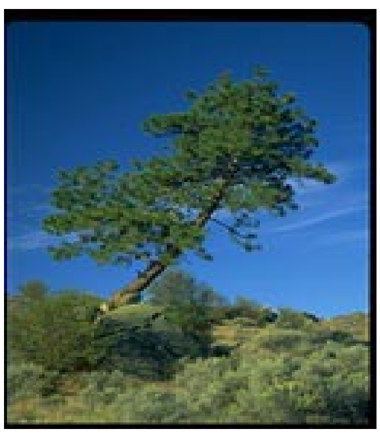
quarterly report for

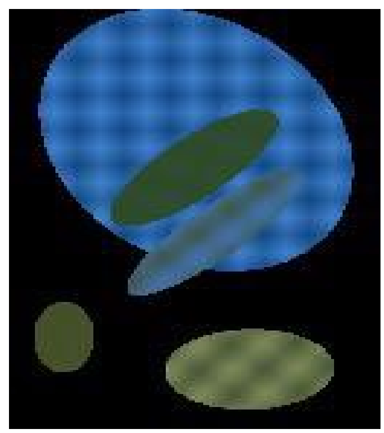
Application of Forest Image Analysis to Monitoring and Modeling of Psychological, Silvicultural, and Wildlife Habitat Attributes

submitted to:

Mr. Victor A. Rudis
USDA Forest Service
Southern Research Station
P. O. Box 928
Starkville, MS 39760-0928
Tel: 662-338-3109, Fax: 662-338-3101
Email: vrudis@usfs.msstate.edu

November 15, 1999





submitted by:

Z. Long, J. Picone
Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University
Box 9571
413 Simrall, Hardy Road
Mississippi State, Mississippi 39762
Tel: 662-325-3149
Fax: 662-325-3149
Email: {long, picone}@isip.msstate.edu

EXECUTIVE SUMMARY

The objective of this continuing project is to extract features from digitized forestry images and develop specific models for monitoring psychological, silvicultural, and wildlife habitat attributes. Previously, we had developed algorithms to extract diverse features such as color histograms, densities of long lines and short lines, entropy, etc., from the forestry images. We had also built an automatic image segmentation system based on those features. Our work this quarter has focused on improving the prototype segmentation system.

The densities of long lines and short lines are very important features for identifying trees, bushes and grass. Therefore, we had incorporated edge and line detectors into our system to extract such information. However, we had not done any comprehensive research on these detectors, which motivated us to study them in detail. In the course of this study, we discovered a bug in the existing code, which was believed to have resulted from a misunderstanding of the Canny algorithm for edge detection. Correction of this bug resulted in an improvement in performance from 83.7% to 80.8%.

Our new research has focused on designing better features. First, we have refined the edge and line detectors, which have been used to generate important line features such as the densities of long and short lines in an image, as we mentioned above. The refinement has been achieved by optimizing the key parameters involved in the algorithms. We estimated the key parameters, then applied them to the segmentation system in various combinations. After a series of carefully designed experiments, we obtained a set of optimal values of the key parameters. These optimized parameters reduced the error rate in segmentations from 80.8% to 70.3%.

Second, we have investigated the Discrete Cosine Transform (DCT) and integrated this technique into the feature extraction process. The DCT is probably the most widely used spectral analysis technique in the area of image processing due to many advantages it has over other techniques, such as its efficiency in packing energy into a small number of DCT coefficients and the existence of fast computational algorithms. We analyzed the amplitude distribution of the DCT coefficients on our forestry image database. Based on this analysis, we designed features which were generated by filtering DCT coefficients. We integrated these DCT-based features into the software, then evaluated their effectiveness in segmenting images. On a small pilot evaluation, the DCT-based features produced a segmentation error rate of 65.7%. This error rate is comparable to the best error performance we achieved before, which was 61.4%. A more extensive evaluation is underway.

Currently we are researching improving feature design using DCT coefficients. There are still some parameters which need to be investigated in more detail. For example, we need to determine an appropriate number of DCT coefficients to be used in generating features. We also need to understand how to optimize the filter size in our feature generation. In addition, we will explore other frequency domain analysis techniques such as Gabor filters. The Gabor filter is a promising analytical tool that imitates the first filtering stages of the human visual system. It works successfully in texture segmentation, which is similar to the segmentation problem of interest in this project. We also plan to try a decision tree-based segmentation system after we have developed features which are significant for distinguishing a particular forestry region. We expect that the decision tree-based system will simplify the segmentation problem.

TABLE OF CONTENTS

ABSTRACT

In this project, we wish to extract useful features from forestry images and build statistical models for forest structure analysis. In our prior efforts, we had developed a prototype automatic image segmentation system with a limited success. This quarter, we have worked on improving the features. We optimized the edge and line detectors which were used to generate important line features. In the process, we fixed an error with the previous implementation of the Canny edge detector. We also designed features based on the Discrete Cosine Transform (DCT), and evaluated their effectiveness in segmenting images. With these DCT-based features we obtained a system which gives us a 65.7% block classification error on a small data set. We are now researching improving feature design using DCT coefficients. We are also investigating other means of analysis in the frequency domain, such as Gabor filters.

1. INTRODUCTION

Image analysis is an important approach to understanding human perception. It takes a key role in building automated systems for monitoring natural resources. The goal of this project is to extract meaningful features from digitized forestry images using image analysis techniques and develop specific models for monitoring psychological, silvicultural, and wildlife habitat attributes. In our previous research, we had extracted diverse features such as color histograms, densities of long and short lines, entropy, etc., from the forestry images [1]. We had developed an automatic image segmentation system based on those features, but its performance was not exciting [2].

A careful study of the prototype system indicated that the failure should be attributed to features which were not good enough to distinguish between the different categories. Consequently, we have been focusing our efforts on developing salient features during this period.

First, we chose to improve the line features, i.e., the densities of long and short lines in a forestry image. These line features are important because long lines identify trees, while short lines are indications of the presence of bushes and grasses. Previously, we had incorporated edge and line detectors into our system to extract the line features, but we had not done much work to optimize the performance of those detectors. Therefore, we elaborated the edge and line detectors in this quarter by tuning the key parameters involved. In the course of this optimization, we discovered and fixed an error with the previous implementation of the Canny edge detector.

Second, we investigated the spectral analysis technique of DCT. Although we had extracted many features before, we did not have any spectral features yet. We chose DCT because it has many advantages over other frequency analysis techniques, such as a high efficiency in packing energy into DCT coefficients, the availability of fast computation algorithms, etc.. We analyzed the amplitude distribution of the DCT coefficients on all the forestry regions of interest in this study, designed various feature vectors based on the DCT coefficients, and evaluated their effectiveness in segmenting images. The best DCT feature we have developed so far presents an error rate of 65.7% for the segmentation problem on a small data set.

Although the preliminary analysis implies that DCT-based feature design is promising, there is still much research work which needs to be done. For example, some parameters involved in the design, such as the filter size and the number of DCT coefficients, have not been investigated in

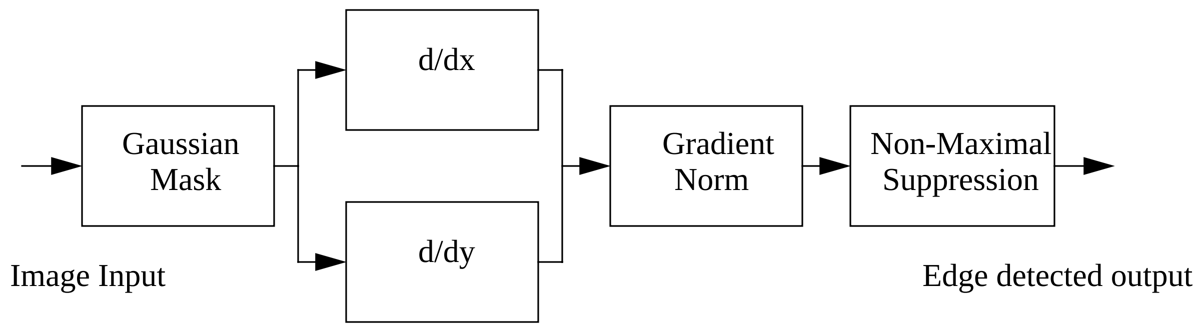


Figure 1. Block diagram for the Canny edge detector.

great detail. Currently, there is a lack of understanding of the physics behind the DCT analysis, which is hindering our progress toward a successful system. In addition to the DCT method, we plan to explore analysis techniques in the frequency domain further. We will also integrate the algorithm of decision tree into our system in the near future.

2. EDGE DETECTOR AND LINE DETECTOR OPTIMIZATION

The densities of long and short lines in a forestry image are important features for automated image classification and segmentation. This importance had been verified by the good classification results we achieved in the previous years of this project [1]. We have traditionally used the Canny algorithm [3] to perform edge detection, and a simple thresholding scheme to connect edges into line segments (the latter is called a line detector). However, there are several parameters involved in these algorithms, and no comprehensive study has been performed to optimize them.

The Canny edge detector works as shown in the block diagram of Figure 1. Firstly, an $n \times n$ Gaussian mask is used to smooth out noises presented in the input image. Then, differentiations in both the horizontal and the vertical directions are performed. Afterwards, gradient values are calculated for every pixel of the image. The final stage of the edge detection is non-maximal suppression. In this step, two threshold values are set. If the gradient value of a pixel is greater than the higher threshold, then the pixel is treated as an edge pixel; otherwise, only if its gradient is greater than the lower threshold and it is connected to an edge, is it still considered as an edge pixel.

For simplicity, we set the Gaussian mask size as 3 x 3. We assume both mean values for the two-dimensional Gaussian distribution are zeros, and both variances are equal. All we need to do is to determine the variance value. The two thresholds in the last step are also key to the performance of the edge detector. If they are too high, we will lose edges. On the other hand, if they are too low, we will have false edges in the output.

Our line detector is simple. It takes the output from the edge detector, then compares the edge lengths with the threshold data. For an edge to be a long line, its length must fall into the range of

Experiment No.	σ	High Edge Threshold	Low Edge Threshold	Short Line Range	Long Line Range	Percentage Error Rate
1	9	200	70	15 - 25	>50	80.8
2	3	200	70	15 - 25	>50	76.9
3	3	180	60	15 - 25	>50	76.1
4	1	200	70	15 - 25	>50	73.2
5	1	180	60	15 - 25	>50	71.8
6	1	120	40	15 - 25	>50	78.7
7	0.1	180	60	15 - 25	>50	88.7
8	0.1	120	40	15 - 25	>50	93.5
9	1	180	60	15 - 25	>25	76.8
10	0.5	180	60	15 - 25	>25	75.6
11	1	180	60	15 - 30	>60	**70.3**
12	0.5	180	60	15 - 30	>60	71.9

Table 1. Optimization experiments with the edge detector and the line detector.

long lines. Similarly, it has to fit the range of short lines to be considered as a short line.

According to the above discussion, we chose the following parameters to optimize: the variance of the Gaussian distribution, the high and low thresholds for the non-maximal suppression, and the threshold values for the line detector. We applied various combinations of these parameters to the image segmentation system, and chose the set of parameters which gave us the best performance in the sense of block classification errors. The experiments are summarized in Table 1.

The optimization was carried out on data set 1 of the USFS Pre-Phase 01 data. The features were set to be “long lines plus short lines,” the frame size was 64 x 64, and the window size was 128 x 128. As a comparison, the system before optimization yielded an error rate of 83.7% under these same conditions.

As we were in the process of the optimization, we discovered an error with the previous implementation of the Canny algorithm. The non-maximal suppression procedure, which is supposed to suppress the lower gradient values, had been implemented as: a pixel is an edge pixel only if its gradient value falls between the two thresholds. In this case, pixels with gradient values greater than the higher threshold will be masked out. Obviously, there had been a misunderstanding of the algorithm. We fixed this error before we optimized the system.

3. DCT ANALYSIS

Previously, we had extracted many features from the forestry images, but none of them presents spectral information of the image regions. Given the fact that most forestry regions display remarkable variations in the spatial domain, and that the patterns of those variations change with each specific kind of region, we believe that there should exist some frequency features helpful for discrimination between those regions. Therefore, we investigated the discrete cosine transform.

The approach of discrete cosine transform is a widely used frequency analysis method in image processing. It has many advantages over other transformation techniques. For example, only real computation is involved in DCT; no spurious spectral components will come out as is the case with the discrete Fourier transform (DFT); fast DCT implementations are available; and most important, with this method, energy is packed efficiently into a small number of DCT coefficients [4].

Here is the definition equation for the two-dimensional forward DCT of an $n \times n$ data block,

$$F(u, v) = \frac{4C(u)C(v)}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} f(j, k) \cos\left[\frac{(2j+1)u\pi}{2n}\right] \cos\left[\frac{(2k+1)v\pi}{2n}\right] \tag{1}$$

where the constants are computed as

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & w = 0 \\ 1 & otherwise \end{cases}. \tag{2}$$

On the other hand, the inverse 2-D DCT (IDCT) is defined as

$$f(j, k) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} C(u)C(v)F(u, v) \cos\left[\frac{(2j+1)u\pi}{2n}\right] \cos\left[\frac{(2k+1)v\pi}{2n}\right] \tag{3}$$

where $C(w)$ is calculated the same way as it is in the forward DCT.

To better understand the idea of DCT analysis, we first carried out forward DCT on the forestry image data. We chose one typical region from our image database for each of the six categories under study, i.e., tree, foliage, bush, grass, background sky and sky. Then we generated the DCT coefficients of the selected data with the block size ranging from 8 to 128. There appeared to be a unique pattern of the coefficients' amplitude distribution for each kind of region. To illustrate this more intuitively, we plotted all the original data and the outcome DCT coefficients in MATLAB. Some example plots are shown in Figure 2-4. The plots shown in Figure 2 and Figure 3 describe amplitude distributions of the DCT coefficients calculated on all kinds of regions with the image block size set as 64 x 64, while the plots in Figure 4 show the DCT coefficients computed on the foliage region with diverse block sizes.

From these plots, we note that the DCT coefficients' amplitude distribution varies significantly

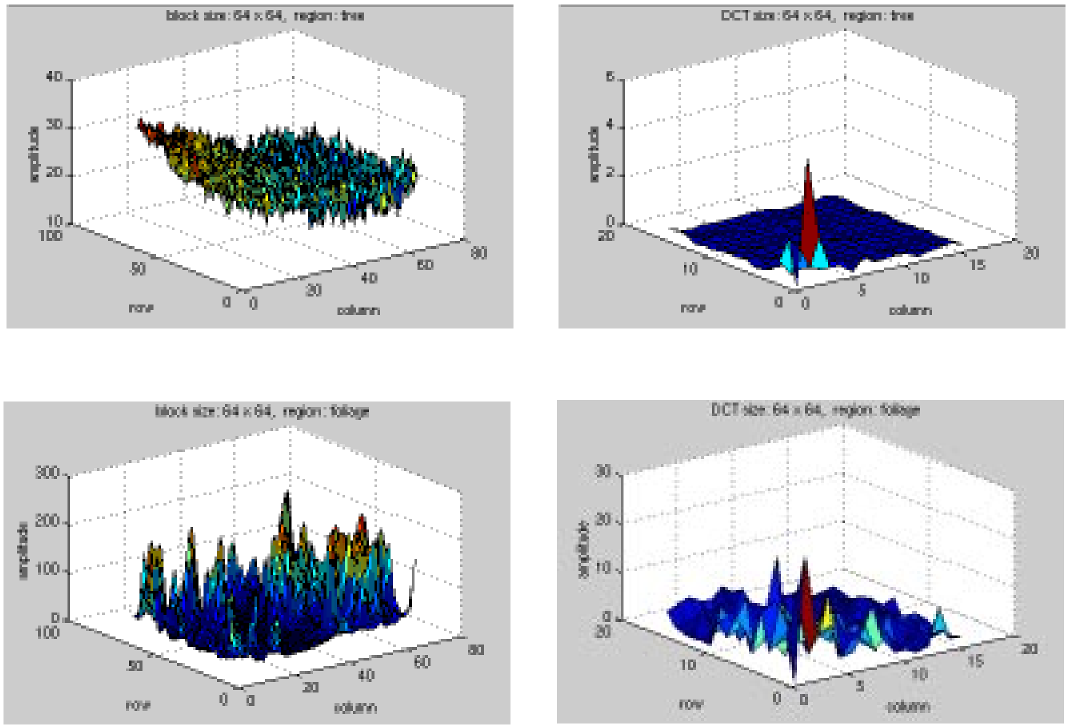


Figure 2. The amplitude distribution plots of DCT coefficients computed on typical forestry regions. Note the left column plots are for the original data, and the right ones are for the DCT coefficients.

between the regions. This variation may indicate a potential of DCT coefficients to discriminate between the forestry regions. Another observation is that the block size for the DCT computation has a noticeable impact on the pattern of the amplitude distribution. The reason for this sensitivity to the block size may be that small sizes are not sufficient for the data to contain enough texture information, resulting in patterns inconsistent with those generated on larger blocks. Also, we notice an interesting phenomenon that whatever the block size is, the energy tends to be packed into the lower frequency components by the transformation. Therefore, the lower frequency components are much more important than the higher components, since they collect the majority of the energy.

Based on these observations, we believe that if we choose a suitable block size and use the lower DCT coefficients to generate features, we will be able to achieve a system with good segmentation performance.

4. DCT-BASED FEATURE DESIGN AND EVALUATION

We designed a few features based on what we had observed in the DCT analysis. The basic idea behind the design is: first compute DCT coefficients on an image block, then filter them in the frequency domain, after that choose the lower frequency components as the feature vector. To be more specific, we designed our baseline system in the following way: perform DCT on the green pixels, average every four of them for the filtering stage, then choose the first 16 filtered outputs to create the feature vector. That means we build the feature vector on the first 64 DCT coefficients.

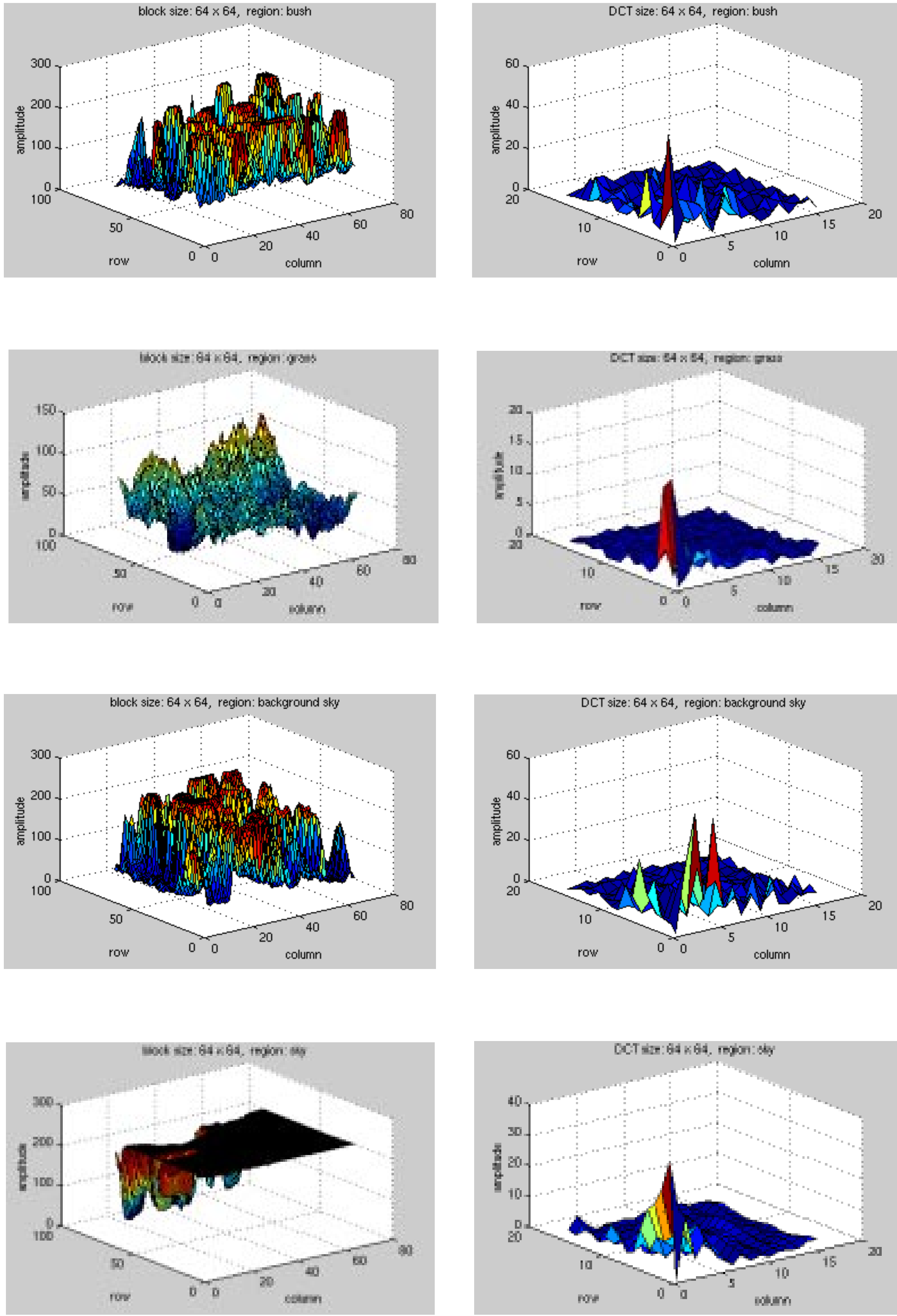


Figure 3. The amplitude distribution plots of DCT coefficients computed on typical forestry regions (continued).

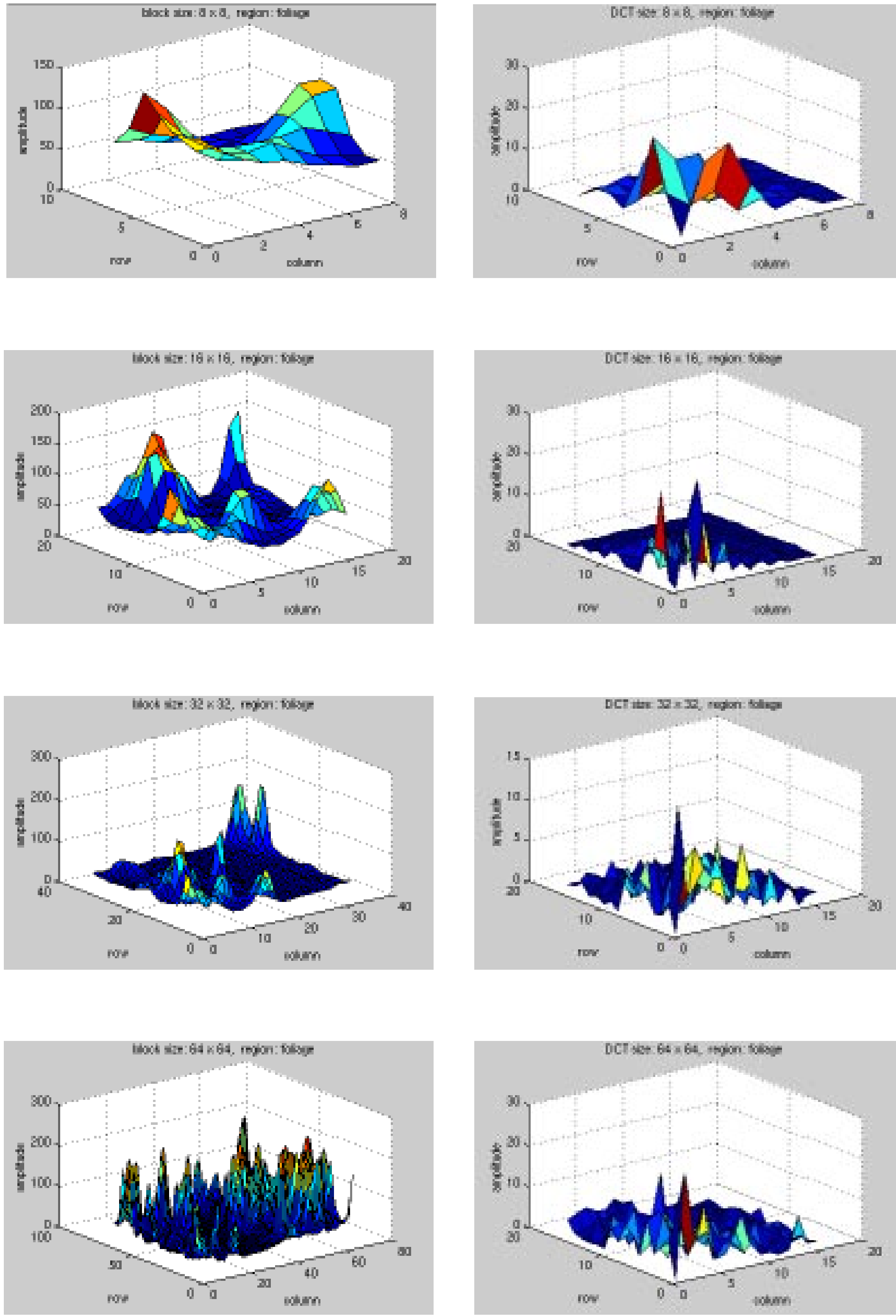


Figure 4. The amplitude distribution plots of DCT coefficients computed on a typical foliage region with different block sizes. Note the left column images are for the original data and the right ones are for the DCT coefficients.

This choice is reasonable since we have already verified that the energy is packed into the lower frequency components, and for those typical regions, the first 64 DCT coefficients concentrate the majority of the energy. If we include all DCT coefficients in the feature computation, we will meet problems with large DCT block size. Suppose we are doing DCT on a 64 x 64 block, and we are averaging every four of all the DCT coefficients to generate a feature vector, then the dimension of the outcome feature vector should be 64 x 64 / 4 = 1024. This large size will unnecessarily increase the computational complexity of training and test processes.

To examine how well our feature design fits for the segmentation problem, we ran several evaluation experiments on a small data set. The training data set is shown in Table 2. Note that we chose different images to train the models. This was for the purpose of enhancing the training efficiency. Since the six categories of regions are distributed unevenly across the database, if we had used only one training set for all the models, the training set would have had to be increased to a much larger size to guarantee a sufficient amount of training data for each model. The test data set consisted of 5 random images with a total of 1406 64 x 64 image blocks.

As a first step, we evaluated the effects of the windowing scheme on this baseline system. The results indicate that the system performance is sensitive to the windowing parameters. As we see from Table 3, when the frame and window sizes are set to 64 x 64 and 96 x 96 respectively, the

Region	Number Of Images	Number Of Blocks	Block Size
tree	5	523	64 x 64
foliage	6	632	64 x 64
bush	5	520	64 x 64
grass	6	523	64 x 64
background sky	10	675	64 x 64
sky	10	61	64 x 64

Table 2. The training data set.

Experiment No.	Frame Size	Window Size	Percentage Error Rate
1	32 x 32	32 x 32	72.6
2	32 x 32	64 x 64	67.6
3	64 x 64	64 x 64	67.9
4	64 x 64	96 x 96	**66.6**
5	64 x 64	128 x 128	67.6

Table 3. The performance evaluations with various windowing schemes.

Experiment No.	Color Component	Percentage Error Rate
1	red	**65.7**
2	green	66.6
3	blue	67.3

Table 4. The performance evaluations with different color components.

Experiment No.	Filter Size	Percentage Error Rate
1	4	66.6
2	2	69.2

Table 5. The performance evaluations with different frequency filter sizes.

Experiment No.	Amount Of DCT Coefficients Involved	Percentage Error Rate
1	64	66.6
2	128	72.5

Table 6. The performance evaluations with different amounts of DCT coefficients involved in the feature computation.

system performs best on the green pixels.

Then we investigated with different color features, i.e., we computed DCT coefficients on the red, green and blue pixels respectively. For these evaluations, we set the frame size as 64 x 64, and the window size as 96 x 96. As shown in Table 4, generally the performances with all three colors are comparable. This similarity is reasonable because DCT coefficients are supposed to describe the spatial variation, or the texture pattern the image block displays. This texture characteristics should not vary noticeably with different color components.

In the schemes discussed so far, we chose the frequency filter size to be 4. However, that filter parameter may not necessarily be the optimal choice. Therefore, we tested a smaller filter size of 2. In this scenario, all other conditions for feature generation were kept the same as the baseline system. The windowing parameters also remained unchanged, that is, we used 64 x 64 frames and 96 x 96 windows. We observe in Table 5 that the system with a larger filter works slightly better. However, the difference in the error performance is not significant. To draw a conclusion as to what filter size is more promising, we will have to evaluate more values for the filter size. There may be an optimal filter size in existence, and that optimal value may change with the colors and the windowing parameters.

Finally, we evaluated the impacts of the amount of DCT coefficients, which are involved in the feature computation, on the system performance. We modified the baseline system with different numbers of DCT coefficients. We used the first 64 and 128 coefficients to generate feature vectors respectively. Hence, for the first experiment, we had a 16-dimensional feature vector, while for the second one, it was of 32-dimension. The results verified the idea that using more DCT coefficients is not necessarily better - with 128 DCT coefficients involved in the feature computation, the system performance is 5% worse than that of the system with the features based on only 64 coefficients.

5. SUMMARY

In this report we described our work during the first quarter of this year. We have mainly worked on the feature extraction. First, we optimized the edge and line detectors by tuning the key parameters. Second, we investigated the technique of discrete cosine transform, which is a promising way to extract frequency domain features. We analyzed the DCT coefficients' amplitude distribution on typical forestry regions and discovered a few basic rules which might be helpful for the segmentation task. Based on this analysis, we designed some feature vectors and evaluated them on a small data set. So far, the best DCT-based feature vector is generated as: compute DCT coefficients on the red color pixels with 64 x 64 frames and 96 x 96 windows, then choose the first 64 DCT coefficients and average every four of them to generate a 16-dimensional feature vector. The corresponding system performance is 65.7% block classification errors on the small data set.

6. FUTURE WORK

We need to improve the frequency features on the basis of our preliminary design with the DCT coefficients. As discussed earlier, there are still some parameters, such as the frequency filter size, and the number of DCT coefficients involved in the computation, which need to be optimized. Moreover, to combine the DCT coefficients more effectively, we will have to understand thoroughly the physics behind the DCT.

In the mean time, we will explore analysis techniques in the frequency domain further. As an example, the Gabor filter is a promising analytical tool. This filtering approach imitates the first filtering stages of the human visual system and helps to generate impressive results for the texture segmentation problem [8]. We are now studying the feasibility of applying Gabor filters to our image segmentation system.

After we succeed in finding discriminating features for the system, we plan to try a decision tree-based segmentation scheme. We will carry out a series of two-way classifications instead of trying to classify an image block into one of the six categories at one process. This decision tree-based algorithm may simplify the segmentation problem and show us great improvement.

7. REFERENCES

[1] N. Kalidindi, V. Ramani and J. Picone, "Scenic Beauty Estimation of Forestry Images," *Final Project Report for the Southern Forest Experiment Station, United States Forest Service*, Institute for Signal and Information Processing, Mississippi State University, December 1998.

[2] Z. Long and J. Picone, "Applications of High Performance Statistical Modeling to Image Analysis of Forest Structure," *Final Project Report for the Southern Research Station, United States Forest Service*, Institute for Signal and Information Processing, Mississippi State University, August 1999.

[3] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, November 1986.

[4] M. Rabani and P. W. Jones, *Digital Image Compression Techniques,* SPIE Optical Engineering Press, Bellingham, Washington, USA, 1991.

[5] J. Lay and L. Guan, "Image Retrieval Based on Energy Histograms of the Low Frequency DCT Coefficients," *ICASSP Proceedings*, vol. 6, pp. 3009-3012, Phoenix, Arizona, USA, March 1999.

[6] Y. Xiao, N. P. Chandrasiri, Y. Tadokoro, and M. Oda, "Recognition of Facial Expressions Using 2D DCT and Neural Network," *Electronics and Communications in Japan*, vol. 82, no. 7, pp. 1-11, July 1999.

[7] Y. Huang and R. Chang, "Texture Features for DCT-Coded Image Retrieval and Classification," *ICASSP Proceedings,* vol. 6, pp. 3013-3016, Phoenix, Arizona, USA, March 1999.

[8] A. K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167-1186, December 1991.