the vertical orientation in Figure 3(a). Only 15 of the 39 people who analyzed this image found the vertical orientation to be dominant. This example suggests that the width of the break effects the perceived strength of the vertical orientation.

Figures 3(c) and 3(d) are similar to Figures 3(a) and 3(b) except that the lines are no longer horizontal and the break is no longer vertical. In Figure 3(c), all the people did detect the orientation of the break. The computer also detected the orientation of the break.

Unlike Figure 3(b) where the smaller width of the break made it less perceptually dominant, the oriented break in 3(d) is perceived by all the subjects. The computer also chooses it as a dominant orientation. Similar to the human results, the strength of the orientation of the break found by the computer is smaller than the strength of the orientation of the continuous lines.

In Figure 3(e) it was interesting to see whether the subjects and the algorithm would pick up the horizontal orientation despite the lack of any explicitly horizontal lines. Indeed, 28 out of 39 subjects chose the horizontal orientation and gave it the highest strength. The computer only found the orientation of the line segments connected together. Similarly, eight of the 39 subjects picked the vertical orientation to be dominant. We count this inability of the algorithm to "group" the lines into their horizontal direction as the first case of where it fails. We discussed fixes to this failure mode in [12]; all of the fixes involve more processing which begins to slow down the "quick glance" approach.

Figure 3(f) illustrates orientations at two different scales. Both the computer and the human subjects chose the orientation of the diagonals at 43°. The humans (by a narrow 24/39 subjects) also picked the vertical orientation to be dominant. The computer did not pick this orientation; instead it found the orientation 22° which the majority of humans did not pick. Also, 4 people chose the horizontal orientation to be dominant which the computer did not find at any of the pyramid levels. The computer did not detect the vertical orientation in any of the pyramid levels. We count this lack of detecting the vertical orientation as the second case where the algorithm failed.

Figure 3 (g) is the famous Zölliner illusion [15]. The computer found the diagonal lines to be parallel. The human data indicates that the subjects also perceived the diagonal lines to be parallel. One possible reason for the illusion not working could be due to the experimental design [12] which had the humans spin a bar on top of the test image. Also since the human data is quantized (same method as [12]), small differences between the orientations chosen are sometimes lost.

At first glance, the "teaser" images may seem trivial. However, the fact that the subjects' responses were not always in agreement indicates that not all the orientations were perceptually obvious and may have required a "high-level" interpretation. The computer did detect 16 out of the 18 dominant orientations perceived by the majority of the subjects even though the filters were of fixed size and only applied at four different scales.

4 Study: Natural Scenes

One big problem in designing an image database system involves teaching the computer to recognize different scenes for annotation and fast retrieval. How can the computer recognize whether an image is of a "country" scene with predominantly natural scenery like trees, grass, mountains, etc. or a "city/suburb" scene with predominantly buildings and other man-made structures found in a city or suburb like cars, roads and sign posts? Obviously this problem can not be completely solved by "low-level" vision. Humans can easily distinguish different scenes using both "low-level" vision and "high-level" knowledge based on past experience and learning. However, an interesting question still can be posed: *How much* of a high-level task can be achieved by using only a collection of low-level features? In this second study, we demonstrate how textural orientation information can be used for this problem.

4.1 Recognizing "city/suburb" scenes

Recognizing "city/suburb" scenes provides an application where textural orientation information can be very useful: many man-made structures found in the city or a suburb such as buildings, cars, sign posts, etc. can be viewed to have a global "textural" appearance with specific dominant orientations. Assuming an upright camera position, which would be the case for a lot of pictures taken, the majority of these dominant orientations are vertical and horizontal. For example, street lights and sign posts in a picture have vertical orientations. A picture taken from a frontal view of a building would have dominant vertical and horizontal orientations. Of course, a picture taken from a building from an angle will have dominant orientations other than the horizontal and vertical. Unless the perspective is severe, the vertical orientation of the buildings changes very little but the horizontal direction can be skewed to other angles usually at most $\pm 45°$ from the horizontal.

A scene is then more likely to be in a city or a suburb, if it has a lot of man-made structures with either a strong vertical orientation or both a vertical orientation and a horizontal orientation which may be skewed (as in the case of a building with possible perspective). Therefore, it is possible to design a classification scheme to search for likely "city/suburb" scenes in a database using the textural orientation information. We designed such a scheme. We apply the orientation-finding algorithm on regions in an image, to determine each region's dominant orientations. Then we label an image as "city/suburb" if either or both of the following conditions are satisfied:

1. More than R regions have only a strong dominant vertical orientation $85° \leq |\theta| \leq 90°$ where $|\ |$ is the absolute value operator.
2. More than R regions have both a dominant vertical orientation and a dominant orientation from $-45° \leq \theta \leq 45°$ (to take care of the perspective problem).

As can be seen, this is an exceedingly simple method. Of course, there will be images of scenes other than "city/suburb" with dominant orientations found by this algorithm. However, in the case of an image database, this method provides a scheme to quickly index through thousands of images to find those which are more likely to be scenes of the desired category. More specific algorithms can then be applied to fully understand the content of each image. The use of texture-like orientation information should reduce the search space, saving time overall. The focus here is on behavior like a human "glancing quickly" through a pile of pictures.

4.2 Image test data

To test the "city/suburb" recognizer, we used a set of 98 512×512 24-bit RGB digitized photos given to us by BT, PLC. These images are of various scene types and were not shot with this classification in mind. The images were labeled by us so that *imgN* corresponds to the Nth image. Eleven of these images are shown in Figure 5.

Three people viewed all the 98 images and labeled those which were "city/suburb" scenes. In future environments, it is typical that each person will annotate their own photos and annotations may differ; this makes developing "ground truth" extremely difficult for the image database query problem. In essence, one person's annotation is sufficient for testing algorithms in database query. By using three people, we essentially ran three different trials for extra robustness. There were 35 images out of the 98 which were labeled by at least two people to be "city/suburb" scenes. There was an ambiguous set of three images only labeled by one out of the three people to be a "city/suburb" scene. For this study we only considered the 35 images judged by at least two out of three people to be "city/suburb" scenes. For determining the dominant orientations, we used the NTSC "Y" component of the images [16]. No other preprocessing was done on the images, nor were any images omitted from the original set we were given.

4.3 Finding orientation in image regions

For natural scenes, we applied the orientation-finding algorithm of [12] except for the modifications described earlier, to regions in the images. There are a huge number of ways to divide the images into regions. Determining the best size and shape for the regions is a very difficult problem since it depends on the scale and content of the image which can vary tremendously through a database. For this experiment, we chose a simple solution of dividing each 512×512 image into square regions of 128×128 resulting in 16 regions.

If a region has only a vertical orientation and an orientation from $-45° \leq \theta \leq 45°$ satisfying the salience measure thresholds shown in Table 1 (these thresholds are slightly different from [12] since only three levels of the pyramid were used), it is considered for the calculation of R. For regions having only a dominant vertical orientation, we use a higher salience measure threshold to ensure that only those with very strong vertical orientations are considered. For this study, the salience measure threshold was the same for all pyramid levels, 0.18.

For each of the 98 images, the orientation-finding algorithm was applied over all 16 regions. To choose the value of R, we calculated a simple histogram shown in Figure 4 where the x-axis corresponds to the number of blocks having the required dominant orientations. The dotted line indicates how many of the 35 "city/suburb" scenes have blocks with the required orientations. The continuous line indicates how many of the scenes other than "city/suburb" have blocks with the required orientations. Different criteria can be considered for choosing R. It is important to find a value for R which minimizes the total classification error. However, since there are only 35 "city/suburb" scenes, it is also important that the majority of these images are classified correctly. Therefore, we searched for a value of R which reduces the total classification error and results in more than 90% of the 35 "city/suburb" scenes to be classified correctly. As can be seen in Figure 4, $R = 3$ satisfies the above criterion resulting in 34/35 "city/suburb" scenes to be classified correctly and 6/63 other scenes to be misclassified as "city/suburb". This choice of R resulted only in a total of 7 misclassified scenes.

Threshold	Values
γ_2	0.3
γ_1	0.04
γ_0	0.014

Table 1: Threshold values chosen for different pyramid levels 0-2. Notice they are biggest at the coarsest (top) level.

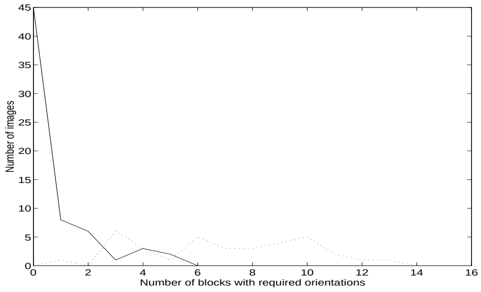

Figure 4: Histogram for determining the value of R

4.4 Misclassified cases

The seven misclassified images along with some correctly classified images are shown in Figure 5. The misclassified images are *img*008, *img*036, *img*047, *img*029, *img*089, *img*056 and *img*067.

In Figure 5, *img*021 was correctly classified by the computer to be a "city/suburb" scene. The *img*047 was not found by the computer to be a "city/suburb" scene although it was labeled by the three humans to be one. One problem is that the details of the buildings in this image compared to *img*021 are not very sharp. So, even though the computer did detect the required orientations in these regions, their salience measures were much smaller than the threshold requirements. It may be the case that the "thresholds" used by humans adapt to context such as a defocused background; this type of adaptivity has not been studied yet, but could be worked into this algorithm.

The *img*001 was correctly classified as a "city/suburb" scene and *img*044 and *img*064 were correctly not classified as "city/suburb" scenes. On the other hand, scenes *img*008, *img*036 (1/3 subjects chose as "city/suburb"), *img*056 (1/3 subjects chose as "city/suburb") and *img*067 were incorrectly classified as "city/suburb". In *img*008, there are a lot of structural objects like the house and the fence which cover more than 25% of the scene and have the dominant orientations searched by the computer. In *img*036, there are houses and cars in the background and the vertical goal post. In both *img*056 and *img*067, there are orientations searched by the algorithm like the vertical structure behind the man and the structure behind the girl and the vertical pole beside her (the strong horizontal lines on her shirt were not considered since there was no vertical orientation in those regions). The human annotators for most cases probably considered only those images with the main subjects directly associated with a city or a suburb as seen in *img*001 where there are a building and a car. The term "city/suburb" can be ambiguous for the misclassified cases mentioned above. It would not be wrong to annotate these images as "city/suburb"; however, for more complete descriptions, other categories are needed.

Figure 5: Example photos. The seven misclassified cases are shown at right.

There will always be scenes like *img*029 and *img*089 which have the required orientations searched by the computer like the vertical orientation of the fence and the trees but which are not considered as "city/suburb" scenes. At a quick glance, humans sometimes also pick these images; the study here is conservative in that it gave subjects ample time to classify, but gave the retrieval algorithm only "a glance." Nonetheless, using only the textural orientation information, the computer was able to reduce the search space by more than half from 98 scenes to 41 scenes, of which 34 corresponded correctly to the human annotation.

5 Summary

In this paper, we introduced the idea of using global textural features such as orientation for a quick way to make "high-level" scene classifications. We used a multi-scale method for finding dominant orientations and compared its output to that of 39 humans analyzing seven "teaser" images. Except for two cases, the computer detected the orientations found by the majority of the humans, thus offering more evidence that the multiscale orientation finding method of [12] can find "perceptually important" dominant orientations.

Second, we showed an application where the dominant textural orientations were used to quickly index through 98 images of natural scenes to find likely "city/suburb" scenes which have strongly oriented man-made structures such as buildings, cars and sign posts. Using only dominant orientation information, the computer was able to reduce the search space by more than half from 98 scenes to 41 scenes of which 34 corresponded correctly to human classification.

Textural orientation is not intended to solve the "high-level" problem; we expect the usage of this feature to be activated by a high-level request and eventually combined with other features. Although the choice of only orientation texture (vs. other combinations of textural features) and particular value of R will vary for other retrieval problems, the texture and pattern recognition methodology applied here is generalizable. Moreover, the success of simple orientation in agreeing with high-level human interpretation found on this large set of images can not be ignored.

There are many areas for future research – exploring more complex interactions between filter outputs to deal with "high-level" effects like grouping and problems with contrast, exploiting context and prior expectations to adapt thresholds for background blur, or attaching categorical descriptions other than "city/suburb" to a combination of textural and other low-level features like color. These features can be pre-computed beforehand and used to further improve the "quick-glance" method in real-time image database retrieval environments like Photobook [17].

Another direct application for using dominant textural orientation is that it can be used to find areas in the image with certain types of directional structure for speeding up object and motion recognition. The dominant global orientation information also gives a quick way to tell if two textures are similar before applying a more computationally expensive or attentive image understanding method.

6 Acknowledgments

We would like to express our gratitude to O. Yip and V. Balard for their help with the human studies and to W. T. Freeman for the steerable pyramid code and many helpful insights.

7 References

[1] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE T. Sys., Man and Cyber.*, SMC-8(6):460–473, 1978.

[2] H. D. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Physiology*, 195:215–243, 1968.

[3] 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 Processing*, pages 253–271, Cambridge, MA, 1991. MIT Press.

[4] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition Journal*, 24:1167–1186, 1991.

[5] M. Kass and A. Witkin. Analyzing oriented patterns. In M. A. Fischler and M. Kaufman, editors, *Readings in Computer Vision*, pages 268–276. M. Kaufman, 1987.

[6] R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP Graphical Models and Image Processing*, 53(2):157–185, 1991.

[7] R. J. Herrnstein, D. H. Loveland, and C. Cable. Natural concepts in pigeons. *J. of Exp. Psych: Anim. Beh. Procs.*, 2:285–302, 1976.

[8] J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *Proc. 1st Int. Conf. Comp. Vis.*, pages 433–438, London, England, 1987.

[9] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE T. Patt. Analy. and Mach. Intell.*, PAMI-13(9):891–906, Sept. 1991.

[10] R. Bajcsy. Computer description of textured surfaces. *Int. Joint. Conf. Artificial Intelligence*, pages 572–578, 1973.

[11] S. Chaudhuri, H. Nguyen, R. M. Rangayyan, S. Walsh, and C. B. Frank. A Fourier domain directional filtering method for analysis of collagen alignment in ligaments. *IEEE Transactions on Biomedical Engineering*, 34(7):509–517, 1987.

[12] R. W. Picard and M. Gorkani. Finding perceptually dominant orientations in natural textures. *Spatial Vision, Spec. Julesz Issue.* To Appear; also avail. as Percep. Comp. TR #229, M.I.T. Media Lab, 1993.

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

[14] W. T. Freeman. *Steerable Filters and Local Analysis of Image Structure.* PhD thesis, Media Arts and Sciences, MIT, 1992.

[15] I. Rock. *The Perceptual World.* W. H. Freeman and Company, New York, NY, 1990.

[16] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[17] R. W. Picard and T. Kabir. Finding similar patterns in large image databases. In *Proc. ICASSP*, pages V–161–V–164, Minneapolis, MN, 1993.