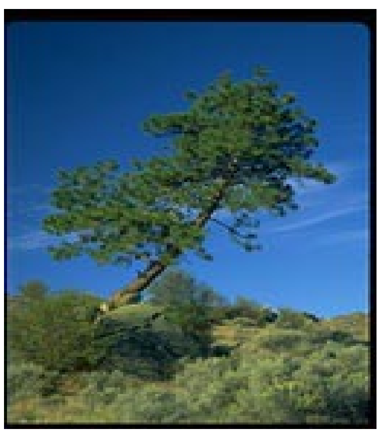
quarterly report for

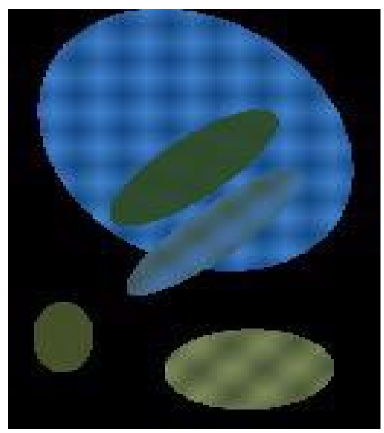
Application of Forest Image Analysis to Monitoring and Modeling of Psychological, Silvicultural, and Wildlife Habitat Attributes

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

February 15, 2000





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

ABSTRACT

This quarter we primarily worked on optimizing the edge and line detectors incorporated in our image analysis software. We changed our optimization scenario and designed a metric for the evaluation procedure. With our optimized detectors, we achieved a match error rate of 29.3%, as well as a low level of insertion, on the test set 01 from the Pre-phase 01 image data. Meanwhile, we started investigating the frequency analysis technique of Gabor filters. We are now in the process of replicating a system which uses Gabor filters to do unsupervised texture segmentation, in an attempt to understand the behavior of Gabor filters.

1. INTRODUCTION

Last quarter, we worked on optimizing edge and line detectors, and reported the corresponding results [1]. However, the scenario for that investigation was not comprehensive enough to cover all significant aspects of edge and line detection. With that scheme, we adjusted the values of several parameters, applied them to the image segmentation system, and made decisions on how good they were at the detection task by examining the segmentation results. No direct evaluation of the actual detection output had been performed.

This deficiency in the old scheme motivated us to continue the optimization work this quarter. We carefully redesigned our paradigm to evaluate the detection data directly. First of all, we manually transcribed significant lines as reference data. Then we carried out experiments to investigate the effects of various parameters on the performance of the detectors.We performed the investigation more systematically than in the previous attempt and covered more aspects of the problem. We also designed an appropriate metric which lays emphasis on the physical properties of lines, such as location, length, and slope, to assist in the performance evaluation.

In the meantime, we started researching Gabor filter techniques. As discussed in [2], the Gabor filter is a promising filtering technique for visual image analysis, mainly because it is a good fit for the receptive field profiles of simple cells in the striate cortex. A bank of Gabor filters has been used to build a successful unsupervised texture segmentation system [2]. Since the majority of the forestry images in our database are texture images (E.g., foliage, grass, bush and sky are all unique texture patterns.), we believe that the Gabor filter-based features will help in our image segmentation system. As a first step, we are replicating the system described in [2], in order to acquire an understanding of the behavior of these filters.

2. OPTIMIZATION OF EDGE AND LINE DETECTION

The densities of long and short lines in a forestry image are important features for automated image classification and segmentation. Last quarter, we worked on optimizing the edge and line detectors embedded in our system [1]. However, at that time, our evaluation paradigm did not match the problem of edge and line detection very well. We applied various settings of edge detection-related parameters to the image segmentation system, and determined the goodness of the settings by the resulting segmentation error rates. This method may help in finding good parameter values for segmentation, but it does not directly determine the quality of the output detection. Hence, the optimality of the tuned parameters is not convincing.

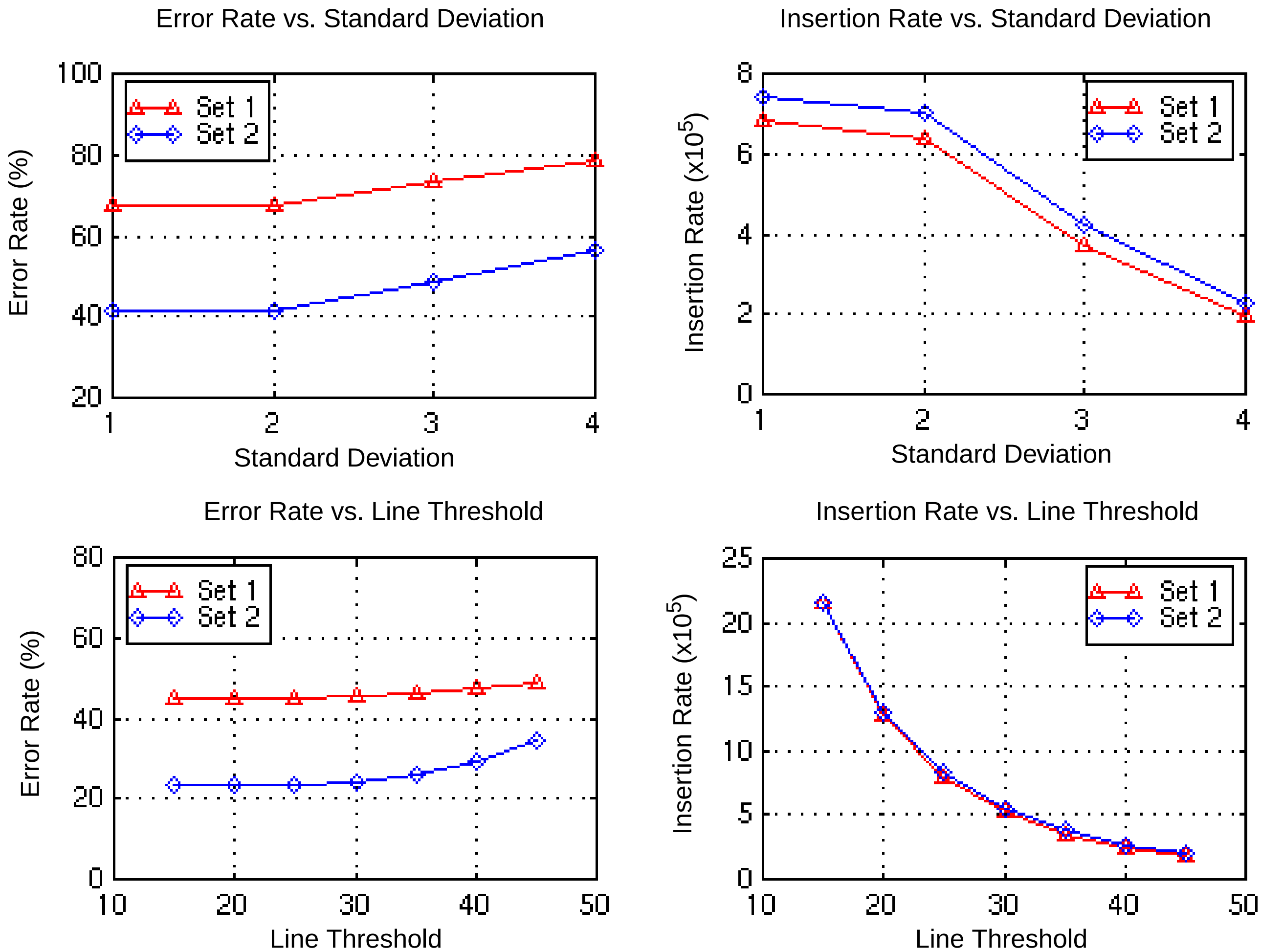


Figure 1. Plots from optimization experiments with standard deviation and line threshold.

In this quarter we redesigned the optimization strategy. First, we created a set of reference data by manually transcribing significant lines from the image database. Then, we carried out edge and line detection on the same images and compared the detection output with the reference data. The performance of the detectors was evaluated on how well the detection data matched the reference data. For evaluation purposes, we designed a metric which compares the physical properties of both detection lines and reference lines.

The evaluation can be described as a two-stage procedure. In the first stage, we loop over all reference lines and find the best match for each reference line in the output detection. In the second step, we evaluate those best matches found in the previous stage.

To find the best match for a reference line, we loop over all detection lines and compute the distance from the middle point of each detection line to the reference line. The one giving the smallest distance is the best match. We count all detection lines without matches in the reference data as “insertion errors,” or false alarms.

In the second step, we need to handle several cases. The first case is with close parallel lines of

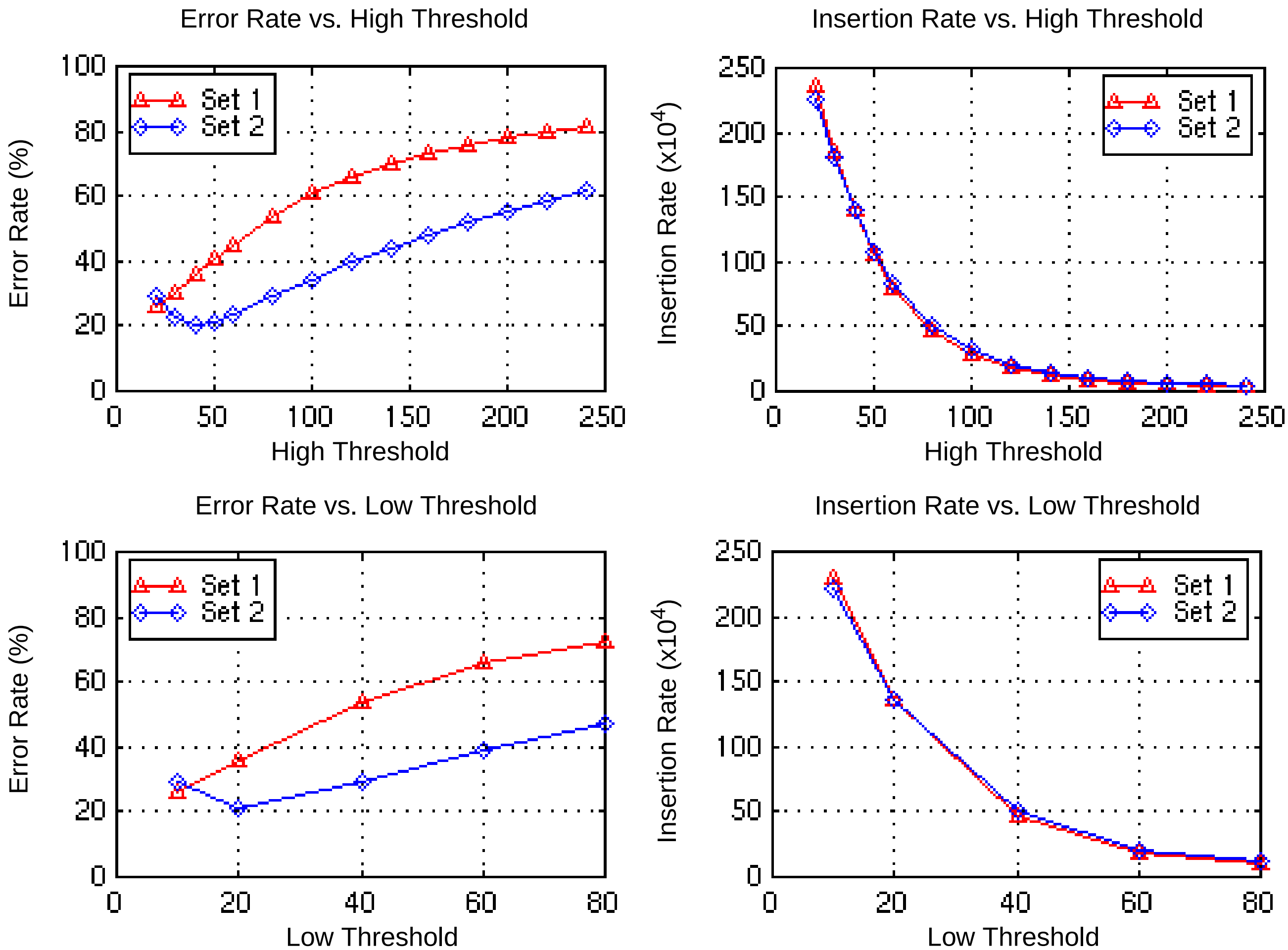


Figure 2. Plots from optimization experiments with high and low edge thresholds.

approximately the same length. We count this case as a correct recognition. The second case is with close parallel lines of unequal length. We count it as a correct detection if the length difference is within 25% of the length of the reference line. The third case involves close non-parallel lines of approximately the same length. We consider it a correct recognition if the angle between the two lines is less than 20 degrees. The fourth case is that of close non-parallel lines of unequal length. We count it as a correct detection if the angle between the two lines is less than 20 degrees and the length difference is within 25% of the length of the reference line. All other cases will be counted as errors.

We chose two data sets for experimentation. One contains 165 images from the training set 01 of Pre-phase 01. The other consists of 159 images from the test set 01 of Pre-Phase 01.

We experimented with the same parameter set we had used previously [1]. That is, we tuned the high and low edge thresholds, the line threshold, and the Gaussian variance. We investigated the influence of these parameters on the edge and line detection output. When we were experimenting with one parameter, we set all the others to be fixed values. For the parameter under investigation, we swept through the range of all possible values.

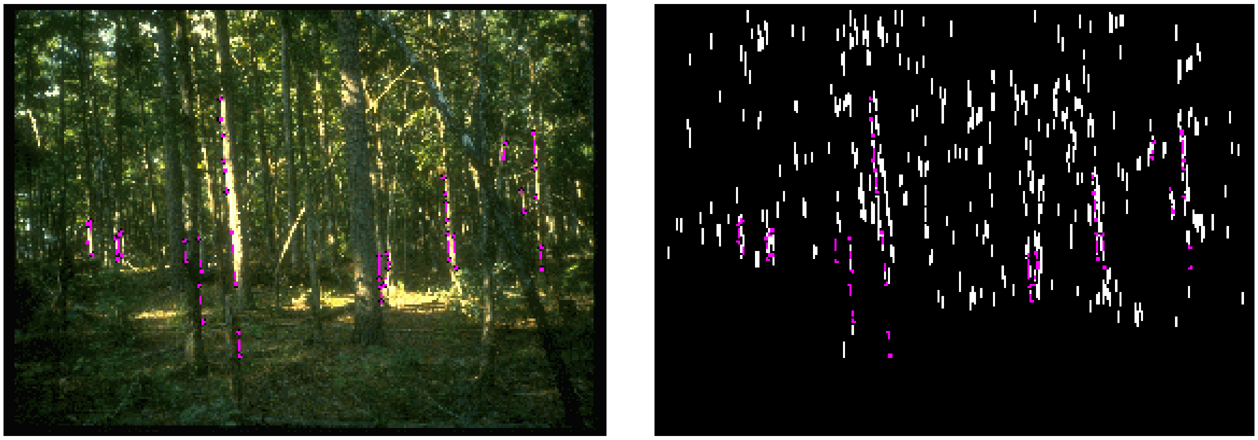


Figure 3. An example detection image with the optimal parameter setting. The left image is the original one. The colored lines are reference lines.

Figure 1 and 2 illustrate the results with various experimental conditions. The corresponding data is documented in Table 1 - 4 in the appendix.

The performance of the detectors is measured by a combination of two resulting values. One is the error rate; the other is the insertion rate. A system which achieves low error rate and low insertion rate at the same time is desirable. However, one interesting phenomenon we noticed from the experiments is that, when we lower the thresholds and the Gaussian variance, the error rate tends to decrease, and the insertion rate increases. The reason for this trade-off is that lower thresholds will result in more lines, which will increase the chance for both correct matches and undesirable insertions, simultaneously.

To find the optimal parameter set, we need to keep a balance between the error rate and the insertion rate. By examining the error rate curves and the insertion rate curves, we found that with the following settings of parameters, both the error rate and the insertion rate are at a comparatively low level: 2.0 for the Gaussian standard deviation, 60 for the high threshold, 30 for the low threshold, and 40 for the line threshold. With these values, we achieved an error rate of 29.3% on data set 2. The corresponding insertion rate is 272073 lines with all 159 images. Given the fact that we had transcribed only significant lines as reference data, some of the "inserted lines" may actually be correct detection, and such an insertion rate seemed acceptable. An example detection image with these optimal conditions is shown in Figure 3.

3. GABOR FILTERS

Gabor filters are important in visual image analysis. They have been found to fit very well for receptive field profiles of simple cells in a striate cortex [2]. The impulse response of an even-symmetric Gabor filter is given by

$$h(x, y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^{\,2}} + \frac{y^2}{\sigma_y^{\,2}}\right]\right\}\cos(2\pi u_0 x) \cdot \qquad (1)$$

Here, u_0 is the frequency of a sinusoidal plane wave along the x axis (or the 0^o orientation), σ_x is the space constant of the Gaussian envelope along the x axis, and σ_y is the other space constant along the y axis. To obtain a Gabor filter with an arbitrary orientation, we need to rotate the $x-y$ coordinate system accordingly.

The Fourier domain representation of (1) is given by

$$H(u, v) = A\exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^{\ 2}} + \frac{v^2}{\sigma_v^{\ 2}}\right]\right\} + A\exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^{\ 2}} + \frac{v^2}{\sigma_v^{\ 2}}\right]\right\} \qquad (2)$$

where $\sigma_u = \frac{1}{2}\pi\sigma_x$, $\sigma_v = \frac{1}{2}\pi\sigma_y$, and $A = 2\pi\sigma_x\sigma_y$. This equation is also referred to as a *modulation transfer function* (MTF) since it specifies the amount by which the filter modifies each frequency component of the input image.

Gabor filters are able to keep an optimal balance between the resolution in the spatial domain and that in the frequency domain [2]. This is a significant property for the texture segmentation problem, where high resolution in the spatial domain is necessary for locating texture boundaries, and smaller bandwidth in the frequency domain is desirable for distinguishing between different textures. The usefulness of Gabor filters in texture segmentation has been demonstrated by the research work described in [2].

We are now replicating the texture segmentation system described in [2]. This is an attempt to obtain an understanding of the behavior of Gabor filters.

4. SUMMARY AND FUTURE WORK

This quarter, we mainly worked on the optimization of the edge and line detectors. We redesigned our optimization scenario. In the process, we also designed a metric to evaluate how well an output detection matches the corresponding reference data. With the new paradigm, we investigated the effects of various parameters involved in the algorithms on detection performance. We acquired an optimal set of parameters which resulted in a low error rate of 29.3%, as well as a low insertion rate, on data set 2. Meanwhile, we proceeded to studying Gabor filters, which are promising for texture image segmentation.

For our future work, we plan to design features on the basis of images filtered from a bank of Gabor filters, and then to test the discrimination ability of those features. We expect good results from these Gabor filter-based features, since forestry images are mostly texture images. Afterwards, we will investigate algorithms which incorporate syntactic information into the block classification procedure.

5. REFERENCES

[1] Z. Long and J. Picone, "Application of Forest Image Analysis to Monitoring and Modeling of Psychological, Silvicultural, and Wildlife Habitat Attributes," *Quarterly Status Report for the Southern Research Station, United States Forest Service*, Institute for Signal and Information Processing, Mississippi State University, November 1999.

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

[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] R. N. Strickland and D. Chang, "An Adaptable Edge Quality Metric," *Proceedings of SPIE*, vol. 1360, pp. 982-995, Bellingham, Washington, USA, October 1990.

APPENDIX

Here is the data for all the optimization experiments on the edge and line detectors.

Exp.	σ	Set 1				Set 2			
		Reference	Detection	Error(%)	Insertion	Reference	Detection	Error(%)	Insertion
1	1.0	3176	681636	67.6	680608	3376	746269	41.7	744302
2	2.0	3176	640911	67.5	639880	3376	703848	41.6	701877
3	3.0	3176	371683	73.1	370830	3376	421944	48.6	420208
4	4.0	3176	197896	78.6	197216	3376	232348	56.5	230879

Table 1. Optimization experiments with the Gaussian variance. Here the high and the low thresholds for the edge detector are 180 and 60 respectively, and the line threshold is 15.

Exp.	Line Threshold	Set 1				Set 2			
		Reference	Detection	Error(%)	Insertion	Reference	Detection	Error(%)	Insertion
1	45	3176	178213	48.8	176586	3376	198994	34.2	196773
2	40	3176	248794	47.2	247118	3376	274461	29.3	272073
3	35	3176	355927	45.9	354210	3376	387083	25.7	384575
4	30	3176	524477	45.3	522741	3376	560715	24.0	558148
5	25	3176	801505	45.1	799761	3376	840532	23.4	837947
6	20	3176	1278402	45.0	1276656	3376	1309756	23.4	1307169
7	15	3176	2163597	45.0	2161851	3376	2160635	23.4	2158048

Table 2. Optimization experiments with the line threshold. Here the high and low thresholds for the edge detector are 60 and 30 respectively, and the standard deviation is 2.0.

Exp.	Low Threshold	Set 1				Set 2			
		Reference	Detection	Error(%)	Insertion	Reference	Detection	Error(%)	Insertion
1	80	3176	95603	72.4	94728	3376	113599	47.0	111810
2	60	3176	186558	65.8	185471	3376	213187	39.0	211128
3	40	3176	460542	53.9	459077	3376	500603	29.6	498226
4	20	3176	1370829	36.0	1368795	3376	1370686	21.4	1368032
5	10	3176	2308121	26.3	2305780	3376	2211064	29.2	2208674

Table 3. Optimization experiments with the low threshold of the edge detector. Here the high threshold is 100, the standard deviation is 2.0, and the line threshold is 25.

Exp.	High Threshold	Low Threshold	Set 1				Set 2			
			Reference	Detection	Error(%)	Insertion	Reference	Detection	Error(%)	Insertion
1	240	120	3176	34555	81.5	33969	3376	42993	61.8	41705
2	220	110	3176	42697	80.0	42063	3376	52557	58.9	51168
3	200	100	3176	53522	78.4	52837	3376	65214	55.1	63698
4	180	90	3176	68390	76.0	67627	3376	82599	52.1	80981
5	160	80	3176	90358	73.0	89501	3376	107854	48.2	106106
6	140	70	3176	125070	70.2	124125	3376	146506	44.3	144626
7	120	60	3176	182904	66.2	181831	3376	209430	39.5	207389
8	100	50	3176	284695	60.6	283444	3376	317499	34.2	315279
9	80	40	3176	468147	53.6	466674	3376	507845	29.0	505449
10	60	30	3176	801505	45.1	799761	3376	840532	23.4	837947
11	50	25	3176	1060511	40.5	1058622	3376	1086917	21.1	1084253
12	40	20	3176	1408473	35.7	1406430	3376	1406020	20.7	1403342
13	30	15	3176	1860484	30.4	1858273	3376	1812818	22.8	1810212
14	20	10	3176	2350541	26.4	2348203	3376	2248794	29.0	2246397

Table 4. Optimization experiments with the high and low thresholds of the edge detector. Here the standard deviation is 2.0, the line threshold is 25, and the low thresholds are set to be half of the high thresholds as recommended in Canny's paper.