EDGE DETECTION USING DYNAMIC OPTIMAL PARTITIONING

Jeffrey D. Scargle[1] and Mahmoud K. Quweider[2]

[1]Space Science Division, NASA Ames Research Center
Moffet Field, CA 95035-1000
[2]CS/CIS Department, University of Texas, Brownsville
Brownsville, Texas 78520
USA

ABSTRACT

In this paper, a new edge detector (boundary extractor) is proposed based on finding major change points in a local one-dimensional window of the image intensity values of the rows or columns. The approach amounts to separating the pixels in the window into sets or regions of constant intensities with the edge pixels providing transition points. The edge points are found based on partitioning the interval in an optimal way using dynamic programming with an appropriate cost function. Different cost functions are introduced for the algorithm with simulation results that show the detector's effectiveness even in the presence of noise.

1. INTRODUCTION

An edge is usually defined as an abrupt transition of intensity or gray levels across a group of relatively uniform pixels [1-4]. Despite their presence in small percentages in any image, and as suggested by biological and physiological evidence, edges are high in their information content and therefore are of fundamental importance in image analysis, computer vision and pattern recognition. This is usually attributed to the edges primary role in perception and recognition and to their ability to reduce the total amount of data present in an image without losing the main features or structures needed for perception and recognition operations.

The literature describes many edge detection techniques [1-4]. These techniques have a wide range of approaches ranging from gradient-based high-pass filtering techniques to optimization techniques based on a specific model, curve or surface fitting, neural network based, genetic programming based, statistical and error minimization or reduction. Most of these techniques follow a two-step process of image smoothing followed by a detection or thresholding operation. The smoothing step is used to reduce the noise and blur the edges while the second step is used to find the edge map based on the strength of the edge.

In this paper we propose a new approach to edge detection. Our algorithm is a one step algorithm that starts by dividing the image into rows and columns and then processes each separately; we will use the image rows as inputs for the rest of the paper with the understanding that the same steps are applied to columns as well. Each row is passed to the dynamic optimal portioning algorithm which finds the edge points, one-dimensional edge map, as the change points of that row. The change points are found based on either a Gaussian-based or Poisson-based cost function which reflects our posterior knowledge that pixels within two successive change points have a constant value and therefore considered non-edge pixels. In addition to the cost function, a prior value is used to find the edge at different scales. A segmentation scheme is also introduced to reduce the computational complexity of the algorithm for each row. Our approach is similar to Mumford and Shah's variational model [8] in that edges are found by optimizing a cost function; our approach, however, relies on dynamic programming for its optimization.

2. EDGE DETECTION MODEL

2.1 Optimal Partitioning on an Interval

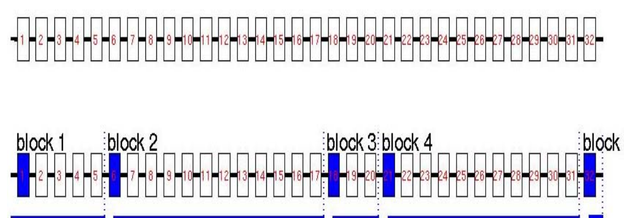

Figure 1. Example of an Optimal Partitioning of an interval

We pose the problem of finding a set edge pixels within an image row (or column) as a problem of optimal partitioning of that row, denoted here as R, into a set of *M* uniform regions

$\{R_i \mid i = 0,2,3,...M-1\}$ With $R = \bigcup_{i=0}^{M-1} R_i$ and $R_i \cap R_j = \varnothing$ *for* i ≠ j, using an appropriate cost function. The number M is different for each row and is not defined a head of time. The optimal partitioning algorithm should find this number based on the cost function. A partition here is defined as a set of one or more contiguous pixels (block) from the row have approximately constant gray level value. Figure 1 shows details of this partitioning process. The final edge pixels map is simply the set of *M-1* change points produced by the optimal partitioning. A cost function, C(R_i), is associated with the i^{th} region or partition with the overall cost over the row *R* denoted as C(*R*) being the sum of the costs of the partitions. This can be expressed as follows:

$$C(R) = \sum_{i=0}^{M-1} C(R_i)$$

Jackson, Scargle and others [5], have recently developed a dynamic programming technique [7], that solves this problem in *O* (L^2) where L is the length of the signal (image row or column) to be partitioned. Associated with each partition is a cost function. The algorithm searches the exponentially large space of partitions of L data points in time *O* (L^2). The cost function required by the algorithm must be additive with the cost of the total signal being the sum of the costs of its subintervals or partitions. The algorithm is *guaranteed* to find the exact global optimum, automatically determines the model order (the number of thresholds) and has a convenient real-time mode. The algorithm is discussed in detail in the reference and we only give a summary of it within the context of our edge detection model:

Step 0: Decide prior, Ncp_Prior, and cost function C
Step 1: Set *optimal* (-1) =0; set n=0
Step 2: Given *optimal* (*j*) for j=0, 1, …, n

- Compute *optimal* (*n+1*) as given in [5], for j=0, 1,…, n+1
- Keep track of j, where maximum occurred as *lastChange*(n+1)
- Set n = n+1
- If n=L Stop

Step 3: Extract the set of M-1 edge locations as:

$$E_{M-2} = lastChange(L-1)$$
$$E_{M-3} = lastChange(E_{M-2}-1)$$
$$E_i = lastChange(E_{i+1}-1)$$
$$...$$
$$E_0 = lastChange(E_1)$$

As we can see from the third step, the edge locations are traced back in reverse order. Their number is different for each input, which is an image row, or a segment of it. These change points are essentially the edge pixels and every thing else is considered background.

2.2 Cost Function

The cost function model we use is adopted from [5], [6] and is based on the Bayesian posterior for a segmented Poisson model and is given by:

$$C\ (R_i) = \log\Gamma(G_i+1) - (G_i+1)\log(N_i+1),$$

Where:

$$G_i = \sum_{j\in R_i} g_j \ \text{ and }\ N_i = \sum_{j\in R_i} n_j$$

n_i is the number of pixels at gray level l_i, which we denote n_i, and g_i represents the total number of photons in R_i or $l_i n_i \equiv g_i$. The derivation of the model is given in [5]. One important observation for our thresholding problem is the fact that the cost function has the desired property of depending only on the sum of all the gray levels present in an interval and the number of cells in that interval which are called sufficient statistic. Additionally, this cost function was derived in the context of a data model representing the measured quantity as constant over each element of the partition—a piecewise constant model. Therefore, the model is basically identifying significant structures, i.e. regions, by their conformity to the piecewise constant model. We have experimented with other cost functions based on the Gaussian distribution and obtained similar results. We are also experimenting with other cost functions such as entropy and cross entropy for different applications.

3. Algorithm

Based on the image model, dynamic partitioning of an interval and the choice of a cost function, we introduce the edge detection algorithm. The algorithm works separately, and in an identical way, on each dimension (rows and the columns of the image.) For each dimension, we break each row into identical number of segments and process each segment separately. We then feed each segment to the dynamic partitioning algorithm

Following are the general steps for the algorithm

1. Break each row into identical segments
2. Feed each segment with the cost function to the optimal partitioning dynamic programming algorithm
3. Mark change points as edge points and others as background
4. Repeat 1-3 for the columns

Segmenting each dimension's data signal is only necessary to reduce the complexity of the algorithm and in off-line applications the whole row can be used as an input to the optimal partitioning algorithm. In addition to the image and its dimensions, the algorithm takes two more parameters: the prior and the cost function. The prior implicitly determines the number of classes with large values giving smaller number of edge pixels in the output image. The cost function can be modified to fit the application at hand. In addition to the Poisson cost function used in this paper, other cost functions based on Gaussian distribution or entropy can be used as well. Some of these cost functions have the advantage of incorporating local features from the image, albeit at higher computational cost.

4. SIMULATION RESULTS

In testing the effectiveness of our algorithm, we have obtained excellent results for many images of low and high details. The results of testing our algorithm on the Lena image are shown in Figure 2 for different priors as well as for a noisy version of the image. Each edge image is produced by running the algorithm on the image rows then the image columns (which can be done in parallel if needed). The algorithm produces the edge pixels, given here as white pixels, with the understanding that every other pixel is considered as background, given here as black pixels.The prior given to the algorithm is adjusted to give the desired level of edge details. We treated this prior as a parameter for the algorithm and allowed it to vary to fit the desired level of edge structure or details. It reflects our prior knowledge about the structure of the image at hand. Therefore, we can think of the prior as a means to produce, hierarchically, edge images at different scales of structure, while the image spatial dimensions or support are not changed. Qualitatively, increasing the prior acts as a penalty against fine image structures and therefore produces smaller number of edges by disregarding any less significant ones. Typical prior values used in the simulation were in the range of 8 to 64, values that were determined experimentally.

The algorithm has shown robustness to different levels and types of noise as well. Part (b) of Figure 2 shows a corrupted Lena image with Gaussian noise of zero mean and standard deviation of 10, variance of 100. Part (f) of the figure shows the result of our edge detection algorithm with a prior of 32. Evaluating the image subjectively, we can see that the algorithm succeeds in producing faithfully all the significant edges of the image. Some of the noise artifacts can easily be eliminated using post processing operations including morphological ones.

5. CONCLUSION

We presented a new edge detection algorithm based on dynamically partitioning the image rows or columns into regions of relatively uniform gray levels separted by edge or transition levels using an appropriate cost function. The algorithm extracts the edges as the statistically significant structures in the image. The method also incorporates a prior value that works as a penalty parameter that controls hierarchically the amount of details present in the image, thus allowing edge generation at different scales of structure while keeping the spatial dimensions of the image intact. Experimental results show excellent results for different cost functions, including the presented Bayesian posterior Poisson model, and robustness in the presence of low to mild amounts of Gaussian noise.

ACKNOWLEDGMENT

Part of this work was conducted at NASA Ames Research Center in Moffet Field, California, with funding from NASA through the National Association for Equal Employment in Higher Education, NAFEO. Their hospitality and support are greatly appreciated.

REFERENCES

[1] I. E. Abdou, and W. K. Pratt, "Quantitative design and evaluation enhancement/thresholding edge detectors," Proceedings of IEEE, vol. 67, pp. 753–763, 1979.

[2] T. Peli, and D. Malah, "A Study of Edge Detection Algorithms," Computer Graphics and Image Processing, vol. 20, pp.1-2, 1982.

[3] D. Marr, D. E. C. and Hildreth, "Theory of edge detection," Proc. Roy. Soc. London B, 207, pp. 187–217, 1980.

[4] J. Canny, "A computational approach to edge detection," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 679–698, 1986.

[5] B. Jackson, B., J. Scargle, et. al.: "An algorithm for optimal partitioning of data on an interval," IEEE Signal Processing Letters, vol. 12, pp. 105-108, 2005.

[6] J. Scargle, "Studies in astronomical time series analysis. V. Bayesian blocks, a new method to analyze structure in photon counting data, "The Astrophysical Journal, vol. 504, pp. 405-418, 1998.

[7] R. Bellman, Dynamic Programming, Princeton, University Press, Princeton, 1957.

[8] D. Mumford and J. Shah, “Optimal approximation by piecewise smooth functions and associated variational problems,” Comm. Pure Appl. Math, vol. 42, pp. 577–685, 1989.

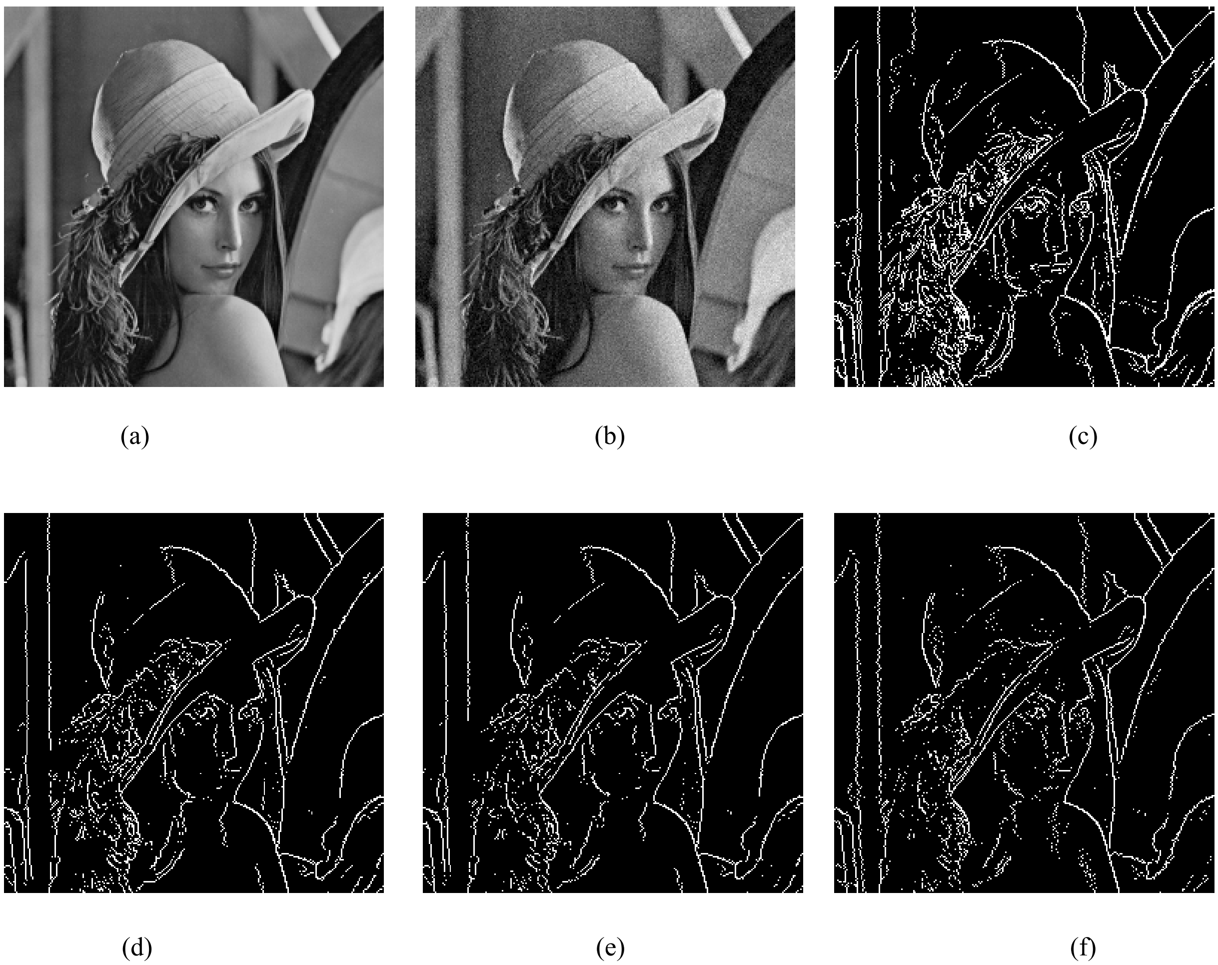

Figure 2. (a) Original Lena Image. (b) Original Lena Image with Gaussian noise of S.D. of 10.
(c) Edge image with Prior of 16 (d) Edge image with Prior of 32
(e) Edge image with Prior of 40 (f) Edge image of the Lena noisy image with Prior of 32