Darmanjian-Shaker Method (DSM)

Novel Edge Detector with Blurring

Shalom Darmanjian [1,2] oxygen@cnel.ufl.edu

Michael Nechyba [1,2] nechyba@mil.ufl.edu

A. Antonio Arroyo [1] arroyo@mil.ufl.edu

Eric M. Schwartz [1] ems@mil.ufl.edu

[1] Machine Intelligence Laboratory

[2] Computational NeuroEngineering Laboratory

Department of Electrical and Computer Engineering, University of Florida, Gainesville, Fl 32611-6200

ABSTRACT

Object recognition is an integral part of computer vision and often requires three stages of processing: (1) pre-processing (2) feature extraction and (3) training-recognition. In the first stage of pre-processing, edge detection is one of the most fundamental and often used techniques. For this paper, we propose a novel edge detector that incorporates a blurring effect; we label this approach the Darmanjian-Shaker-Method (DSM). The discussion is first motivated with a qualitative explanation of DSM and its origin. Following the motivation, we express DSM with a formal structure and relate it to a 2D convolution mask. Finally, using a subset of images we conclude the paper with a comparison of DSM results to the results of other well-known edge detectors (outlining benefits and disadvantages).

I. INTRODUCTION

Increasingly, intelligent and autonomous systems are employing computer vision and image-processing techniques in order to identify and recognize/classify objects. Incorporated within these classification systems are essential pre-processing and feature extraction components. These core techniques simplify the input image and assist the complex classifiers in their end goal of recognition. One of these fundamental techniques is edge detection.

Edge detection uses gradient profiles to isolate structures within an image. These structures in turn can be used for another pre-processing layer or final classification/recognition. Although there are many edge detection methods available, they all tend to produce similarly thin edges on the post-processed image. This often makes it necessary for the edges to be thickened or blurred so that more advanced methods can be applied. Unfortunately, this blurring or thickening normally occurs before/after the initial edge detection, adding processing time and complexity. Our internal object recognition-classification work led us to a simple computation (DSM) that can combine edge detection with blurring and mitigate some of these issues.

DSM was inspired while driving through a narrow congested urban street. We discovered that as people and cars darted in front of the car, our eyes would repeatedly dart from left to right. This horizontal 'shaking' motion seemed to allow our peripheral vision to render large moving objects quickly. Referring to medical literature, it is explained that the foveal (center) vision is optimized for acute details whereas the peripheral is optimized for coarser details (Figure 1) [1]. Therefore, by darting our eyes, we sacrifice acute resolution in order to see as many moving objects in the smallest amount of time.

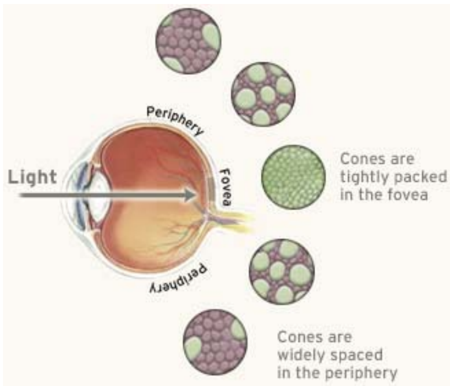

Figure 1: Tightly packed cones generate better resolution

In this paper, we show that 'image shaking' can render thicker edges because the image becomes blurred with the 'shaking' motion. We first examine this approach qualitatively and then move to express DSM with a formal structure. Finally, we conclude the paper with a comparison of DSM results to the results of other well-known edge detectors, outlining the benefits and the disadvantages.

II. Approach

In this section, we first give a qualitative understanding of how our method works. We then present DSM with a formal structure and relate it to a 2-D convolution mask, comparing its structure to that of other edge detectors.

2.1 Qualitative

Our method is similar to many other pixel operations used in the fields of computer vision and image processing. For each pixel going along the columns and rows of the image, we apply a series of mathematical operations. In particular, our method incorporates eight subtractions of neighboring RGB pixel values from a scaled version of a center RGB pixel value. To illustrate this succinctly, we use the following thought experiment: imagine taking an image and shaking it to produce eight images displaced around the original image. For example shaking it two pixels in eight directions, up, down, left, right, diagonal upper left, diagonal upper right, diagonal lower right diagonal lower left. Once you have these eight images, you then scale your original image RGB pixel-values by a constant, so that when you subtract your eight 'shaken' images, all pixels surrounded by the same RGB pixel values are removed (Figure 2).

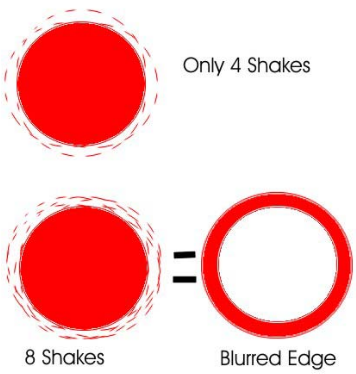

Figure 2: Subtracting displaced versions of the object showing 4, 8 and final blurred edge.

We note that 'image shaking' is analogous to the video processing technique of image subtraction[5], where a current image frame in the video stream is subtracted from the previous image frame in order to determine the movement difference between the images. This is similar to image shaking since with the artificially produced motion DSM also computes image subtractions to see what 'movement' difference exists between these 'shaken' images.

2.2 Quantitative

Now that we provided a qualitative overview of our simple method, we seek to express DSM with a more formal structure. Using the 'shaking' subtractions described in section 2.1, we can define our structure with a 2D convolution mask (see Figure 3 below).

-1	0	0	0	0	0	-1
0	0	0	-1	0	0	0
0	0	0	0	0	0	0
0	-1	0	8	0	-1	0
0	0	0	0	0	0	0
0	0	0	-1	0	0	0
-1	0	0	0	0	0	-1

Figure 3: DSM 2-D convolution mask.

In Figure 3, we observe that two regions of pixel neighbors are subtracted from the desired center pixel (inner four and outer four). We also see a scaling coefficient with a value of eight being used for the center pixel (discussed later). We first note the four diagonal outer subtractions (or negative one in Figure 3) are similar to the Robert cross operator. In the Robert cross operator, (Figure 4) two disjoint operations are applied to the desired pixel [3]. Specifically, the diagonals are subtracted from each other for both a horizontal and vertical operation. DSM has an advantage since it encompasses both horizontal and vertical components in one convolution mask as opposed to two.

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Figure 4: Robert Cross operator takes two iterations (horizontal and vertical)

Notice that the DSM inner region of pixel neighbor subtractions is very similar to the Laplace operator in which the second derivative is computed on the center pixel relative to the immediate neighbors (Figure 5) [4]. Note that the Laplace operator also uses a scaling coefficient to 'zero out' center pixels that have the same RGB values as its neighbors.

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

−1	2	−1
2	−4	2
−1	2	−1

Figure 5: Some common Laplace 2D Masks

Overall, we see that DSM combines some of the qualities of the Robert Cross operator and the Laplace operator to thicken edges with burring. Despite these incorporated qualities, DSM differs from both the Laplace and Robert cross operators because it uses pixels neighbors that farther away from the center-desired pixel which add to the blurring effect. DSM also differs from most of the other edge masks since they are usually application specific and target certain gradient patterns in the image, hoping to isolate particular features (Figure 6). DSM on the other hand, is more general and does not target specific gradient patterns but just provides an overall edge detection and thickening.

	0°			45°		
Sobel	-1	0	1	0	1	2
	-2	0	2	-1	0	1
	-1	0	1	-2	-1	0
Kirsch	-3	-3	5	-3	5	5
	-3	0	5	-3	0	5
	-3	-3	5	-3	-3	-3
Robinson	-1	0	1	0	1	1
	-1	0	1	-1	0	1
	-1	0	1	-1	-1	0

Figure 6: Other common 2-D convolution masks used in edge detection

III. Results

Although most edge detectors are application specific, we believe the majority of recognition tasks can benefit from edge thickening. Consequently, we use two dissimilar images to compare DSM to the other edge detection methods. One of the images is a difficult generic scene with a person and the other test image involves specific objects (sport balls) to be isolated. For reasons of space constraints, we do not go beyond these two sample image scenes.

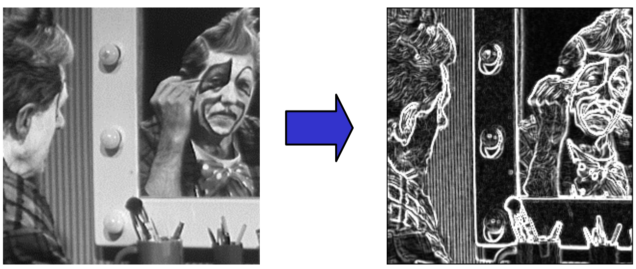

Figure 7: DSM Binary Output

In Figure 7 above, we display the results from using DSM. We notice that the edges are thicker than the other image outputs (Figures 8 and 9) and that we retained some of the textural information of the image. Despite the image being difficult, we are able to maintain most of the edge structure in the image.

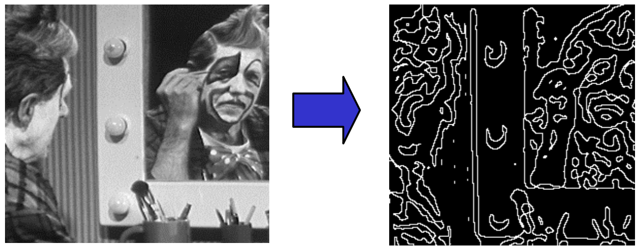

Figure 8: Laplace Binary Output

In Figure 8 above, we display the results from using the Laplace operator. We notice that the edges are thinner than the DSM output and that we lose much of the textural information in the image. We also notice that some of the thin lines on the wall and on the clown's shirt are removed entirely.

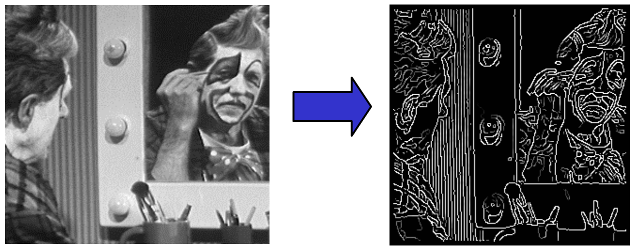

Figure 9: Sobel Operator Binary Output

In Figure 9 above, we display the results from using the Sobel operator. We notice that the edges are also thinner than the DSM output and that we also lose much of the textural information in the image. In this output though, we retain some of the wall lines and other features that are lost with the Laplace operator.

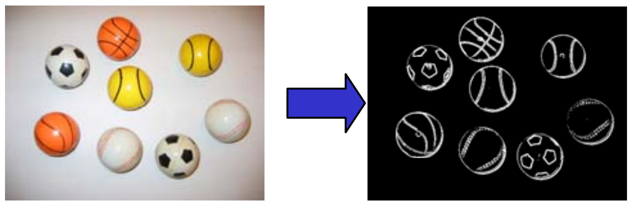

Figure 10: Sobel Operator Binary Output

In Figure 10 above, we display the results from using Sobel operator on a simple object recognition task. We notice that the edges are thinner than the DSM output, which could pose problems in later processing. Figure 11 is another example of how DSM does not produce thin edges.

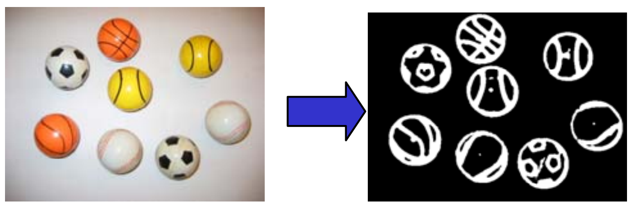

Figure 11: DSM Binary Output

IV. CONCLUSION

The results from section 3 show that DSM provides superior information for classification systems that only require a rough segmentation of an image to train and classify. In other niche applications, DSM may provide too 'rough' of an image segmentation for the classifier to work reliably, especially if the classifier is easily prone to noise or the object recognition must be extremely pixel accurate. To work around this, DSM could use closer pixel neighbors to decrease the blurring effect. Overall, DSM essentially computes a row, column, and blurring operation with a single convolution mask. This makes DSM more efficient, more convenient, and easier to understand and implement.

REFERENCES

[1] J. R. Boyle and R. Thomas Computer Vision: A First Course, Blackwell Scientific Publications, 1988, pp 50 - 51.

[2] E. Davies *Machine Vision: Theory, Algorithms and Practicalities* Academic Press, 1990, Chap 5.

[3] L. Roberts *Machine Perception of 3-D Solids*, Optical and Electro-optical Information Processing, MIT Press 1965.

[4] D. Vernon *Machine Vision*, Prentice-Hall, 1991, Chap 5.

[5] S. Darmanjian, A. Arroyo and E. Schwartz, " An Alternative Real-Time Image Processing Tool, ," 2003 Florida Conference on Recent Advances in Robotics, Dania Beach, May, 2003.