Locating Deciduous Trees

Niels Haering
Zarina Myles
Niels da Vitoria Lobo

Department of Computer Science
University of Central Florida
Orlando, Florida 32816

Abstract

In this paper, we present a method to obtain information about the presence of deciduous trees in images. Since a single measure, observation or model is unlikely to yield robust recognition of trees, we present an approach that combines color measures, and estimates of the complexity, structure, roughness and directionality of the image based on Entropy Measures, Greylevel Co-occurrence Matrices, Fourier Transforms, Multi-resolution Gabor Filter sets, Steerable Filters and the Fractal Dimension. A standard back-propagation neural network is used to arbitrate between the different measures and to find a set of robust and mutually consistent "tree experts".

1 Introduction

Abstract descriptions of the image content, such as the location and identity of objects in a scene are essential for efficient image retrieval. Recent work in this area includes MIT's Photobook project [17], ETL's Trademark and Art applications [12] and IBM's QBIC project [6]. Often the tasks these systems have to perform range widely. For example, they may perform face recognition given a person's mug shot and/or image retrieval given a simple sketch. In the general framework we are interested in, it is likely that a single indexing method will not suffice. The features that need to be extracted from images depend on the task the system is asked to perform. Unfortunately, these features cannot be extracted on-line especially if we are trying to manage large databases. Therefore, it is necessary to preprocess images and to annotate them with semantics indicating their content.

We present an approach that can be used to locate deciduous trees in images. In this context, we would like to argue that a single measure, observation or model is unlikely to yield robust recognition of trees. Since the appearance and shape of trees, even if unoccluded and from one deciduous species, vary a great deal, no one feature, such as texture, color, or inferred shape, can classify them correctly. In this paper, we try to show the robustness of a combination of simple filter methods for the classification of tree and non-tree areas in images. The analysis and combination of the responses of these methods is complex and probably requires non-linear arbitration. Therefore, a standard back-propagation network is being used to learn the arbitration [5]. The classification is based on the weights and correlations found by the network during the training phase where images are presented in feature space together with their tree/non-tree segmentation. Note that no attempt is made at recognizing the various species and subspecies of deciduous trees. Hence locating deciduous trees is certainly a simpler problem than tree species classification. However, it constitutes a valuable process that may simplify such classification and help build a content index for image databases. In addition, the method proposed here is valuable for outdoor navigating robotic devices, that need to recognize tree and non-tree areas in order to flexibly react and interact with their environments. Section 2 discusses the measures used for our method. In Section 3, we report how these measures are combined for classification. Section 4 shows and analyses the results before Section 5 closes with a discussion of open issues and a reflection on the success of the presented method.

2 The Measures

We classify trees using features extracted from color images. Some of these features are representations of individual pixels in the image, such as the hue, saturation and value representation. Others are the outputs of image patch based filters, such as the features obtained based on the Fourier Transform, multi-resolution Gabor filters, the oriented energy of bar and step edges, Greylevel Co-occurrence Matrices (GLCM), the Fractal Dimension, and some entropy measures. Since we are aiming at locating tree areas in images, the set of features is primarily designed to
recover the level of the structure in an image and to a lesser degree its type. For instance, we are not concerned with the orientation of an image patch; in fact, for features that are inherently directional we often combine the responses at a number of orientations and use their range and average to make them (in-plane) rotation invariant (as done in [11]).

2.1 Fourier Transform Measures

Some measures commonly used with Fourier Based Methods are i) wedge sampling, ii) annular-ring sampling, and iii) parallel-slit sampling.

Many textures differ significantly in the domains of the annular-ring and parallel-slit measures. We found that for our purpose of discriminating tree and non-tree areas angular wedge sampling is most expressive. Fourier Transforms (FTs) of images and image patches containing human-made structures often have line or wedge shaped areas of high spectral power that pass through the center as shown in the image FT pair in Figure 1: An image containing human-made and tree areas (a) and its Fourier Transform (b). An image of leaves of a tree (c) and its Fourier Transform (d). The numbers associated with (b) and (d) are the structure measure (described in Section 2.1) for images (a) and (c).

Figures 1 (a) and (b). Summing the power in fixed angular intervals for all directions in the FT of the image lets us separate common from uncommon orientations in the image. The shaded wedge in Figure 2 shows such an angular interval. A circular mask has been imposed so that the power in the diagonal directions is not unfairly biased.

Figure 2: The sum of the power of the Fourier Transform inside the shaded vertical angular interval is a measure of the "structure" present in an image patch.

Once the power in each angular interval has been determined, we obtain the minimum and maximum angular power and use the normalized ratio \(\frac{\text{max}}{\text{min}}\) to determine the amount of structure in the patch.

Larger values for this wedge measure indicate greater "regularity" in some direction in the image patch, smaller values indicate less "regularity", in terms of parallel lines, bars and edges. Since we are comparing the ratio between the maximum and minimum value, this measure is rotation invariant.

Performing the above procedure on fixed-size image patches, we obtain local measures of the regularity of these patches. We obtained very similar results for patches of size 16 \(\times\) 16, 32 \(\times\) 32, and 64 \(\times\) 64 pixels.

2.2 Gabor Filter Measures

The image (in the spatial domain) is described by its 2-D intensity function. The Fourier Transform of an image represents the same image in terms of the coefficients of sine and cosine basis functions at a range of frequencies and orientations. Similarly, the image can be expressed in terms of coefficients of other basis functions. Gabor [9] used a combined representation of space and frequency to express signals in terms of "Gabor" functions:

\[
F_{\theta, \nu}(x) = \sum_{i=1}^{n} a_i(x) g_i(\theta, \nu)
\]

(1)

where \(\theta\) represents the orientation and \(\nu\) the frequency of the complex Gabor function:

\[
g_i(\theta, \nu) = e^{i\nu(\nu\cos(\theta) + \gamma\sin(\theta))} e^{-\frac{x^2 + y^2}{2}}
\]

(2)

Gabor filters have gained popularity in multi-resolution image analysis, despite the fact that they do not form an orthogonal set, which means that their coefficients cannot be obtained by convolution of the image with the basis functions as is done in the next section, where we employ steerable bar and step edge detectors. Gabor filter based wavelets have recently been shown by Manjunath and Ma [15] to be fast and useful for the retrieval of image data.

We convolve each image patch with Gabor filters tuned to four different orientations at 3 different scales. Next, the average and range of the four measures at each scale are computed. Finally, to facilitate scale invariance introduced the following four texture measures:

- The average over all orientations and scales.
The average of each scale’s range of orientations.

The range of each scale’s orientation average.

The range of each scale’s range of orientations.

2.3 Steerable Bar and Step-Edge Filters

Since many human-made structures exhibit a large amount of regularity in the form of parallel lines and bars, patches with few dominant orientations are less likely to represent trees. On the other hand, the irregular leaf and branch structure of trees often exhibits a greater variety of orientations.

Binning orientations appropriately, we can use the number and strength of different orientations in an image patch to distinguish between patches belonging to human-made scenes (which usually have fewer but stronger distinct orientations) and natural scenes.

To obtain the dominant orientation for each image patch, steerable filters are used in a manner similar to [1, 8]. The result of this routine is an orientation image indicating the orientation of the predominant step or bar edge at each location.

Perona [19] demonstrated a general constructive method to construct basis and interpolating functions for steerable bar and step edge filters and showed that all functions that are polar-separable with sinusoidal $\theta$ components are steerable. Examples of such functions are shown in Figure 3.

![Figure 3: Examples of polar separable functions with sinusoidal $\theta$ component corresponding to $a_0, \ldots, a_7$.](image)

We used this method to obtain a steerable function set for a quadrature pair $(G_{yy}, H_{yy})$, where $G_{yy}$ is the second derivative along the y-axis of an elongated Gaussian kernel $G(x, y, \sigma_x, \sigma_y) = e^{-(x^2/\sigma_x^2 + y^2/\sigma_y^2)}$ shown in Figure 4 (a) and $H_{yy}$ is the Hilbert transform of $G_{yy}$ shown in Figure 4 (b).

For multiple occurrences of lines and step edges, good angular resolution (orientation selectivity) was obtained when the ratio $\sigma_x/\sigma_y$ was at least $\frac{1}{4}$. Perona [19] shows an efficient method that places the second derivative of the Gaussian in the real part of the complex kernel and its Hilbert transform in the imaginary part.

![Figure 4: Filters used to measure the energy in the image. The second derivative of an elongated Gaussian (left) is used to detect lines in an image. Its Hilbert transform (right) is used to detect step edges in an image.](image)

The n-term approximation of the function we want to steer can be written as:

$$ F_{\theta}^{[n]} = \sum_{i=1}^{n} \sigma_i a_i(x) b_i(\theta) $$

where the $\sigma_i$ weight the product of the $i^{th}$ filter basis function $a_i$ (the coefficients of the 2D Fourier series) and the corresponding interpolating function $b_i$ (note that the $b_i$ are the frequency basis functions of the Fourier Series).

The values for the $\sigma_i$, $a_i$ and $b_i$ are obtained by finding the Fourier series of the function $h(\theta)$, which is the integral of the product of the function with rotated versions of itself:

$$ h(\theta) = \int_{\mathbb{R}^2} F_{\theta}(x) \overline{F_{\theta=0}(x)} \, dx $$

where the integral ranges over all 2D space ($\mathbb{R}^2$) and $\overline{()}$ represents the complex conjugate. Note that $F_{\theta=0}(x) = F(x)$.

Expanding $h(\theta)$ as a Fourier series we can read off the filter’s (2D) basis functions $a_i$ and the corresponding interpolating functions $b_i$.

$$ \sigma_i = h(\nu_i) $$

$$ b_i(\theta) = e^{i\nu \theta} $$

$$ a_i(x) = \sigma_i^{-1} \int_{\mathbb{R}^2} F_{\theta}(x) e^{ik\theta} \, dx $$

The $\sigma_i$ terms are used only for error analysis. For details see [19].

These filters are used to obtain the oriented energy of both step as well as bar edges. Although we initially envisaged them to aid the recognition of deciduous trees in winter, when their leaves are missing, the orientation analysis also turned out to be useful for the detection of leaves and trees in summer.
2.4 Greylevel Co-occurrence Matrix Measures

Let \( p(i, j, d, \theta) = \frac{P(i, j, d, \theta)}{R(d, \theta)} \) where \( P(\cdot) \) is the greylevel co-occurrence matrix of pixels separated by distance \( d \) whose orientation is \( \theta \) and where \( R(\cdot) \) is a normalization constant that causes the entries of \( P(\cdot) \) to sum to one.

The following are some of the measures defined for texture classification, see [2, 11]. The complete set of the features used in our work is listed in [10]. The Angular Second Moment (E) (also called the Energy) assigns larger numbers to textures whose co-occurrence matrix is sparse:

\[
E(d, \theta) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} [p(i, j, d, \theta)]^2
\]

The Contrast (Con) is the moment of inertia around the co-occurrence matrix’s main diagonal. It is a measure of the spread of the matrix values and indicates whether pixels vary smoothly in their local neighborhood.

\[
\text{Con}(d, \theta) = \sum_{n=0}^{N_x-1} n^2 \left[ \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} p(i, j, d, \theta) \right]
\]

The Inverse Difference Moment (IDM) measures the local homogeneity of a texture. It weighs the contribution of the co-occurrence matrix entries inversely proportional to their distance to the main diagonal.

\[
\text{IDM}(d, \theta) = \sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} \frac{1}{1 - (i - j)^2} p(i, j, d, \theta)
\]

The Mean (M) is similar to the contrast measure above but weights the off-diagonal terms linearly with the distance from the main diagonal, rather than quadratically as for the Contrast.

\[
\text{M}(d, \theta) = \sum_{n=0}^{N_x-1} n \left[ \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} p(i, j, d, \theta) \right]
\]

Similar to the Angular Second Moment the Entropy (H) is large for textures that give rise to co-occurrence matrices whose sparse entries have strong support in the image. It is minimal for matrices whose entries are all equally large.

\[
H(d, \theta) = -\sum_{i=1}^{N_y} \sum_{j=1}^{N_x} p(i, j, d, \theta) \log(p(i, j, d, \theta))
\]

The Correlation (Cor) measure is an indication of the linearity of a texture. The degree to which rows and columns resemble each other strongly determines the value of this measure. This measure uses \( \mu_x = \sum_i \sum_j p(i, j, d, \theta) \) and \( \mu_y = \sum_j \sum_i p(i, j, d, \theta) \).

\[
\text{Cor}(d, \theta) = \frac{\sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} ijp(i, j, d, \theta) - \mu_x \mu_y}{\sigma^2}
\]

Note that the directionality of a texture can be measured by comparing the values obtained as \( \theta \) is changed. The above measures were computed at 4 angles \( (0^\circ, 45^\circ, 90^\circ, \text{and } 135^\circ) \) using \( d = 1 \). To make the measures rotation invariant, we use the average and range over the 4 orientations to obtain 2 features for each type of measure. For further discussion of these greylevel co-occurrence matrix measures, see [11, 2].

2.5 Fractal Dimension Measures

The underlying assumption for the use of the Fractal Dimension (FD) for texture classification and segmentation is that images or parts of images are self-similar at some scale.

Various methods that estimate the FD of an image have been suggested:

- Fourier-transform based methods [16],
- box-counting methods [13, 3], and
- 2D generalizations of Mandelbrot’s methods [18].

The principle of self-similarity may be stated as: If a bounded set \( A \) (object) is composed of \( N_r \) non-overlapping copies of a set similar to \( A \), but scaled down by a reduction factor \( r \), then \( A \) is self-similar.

From this definition, the Fractal Dimension \( D \) is given by

\[
D = \frac{\log N_r}{\log r}
\]

The FD can be approximated by estimating \( N_r \) for various values of \( r \) and then determining the slope of the least-squares linear fit of \( \log N_r \). The differential box-counting method outlined in [3] is used to achieve this task.

Three features are calculated based on

- the actual image patch \( I(i,j) \),
- the high-greylevel transform of the image patch,
  \[
  I_1(i,j) = \begin{cases} 
  I(i,j) - L_1 & I(i,j) > L_1 \\
  0 & \text{otherwise}
  \end{cases}
  \]
- the low-greylevel transform of the image patch,
  \[
  I_2(i,j) = \begin{cases} 
  255 - L_2 & I(i,j) > 255 - L_2 \\
  I(i,j) & \text{otherwise}
  \end{cases}
  \]
where \( L_1 = g_{\min} + \frac{g_{\avg}}{2} \), \( L_2 = g_{\max} - \frac{g_{\avg}}{2} \), and \( g_{\min} \), \( g_{\max} \), and \( g_{\avg} \) are the minimum, maximum and average greyvalues in the image patch, respectively.

The fourth feature is based on multi-fractals which are used for self-similar distributions exhibiting non-isotropic and inhomogeneous scaling properties. Let \( k \) and \( l \) be the minimum and maximum greylevel in an image patch centered at position \((i,j)\), let \( n_r(i,j) = l - k + 1 \), and let \( N_r = \frac{n_r}{N} \). Then the multi-fractal, \( D_2 \) is defined by

\[
D_2 = \lim_{r \to 0} \frac{\log \sum_{i,j} N_r^2}{\log r}
\]

A number of different values for \( r \) \((0 \ldots 1)\) are used and the linear regression of \( \frac{\log \sum_{i,j} N_r^2}{\log r} \) yields an estimate of \( D_2 \).

2.6 Hue, Saturation and Value Measures

While the intensities of the red, green and blue components of a color image are highly correlated, the hue, saturation, and value decomposition offers a more independent representation that captures complementary information of the image.

The Hue component \((\theta)\) can be computed by finding the angle between the color of a pixel with respect to the red corner of the color triangle in Figure 5(a) (see for example [14] for details).

\[
\theta = \cos^{-1} \left( \frac{2r - g - b}{2\sqrt{(r-g)^2 + (r-b)(g-b)}} \right)
\]

where \( r \), \( g \), and \( b \) are the intensity of the red, green and blue components of the corresponding pixel.

![Color Triangle](image)

Figure 5: The color triangle and the function mapping a pixel's hue to its probability of being a leaf pixel.

The saturation \((S)\) and value \((V)\) components are also defined in terms of \( r \), \( g \), and \( b \):

\[
S = 1 - \frac{3}{r + g + b} \min(r, g, b)
\]

\[
V = \frac{1}{3} (r + g + b)
\]

The color-value discontinuity between magenta-red and orange-red (which have maximally different feature values on the hue scale but appear very similar in images) complicates the task unnecessarily. Therefore each hue value is assigned the corresponding value of a function that transforms the hue into the likelihood of the pixel being a leaf pixel (see Figure 5(b)).

2.7 Entropy Measures

Since leaves and branches appear as rough and "messy" areas at most scales at which trees can be identified, we can use the entropy of image patches to separate them from uniform, smooth, and smoothly varying object surfaces. If \( V_{\max} \) is the maximum value in an image patch, the entropy is defined as

\[
\text{Entropy} = -\sum_{i=0}^{V_{\max}} h_i \log(h_i)
\]

where \( h_i = \frac{n_i}{N} \) is the \( i \)th histogram count \( n_i \) divided by the total number of pixels in the image patch \((N)\). We measure the entropy in both the gray value image as well as the orientation image described above, both measures are largely rotation invariant.

3 Combining the Measures

Combining all the features above we represent an image location with 43 feature values. Since the analysis of their usefulness and contribution is complex, we used a standard back-propagation algorithm with randomly selected initial weights, and the sigmoidal activation function \( \Phi(\text{act}) = \frac{1}{1+e^{-\text{act}}} + 0.5 \), where \( \text{act} \) is the activation of the unit before the activation function is applied. One factor complicating the task for the network is that often there is no correct classification. For instance, should bushes be labeled as tree or non-tree areas? What if a bush is actually a small tree? A second complicating factor is that we did not take excessive care labeling our training data so that at border regions pixels of opposite classes are often labeled the same. Therefore, we should keep in mind that despite the statistical nature of neural networks, which makes them robust against a few outliers and mislabeled training patterns, we can not expect 100 percent correct classification on the training data. In fact, there is a greater problem associated with overtraining the network so that it learns the training set too well while generalizing poorly for new images. A third problem arises from the use of patch based measures that often assigns in-between values to pixels on class boundaries.

3.1 The Neural Network Architecture

To facilitate non-linear decisions the network needs to have at least one hidden layer. On the other hand,
too many hidden layers will slow convergence since more units need to be updated following the presentation of each training pattern. Cybenko [4] has shown that a single hidden layer is sufficient to uniformly approximate any continuous function (to arbitrary accuracy). Although this existential proof doesn't state that the best network for any task has only a single hidden layer, we use only one hidden layer for our network. The architecture of the network is shown in Figure 6. The back-propagation algorithm propagates the (input) function values layer by layer, left to right (input to output) and back-propagates the errors layer by layer, right to left (output to input) as shown in Figure 6.

![Network Architecture Diagram](image)

**Figure 6:** The network architecture.

Although all the computed features are bounded, some have much larger means than others, (the means vary by more than 10 orders of magnitude!). To prevent some features from overriding the contributions of others all measures are normalized by subtracting their mean (over all training images) before dividing their values by their range and "squashing" them with a tanh function to make sure that all unseen features also fall within this range.

Training causes the network to come up with "tree experts" which together can classify most (above 95 percent) of the pixels in the training images correctly. As stated above, over-training the network will achieve better performance on the training images, but often results in poorer performance on test images.

4 Results

The first columns of Figures 7 and 8 show previously unseen images, while the second columns show the corresponding classification results. Brighter regions represent areas that are likely to belong to deciduous trees and darker regions represent areas that are less likely to belong to deciduous trees. A larger set of the color images together with their classification re-

![New Images and Classification Results](image)

**Figure 7:** New images and the classification results.
sults can be seen at http://longwood.cs.ucf.edu/haer-
ing/ir/trees.html.

Measures at every third pixel location of these images were obtained (just under half a million data points) and combined with labeled images to train the network. Subsampling speeds up the training process without (noticeably) affecting its outcome, since neighboring pixel locations are highly correlated. Linear and non-linear optimizers like the back-propagation network are usually better at interpolating between known data points than they are at extrapolating. For the network to be able to recognize deciduous trees of different species at a wide range of scales, view points, lighting conditions and during spring, summer and fall, it is important that the training images reflect this variety.

We would like to point out that some of the test images show trees in fall with the leaves’ colors ranging from green, through yellow, orange and red to a magenta-ish red. Although color is often a useful cue, the network has also learned to recognize trees that have unusually colored leaves.

The second image in Figure 8 shows the performance of the approach for an image taken on a foggy day, with low contrast and color saturation. The second last pair of input and output images in Figure 7 shows the robustness of the approach with respect to scale and color. This fall image shows trees at distances ranging between 5 meters and over 500 meters whose colors range from magenta to green. The output image shows that almost all tree regions were correctly labeled. The last image pair in the same figure, though, shows that both false positives and negatives occur. The distant and hazy trees causes the network to label tree regions as non-tree regions, while some of the gravel on the road in the foreground was labeled as a tree region.

The test and training images were in GIF and JPEG format. JPEG is a lossy, subjective compression method that does not guarantee that the complexity and fine structure of compressed image blocks is preserved. More emphasis is placed on preserving the appearance of the image patch which heavily depends on the peculiarities of our own visual system. Some of the features including the hue, saturation, and value were obtained using MATLAB. This introduced a further deterioration of many color images since MATLAB can only handle 256 colors. The poorer classification results in the second and fourth image pair in Figure 8 could partially be due to this severe image quality reduction.
5 Conclusion

The use of a back-propagation network offers a simple solution to the laborious task of finding a good combination of the described color, greylevel co-occurrence, Fourier transform, Gabor, and fractal features. We have shown that feature sets like the one presented have sufficient expressive power to allow good generalization from only a few training images. Since the back-propagation algorithm is well understood and analyzed [5] it is also a straightforward matter to determine the usefulness of a specific feature if we had to reduce the number of features used to an optimal subset. The neural network approach offers a synthetic solution to the sensor fusion problem that is concerned with combinations of (possibly dependent) features for the purpose of classification and/or recognition. An analytical approach, on the other hand, would be difficult to conduct since the interactions even between modest numbers of dependent features is complex.

In future work, though, it may be useful to analyze the weights of the network to estimate their relative importance for the classification task. For example, it is conceivable that an eigen analysis of the feature space could reveal relevant and redundant features. Reducing the dimensionality of the feature space may improve the per-image processing time.

Acknowledgments

We would like to thank Guy Smith and Ian Burns from the University of Queensland for their excellent texture analysis package MEASTEX.

References


