Summer 2010

Multi-Scale Edge Detection Algorithms and Their Information-Theoretic Analysis in the Context of Visual Communication

Bo Jiang
Old Dominion University

MULTI-SCALE EDGE DETECTION ALGORITHMS AND THEIR INFORMATION-THEORETIC ANALYSIS IN THE CONTEXT OF VISUAL COMMUNICATION

by

Bo Jiang
B. S. June 2004, Chengdu University of Technology
M. S. June 2007, University of Electronic Science and Technology of China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2010

Approved by:

Zia-ur Rahman (Director)

Frederic McKenzie (Member)

Jiang Li (Member)

Roland Lawrence (Member)

ABSTRACT

MULTI-SCALE EDGE DETECTION ALGORITHMS AND THEIR INFORMATION-THEORETIC ANALYSIS IN THE CONTEXT OF VISUAL COMMUNICATION

Bo Jiang
Old Dominion University, 2010
Director: Dr. Zia-ur Rahman

The unrealistic assumption that noise can be modeled as independent, additive and uniform can lead to problems when edge detection methods are applied to low signal-to-noise ratio (SNR) images. The main reason for this is because the filter scale and the threshold for the gradient are difficult to determine at a regional or local scale when the noise estimate is on a global scale. Therefore, in this dissertation, we attempt to solve these problems by using more than one filter to detect the edges and discarding the global thresholding method in the edge discrimination. The proposed multi-scale edge detection algorithms utilize the multi-scale description to detect and localize edges. Furthermore, instead of using the single default global threshold, a local dynamic threshold is introduced to discriminate between edges and non-edges. The proposed algorithms also perform connectivity analysis on edge maps to ensure that small, disconnected edges are removed. Experiments where the methods are applied to a sequence of images of the same scene with different SNRs show the methods to be robust to noise. Additionally, a new noise reduction algorithm based on the multi-scale edge analysis is proposed. In general, an edge—high frequency information in an image—would be filtered or suppressed after image smoothing. With the help of multi-scale edge detection algorithms, the overall edge structure of the original image could be preserved when only the isolated edge information that represents noise gets filtered out. Experimental results show that this method is robust to high levels of noise, correctly preserving the edges. We also propose a new method for evaluating the performance of edge detection algorithms. It is based on information-theoretic analysis of the edge detection algorithms in the context of an end-to-end visual communication channel. We use the information between the scene and the output of the edge-detection algorithm, ala Shannon, to evaluate the performance. An edge detection algorithm is considered to have high performance only if the information rate from the scene to the edge approaches the maximum possible. Therefore, this information-theoretic analysis becomes a new method to allow comparison between different edge detection operators for a given end-to-end image processing system.

To Mom, Dad, and my families
for their endless love and support.

ACKNOWLEDGEMENTS

I would like to thank everybody who was important to the successful realization of my dissertation, as well as apologize that I could not mention personally one by one.

First and foremost, I would like to thank my advisor, Dr. Zia-ur Rahman, for his guidance, knowledge, enthusiasm and support. I am honored to be one of his Ph.D. students and learned many things from him, especially in my research, which will always be a treasure in my whole life. I appreciate his help and support.

Also, I would like to thank my dissertation committee, Dr. Frederic McKenzie, Dr. Jiang Li, and Dr. Roland Lawrence, for their suggestive comments on this dissertation and full support of my research work.

I need to thank all current and previous members of Computational Intelligence & Machine Vision Laboratory for their friendship.

I also need to thank all my friends, colleagues and staff, Yanxiao, Rui, Jianqiao, Jie, Xiaoyan, Yufei, Romina, and Linda, in the Department of Electrical and Computer Engineering who helped me and rendered me precious memories and experiences.

To those friends who are living or have lived in Norfolk, Hui, Jianhu, Linna, Luni, Anup and Greg, I am so glad to recognize you and appreciate your help and friendship. You made my time and gave me extraordinary and remarkable memories.

To those friends in China, Jieyi, Weijun, Huan, and Liping, I am so lucky to know you. Thank you for your support and care.

Last, I would like to thank my parents, Kunfu, Tongqing, and my families for their endless love and support. Without their support, I can't make it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

I.1 MOTIVATION AND BACKGROUND

I.1.1 Edge Detection

Edge detection is an integral part of many digital image processing tasks. In general, the edge detection operation has two main steps: filtering and localization. Depending upon the final application, different algorithms implement these steps differently. The most commonly used edge-detection methods compute edges from the derivative of the intensity values. This localizes edges at pixels where intensity transitions occur. Most first derivative operators, e.g., Roberts [1], Sobel [2], or Prewitt [3] methods, are isotropic. The edges are defined at the point that has the maximum magnitude in the gradient direction. A threshold on the gradient magnitude is often used to eliminate weak edges. However, finding the optimal threshold is an ill-posed problem, especially when a single—global—threshold is used over the entire edge map of a noisy image. Even when the gradient threshold is based on noise estimation [4], it may be large in smooth areas or produce low contrast edges, due to the noise. Consequently, thresholding the gradient can lead to many errors.

To compensate for the threshold problem, one approach is to suppress the non-maximum point in the gradient direction [5]. This is equivalent to computing the second

This dissertation follows the style of *IEEE Transactions.*

derivative in the direction of the gradient and looking for the zero-crossing points that correspond to the local maximum point of the first derivative. Haralick [6] was the first to use zero-crossing to detect edges. The Laplacian operator [7] is the simplest second derivative operator. However, these methods are not always effective for noisy images because the derivative approach is very sensitive to noise [7]. Generally, smoothing filters are used in regularization techniques to make differentiation more immune to noise. Marr and Hildreth [8] combined the Gaussian filter and the Laplacian operator to construct the Laplacian-of-Gaussian (LoG) operator, which is quite effective. However, the optimal width of the Gaussian is hard to find and is image dependent [9]. Canny [5] also utilized the Gaussian filter to smooth the image.

The LoG and Canny operators show that the Gaussian filter is close to the optimal pre-filter for edge-detection. The Gaussian filter has been applied in multi-scale image analysis. Based on the research on cat and primate vision systems [10], the multi-resolution (multi-scale) description has been suggested to process natural images. In 1971, the importance of multi-scale description of images was recognized by Rosenfield and Thurston [11]. They proposed an edge and curve detection method by using different sizes (scales) of Gaussian filters. Then, Witkin [12] proposed scale-space filtering by Gaussian filters to smooth an image and detect its edges. This method is used to reduce noise while protecting features. The scale of the Gaussian filter could be considered as a continuous parameter, thereby generalizing the existing notion of Gaussian pyramids, which has been further generalized into scale-space theory [13].

In 1987, multi-resolution theory based on wavelet transform was first shown to be the foundation of a powerful new approach to signal processing and analysis [7]. This approach is concerned with the representation and analysis of signals (or images) at more than one resolution. The appeal of such an approach is obvious—features that might go undetected at one resolution may be easy to detect at another. Mallat and Zhong [14] expanded the field of multi-scale analysis to edge detection. They related multi-scale edge detection with the discrete wavelet transform (DWT). Beltran et al. [15] replaced the DWT with Gaussian filters and proved that multi-scale detection can be performed using Gaussian filters if the parameters are properly selected.

Multi-scale edge processing can avoid many of the problems of the edge-detectors that use a single filter. For example, a filter with a single global scale might under-smooth areas of high noise but over-smooth less noisy areas, while one with different threshold values at different scales may be able to avoid this problem. Additionally, different scales represent different details and, hence, different levels of edge significance. Thus, multi-scale analysis turns out to be a very useful tool for edge-detection and analysis.

I.1.2 Noise Reduction

The impact of noise on image quality in scenes acquired under poor visibility conditions is quite significant. Additionally, the presence of noise also significantly restricts how the image can be used for computer vision and pattern recognition applications. For this reason, noise reduction in digital images has been an active topic of research in recent years. Several different approaches have been used for noise reduction, and they encompass a wide variety of processing methods. Linear filtering methods such as low-pass (smoothing) filter techniques assume that noise (mostly) is comprised of high spatial-frequency components in the spatial-frequency representation of an image. However, the reduction of noise using such techniques also often leads to the suppression of significant edge features. Since the presence of edges is perceptually related to sharpness and, hence, contrast [16], these approaches impact image quality significantly. There are smoothing methods that can also preserve edges, and Winkler et al. [17] provide a good synopsis of such methods.

Several techniques for noise reduction have been based on the analysis of systems used to generate digital images, e.g., scanners and digital cameras. This system analysis allows one to develop image restoration filters [18, 19, 20, 21, 22] that take into account the different sources of noise and attempt to reduce their impact on the output image.

Other researchers have used edge-preserving, non-linear filters like the median filter [7, pp. 156–157]. While such techniques are quite useful, they have significant shortcomings in the presence of a high level of noise due to the static nature of the filter extent. For this reason, several adaptive methods for noise reduction that preserve edges have also been developed in the literature. These are usually variants on the median filter, such as the adaptive median filters (AMF) [23, 24] and the adaptive threshold median filter (ATMF) [25] for speckle and salt and pepper noise reduction, but they do not work as well for additive white Gaussian noise. Lian et al. [26] and Smolka et al. [27] have extended the idea of adaptive median-like filtering to color images. Hamza and Kim [28] use robust estimation techniques to derive non-linear filters that can be used to denoise both impulse and Gaussian noise. Adelmann [29] presents an algorithm that uses the idea of non-maximal suppression for edge preservation. Witkin [12] proposed scale-space filtering by Gaussian filters for noise reduction and feature preservation. To protect edges during smoothing, Perona and Malik [30] extended scale-space filtering by anisotropic diffusion. Lu et al. [31] proposed a new anisotropic diffusion based on proportional-integral-derivative (PID) control law together with a stopping mechanism. There are several wavelet based noise-reduction algorithms derived from the work done by Donoho and Johnstone [32]. The algorithm by Zhan and Karam [33] uses correlation analysis on the dyadic wavelet decomposed image to form noise and non-noise features and uses this information for noise reduction. Van De Ville et al. [34] present a fuzzy image filtering approach that uses the idea of fuzzy derivatives and fuzzy smoothing to reduce the impact of noise. Peters [35] develops an algorithm for edge-preserving noise reduction using mathematical morphology. Several researchers have shown the importance of using edge primitives as a basis for recognition in visual perception [36, 37]. This edge pattern analysis can be used for both automatic assessment of spatially variable noise and as a foundation for new noise reduction methods [38].

Recently, the trend in edge-detection has been moving away from using neighborhood pixel differences to estimate local derivatives for detecting intensity changes, i.e., edges. More attention is being paid to edge feature analysis and, based on this, in trying to design new and effective noise reduction methods. Edges can be divided into basic categories [39, 40] such as ramp, step, stair, and pulse: different types have different shapes. These edges can be filtered with a Gaussian to estimate their slope [15, 41]. Because noise can generally be assumed to be independent of signal, have little regional connectivity, and have random orientation, its estimate would be small under a Gaussian filter. Furthermore, it has been shown that the Gaussian is close to the optimal operator for different edges [5]. This kind of image analysis can be used as the basis of a promising solution for image denoising. If we divide an edge image into signal and noise pixels based upon local edge analysis, then we can apply different filters to the signal pixels and the noise pixels, reducing the overall impact of noise on image quality [42].

I.1.3 Edge Detection Evaluation

Research on edge detection methods has been going on for more than 40 years and encompasses everything from well known, traditional methods such as Roberts, Sobel, Laplacian-of-Gaussian (LoG), Canny, scale-space, etc., to some of the latest methods based on or combining techniques such as nonlinear derivatives [43], fuzzy sets [44], neural networks [45], wavelets [46], and so on. All of these methods have their pros and cons. In many cases the algorithms are designed for some specific application and perform extremely well for that particular application. However, their performance is application dependent and, hence, *not* general. Additionally, until now there has not been a common quantitative metric that allows one to judge the effectiveness of edge detection methods. Generally, qualitative analysis is used. As Pal and Pal [47] point out, people are considered to be the ultimate judge in making an evaluation of the edge detection result. However, this is not a practical solution for automated systems that rely on edge detection; in those cases it is much better to have a universal metric/environment that allows one to measure how well an edge detection algorithm performs for a class of images.

To objectively measure the performance of an edge detection algorithm, several authors [48, 49, 50, 51, 52, 53, 54] have proposed performance measures to evaluate the output of edge detectors. Abdou and Pratt [48] proposed a figure of merit which is a combination of three factors: (i) non-detection of true edges, (ii) detection of false edges, and (iii) edge localization error. However, it is hard to measure the error based on this metric because complete information about the true edges is required. Also, as pointed out by Peli and Malah [55], this figure of merit does not consider the performance of edge detector in terms of factors such as edge thickness and edge continuation. Michelli et al. [49] analyzed edge detectors based on the accuracy of edge localization and sensitivity to noise, but the problem with their approach is that the analysis was performed for the case where the SNR is good. Thus, the noise did not impact the performance of the algorithm in a significant way. In actuality, the low SNR case must not be neglected. Kitchen and Rosenfeld [51] evaluated edge detectors using edge coherence, which measures the continuation and thinness of the detected edges. Fram and Deustch [50] developed a quantitative measure which was used to evaluate detected true edges in the presence of noise. Heath et al. [52] proposed an evaluation method by combining subjective and objective evaluations. Humans often compare different edge detectors based on the significance of observed differences.

I.1.4 Information-theoretic Analysis

In fact, even today the trend in digital image processing is still to focus on narrowly defined tasks. For example, image restoration and enhancement are researched as independent processes. While image restoration and enhancement have been shown to improve, often dramatically, the quality of degraded images to clearly reveal what could perhaps be barely discerned before, experiments show that these processes are not independent of the image gathering and display devices. Hence, image processing algorithms should be analyzed in the context of the end-to-end imaging system that takes into account the image gathering and display processes as well as the processing algorithms. For example, an improperly designed image restoration filter can lead to image artifacts such as enhanced aliasing noise caused by the image gathering and display devices. Thus, it is often unclear whether the quality of the improved images actually approaches the optimal and, hence, what further improvements could be made.

As Gabor [56] said: “Experiments unguided by theory do not appear very promising.” Huck, et al. [57] proposed a definitive analysis of visual communication channels, where the assumed independent parts such as image gathering, display devices, and the digital image processing for image coding and restoration are analyzed and assessed using an integrated platform (Figure 1) by extending Shannon’s information theory. They performed an end-to-end, information theory based system analysis to assess image restoration methods. They evaluated the performance of the different algorithms as a function of the characteristics of the scene, and the parameters, such as inter-sample distance, additive noise etc., that define the image gathering system. The image restoration algorithm is regarded to have high performance only if the information rate from the scene to the display approaches the maximum possible. This goal can be achieved only by jointly optimizing all processes.

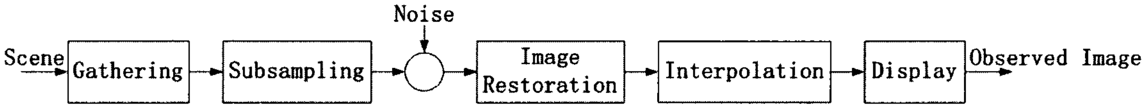

FIG. 1: Model of image gathering and display with digital processing and interpolation.

The mathematical development of the end-to-end analysis proposed by Huck et al. [58, 59] is based on the two classical works that can be considered to be the foundation of modern communication theory. The first, by Claude Shannon [60], introduces the concept of the rate of transmission of information in a noisy channel; the second, by Norbert Wiener [61], introduces the concept of the minimum mean-square error restoration of signals corrupted by noise. Huck et al. [57] combine these two concepts and provide the mathematical formulations that describe, for a Gaussian signal, the relationships that exist among information rate, theoretical minimum data rate and maximum-realizable fidelity. In that system analysis, they put the electro-optical design of image gathering and the digital processing for image coding, restoration and display devices together. Thus, this system or channel would be considered as high quality only if the information rate from the scene to the observer approaches the maximum possible and the required data rate approaches the minimum possible. Thus, the goal would be only achieved by jointly optimizing all those parts. These relationships, which are affected by image gathering and display device responses, allow a quantitative assessment of the visual communication channel which includes such processes as edge detection.

I.2 PROPOSED SOLUTIONS

In the following sections, we describe our approach to solving many of the problems that impact edge-detection and noise reduction in the presence of a high degree of noise, i.e. for the low SNR case.

I.2.1 Multi-scale Edge Detection

In this dissertation, we present a new approach to edge detection that is based on higher derivatives of the intensity image. Compared with the approach that estimates local derivatives by neighboring pixel differences, we modify the edge detection method presented by Beltran [15] that uses a multi-scale analysis. Because edge features due to just the signal can generally be assumed to have regional connectivity and specific orientations, we can separate the signal features from noise by exploiting this idea. Prominent edges should have features at all scales while features due to noise should be regional and disappear at certain scales. Hence, by analyzing the generated scale-space, the edge features can be recovered. Furthermore, in the edge discrimination process, a local dynamic threshold generated by estimating local noise is introduced to avoid the under- and over-smoothing problems that can occur with global thresholding. In addition to the multi-scale analysis, we also perform connectivity analysis on the edge map to eliminate features that exist across scales but are of relatively small extent. These signal features are also associated with noise in scenes with heavy noise. The idea of connectivity analysis presented by Rahman and Jobson [42] also relies on the connectivity characteristics of noise and features, as stated above. Therefore, using the idea of regional connectivity, Rahman and Jobson classify pixels on edges that have lengths smaller than a given threshold as "noise." The algorithm proposed in this thesis performs extended connectivity analysis on the edge-map to make sure that only features smaller than a predefined edge-length, i.e., features that only represent noise, get filtered out.

I.2.2 Image Denoising

The multi-scale edge detection algorithm allows the edges due to the signal and the “non-edges” due to noise, to be differentiated, even when the SNR is very low. This points to one possible method for suppressing noise and preserving signal for noise reduction. As stated previously, the most common way of dealing with additive white Gaussian noise is to apply a low-pass filter to the noisy image. In general, this results in blurring, i.e., suppressing the edge features—high frequency information—in the image, especially when SNR is very low. Thus, while the noise is attenuated, the image loses sharpness and, hence, contrast and clarity. This trade-off between noise reduction and sharpness retention makes making use of the processed image in additional tasks considerably harder. To compensate for this drawback Kao and Chen [62], for example, add an edge preserving stage in their noise reduction approach. Consequently, in this dissertation, we combine the proposed edge detection algorithm with connectivity analysis to detect and mitigate noise. In the smoothing process, only the edges that have lengths smaller than a given threshold are classified as “noise,” and pixels at those locations in the original image are replaced with an average of their neighbors. This reduces the impact of noise at that location while preserving the overall edge structure of the original image because only locations with edges due to the noise are blurred.

I.2.3 Information-theoretic Analysis

All the methods mentioned in Section I.1.3 measure edge detector performance without consideration of the properties of the visual communication system, such as the image-gathering device response, sampling and transmission rates, etc., that affect the input images. Inspired by Huck et al. [63, 58, 59] we analyze and compare the edge detection methods using information-theoretic analysis in the context of the end-to-end imaging system. The edge detection process analysis is used within the end-to-end system model to allow quantitative analysis of the algorithms. We analyze the performance of the edge detectors in terms of the amount of the mutual information between the original scene and the edge image. If the image-gathering and the communication environment parameters are fixed for all the edge detection operators, then any variation in the amount of information being transmitted is directly attributable to the edge-detection algorithm. Hence, the performance of the algorithms can be measured for a given set of system parameters. In order to use mutual information as a metric for evaluating the performance of different edge detection methods, we need their associated power spectral densities (PSDs). Therefore, we first derive the PSDs for the traditional edge detection methods, namely the Sobel, Prewitt, Roberts, the LoG and the Canny algorithms, and then use them in our information-theoretic evaluation.

I.3 DISSERTATION OUTLINE

The remaining parts of this dissertation are organized as follows. The multi-scale edge detection algorithm design and description is provided in Chapter II. Additionally, the mechanics of the connectivity analysis that is used to perform edge-preservation and non-edge reduction are described, and the results of the application of the proposed edge detection algorithms on noise reduction are shown. In Chapter III, the fundamentals of an end-to-end digital image processing system in the visual communication channel are reviewed. Inspired by this system, an information-theoretic analysis based on Shannon's information theory is described. Then, the proposed edge detection algorithm evaluation mechanism is presented. In Chapter IV, experimental results and analysis of combining the various edge-detection algorithms with connectivity analysis for edge detection and noise reduction are shown. Test images are drawn from both computer generated images and real/natural images. A performance comparison using the image fidelity metric is also presented. Chapter V provides the results of the information-theoretic analysis. First, the PSDs of traditional edge detectors are derived by theoretical and experimental methods. Then, by setting different parameters of the visual communication channel, such as those controlling the mean spatial detail in the scene, the blur of image acquisition device, system noise, etc., the performance of various edge detectors are theoretically evaluated and compared. Finally, conclusions and future work are given in Chapter VI.

CHAPTER II

MULTI-SCALE EDGE DETECTION ALGORITHMS AND NOISE REDUCTION

In this chapter, we describe the design and assessment of the multi-scale edge detection algorithms. We then use these algorithms in conjunction with connectivity analysis to devise a noise-reduction process.

II.1 MULTI-SCALE EDGE DETECTION

The multi-scale edge detection process for low SNR imagery is made up of several operations. These include the selection and design of the scale-space (II.1.1), edge localization (II.1.2), multi-scale analysis (II.1.3), local threshold assessment based on a local noise estimate (II.1.4), connectivity analysis for differentiating between edges due to the signal and edges due to the noise (reduction II.1.5), and noise reduction (II.1.6) based on all of these operations.

II.1.1 Scale-space Design

Once an intensity image has been filtered with a Gaussian, the high frequency information—edges and noise—are both attenuated. An important consideration in the design of the filter is the scale of the Gaussian: small scales let more noise and edges through, while larger scales suppress both noise and edges. This trade-off between noise reduction and sharpness retention impacts edge detection and localization. The optimal selection of the scale is still an open question. Jeong and Kim [64] proposed an adaptive method to determine the global Gaussian filter scale, but relying on a single global scale might lead to errors because it might be large where the intensity changes slowly and small at areas of sudden intensity changes. Lindeberg [65] pointed out that the "right scale" does indeed seem to be problem dependent. Since multi-scale analysis uses a combination of Gaussian filters with differing scales, this can ameliorate the problem, but the selection of scales that provide good edge-detection capabilities in the presence of different SNR values is still tricky. We conducted a series of experiments to find a set of scales that balances the trade-off between noise reduction and feature preservation [66].

II.1.2 Edge Detection and Localization

In a recent paper [66], we showed that multi-scale analysis is a good and simple method to detect edges. However, we used only the gradient and its orientation to classify edge and non-edge pixels. Furthermore, in noisy or blurred conditions, the edge detector found more than one local maximum gradient along the cross-section of the edge, so it was difficult to localize the edge. Thus, we propose a modification of our previous approach and additionally use the zero-crossings of the second derivative. The zero-crossings have been shown to be a sufficient statistical analysis model to detect and localize edges [4]. Thus, in order for a pixel to be classified as an edge, it must meet two criteria: (1) the gradient magnitude should be significantly greater than zero, and (2) there should be a significant zero-crossing in the second derivative in the direction of the gradient.

As pointed out by Clark [67], the edges after Gaussian filtering that correspond with zero-crossings can be divided into two groups: authentic and phantom. To differentiate between them, we also require that the authentic edge should satisfy the condition that the product of its first and third derivatives is negative. Since noise causes zero-crossings to occur in locations where edges due to signals do not exist, we still use gradient magnitude and orientation to eliminate spurious zero-crossings.

II.1.3 Multi-scale Analysis

There is no standard definition of multi-scale analysis (MSA) for edge-detection, so the MSA has been used in myriad ways. Edge focusing [68] uses the notion of coarse-to-fine tracking. The premise of this approach is to select the really significant events, i.e., those that survive at the largest scales and then track these events through scale-space as the scale is decreased. The localization of these events at the smaller scales accurately localizes the edges. Estrada [69] also developed a coarse-to-fine method using Bayesian probabilistic methods. The candidate contours are extracted at a coarser scale and then used as priors on the location of possible contour at finer scale. Aside from the computational complexity of these approaches, it is probably not appropriate to use only the large scale of scale-space to judge edge significance. Also, the assumption that optimal localization accuracy can be attained at the smallest scale is not rational. For example, if noise is high, localization accuracy can be very poor at smaller scales. Instead of using the coarse-to-fine tracking, we use the logical AND operation on the derivative images at one or more scales to find the edge.

The multi-scale edge detection method described in Section II.2.1 uses six scales where each scale is double the size of the previous scale. Using images from all six scales typically results in very thick edges because of the heavy blurring associated with large values of j in Eq. 5. Experimental results led to the development of a 3-of-6 rule in which we use a combination of any three out of the six possible scales to generate the output image. Different results can be obtained depending upon which three scales are used. Small scales give finer edges but are more prone to letting noise through. Larger scales eliminate the noise but produce thick edges. Once the three optimal scales have been determined, a second variation results in producing more robust edge results. In this variation, we accept a pixel to be an edge pixel if there is a sufficiently large value at that location in at least two of the three scales. We call this the 2-of-3 rule. The idea of "sufficiently large" is related to the threshold of the magnitude which is described in a later section.

One important reason to utilize the logical AND operation is that the signal is assumed to be strong enough to exist over all scales, but the edge can't always be assumed to have high contrast and sharpness. Some long edges may be blurrier and of lower contrast than relatively short edges. The goal of edge detection should be to detect all edges in the image, over the broad range of contrasts and blur scales with which they occur [9]. This is one of the reasons for the 2-of-3 rule. When the signal is strong enough, such as that due to the main structures, it would survive at almost all scales and would be easily be detected with the logical AND. However, weak signals, such as minor details, do not survive across the whole scale-space: they probably only survive at some scales. Hence, by using the 2-of-3 rule, we can take advantage of the characteristics of small and large scales.

II.1.4 Local Noise Estimate

Although we found the regularity of threshold for edge detection corresponding to its specific SNR [66], the threshold set as global one for a whole image is still an ill-posed problem. It means that the details such as edges and noise are uniformly distributed across the whole image. For real images, however, both the signal and the noise can vary from area to area, even from pixel to pixel. Thus, using a local, dynamic threshold might be a better solution.

Elder and Zucker [9] proved the threshold for asserting a non-zero gradient to be

$$s_1(\sigma_1) = 3\sigma_n/\sqrt{\pi}\sigma_1^2, \tag{1}$$

where σ_n is the standard deviation of the noise, and σ_1 is the scale of the Gaussian first derivative filter. The prior computation of a significance function s_1, which determines the lower bound on the smoothing kernel, needed to reliably assert a non-zero gradient. Thus, the pixel would be assigned as an edge if its gradient is larger than $s_1(\sigma_1)$.

Elder and Zucker [9] use the significance function to estimate the minimum reliable scale for asserting a non-zero gradient. In other words, the significant function can be used as a guide for finding a reliable gradient which can differentiate between edges and non-edges. While Elder and Zucker use Eq. 1 to determine a global threshold based on image characteristics, we extended this idea to compute the local dynamic threshold. Since we know the scales of the Gaussian filters that are used in the multi-scale edge detection algorithm, the standard deviation of the local noise can be estimated.

One potential problem in the local scale control presented by Elder and Zucker as pointed out by Liang and Wang [70] is that the noise is assumed as uniformly distributed in the whole area and uncorrelated with the image signal. Thus, the same noise variance σ_n would be utilized for the entire image. If a global noise variance is used in the local area, the area with less noise would be over-smoothed, while the area with more noise would be under-smoothed. Also, for 2D images, the noise could possibly vary between neighboring pixels at different orientations.

To compensate for this problem and to make the threshold adaptive to the amount of noise instead of being fixed, it is necessary to estimate the noise for each pixel in every direction. Olsen [71] gave a complete description and comparison of some earlier estimation algorithms. They are classified into two different approaches: filter-based (or smoothing based) and block-based. Some recent methods are proposed by Tai and Yang [72], Shin et al. [73], and Liu et al. [74]. However, these more recent algorithms can also be categorized as filter-based, block-based, or a combination of the two. Compared to global noise estimation, local noise estimation has not so far been researched extensively.

Liang and Wang [70] proposed a local noise estimate, but experiments show that it is not appropriate for our edge detection algorithm. We developed a local noise estimate by experimentation. Unlike Liang and Wang's method, we examine every scale to estimate the local noise. Examining each scale is more accurate than analyzing only the original image because the image representations generated at each scale using Gaussian filters of different extents are not linearly related. Our method is based on the number of zero-crossings of the second order derivative in a given direction and the absolute sum of the first derivative, or the amount of unipolar energy, in the same direction:

$$\sigma_{n_j} = \beta \frac{Z(m,n,\phi,\gamma)S(m,n,\phi,\gamma)}{(2\gamma+1)}, \tag{2}$$

where σ_{n_j}, slightly different from σ_n, are the standard deviations of the local noise at the jth scale, β is a predefined positive constant, $Z(m,n,\phi,\gamma)$ is the number of zero-crossings of the second order derivative at (m,n) in the direction ϕ over the distance γ, from $(m,n)-\gamma$

to $(m,n)+\gamma$ and $S(m,n,\phi,\gamma)$ is the absolute sum of first derivative at (m,n) in the same direction and across the same pixels. Thus, the local threshold would be computed for each pixel based on local noise estimate (LNE):

$$\tau_j(\sigma_j,m,n,\phi,\gamma)=\sigma_{n_i}/\sigma_j^2,\quad \phi=0,\pi/2,\pi,3\pi/4, \tag{3}$$

where $\sigma_j=2^{j-1}\sigma_1$, $\sigma_1=2$.

II.1.5 Connectivity Analysis

The connectivity analysis algorithm is very useful for images with low SNR. In these noisy images, some edges, due to noise, survive the multi-scale analysis and detection. Connectivity analysis can be used to preserve the authentic edges in an image by differentiating between edge features due to signal and those due to noise using the connectivity of features as the discriminant. The rationale for this is that edges due to noise will usually be isolated and will form either singular points or very short edges in the edge image. This idea expands on the ideas presented by Jiang and Rahman [66] and Rahman and Jobson [42] by looking at larger connectivities to differentiate between pixels due to noise and those due to signal. Consequently, the connectivity analysis step is adopted to make sure that edge information that is associated with noise gets filtered out, hence preserving the overall edge structure of the original image.

II.1.6 Noise Reduction

In image processing, edges, as one important feature of the image content, would be of great value to the post-processing processes, such as pattern recognition, object detection, image registration, etc. As stated in Chapter I, our starting point is the idea that edges in an image should be preserved in order to preserve significant detail about the structure and the texture in an image. Operations such as image smoothing and median filtering, will impact the contrast of an image, and hence the strength of edges. Therefore, to avoid this problem, the solution is to perform noise reduction operations only on those pixels that are not part of an edge as was proposed by Rahman and Jobson [42].

II.2 DESCRIPTION OF THE ALGORITHMS

In the following sections, we describe our edge detection and noise-reduction algorithms in detail. The multi-scale edge detection algorithms, adapted from Beltran et al. [15], are described in Section II.2.1. Connectivity analysis that is used to perform edge-preserving noise reduction is described in Section II.2.2. Noise reduction, as one application of our proposed edge detection algorithms, is introduced in Section II.2.3.

II.2.1 Multi-scale edge detection

The multi-scale edge-detection algorithm proposed by Beltran et al. [15] uses the idea of the difference-of-Gaussian operator in a slightly different way. Instead of using the difference between two representations of an image obtained by filtering with a Gaussian of different scales (widths), the idea here is to use the logical AND operation on the image at one or more resolution to find the edge. This process can be described using the following steps:

1. Generate the multi-resolution image representations:

$$G_j(m,n) = G(m,n) * F_j(m,n), \quad j = 1,\ldots,6, \tag{4}$$

$$F_j(m,n) = \exp\left(-\frac{m^2+n^2}{\sigma_j^2}\right), \; \sigma_j = 2^{j-1}\sigma_1, \; j = 1,\ldots,6, \tag{5}$$

where $G(m,n)$ are the pixels from the input image, G. G_j represents one image in the resolution pyramid of level j. F_j denotes the Gaussian filter with standard deviation of σ_j. σ_1 is the standard deviation of $F_1(m,n)$ and can be varied depending upon the image under consideration.

2. Each image $G_j, j = 1,\ldots,6$ has associated modulus M_j and phase P_j images. M_j and P_j are computed as:

$$DG_j^x = G_j D_x, \quad D_x = \begin{bmatrix} -1 & 1 \end{bmatrix}, \tag{6}$$

$$DG_j^y = G_j D_y, \quad D_y = \begin{bmatrix} -1 & 1 \end{bmatrix}^T, \tag{7}$$

$$M_j = \sqrt{\left(DG_j^x\right)^2 + \left(DG_j^y\right)^2}, \tag{8}$$

$$P_j = \tan^{-1}\left[\frac{DG_j^y}{DG_j^x}\right], \tag{9}$$

where $[\,]^T$ indicates vector transposition.

3. The M_j and P_j images are used to obtain the edge using a top-down search algorithm. For a pixel to be classified as an edge, it must satisfy the condition:

$$(M_j(m,n) > \tau_m) \wedge (|P_j(m,n) - \Phi(m,n)| < \tau_p), \; j = 1,\ldots,6, \tag{10}$$

where $\wedge$ is the logical AND operator, τ_m and τ_p are magnitude and phase thresholds, respectively, and $\Phi(m,n) = P_1(m,n)$. τ_m can be either a predefined global value,

τ_{gm}, or a local estimate, τ_{lm} as is defined by Eq. 3. The value of τ_p tends to be small and is used to take into account the discrete nature of the computation of the phase, P_j. When Eq. 10 is satisfied, the pixel at (m,n) is classified as an edge pixel.

4. As pointed out by Clark [67], the edges after Gaussian filtering that correspond with zero-crossings can be divided into two groups: the authentic edge and the phantom edge. To further accurately localize the authentic edge, an edge also needs to satisfy two other conditions:

   (a) The edge is located at the pixel where the sign of second derivative would be different in at least one orientation. We use four predefined orientations to perform this assessment:

$$\begin{aligned} G''_j(m,n)G''_j(m,n+1) &< 0, & 0^\circ, \\ G''_j(m,n)G''_j(m+1,n-1) &< 0, & 45^\circ, \\ G''_j(m,n)G''_j(m+1,n) &< 0, & 90^\circ, \\ G''_j(m,n)G''_j(m+1,n+1) &< 0, & 135^\circ. \end{aligned} \tag{11}$$

   This condition computes the direction of the zero-crossing.

   (b) If the first condition is satisfied for a specific orientation, ϕ, then, for this potential point, the product of first derivative and third derivative should be negative in the same orientation:

$$G'_j(m,n)G'''_j(m,n) < 0. \tag{12}$$

If all the conditions specified by Equations 10–12 are satisfied, then the pixel at (m,n) in the image would be judged to lie on an edge. Once all the edge pixels have been found, an edge map is generated. This edge-map is then used with connectivity analysis to eliminate the edges due to noise. Either the 3-of-3 or the 2-of-3 rule can be employed to generate the edge map. The latter rule leads to more connected edges but also allows more noise pixels to be classified as edges.

II.2.2 Connectivity Analysis

For each non-zero pixel in the edge image obtained from the multi-scale analysis described in Section II.2.1 we perform connectivity analysis which dictates that we search for edges of a prescribed length, *EL*, in the image. The rationale for this is that edges due to noise will usually be isolated and will form either singular points or very short edges in the edge image. This idea expands on the ideas presented by Rahman and Jobson [42] and Jiang and Rahman [66] by looking at larger connectivities to differentiate between pixels due to noise and those due to signal. As an example, consider the case when $EL = 2$. In this case the minimum length of a signal feature would be 2, and all isolated single point edge pixels would be eliminated. Similar procedures apply for $EL > 2$, where longer and longer edge features are classified as signal and shorter ones eliminated as noise. The algorithm can be described by the following steps:

1. Search the 3×3 allowed neighborhood area. If another edge pixel is found, and $EL > 1$, go to Step 2. If an edge pixel is found but $EL = 1$, then go to Step 3. If there is no edge pixel, the pixel would be classified as noise.

2. Move to the new 3×3 area, the center pixel of which is the edge pixel found in Step 1. Not all the pixels in the new 3×3 neighborhood would be examined because

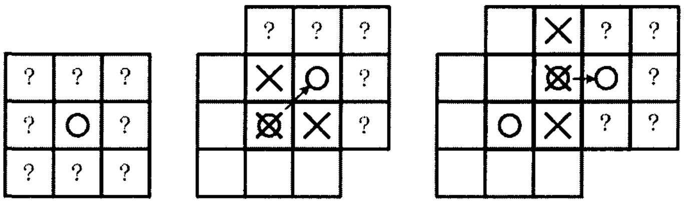

FIG. 2: Edge front: only pixels in front of the current pixel are examined for classification.

some of these pixels have already been tested for connectivity in the previous step, and there is a chance that the edge features would loop back unless such pixels are eliminated from the search space. For this reason only pixels in *front* of the current edge-pixel are tested (Figure 2). This corresponds to examining only those pixels whose distance from the pixel being classified is larger than 1 in the Euclidean sense. The rule specified in Step 1 is used to classify the edge pixel as either noise or signal. In Figure 2 'o' denotes classified edges, '?' denotes potential edge pixels, and '×' denotes pixels that are not examined since they can cause the search to loop. Only the pixels marked as '?' are tested in each analysis step. If a connecting pixel is found, such as that shown in Figure 2 (middle) and $EL > 1$, the area of search is shifted to that pixel and centered on it. Using this new center, the new 3×3 search area is examined, except for the pixels marked with the '×' symbol. This process is repeated until the requisite EL has been achieved.

3. There is a final condition that should also be satisfied

$$|i_0 - i| \geq EL \vee |j_0 - j| \geq EL, \tag{13}$$

where $\vee$ is the logical OR operator, (i_0, j_0) is the location of the pixel under consideration, and (i, j) is the current edge location. This condition is invoked because at times the pixel being classified satisfies the edge length criteria in the Manhattan-distance sense but in the Euclidean sense. This condition places a limit on how short a Manhattan distance is allowed to satisfy the *EL* constraint.

In fact, the determination of how large an *EL* should be used is done by defining what constitutes a feature edge. Because of this, *EL* can be application dependent. In this dissertation, we examine the performance of the noise reduction algorithm as a function of *EL*. We expect that while larger *EL* increases the complexity of the computation, it will reduce the overall impact of noise on the image. This procedure is similar to the analysis used for hysteresis thresholding in the Canny edge operator [5] where pixels in between the two thresholds are considered as edge pixels only if they are already connected to edge pixels.

II.2.3 Noise Reduction

The noise reduction process typically follows the multi-scale edge detection and connectivity analysis processes. To eliminate noise, the final edge maps after connectivity analysis would be utilized to guide the smoothing filter. Edge pixels that are classified as noise are used to mark the pixels at the same location in the original image as noise. These pixels are replaced in the original image by the output of a 3×3 smoothing filter operating on the pixel neighborhood. An important aspect of this algorithm is that while the analysis for determining pixels due to noise is performed on the edge image, the action for eliminating noise is taken on the original image. Hence, this procedure finds the noise pixels in the image and replaces them by a reduced resolution, blurred representation of its neighbors. As shown by Rahman and Jobson [42], this noise reduction mechanism can preserve the edge information while performing noise-reduction.

Thus far, we have introduced the multi-scale edge detection algorithms and the noise reduction mechanism. In Chapter III, we will define the information-theoretic assessment procedure that we use to evaluate the performance of the edge detection algorithms.

CHAPTER III

INFORMATION-THEORETIC ANALYSIS IN VISUAL COMMUNICATION SYSTEM

III.1 INFORMATION-THEORETIC ANALYSIS

This chapter introduces the information-theoretic analysis that we use to assess the performance of edge detection operators in a visual communication system. This analysis is meaningful because there appears to exist a close correlation between the mutual information between the original and the edge image and the quality of the output.

III.1.1 End-to-end Image Processing System

As Shannon [60] stated, the fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

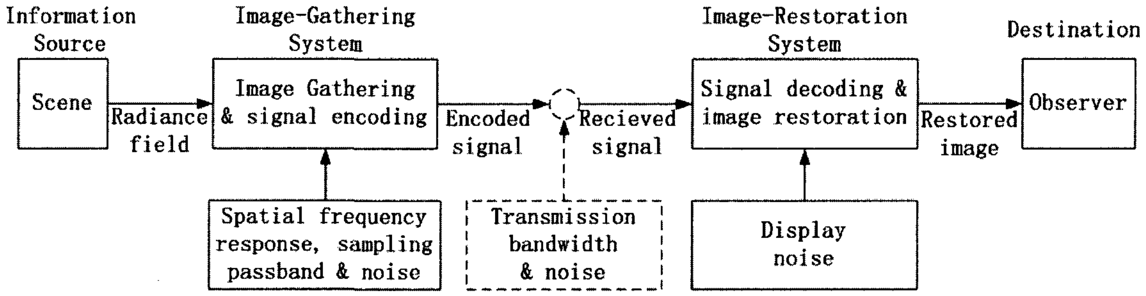

FIG. 3: Model of visual communication channel with the critical limiting factors that constrain its performance.

Figure 3 depicts a typical visual communication channel in which image gathering is combined with encoding to compress the data, and image display is combined with image restoration to enhance the image quality. This channel can be described in terms of three major processes: the image gathering process, the data transmission process and the image restoration process. Huck et al. [57] assumed that transmission errors could be corrected with well-established error-correcting codes, so they did not consider the data transmission process as part of their model. Instead, they developed a framework where the imaging system including digital image processing could be examined as a whole. Since we are not considering image display as part of our analysis, we use the simplified end-to-end imaging model which is shown in Figure 4.

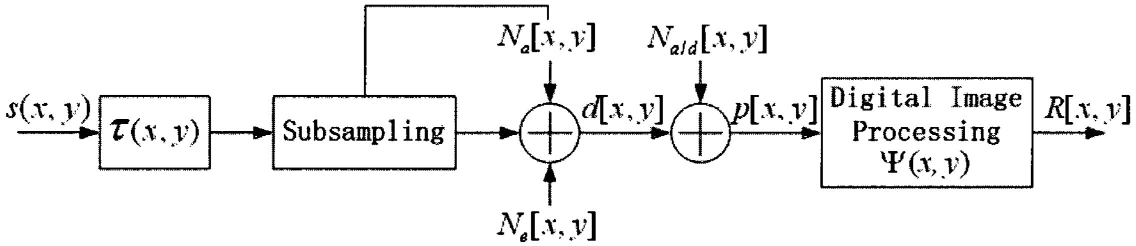

FIG. 4: Simplified end-to-end imaging model.

As can be seen from Figure 4, several sources of noise impact the acquired digital image $p[x,y]$. Three major sources of noise are thermo-electric noise due to the sensor CCD, quantization noise due to the analog-to-digital (A/D) conversion, and aliasing noise due to sampling. The end-to-end imaging process can, thus, be modeled as

$$\begin{aligned} p[x,y] &= (s(x,y)*\tau(x,y)))\underline{|||}(x,y)+N_{a/d}[x,y]+N_e[x,y], \\ R[x,y] &= p[x,y]*\Psi(x,y), \end{aligned} \qquad (14)$$

where $s(x,y)$ is the scene with PSD $\hat{\Phi}_s[\mu,\nu]$; $\tau(x,y)$ is the point spread function (PSF) of the combined sensor and system optics; $\underline{|||}(x,y) = \sum_{m,n\in Z}\delta(x-m,y-n)$ represents the sampling lattice with unit sampling distance; $N_a[x,y]$ is the aliasing noise due to sampling; $N_e[x,y]$ is the discrete system noise due to thermo-electric effects; $N_{a/d}[x,y]$ is the discrete quantization noise. $d[x,y]$ is the discrete signal with continuous amplitude; and $p[x,y]$ is the digital signal. The digital image processing can refer to any processing, including image restoration and edge detection. For example, for image restoration, $\Psi(x,y)$ is the joint restoration and reconstruction filter that corrects for signal attenuation due to noise sources and interpolates the sub-sampled discrete signal $p[x,y]$ onto the continuous display, and produces the restored image $R[x,y]$, which is the same size as the scene, $s(x,y)$.

In the spatial-frequency domain, the end-to-end imaging process can be described as [63]

$$\begin{aligned} \tilde{p}(\mu,\nu) &= (\hat{s}[\mu,\nu]\hat{\tau}[\mu,\nu]) * \hat{\underline{|||}}[\mu,\nu] + \tilde{N}_{a/d}[\mu,\nu] + \tilde{N}_e[\mu,\nu], \\ \hat{R}[\mu,\nu] &= \tilde{p}(\mu,\nu)\hat{\Psi}(\mu,\nu), \end{aligned} \tag{15}$$

where^shows the continuous Fourier transform,~shows the discrete Fourier transform, and $\hat{\underline{|||}}[\mu,\nu] = \sum_{m,n\in Z}\delta(\mu-m,\nu-n)$. The associated sampling passband is given by $\hat{B} = \{\mu,\nu : |\mu|,|\nu| \leq 0.5\}$.

In our research, the optical transfer function (OTF), which is the Fourier transform of the PSF of the image-gathering device can simply, but fairly accurately, be modeled using a Gaussian function [63]:

$$\begin{aligned} \tau(x,y) &= \pi\rho_c^2 \exp\left[-\pi^2\rho_c^2(x^2+y^2)\right], \\ \hat{\tau}(\mu,\nu) &= \exp\left[-(\mu^2+\nu^2)/\rho_c^2\right]. \end{aligned} \tag{16}$$

The value of ρ_c determines the extent of blurring: larger values of ρ_c imply less blurring but more aliasing, and smaller values of ρ_c (< 0.3) imply more blurring but less aliasing.

The probability density function of the quantization noise $N_{a/d}[x,y]$ is assumed to be uniform. It can be written as [63]

$$p(N_{a/d}[x,y]) = \frac{\kappa}{s_{p_{max}}(x,y) - s_{p_{min}}(x,y)} = \frac{\kappa}{2k\sigma_s}, \tag{17}$$

where $s_{p_{max}} = k\sigma_s$ and $s_{p_{min}} = -k\sigma_s$ specify the range of the signal; κ is the number of quantization levels of A/D converter, and $\sigma_p^2 = \int_{\hat{B}} \tilde{\Phi}_p[\mu,\nu]d\mu d\nu$. For $k = \sqrt{3}$, the dynamic range encompasses 92% of the signal. The PSD of the signal $\tilde{\Phi}_p[\mu,\nu]$ prior to quantization is [63]

$$\tilde{\Phi}_p[\mu,\nu] = E\{\tilde{p}(\mu,\nu)\tilde{p}^*(\mu,\nu]\} = \left[K^2\hat{\Phi}_s\left|\hat{\tau}(\mu,\nu)^2\right|\right] * \underline{\widehat{|||}} + \tilde{\Phi}_{N_e}[\mu,\nu], \tag{18}$$

where E is the expectation operator, and * indicates complex conjugation. K is the linear signal-to-radiance conversion constant. Assuming that the error within each quantization interval is uncorrelated with errors within other intervals, the PSD of quantization noise is [63]

$$\tilde{\Phi}_{N_{a/d}}[\mu,\nu] = \sigma^2_{N_{a/d}} = \frac{1}{3}\left(\frac{k\sigma_p}{\kappa}\right)^2 = \left(\frac{\sigma_p}{\kappa}\right)^2. \tag{19}$$

The aliasing "noise" $\tilde{\Phi}_a(\mu,\nu)$ is due to sampling and is given by

$$\tilde{\Phi}_a(\mu,\nu) = K^2\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2 * \sum \delta(\mu - m, \nu - n), \tag{20}$$

where $m,n \in \mathcal{Z}$ and $m,n \neq 0,0$.

III.1.2 Information Rate

The visual communication channel, as modeled above in Section III.1.1, contains three critical transformations: (a) the continuous-to-discrete transformation of the continuous radiance field $s(x,y)$ into the discrete signal $d[x,y]$ with analog magnitudes, (b) the analog-to-digital transformation of the discrete signal $d[x,y]$ into the digital signal $p[x,y]$, and

(c) the digital-to-continuous transformation of the resorted digital image $R[x,y]$ into the continuous observed image $R_o(x,y)$.

According to Shannon [60], the rate of transmission of information $\mathcal{H}$ that the image-gathering process produces or, equivalently, the mutual information between the input radiance field $s(x,y)$ within the sampling passband and the output digital signal $p[x,y]$ is defined by [63]

$$\mathcal{H} = \varepsilon[p[x,y]] - \varepsilon[p[x,y]|s(x,y)] = \varepsilon[\tilde{p}(\mu,\nu)] - \varepsilon[\tilde{p}(\mu,\nu)|\hat{s}(\mu,\nu)], \tag{21}$$

where the first term, $\varepsilon[\cdot]$, in each equation represents the entropy of the received signal in spatial and spatial frequency coordinates, respectively, and the second term, $\varepsilon[\cdot|\cdot]$, represents the conditional entropy of the received signal given the radiance field. The conditional entropy is the uncertainty of the digital signal $\tilde{p}(\mu,\nu)$ when the radiance-field spectrum $\hat{s}(\mu,\nu)$ is known. Thus, the information rate $\mathcal{H}$ given by this defining equation measures the amount of information received less the part of this which is due to noise.

The assumptions that we have made above in Section III.1.1, show that the aliased signal components $N_a(x,y)$ caused by insufficient sampling and the quantization noise $N_{a/d}(x,y)$ caused by the analog-to-digital transformation can both be treated as independent, additive noises. This allows us, according to Shannon's Theorem 16 [60], to simplify $\mathcal{H}$ given by Eq. 21 to [63]

$$\mathcal{H} = \varepsilon[p[x,y]] - \varepsilon[N[x,y]] = \varepsilon[\tilde{p}(\mu,\nu)] - \varepsilon[\tilde{N}[\mu,\nu]], \tag{22}$$

where

$$\tilde{p}(\mu,\nu) = K\hat{s}(\mu,\nu)\hat{\tau}(\mu,\nu) + \tilde{N}(\mu,\nu) \tag{23}$$

and

$$\tilde{N}(\mu,\nu) = \hat{N}_a(\mu,\nu) + \tilde{N}_e(\mu,\nu) + \tilde{N}_{a/d}(\mu,\nu). \tag{24}$$

A full interpretation of Eq. 22 in terms of the probability densities $p_s[\tilde{p}(\mu,\nu)]$ and $p_n[\tilde{N}(\mu,\nu)]$ of the signal and the total noise, respectively, is given by the expression [63]

$$\begin{aligned}\mathcal{H} = \ & \frac{1}{2}\iint\Big\{\iint p_s[\tilde{p}(\mu,\nu)]\log_2 p_s[\tilde{p}(\mu,\nu)]\,d\tilde{p}(\mu,\nu)\\ & \qquad - \iint p_N[\hat{N}(\mu,\nu)]\log_2 p_N[\hat{N}(\mu,\nu)]d\hat{N}(\mu,\nu)\Big\}\,d\mu d\nu,\end{aligned} \tag{25}$$

where $d\tilde{p}(\mu,\nu)$ and $d\tilde{N}(\mu,\nu)$ denote the real differential area elements comprising the real and imaginary parts of $\tilde{p}(\mu,\nu)$ and $\tilde{N}(\mu,\nu)$, respectively. The Gaussian probability density $p_s[\tilde{p}(\mu,\nu)]$ of the signal $\tilde{p}(\mu,\nu)$ is

$$p_s[\tilde{p}(\mu,\nu)] = \frac{1}{\pi\tilde{\Phi}_p(\mu,\nu)}\exp\left[-|\tilde{p}(\mu,\nu)|^2/\tilde{\Phi}_p(\mu,\nu)\right]. \tag{26}$$

Similarly, the Gaussian probability density $p_N[\tilde{N}(\mu,\nu)]$ of the noise $p_N[\tilde{n}(\mu,\nu)]$ with the PSD $\tilde{\Phi}_N(\mu,\nu)$ is [63]

$$p_n[\tilde{p}(\mu,\nu)] = \frac{1}{\pi\tilde{\Phi}_n(\mu,\nu)}\exp\left[-|\tilde{n}(\mu,\nu)|^2/\tilde{\Phi}_n(\mu,\nu)\right]. \tag{27}$$

The PSD $\tilde{\Phi}_p(\mu,\nu)$ of the signal is

$$\tilde{\Phi}_p = [\hat{\Phi}_p|\hat{\tau}(\mu,\nu)|^2] * \underline{\hat{|||}} + \tilde{\Phi}_e + \tilde{\Phi}_{a/d}, \tag{28}$$

and the PSD $\tilde{\Phi}_N(\mu,\nu)$ of the signal is

$$\tilde{\Phi}_N = [\hat{\Phi}_p|\hat{\tau}(\mu,\nu)|^2] * \underline{\hat{|||}}_s + \tilde{\Phi}_e + \tilde{\Phi}_{a/d}, \tag{29}$$

where $\underline{\hat{|||}} \equiv \underline{\hat{|||}}(\mu,\nu) = \delta(\mu,\nu) + \underline{\hat{|||}}_s(\mu,\nu)$, i.e., $\underline{\hat{|||}}_s$ represents the aliasing sidebands. Substituting the above into Eq. 22 yields the following results:

$$\begin{aligned}\mathcal{H} &= \frac{1}{2}\iint \log_2 \tilde{\Phi}_p(\mu,\nu)d\mu d\nu - \frac{1}{2}\iint \log_2 \tilde{\Phi}_n(\mu,\nu)d\mu d\nu \\ &= \frac{1}{2}\iint \log_2 \frac{\tilde{\Phi}_s(\mu,\nu)}{\tilde{\Phi}_n(\mu,\nu)} d\mu d\nu = \frac{1}{2}\iint \log_2 [1 + \frac{\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2}{\tilde{\Phi}_n(\mu,\nu)}] d\mu d\nu \\ &= \frac{1}{2}\iint \log_2 \left[1 + \frac{\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2}{\tilde{\Phi}_a(\mu,\nu) + \tilde{\Phi}_{N_e}(\mu,\nu) + \tilde{\Phi}_{N_{a/d}}(\mu,\nu)}\right] d\mu d\nu.\end{aligned} \tag{30}$$

Actually, the above formulations of $\mathcal{H}$ can be tied to the Wiener restoration filter $\hat{\Psi}$, which is given by [63]

$$\hat{\Psi}(\mu,\nu) = \frac{\hat{\Phi}_s(\mu,\nu)\hat{\tau}^*(\mu,\nu)}{K^2\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2 + \tilde{\Phi}_a(\mu,\nu) + \tilde{\Phi}_{N_e}(\mu,\nu) + \tilde{\Phi}_{N_{a/d}}(\mu,\nu)}, \tag{31}$$

where

$$\tilde{\Phi}_a(\mu,\nu) = K^2\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2 * \sum \delta(\mu - m, \nu - n) \tag{32}$$

is the aliasing noise, where $m,n \in Z$ and $m,n \neq 0,0$. Therefore, the information capacity $\mathcal{H}$ of a system is defined as the mutual information between the radiance field and the sampled image [63]:

$$\mathcal{H} = \frac{1}{2}\iint \log_2 \left[1 + \frac{\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2}{\tilde{\Phi}_a(\mu,\nu) + \tilde{\Phi}_{N_e}(\mu,\nu) + \tilde{\Phi}_{N_{a/d}}(\mu,\nu)}\right] d\mu d\nu. \tag{33}$$

For the simplifying assumptions that (a) the photodetector noise is white so that the PSD $\tilde{\Phi}_{N_e}(\mu,\nu)$ is equal to its variance $\sigma^2_{N_e}$ and (b) the PSD $\tilde{\Phi}_{N_{a/d}}(\mu,\nu)$ of the quantization noise is equal to its variance $\sigma^2_{N_{a/d}}$, Eq. 33 can be expressed as a function of the SNRs $K\sigma_s/\sigma_{N_e}$ and $(K\sigma_s/\sigma_p)\kappa$ as [63]

$$\mathcal{H} = \frac{1}{2}\iint \log_2 \left[1 + \frac{\hat{\Phi}'_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2}{\tilde{\Phi}'_a(\mu,\nu) + (K\sigma_s/\sigma_{N_e})^{-2} + (K\sigma_s/\sigma_p)^{-2}\kappa^{-2}}\right] d\mu d\nu, \tag{34}$$

where $\hat{\Phi}'_s(\mu,\nu) = \sigma_s^{-2}\hat{\Phi}_s(\mu,\nu)$ and $\hat{\Phi}'_a(\mu,\nu) = \sigma_s^{-2}\hat{\Phi}_a(\mu,\nu)$.

Huck et al. [63] proposed that $\sigma_{\tilde{p}}^2$ can be assumed the same as σ_s^2, for $\sigma_{\tilde{s}}^2 = K^2\sigma_{\tilde{p}}^2$. The information capacity equation can be further simplified as [63]

$$\mathcal{H} = \frac{1}{2}\iint \log_2 \left[1 + \frac{\hat{\Phi}_s'(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2}{\tilde{\hat{\Phi}}_a'(\mu,\nu) + (1/SNR)^2 + (1/\kappa)^2}\right] d\mu d\nu. \tag{35}$$

The one "unknown" in Equations 33–35 is the power spectral density of the scene, $\hat{\Phi}_s$. We assume that

$$\hat{\Phi}_s(\mu,\nu) = \frac{2\pi{\sigma_s}^2\zeta^2}{[1+4\pi^2\zeta^2(\mu^2+\nu^2)]^{3/2}}, \tag{36}$$

where ζ is the mean spatial detail in the scene with respect to the inter-sample distance, and σ_s is the standard deviation of the radiance field. This description represents the ensemble average of scenes acquired from remote sensing platforms.

In the end-to-end system model described in Figure 3, image processing algorithms like edge-detection are applied prior to image restoration. Hence, we will further restrict our evaluation to just before the formation of the image restoration filter which, consequently, plays no part in this analysis.

Edge detection is a typical digital image processing algorithm that can be evaluated within the context of the end-to-end imaging model. Edges are usually modeled by the high frequency information in an image. For this reason, edge detectors are often regarded as high-pass or band-pass filters, but the properties and parameters of the edge-detection filters vary for different edge operators. Here, we incorporate the edge-detection process into the model shown in Figure 4 by letting the image processing module be an edge-detection module. To evaluate the edge detection, the model would have to be adjusted as shown in Figure 5.

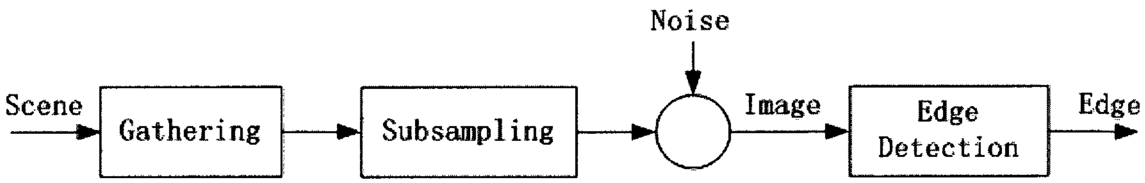

FIG. 5: End-to-end edge detection model.

We again neglect anything beyond the edge-detection process because we are only interested in the information throughput between input scene at image acquisition and the output of the edge-detection algorithm. Consequently, the associated information capacity $\mathcal{H}_e$ is

$$\mathcal{H}_e = \frac{1}{2}\iint \log_2\left[1+\frac{\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2|\hat{\tau}_e(\mu,\nu)|^2}{\tilde{\Phi}_A(\mu,\nu)+\tilde{\Phi}_{N_e}(\mu,\nu)+\tilde{\Phi}_{N_{a/d}}(\mu,\nu)}\right] d\mu d\nu, \tag{37}$$

where

$$\tilde{\Phi}_A(\mu,\nu) = K^2\hat{\Phi}_s(\mu,\nu)|\hat{\tau}(\mu,\nu)|^2|\hat{\tau}_e(\mu,\nu)|^2 * \sum\delta(\mu-m,\nu-n) \tag{38}$$

is the aliasing noise and where $m,n \in \mathcal{Z}$ and $m,n \neq 0,0$, and $\hat{\tau}_e(\mu,\nu)$ is the SFR of edge detector.

III.2 EDGE DETECTORS

To analyze traditional edge detection performance in the context of visual communication, we use the following edge detection operators in this dissertation:

1. Sobel [2],

2. Prewitt [3],

3. Roberts [1],

4. Laplacian-of-Gaussian (LoG) [8],

5. Canny [5].

III.2.1 Sobel Operator

The Sobel operator is one of the most popular edge detection operators. In its simplest form, it consists of two convolution kernels: one, S_H, is designed to detect primarily horizontal edges and the other, S_V, primarily vertical edges.

$$S_H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad S_V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{39}$$

The edge pixel, $e_s(m,n)$, is then given by

$$e_s(m,n) = |G(m,n) * S_H(m,n)| + |G(m,n) * S_V(m,n)|, \tag{40}$$

where $G(m,n)$ are the pixels from the input image, G.

III.2.2 Prewitt Operator

The Prewitt horizontal and vertical are very similar to the Sobel kernels. The main difference is that whereas the Sobel operator weights the central column and row values twice as much as the border values, the Prewitt operator uses a weight of 1. The two convolution kernels P_H and P_V are given as

$$P_H = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad P_V = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}. \tag{41}$$

The edge pixel, $e_s(m,n)$, is then given by

$$e_s(m,n) = |G(m,n) * P_H(m,n)| + |G(m,n) * P_V(m,n)| . \tag{42}$$

III.2.3 Roberts Operator

Except Sobel and Prewitt are first-order partial derivative kernel operators, the Robert operator is another first-order partial derivative kernel, but its mask size is 2 × 2. It has two convolution kernels: R_x and R_y.

$$R_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad R_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{43}$$

The edge pixel, $e_s(m,n)$, is then given by

$$e_s(m,n) = |G(m,n) * R_x(m,n)| + |G(m,n) * R_y(m,n)| . \tag{44}$$

III.2.4 LoG Operator

The LoG is a second-order derivative operator. The spatial response and SFR of this operator, respectively, are

$$\tau_e(x,y) = \frac{1}{\pi\sigma_e^4}\left(1 - \frac{r^2}{2\sigma_e^2}\right)\exp\left[-\frac{r^2}{2\sigma_e^2}\right], \tag{45}$$

and

$$\hat{\tau}_e(\mu,\nu) = (2\pi\rho)^2 \exp\left[-2(\pi\sigma_e\rho)^2\right], \tag{46}$$

where $r^2 = x^2 + y^2$, $\rho^2 = \mu^2 + \nu^2$, σ_e is the standard deviation of the Gaussian function, and $\rho = 1/\sqrt{2}\pi\sigma_e$ is the spatial frequency at which $\hat{\tau}_e$ is maximum.

A 5×5 approximation spatial kernel of the LoG is given by

$$L = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}. \tag{47}$$

The edge pixel, $e_s(m,n)$, is then given by

$$e_s(m,n) = G(m,n) * L(m,n). \tag{48}$$

III.2.5 Canny Operator

The Canny operator [5] is based upon three basic performance criteria: good SNR, good detection and localization, and only one response to a single edge. It is implemented as the following four steps:

1. Smoothing the image using Gaussian filtering: this step is used to low-pass filter the image to reduce some of the high-frequency noise.

2. Calculating the gradient magnitude and direction by using first order finite differences: this step provides the candidate map for edge-pixels. Typically an operation like Sobel edge detection is used.

3. Imposing non-maximum suppression on the gradient value: this step ensures that there is only one response to an edge. In other words, this is an edge-thinning process.

4. Detecting and connecting the edges by using hysteresis thresholding. The idea of hysteresis thresholding is simple yet elegant. Two thresholds τ_u and τ_l, which are often linearly related, are used. A candidate edge pixel $c(m,n)$ is classified as an edge pixel if $|c(m,n)| > \tau_u$ and as a non-edge if $c(m,n) < \tau_l$. If neither condition is satisfied, then $c(m,n)$ is classified as an edge if it is 4-adjacent to a previously classified edge-pixel. The idea of 4-adjacency simply means that the search space for adjacency is restricted to the East, West, North, and South neighbors. This reduces the number of spurious pixels due to noise.

So far we have described the complete system model and introduced the edge detectors that we compare using the information-theoretic analysis. As stated earlier, the information-theoretic analysis requires that we know the spatial-frequency response and the PSD of the edge detection operators. We will investigate edge detection and noise-reduction using edge and connectivity analysis in Chapter IV and derive the PSDs and evaluate the performance of the operators in Chapter V.

CHAPTER IV

RESULTS OF EDGE DETECTION AND NOISE REDUCTION

We compare the performance of the multi-scale edge detection algorithms with several edge detection algorithms. We also use edge-detection in conjunction with connectivity analysis to perform noise reduction. For noise reduction, we use both edge detection algorithms, including the proposed algorithms, and some traditional edge preserving noise reduction filters. For the edge detection operators, we perform connectivity analysis after edge detection to denoise the original image. For the linear or non-linear filters, we apply them directly to the image. Each of these algorithms is discussed in some detail in Section IV.1, except those that were described in Section III.2. The results of edge detection are shown in Section IV.2, and the noise reduction performance of different methods is shown in Section IV.3.

IV.1 ALGORITHMS USED FOR EDGE DETECTION OR NOISE REDUCTION

In addition to the previously described Sobel, Prewitt, Roberts, LoG, and Canny operators (Chapter III), we use the following algorithms for denoising:

1. 3×3 smallest difference-of-Gaussian, or, lateral-inhibition (LIH) [63, 59],

2. Median filter [7, pp. 165–167],

3. Adaptive median filter (AMF) [7, pp. 332–335],

4. Adaptive threshold median filter (ATMF) [25].

Of these, the LIH is an edge-detection algorithm, and the other three are traditionally used noise-reduction algorithms.

IV.1.1 Lateral Inhibition

The smallest difference-of-Gaussian (DOG), or the lateral inhibition (LIH) [63, 59], operator derives its origin from the natural vision literature. The lateral inhibition terminology comes from its physical renditions, where a positive center pixel in a 3×3 neighborhood is inhibited by all of its neighbors to produce a high pass signal. Mathematically, the output of the LIH is given by

$$e_l(m,n) = G(m,n) * L(m,n), \tag{49}$$

where $e_l(m,n)$ is the edge output, and

$$L = \begin{bmatrix} -0.0675 & -0.1820 & -0.0675 \\ -0.1820 & 1.0000 & -0.1820 \\ -0.0675 & -0.1820 & -0.0675 \end{bmatrix}. \tag{50}$$

IV.1.2 Median Filter

The best-known median filter [7, pp. 165–167] replaces the pixel value $f(m,n)$ with the median of the gray levels in the 3×3 neighborhood $\mathscr{A}$ of that pixel:

$$f(m,n) = \text{median}\left(g(m,n)\right), \quad (m,n) \in \mathscr{A}. \tag{51}$$

The original value of the pixel is included in the computation of the median. The median filter is very popular because, for certain types of random noise, it provides excellent noise-reduction capabilities, with considerably less blurring than the linear mean filter of the same size.

IV.1.3 Adaptive Median Filter

Despite its effectiveness in eliminating noise, the median filter tends to remove fine details when applied uniformly to an image. To eliminate this drawback, the adaptive median filter (AMF) [7, pp. 332–335] has been proposed. The AMF uses variable window sizes for removing impulse noise while preserving sharpness. In this way, the integrity of edge and detail information becomes better. The basic window size of AMF is 3, which is the same as the median filter. The following algorithm is used to implement the AMF:

1. In the $P \times P$ neighborhood around the pixel $p(m,n)$, compute the median z_{med}, the maximum value z_{max}, and the minimum value z_{min}. If $z_{min} < z_{med} < z_{max}$ go to step 3; otherwise go to step 2.

2. Increase the size of the neighborhood to the next higher odd number, i.e., $P \leftarrow P+2$. If the new P is larger than the allowed maximum size, go to step 4. Otherwise, go back to step 1.

3. Compare $p(m,n)$ with z_{max} and z_{min}. If $z_{min} < p(m,n) < z_{max}$, output $p(m,n)$.

4. Output $p(m,n) \leftarrow z_{med}$.

IV.1.4 Adaptive Threshold Median Filter

Images are often corrupted by multiple-impulse noise, which means that the magnitude of impulse noise is not constant as is generally assumed. Its magnitude might be slightly varying but still close to extreme (pure black or while) values in an image. The performance of the AMF in the presence of such noise is not very good, so a modified AMF called the adaptive threshold median filter (ATMF) [25] is used. The basic steps of the ATMF are the same as those of the AMF as described in Section IV.1.3. The major difference is that the ATMF integrates the AMF with two dynamic thresholds. An in-depth analysis of the AMF shows that the values of z_{max} and z_{min} are critical in judging the noise. Since the ATMF attempts to handle multiple impulse noise, the z_{max} and z_{min} of AMF are replaced with the least maximum and the largest minimum with the help of two dynamic thresholds. The dynamic thresholds enhance the ability of the filter to detect the multiple noises and balance the noise removal and image quality.

IV.2 EDGE DETECTION RESULTS

IV.2.1 Experimental Evaluation

We have tested our methods on color and grayscale images with root-mean-square (RMS) SNRs varying from 1 to 10. For color images, the algorithms are applied to each channel individually. The baseline images used in the experiments were computer generated so they are noise free, allowing us to control the SNR of the images precisely by adding white Gaussian noise of requisite amplitude to the data. This provides a controlled environment in which the performance of the algorithms can be evaluated as a function of the different system parameters which can be modified. Also, this allows us to compare the effectiveness of the noise-reduction algorithms since we can access the original noise-free image and compare the denoised directly with the original image. To test the performance of the algorithms on real images, “Barbara” and “Lena” were used. The original test images are shown in Figure 6 for the noise-free and the extremely noisy—SNR = 1—case.

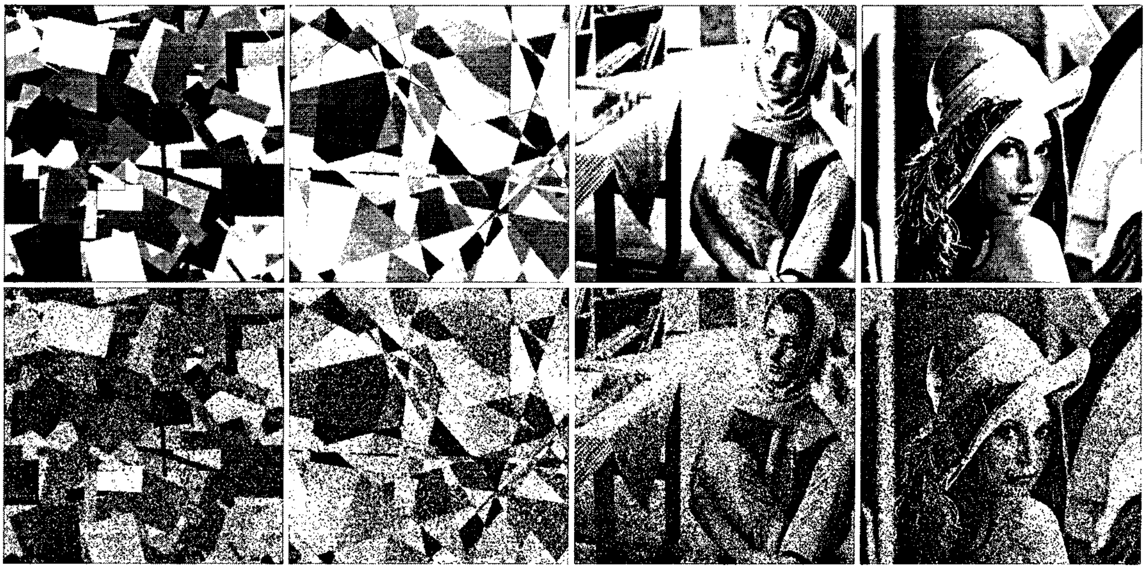

FIG. 6: Top-row: noise-free, and bottom-row SNR = 1: (left-column) Rect; (second-column) Randpoly; (third-column) Barbara; (right-column) Lena.

Figure 7 shows an original, noise-free, image G and three noisy images, $G_{SNR=10}$, $G_{SNR=5}$, and $G_{SNR=1}$. In general, except when the SNR $\approx$ 1, traditional edge detection methods can be used to find the edges that have been impacted by noise. Figure 8 shows G_j for the original noise free image, G, and Figure 9 shows the multi-resolution images for $G_{SNR=1}$. The F_j (Eq. 5) used for these images were generated using $\sigma_1 = 2$. This value of σ_1 was deemed to be optimal after conducting a series of experiments based on different

FIG. 7: (left) Original image, G; (second) noisy image $G_{SNR=10}$; (third) noisy image, $G_{SNR=5}$; and (right) noisy image, $G_{SNR=1}$.

images, which are noise-free or corrupted by white Gaussian noise with various SNR from 1 to 10. This scale makes the resolution differences between each neighbor scales large enough to differentiate details and noise but not so overwhelming so as to lead to too much blurring.

As can be seen from Figures 8 and 9, while noise suppression due to the Gaussian filters is small for $j = 1,2,3$, i.e., $\Delta(G) = |G_j - G_{SNR=1j}|$ is large, $\Delta(G)$ is relatively small for $j = 4,5,6$, i.e., noise suppression is large.

From our experiments we determined that it is not necessary to use all six images with different resolutions to obtain an edge image: in general, three resolutions are enough. The question, then, is: which three resolutions should be used? For the noise-free image or good SNR, i.e., SNR$\geq$ 10, this is not hard to choose. Since we do not want to attenuate too much original information, smaller scales of the Gaussian filters, i.e., $j = 1,2,3$, would be used to create the three resolution images. The selection is not so obvious for the extremely noisy condition, such as SNR$=$ 1. We will discuss this low SNR condition in detail below.

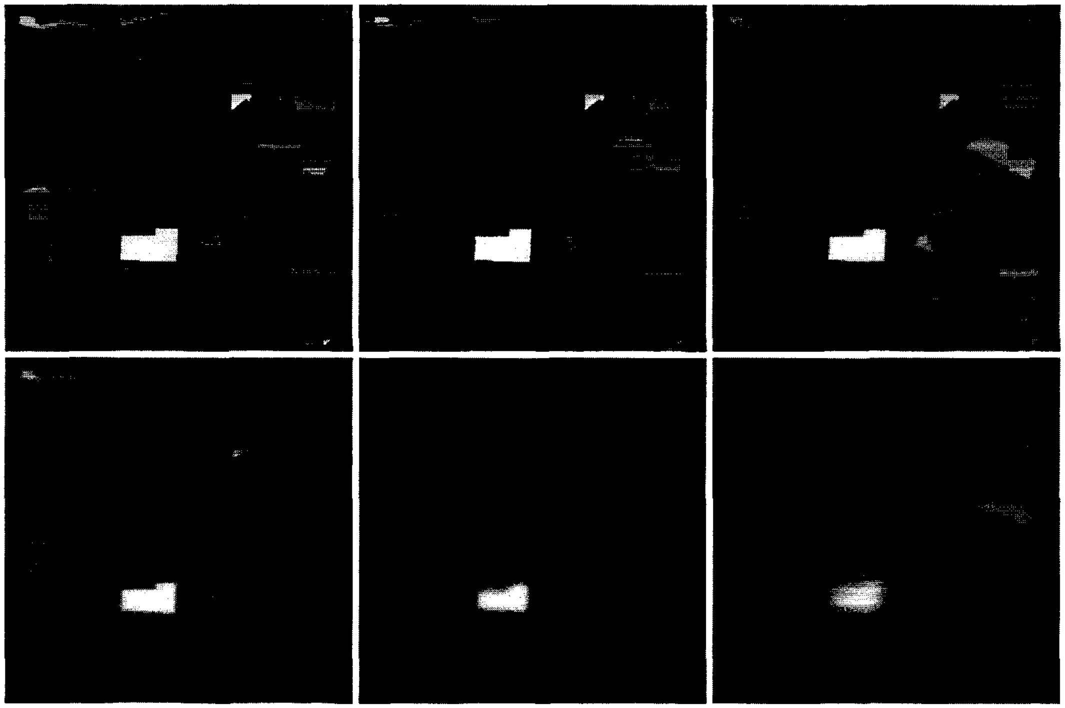

FIG. 8: Multi-resolution images: $G_j = G * F_j,\ \ j = 1,\ldots,6,\ \sigma_1 = 2$ (Equation 5).

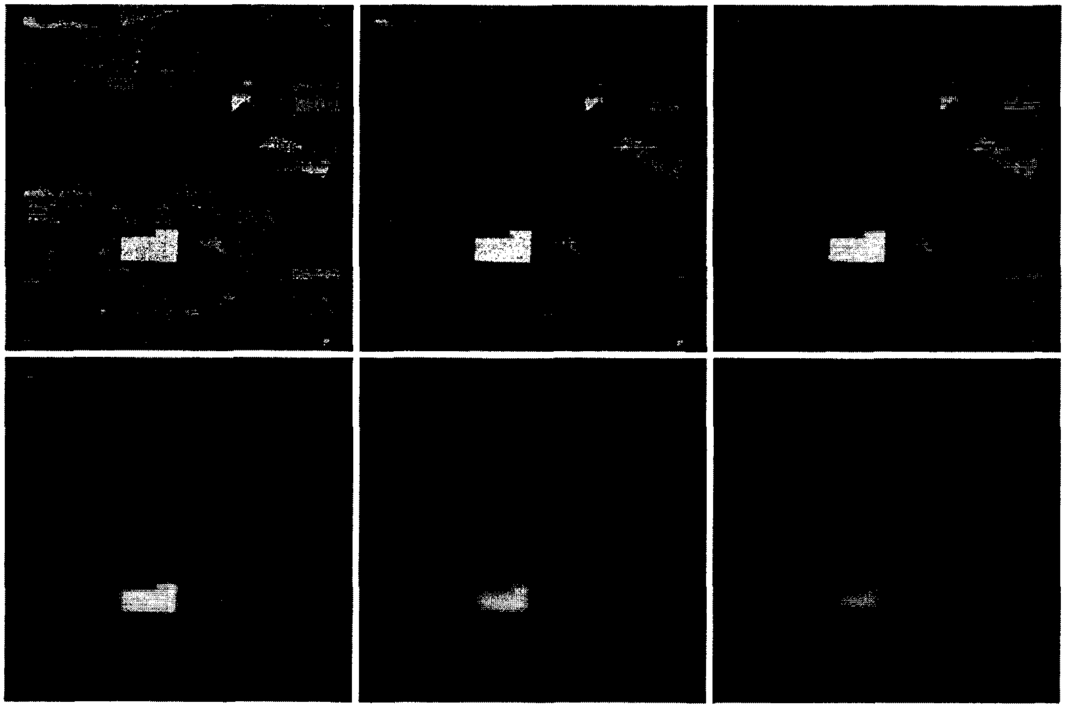

FIG. 9: Multi-resolution images: $G_j = G_{SNR=1} * F_j,\ \ j = 1,\ldots,6,\ \sigma_1 = 2$ (Equation 5).

From Figure 9, we see that smaller σ_j keep the image sharp and noisy and vice versa. Using just the smallest or largest σ_j does not provide good performance. Experiments show that the larger σ_j, such as the one shown in Figure 9 (bottom-right) lose almost all the high-frequency information and hardly give any help for edge detection. In Figure 10, different combinations of resolutions are shown. In each case, three neighboring scaled images were selected. We found that different combinations of resolutions required different values for τ_m and τ_p. While we had to relax the requirement on τ_p, its impact on edge detection was not critical. However, varying τ_m has a significant impact on the performance of the algorithm. Increasing τ_m removes more noise but also loses more edges.

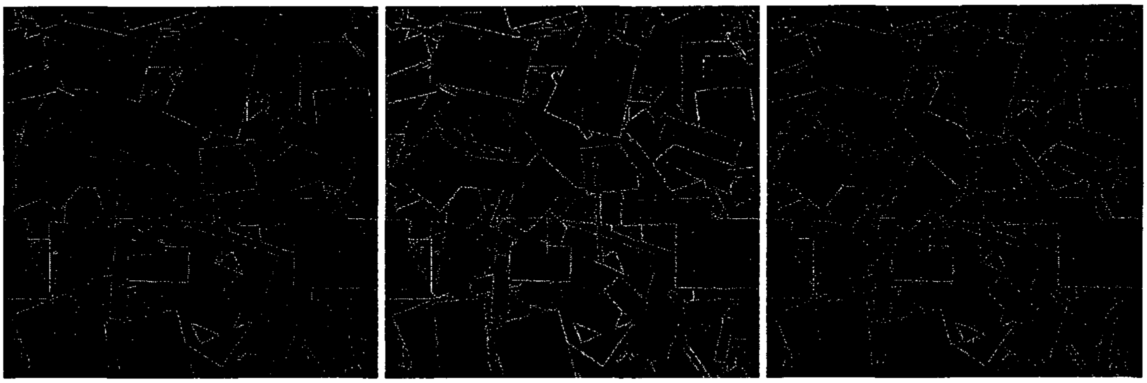

FIG. 10: The results of combining 3 different layers on $G_{SNR=1}$: (left) layers 1, 2, and 3; (center) layers 2, 3, and 4; and (right) layers 3, 4, and 5.

Using the same σ_1, another new scheme was introduced. A pixel at (m,n) was considered to be an edge pixel if $M_j(m,n) > \tau_m$ for two out of the three (2-of-3) resolutions, rather than for all three (3-of-3) resolutions as shown in Figure 10. Results for this scheme are shown in Figure 11. While this new scheme leads to more connected edges, it also allows more noise pixels to be classified as edges. However, the visual impact is better than the 3-of-3 scheme because the edges are finer. The value for τ_m changes slightly for optimal results, but τ_p is the same as that used in Figure 10.

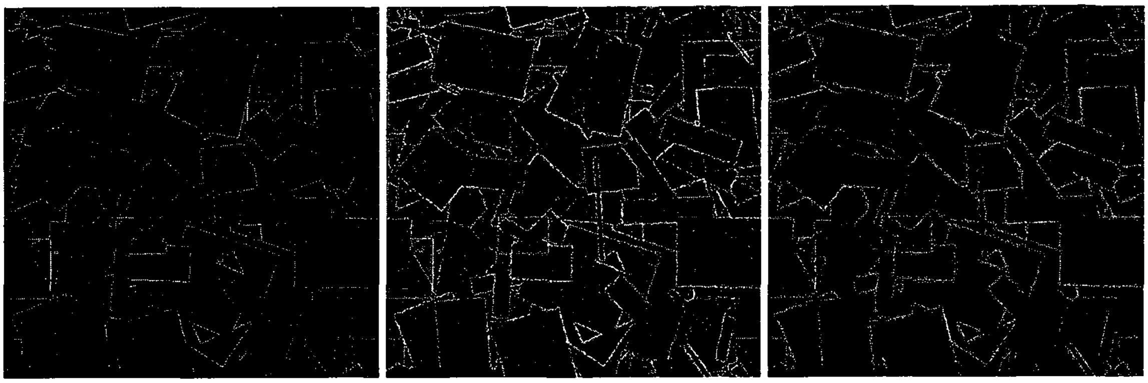

FIG. 11: The results of using the 2-of-3 rule on $G_{SNR=1}$: (left) layers 2, 3, and 4; (center) layers 3, 4, and 5; and (right) layers 4, 5, and 6.

For the 3-of-3 and 2-of-3 schemes, using different Gaussian filters to suppress the noise would make the edges thicker as σ_1 increases. Thinner edges can be produced only at the cost of classifying more noise pixels as edge pixels, especially for very low SNRs. We have determined experimentally that if SNR ≥ 10, then the proposed algorithms can produce edges that are as thin as those produced by traditional algorithms for the noise-free case.

IV.2.2 Results of Comparisons

In order to evaluate the performance of the proposed algorithms, we compare its performance with the traditional edge detection methods described in Sections III.2 and IV.1.

Noise-free case

In the noise-free case, all the methods give excellent results, with relatively minor differences in performance, for all the test cases. In Figures 12, 13, 14, and 15, we show the results for two computer-generated images, "Rect" and "Randpoly," and two real images, "Barbara" and "Lena." For the computer-generated cases, both 3-of-3 and 2-of-3 methods give good results, but the edges for the latter are finer than those produced by the former. This is because fewer pixels are eliminated for the 2-of-3 methods since the comparison takes place over fewer scales. The LNE algorithm extracts additional finer details and produces cleaner, straighter, and thinner edges. While the Sobel operator correctly finds the edges in the image, the edges it produces are thicker than those produced by other operators. Prewitt and Roberts also have almost the same results—since the image is noise-free, all results of first derivative based operators are very good. LIH marks the location of the edges precisely and the edges are thin. The result of LoG is not as good as the others because of the single kernel approximation which balances the trade-off between the contour and details. Thus, while the small details are not picked up, the main structures are. The Canny operator also performs well and produces thin edges.

Since the "Barbara" image contains fine detail, it is a good exemplar for testing the performance of edge detectors. Our proposed algorithms and the first derivative methods—Sobel, Prewitt, Roberts, and LIH—both produce good results. All of them not only effectively extract edge from main structures, such as the frame, the books, and the "Barbara," but also extract the edge exists in high edge density areas like the scarf, the trousers and the table cover. The LoG and the Canny operators are good at extracting main structures

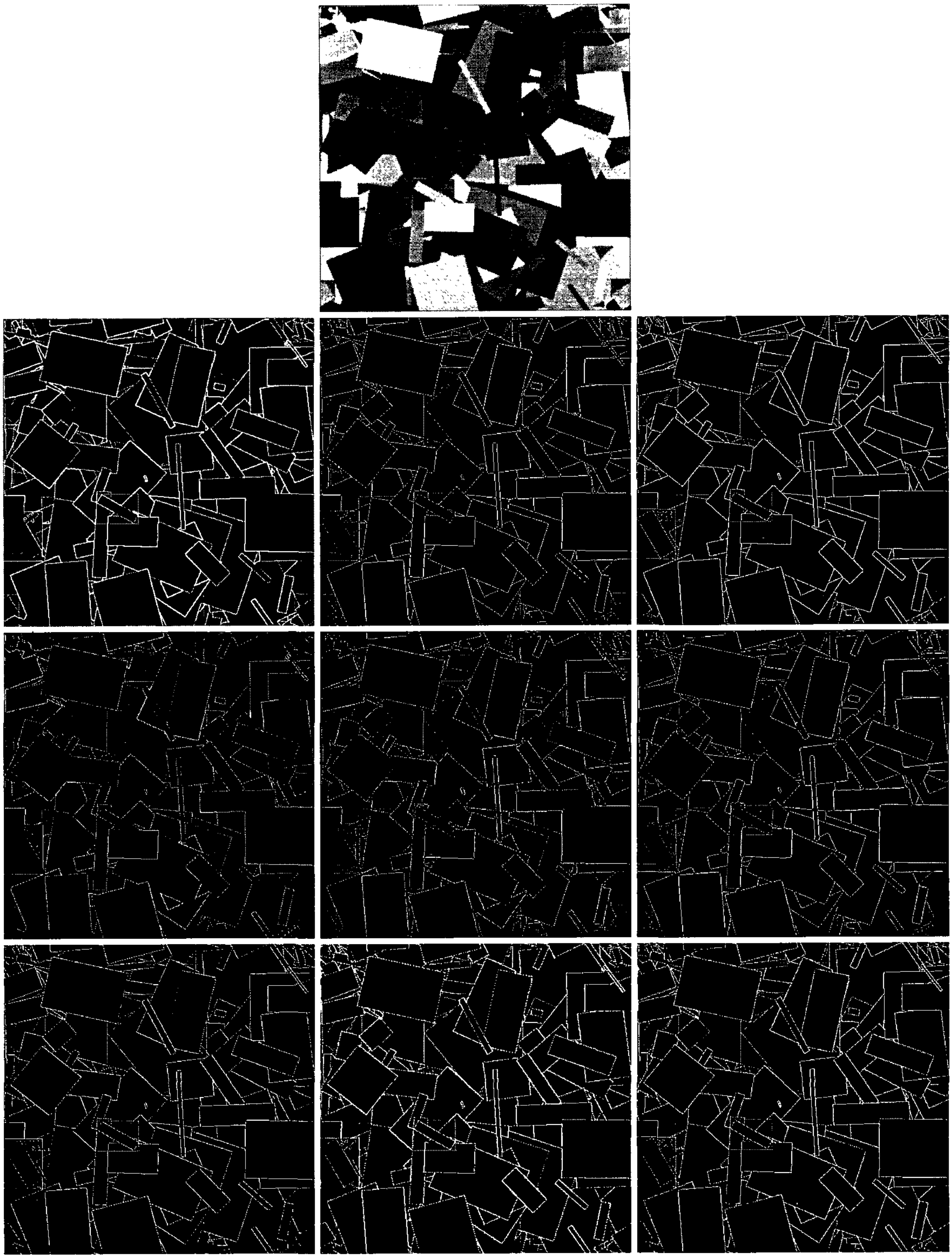

FIG. 12: “Rect”: (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=1); (third-row-center) 2-of-3 (EL=1); and (third-row-right) LNE (EL=1).

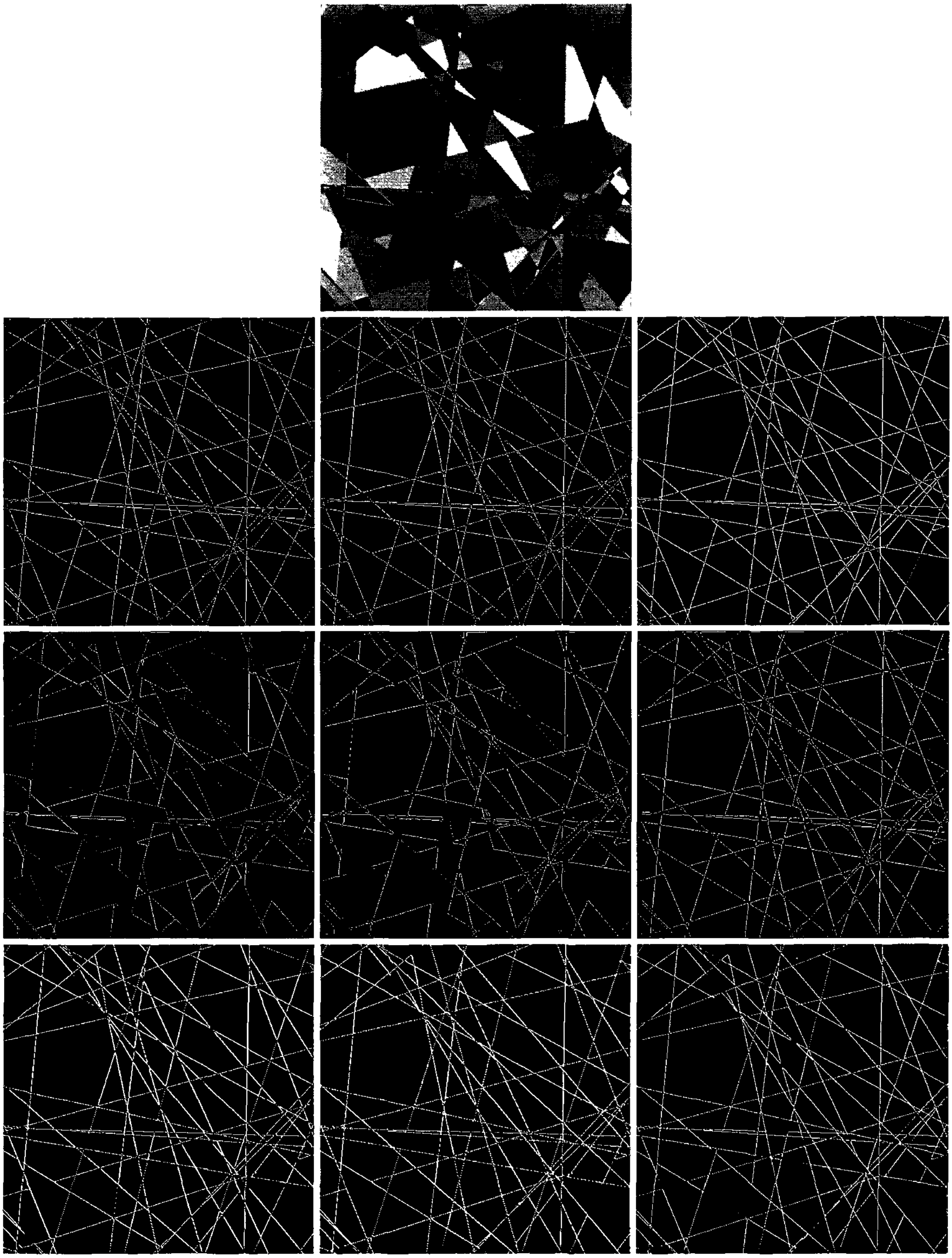

FIG. 13: "Randpoly": (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=1); (third-row-center) 2-of-3 (EL=1); and (third-row-right) LNE (EL=1).

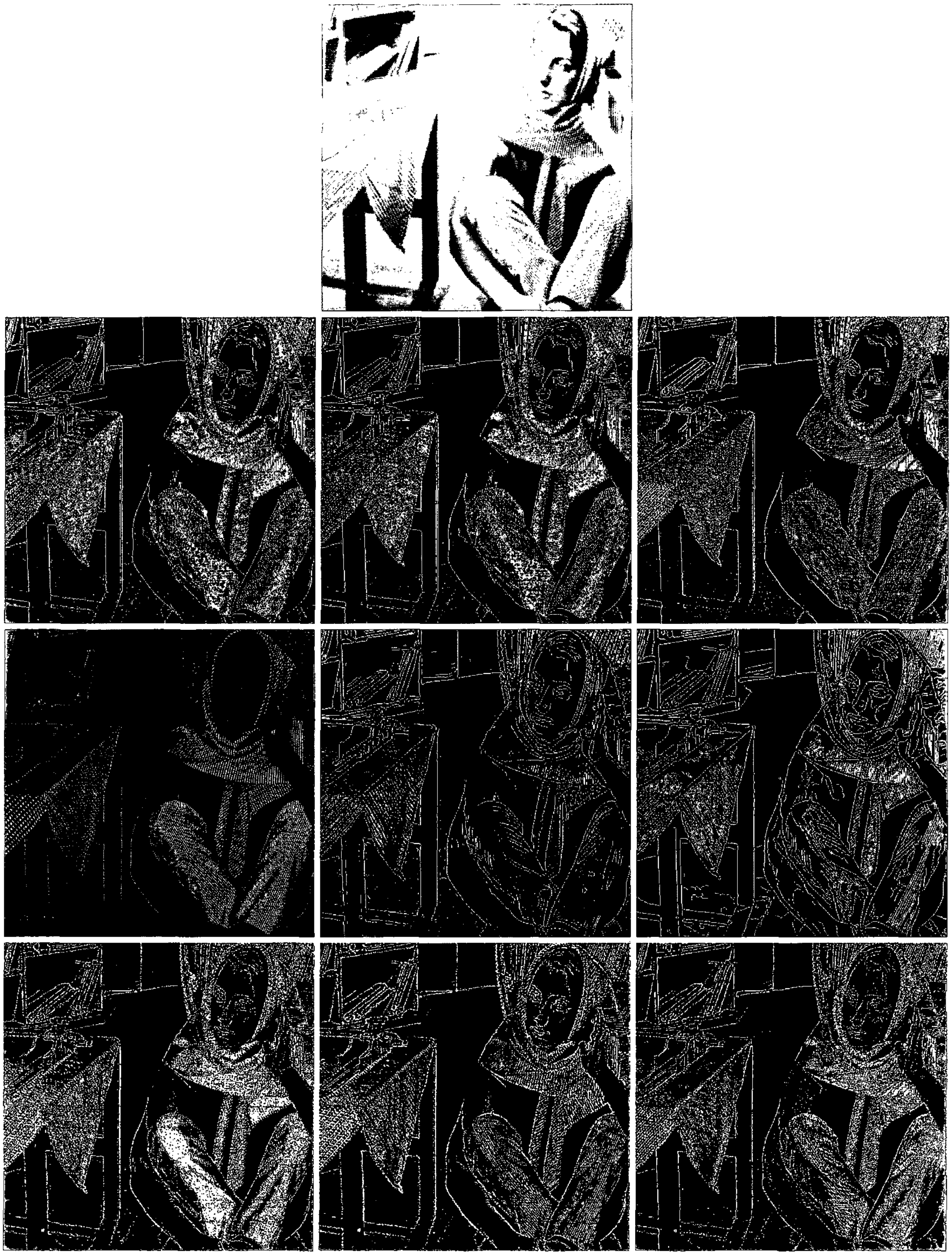

FIG. 14: "Barbara": (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=2); (third-row-center) 2-of-3 (EL=0); and (third-row-right) LNE (EL=2).

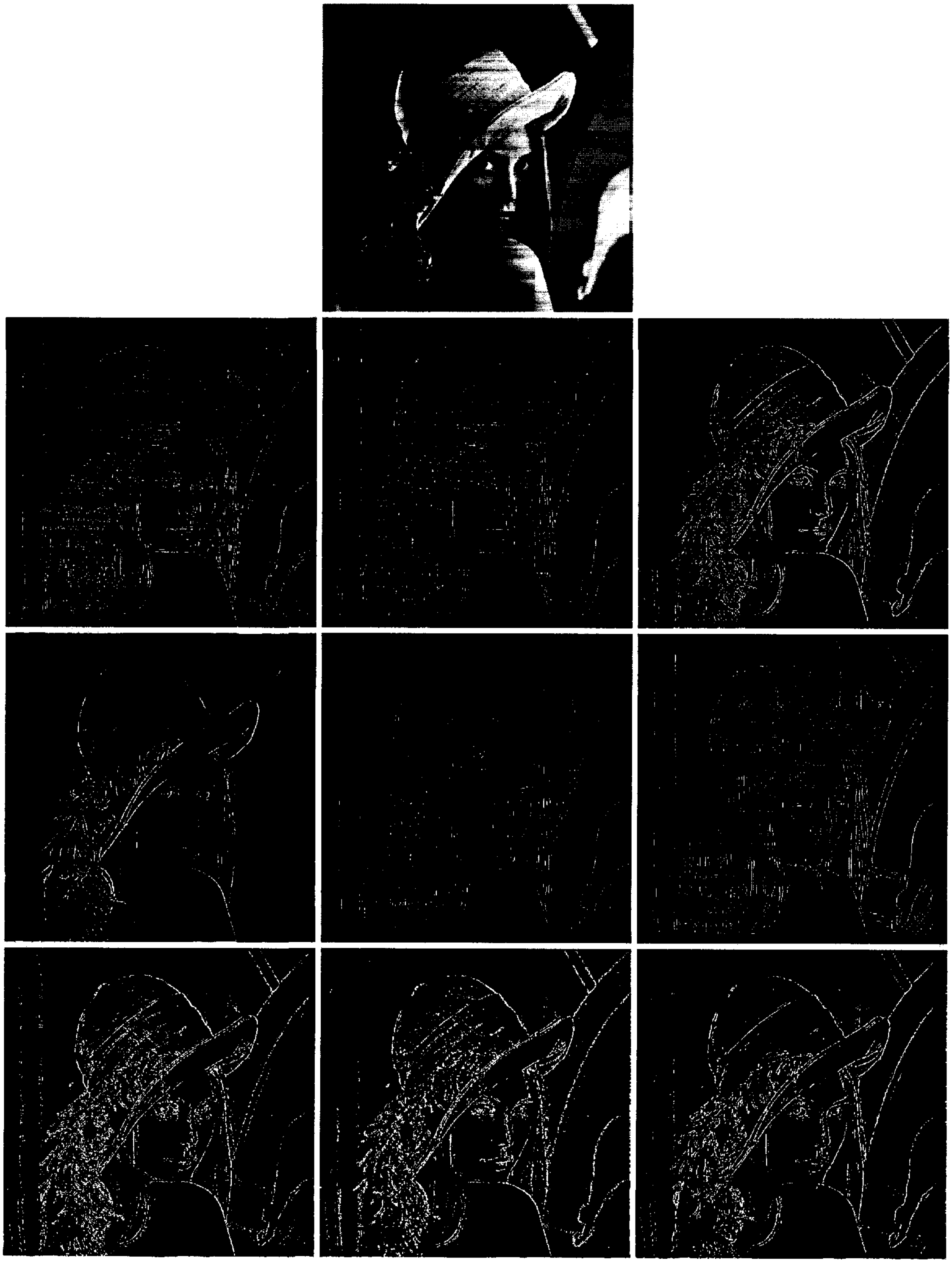

FIG. 15: "Lena": (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=2); (third-row-center) 2-of-3 (EL=2); and (third-row-right) LNE (EL=1).

but not very good at the finer details because of the inherent blurring associated with the processes.

The original "Lena" image exhibits some minor noise. The first derivative methods would produce spurious edges as can be seen in Figure 15 because noise introduces gradient transitions. Our proposed methods produce strong edges and are comparable in performance to the other methods. The LNE method produced relatively thinner edges and less noise than others but lost some details. Both the 3-of-3 and LNE methods produce finer edges than the 2-of-3 method. While the edge results using the LoG are good, some small details are lost. Canny produces the cleanest results and single response for each edges, but edge terminals are slightly distorted.

Noise with SNR = 1

Figures 16 and 17 show the edge images obtained with the different methods used in Figures 12 and 13 except the case where heavy noise (SNR = 1) has been introduced into the image. Here, for the multi-scale methods, relatively larger scales are used to decrease the impact of noise, which results in some details not being preserved in all three scales and, hence, not being detected as edges. The 2-of-3 algorithm displays to advantage in this case: while it finds more pixels as edges and is, hence, noisier, we can use connectivity analysis to eliminate the edges due to noise. The LNE algorithm is better than the 3-of-3 algorithm but a little worse than the 2-of-3 method. The first derivative methods such as Sobel, Prewitt, and Roberts operator are badly affected by the noise since noise causes edge transitions. LIH seems to lose connectivity since it is primarily a point detector. LoG exhibits better performance than the first derivative methods but is not especially good because of the single kernel. The Canny operator is good at suppressing noise and detecting edges because it uses a Gaussian filter to smooth the image before edge localization, but greater noise reduction results in a number of small distortions—curves or projections at the edge terminals.

Additional experimental results for the SNR = 1 case are shown in Figure 18. For "Barbara," our proposed algorithms give better results than other algorithms. Even in this case of extremely poor SNR, they can separate the small edges, for example, on the table cover. Furthermore, they robustly detect the main structures in this image, such as the contour of Barbara, the bookshelf, and the tables. While the edges produced by the 3-of-3 and LNE methods are thinner than those produced by the 2-of-3 method, the 2-of-3 detected more complete edges. The first derivative methods are not effective at all because they do not employ noise-reduction. Although LoG and Canny do perform noise-reduction, their results are not as good as those produced by our algorithms.

Similarly, for the case when the "Lena" image is corrupted by noise with SNR= 1 (Figure 19), the first derivative methods do not properly detect the edges. The LoG managed to detect a few edges scattered across the image, but it also let more noise through. The filtering process of Canny introduced bad distortions in the result but got a clean and partial contour. Our algorithms obviously produce better results than others. The distortion is less than Canny, and more edges are extracted such as the long straight lines in the left side of the image. While the edges produced by the 3-of-3 and LNE methods are thinner than those produced by the 2-of-3 method, the 2-of-3 found more complete edges.

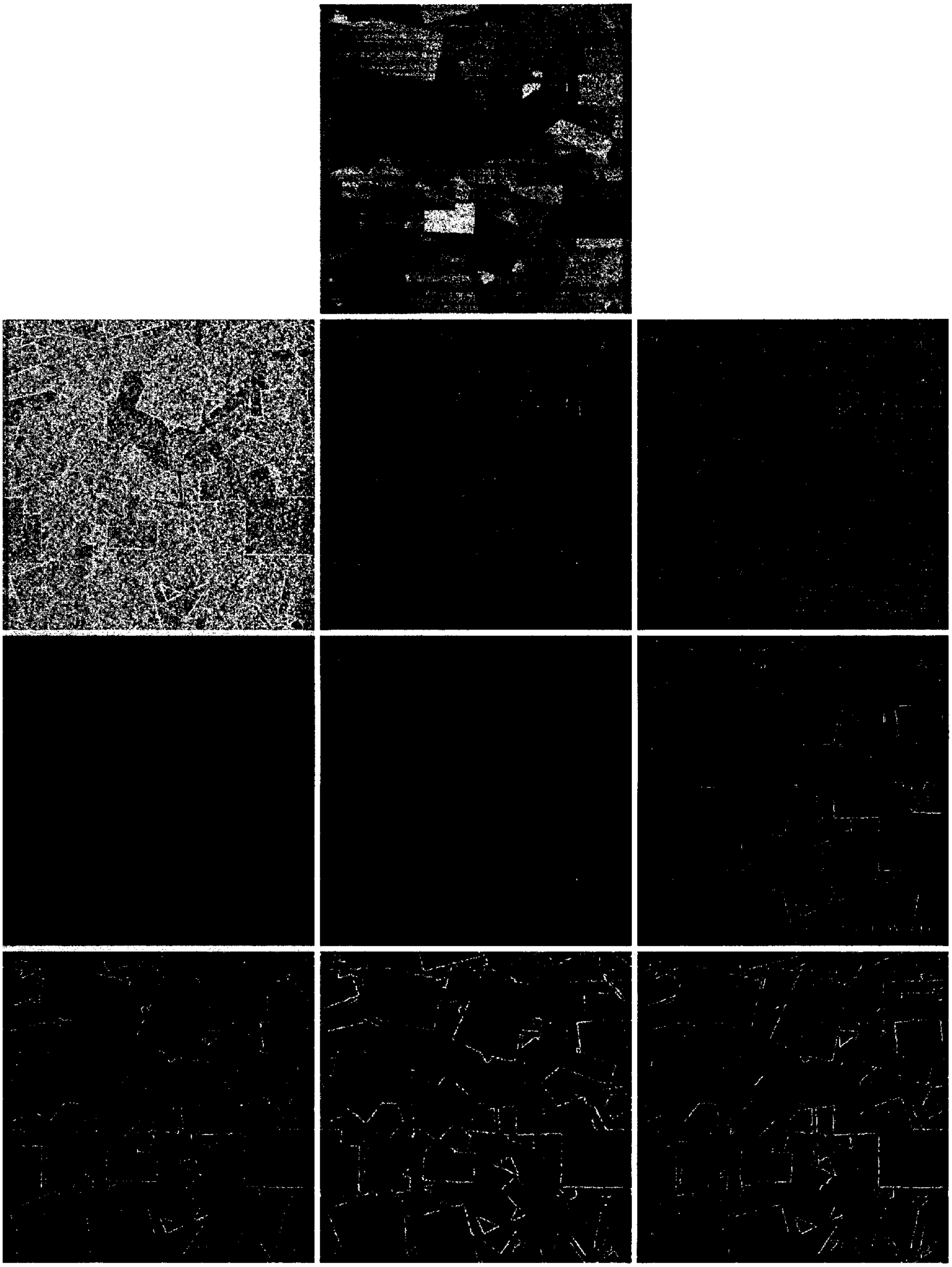

FIG. 16: “Rect” SNR= 1: (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=3); (third-row-center) 2-of-3 (EL=3); and (third-row-right) LNE (EL=3).

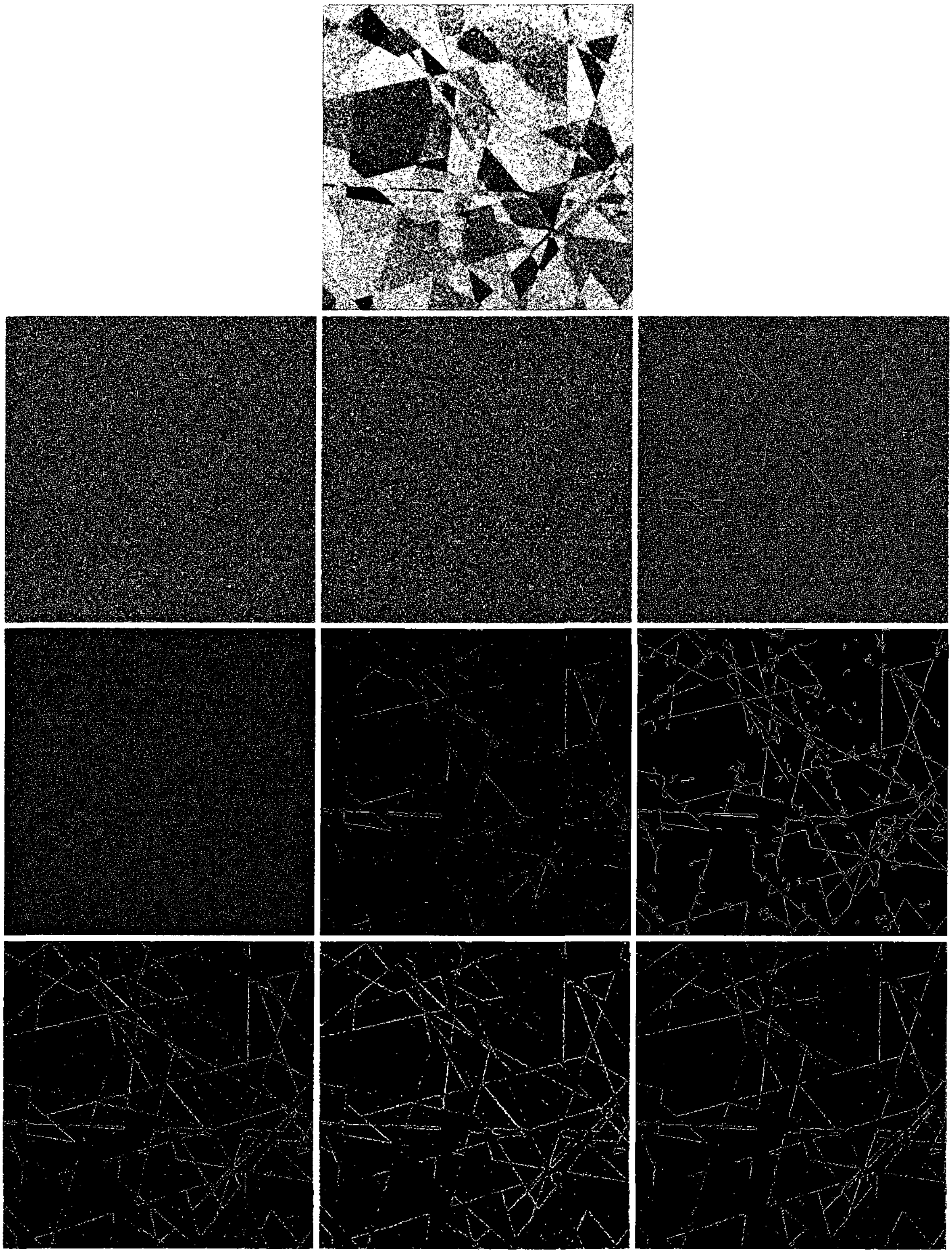

FIG. 17: "Randpoly" SNR= 1: (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=2); (third-row-center) 2-of-3 (EL=3); and (third-row-right) LNE (EL=2).

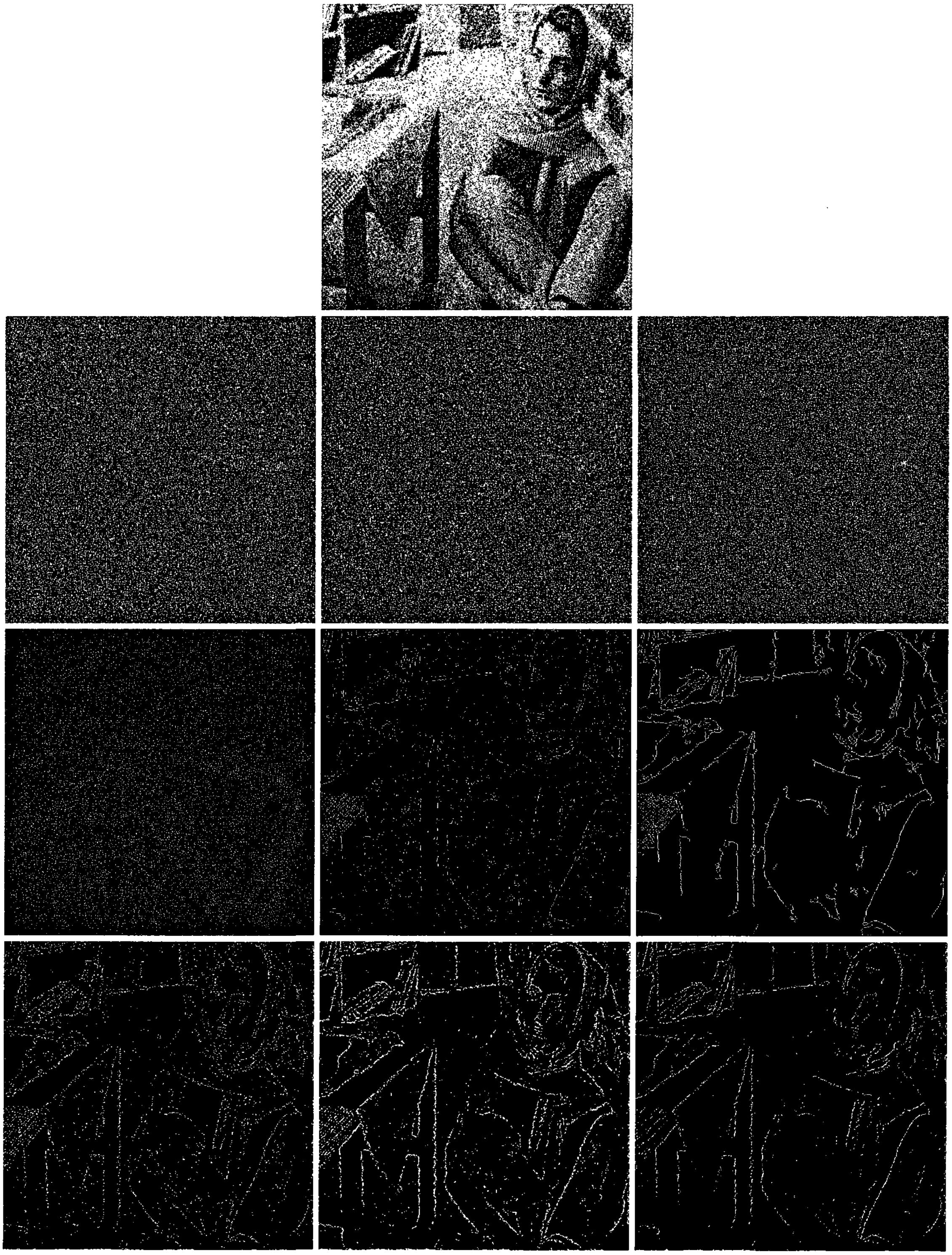

FIG. 18: "Barbara" SNR= 1: (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=2); (third-row-center) 2-of-3 (EL=3); and (third-row-right) LNE (EL=2).

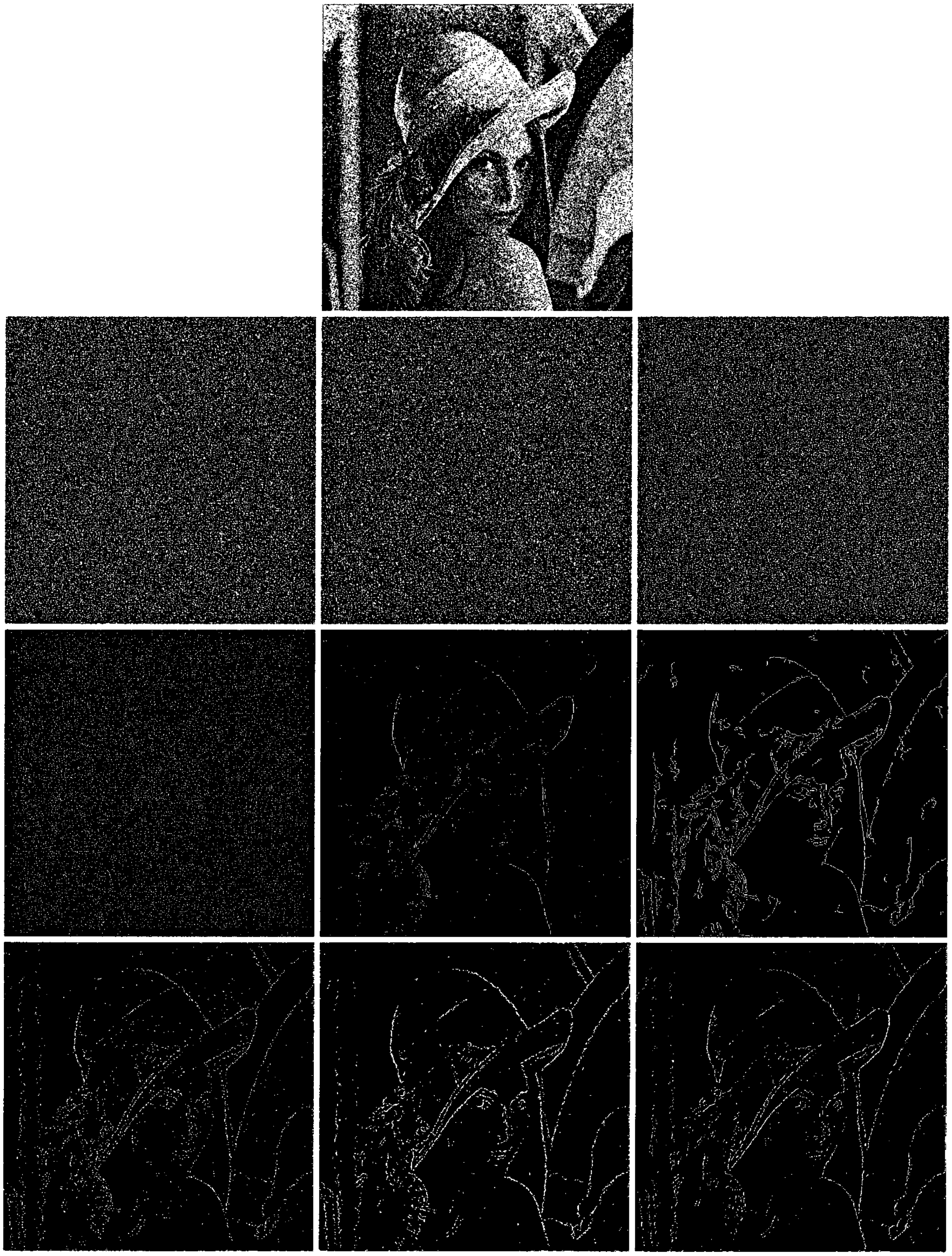

FIG. 19: "Lena" SNR= 1: (top-row-left) Sobel; (top-row-center) Prewitt; (top-row-right) Roberts; (second-row-left) LIH; (second-row-center) LoG; (second-row-right) Canny; (third-row-left) 3-of-3 (EL=2); (third-row-center) 2-of-3 (EL=3); and (third-row-right) LNE (EL=2).

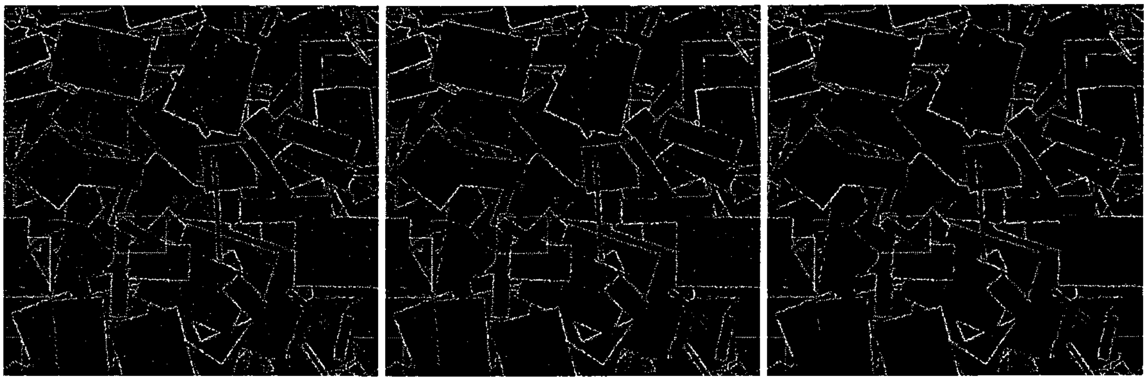

FIG. 20: The effect of connectivity analysis: (left) No connectivity analysis; (center) EL = 3; and (right) EL = 5.

Figure 20 shows the effect of the connectivity analysis. The original edge image is generated by using the 2-of-3 algorithm. The condition $EL = 1$ represents the case where all the edges are passed through, i.e., no connectivity analysis is performed. For $EL > 1$, the difference is obvious: as EL increases, the small edges appearing frequently in the center of blocks of smooth regions away from the authentic boundaries are increasingly removed. However, the algorithm takes longer to process the image since multiple neighborhoods need to be examined at each pixel location (See Section II.2.2). In our experiments, the size of the image would be 512×512. When $EL < 4$, the computation time is less than 1 second in C/C++ compiler on 2.40GHz Intel Core2 CPU. However, when $EL \geq 5$, the process would take about 5 seconds or more to finish connectivity analysis. Thus, by help of connectivity analysis, the edges with prescribed lengths are classified as signal and kept, while those with lengths shorter than EL are classified as noise and eliminated using smoothing.

IV.3 NOISE REDUCTION RESULTS

IV.3.1 Comparison for Computer-generated Images

Recall that the overall denoising process relies on the edge-detection mechanism followed by noise suppression at those locations where the pixels are not classified as edge pixels. We also introduced the idea of edge connectivity analysis to determine which edge pixels were actually produced due to noise. In order to determine the effectiveness of this approach, we performed two different group tests. Figure 21 shows the output of the traditional noise reduction filters: the median filter, the AMF, or the ATMF. Also, as a final comparison, the noise-reduction achieved by applying just the smoothing filter is shown. In Figure 22, the results of applying edge detection for noise reduction are shown for our proposed methods and other edge detection algorithms.

After edge detection, we can either apply:

1. a blurring filter to every pixel that is not an edge pixel without performing edge connectivity analysis to determine if it were a noise pixel or not, or

2. edge analysis with different edge length requirements and further reduce the total number of edge pixels by eliminating those that were classified as noise.

Experiments show that using connectivity analysis, for example, with the 3×3 edge neighborhood, eliminates noise along the boundaries of regions with intensity transitions and produces an overall less noisy image. Increasing EL would, additionally, remove isolated small edges, further reducing the noise along the boundaries of regions. Larger connectivity neighborhoods produce smoother results, yet, the impact on sharpness and contrast is

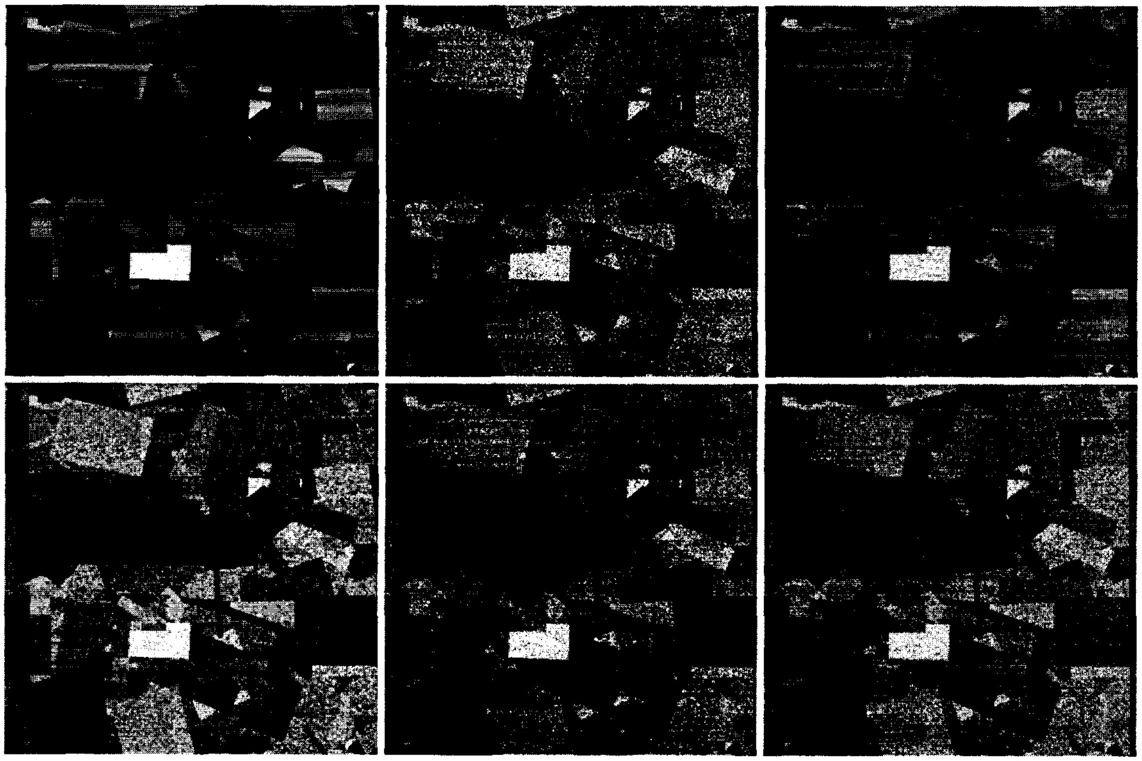

FIG. 21: (top-row-left) Original image, G; (top-row-center) noisy image, $G_{SNR=1}$, SNR = 1; (top-row-right) Mean filter; (second-row-left) 3 × 3 Median filter; (second-row-center) 7 × 7 AMF; and (second-row-right) 7 × 7 ATMF.

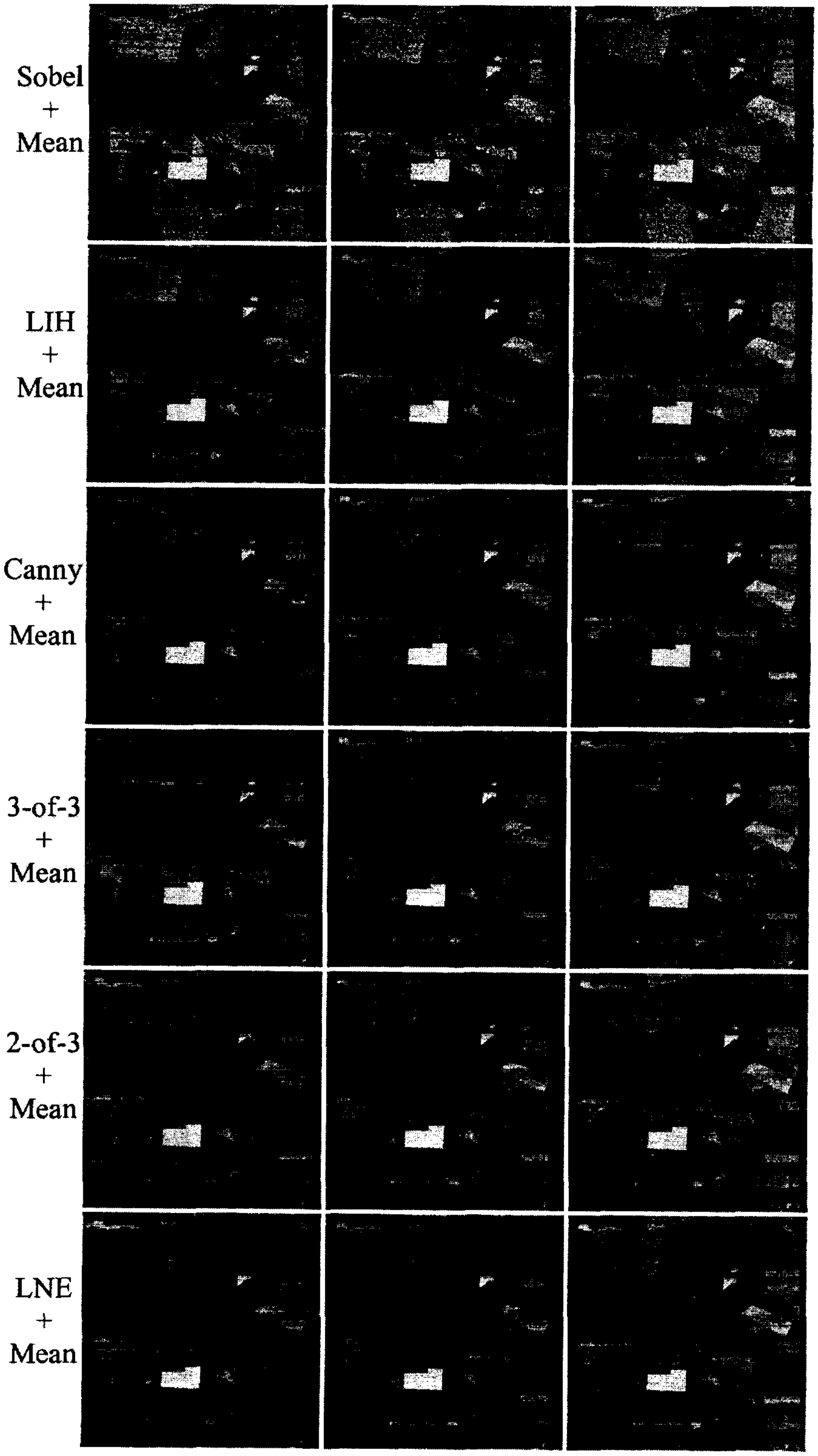

FIG. 22: (left) without edge analysis; (center) EL = 3; and (right) EL = 5.

minimal. The 3-of-3, 2-of-3 and LNE methods are able to eliminate more noise than Sobel and LIH because of better edge detection and localization. The performance of the Canny operator is similar to that of the proposed method. Compared with our proposed method, the traditional methods are less effective in very low SNR environments. While the mean filter produces good results in terms of eliminating visible noise distortion, it blurs the edges severely—depending on the SNR—limiting the utility of the processed image for further operations. The AMF does not produce good results for the additive white Gaussian noise. Similarly, the ATMF does not perform well even though it is good at reducing multi-layer impulse noise [25].

IV.3.2 Fidelity Assessment

A commonly used metric of similarity between two images G_1 and G_2 is the *fidelity*, $\mathcal{F}(G_1, G_2)$ defined as

$$\mathcal{F}(G_1,G_2) = 1 - \frac{\sum\limits_{m=0}^{M-1}\sum\limits_{n=0}^{N-1} (G_1(m,n) - G_2(m,n))^2}{\sum\limits_{m=0}^{M-1}\sum\limits_{n=0}^{N-1} G_1(m,n)^2}, \tag{52}$$

where $\mathcal{F} = 1$ when $G_1 = G_2$. The fidelity metric corresponds fairly closely with visual judgment for comparing images. In order to measure the performance of our noise-reduction approach, we compute $\mathcal{F}(G, G_p)$, where G_p is variously produced by edge-directed noise reduction using the Sobel, LIH, Canny, and our two methods, and G is the original noise-free image. The results are tabulated in Table 1.

Using $\mathcal{F}$ for assessing the performance, it is clear that the connectivity analysis method indeed improves noise reduction for LIH, Canny, and our proposed method. Furthermore,

Table 1: Fidelity improvement with noise reduction.

	Fidelity ($\mathcal{F}$)		
	Without connectivity analysis	**With connectivity analysis**	
		$EL = 3$	$EL = 5$
Noisy Image (SNR=1)	0.779	-	-
Sobel+Mean filter	0.801	0.801	0.801
LIH+Mean filter	0.915	0.928	0.938
Canny+Mean filter	0.958	0.958	0.958
3-of-3+Mean filter	0.957	0.958	0.959
2-of-3+Mean filter	0.954	0.955	0.956
LNE+Mean filter	0.955	0.957	0.958
Mean filter (3×3)	0.959	-	-
Median filter (3×3)	0.805	-	-
AMF (7×7)	0.884	-	-
ATMF (7×7)	0.873	-	-

increasing the size of the connectivity neighborhood can improve $\mathcal{F}$ by as much as 2% for LIH. However, there is no obvious impact when the Sobel operator is used. Also, $\mathcal{F}$ for median filter and AMF are not good: both produce denoised images with a contrast that is poorer than that of the original image. The effect on ATMF is also not good: while it removes a little more noise than the AMF, its fidelity is worse. The mean filter performs about as well, in terms of $\mathcal{F}$, as the proposed noise reduction. We can explain this (slightly) unexpected result by the observation that $\mathcal{F}$ is a gross measure of visual similarity, so a blurred image compared with its original unblurred version would typically result in a high $\mathcal{F}$. Additionally, the experimental results show that both the proposed methods and the Canny operator are effective in preserving edges and removing noise.

IV.3.3 Comparison of Real Images

In addition to the results based on computer generated images, we also applied the proposed algorithms to real images. In Figure 23(a), the "Barbara" image, which has areas of varying edge densities, was used to test our algorithms. Since we are looking at the extreme case of SNR=1, we do not show the result for the derivative-based methods because their performance is extremely poor for this condition (Figure 12). Therefore, we only show the results of the Canny algorithm and our algorithms. All methods performed well in detecting edges. Then, we used the computed edge images to direct the mean filter to reduce image noise. In comparison with the result of the mean filter alone, the results generated by combining edge detection algorithms and the mean filter are better because the edges were preserved during smoothing. The Canny operator helped the mean filter to remove the most noise but also resulted in phantom edges due to hysteresis thresholding. Our methods extracted more edges than Canny, especially the 2-of-3 method, but also allowed more noise to go through. However, the greater number of edges led to a greater number of features being preserved. For the LNE algorithm, the edges were thinner and cleaner. It also had less noise than the 3-of-3 and the 2-of-3 methods. The fidelity results of the "Barbara" image are tabulated in Table 2. All the methods achieved good results in terms of the fidelity metric.

In Figure 23(b) another example, "Lena," has been used. This image not only has straight edges but also curved and vanishing ones. The mean filter attenuated both features and noise. The Canny operator did well in noise reduction, but, since it used a relatively larger scale for the Gaussian filter, some of the features, for example Lena's eyes, were

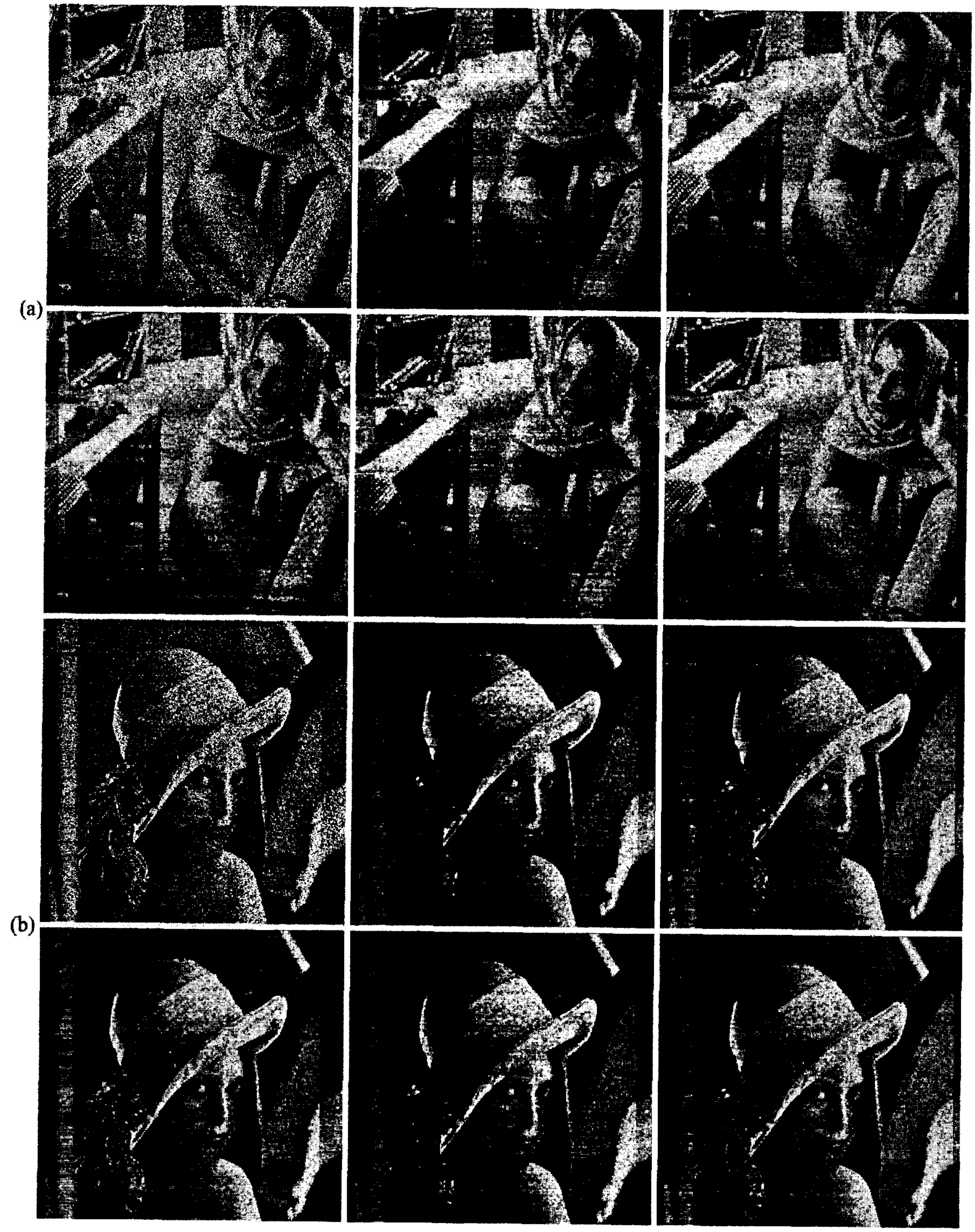

FIG. 23: Noise reduction for real images: (a) **Barbara**; (b) **Lena**: clockwise from top-left: Image with SNR = 1; Mean filter (MF); Canny + MF; 3-of-3 + MF; 2-of-3; LNE + MF.

Table 2: Fidelity improvement with noise reduction.

Fidelity	Original Image	Noisy Image (SNR = 1)	Canny + MF	3-of-3 + MF	2-of-3 + MF	LNE + MF	Mean filter
Barbara	1.00	0.82	0.97	0.97	0.97	0.97	0.97
Lena	1.00	0.85	0.96	0.96	0.96	0.97	0.97
MF = Mean filter							

not completely preserved. The 3-of-3 and the 2-of-3 methods are better than Canny in preserving the features, but they have more noisy outputs. This depicts the trade-off between signal preservation and noise reduction. LNE is better than other methods because it preserved the most edge structures, such as the hat and the mirror, but let less noise through. Table 2 shows the LNE is best in terms of the fidelity comparison.

During our experiments we found that the value of the parameters in Section II.2.1 needed to be changed depending upon the value of the SNR. From observation we see that to find a thinner edge σ_1 should be decreased and τ_m should be increased; otherwise, more edges and more noise would be produced. At the same time, due to small σ_1, the residual noise would be higher and affect the edges, making it harder to differentiate between edges and noise. Under these conditions, though the edge might be thinner, the noise would be worse not just around the true edges but also spread across non-edge areas. However, since we do not use all the resolutions, reducing σ_1 is equivalent to using a different set of resolutions that correspond to the same σ_1 value, so the overall impact of changing σ_1 is in selecting different resolutions that do not impact overall performance. The same cannot be said, however, about τ_m. Figure 24 shows that the SNR and τ_m are inversely related: as SNR increases, τ_m should decrease. This regularity was found by statistical analysis

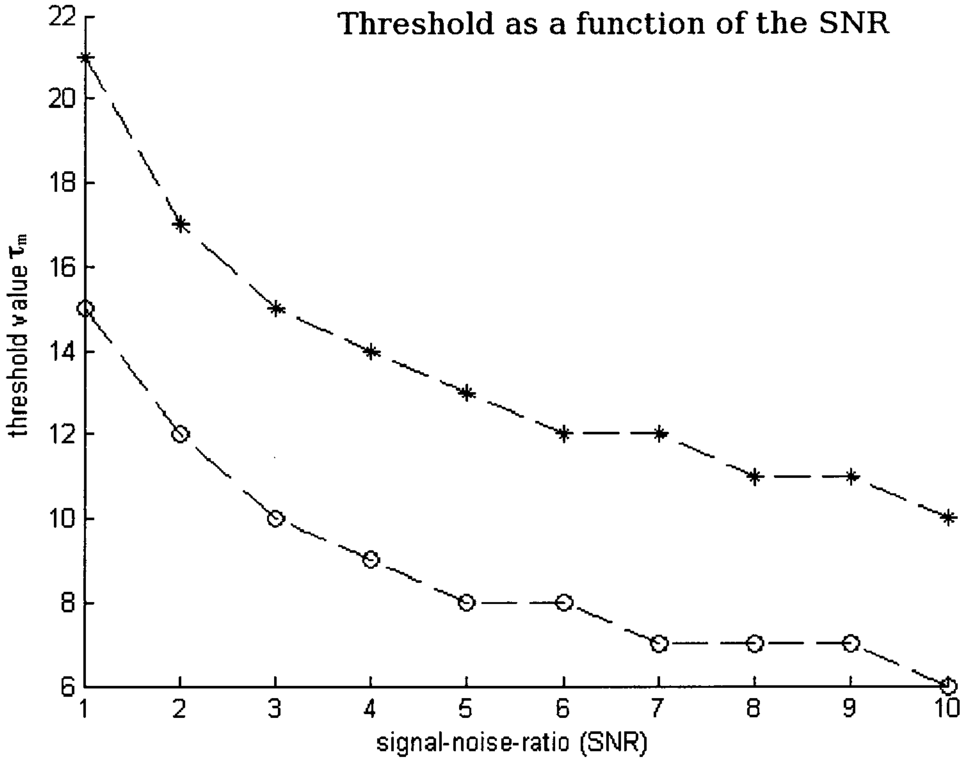

FIG. 24: Magnitude threshold τ_m as a function of the SNR: $*$ color data; $\circ$ grayscale data.

across different color and gray images. We generated 40 noise-free images for both color and gray images. They were corrupted by Gaussian white noise with different SNR from 1 to 10. Then, we applied our edge detection methods to those images and examined the regularity of τ_m. This regularity makes intuitive sense since less noise in the image (higher SNR) corresponds to fewer false edges in the image corresponding to noise and, hence, does not require a larger threshold to eliminate such edges.

CHAPTER V

RESULTS OF INFORMATION-THEORETIC ANALYSIS

V.1 PSDS OF EDGE DETECTORS

As shown in Section III.1, the information capacity $\mathcal{H}$ requires the PSD of its corresponding process, such as the image gathering function or the scene. Similarly, to evaluate the edge detection process, τ_e, in the context of the visual communication channel, we need to determine its PSD. The PSD of the edge-detection process falls into one of three major categories:

1. The edge detection algorithm has a close-formed analytic expression in the spatial domain;
2. It does not have a close-formed analytic expression, but it is linear and shift-invariant;
3. It does not belong to either category 1 or category 2. For example, it is a non-linear and shift-variant operator.

The SFR of the LoG or other traditionally used edge-detection operators that can be described with close-formed analytic functions in the spatial domain can be easily obtained by the Fourier transform (FT) of the analytic function:

$$\hat{\tau}_e(\mu, \nu) = \int\int \tau_e(x, y) \exp(-i2\pi(\mu x + \nu y)dx dy. \tag{53}$$

The PSD of a function is defined as the FT of its autocorrelation. This is equivalent to taking the magnitude of the SFR for stationary signals.

Typically, first derivative operators used for edge-detection are defined strictly in the spatial domain, and their SFR is not examined. However, for the information theoretic analysis, a spatial frequency domain representation is needed. To evaluate Sobel, and other kernel-based operators which do not have closed-form analytic expressions, we use two methods to determine their PSDs. The first method is simulation based and physically computes the DFT of the convolution kernels, while the second method derives the DFT theoretically. Identical results are obtained using the two methods.

V.1.1 Simulation Derivation of PSDs

The procedure for computing the PSDs of the edge-detectors using simulations is as follows:

1. Build a blank $M \times N$ image I (all pixel values are 0);
2. Place the $p \times q$ edge operator kernel in the center of I;
3. Compute the frequency response $\hat{I}$ using the DFT;
4. Compute $|\hat{I}|^2$ which is the PSD for a zero-mean process.

The PSD of the operator is then equivalent to $|\hat{I}|^2$.

V.1.2 Theoretical Derivation of PSDs

The theoretical derivation makes use of the commonly used kernels for the Sobel, Roberts and Prewitt operators and expands them using signal processing techniques.

Sobel

The Sobel operator is typically described using two convolution kernels: $S_H(x,y)$ or $S_V(x,y)$. We can expand these kernels using the Dirac delta function $\delta(\,)$ in the following manner:

$$\begin{aligned} S_H(x,y) &= \sum S_H(m,n)\delta(x-m,y-n), \\ S_V(x,y) &= \sum S_V(m,n)\delta(x-m,y-n), \end{aligned} \tag{54}$$

where $S_H(m,n)$ and $S_V(m,n)$ are the values of $S_H(x,y)$ and $S_V(x,y)$ at the nodes of the sampling lattice. Based on standard Sobel kernels [7], the two kernels can then be expressed as

$$\begin{array}{rllllll} S_H(x,y) = & -1\times\delta(x+1,y+1) & + & 0\times\delta(x+1,y) & + & 1\times\delta(x+1,y-1) \\ & -2\times\delta(x,y+1) & + & 0\times\delta(x,y) & + & 2\times\delta(x,y-1) \\ & -1\times\delta(x-1,y+1) & + & 0\times\delta(x-1,y) & + & 1\times\delta(x-1,y-1), \\ S_V(x,y) = & -1\times\delta(x+1,y+1) & - & 2\times\delta(x+1,y) & - & 1\times\delta(x+1,y-1) \\ & +0\times\delta(x,y+1) & + & 0\times\delta(x,y) & + & 0\times\delta(x,y-1) \\ & +1\times\delta(x-1,y+1) & + & 2\times\delta(x-1,y) & + & 1\times\delta(x-1,y-1). \end{array} \tag{55}$$

Simplifying,

$$\begin{array}{rllllll} S_H(x,y) = & -\delta(x+1,y+1) & + & \delta(x+1,y-1) & - & 2\delta(x,y+1) \\ & +2\delta(x,y-1) & - & \delta(x-1,y+1) & + & \delta(x-1,y-1), \\ S_V(x,y) = & -\delta(x+1,y+1) & - & 2\delta(x+1,y) & - & \delta(x+1,y-1) \\ & +\delta(x-1,y+1) & + & 2\delta(x-1,y) & + & \delta(x-1,y-1). \end{array} \tag{56}$$

To obtain the SFR, we take the Fourier transforms of the kernels. Using the shift-in-space relationship between the spatial and spatial frequency domain representations, and the identity that $\mathcal{F}(\delta) = 1$, we can combine the various terms and obtain the following simplified forms for the two kernels:

$$\begin{aligned}\tilde{S}_H(\mu,\nu) &= -i4\sin(2\pi\nu)(1+\cos(2\pi\mu)),\\ \tilde{S}_V(\mu,\nu) &= -i4\sin(2\pi\mu)(1+\cos(2\pi\nu)).\end{aligned} \tag{57}$$

Note that the presence of the i in both terms indicates that these are derivative operators. Hence, we can write the Sobel operator as

$$\hat{\tau}_e(\mu,\nu) = \left|\tilde{S}_H(\mu,\nu)\right| + \left|\tilde{S}_V(\mu,\nu)\right|. \tag{58}$$

Prewitt

Following the development in Section V.1.2, the Prewitt kernels can be expressed as

$$\begin{aligned}P_H(x,y) = {} & -\delta(x+1,y+1) + \delta(x+1,y-1) - \delta(x,y+1)\\ & + \delta(x,y-1) + \delta(x-1,y+1) + \delta(x-1,y-1),\\ P_V(x,y) = {} & -\delta(x+1,y+1) - \delta(x+1,y) - \delta(x+1,y-1)\\ & + \delta(x-1,y+1) + \delta(x-1,y) + \delta(x-1,y-1).\end{aligned} \tag{59}$$

After computation, their frequency responses are given as

$$\begin{aligned}\tilde{P}_H(\mu,\nu) &= -i4\sin(2\pi\nu)(1/2+\cos(2\pi\mu)),\\ \tilde{P}_V(\mu,\nu) &= -i4\sin(2\pi\mu)(1/2+\cos(2\pi\nu)).\end{aligned} \tag{60}$$

Consequently,

$$\hat{\tau}_e(\mu,\nu) = \left|\tilde{P}_H(\mu,\nu)\right| + \left|\tilde{P}_V(\mu,\nu)\right|. \tag{61}$$

Roberts

As in V.1.2, the Roberts masks can be explicitly expressed as two separate two dimensional signals. Then, they can be expressed as

$$\begin{aligned} R_x(x,y) &= -\delta(x,y)+\delta(x-1,y-1), \\ R_y(x,y) &= -\delta(x,y-1)+\delta(x-1,y). \end{aligned} \tag{62}$$

After computation, their frequency responses are

$$\begin{aligned} \tilde{R}_x(\mu,\nu) &= -1+\cos(2\pi(\mu+\nu))-i\sin(2\pi(\mu+\nu)), \\ \tilde{R}_y(\mu,\nu) &= \cos(2\pi\mu)-\cos(2\pi\nu)-i[\sin(2\pi\mu)-\sin(2\pi\mu)]. \end{aligned} \tag{63}$$

Consequently,

$$\hat{\tau}_e(\mu,\nu) = \left|\tilde{R}_x(\mu,\nu)\right| + \left|\tilde{R}_y(\mu,\nu)\right|. \tag{64}$$

Laplacian-of-Gaussian

The LoG is a second-order derivative operator. The spatial and spatial frequency response of this operator, respectively, are

$$\tau_e(x,y) = \frac{1}{\pi\sigma_e^4}\left(1-\frac{r^2}{2\sigma_e^2}\right)\exp\left(-\frac{r^2}{2\sigma_e^2}\right), \tag{65}$$

and

$$\hat{\tau}_e(\mu,\nu) = (2\pi\rho)^2\exp(-2(\pi\sigma_e\rho)^2), \tag{66}$$

where $r^2 = x^2+y^2$, $\rho^2 = \mu^2+\nu^2$, σ_e is the standard deviation of the Gaussian function, and $\rho = 1/\sqrt{2}\pi\sigma_e$ is the spatial frequency at which $\hat{\tau}_e$ is maximum.

Canny

There are a number of edge detection algorithms, such as our proposed edge detection algorithms, that fall into the third category: i.e., they neither have closed form analytic descriptions, nor are they linear and shift-invariant. In this dissertation, we describe how the SFR of these operators can be approximated using experimental methods. We use the Canny operator as an example for this process.

Recall that we use the simplified model of the end-to-end imaging process to evaluate the performance of the edge detection operators. The output image was given by Equation 14 which is reproduced below:

$$R[x,y] = ((s(x,y) * \tau(x,y))\underline{|||}(x,y) + N_{a/d}[x,y] + N_e[x,y]) * \Psi(x,y).$$

If we let $\Psi(x,y) = \tau_e(x,y)$ be the Canny operator, then $R(x,y)$ is the edge-image. Additionally, if we ignore the influence of every noise source, we can rewrite Eq. 14 as

$$R[x,y] = (s(x,y) * \tau(x,y)) * \tau_e(x,y). \tag{67}$$

The SFR τ_e could then be described using the following:

$$\tilde{R}[\mu,\nu] = (\hat{s}(\mu,\nu)\tilde{\tau}(\mu,\nu))\hat{\tau}_e(\mu,\nu). \tag{68}$$

If the scene $\hat{s}(\mu,\nu)$, the image-gathering process $\tilde{\tau}(\mu,\nu)$ and the edge image $\tilde{R}[\mu,\nu]$ are known, the SFR of the Canny operator in frequency domain could be approximated by

$$\hat{\tau}_e(\mu,\nu) = \frac{\tilde{R}[\mu,\nu]}{\hat{s}(\mu,\nu)\tilde{\tau}(\mu,\nu)}. \tag{69}$$

This can be used in Eq. 37 to compute the information capacity associated with the Canny operator. However, since this result is valid for only a single input image $s(x,y)$, we use an ensemble average to actually approximate the PSD.

V.1.3 PSDs Results

Derivative-based Algorithms

In Figures 25 and 26, the PSDs of different derivative edge detectors are shown as 3-dimensional plots. These were individually computed using both the simulation technique outlined in Section V.1.1 and the theoretical derivation in Section V.1.2. In our experiments, the size of the image was set as 1024 × 1024 for both methods.

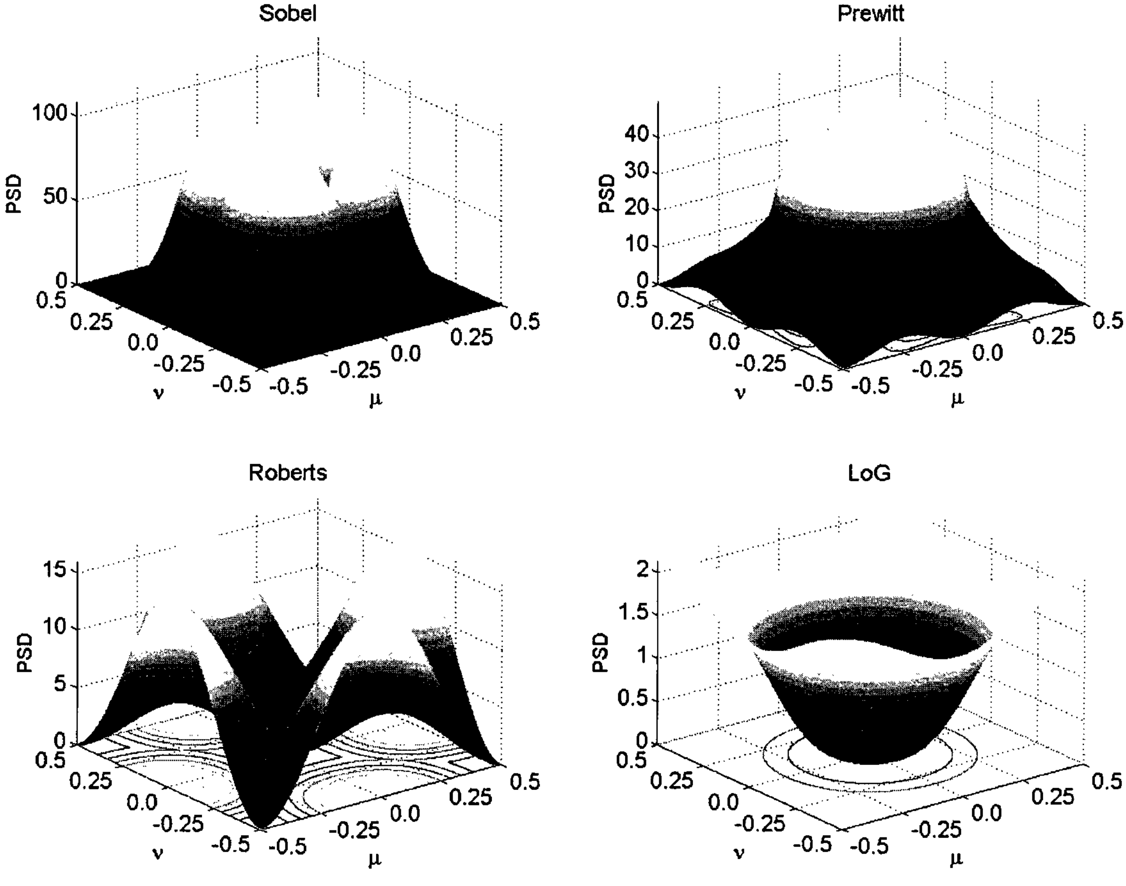

FIG. 25: Power Spectral Densities of Edge Detectors by Real Simulation.

For the Sobel, Prewitt and Roberts operators, the PSDs generated by using the simulation based method agree perfectly with the theoretical derivation. For the LoG operator, the PSD of its 5 × 5 approximation mask is shown in Figure 25, and, the PSD of the LoG based on its SFR, where $\sigma_e = 0.5$, is shown in Figure 26. They are almost identical, but the amplitude and the point of inflexion are slightly different.

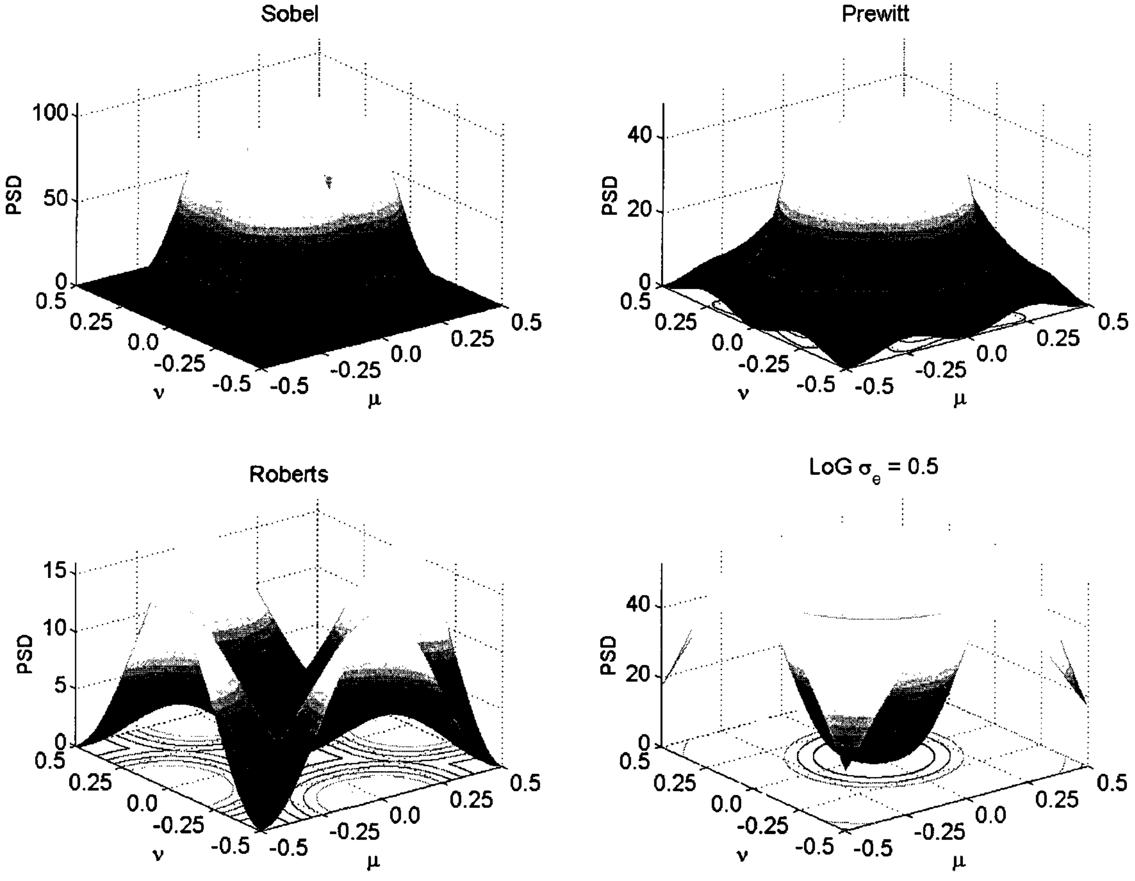

FIG. 26: Power Spectral Densities of Edge Detectors by Theoretical Derivation.

Canny Operator

Since the Canny operator is not a linear and shift-invariant operator, we have to estimate its PSD associated with different conditions, such as the mean spatial detail and the image-gathering process parameters. To compute its PSD, a group of computer generated images with mean spatial detail ranging from 1 to 10 were used. A set of 100 images was used for each mean spatial detail, and the ensemble averages were used to compute the PSD. For the image-gathering process, the ρ_c was changed from 0.1 to 1.0. The sharpness or blurring of digital image $p[x,y]$ varied based on different ρ_c as stated in Section III.1.1. Therefore, images with different sharpness and blurring conditions were used to find the properties of the Canny operator.

Returning to the steps of the Canny operator, because we are using a noise-free image, the filtering procedure in the first step of the Canny algorithm needs to be adjusted. We set the scale of the Gaussian filter as a constant in the first step with $\sigma = 0.3$, which was determined by a series of experiments. The second and third steps of the Canny algorithm remain unchanged. Looking at the Canny operation more closely, the output of the Canny operator after the first three steps is actually an edge image. However, these edges have different magnitudes. To further differentiate the edges and non-edges, the Canny algorithm makes use of a fourth step: hysteresis thresholding. Since our images are noise-free, the fourth step is omitted in our PSD approximation.

Thus, the procedure is that for each set of images with a given ρ_c and ζ, the simulation method is used to compute the individual PSD. The SFR τ_e is computed by averaging the results across all the different images that correspond to a given set of ζ and ρ_c. This does not provide a "general" SFR for the Canny operator. Rather, one can choose one of several depending upon the blurring coefficient ρ_c and the mean spatial detail ζ. The results obtained from the process are not "smooth." Thus, a curve fitting procedure is used to model the final PSD. During the curve fitting process, we tried several models such as the Gaussian, polynomial, exponential, and power functions. The Gaussian model was found to have the most suitable fit to the data.

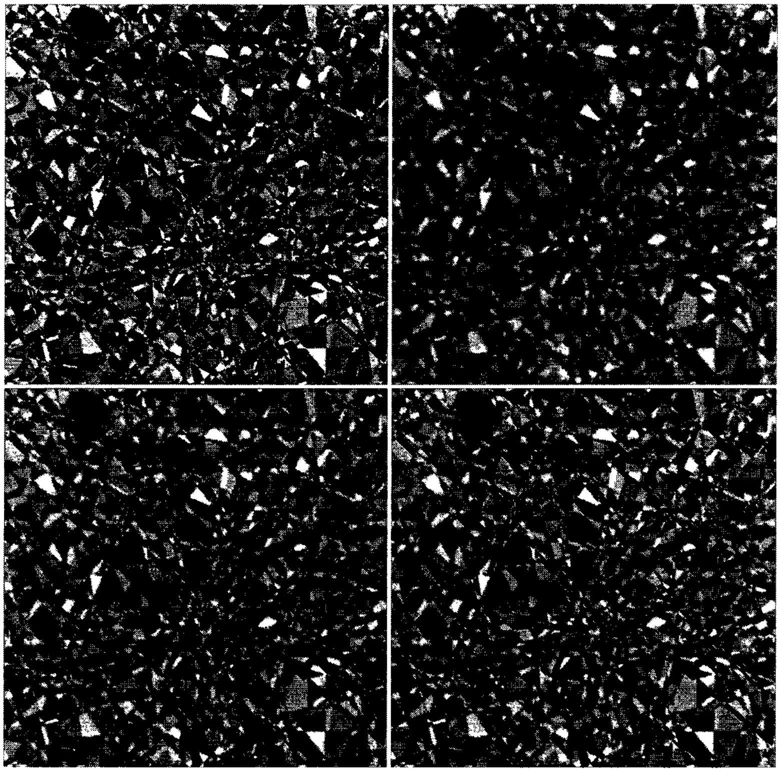

FIG. 27: One simulation example: (top-left) original image; (top-right) blurred by $\rho_c = 0.3$; (bottom-left) blurred by $\rho_c = 0.5$; and (bottom-right) blurred by $\rho_c = 0.7$.

Figure 27 shows one example of a computer generated image with $\zeta = 1$, which corresponds to the finest mean spatial detail with respect to the inter-sample distance. At this mean spatial detail the average distance between areas of different intensity values is equal to the distance between sampling elements on the CCD, so the image contains a lot of details (Figure 27 (top-left)). The image is generated at a high resolution and is used as the "scene" in the end-to-end model. The other three images in Figure 27 show the results after the image-gathering process for $\rho_c = \{0.3, 0.5, 0.7\}$.

A group of results for the Canny operator based on Figure 27 is shown in Figure 28. It is obvious that the larger the ρ_c the clearer and more complete the edges in the output

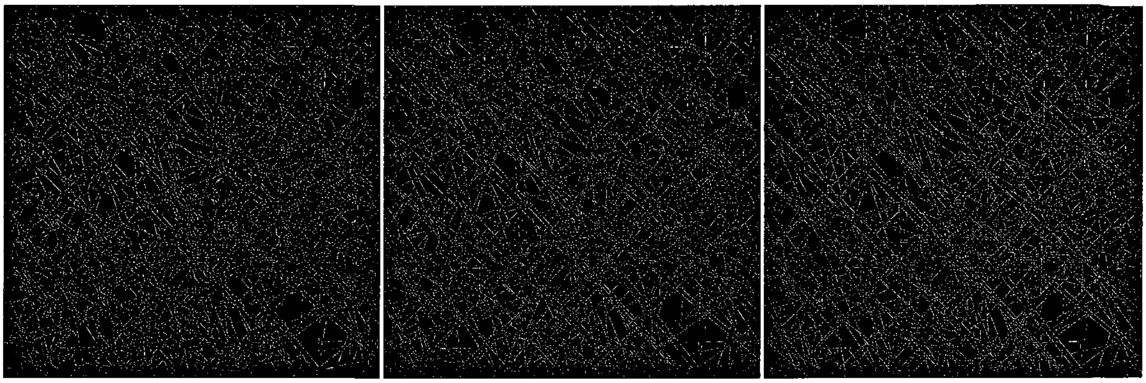

FIG. 28: Examples of Canny operator: (left) blurred by $\rho_c = 0.3$; (middle) blurred by $\rho_c = 0.5$; and (right) blurred by $\rho_c = 0.7$.

image. The corresponding PSDs for the Canny operator are shown in Figure 29. The data is then curve fitted using the Gaussian function:

$$f(x) = a\exp\left[-\left(\frac{x-b}{c}\right)^2\right]. \tag{70}$$

Table 3 shows the parameters of the Gaussian model for different imaging conditions. In Figure 30, the curve fitted results for the Canny PSDs are shown.

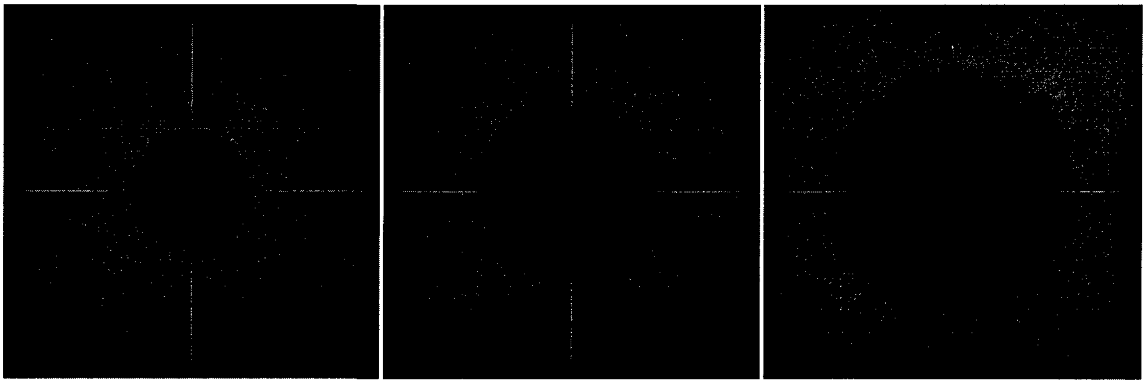

FIG. 29: Canny PSDs: (1st) blurred by $\rho_c = 0.3$; (2nd) blurred by $\rho_c = 0.5$; and (3rd) blurred by $\rho_c = 0.7$.

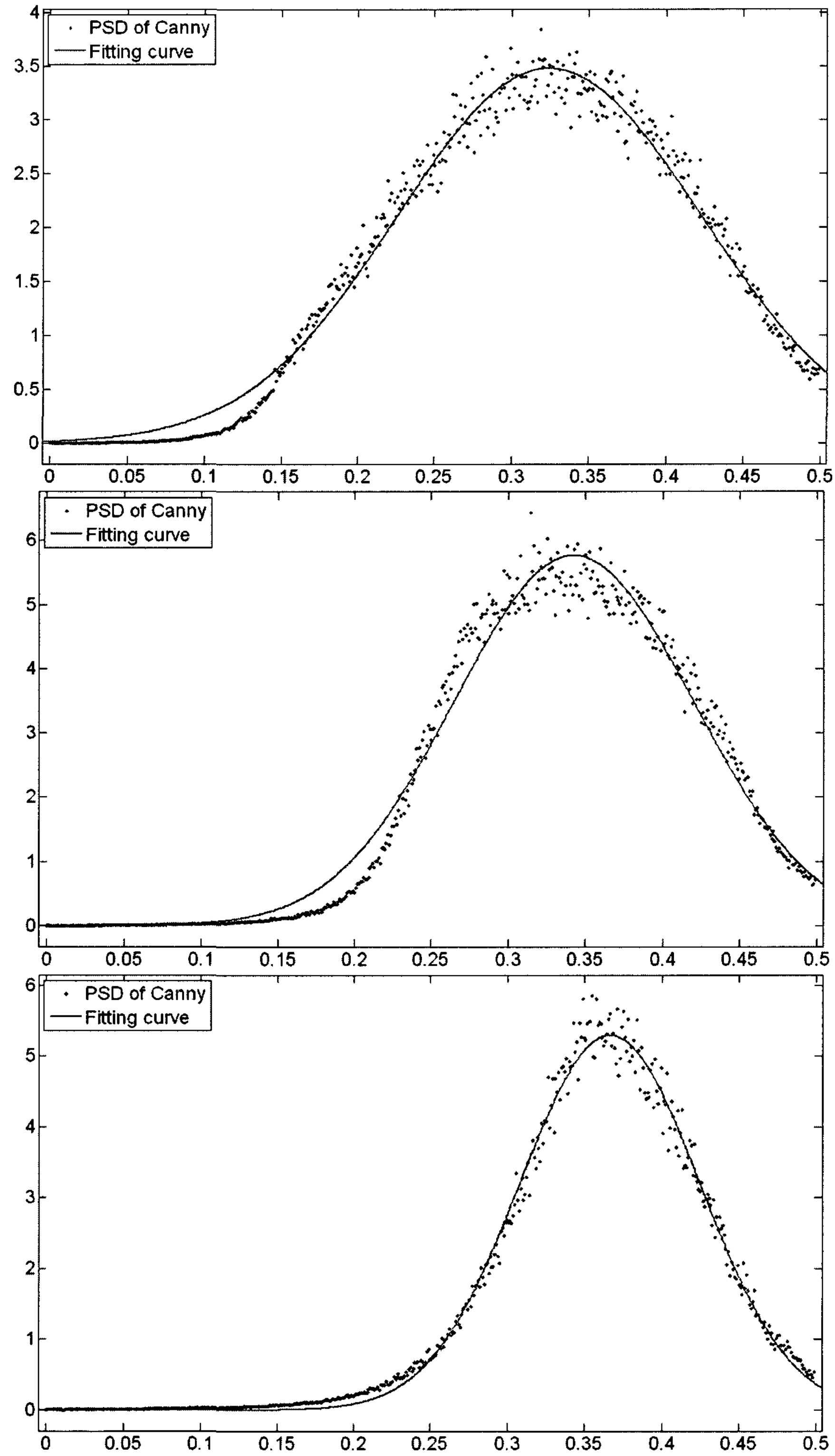

FIG. 30: Results of curve fitting: (1st) blurred by $\rho_c = 0.3$; (2nd) blurred by $\rho_c = 0.5$; and (3rd) blurred by $\rho_c = 0.7$.

Table 3: Canny PSD Parameters.

Canny PSD		ζ=1	ζ=2	ζ=3	ζ=4	ζ=5	ζ=6	ζ=7	ζ=8	ζ=9	ζ=10
ρ_c **= 0.1**	$a=$	0.70	0.82	0.88	0.96	0.96	0.95	0.96	0.94	0.89	0.83
	$b=$	0.25	0.26	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
	$c=$	0.21	0.20	0.19	0.19	0.18	0.18	0.18	0.18	0.18	0.18
ρ_c **= 0.2**	$a=$	1.8	2.32	2.5	2.8	2.90	2.74	2.65	2.65	2.31	2.2
	$b=$	0.30	0.31	0.31	0.32	0.31	0.31	0.31	0.31	0.31	0.31
	$c=$	0.16	0.15	0.14	0.14	0.14	0.14	0.13	0.13	0.14	0.13
ρ_c **= 0.3**	$a=$	3.47	4.38	4.63	4.81	4.60	4.46	4.27	4.18	3.62	3.66
	$b=$	0.32	0.32	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	$c=$	0.13	0.13	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
ρ_c **= 0.4**	$a=$	4.96	5.87	5.88	5.71	5.54	5.21	4.76	4.75	4.36	4.07
	$b=$	0.33	0.32	0.32	0.31	0.31	0.31	0.31	0.31	0.30	0.31
	$c=$	0.12	0.11	0.11	0.11	0.11	0.10	0.10	0.10	0.10	0.10
ρ_c **= 0.5**	$a=$	5.76	6.37	6.20	6.02	5.67	5.33	5.0	5.08	4.43	4.23
	$b=$	0.34	0.33	0.32	0.32	0.32	0.32	0.32	0.31	0.31	0.31
	$c=$	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
ρ_c **= 0.6**	$a=$	5.92	6.16	6.10	5.8	5.77	5.28	4.94	5.06	4.48	4.20
	$b=$	0.35	0.34	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	$c=$	0.094	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
ρ_c **= 0.7**	$a=$	5.29	5.65	5.44	5.33	5.34	4.87	4.44	4.48	4.11	3.91
	$b=$	0.36	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
	$c=$	0.08	0.07	0.07	0.07	0.08	0.07	0.08	0.08	0.08	0.08
ρ_c **= 0.8**	$a=$	3.96	4.90	5.29	5.61	5.65	5.39	5.23	5.26	4.98	4.58
	$b=$	0.37	0.36	0.36	0.35	0.35	0.35	0.35	0.35	0.35	0.35
	$c=$	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
ρ_c **= 0.9**	$a=$	2.27	2.32	2.40	2.25	2.34	1.96	1.97	1.97	1.76	1.64
	$b=$	0.38	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.36	0.37
	$c=$	0.09	0.08	0.08	0.08	0.08	0.08	0.09	0.09	0.09	0.09
ρ_c **= 1.0**	$a=$	1.44	1.48	1.52	1.43	1.49	1.26	1.29	1.27	1.19	1.09
	$b=$	0.4	0.38	0.38	0.37	0.37	0.37	0.37	0.37	0.37	0.37
	$c=$	0.11	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09

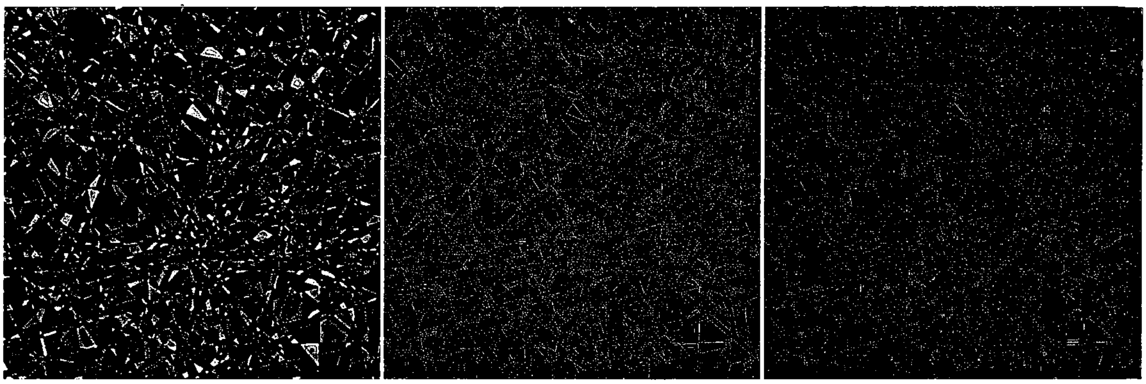

FIG. 31: Validation of Canny PSDs: (1st) blurred by $\rho_c = 0.3$; (2nd) blurred by $\rho_c = 0.5$; and (3rd) blurred by $\rho_c = 0.7$.

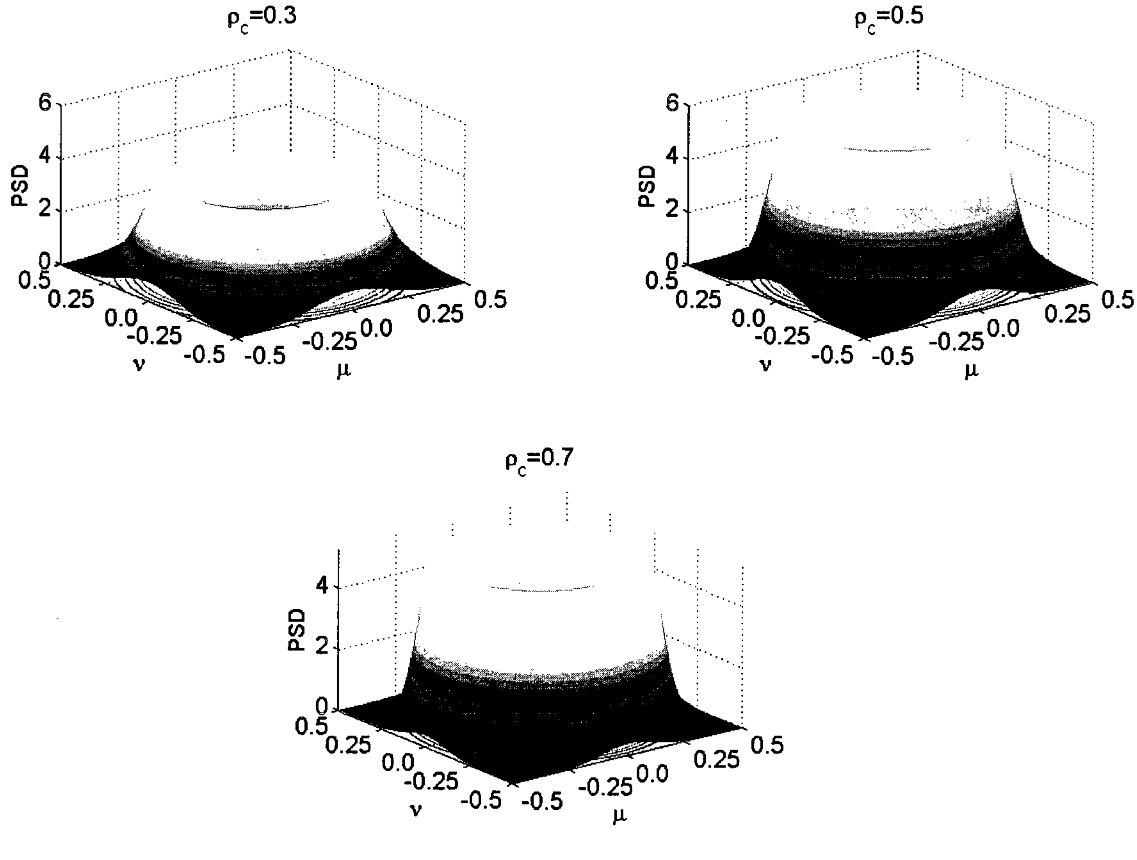

FIG. 32: Theoretic Canny PSDs.

To validate the final form of the Canny PSDs from the Gaussian models, the PSDs were used to perform edge detection. Some results are shown in Figure 31. Though those edge outputs are not very accurate, they are close to the edge images in Figure 28. Therefore, we would use the approximate theoretic Canny PSDs (Figure 32) in the information-theoretic assessment of the performance of the Canny operator.

V.2 INFORMATION ANALYSIS

Since all the PSDs for the edge detectors listed in Section III.2 have been obtained, they can be used in Eq. 37 to determine empirically the impact of the edge detection algorithm on the mutual information between the input scene and the edge output.

V.2.1 Kernel-based Algorithms

First, the comparisons between the traditional first derivative edge detectors and the kernel defined LoG operator are shown. Figure 33 shows the information rate $\mathcal{H}_e$ for the four edge-operators as a function of the system design parameters ρ_c and the overall SNR. We used $\zeta = 1$ in Eq. 36 to model the finest mean spatial detail with respect to the inter-sample distance. We examine the cases for SNR values varying from 1—where it is very difficult to discern between signal and noise—to 256, where the signal is essentially noise free. The blurring parameter ρ_c on the x-axis controls the amount of aliasing noise that is introduced in the image. We also assume that the A/D converter uses 8-bits per pixel to convert the signal. Thus, $\kappa = 256$ and $\tilde{\Phi}_{N_{a/d}}$ is insignificant when compared to the other noise sources. Hence, the two primary noise PSDs that we examine are $\tilde{\Phi}_a$ and $\tilde{\Phi}_{N_e}$.

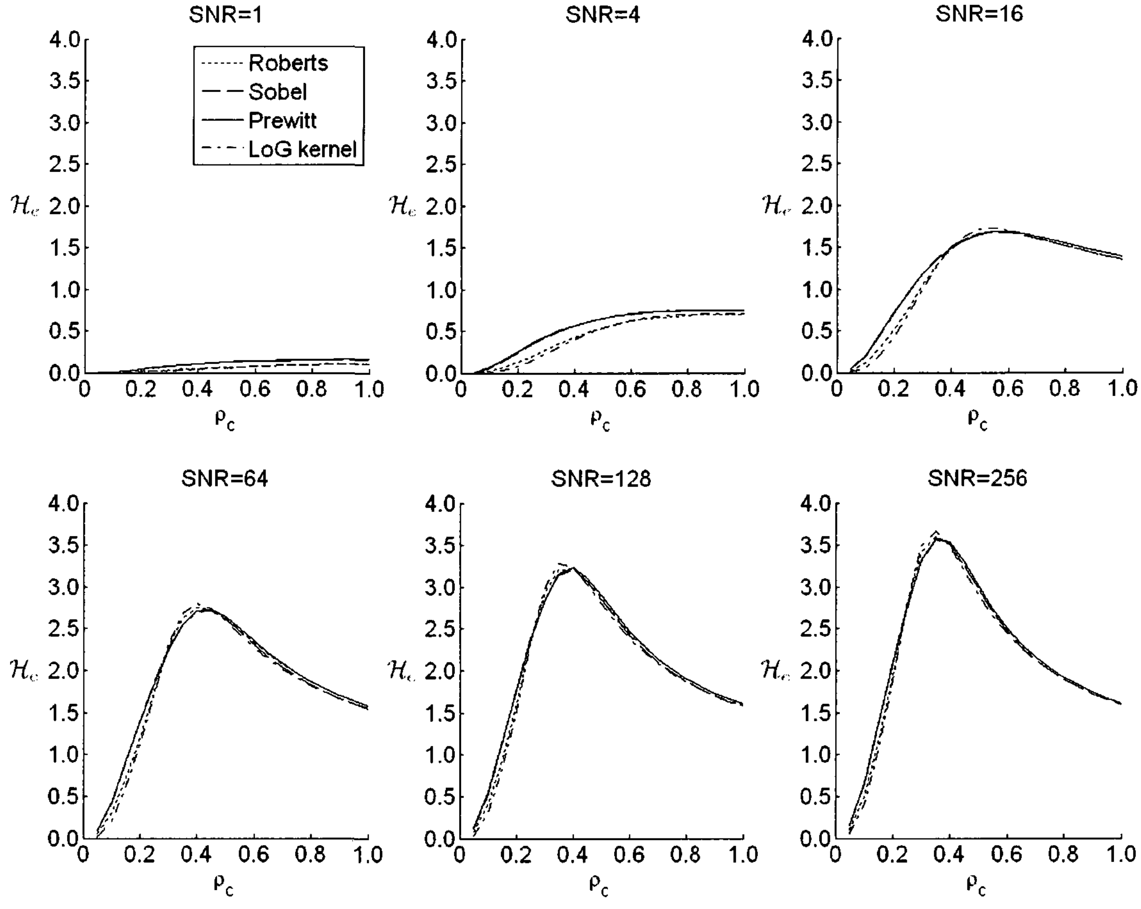

FIG. 33: Information rate $\mathcal{H}_e$ for the kernel-based operators as a function of the optical-response index ρ_c for several SNRs and $\zeta = 1$.

The general observation that we can make is that the performance of the different edge operators is quite similar overall, at least as far as the information is concerned. Looking at the different cases plotted in Figure 33, Prewitt offers the best information performance for the low SNRs. This can probably be attributed to the fact that the Prewitt operator has the weakest edge detection performance of the four operators that we examined. This implies that for the low SNR cases, where the noise and signal are difficult to discriminate, the operator that detects the fewest edges is likely to provide the greatest overall information.

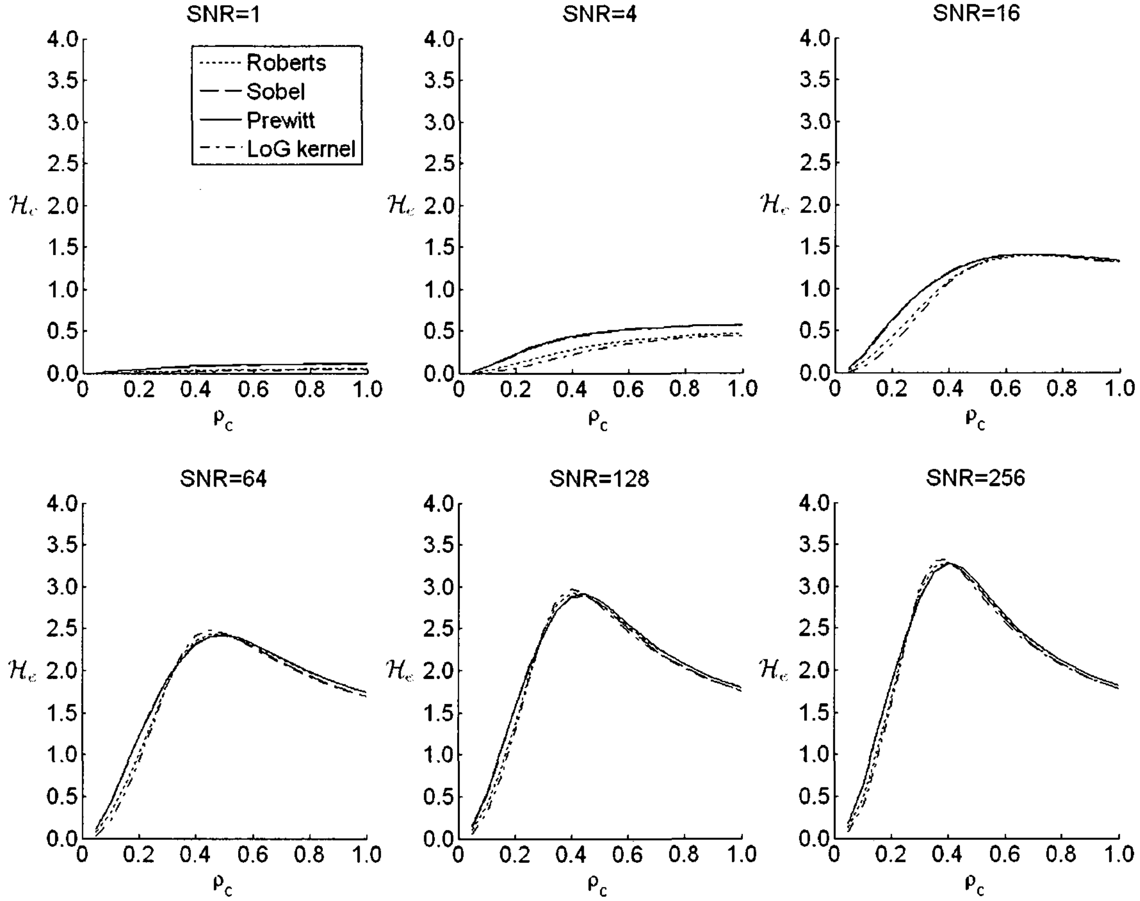

FIG. 34: Information rate $\mathcal{H}_e$ for the kernel-based operators as functions of the optical-response index ρ_c for several SNRs and $\zeta = 4$.

As the SNR increases to 16, $\mathcal{H}_e$ is maximized at about $\rho_c = 0.5$. At this point the maximum response is provided by the LoG operator. There is very little difference between the other three operators though a closer look shows that the performance of the Prewitt operator is better than that of the Sobel operator. This trend is repeated as the SNR increases from 16 to 256.

An interesting aspect of the information rate, $\mathcal{H}_e$, is its dependence on the aliasing noise for a fixed SNR. As can be seen in Figure 33, the value of ρ_c at which $\mathcal{H}_e$ reaches

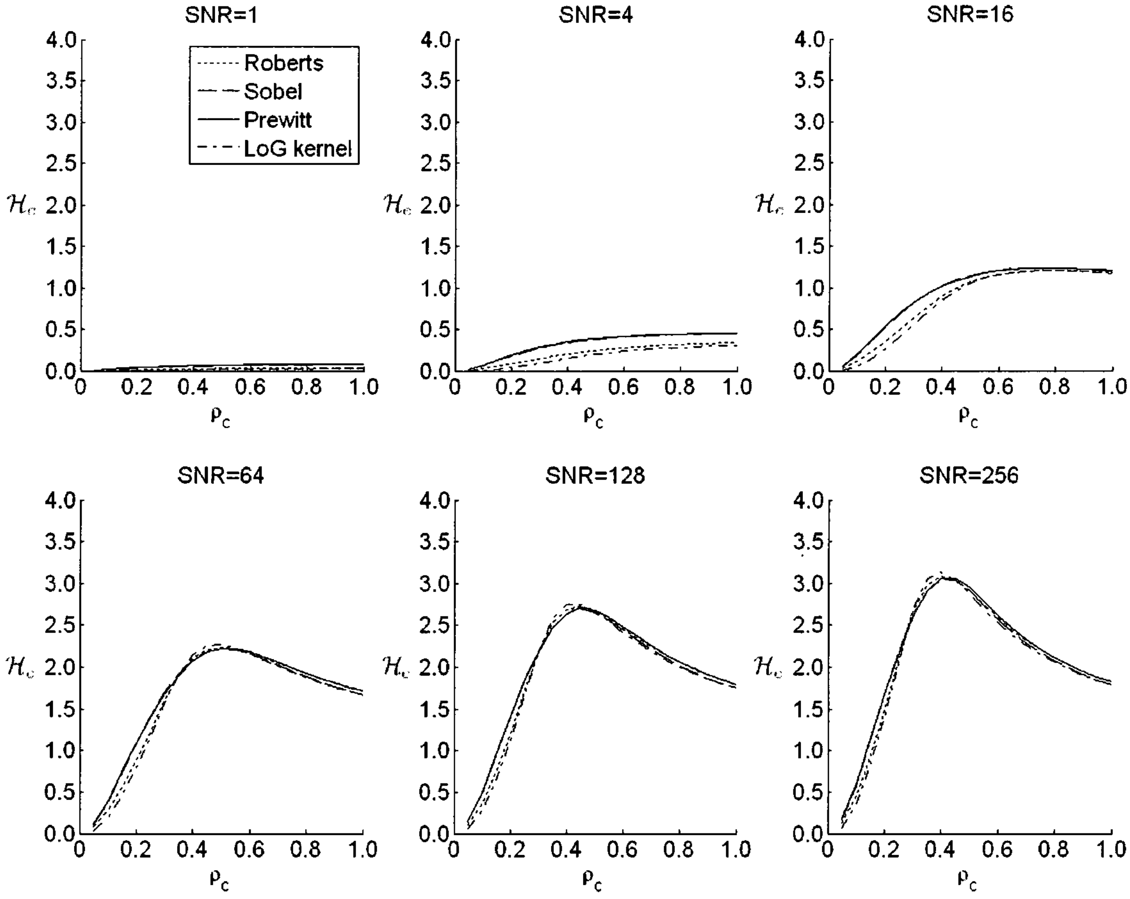

FIG. 35: Information rate $\mathcal{H}_e$ for the kernel-based operators as functions of the optical-response index ρ_c for several SNRs and $\zeta = 7$.

maximum decreases as SNR increases. This is readily explainable when one considers the imaging system model. As the SNR increases, there is less noise in the final image; hence, the impact of aliasing is more easily discernible. When the SNR is low, additional noise due to aliasing does not significantly impact the already poor image quality, so one can continue to decrease blurring without impacting the information. This provides, in essence, a guideline for designing image processing systems and selecting the edge-detection algorithm. For systems with high SNR we can blur the signal more to get rid of the noise

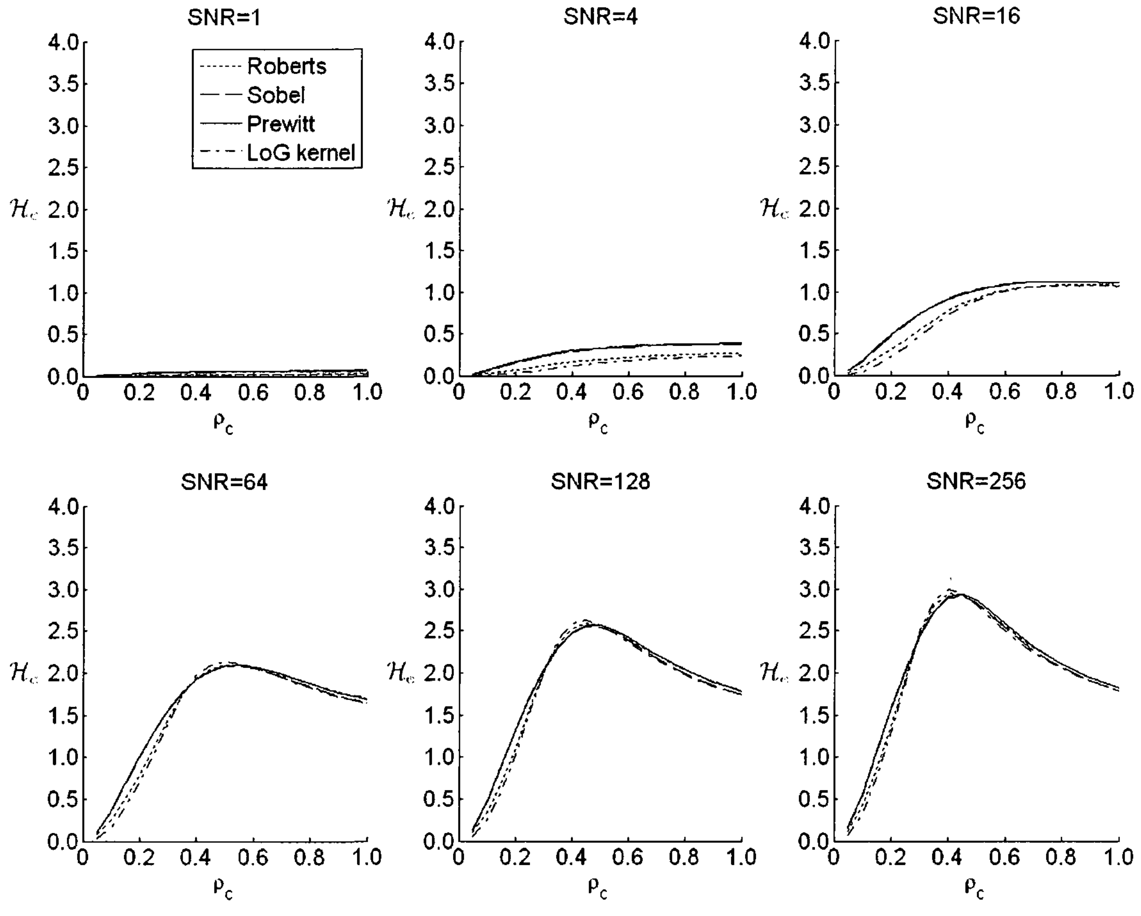

FIG. 36: Information rate $\mathcal{H}_e$ for the kernel-based operators as functions of the optical-response index ρ_c for several SNRs and $\zeta = 10$.

because the restoration process is not hampered by noise. Additionally, since the information is maximized by the LoG operator, that is the edge-detection operator of choice for such systems. As the SNR decreases, more noise is introduced into the image so additional aliasing noise does not have much of an impact on the image quality and less blurring gives sharper results, so information is maximized at higher values of ρ_c. This trend continues for the lowest SNR case where one can essentially let the signal through unblurred without impacting the overall image quality or information throughput.

Figures 33, 34, 35, and 36 show the information capacity for low, medium, and high SNR as a function of ρ_c for four different mean spatial details. It is interesting to note that information is maximized at roughly the same ρ_c for a given SNR, regardless of the mean spatial detail. However, there is some variation. The SFR, in turn, is controlled by the optical-response index ρ_c. The curve shows that the optical-response index ρ_c that maximizes $\mathcal{H}_e$ is independent of ζ but that the losses in $\mathcal{H}_e$ with increasing ζ are critically dependent on ρ_c.

Maximizing the information of the system leads to better edge detection because the ratio between signal and noise achieves its maximum value when $\mathcal{H}_e$ reaches a maximum. Thus, the selection of the ρ_c for which information is maximum for a particular SNR and a specific edge detector gives us a system design criterion in terms of the SFR of the image-gathering device and the SNR: the former controls the amount of aliasing, and the latter affects the system noise.

V.2.2 LoG Operator

Since the derived PSD for the first derivative methods is the same as that computed using simulations, we do not compare the two techniques. Additionally, the first derivative methods that we consider—Sobel, Prewitt and Roberts—are non-parametric, so the form of the PSD is fixed. However, the SFR of the second-derivative based LoG method is controlled by the parameter σ_e. We further analyzed this SFR for a different value of σ_e. There are some differences in the PSD of the LoG as a function of σ_e, though the basic shape is generally the same.

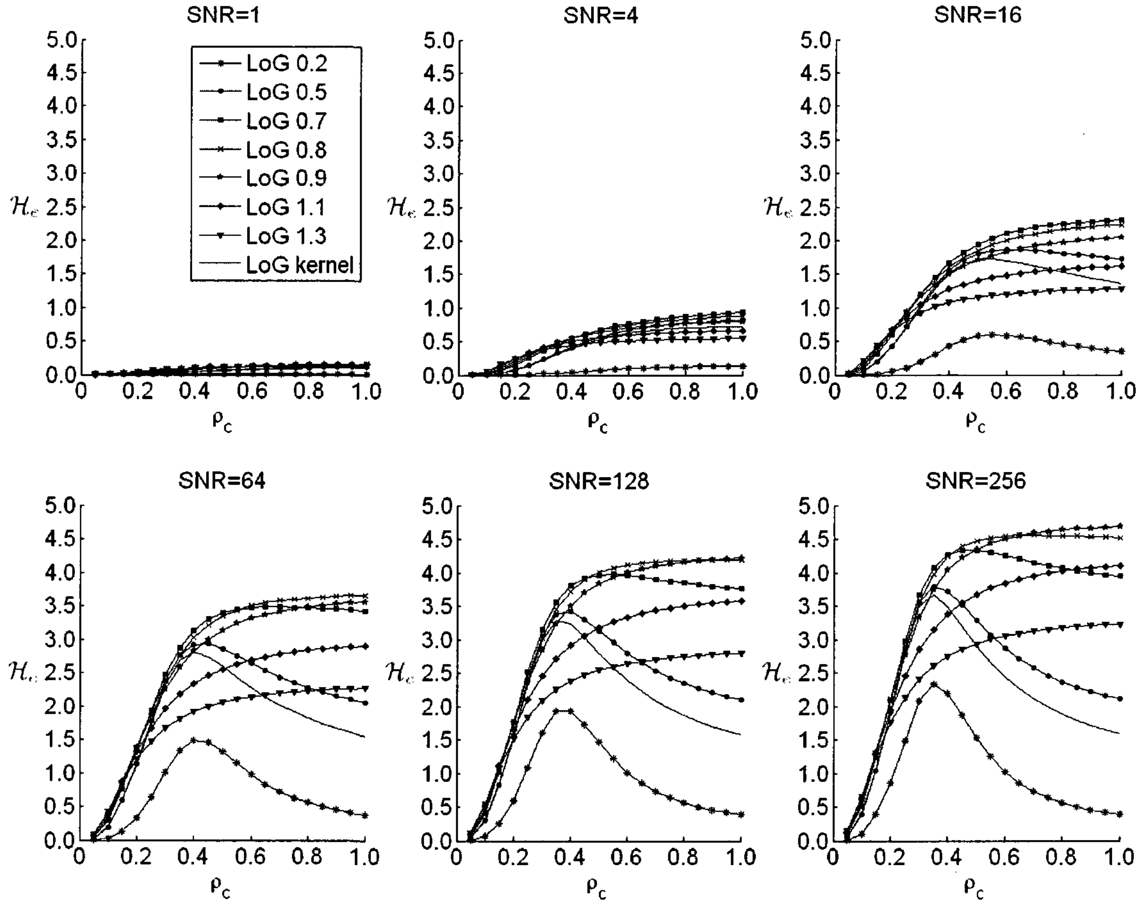

FIG. 37: Information rate $\mathcal{H}_e$ for the LoG operator with different σ_e as functions of the optical-response index ρ_c for several SNRs and $\zeta = 1$.

Figure 37 characterizes the information rate $\mathcal{H}_e$ of the LoG operator for σ_e. To make the explicit comparison for $\mathcal{H}_e$, the LoG kernel has also been included. Since the LoG kernel and LoG with $\sigma_e = 0.5$ have almost the same PSD as shown in Section V.1.3, the curves representing their information should also be similar. In the actual graphs (Figure 37), we see that while they are similar for different SNR and ρ_c, there is a slight difference in amplitude. As ρ_c increases, $\mathcal{H}_e$ of the LoG with $\sigma_e = 0.5$ is slightly larger than that of the LoG kernel.

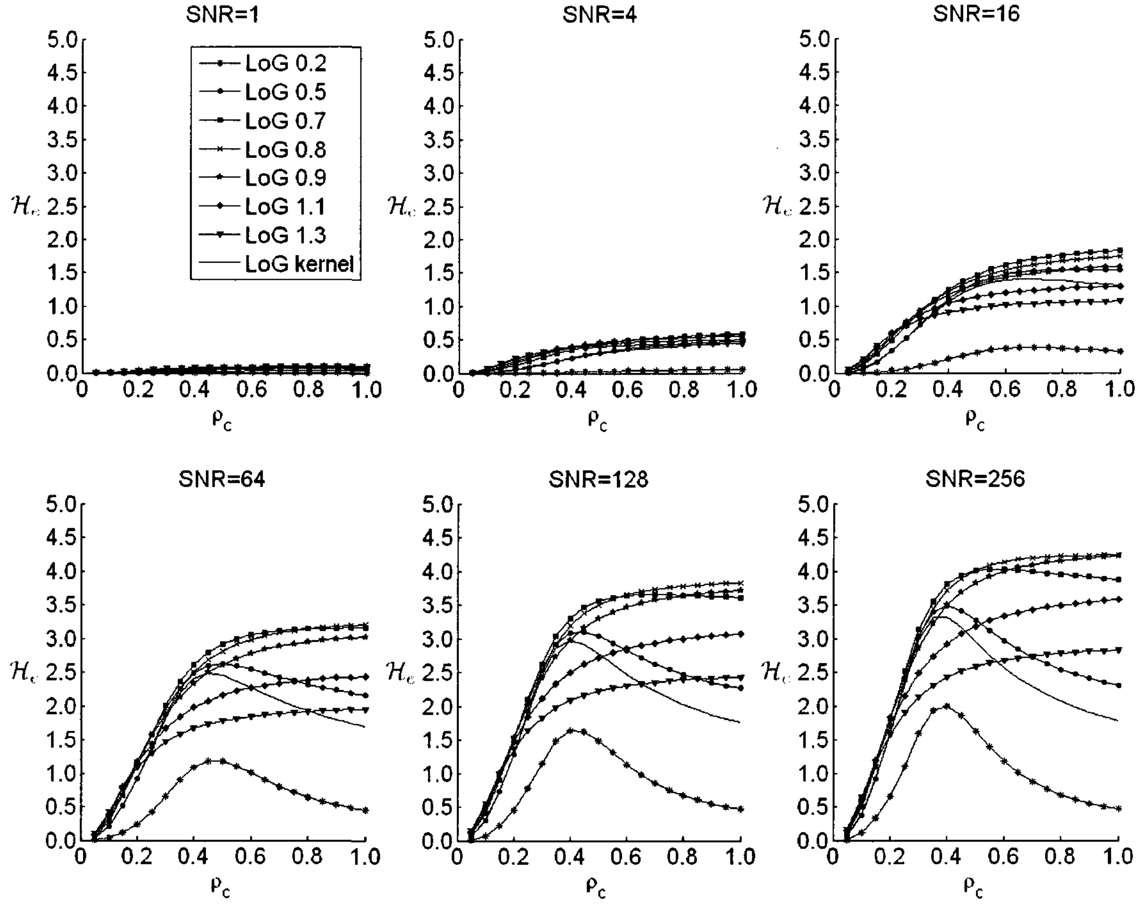

FIG. 38: Information rate $\mathcal{H}_e$ for the LoG operator with different σ_e as functions of the optical-response index ρ_c for several SNRs and $\zeta = 4$.

When σ_e is larger than 0.8, the information capacity becomes (almost) constant for $\rho_c > 0.5$. The reason for this is that the LoG with the larger σ_e blurs the edge image, thus eliminating the high aliasing noise caused by high ρ_c. Therefore, the PSD of the signal and the aliasing noise decrease simultaneously, but the PSD of aliasing noise decreases faster than the PSD of the signal. Thus, $\mathcal{H}_e$ for the LoG operator behaves differently for different values of σ_e. For $\sigma_e < 0.5$, we see the "expected" behavior with $\mathcal{H}_e$ reaching a maximum for some ρ_c and decreasing afterwards. For $\sigma_e > 0.5$, we see $\mathcal{H}_e$ achieve a maximum value

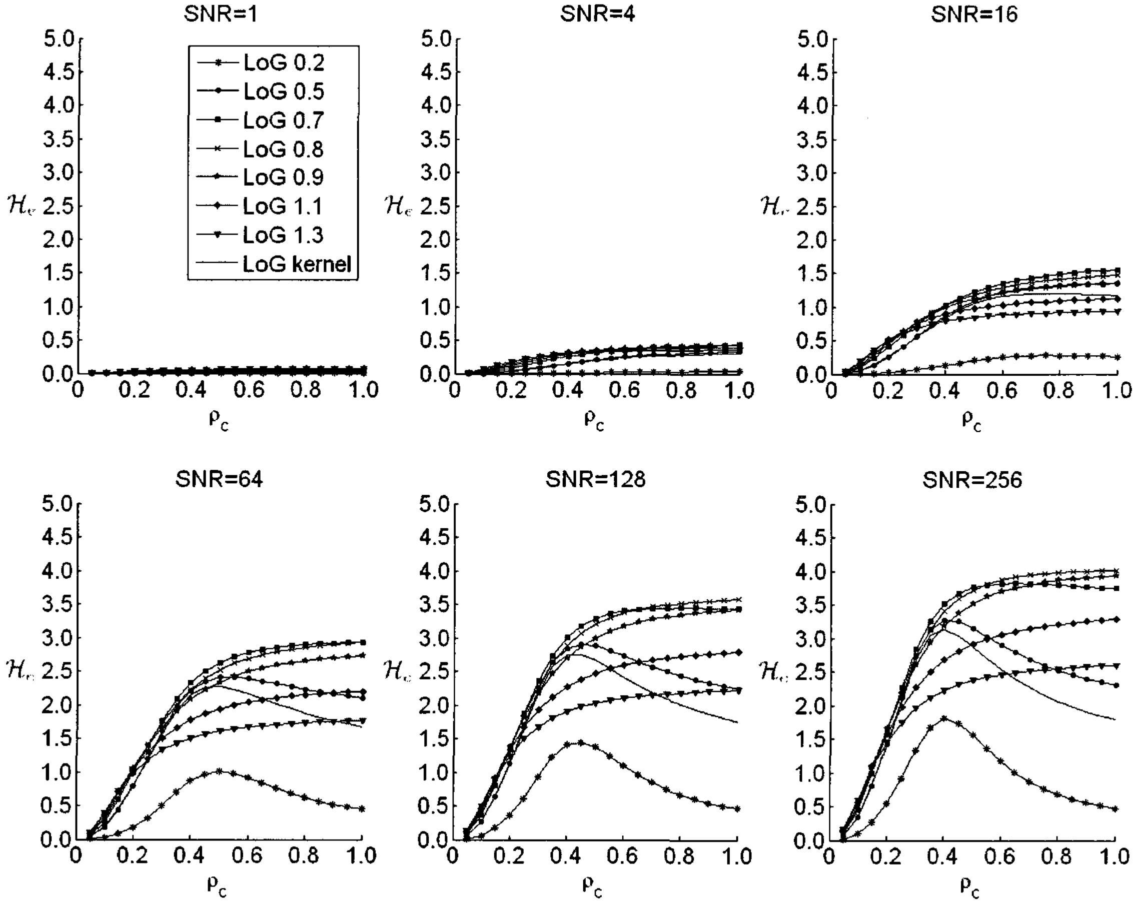

FIG. 39: Information rate $\mathcal{H}_e$ for the LoG operator with different σ_e as functions of the optical-response index ρ_c for several SNRs and $\zeta = 7$.

and essentially remain there. Consequently, based on Figure 37, the LoG with $\sigma_e = 0.7$, 0.8 or 0.9 seems to be the best option, especially when ρ_c is large.

In Figures 38, 39, and 40, the information for low, medium, and high SNR as a function of ρ_c for other mean spatial details is displayed. The information of LoG is also maximized at roughly the same ρ_c for a given SNR regardless of the mean spatial detail, but some variation exists. The curves show that the optical-response index ρ_c that maximizes $\mathcal{H}_e$ is independent of ζ and that the losses in $\mathcal{H}_e$ with increasing ζ are critically dependent on ρ_c.

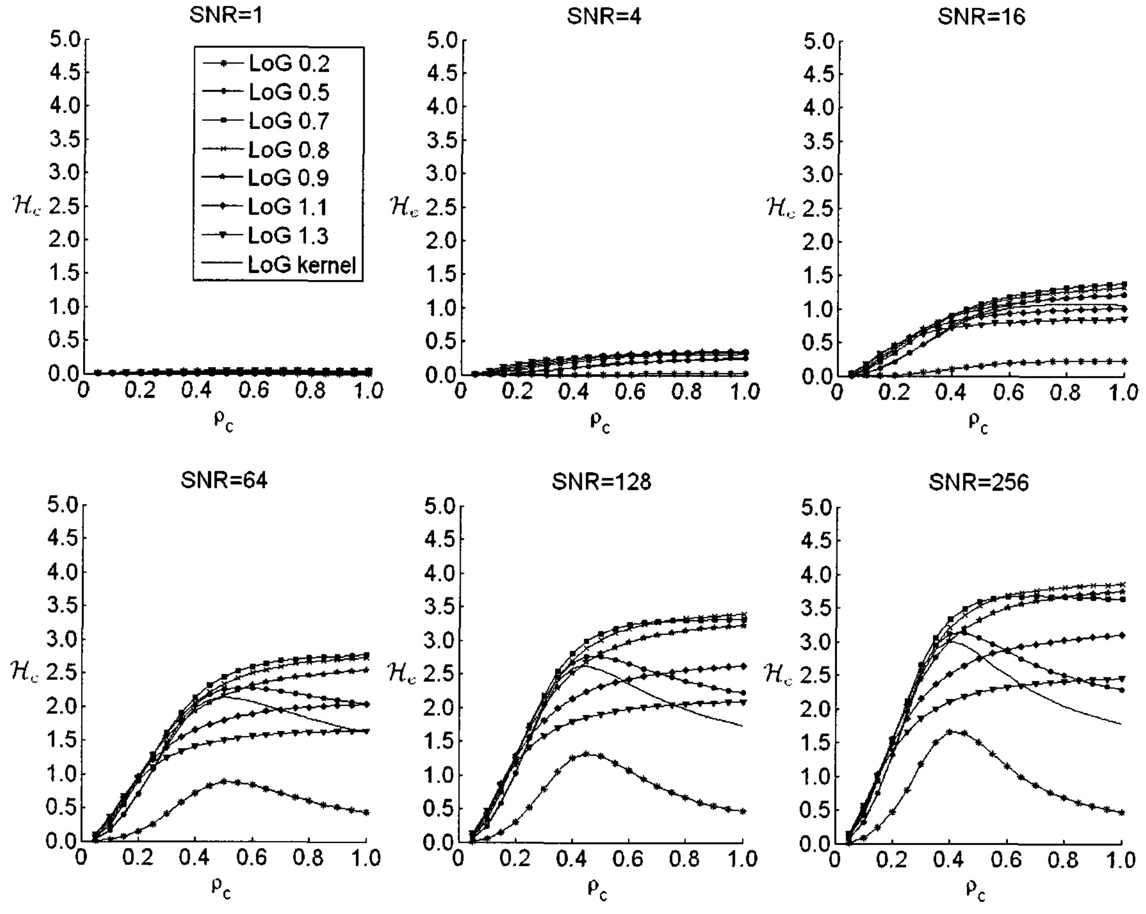

FIG. 40: Information rate $\mathcal{H}_e$ for the LoG operator with different σ_e as functions of the optical-response index ρ_c for several SNRs and $\zeta = 10$.

V.2.3 Canny Operator

As stated in Section V.1, the PSD of the Canny operator for different system parameters can be approximated using the simulation method. In this way, the information of the Canny operator can be examined and assessed. Figure 41 shows the information rate $\mathcal{H}_e$ of the Canny operator for the noise-free case as a function of ρ_c for four different mean spatial details. Since we only considered the noise-free condition, the results and comparisons are shown only for the different mean spatial details ζ. Furthermore, to compare with other

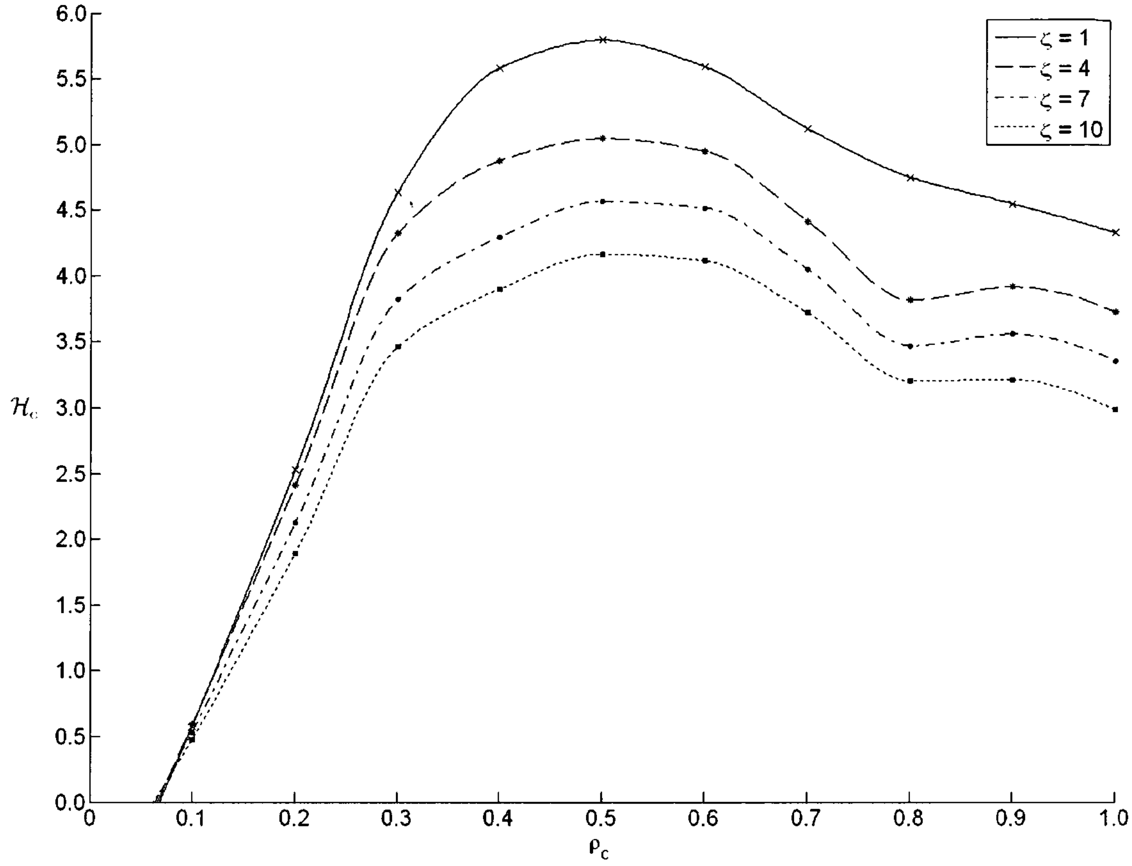

FIG. 41: Information rate $\mathcal{H}_e$ of the Canny operator as functions of the optical-response index ρ_c for several mean spatial details.

methods, we used SNR=256, which is close to the noise-free condition. For different ζ, $\mathcal{H}_e$ is always maximized at about $\rho_c = 0.5$. Also, as expected, the information rate decreases as mean spatial detail increases at the same point ρ_c, because fewer edges lead to a smaller signal component.

Comparing with other results in Section V.2.1, the information is obviously largest for the Canny operator. For example, when $\zeta = 1$, the maximum value $\mathcal{H}_e = 5.8$ for Canny, but $\mathcal{H}_e \leq 4.0$ for those derivative kernel-based methods. Even for the parametric form of the LoG operator, $\mathcal{H}_e < 5.0$ even with the optimal value of σ_e. Based on the information-theoretic analysis, the Canny operator is better than the traditional edge detectors defined in Section III.2. This is also borne out by experiments [5].

There is a slight anomaly in Figure 41: there is a dip in the curve for $\mathcal{H}_e$ around $\rho_c = 0.7$, when $\zeta \geq 4$. This anomalous behavior of the $\mathcal{H}_e$ curve may be attributed to the fact that we are modeling an essentially non-linear, shift-variant process using a linear, shift-invariant process.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

This chapter summarizes the main contributions and conclusions in Section VI.1 and presents possible future research directions in Section VI.2.

VI.1 CONCLUSIONS

We have examined three main topics in this dissertation. In the sections that follow we state the contributions that we made to each of these areas and the conclusions that we drew from our research.

VI.1.1 Edge Detection

We have proposed new algorithms based on multi-scale processing and higher order derivatives of image intensity to detect and localize an edge. A group of Gaussian filters was used to generate multi-scale representations of the original image. By recognizing that edges due to the signal have regional connectivity and specific orientation across the scales, and that edges due to noise do not, we can differentiate between the edge-features due to signal and those due to noise. Using these characteristics, pixels corresponding to edges can be preserved, while those due to noise can be eliminated. To further extend our proposed edge detection algorithm, the local noise estimate is introduced to help determine local dynamic threshold. It helps to reduce the effect of an ill-posed global threshold and makes the edge more complete.

The other key idea that emerges from this research is that of the efficacy of connectivity analysis in further reducing the impact of noise. This idea arose from the recognition that the structure of edge patterns remains substantially unchanged even in the presence of a high degree of noise.

The proposed edge detection algorithm has been applied to a variety of images ranging from ideal, noise free images to images with very low SNR and from computer generated images to natural images. In all cases, the algorithm produces good results, extracting edge structure from extremely noisy—low SNR—images quite successfully.

VI.1.2 Noise Reduction

Instead of using global noise-reduction mechanisms such as a smoothing filter or a median filter, we made use of edge pattern analysis both for noise assessment and as a foundation for noise reduction methods. Starting with the premises that edges in the image should be preserved, and that the overall impact of noise is to reduce resolution, our new edge detection algorithms that are based on multi-scale processing and connectivity analysis are applied to reduce noise by robustly extracting authentic edges.

The significance of this method is to leave the resolution of all connected edges intact while reducing resolution in an area where noise is detected. Since the resolution of an image is reduced by the overall impact of noise, reducing resolution in noisy areas by blurring eliminates the appearance of noise while not affecting overall image quality. Experimental results and the image comparison metric shows that this new method in noise reduction is effective even for very low SNR values.

VI.1.3 Edge Detection Evaluation

We theoretically analyzed edge detection algorithms within the context of an end-to-end visual communication channel. The impact of the different parts, such as image gathering, sampling, and A/D conversion that comprise the image acquisition system are analyzed and assessed in an integrated manner using information theory. By setting initial conditions, such as image-gathering response, sampling, additive noise, etc., the edge detection methods can be theoretically quantified and compared. Edge detection is regarded to be high performing only if the information rate from the scene to the edge approaches the maximum possible.

One key idea that emerged from this dissertation is that of the PSD of commonly used edge detectors. Since the PSD is required for computing information capacity, we first derived the PSDs for kernel-based methods, such as Sobel, Prewitt and Roberts. We also examined the impact of the σ_e parameter on the performance of the LoG function and determined the optimal solution for specific visual communication conditions.

Non-linear operators like the Canny operator do not conform to the PSD analysis that we used earlier. Instead, we developed a family of PSDs for the Canny operator based on system parameters such as the mean spatial details of scene and the parameters of image acquisition process. The "PSD" was then approximated experimentally using an average of 100 instances of the application of Eq. 37. Results demonstrate that the Canny operator is better than traditional edge detectors in terms of its information capacity.

In assessing edge detectors, people generally subjectively judge and compare the detectors. Objective measures are often based on simulations and not necessarily on theoretic analysis or consideration of the visual system. The information-theoretic analysis presented in this dissertation becomes a new tool for analyzing and comparing different edge detection operators, especially in the visual communication environment.

VI.2 FUTURE WORK

There are several directions to further extend this research, which are briefly discussed below:

1. **Noise Estimation:** To solve the problem posed by static global thresholding, a local noise estimate was proposed. The initial tests show that this strategy is promising, but it is simple and primitive. This research can be continued to find more robust solutions.

2. **Connectivity Analysis:** In this dissertation, we have presented some early results for edge detection and noise reduction using connectivity analysis that are quite promising. However, there are still significant issues to tackle. For example, in very noisy conditions, some structures can be easily discerned by a human, but it is hard to consider them as features because their edges are not connected. The connectivity analysis is currently rather primitive and needs to be made more robust and "intelligent."

3. **Canny Operator Evaluation:** In the PSD derivation, we simplified the Canny operator by assuming a noise-free image and computing the PSD as an "inverse" operation. Additionally, we did not use hysteresis thresholding because we wanted a

grayscale image. Examining the whole operation in the mean-square-error environment should provide a more robust framework.

BIBLIOGRAPHY

[1] L. G. Roberts, "Machine perception of three-dimensional solids," *Optical and Electro-optical Information Processing, MIT Press, Cambridge, MA*, pp. 159–197, 1965.

[2] I. Sobel, "Camera models and machine perception," Ph.D. dissertation, Stanford Artificial Intelligence Lab, Stanford University, May 1970.

[3] J. M. S. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics, LipKin, B. S., and Rosenfeld, A. (eds.), Academic Press, New York*, pp. 75–149, 1970.

[4] D. H. Marimont and Y. Rubner, "A probabilistic framework for edge detection and scale selection," in *Computer Vision, 1998. Sixth International Conference on*, 1998, pp. 207–214.

[5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[6] R. M. Haralick, "Digital step edges from zero crossing of second directional derivatives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-6, no. 1, pp. 58–68, Jan. 1984.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Pearson Prentice Hall, Upper Saddle River, New Jersey 07458, third ed., 2008.

[8] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[9] J. H. Elder and S. W. Zucker, "Local scale control for edge detection and blur estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 699–716, 1998.

[10] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[11] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers*, vol. 100, no. 20, pp. 562–569, 1971.

[12] A. Witkin, "Scale-space filtering," in *Proc. 8lh Int. Joint Conf. Ari. Inlell.', Karlsruhe, West Germany*, 1983, pp. 1019–1022.

[13] J. J. Koenderink, "The structure of images," *Biological cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.

[14] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 7, pp. 710–732, Jul 1992.

[15] J. R. Beltran, F. Beltran, and A. Estopanan, "Multiresolution edge detection and classification: noisecharacterization," in *1998 IEEE International Conference on Systems, Man, and Cybernetics, 1998*, vol. 5, 1998, pp. 4476–4481.

[16] W. F. Schreiber and R. A. Haddad, "Fundamentals of electronic imaging systems," *Applied Optics*, vol. 29, p. 17, 1990.

[17] V. Aurich, G. Winkler, K. Hahn, A. Martin, , and K. Rodenacker, "Noise reduction in images: some recent edge-preserving methods," *Pattern Recognition and Image Analysis*, vol. 9, no. 4, pp. 749–766, 1999.

[18] J. F. Abramatic and L. M. Silverman, "Nonlinear restoration of noisy images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 3, pp. 141–149, 1982.

[19] R. Chellappa and R. Kashyap, "Digital image restoration using spatial interaction models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, no. 3, pp. 461–472, 1982.

[20] C. W. Helstrom, "Image restoration by the method of least squares," *J. Opt. Soc. Am*, vol. 57, pp. 297–303, 1967.

[21] C. L. Fales, F. O. Huck, and R. W. Samms, "Imaging system design for improved information capacity," *Applied optics*, vol. 23, no. 6, pp. 872–888, 1984.

[22] F. O. Huck, C. L. Fales, J. A. McCormick, and S. K. Park, "Image-gathering system design for information and fidelity," *Journal of the Optical Society of America A*, vol. 5, no. 3, pp. 285–299, 1988.

[23] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Transactions on image processing*, vol. 4, no. 4, pp. 499–502, 1995.

[24] R. H. Chan, C. W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization,” *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005.

[25] B. Jiang and W. Huang, “Adaptive threshold median filter for multiple-impulse noise,” *Journal of Electronic Science and Technology of China*, vol. 5, no. 1, pp. 70–74, 2007.

[26] N. Lian, V. Zagorodnov, , and Y. Tan, “Edge-preserving image denoising via optimal color space projection,” *IEEE Transactions on Image Processing*, vol. 15, no. 9, p. 2575, 2006.

[27] B. Smolka, K. N. Plataniotis, A. Chydzinski, M. Szczepanski, A. N. Venetsanopoulos, and K. Wojciechowski, “Self-adaptive algorithm of impulsive noise reduction in color images,” *Pattern Recognition*, vol. 35, no. 8, pp. 1771–1784, 2002.

[28] A. B. Hamza and H. Krim, “Image denoising: A nonlinear robust statistical approach,” *IEEE Transactions on Signal Processing*, vol. 49, no. 12, p. 3045, 2001.

[29] H. G. Adelmann, “An edge-sensitive noise reduction algorithm for image processing,” *Computers in biology and medicine*, vol. 29, no. 2, pp. 137–145, 1999.

[30] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[31] R. Lu, Y. Shen, and Y. Wang, “Novel Anisotropic Diffusion Algorithm Based on PID Control Law Together with Stopping Mechanism,” in *Engineering in Medicine and*

Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the, 2005, pp. 3402–3405.

[32] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[33] C. Q. Zhan and L. J. Karam, "Wavelet-based adaptive image denoising with edge preservation," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1, 2003.

[34] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. E. Kerre, W. Philips, and I. Lemahieu, "Noise reduction by fuzzy image filtering," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 429–436, 2003.

[35] R. A. Peters, "A new algorithm for image noise reduction using mathematical morphology," *IEEE Transactions on Image Processing*, vol. 4, no. 5, 1995.

[36] D. Walters, "Selection of image primitives for general-purpose visual processing," *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 2, pp. 261–298, 1987.

[37] D. J. Jobson, "Spatial vision processes: From the optical image to the symbolic structures of contour information," NASA, Washington, DC, Tech. Rep. 2838, November 1988, nASA Technical Paper.

[38] D. J. Jobson, Z. Rahman, G. A. Woodell, and G. D. Hines, "Automatic assessment and reduction of noise using edge pattern analysis in non-linear image enhancement," in *Proceedings of SPIE*, vol. 5438, 2004, pp. 141–141.

[39] D. J. Williams and M. Shah, "Edge characterization using normalized edge detector," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 4, pp. 311–318, 1993.

[40] J. R. Beltrán, J. García-Lucía, and J. Navarro, "Edge detection and classification using Mallat's wavelet," in *IEEE International Conference Image Processing, 1994. Proceedings. ICIP-94.*, vol. 1, 1994, pp. 293–297.

[41] G. Palacios and J. R. Beltran, "Cell Nuclei Segmentation Combining Multiresolution Analysis, Clustering Methods and Colour Spaces," in *Machine Vision and Image Processing Conference, 2007. IMVIP 2007. International*, 2007, pp. 91–97.

[42] Z. Rahman and D. J. Jobson, "Noise, edge extraction, and visibility of features," in *Proceedings of SPIE*, vol. 5817, 2005, pp. 200–200.

[43] O. Laligant and F. Truchetet, "A nonlinear derivative scheme applied to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 2, pp. 242–257, Feb. 2010.

[44] L. R. Liang and C. G. Looney, "Competitive fuzzy edge detection," *Applied Soft Computing Journal*, vol. 3, no. 2, pp. 123–137, 2003.

[45] H. Mehrara, M. Zahedinejad, and A. Pourmohammad, "Novel edge detection using BP neural network based on threshold binarization," in *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, vol. 2, Dec. 2009, pp. 408–412.

[46] W. Jiang, K.-M. Lam, and T.-Z. Shen, "Efficient edge detection using simplified gabor wavelets," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 4, pp. 1036–1047, Aug. 2009.

[47] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[48] I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 753–763, May 1979.

[49] E. De Micheli, B. Caprile, P. Ottonello, and V. Torre, "Localization and noise in edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 10, pp. 1106–1117, Oct 1989.

[50] J. R. Fram and E. S. Deutsch, "On the quantitative evaluation of edge detection schemes and their comparison with human performance," *Computers, IEEE Transactions on*, vol. C-24, no. 6, pp. 616–628, June 1975.

[51] L. Kitchen and A. Rosenfeld, "Edge evaluation using local edge coherence," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 11, no. 9, pp. 597–605, Sept. 1981.

[52] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1338–1359, 1997.

[53] Q. Ji and R. M. Haralick, "Efficient facet edge detection and quantitative performance evaluation," *Pattern Recognition*, vol. 35, no. 3, pp. 689–700, 2002.

[54] T. Kanungo, M. Y. Jaisimha, J. Palmer, and R. M. Haralick, "A methodology for quantitative performance evaluation of detection algorithms," *IEEE Transactions on Image Processing*, vol. 4, no. 12, pp. 1667–1674, 1995.

[55] T. Peli and D. Malah, "A study of edge detection algorithms," *Computer Graphics and Image Processing*, vol. 20, no. 1, pp. 1–21, 1982.

[56] D. Gabor, "Theory of communication," *Proc. of IEEE*, vol. 93, pp. 429–457, 1946.

[57] F. O. Huck, C. L. Fales, and Z. Rahman, *Visual communication: an information theory approach.* Kluwer Academic Pub, 1997.

[58] ——, "Information theory of visual communication," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 354, no. 1716, pp. 2193–2248, 1996.

[59] C. L. Fales, F. O. Huck, R. Alter-Gartenberg, and Z. Rahman, "Image gathering and digital restoration," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 354, no. 1716, pp. 2249–2287, 1996.

[60] C. E. Shannon and W. Weaver, "A mathematical theory of communication," *Bell Syst. Tech. J*, vol. 27, pp. 379–423, 1948.

[61] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications.* Technology Press of the Massachusetts Institute of Technology Cambridge, 1949.

[62] W. C. Kao and Y. J. Chen, "Mltistage bilateral noise filtering and edge detection for color image enhancement," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1346–1351, 2005.

[63] F. O. Huck, C. L. Fales, and Z. Rahman, *Visual Communication: An Information Theory Approach.* Kluwer Academic Publishers, Norwell, MA, 1997.

[64] H. Jeong and C. I. Kim, "Adaptive determination of filter scales for edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 5, pp. 579–585, 1992.

[65] T. Lindeberg, *Scale-Space Theory in Computer Vision.* Dordrecht, The Netherlands: Kluwer, 1994.

[66] B. Jiang and Z. Rahman, "Noise reduction using multi-resolution edge detection," in *Proceedings of SPIE*, vol. 7245, 2009, p. 724507.

[67] J. J. Clark, "Authenticating Edges Produced by Zero Crossing Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 43–57, 1989.

[68] F. Bergholm, "Edge focusing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 6, pp. 726–741, 1987.

[69] F. J. Estrada and J. H. Elder, "Multi-scale contour extraction based on natural image statistics," in *Workshop on Perceptual Organization in Computer Vision (POCV)*, vol. 8, 2006.

[70] P. Liang and Y. F. Wang, "Local scale controlled anisotropic diffusion with local noise estimate for image smoothing and edge detection," in *Proceedings of the Sixth International Conference on Computer Vision.* IEEE Computer Society Washington, DC, USA, 1998, p. 193.

[71] S. Olsen, "Estimation of noise in images: An evaluation," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 4, pp. 319–323, 1993.

[72] S.-C. Tai and S.-M. Yang, "A fast method for image noise estimation using laplacian operator and adaptive edge detection," in *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on*, 12-14 2008, pp. 1077 –1081.

[73] D. Shin, R. Park, S. Yang, and J. Jung, "Block-based noise estimation using adaptive Gaussian filtering," in *Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on*, 2005, pp. 263–264.

[74] C. Liu, W. Freeman, R. Szeliski, and S. Kang, "Noise estimation from a single image," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006.

VITA

Bo Jiang
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

EDUCATION

Master of Science in Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China, June 2007.

Bachelor of Science in Information Engineering, Chengdu University of Technology, Chengdu, Sichuan, China, June 2004.

PUBLICATIONS

Journal Articles

1. Bo Jiang, Wei Huang, "Adaptive Threshold Median Filter for Multiple-Impulse Noise," Journal of Electronic Science and Technology of China, pp. 70-74, March 2007.
2. Bo Jiang, Wei Huang, "Speech Recognition Based on Multi-Band CHMM and BPNN Fusion," Application of Electronic Technique, pp. 126-128, February 2007.

Selected Conference Proceedings and Presentations

1. Bo Jiang, Zia-ur Rahman, "Information-theoretic Analysis of Edge Detection in Visual Communication," SPIE Optics and Photonics 2010, Optical Engineering and Applications, Mathematical Methods in Pattern and Image Analysis, August 2010.
2. Bo Jiang, Zia-ur Rahman, "Multi-scale Edge Detection with Local Noise Estimate," SPIE Optics and Photonics 2010, Optical Engineering and Applications, Application of Digital Image Processing, August 2010.
3. Bo Jiang, Zia-ur Rahman, "Noise Reduction Using Multi-resolution Edge Detection," IS&T/SPIE's 21st Annual Symposium, Electronic Imaging 2009, Image Processing: Algorithms and System VII, Proceedings of SPIE vol. 7245, 2009.