A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms

Michael D. Heath, Sudeep Sarkar, *Member, IEEE Computer Society,*
Thomas Sanocki, and Kevin W. Bowyer, *Senior Member, IEEE*

Abstract—A new method for evaluating edge detection algorithms is presented and applied to measure the relative performance of algorithms by Canny, Nalwa-Binford, Iverson-Zucker, Bergholm, and Rothwell. The basic measure of performance is a visual rating score which indicates the perceived quality of the edges for identifying an object. The process of evaluating edge detection algorithms with this performance measure requires the collection of a set of gray-scale images, optimizing the input parameters for each algorithm, conducting visual evaluation experiments and applying statistical analysis methods. The novel aspect of this work is the use of a visual task and real images of complex scenes in evaluating edge detectors. The method is appealing because, by definition, the results agree with visual evaluations of the edge images.

Index Terms—Experimental comparison of algorithms, edge detector comparison, low level processing, performance evaluation, analysis of variance, human rating.

1 Introduction

The continued development of edge detectors is producing increasingly complex edge detection algorithms. While the field has come a long way since the algorithms of Roberts [1] and Sobel [2], there is a belief that the "increased sophistication (of newer algorithms) is not producing a commensurate improvement in performance" [3]. This conjecture, however, is hard to confirm or disprove since most of the published methods for evaluating edge detectors have not been widely accepted by researchers in the edge detection community. This is evident from their limited application in publications. Table 1 lists 21 algorithms published in three major journals within just the last three years. None of these give any objective performance comparison.

This paper presents a new edge detector evaluation method that was motivated by three ideas. The first idea is that a comparison of edge detectors should be done using real images. This was noted by Zhou et al. [4]: "Any conclusions based on these comparisons of synthetic images have limited value. The reason is that there is no simple extrapolation of conclusions based on synthetic images to real images!" The second idea is that an evaluation method should produce results that correlate with the perceived quality of edge images. This was noted by Cinque et al. [5]: "Although it would be nice to have a quantitative evaluation of performance given by an analytical expression, or more visually by means of a table or graph, we must remember that the final evaluator is man and that his subjective criteria depend on his practical requirements." The third idea is that edge detectors should be evaluated within a vision system performing a task. This was expressed by Cinque et al. [5], "We strongly appreciate the attempt to characterize the quality of an image processing system independently from the task it is performing, and, as mentioned above, we realize that many difficulties in achieving such a goal may be encountered. We believe that we still have a long way to go and therefore must now principally rely on human judgment for obtaining a practical evaluation; for some specific applications we feel that this is doomed to be the only possibility."

Assessing the performance of edge detection algorithms is difficult because the performance depends on several factors. At a minimum, these are:

1) the algorithm itself,
2) the type of images used to measure the performance of the algorithm,
3) the edge detector parameters used in the evaluation, and
4) the method for evaluating the edge detectors.

The approach taken to evaluate edge detectors in this work was to measure their performance using a general purpose evaluation function, to use real images in the evaluation and to select the parameters for each algorithm in a meaningful way that was not biased towards any algorithm. The human visual system was selected for the evaluation function because at the present time it is the most general purpose vision system.

The method presented in this paper is a refinement of a method previously introduced in [6]. In that work, the relative performance of the Sobel (with added hysteresis), Nalwa-Binford [8], Canny [9], and Sarkar-Boyer [10] edge detectors were measured using eight images. This paper expands on that initial study:

- *M.D. Heath, S. Sarkar, and K. Bowyer are with the Department of Computer Science & Engineering, University of South Florida, Tampa, FL 33620. E-mail: {heath, sarkar, kwb}@csee.usf.edu.*
- *T. Sanocki is with the Department of Psychology, University of South Florida, Tampa, FL 33620. E-mail: sanocki@chuma.cas.usf.edu.*

Manuscript received 17 Dec. 1996; revised 28 Aug. 1997. Recommended for acceptance by V.S. Nalwa.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 105704.

TABLE 1
RECENTLY PUBLISHED EDGE-DETECTION ALGORITHMS

Source	Nature of the algorithm	Performance presented on	Real image ground truth	Algorithms compared
[11](PAMI, 1995)	Logical/Linear	2 real	0	Canny
[12](PAMI, 1995)	covariance models	3 real	0	none
[13](PAMI, 1994)	expansion matching	1 real	0	Canny
[14](PAMI, 1993)	dispersion of gradient direction	1 real	0	Sobel
[15](PAMI, 1993)	regularization	2 real	0	LoG, Canny
[16](CVGIP, 1994)	voting based	3 real, 3 range, 2 synth	0	Canny
[17](CVGIP, 1994)	linear filtering	1 real, 1 synth	0	LoG
[18](PR, 1995)	filtering	1 synth, 1 real	0	zero-crossing
[19](PR, 1995)	statistical	1 synth, 3 real	0	Sobel
[20](PR, 1995)	filtering	7 synth, 1 real	0	none
[21](PR, 1995)	filtering	4 real	0	none
[22](PR, 1995)	statistical	4 real	0	Canny, LoG
[23](PR, 1995)	search	1 synth, 3 real	0	Canny, LoG, Ashkar&Modestino
[24](PR, 1995)	filtering	4 real	0	none
[25](PR, 1994)	neural nets	1 synth, 1 real	0	Canny
[26](PR, 1994)	genetic opt.	1 synth, 1 real	0	simulated anneal local search
[27](PR, 1994)	co-occurrence	4 synth, 2 real	0	Canny Jain's stochastic
[28](PR, 1994)	statistical	1 synth, 1 real	0	Sobel, DoG, Haralick, Anisotropic diffusion
[29](PR, 1993)	local masks	2 synth, 2 real	0	other hierarchical
[30](PR, 1993)	filtering	1 real	0	none
[31](PR, 1993)	statistical	3 real	0	Nalwa, DoG

Edge detection algorithms in PAMI, CVGIP: Image Understanding (renamed Computer Vision and Image Understanding in January 1995) and PR from 1993 through 1995. The number of images corresponds to the images presented in the paper. Ground truth is counted as objective specification of correct edge pixels. The last column lists the edge algorithms considered in the comparison of algorithms. Note that the Canny edge detector is the one most frequently used for comparison in the papers presenting new algorithms.

1) to compare a broad sampling of recently proposed and classic edge detection algorithms,
2) to use a larger sample of images,
3) to improve the method of selecting input parameters, and
4) to examine how the relative performance of the algorithms might differ between manmade or natural and textured or nontextured images.

The paper is organized as follows. Related work is reviewed in Section 2. The images used for the evaluation are presented in Section 3. The edge detectors that were compared are described in Section 4. The methods used in selecting parameters for the algorithms and the resulting parameter selections are described in Section 5. A description of the evaluation experiment and analysis of the results are in Section 6. A discussion of the methods and results is presented in Section 7.

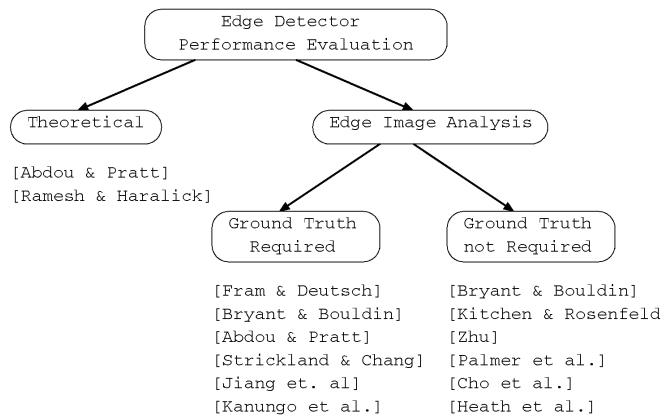


Fig. 1. Published methods for edge detector performance evaluation.

2 RELATED WORK

A variety of methods have been proposed for assessing the performance of edge detectors. These methods can be categorized as shown in Fig. 1. At the highest level, they can be categorized according to whether they employ a theoretical analysis or an experimental analysis of the edge pixels produced by an algorithm. Edge image analysis methods can be further categorized by whether or not they require "ground truth" locations of the "true" edges.

2.1 Theoretical Evaluation

A theoretical evaluation is done by applying a mathematical analysis without the algorithm(s) ever being applied to an image. Instead, the input to the algorithm is mathematically characterized and the performance is determined analytically or by simulation. Abdou and Pratt [32] and Ramesh and Haralick [33] developed evaluation measures of this type. The major limitations of these methods are the simplistic mathematical models used to characterize input signals and noise, and the difficulty in

TABLE 2
SUMMARY OF EDGE DETECTION EVALUATION METHODS

Authors	Image Type	Images Used	Requires Ground Truth	Comments
Fram & Deutsch [34] 1975	Synth.	1 Synth. 1 Real	Yes	Vertical step edge image. Real edge images not evaluated.
Bryant & Bouldin [35] 1979	Real	1 Real	Yes and No	Subjective ground truth specified for a single edge by hand.
Abdou & Pratt [32] 1979	Synth.	1 Synth. 3 Real	Yes	Horizontal, vertical and diagonal step edges used.
Kitchen & Rosenfeld [39] 1981	Real	2 Synth.	No	Method demonstrated using only synthetic images.
Ramesh and Haralick [33] 1992	Synth.	1 Synth. 2 Real	Yes	Synthetic images were generated for a ramp edge embedded in noise.
Strickland & Chang [36] 1993	Synth.	1 Synth.	Yes	Adaptable metric that is difficult to use with inclined or curved edges.
Jiang et al. [37] 1995	Real	80 Real (Range)	Yes	The use of simple scenes allowed accurate ground truth by hand.
Kanungo et al. [38] 1995	Synth.	1 Synth.	Yes	Vertical edge with added square wave noise. Detection task was used.
Cho et al. [42] 1996	Real	1 Real	No	Evaluates edge detectors that rely on the same edge model.
Heath et al. [6] 1996	Real	8 Real	No	Performance evaluated using complex scenes and an object recognition task.
Zhu [40] 1996	Real	2 Synth. 2 Real	No	The method is very similar to the one used in [39].
Palmer et al. [41] 1996	Real	1 Synth. 5 Real	No	Edge detection is evaluated within a line detection system.

This table lists the papers that have presented methods for doing edge detection evaluation. The image type describes the type of images used by the evaluation method. The number of images used was measured by counting the number of real images and the number of synthetic images presented in the paper. In the case of synthetic images, the images that were derived by manipulating a synthetic image were not counted as separate images.

applying such methods to many of the more modern edge detectors because of the complexity of their algorithms.

2.2 Evaluation Using Ground Truth

The basic idea of these approaches is to measure the difference between the detected edges and the ground truth. Fram and Deutsch [34] did this by measuring the ratio of the number of correctly detected edge pixels divided by the number of detected edge pixels and the fraction of a line "covered" by edge pixels. Abdou and Pratt [32] formulated a figure of merit that incorporated the displacement of detected edges from their true location. Bryant and Bouldin [35] measured the correlation between the detected edges and the ground truth using a method termed absolute grading. Strickland and Chang [36] described a metric for calculating an edge quality score as linear combination of individual measures of edge continuity, smoothness, thinness, localization, detection and noisiness. Jiang et al. presented a method [37] for evaluating edge detectors using sixteen performance measures related to correct edges, spurious edges and missing edges in range image data using hand specified ground truth. Finally, Kanungo et al. [38] presented a method for evaluating edge detectors within a system that did edge detection followed by line detection.

The limitations of these methods are that they depend on ground truth so they either rely on synthetic images or on simple real images for which it is relatively easy to specify the ground truth. Neither type of image captures the complexity of real scenes so none of these methods measure the performance of edge detectors under realistic conditions.

2.3 Evaluation Without Ground Truth

Kitchen and Rosenfeld [39] evaluated edge detectors using edge coherence, which measures the continuation and thinness of the detected edges. Bryant and Bouldin [35] used synthesized ground truth, obtained from a consensus decision from a suite of edge detection algorithms, to evaluate edge detectors. Zhu [40] developed a different way to compute and express the method of local edge coherence developed by Kitchen and Rosenfeld [39]. Palmer et al. [41] developed a method for evaluating a system that performs edge detection followed by line detection. A performance measure was developed that computes a nonlinear combination of the support for detected lines. Cho et al. [42] applied bootstrapping to measure the performance of an edge detection algorithm.

All these methods evaluate either the form of the edges, the likelihood of a detected edge being a true edge given the local edge pixel intensities, or the similarity of the detected edges with a synthetic "ground truth." The principal limitation of these methods is that they cannot measure the displacement of the edges from their true locations. Therefore, they do not suitably capture a necessary component of the quality of the edges for doing any task. For this reason, modern edge detectors that blur the image before detecting the edges can score very highly with these measures by producing very distorted edges with good form, but that are not usable for any task.

2.4 The Proposed Method

The emphasis in [6] and in this work, is that real images of common scenes should be used in the evaluation of edge detection algorithms. Most methods for evaluating an edge

detector rely on the specification of ground truth. This is very difficult for real images of common scenes. While it is possible for people to label some of the edges in an image, the difficulty in labeling all of the edges is clear to anyone who has tried to do it. For example, how does one label the edges on a tree or the edges on a desk with wood grain?

Admittedly, the methods proposed in [39], [35], [40], [41], and [42] can operate on real images and do not required ground truth. Unfortunately, each of these methods leave something to be desired. The methods in [39] and [40] do not consider the displacement of detected edges from their true locations. The method in [35] can punish an edge detector for detecting correct edges if they were missed by the majority of the edge detectors used to establish the ground truth. The method in [41] can only be applied to straight line edges. Finally, the measured performance by the method in [42] depends on the choice of the algorithm for establishing the support for perturbing the input.

Our proposed evaluation method relies on the subjective evaluation of edge images by people. This removes the need for explicitly specifying the ground truth. Hence it allows the use of complex real images in the evaluation that would not be usable by other methods because the ground truth would be too difficult to specify.

3 Image Selection and Categorization

A set of images was collected that contained objects that people could readily recognize and name. To ensure wide variety, objects were categorized as manmade or natural and as textured or nontextured, and images of each type were collected. Images were obtained by photographing common objects in their natural settings. All photographs were taken with a 35-mm camera on color negative film using a 50-mm lens. The images were then scanned onto PhotoCD by a commercial lab and were then extracted from the CD in the 768×512, 24 bit/pixel format. Each image was then converted to gray scale by combining the color planes in a ratio of 0.299 RED + 0.587 GREEN + 0.114 BLUE. The gray scale images were then cropped to obtain images nearly 512×512 in size, in which the object of interest was clearly in the center of the image. Finally, the images were scaled to adjust the brightness and contrast to look good on a computer monitor.

The images were screened to be sure that people could recognize the central object in each photograph. This was done because the object recognition task to be performed with the edge images would not be meaningful if people could not recognize the object in a photograph.

Each image was then labeled as manmade or natural and as textured or nontextured, according to the properties of the central object. While no claim is made that the images in each category are representative of the corresponding class of images, the categorization was done to allow a check to be done on the consistency of the results across meaningful subsets of the images.

All together, 28 images were selected for use in the evaluation of the edge detectors. Twenty of these images were categorized as manmade versus natural and as textured versus nontextured with five images in each category. Eight of the 28 images were used in [6] and were carried forward to these experiments. Fig. 2 shows the 28 images.

4 Edge Detection Algorithms

Three criteria for selecting edge detectors were:

1) to include a diverse mix of algorithms including representatives of the state of the art in edge detection,
2) to evaluate only edge detection algorithms that had been presented to the vision community through a refereed publication, and
3) to evaluate only algorithms for which code was readily available.

Based on these criteria, algorithms by Canny [9], Nalwa [8], Iverson [11], Bergholm [43], and Rothwell [44] were selected. The choice to include only five algorithms was made to keep the experimental comparisons within a reasonable size.

Whenever possible, an implementation of an algorithm was obtained from the authors who developed it. Ideally, our objective was not to modify the code we received. Unfortunately, this was necessary in several cases. Some reasons for doing this were that several algorithms did not output edges in a binary image format and we decided to add non-maximal suppression or hysteresis thresholding. Every program, except the Bergholm edge focusing program, was modified to a small degree. Strictly speaking, the results of the performance evaluation should be attributed to the implementation of the algorithm that was used. We take full responsibility for all of the changes we made to the algorithms.

4.1 Canny Algorithm

The Canny edge detection algorithm is considered a "standard method" used by many researchers. Canny edge detection uses linear filtering with a Gaussian kernel to smooth noise and then computes the edge strength and direction for each pixel in the smoothed image. This is done by differentiating the image in two orthogonal directions and computing the gradient magnitude as the root sum of squares of the derivatives. The gradient direction is computed using the arctangent of the ratio of the derivatives. Candidate edge pixels are identified as the pixels that survive a thinning process called nonmaximal suppression. In this process, the edge strength of each candidate edge pixel is set to zero if its edge strength is not larger than the edge strength of the two adjacent pixels in the gradient direction. Thresholding is then done on the thinned edge magnitude image using hysteresis. In hysteresis, two edge strength thresholds are used. All candidate edge pixels below the lower threshold are labeled as nonedges and all pixels above the low threshold that can be connected to any pixel above the high threshold through a chain of edge pixels are labeled as edge pixels. The implementation of the Canny edge detector used was originally written at the University of Michigan. Parts of the code were rewritten to have better structure and to allow the processing of nonsquare images.

Fig. 2. The images that were used to evaluate the edge detectors. (continued on next page)

The Canny edge detector allows the user to specify three parameters. The first is *sigma*, the standard deviation of the Gaussian filter specified in pixels. The second parameter, *low*, and the third parameter, *high*, are, respectively, the low and high hysteresis thresholds. The high threshold is a fraction of the graient magnitude and the low threshold is a fraction of the calculated high threshold value.

4.2 Nalwa Algorithm

The Nalwa edge detection algorithm represents the method of edge detection by surface fitting. It differs from the linear filtering approach used in the Canny edge detector because the derivative of the image is not computed. Instead, hyperbolic tangent and quadratic functions are fit to the image intensities in a 5×5 pixel window that is scanned across the image. If the hyperbolic tangent fit has a lower error than the quadratic fit, then a candidate edge is marked. The contrasts of the candidate edge pixels are then thresholded to reduce the number of spurious edges.

The implementation of the algorithm was obtained from Vic Nalwa. The output format was originally a text file that

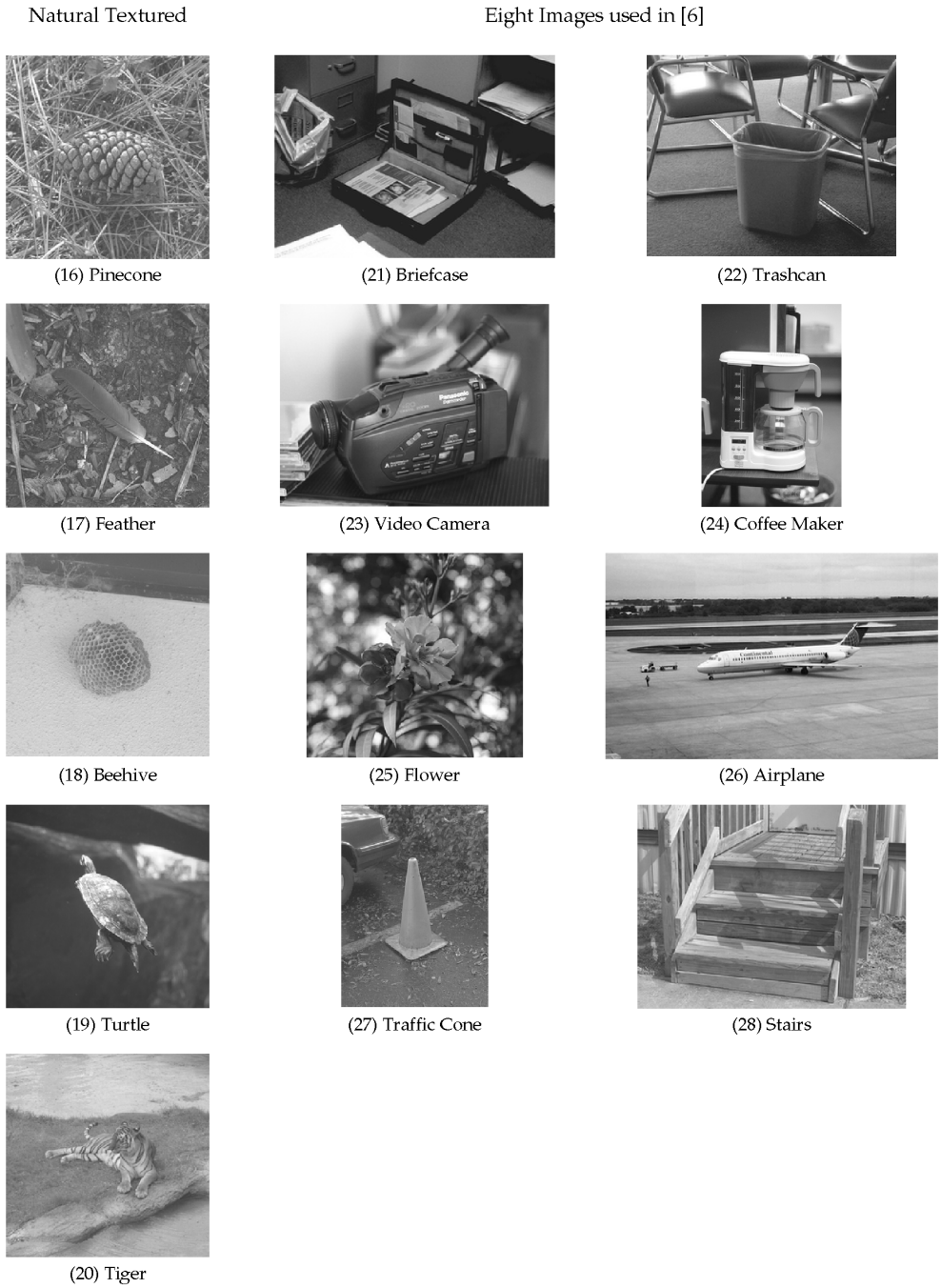


Fig. 2. (continued)

indicated the subpixel location of each edgel (edge element) as well as the edge contrast and orientation. To use our evaluation methodology, it was necessary to modify the algorithm to produce an edge image by plotting the edge points on a grid with the dimensions of the image. The edge images produced by the modified algorithm were in some cases more than one pixel thick. To take care of this, the nonmaximal suppression algorithm from the Canny edge detector was added to thin the edges. Twice the edge contrast was used in place of the gradient magnitude in this process to reduce the noise from quantization. Hysteresis thresholding was added following the suggestion of Vic Nalwa.

The original Nalwa edge detector allows the user to specify one parameter, but the modified program required the user to specify three parameters. The range used for the *blur* parameter was 0.6 to 1.5 (determined from a recommendation from Vic Nalwa). The values of the hysteresis thresholds, *low* and *high*, were obtained through experimentation.

4.3 Iverson Algorithm

The Iverson logical/linear edge detector was included in the evaluation because it presented a method to improve the performance of a linear edge detection algorithm by

including logical checks for the existence of an edge. The motivation for doing this was to reduce the number of false positive edges detected with linear edge detectors without losing sensitivity in detecting true edges.

The edge detector implementation (version 1.0.3) was down loaded from an FTP site (`ftp://ftp.cim.mcgill.ca/pub/people/leei/loglin.tar.gz`). The program itself was not modified at all, but post processing was applied to the output. This was done because the algorithm outputs a postscript file of edge segments plotted on a grid with a resolution higher than the original image. Since we required all of the algorithms to represent the edges as a binary image with the dimensions of the original image, each edgel was plotted as a pixel in this image. Since the algorithm initially allowed for detecting multiple edgels at the same position with different orientations, the edge direction was taken to be the direction with the largest associated edge strength. Nonmaximal suppression was added to ensure that thin edges were produced. Hysteresis was added to allow greater flexibility for tuning the algorithm to each image. This edge detector is capable of separately detecting step edges and both positive and negative contrast lines. We search for only step edges in this evaluation.

The modified Iverson-Zucker algorithm allowed the user to specify three parameters. Default values were specified for the degree and the threshold. The *direction* parameter controls the number of directions considered to detect edges. This parameter was set at four values between four and ten. The *low* and *high* hysteresis values were obtained through experimentation. To consider the case of no hysteresis thresholding, we included the choices of zero *low* and *high* thresholds in the parameter set.

4.4 Bergholm Algorithm

The Bergholm edge focusing algorithm was selected because it represented an approach that used a scale space representation to try to find edges that are "significant." Edges are first detected at a coarse resolution. This is done by blurring the image with a Gaussian filter, and finding the pixels that have gradient that is both a local maximum and that is greater than a threshold value. The algorithm then "focuses" these edges by tracking them through scale space to finer resolutions (images that were smoothed with small Gaussian filters). Edges at coarse scales were used to guide the search for edge pixels at successive fine scales.

The implementation of the Bergholm detector was obtained as part of the Candela image processing package obtained by anonymous FTP (from `ftp.bion.kth.se/cvap/2.1`). The program was not changed at all. The edge images were remapped to display the edges in black on a white background.

The algorithm required three parameters to be set; the *starting sigma*, the *ending sigma* and an edge *threshold*. The range of values for the *starting sigma* and the *ending sigma* were 5 to 0.5. This is consistent with the parameter values in the journal paper that presented the algorithm [43]. The range of values for the *threshold* was 5 to 20 and was determined through experimentation.

4.5 Rothwell Algorithm

The last algorithm included in the experiment was unique in that it employed dynamic thresholding that varied the edge strength threshold across the image. Overall, the algorithm was very similar to the Canny algorithm because Gaussian smoothing was followed by differentiation. The difference between the two algorithms was that the Rothwell algorithm does edge thinning as a post edge detection process and that dynamic thresholding is used instead of hysteresis. The reason for not using nonmaximal suppression to do the thinning was a claim that it fails at the junctions in images because of the smoothing process. The reason for not using hysteresis was a belief that the strength of an edge has no particular relevance to its value for higher-level vision processing (such as object recognition). The implementation of this algorithm was performed by combining pieces of the Canny edge detector code and pieces of C++ code obtained from the authors of the paper [44].

This algorithm required that the user input three parameters. These are the smoothing amount *sigma*, the edge *threshold* and a parameter *alpha* that adapts the edge threshold to increase the detection of pixels that are near other edges. The range of values for *sigma* was 0.5 to 2.0 pixels. The value of *alpha* was set between 0.8 and 0.95, to include the value of 0.9 used in [44].

5 Parameter Selection

5.1 Overview

Selecting the input parameters of each algorithm is a critical step in edge detector performance evaluation because the resulting edge quality varies greatly with the choice of parameters. In this evaluation, the input parameters were selected to maximize the quality of the edges for the purpose of recognizing an object in the image. We devoted equal effort in searching the parameter space for each algorithm. While this method does not guarantee that the optimal input parameter set was identified,[1] it does avoid biasing the results toward any of the algorithms.

Parameter selection involved multiple steps. We began by identifying a large, fixed number of parameter combinations. We then had an individual, the parameter prescreener select a subset of parameters. This subset of parameters was then reduced to the final parameter set using visual ratings collected in an experiment.

5.2 Initial Parameter Specification

The initial parameter specification involved selecting 64 sets of parameters that were consistent with the parameters used by the respective authors. The objective of this step was to select a range of parameters that samples the space broadly enough for each parameter without sampling it too coarsely. These parameters sets were then used to generate 64 edge images.

The initial 64 Canny parameters were all combinations of *sigma*, *low*, and *high*, where *sigma* $\in$ {0.60, 1.20, 1.80, 2.40}, *low* $\in$ {0.20, 0.30, 0.40, 0.50}, and *high* $\in$ {0.60, 0.70, 0.80, 0.90}. The initial 64 Nalwa parameters were all combinations of *blur*, *low*, and *high*, where *blur* $\in$ {0.60, 0.90, 1.20, 1.50}, *low* $\in$ {0.05,

1. There is currently no accepted method that will guarantee finding the optimal input parameters without ground truth.

TABLE 3
THE PARAMETERS THAT WERE USED IN THE PARAMETER SELECTION EXPERIMENT

Canny Edge Detector												
	Combination Number											
Parameter	1	2	3	4	5	6	7	8	9	10	11	12
sigma	1.2	1.8	0.6	1.2	0.6	1.2	1.2	1.2	2.4	0.6	1.2	1.8
low	0.4	0.2	0.3	0.2	0.5	0.3	0.4	0.2	0.2	0.3	0.4	0.3
high	0.8	0.7	0.9	0.6	0.9	0.8	0.6	0.8	0.6	0.8	0.9	0.9

Nalwa Edge Detector												
	Combination Number											
Parameter	1	2	3	4	5	6	7	8	9	10	11	12
blur	1.50	1.50	0.60	1.20	1.50	0.60	1.50	1.20	1.20	1.50	0.60	1.50
low	0.10	0.20	0.15	0.10	0.15	0.10	0.15	0.05	0.05	0.05	0.05	0.05
high	0.60	0.60	0.60	0.60	0.15	0.60	0.45	0.15	0.30	0.45	0.30	0.15

Iverson Edge Detector												
	Combination Number											
Parameter	1	2	3	4	5	6	7	8	9	10	11	12
directions	8	8	8	4	8	8	4	6	8	8	6	10
low	0.00	0.20	0.00	0.20	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00
high	0.55	0.05	0.00	0.55	0.05	0.30	0.05	0.00	0.30	0.55	0.30	0.05

Bergholm Edge Detector												
	Combination Number											
Parameter	1	2	3	4	5	6	7	8	9	10	11	12
start sigma	2.0	2.0	3.0	3.0	2.0	2.0	3.0	4.0	4.0	3.0	4.0	5.0
end sigma	2.0	1.5	1.5	2.0	1.0	2.0	2.0	1.5	2.0	2.0	1.5	1.5
threshold	20	15	10	20	15	15	5	20	10	15	5	10

Rothwell Edge Detector												
	Combination Number											
Parameter	1	2	3	4	5	6	7	8	9	10	11	12
sigma	1.0	1.0	1.0	1.5	1.0	1.5	1.5	2.0	0.5	1.0	1.0	1.5
threshold	8	8	13	8	13	3	3	3	18	18	18	13
alpha	0.90	0.95	0.85	0.85	0.80	0.90	0.80	0.95	0.80	0.80	0.85	0.80

These tables list 12 parameter combinations that were chosen for each edge detector. These parameters were obtained by evaluating 64 parameters combinations for each picture and then selecting a subset of 12 parameter combinations that provided at least one good edge image for each picture.

0.10, 0.15, 0.20}, and *high* ∈ {0.15, 0.30, 0.45, 0.60}. The initial 64 Iverson parameters were all combinations of *directions*, *low*, and *high*, where *directions* ∈ {4, 6, 8, 10}, *low* ∈ {0.00, 0.20, 0.40, 0.60}, and *high* ∈ {0.05, 0.30, 0.55, 0.80}. Note: The combinations containing both *low* = 0.60 and *high* = 0.80 were replaced with *low* = 0.00 and *high* = 0.00. The initial 64 Bergholm parameters were all combinations of *start sigma*, *end sigma*, and *threshold*, where *start sigma* ∈ {2.0, 3.0, 4.0, 5.0}, *end sigma* ∈{0.5, 1.0, 1.5, 2.0}, and *threshold* ∈{5.0, 10.0, 15.0, 20.0}. The initial 64 Rothwell parameters were all combinations of *sigma*, *threshold*, and *alpha*, where *sigma* ∈ {0.50, 1.00, 1.50, 2.00}, *threshold* ∈ {3.0, 8.0, 13.0, 18.0}, and *alpha* ∈ {0.80, 0.85, 0.90, 0.95}.

5.3 Parameter Prescreening

5.3.1 Methodology

A person, the parameter prescreener, viewed the 64 edge images and selected the best five for each gray-scale image. After doing this, the results were input to a greedy search algorithm that selected a subset of 12 of the 64 parameter combinations to use in a final parameter selection experiment. The objective of the greedy search was to find a subset of the parameters that produced at least one good edge image for each gray-scale image. More specifically, at each step, the search maximized the minimum number of images that were selected by the parameter prescreener across the set of gray-scale images.

5.3.2 Results

The parameter prescreening process reduced the 64 parameter sets to 12 parameter sets for each algorithm. The results are listed in Table 3.

5.4 Parameter Selection Experiments

5.4.1 Methodology

One parameter selection experiment was conducted for each edge detector. In these experiments, participants evaluated sets of edge images created from 28 gray-scale images. Each set consisted of the gray-scale image and 12 edge images created from it by the same algorithm using different input parameters. Nine students from a graduate computer vision class volunteered to participate in the experiments and gave their informed consent.

The participants were verbally instructed to rate each edge image according to how well they thought they could recognize the central object from the edges. The ratings were recorded on a scale of one to seven. A score of seven indicated that the "Information allows for easy, quick and accurate recognition of the object," and a one indicated that there was "No coherent information from which to recognize the object." Intermediate numbers indicated intermediate ratings. An example edge image with the rating scale can be seen in Fig. 3.

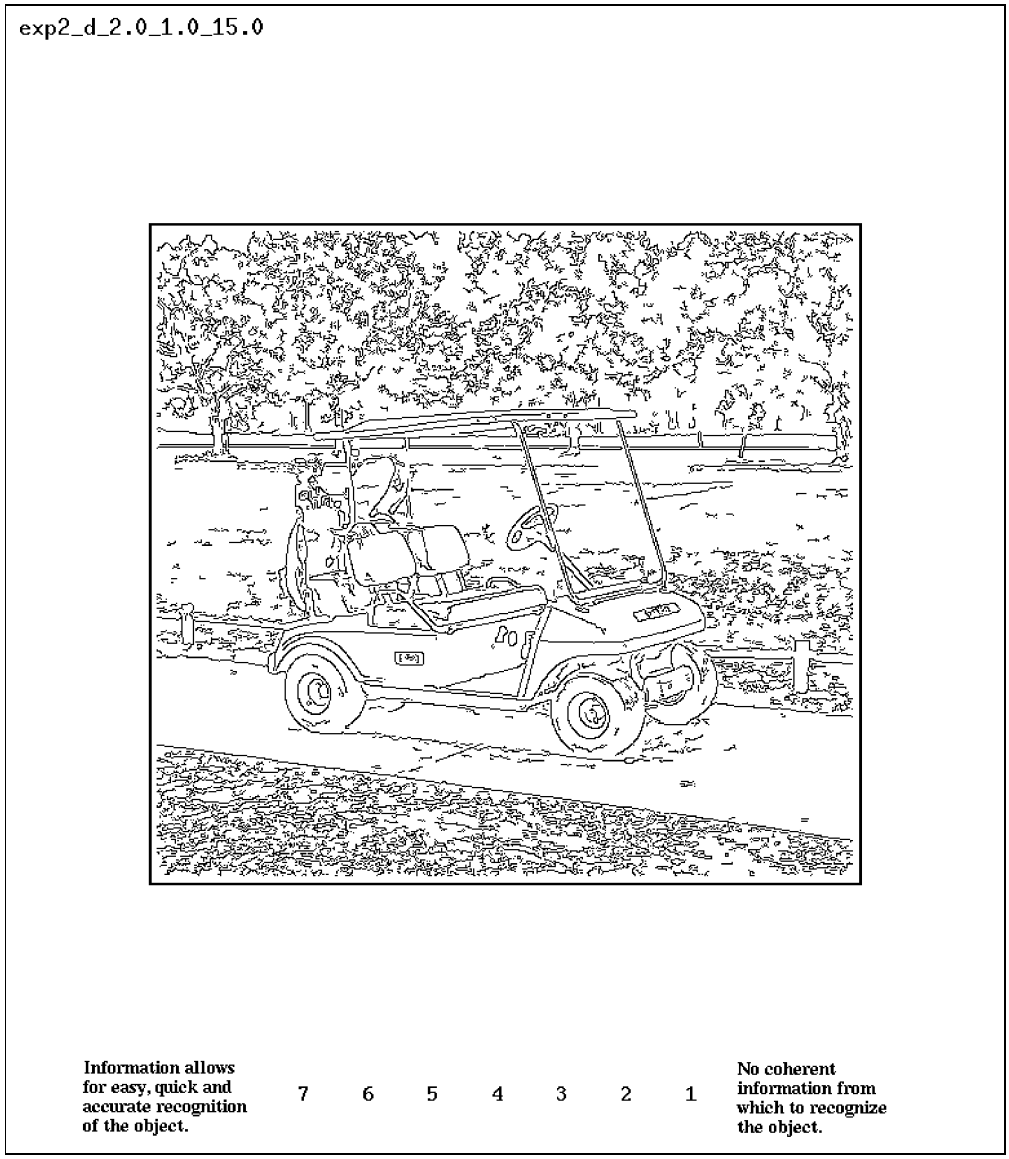


Fig. 3. An example evaluation sheet. The label in the upper left corner is a coded identifier for this image. The edges appear in black on a white background. The rating scale appeared on each evaluation sheet.

Each participant had a table on which to spread out the evaluation sheets for each image. During the experiment, they were allowed to rearrange the sheets to allow for side by side comparisons between edge images. This also allowed the participants to sort the images by edge quality if they wanted to.

There was no time limit for the experiments. Individual times varied between one and two hours. The experiment was repeated for each of the edge detectors by the same nine students with two to three weeks between each experiment.

5.4.2 Results

Fig. 4 illustrates the rating scale using some of the ratings collected in the experiment. It shows the 12 edge images generated by the Canny algorithm from the tire image using the parameters in Table 3. Below each image, the average rating calculated from the nine participants' responses is shown. It is easy to see the that the ratings track the quality of the edge images.

Consistency of the Participants' Ratings. To establish the validity of comparing mean ratings, it is important to know whether the ratings were consistent across the participants. This was estimated using one form of the Intraclass Correlation Coefficient [45]. The ICC(3, k) form of this statistic is appropriate for estimating this because it measures the consistency in the participants' mean rating of a particular parameter settings edge image to the overall mean of the edge images for that edge detector. The ICC(3, k) is defined as:

$$\mathrm{ICC}(3,k) = \frac{BMS - EMS}{BMS}$$

where BMS is the mean square value of the rating between targets, EMS is the total mean square error, and k is the number of judges. The values of the ICC can range from zero (no consistency) to one (complete consistency). The SAS statistical package was used to compute the components of the ICC statistic.

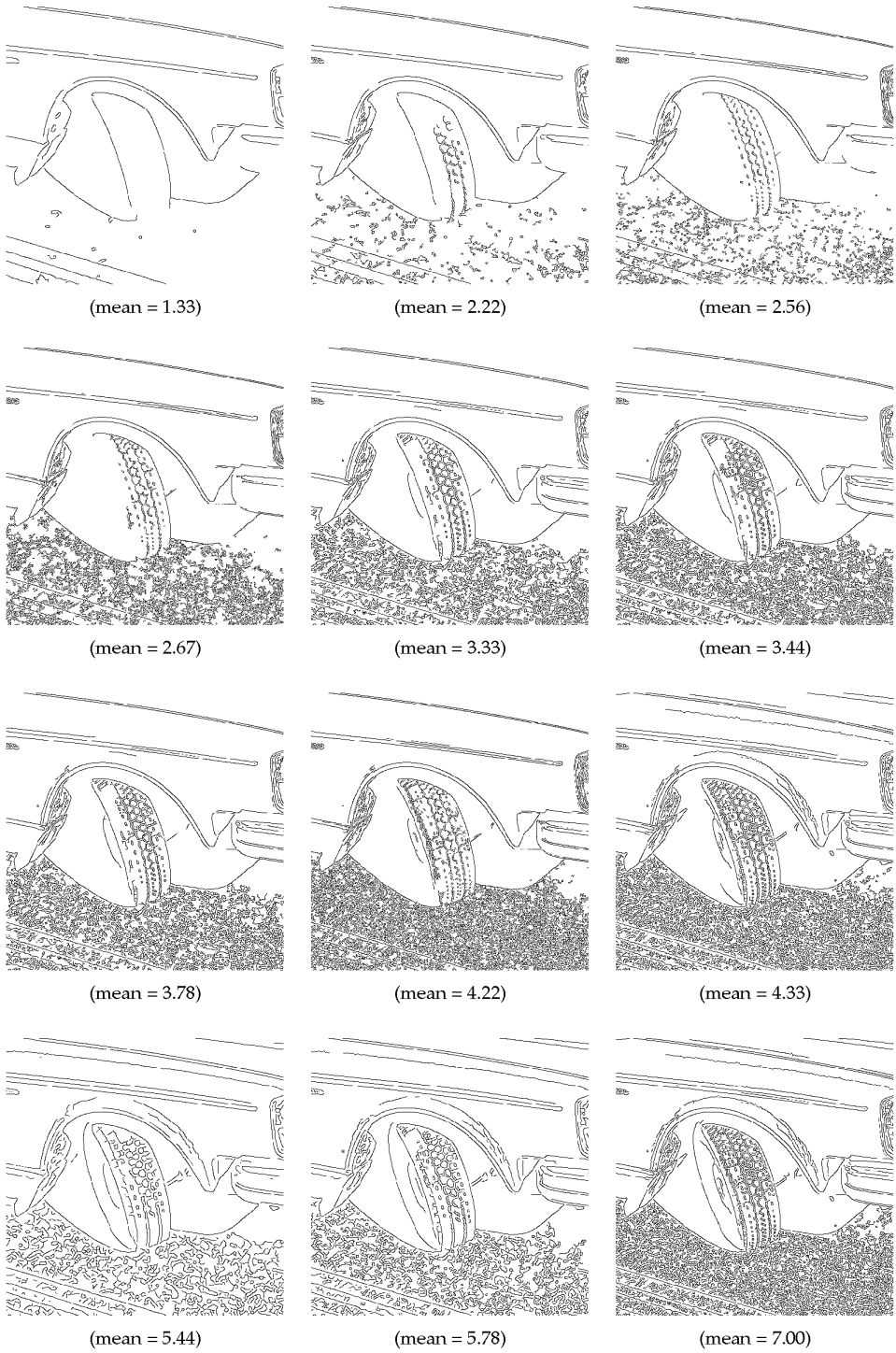


Fig. 4. A sample of the rating collected in the parameter selection experiment. These are the 12 edge images generated from the tire image by the Canny edge detector. The mean below each image is the average of the ratings from the nine participants.

The ICC results in Table 4 show that there was a strong agreement between the participants responses. This means that the participants rated the edge images in a similar fashion and indicates that there is a concept of "edge goodness" that the subjects share. This was an important result because it validated the ability of human subjects to evaluate edge images.

TABLE 4
THE CORRELATION IN RATINGS IN EACH OF THE PARAMETER SELECTION EXPERIMENTS

Detector	20 Image ICC(3, 9)	28 Image ICC(3, 9)
Canny	0.913	0.917
Nalwa	0.923	0.921
Iverson	0.893	0.895
Bergholm	0.908	0.897
Rothwell	0.926	0.925

TABLE 5
THE BEST ADAPTED PARAMETERS LISTED FOR EACH IMAGE

Image	Edge Detection Algorithm				
	Canny	Nalwa	Iverson	Bergholm	Rothwell
golf cart	0.60 0.30 0.90	1.50 0.10 0.60	8 0.20 0.55	2.0 2.0 20	0.5 18 0.80
pitcher	0.60 0.30 0.80	0.60 0.15 0.60	4 0.00 0.05	2.0 1.5 15	0.5 18 0.80
stapler	0.60 0.30 0.90	1.50 0.20 0.60	8 0.20 0.30	5.0 1.5 10	1.0 8 0.95
mailbox	1.20 0.40 0.60	0.60 0.15 0.60	8 0.00 0.30	2.0 1.0 15	1.0 8 0.90
pillow	0.60 0.30 0.80	1.50 0.20 0.60	8 0.20 0.30	3.0 2.0 15	0.5 18 0.80
brush	0.60 0.30 0.90	0.60 0.15 0.60	8 0.00 0.55	2.0 1.5 15	0.5 18 0.80
shopping cart	0.60 0.30 0.90	1.50 0.20 0.60	8 0.20 0.55	2.0 2.0 20	1.0 18 0.80
tire	1.20 0.20 0.60	1.20 0.05 0.15	8 0.00 0.00	3.0 2.0 5	0.5 18 0.80
grater	0.60 0.30 0.90	1.50 0.10 0.60	8 0.20 0.30	2.0 1.0 15	0.5 18 0.80
picnic basket	1.20 0.40 0.60	0.60 0.05 0.30	6 0.00 0.00	3.0 1.5 10	0.5 18 0.80
orange	1.20 0.40 0.80	1.20 0.10 0.60	8 0.00 0.00	3.0 1.5 10	1.0 8 0.90
banana	0.60 0.30 0.80	1.20 0.05 0.30	8 0.00 0.00	4.0 1.5 5	1.5 3 0.90
egg	1.20 0.40 0.80	1.20 0.10 0.60	4 0.00 0.05	4.0 1.5 5	1.5 3 0.90
elephant	0.60 0.30 0.90	1.50 0.20 0.60	8 0.00 0.00	2.0 2.0 20	1.0 13 0.80
pond	0.60 0.30 0.80	0.60 0.10 0.60	8 0.00 0.55	2.0 1.0 15	0.5 18 0.80
pine cone	1.20 0.40 0.60	1.50 0.10 0.60	8 0.00 0.30	3.0 2.0 20	0.5 18 0.80
feather	0.60 0.30 0.80	1.20 0.05 0.30	8 0.00 0.30	2.0 1.0 15	0.5 18 0.80
beehive	1.80 0.30 0.90	1.50 0.20 0.60	8 0.00 0.00	5.0 1.5 10	1.0 13 0.80
turtle	0.60 0.50 0.90	0.60 0.15 0.60	8 0.00 0.30	2.0 1.0 15	1.0 8 0.90
tiger	0.60 0.50 0.90	1.50 0.20 0.60	8 0.00 0.00	2.0 2.0 20	0.5 18 0.80
briefcase	1.20 0.20 0.80	1.50 0.05 0.45	8 0.00 0.05	2.0 1.0 15	0.5 18 0.80
trash can	1.80 0.20 0.70	1.50 0.05 0.45	6 0.00 0.00	3.0 1.5 10	1.0 8 0.95
video camera	1.80 0.20 0.70	1.50 0.10 0.60	8 0.00 0.00	3.0 1.5 10	1.0 8 0.90
coffee maker	1.20 0.30 0.80	1.50 0.10 0.60	8 0.20 0.05	2.0 1.0 15	0.5 18 0.80
flower	0.60 0.30 0.90	1.50 0.10 0.60	8 0.20 0.30	2.0 1.0 15	0.5 18 0.80
airplane	0.60 0.50 0.90	1.50 0.20 0.60	8 0.00 0.55	4.0 1.5 20	1.0 18 0.80
traffic cone	0.60 0.30 0.90	1.20 0.10 0.60	8 0.00 0.30	2.0 2.0 20	0.5 18 0.80
stairs	0.60 0.30 0.80	1.20 0.10 0.60	8 0.20 0.55	2.0 2.0 20	0.5 18 0.80

The Canny parameters are sigma, low, *and* high. *The Nalwa parameters are* blur, low, *and* high. *The Iverson parameters are* directions, low, *and* high. *The Bergholm parameters are* start sigma, end sigma, *and* threshold. *The Rothwell parameters are* sigma, threshold, *and* alpha.

The Parameters Selected for the Evaluation. The ratings were used to determine the best parameter sets using two different criteria.[2] The best single overall parameter set, termed the *fixed parameters*, was identified by averaging the ratings across the subjects, averaging these results across images, and finding the parameter set with the largest average. When two or more parameter sets had nearly the same average, the number of images that each parameter set performed the best on was considered in making the decision of fixed parameters. The best parameter set for each individual image, termed the *adapted parameters*, was also found. This was done by averaging the ratings across subjects and identifying the parameters that had the largest average rating for each image. When more than one parameter set had the same average on an individual image, the parameter set that had the larger average across all of the images was selected. Thus, two types of parameter selections were made for each edge detector. Note that the parameters could actually have the same values because the best fixed parameters could be the same as the best adapted parameters for some image.

The adapted parameters are listed for each image in Table 5. The best fixed parameters for each edge detector are: Canny (0.60 0.30 0.90), Nalwa (1.50 0.20 0.60), Iverson (8 0.00 0.00), Bergholm (2.0 1.0 15), and Rothwell (0.5 18 0.80).

2. Note that only the 20 images that had been categorized by image type were used in determining the best parameters. This was done because the additional eight images (from our first experiment in [6]) did not fit equally into the four categories. Therefore, if they were used, they may have biased the performance in favor of the more prevalent image type.

Consistency Between the Prescreening and Parameter Selection Results. The multistep process used to find parameter combinations for each algorithm raises the question of the consistency between the preference of the parameter prescreener and the preferences of the participants in the parameter selection experiment. The parameter prescreening process reduced the number of parameter combinations from 64 to 12 to limit the scope of the parameter selection experiment. The consistency between the prescreener preferences and the subsequent parameter selection was estimated by calculating the relative score of the edge images selected by the parameter prescreener to the edge images he did not select. This was done for each image and then the results were averaged across the 28 images.

For each picture, the maximum and minimum average ratings, *MAX* and *MIN* were determined, as was the maximum average rating of the images selected by the parameter prescreener, *P*. A relative rating of each edge image selected by the parameter prescreener was then computed as $\frac{P-MIN}{MAX-MIN}$. The relative ratings were then averaged across the 28 images and are listed for each edge detector in Table 6.

These results show that the edge images that were rated highly by the parameter prescreener were also rated highly by the participants in the parameter selection experiments. Given the noise in participant ratings as expressed by the participant rating correlations in Table 4, the relative ratings of 84.2 percent to 92.0 percent for these images are good.

TABLE 6
SUBJECTS RATINGS OF INITIALLY SELECTED PARAMETERS FOR EACH DETECTOR

Detector	Relative Rating
Canny	92.0 percent
Nalwa	84.2 percent
Iverson	88.4 percent
Bergholm	91.9 percent
Rothwell	91.2 percent

6 EDGE DETECTOR EVALUATION

6.1 Methodology

In the edge detector comparison experiment, edge images produced by all of the algorithms were evaluated. The best fixed and adapted parameters, as identified in the parameter selection experiments, were used to generate the edge images. Since there were 28 gray-scale images and five edge detection algorithms with two parameter settings each, there were 280 edge images to evaluate. Sixteen people from the computer vision lab at the University of South Florida volunteered to participate in the experiment. A larger number of participants was used in this experiment than had been used in the parameter selection experiments because the process of selecting the best parameters for each algorithm reduced the range in quality of the edge images and it was thought that more observations might be needed to statistically differentiate between the edge images.

The experiment was conducted in a manner similar to the parameter selection experiments were. To remove the potential for bias towards any algorithm, the evaluation sheets were labeled with codes that did not identify the algorithm. To remove other potential sources of bias for any algorithm, the order of the edge images was randomized separately for each subject.

All 16 participants did the evaluation on the same day. The average time taken to evaluate all of the images was around one and a half hours.

The consistency of the ratings was examined and then the ratings were analyzed to answer three questions. These are:

1) Does the performance of the algorithms improve substantially when the parameters are adapted for each image rather than held fixed for a set of images?
2) What is the relative performance of the edge detection algorithms?
3) Does the measure of the relative performance of the edge detectors depend on the selection of images used in the evaluation?

6.2 Results

6.2.1 Correlation Between Participant Responses

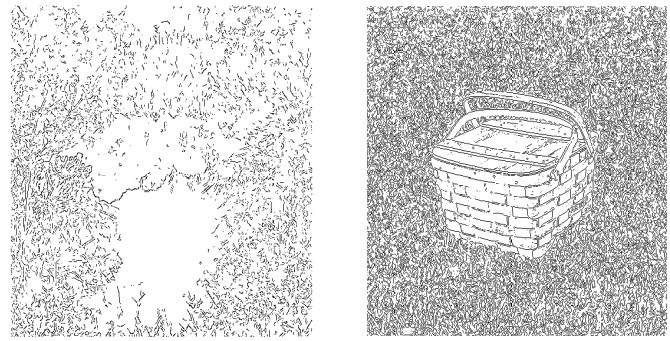
The same interclass correlation measure that was applied to the parameter setting data above was applied to the data collected in this experiment. Again, it is important to have a high correlation between participants because the ratings from different participants will be used as multiple observations in the data analysis. Table 7 lists the interclass correlations for the 20 and 28 image results. The correlation is strong, indicating a good agreement between the participants' ratings.

TABLE 7
THE INTERCLASS CORRELATION COEFFICIENT FOR THE EDGE DETECTOR COMPARISON EXPERIMENT

Detectors	20 Image ICC(3,16)	28 Image ICC(3,16)
All	0.939	0.928

6.2.2 Analysis of the Edge Detector Ratings

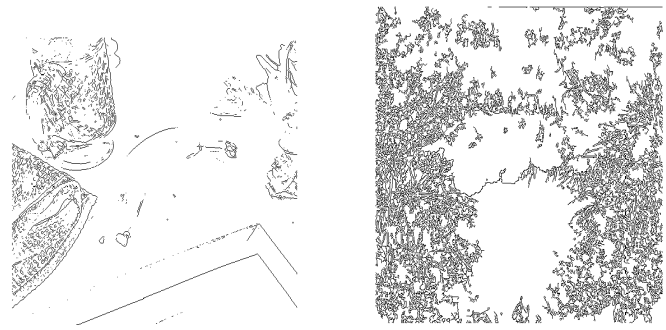
Before presenting the numerical results for the evaluation, selected results are presented to visually calibrate the reader to the numerical scale of edge ratings. The lowest and the highest ratings (averaged across participants) for any single image in the evaluation are displayed in Fig. 5.



(a) Pond (mean = 1.44) (b) Basket image (mean = 6.31)

Fig. 5. The lowest and highest rated edge images. Image (a) was produced by the Nalwa edge detector with input parameters *blur* = 1.50, *low* = 0.20, and *high* = 0.60. Image (b) was produced by the Rothwell edge detector with input parameters *sigma* = 0.50, *low* = 18.0, and *alpha* = 0.80.

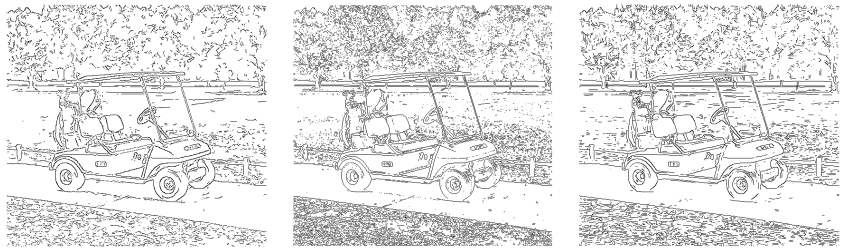
The correlation between the participant responses reported in Section 6.2.1 showed a strong agreement between the relative ratings of participants. Of course, this agreement was not perfect. Fig. 6 shows the two edge images that had the lowest, and the highest, variance of the 16 participant ratings.



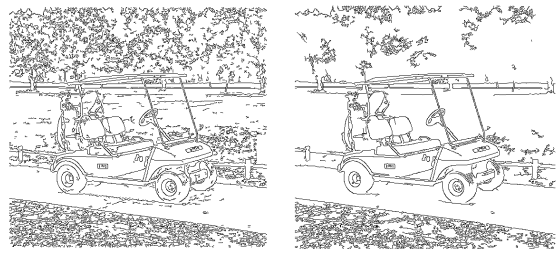
(a) Banana (mean = 1.56, std = 0.75) (b) Pond (mean = 2.63, std = 2.00)

Fig. 6. Edge images that had the lowest and highest variance in the ratings. Image (a) was the image with the lowest variance in ratings and was produced by the Rothwell edge detector with input parameters *sigma* = 0.50, *low* = 18.0, and *alpha* = 0.80. Image (b) was edge image with the highest variance in ratings and was produced by the Canny edge detector with input parameters *sigma* = 0.60, *low* = 0.30, and *high* = 0.90.

To help further understand the range of the scale, the five edge images that had the smallest range of average ratings

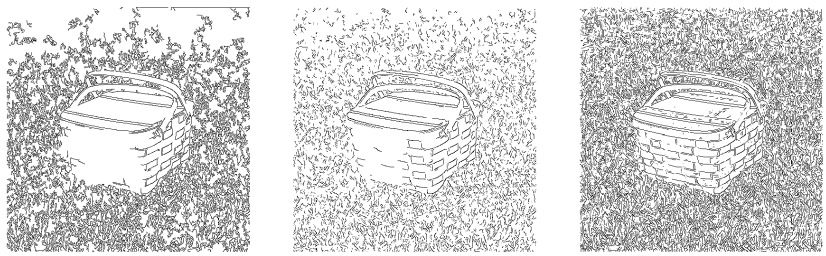


(a) Nalwa (mean = 5.31) (b) Rothwell (mean = 5.38) (c) Iverson (mean = 5.56)

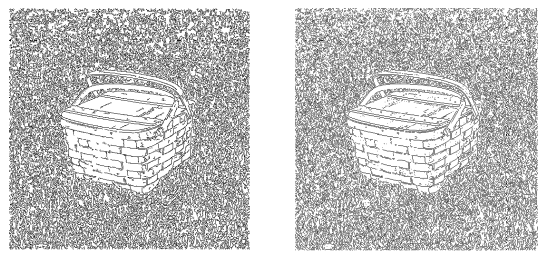


(d) Bergholm (mean = 5.69) (e) Canny (mean = 6.13)

Fig. 7. Edge images that had the smallest range in average ratings. Image (a) was produced by the Nalwa edge detector with input parameters *blur* = 1.50, *low* = 0.20, and *high* = 0.60. Image (b) was produced by the Rothwell edge detector with input parameters *sigma* = 0.50, *low* = 18.0, and *alpha* = 0.80. Image (c) was produced by the Iverson edge detector with input parameters *direction* = 8, *low* = 0.000, and *high* = 0.00. Image (d) was produced by the Bergholm edge detector with input parameters *starting sigma* = 2.00, *ending sigma* = 1.00, and *threshold* = 15. Image (e) was produced by the Canny edge detector with input parameters *sigma* = 0.60, *low* = 0.30, and *high* = 0.90.



(a) Canny (mean = 3.06) (b) Nalwa (mean = 3.63) (c) Iverson (mean = 5.63)



(d) Bergholm (mean = 5.81) (e) Rothwell (mean = 6.25)

Fig. 8. Edge images that had the largest range in average ratings. Image (a) was produced by the Canny edge detector with input parameters *sigma* = 0.60, *low* = 0.30, and *high* = 0.90. Image (b) was produced by the Nalwa edge detector with input parameters *blur* = 1.50, *low* = 0.20, and *high* = 0.60. Image (c) was produced by the Iverson edge detector with input parameters *direction* = 8, *low* = 0.000, and *high* = 0.00. Image (d) was produced by the Bergholm edge detector with input parameters *starting sigma* = 2.00, *ending sigma* = 1.00, and *threshold* = 15. Image (e) was produced by the Rothwell edge detector with input parameters *sigma* = 0.50, *low* = 18.0, and *alpha* = 0.80.

are displayed in Fig. 7, and the five edge images that had the largest range of average ratings are displayed in Fig. 8.

Does the performance of the edge detectors depend on whether the parameters are held constant for all of the images or are adapted to each image?

The edge images produced with the adapted parameters were rated 4.60 on average and the edge images produced with the fixed parameters were rated 4.15 on average. Table 8 shows results from an analysis of variance[3] computed using a model that contains the parameter combination term in each source of variation measured. The result in the first row indicates that there is a significant difference ($Pr > F = 0.001$) in the ratings of the edge images that were generated with fixed and adapted parameters. Thus, the edge detectors

3. Analysis of variance is a general statistical method for analyzing experimental data. Many texts on statistics describe analysis of variance methods. For example, see [46] for an introduction to the subject.

TABLE 8
ANOVA RESULTS FOR A TEST OF THE SIGNIFICANCE OF THE EFFECT OF FIXING THE INPUT PARAMETERS ACROSS ALL IMAGES OR ADAPTING THEM FOR EACH IMAGE

Source	DF	ANOVA SS	Mean Square	Pr > F
Parameter combination	1	163.81	163.81	0.0001
Parameter combination × Edge detector	8	120.06	15.01	0.0001
Parameter combination × Man-made/Natural	2	1,095.70	547.85	0.0001
Parameter combination × Textured/Nontextured	2	24.35	12.17	0.0052
Parameter combination × Edge detector x Man-made/Natural	8	41.07	5.14	0.0230
Parameter combination × Edge detector × Textured/Nontextured	8	165.06	20.63	0.0001
Parameter combination × Edge detector × Man-made/Natural × Textured/Nontextured	10	115.02	11.50	0.0001
Error	3,160	7,209.93	2.31	

performed significantly better with adapted rather than fixed parameters. The table also indicates that all of the interactions between the factors are significant. This implies that the size of the difference in the ratings obtained with the fixed and adapted parameters varies with the other factors.

What is the relative performance of the five edge detection algorithms?

The data collected in this experiment were split into two pieces to answer this question. One subset contained the data for the adapted parameters and the other contained the data for the fixed parameters. A separate analysis was done on each of these data sets because the performance of an edge detector should be evaluated using either adaptive or fixed parameters, but not both.

Statistically significant differences in the mean performance of the algorithms were determined using the Bonferroni test [47]. This test applied individual statistical tests for differences in the mean performance between each pair of edge detection algorithms using a one-way analysis of variance. Ten separate analyses of variance were done. Since the same data was used in multiple tests whose results were to be compared, the level of significance was adjusted from 0.05 to 0.005 for each test. This was done so the family-wise error was approximately 0.05. The analysis was the same for the adapted and fixed parameter data.

Results of the Adapted Parameter Comparison. Table 9 lists the results of the comparison of edge detectors using the parameters that were optimized and set individually for each of the 20 images. The table lists the mean performance of each edge detector and the significant differences in the mean performance between the algorithms. Because the significant differences account for the distribution of scores for each edge detector, they are more accurate measures of the true difference in edge detector performance than a simple difference in the means. The statistically significant differences in the performance of the algorithms indicate that the Rothwell, Bergholm, and Canny edge detectors all performed significantly better than both the Iverson and Nalwa edge detectors.

TABLE 9
RELATIVE EDGE DETECTOR PERFORMANCE USING ADAPTED PARAMETERS

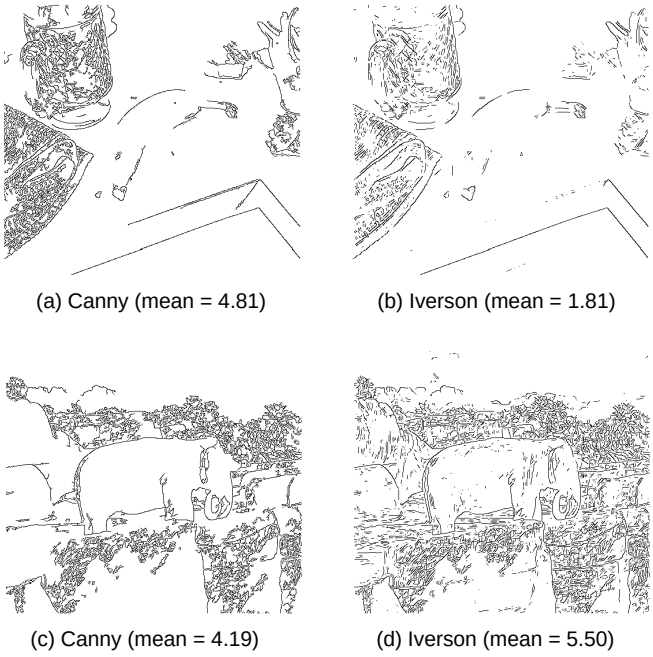
Edge Detector	Mean	Significant Differences
Canny (C)	4.80	(I, N) < (R, B, C)
Bergholm (B)	4.78	
Rothwell (R)	4.76	
Nalwa (N)	4.43	
Iverson (I)	4.26	

Although the Canny edge detector performed significantly better than the Iverson edge detector on average, the performance can be quite different on any particular image. Fig. 9 shows the image where the Canny edge detector performed better than the Iverson edge detector by the largest amount and an image where the Iverson edge detector outperformed the Canny edge detector by the largest amount.

Results of the Fixed Parameter Comparison. Table 10 lists the results of the comparison of edge detectors using the parameters that were optimized and fixed for the set of 20 images. The statistically significant differences reveal that the Bergholm edge detector outperformed both the Canny and Nalwa edge detectors. The Iverson and Rothwell edge detectors did not perform significantly differently from any of the edge detectors in the pairwise tests.

TABLE 10
RELATIVE EDGE DETECTOR PERFORMANCE USING FIXED PARAMETERS

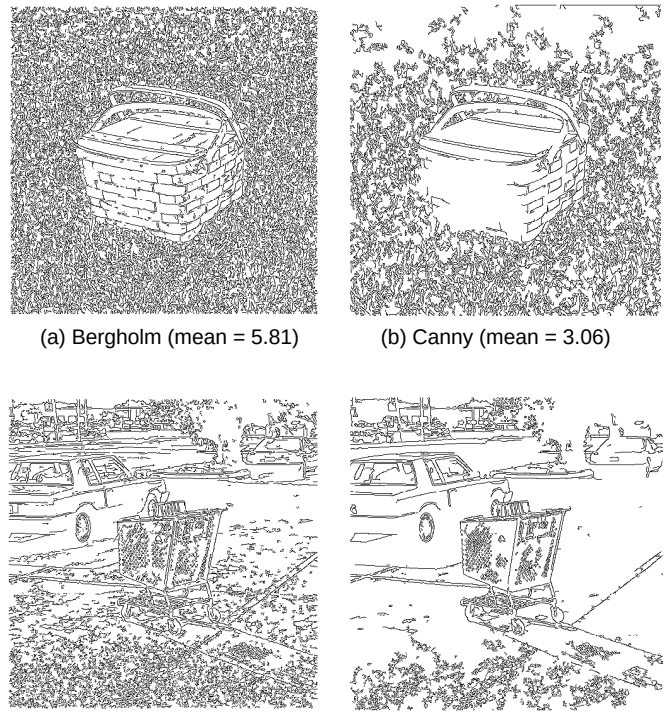
Edge Detector	Mean	Significant Differences
Bergholm (B)	4.38	(C, N) < B
Iverson (I)	4.24	
Rothwell (R)	4.21	
Nalwa (N)	3.97	
Canny (C)	3.96	



(a) Canny (mean = 4.81) (b) Iverson (mean = 1.81)

(c) Canny (mean = 4.19) (d) Iverson (mean = 5.50)

Fig. 9. The adapted parameter edge detector evaluations are image dependent. In the banana image, the edges detected with the Canny algorithm (a) were rated higher than the edges detected with the Iverson algorithm (b). Image (a) was produced with input parameters *sigma* = 0.60, *low* = 0.30, and *high* = 0.90. Image (b) was produced with input parameters *direction* = 8, *low* = 0.000, and *high* = 0.000. In the elephant image, the edges detected with the Canny algorithm (c) were rated lower than the edges detected with the Iverson algorithm (d). Image (c) was produced with input parameters *sigma* = 0.60, *low* = 0.30, and *high* = 0.90. Image (d) was produced with input parameters *direction* = 8, *low* = 0.000, and *high* = 0.000.



(a) Bergholm (mean = 5.81) (b) Canny (mean = 3.06)

(c) Bergholm (mean = 4.56) (d) Canny (mean = 5.31)

Fig. 10. The fixed parameter edge detector evaluations are image dependent. In the basket image, the edges detected with the Bergholm algorithm (a) were rated higher than the edges detected with the Canny algorithm (b). Image (a) was produced with input parameters *starting sigma* = 2.0, *ending sigma* = 1.0, and *threshold* = 15. Image (b) was produced with input parameters *sigma* = 0.60, *low* = 0.30, and *high* = 0.90. In the shopping cart image, the edges detected with the Bergholm algorithm (c) were rated lower than the edges detected with the Canny algorithm (d). Image (c) was produced with input parameters *starting sigma* = 2.0, *ending sigma* = 1.0, and *threshold* = 15. Image (d) was produced with input parameters *sigma* = 0.60, *low* = 0.30, and *high* = 0.90.

As with the adapted parameters, the performance of the algorithms depends on the image when fixed parameters are used. This is illustrated in Fig. 10. The Bergholm edge detector performed better than the Canny edge detector on the basket image, whereas the Canny edge detector outperformed the Bergholm edge detector on the shopping cart image.

How does the measured relative performance of the edge detectors depend on the selection of images used in the evaluation?

An analysis of variance was done on the adapted parameter data to examine the interaction between the performance of the edge detectors and the image type, manmade, natural, textured, and nontextured. Results from the analysis of variance in Table 11 show that there are also significant interactions between the edge detector performance and the image type (Pr > F <= 0.0023 for all the tests) in the adapted parameter data. This means that the size of the difference in the performance of the edge detectors changes with the type of images used to compare the algorithms.

Because there was an interaction between the performance of the algorithms and the image type, the relative performance of the algorithms was calculated separately for each image type. The results are displayed in Table 12. The table shows that the relative ranking of the edge detectors in each image category is consistent; no two algorithms change significantly in relative performance to each other.

TABLE 11
ANOVA RESULTS FOR THE RATINGS OBTAINED FOR USING ADAPTED PARAMETERS

Source	DF	ANOVA SS	Mean Square	Pr > F
Edge detector	4	77.27	19.32	0.0001
Edge detector × Man-made/Natural	5	414.44	82.88	0.0001
Edge detector × Textured/Nontextured	5	67.12	13.42	0.0001
Edge detector × Man-made/Natural × Textured/Nontextured	5	40.99	8.20	0.0023
Error	1,580	3,476.95	2.20	

The low performance of the Iverson algorithm on the Natural, Nontextured images stands out. A study of the results shows that the ratings are low across four of the five images in this category. This means that the low overall performance is not due to a "problem image" but is due to the whole set of images in this category. Currently, we are not able to conjecture a plausible explanation for this result.

TABLE 12
RELATIVE PERFORMANCE OF THE EDGE DETECTORS ON SUBSETS OF THE ADAPTED PARAMETER IMAGES

Man-made Nontextured		
Edge Detector	Mean	Significant Difference
Canny (C)	5.54	I < C
Bergholm (B)	5.20	
Rothwell (R)	5.15	
Nalwa (N)	5.06	
Iverson (I)	4.85	

Man-made Textured		
Edge Detector	Mean	Significant Difference
Bergholm (B)	5.38	N < (R, B)
Rothwell (R)	5.33	
Canny (C)	5.03	
Iverson (I)	4.95	
Nalwa (N)	4.59	

Natural Nontextured		
Edge Detector	Mean	Significant Difference
Nalwa (N)	4.31	I < (B, C, R, N)
Rothwell (R)	4.24	
Canny (C)	4.19	
Bergholm (B)	4.13	
Iverson (I)	2.96	

Natural Textured		
Edge Detector	Mean	Significant Difference
Canny (C)	4.44	NONE
Bergholm (B)	4.41	
Rothwell (R)	4.31	
Iverson (I)	4.26	
Nalwa (N)	3.76	

A similar analysis was done on the fixed parameter data. Table 13 shows results from an analysis of variance used to examine the significance of the interactions between the edge detector and the image type. The results indicate that there is a significant difference in the average ratings of the edge detectors ($Pr > F = 0.0015$) and that there are significant interactions between the edge detector and the image type ($Pr > F <= 0.0001$ for all the tests). This means that the size of the difference in the performance of the edge detectors changes with the type of images used to compare the algorithms.

TABLE 13
ANOVA RESULTS FOR THE RATINGS OBTAINED FOR USING FIXED PARAMETERS

Source	DF	ANOVA SS	Mean Square	Pr > F
Edge detector	4	42.79	10.70	0.0015
Edge detector × Man-made/Natural	5	722.33	144.47	0.0001
Edge detector × Textured/Nontextured	5	122.29	24.45	0.0001
Edge detector × Man-made/Natural × Textured/Nontextured	5	74.03	14.81	0.0001
Error	1,580	3,813.98	2.41	

Because there was an interaction between the performance of the algorithms and the image type, separate analyses were conducted for each of the four subsets of the data. Table 14 shows the results of the analysis. The significant differences between the performance of the algorithms are generally consistent with each other. The only inconsistency was in the relative performance of the Rothwell and the Nalwa algorithms. The Nalwa algorithm was determined to be significantly better than the Rothwell algorithm on the Natural Nontextured objects while the Rothwell algorithm was determined to be significantly better than the Nalwa algorithm on the Man-made Textured images. This flip-flop may have masked the Rothwell algorithm from being significantly different from the other algorithms in the analysis based on the 20 images. This illustrates the dependence of the measured performance of an edge detector on the images that are used to evaluate them. In general, we would strongly caution against reading too much into the differences in rankings for different five-image groups because of the small data size in each category.

TABLE 14
RELATIVE PERFORMANCE OF THE EDGE DETECTORS ON SUBSETS OF THE FIXED PARAMETER IMAGES

Man-made Nontextured		
Edge Detector	Mean	Significant Difference
Bergholm (B)	5.28	NONE
Rothwell (R)	4.86	
Iverson (I)	4.80	
Canny (C)	4.79	
Nalwa (N)	4.76	

Man-made Textured		
Edge Detector	Mean	Significant Difference
Rothwell (R)	5.13	(N, C) < (I, B, R)
Bergholm (B)	5.10	
Iverson (I)	5.04	
Canny (C)	4.30	
Nalwa (N)	4.04	

Natural Nontextured		
Edge Detector	Mean	Significant Difference
Nalwa (N)	3.61	R < N
Canny (C)	3.39	
Bergholm (B)	3.13	
Iverson (I)	2.90	
Rothwell (R)	2.84	

Natural Textured		
Edge Detector	Mean	Significant Difference
Iverson (I)	4.24	(C, N) < I
Rothwell (R)	4.03	
Bergholm (B)	4.00	
Nalwa (N)	3.46	
Canny (C)	3.35	

7 DISCUSSION

7.1 Discussion of Results

The results show that significantly better results are obtained when the parameters are adapted to each image than when one set of fixed parameters are used. While this is to be expected, it is a significant result because it implies that the amount of effort expended in parameter optimization can influence the measured performance of the algorithm.

Therefore, equal effort must be applied in optimizing the parameters for all of the algorithms to do a fair performance evaluation.

The analysis of the relative performance of the algorithms resulted in a ranking of the algorithms as (Canny, Nalwa) < Bergholm for fixed parameters and as (Iverson, Nalwa) < (Rothwell, Bergholm, Canny) for adapted parameters. The performance increases from left to right and the parentheses group algorithms whose difference in performance was not statistically significant. It is evident that the newer algorithms have achieved an increase in performance over the older algorithms when the parameters are fixed. When the parameters are adapted for each image, however, the newer algorithms did not show the same significant increase in performance. This performance increase for fixed parameters is a real achievement because there is at present no general way to adapt the parameters of an algorithm for every image. If there were such a method, it would be part of the edge detection algorithm itself.

The Canny algorithm had the highest performance when the parameters were adapted for each image, but the lowest performance when the parameters were fixed. Although the performance was not statistically significantly better than the performance of the Rothwell or Bergholm algorithms, improving from the worst performance in fixed parameters to the best performance in adapted parameter performance is striking. This suggests that the parameters of the Canny algorithm are "good knobs to turn."

The choice of the edge detection algorithm may depend on its application. For example, computer vision researchers developing higher level vision processing methods may prefer to use the Canny algorithm because it can produce better edge images if care is taken in adjusting the parameters manually. This would provide them with better edges to use in investigating higher level vision processing algorithms, however, researchers implementing "production" vision systems that cannot manually adapt the parameters may benefit from selecting one of the newer algorithms to incorporate into their system.

Finally, it is interesting that there were no significant differences between the performance of the algorithms when they were applied with fixed parameters to images containing Man-Made/Nontextured objects. These are the types of images that these algorithms were designed to process.

7.2 Discussion of the Evaluation Methodology

The methodology we presented in this paper is novel in two ways. It relies on using real images in the evaluation and it measures the performance of edge detection algorithms using the human visual system to estimate the ability to recognize objects in the images.

Measuring the performance of edge detection algorithms using a sample of real images limits the evaluation because the results that are obtained are conditioned to the images used in the evaluation. Recall that the performance of the algorithms changed with the image properties as specified by the crude categorization of images as Man-made/Nontextured, Man-made/Textured, Natural/Nontextured, and Natural/Textured. This means that a large number of images should be used to evaluate an edge detector. In this work, 20 images were used in the overall evaluation. While this is not a large number of images, they have diverse image characteristics, and it is a larger number of images than those which have been used in any previous edge detector comparison study.

To cope with the errors introduced by using a sample of real images, a moderate sized, diverse set of images were used in the evaluation. Images of Man-Made and Natural and Textured and Nontextured objects were used in the evaluation because we believed that such images would test the algorithms over a broad range of image attributes and they could be used to test if the relative performance of the algorithms depended on the image characteristics. It is important to point out however that the analysis for each type of image (i.e., Man-made/Textured) was done on a set of only five images. While statistically significant differences in performance were found, the ability to generalize the results may be very limited. For example, it is unlikely that the particular five images we used adequately sample the space of all Man-made/Textured images. Therefore, we do not claim that any particular algorithm works best on any of the general image categories ("Textured," "Man-Made," etc.). We only defined and used the categories to test the algorithms on meaningful subsets of the images.

Selecting the task of object recognition to evaluate the performance of the algorithms from the edge images also places constraints on the generalizations of the results. Since the performance of the algorithms was measured using an object recognition task, the results are only directly meaningful for that task. For example, using these results to select an algorithm for isolating the features to use in stereo correspondence processing may be risky.

7.3 Comparison With Signal Based Characterization of Edges

A natural question to ask is how does the evaluation strategy presented here compare with signal based strategies. We explore this now. Typically signal based strategies compare algorithms with respect to error criteria such as the probability of false alarm and missed detection rates. However, computing these errors requires the enunciation of ground truth edges which is readily possible only for synthetic images.

We created a synthetic image consisting of a bright circular area against a dark background, similar to the one shown in [7, p. 88]. First, we created a 256×256 sized image consisting of a circle with a gray level of 140 and with radius 96 pixels against a gray background level of 60. We then subsampled this image by first averaging in a 4×4 window and then resampling to arrive at a 64×64 image. This subsampling effectively smoothes the intensity transition between the circle and the background, i.e., introduces some partial pixel effects. To produce noisy versions of this pure image, we added Gaussian noise to reduce the signal to noise ratio to four and to produce the image shown in Fig. 11.

A three-labeled ground truth for the image was then made. Pixels along the boundary of the circle were marked are true-positive edges. Pixels which were connected to true-positive marked pixels were marked as don't care to allow for varied connectedness. And the remainder of the image was marked as a false-positive region.

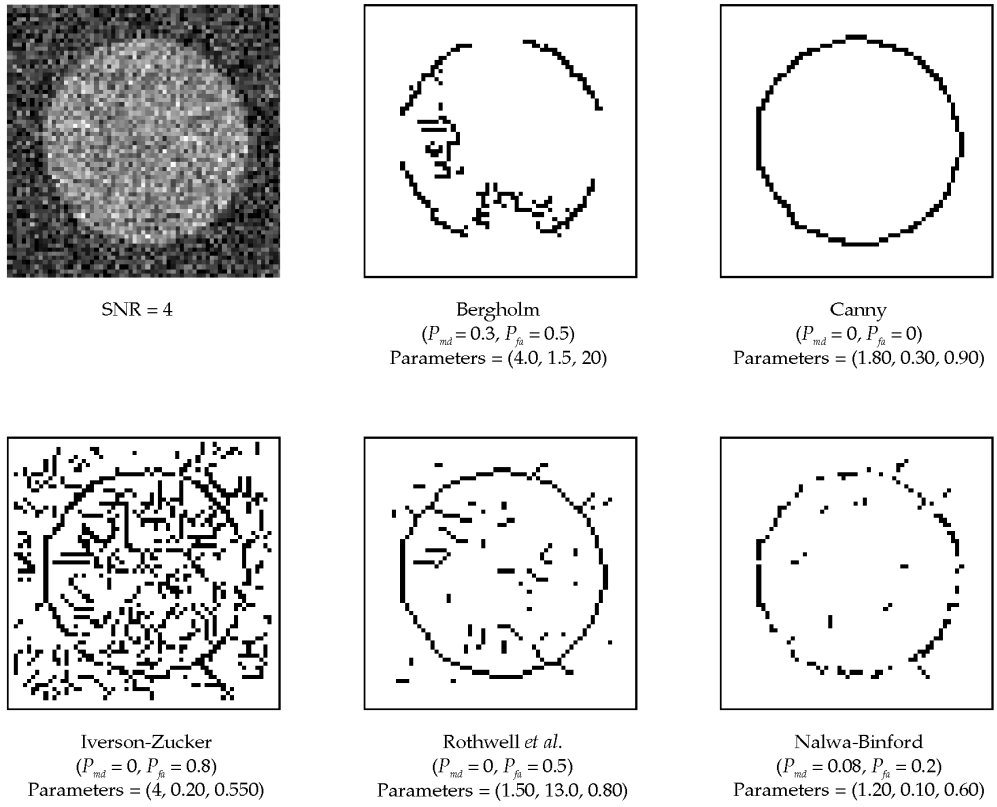


Fig. 11. Edge images for the five edge detectors with best of the 12 parameters sets used in the rating experiment as shown in Table 3. The missed detection (P_{md}) and the false alarm (P_{fa}) rates are shown below each image along the best parameter choices.

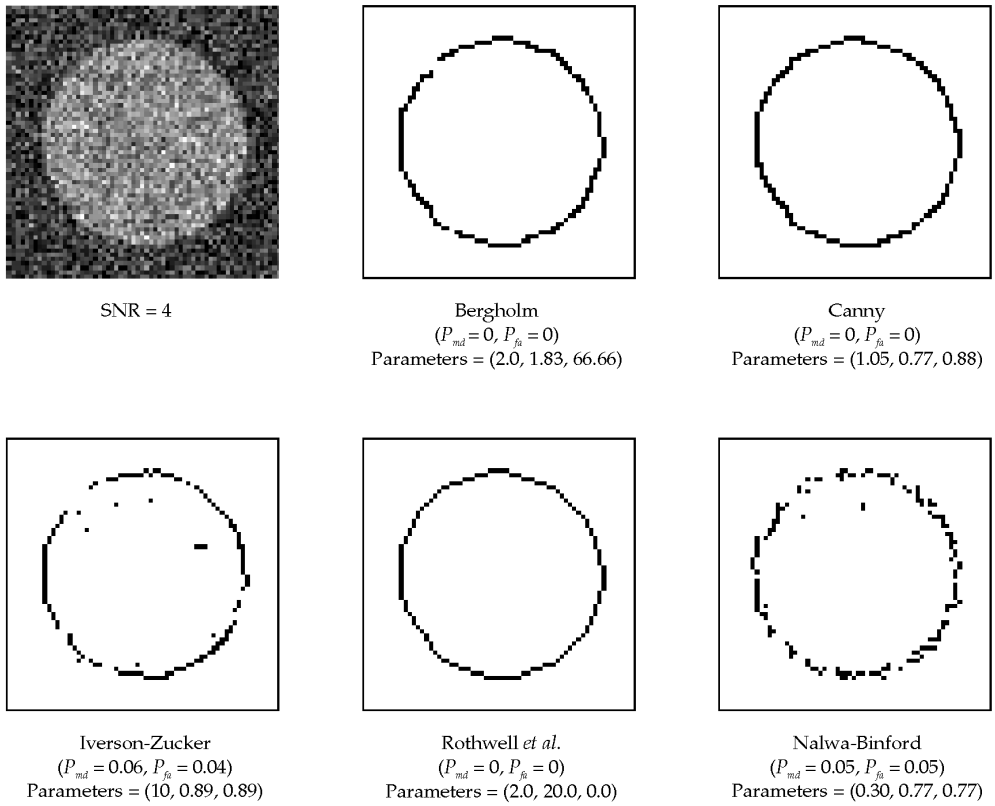


Fig. 12. Edge images of the noisy image (with SNR = 4) in (a) for the five edge detectors with the best of a $10 \times 10 \times 10$ sampling of the parameter space. The missed detection (P_{md}) and the false alarm (P_{fa}) rates are shown below each image along the parameter choices.

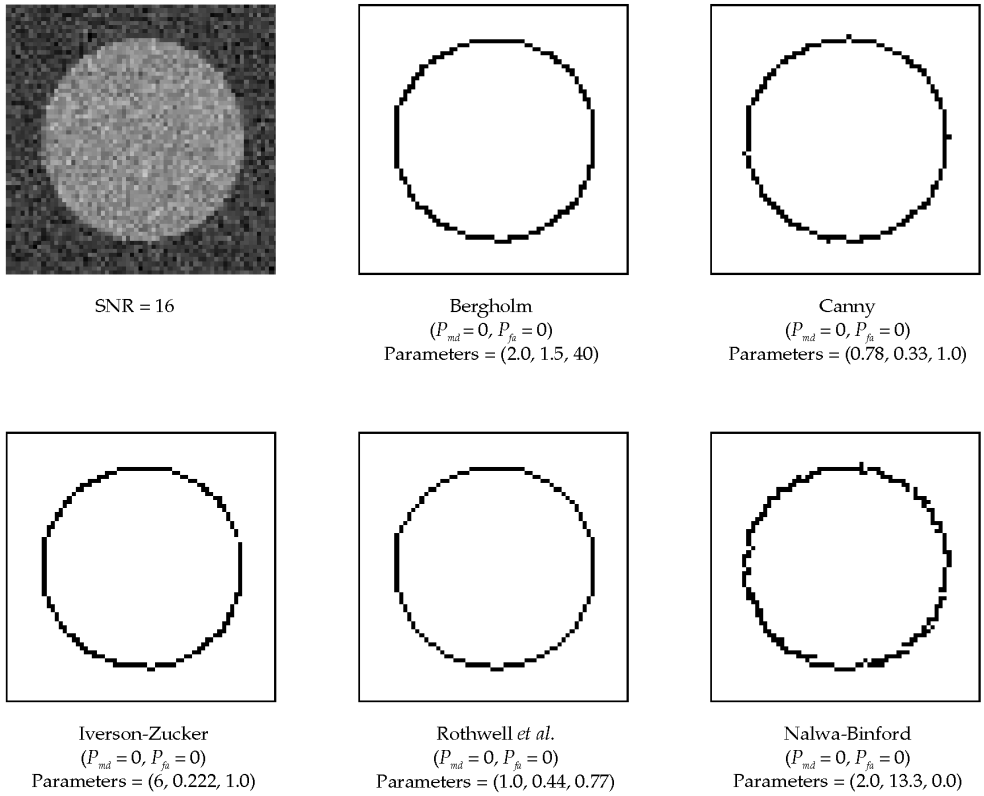


Fig. 13. Edge images of the noisy image (with SNR = 16) in (a) for the five edge detectors. The results are with the best of a $10 \times 10 \times 10$ sampling of the parameter space. The missed detection (P_{md}) and the false alarm (P_{fa}) rates are shown below each image along the parameter choices.

In order to evaluate the edge images produced from the synthetic images, we counted the pixels that fall into the various ground truth labeled regions. To account for locational error in the ground truth or edge images, a circular search radius of three pixels was used to match true-positive edges. (Choosing a smaller radius of two pixels did not produce a significantly varied result.) However, no search range was used for counting false-positive pixels. The unmatched true-positive ground truth pixels determines the missed detection rate (P_{md}). And false-positive pixels contribute to the false alarm error of the edge detector (P_{fa}). Ideally, we would like both these errors to be zero.

We sampled the edge detector parameter spaces in two different ways. First, for each edge detector we used the same 12 parameter sets used in the rating experiment, as shown in Table 3. Each edge image was evaluated in terms of the missed edge pixels and false alarms. The best results are shown in Fig. 11. The missed detection (P_{md}) and the false alarm (P_{fa}) rates are shown below each image. We notice that except for the Canny detector and to some extent the Nalwa-Binford detector, results of the other detectors are unsatisfactory. This might be due to the inadequate sampling of the parameter space using the 12 parameter sets from the rating experiment. Indeed, one of the primary conclusions from the rating experiment is that the parameters need to be tuned on a per-image basis to get best performance. Because the synthetic image is so different from the real images, it may need very different parameter settings.

Since we are working with a synthetic image, we can choose a finer sampling of parameters in a more automated way than done for the rating experiment. Recall that each edge detector has three parameters that can be chosen. Thus, in the second comparison we choose a $10 \times 10 \times 10$ uniform sampling of the parameter space for each detector. The best results for each detector are displayed in Fig. 12. Notice that all the detectors get near-perfect results. As we can see from Figs. 13 and 14, all the edge detectors achieve perfect results on a less noisy image (SNR = 16) and (of course) on a noiseless image. This seems to suggest that signal based comparison on synthetic images might not be sufficient by itself to distinguish between edge detectors. It needs to be complemented by a more global assessment as offered by the rating experiments in this paper. We base this observation not just on the result of one synthetic image but also on our practical experience with edge detectors. We believe that present day edge detectors perform extremely well in terms of signal based criteria measured on such simple synthetic images.

7.4 Extended Application of the Evaluation Method

It is important to realize that the ratings collected in this experiment are relative evaluations. Therefore, it would be a mistake to measure the performance of another algorithm by collecting data for it and comparing the numerical scores to the results collected in this experiment. Doing so would ignore many potentially significant sources causing differences in performance. However, there is another way to leverage off the results of this evaluation to measure the relative performance of a new algorithm.

Reapplying the evaluation method in its complete form would require obtaining new images, categorizing the images

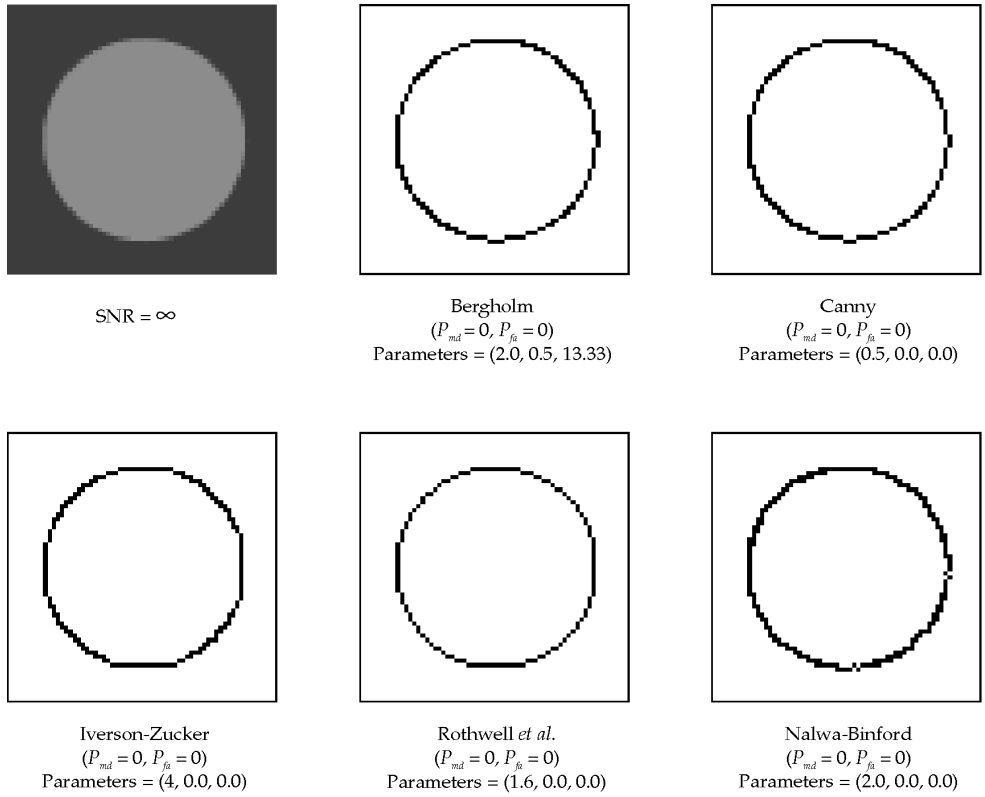


Fig. 14. Edge images of the noise less image in (a) for the five edge detectors. The results are with the best of a $10 \times 10 \times 10$ sampling of the parameter space. The missed detection (P_{md}) and the false alarm (P_{fa}) rates are shown below each image along the parameter choices.

by their properties, determining the parameters to use for each algorithm and then comparing the relative performance of the algorithms. This is clearly time consuming.

To make it easier for other researchers to compare edge detection algorithms, the images that were used in this evaluation are being made available on the world wide web (`http://marathon.csee.usf.edu/edge/edge_detection.html`) and by anonymous ftp (`ftp://figment.csee.usf.edu/pub/Edge_Comparison/images`). Using these images, the only steps required to evaluate a new algorithm are:

1) to identify the parameters for the new algorithm using the parameter selection methodology outlined in this paper,
2) to conduct the edge detector comparison experiment with both the new algorithm and at least a few algorithms used in this study (we recommend the Canny and Bergholm), and
3) to perform the statistical analysis.

Therefore, the steps involved in evaluating a new edge detection algorithm are to:

1) Decide which of the algorithms the new algorithm is to be compared with. In principle, one could compare a new algorithm with only the highest performance algorithm. This, however, is not the recommended approach because comparing several edge detectors is much more informative, and is little more work than comparing only two algorithms.
2) Acquire the images from the ftp site.
3) Select 64 initial parameter combinations for the new algorithm and generate the 64 edge images for each of the 20 gray-scale images. View the edge images and select the best five for each of the 20 images (12 hours time). Apply a greedy search to find a subset of 12 of the 64 parameters. At each step in the search, select the parameter set that reduces the number of best edge images (top 5/64) included across all 20 images the least number of times, by the largest amount possible. This means that one first searches for a subset of parameters that includes all images at least once, then includes each image at least twice, etc. until 12 parameter sets are included.
4) Print out evaluation sheets similar to those in Fig. 3 for each of the 240 edge images (20 images $\times$ 12 parameters). Print out the 20 gray-scale images. Make a set of evaluation images for each participant by photocopying the prints. Have a number of participants (we used nine) evaluate all 240 edge images in one session. Note that paper is used because it allows the participants to view the whole set of edge images at once. This is not possible on a computer monitor due to its limited resolution. The experiment should take a couple of hours.
5) Calculate the ICC(3, k) correlation coefficient to check that the subjects shared a common rating scheme. Calculate the average ratings for each parameter set for each image. To identify the best adapted parameters, find the best average rating for each image. To find the best fixed parameters, calculate the mean of

the average ratings (across images) and find the parameter set with the largest mean. Please note, that the parameters identified by this process may (and probably are not) the very best fixed or adapted parameters attainable. If someone spent more time adjusting the parameters to each image (starting with more than 64 initial parameter sets), they may get better results. Therefore, the above procedure should be used because it applies equal effort in the parameter optimization process for all of the algorithms.

6) Print out the edge images for each algorithm for each image. Also print the gray-scale images. Make a number of sets of evaluation images (we made one set for each of 16 participants) by photocopying the prints. Randomize the edge images within each of the 20 sets of images separately for each of the evaluation sets. Then randomize the order of the 20 sets of images. Have each participant evaluate all of the edge images in one session. The time required for this step will depend on the number of edge detectors being compared. If the new edge detector is compared to all five of the edge detectors evaluated in this paper then there will be 240 edge images for each participant to evaluate. It should take a couple of hours to rate 240 edge images.
7) Calculate the ICC(3, k) correlation coefficient to determine if the subjects shared a common rating scheme. Divide the data in half (adapted and fixed parameter data) and analyze each subset of data the same way. Perform a set of one-way analysis of variance tests using the ratings obtained for each pair of algorithms. The number of tests will depend on the number of algorithms being compared. In deciding whether each one way ANOVA test is significant, use $\alpha = \frac{0.05}{c}$, where c is the number of statistical tests being done. We used $\alpha = 0.005$ because c is 10 (five choose two). If all six algorithms are compared then there will be 15 one-way ANOVAs and $\alpha = 0.00\overline{3}$. Calculate the means for each detector, order them, and group the means using the results of the ANOVA tests to identify statistically significant differences in the ratings. This will provide the relative performance of the algorithms.

We estimate that a comparison of edge detectors using the above method could realistically be conducted in three or four weeks. This is worthwhile because it clearly demonstrates the performance of an algorithm. Given the amount of time it takes to develop a new edge detection algorithm, investing one month to demonstrate its performance is, we believe, a good use of that time.

It is important to note that repeated application of this evaluation method can eventually "wear out" the image set. This is because new edge detectors might be designed to give good performance on the particular set of 20 images used in the evaluation without providing good performance on other (unseen) images. This is an inherent problem with making an evaluation data set public because an algorithm can be over trained to perform well on the test data. To minimize the possibility of this, a new set of images could occasionally be substituted for the 20 images presently used in the evaluation. The cost of doing this would be that the object screening must be performed with the new images and the parameter setting experiment must be reapplied for each algorithm.

ACKNOWLEDGMENTS

We would like to acknowledge Mr. Sean Dougherty's effort in generating the results on the synthetic image. We want to express our thanks to Vic Nalwa, Steven Zucker, Fredrik Bergholm, and Charlie Rothwell for making their edge detection programs available to us for evaluation. We would also like to thank Vic Nalwa for his input regarding parameter selection and his comments on the modification of his algorithm to add nonmaximal suppression and hysteresis. This work was supported in part by a NASA Florida Space Grant Consortium graduate fellowship.

REFERENCES

[1] L.G. Roberts, "Machine Perception of Three-Dimensional Solids," *Optical and Electro-Optical Information Processing*, J.T. Tippett, D.A. Berkowitz, L.C. Clapp, C.J. Koester, and A. Vanderburgh, Jr., eds., pp. 159-197. Cambridge, Mass.: MIT Press, 1965.

[2] I.E. Sobel, *Camera Models and Machine Perception*, PhD thesis, Stanford Univ., 1970.

[3] K.L. Boyer and S. Sarkar, "Assessing the State of the Art in Edge Detection: 1992," *SPIE*, vol. 1,708, *Applications of Artificial Intelligence X: Machine Vision and Robotics*, pp. 353-362, 1992.

[4] Y.T. Zhou, V. Venkateshwar, and R. Chellappa, "Edge Detection and Linear Feature Extraction Using a 2D Random Field Model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 84-95, Jan. 1989.

[5] L. Cinque, C. Guerra, and S. Levialdi, "Reply: On the Paper by R. M. Haralick," *CVGIP: Image Understanding*, vol. 60, no. 2, pp. 250-252, Sept. 1994.

[6] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of Edge Detectors: A Methodology and Initial Study," *Computer Vision and Pattern Recognition*, San Francisco, June 1996.

[7] V.S. Nalwa, *A Guided Tour of Computer Vision*. Reading, Mass: Addison-Wesley Publishing Company, 1993.

[8] V.S. Nalwa and T.O. Binford, "On Detecting Edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 699-714, Nov. 1986.

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

[10] S. Sarkar and K.L. Boyer, "Optimal Infinite Impulse Response Zero Crossing Based Edge Detectors," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 54, pp. 224-243, Sept. 1991.

[11] L.A. Iverson and S.W. Zucker, "Logical/Linear Operators for Image Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 982-996, Oct. 1995.

[12] F. vander Heijden, "Edge and Line Feature Extraction Based on Covariance Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 16-33, Jan. 1995.

[13] K.R. Rao and J. Ben-Arie, "Optimal Edge Detection Using Expansion Matching and Restoration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 12, pp. 1,169-1,182, Dec. 1994.

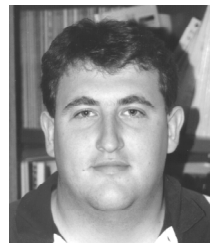
[14] P.H. Gregson, "Using Angular Dispersion of Gradient Direction for Detecting Edge Ribbons," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 682-696, 1993.

[15] M. Gokmen and C.C. Li, "Edge Detection and Surface Reconstruction Using Refined Regularization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 492-498, May 1993.

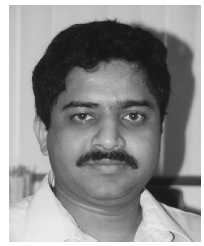
[16] D. Mintz, "Robust Consensus Based Edge Detection," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 59, pp. 137-153, Mar. 1994.

[17] S. Zhang and R. Mehrotra, "A Zero Crossing Based Optimal 3D Edge Detector," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 59, pp. 242-253, Mar. 1994.

[18] J. Shen, "Multi-Edge Detection by Isotropical 2-D ISEF Cascade," *Pattern Recognition*, vol. 28, no. 12, pp. 1,871-1,885, 1995.
[19] P.J. Tadrous, "A Simple and Sensitive Method for Directed Edge Detection," *Pattern Recognition*, vol. 28, no. 10, pp. 1,575-1,586, 1995.
[20] T.N. Tan, "Texture Edge Detection by Modeling Visual Cortical Channels," *Pattern Recognition*, vol. 28, no. 9, pp. 1,283-1,298, 1995.
[21] J. Shen and W. Shen, "Image Smoothing and Edge Detection by Hermite Integration," *Pattern Recognition*, vol. 28, no. 9, pp. 1,159-1,166, 1995.
[22] D.J. Park, K.N. Nam, and R.H. Park, "Multiresolution Edge Detection Techniques," *Pattern Recognition*, vol. 28, no. 1, pp. 211-229, 1995.
[23] A.A. Farag and E.J. Delp, "Edge Linking by Sequential Search," *Pattern Recognition*, vol. 28, no. 5, pp. 611-633, 1995.
[24] J. Shen and W. Shen, "Image Smoothing and Edge Detection by Hermite Integration," *Pattern Recognition*, vol. 28, no. 8, pp. 1,159-1,166, 1995.
[25] V. Srinivasan, "Edge Detection Using Neural Networks," *Pattern Recognition*, vol. 27, no. 12, pp. 1,653-1,662, 1994.
[26] S.M. Bhandankar, Y. Zhang, and W.D. Potter, "An Edge Detection Technique Using Genetic Algorithm Based Optimization," *Pattern Recognition*, vol. 27, no. 9, pp. 1,159-1,180, 1994.
[27] D.J. Park, K.M. Nam, and R.H. Park, "Edge Detection in Noisy Images Based on the Co-Occurrence Matrix," *Pattern Recognition*, vol. 27, pp. 765-775, June 1994.
[28] W.E. Higgins and C. Hsu, "Edge Detection Using 2D Local Structure Information," *Pattern Recognition*, vol. 27, no. 2, pp. 277-294, 1994.
[29] W.B. Thompson, P. Lechleider, and E.R. Stuck, "Detectng Moving Objects Using the Rigidity Constraint," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, Feb. 1993.
[30] D. Ziou and S. Tabbone, "A Multiscale Edge Detector," *Pattern Recognition*, vol. 26, no. 9, pp. 1,305-1,314, 1993.
[31] E. Chuang and D. Sher, "Chi-Square Test for Feature Detection," *Pattern Recognition*, vol. 26, no. 11, pp. 1,673-1,682, 1993.
[32] I.E. Abdou and W.K. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Proc. IEEE*, vol. 67, no. 5, pp. 753-763, May 1979.
[33] V. Ramesh and R.M. Haralick, "Performance Characterization of Edge Detectors," *SPIE*, vol. 1,708, *Applications of Artificial Intelligence X: Machine Vision and Robotics*, pp. 252-266, 1992.
[34] J.R. Fram and E.S. Deutsch, "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison With Human Performance," *IEEE Trans. Computers*, vol. 24, no. 6, pp. 616-628, June 1975.
[35] D.J. Bryant and D.W. Bouldin, "Evaluation of Edge Operators Using Relative and Absolute Grading," *Proc. IEEE Computer Society Conf. Pattern Recognition and Image Processing*, pp. 138-145, Chicago, 1979.
[36] R.N. Strickland and D.K. Cheng, "Adaptable Edge Quality Metric," *Optical Eng.*, vol. 32, no. 5, pp. 944-951, May 1993.
[37] X.Y. Jiang, A. Hoover, G. Jean-Baptiste, D. Goldgof, K. Bowyer, and H. Bunke, "A Methodology for Evaluating Edge Detection Techniques for Range Images," *Proc. Asian Conf. Computer Vision*, pp. 415-419, 1995.
[38] T. Kanungo, M.Y. Jaisimha, J. Palmer, and R.M. Haralick, "A Methodology for Quantitative Performance Evaluation of Detection Algorithms," *IEEE Trans. Image Processing*, vol. 4, no. 12, pp. 1,667-1,674, Dec. 1995.
[39] L. Kitchen and A. Rosenfeld, "Edge Evaluation Using Local Edge Coherence," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 11, no. 9, pp. 597-605, Sept. 1981.
[40] Q. Zhu, "Efficient Evaluations of Edge Connectivity and Width Uniformity," *Image and Vision Computing*, vol. 14, pp. 21-34, 1996.
[41] P.L. Palmer, H. Dabis, and J. Kittler, "A Performance Measure for Boundary Detection Algorithms," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 476-494, May 1996.
[42] K. Cho, P. Meer, and J. Cabrera, "Quantitative Evaluation of Performance Through Bootstrapping: Edge Detection," *IEEE Int'l Symp. Computer Vision*, pp. 491-496, Coral Gables, Fla., Nov. 1996.
[43] F. Bergholm, "Edge Focusing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 6, pp. 726-741, Nov. 1987.
[44] C.A. Rothwell, J.L. Mundy, W. Hoffman, and V.-D. Nguyen, "Driving Vision by Topology," *Int'l Symp. Computer Vision*, pp. 395-400, Coral Gables, Fla., Nov. 1995.
[45] P.E. Shrout and J.L. Fleiss, "Intraclass Correlation: Uses in Assessing Rater Reliability," *Psychology Bulletin*, vol. 86, no. 2, pp. 420-428, 1979.
[46] R.E. Walpole and R.H. Myers, *Probability and Statistics for Scientists and Engineers*, third ed., chap. 11-13. New York: Macmillan Publishing Company, 1985.
[47] G. Keppel, *Design of Analysis*. Englewood Cliffs, N.J.: Prentice Hall, 1991.



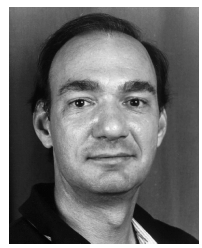
Michael D. Heath received his BS in imaging science from Rochester Institute of Technology, Rochester, N.Y., in 1992 and an MS in computer science and engineering from the University of South Florida in 1996. After completing his BS degree, he joined the Eastman Kodak Company, Rochester. He is currently a PhD candidate in the Department of Computer Science and Engineering at the University of South Florida, Tampa. He is the recipient of the 1996 University Graduate Fellowship and the NASA Florida Space Grant Fellowship.



Sudeep Sarkar received the BTech degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1988, where he was judged the best graduating electrical engineer. He received the MS and PhD degrees in electrical engineering from the Ohio State University, Columbus in 1990 and 1993, respectively. He is currently an assistant professor in the Department of Computer Science and Engineering at the University of South Florida, Tampa. He is the recipient of the U.S. National Science Foundation CAREER award in 1994 and the Teaching Incentive Program Award for undergraduate teaching excellence in 1997. He is the coauthor of the book *Computing Perceptual Organization in Computer Vision*, published by World Scientific (1993).

His research interests include low-level image segmentation, perceptual organization in single- and multiple-image sequences, probabilistic reasoning, color-texture analysis, nonrigid body modeling, and performance evaluation of vision systems. His recent research projects are listed at `http://marathon.csee.usf.edu/~sarkar/sarkar.html`.

Thomas Sanocki received a BS in psychology from Northern Michigan University and a PhD in cognitive psychology from the University of Wisconsin-Madison in 1986. Since then, he has been on the psychology faculty at the University of South Florida, where he is currently an associate professor. He has published numerous research articles on human perception of letters, words, objects, and scenic layout.



Kevin W. Bowyer completed his bachelor's degree at George Mason University and his PhD in computer science at Duke University. He is currently a professor in the Department of Computer Science and Engineering at the University of South Florida. Prior to joining USF, he was a member of the faculty of the Department of Computer Science at Duke University and then a member of the faculty of the Institute for Informatics at the Swiss Federal Technical Institute (Zurich). Professor Bowyer's current research interests are in the general areas of image understanding, pattern recognition, and medical image analysis. Professor Bowyer received an Outstanding Undergraduate Teaching Award from the USF College of Engineering in 1991 and Teaching Incentive Program Awards in 1994 and 1997. Professor Bowyer serves as North American Editor of the *Image and Vision Computing Journal*, and is a member of the editorial boards of *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Computer Vision and Image Understanding*, *Machine Vision & Applications*, and the *International Journal of Pattern Recognition and Artificial Intelligence*. He is author of the book *Ethics and Computing*, published by IEEE Computer Society Press (1995); coauthor, with Louise Stark, of the book *Generic Object Recognition Using Form and Function*, published by World Scientific (1996); coeditor, with Narendra Ahuja, of the book *Advances in Image Understanding*, published by IEEE Computer Society Press (1996); and coeditor, with Sue Astley, of *State of the Art in Mammographic Image Analysis*, published by World Scientific (1994).