An Experimental Investigation in the Use of Robust Statistics in Edge Detection

A Thesis

Submitted to the College of Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Department of Computational Science

University of Saskatchewan

Saskatoon, Saskatchewan

by

Zhanhong Wu

August, 1993

Distribution Notice

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computational Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 0W0

An Experimental Investigation in the Use of Robust Statistics in Edge Detection

Candidate: Zhanhong Wu

Supervisor: Herb Yang

Master of Science Thesis

Presented to the College of Graduate Studies and Research
University of Saskatchewan

August, 1993

ABSTRACT

Edge detection is one of the fundamental algorithms in computer vision and image processing. In edge detection, abrupt intensity changes are detected at boundaries of regions and these changes are characterized into an edge map.

Many edge detection schemes have been studied during the last 30 years. A fundamental conflict exists between the two performance requirements in edge detection: noise immunity and accurate localization. Most edge detection schemes assume that noise has a Gaussian distribution and the least-sum-of-squares (LSS) method is used widely. In the presence of outliers, data which do not conform to any known models of noise, the performance of these detectors degrades significantly. Another issue is the lack of methods for quantitative performance evaluation and comparison algorithms. Qualitative analysis based on human evaluation with no clearly-stated standard is often used as a means to judge the goodness of an edge detector.

This thesis addresses the above two issues. In particular, the application of robust statistics in edge detection and evaluation is explored. Two new robust edge detection schemes based on least-median-of-squares techniques are proposed. Experimentally, the new detectors are found to achieve better noise immunity especially when there are outliers in the data. A novel quantitative evaluation scheme, which can be applied to real images, is also proposed. Furthermore, the new evaluation scheme can also be applied to improve edge maps generated by any given edge detector. The performance of several existing detectors and the proposed robust detectors using synthetic and real images is studied in this thesis. Experimental results show that the robust detectors perform better than others in the presence of outliers.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Herbert H.Y. Yang for his expert guidance throughout this research. You have always been available to discuss various problems with me. Thank you, Herbert, for your constant support and encouragement. I feel very fortunate to have worked with you. The experience I have gained here will certainly benefit me in years to come.

To my thesis committee, Dr. Derek Eager, Dr. Carey Williamson and Dr. Richard Barton, thank you for the time, effort and attention you bestowed on my thesis. It was an honour to have had each of you on my committee. Thank you for your valuable advice and comments.

To the faculty, staff and graduate students of the Department of Computational Science, thank you for making my time here as a graduate student enjoyable. Special thanks go to J. Jennifer Liu, Sanjay Bakshi, Russell Muzzolini, Gamal Hassan Abdel-Hamid for all your help during my work in the vision lab.

The financial support provided by the College of Graduate Studies and the Department of Computational Science is gratefully acknowledged.

To my friends, Yuru Wang, Yi Chen, Jiabi Zhu, Yijun Jiang and others too numerous to list, thank you for your friendship and innumerable assistance. You made my life here colourful!

To my father, Shixian Wu, my mother, Yunqing Wang and my sister, Zhanbing Wu, thank you for your constant support. Your good nature always influences me.

To my husband, Lei Zhu, thank you for your love and your phone calls at 2am which made my work in the lab so enjoyable!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Edge Detection

An edge in an image corresponds to a discontinuity in the intensity surface of the underlying scene. The discontinuity may be due to differences in the depth or changes in the surface normal of the object, changes in the reflectance properties or changes in the illumination [19]. Edge detection, one of the fundamental algorithms in computer vision and image processing, transforms the intensity changes at the boundaries of regions into an edge map. An edge map is essential in computer vision because edges represent important information about the objects in an image. The accuracy in the locations of detected edges will influence the performance of higher level processes such as object recognition. Because of the importance of edge detection, much work has been done in the last 30 years [1, 2, 5, 6, 7, 12, 16, 21, 23, 24, 28, 29, 30].

Intuitively, edges correspond to abrupt intensity changes. Formally, a function $f(t)$ is said to have a discontinuity of order k ($k \geq 0$) at t_0, if the left and right kth-derivatives at t_0 are different, i.e., $f^{(k)}(t_0+) \neq f^{(k)}(t_0-)$, where $f^{(k)}(t_0+)$ denotes $\text{limit}_{\epsilon \to 0}\ f^{(k)}(t_0+\epsilon)$ and $f^{(k)}(t_0-)$ denotes $\text{limit}_{\epsilon \to 0}\ f^{(k)}(t_0-\epsilon)$ with $\epsilon > 0$. The difference $f^{(k)}(t_0+) - f^{(k)}(t_0-)$ is the size of the discontinuity [19]. Thus, an ideal

(a) Step edge (b) Roof edge

Figure 1.1: Step edge and roof edge.

step edge is a zeroth-order discontinuity and a roof edge is a first-order discontinuity in intensity. An ideal step edge and a roof edge are shown in Figure 1.1.

Ideal edges are edges in an image with no noise. In practice, images produced by scanners and other imaging devices are contaminated by various kinds of noise [9]. More specifically, due to the effects of the optical system, the observed intensity image is $f*G$, where f represents the intensity surface of the underlying scene, G, the point spread function of the optical system, and $*$, the convolution operator. An edge in $f*G$ corresponds to a discontinuity in the original signal f. Usually the function G can be approximated by a Gaussian function, which has many desirable properties. For example, the statistical properties of Gaussian noise can be represented by two parameters, namely, the mean and the standard deviation.

Besides Gaussian noise, there are other kinds of noise, such as outliers. Outliers are data points with values that are generated by an unknown process, e.g., a bad element in the camera or a transient glitch in the sensor [33]. Outliers do not conform to any parametric model of errors. In many of the existing edge detection algorithms, noise is assumed to have a Gaussian distribution. The presence of non-Gaussian distributed outliers can degrade estimates calculated by least-sum-of-squares (LSS) algorithms. The lack of immunity to outliers is a barrier to the successful use of computer vision techniques in practical applications [33].

The detection of sharp changes in image intensity requires the evaluation of several different derivatives of the noisy image. It is well known that derivatives emphasize high frequency noise. Also a fundamental conflict exists between the two performance requirements in edge detection: noise immunity and accurate localization. To attain high noise immunity requires more global information and hence, favours edge detectors with a large spatial size. The strong averaging ability of a large operator can suppress the noise effect. On the other hand, accurate localization prefers local information, and hence, favours edge detectors with a small spatial size. When applied to an input image, small detectors can reduce the interference from other nearby edges. In theory, an ideal edge detector should provide both high noise immunity and accurate localization.

Sections 2.1 to 2.5 provide a review of many edge detection schemes. They are classified into several categories. These categories may not be independent because many schemes may contain features of other categories. Some of the operators which claim to be optimal are designed to satisfy certain optimality criteria. An *optimal* solution holds only as long as the assumptions and optimality conditions are valid. There is no convincing evidence that a particular algorithm is superior than the rest.

1.2 Performance Evaluation

In edge detection, quantitative performance evaluation and comparison is not a common practice. Usually, the criteria that an edge detector is expected to satisfy are:

- Good detection: The detector must be able to mark the edge points while reducing the probability of falsely marking non-edge points.
- Good localization: The location of the marked edges should be as close as possible to the real edges.
- Robustness: The performance of the algorithm should be robust to noise and

should be consistent for various images.

- Efficiency: An algorithm that satisfies the above conditions should lend itself to an efficient implementation.

How to quantify the first three items is not a trivial issue. In practice, qualitative analysis based on human evaluation is often used as the means to judge the goodness of an edge detection scheme; if an edge detection scheme works well with a set of real life images, then it is "declared" as a good scheme in general. But in an arbitrarily selected set of real images, there are no systematic ways to tell what and where the ideal edges are. Furthermore, human perception of an edge is a complicated procedure.

To set a common framework for performance evaluation and comparison, a thorough quantitative analysis with a well designed performance evaluation model is required. Simulated images (synthetic images) with known locations of edges and controlled noise are often used. Different edge detectors are applied to the same set of images and performance measurements (using a pre-defined evaluation model) are computed. If an edge detector systematically performs better, it is considered to be a good detector. To demonstrate the ability of a detector more convincingly, the synthetic images should have a wide range of different types of edges.

As well, an edge detection scheme often has one or more parameters. Inappropriate parameter settings may yield poor results. In addition to facilitating the comparison of different edge detection schemes, performance measurements can also be used to choose the optimal parameter settings for a given edge detection scheme.

1.2.1 Outline of the Thesis

This thesis work focuses on the application of robust statistics in edge detection. The objectives are to develop robust edge detectors and an appropriate evaluation model. Two new robust edge detection schemes which are based on *least-median-of-squares* (LMS) techniques are proposed. The LMS technique is a robust statistical technique which is described in more detail in this thesis. The new detectors aim to achieve better noise immunity especially when there are outliers in the data. A new quantitative evaluation scheme is also proposed in this thesis. This new model can be used with both synthetic and real images. In the presence of outliers, experimental results show that the new robust edge detectors perform better in the presence of outliers when compared with several existing edge detection schemes.

The organization of the thesis is as follows. Chapter 2 reviews major edge detection algorithms and evaluation schemes. Chapter 3 studies the performance of the LMS estimator in robust statistics and proposes two new robust edge detectors using such an estimator. The performance of these detectors on synthetic images is compared with other operators using an existing performance evaluation model. Chapter 4 proposes a new performance evaluation model and presents the experimental results of using it in performance evaluation as well as in refining edge detection results. Chapter 5 summarizes the findings of this thesis.

Chapter 2

Edge Detection and Evaluation Schemes

Major approaches to edge detection are reviewed in this chapter. These approaches can be classified into the following categories:

1. Gradient and template approaches

2. Gaussian-like approaches

3. Regularization and residual analysis approaches

4. Surface fitting approaches

5. Robust approaches

Many schemes contain features from several categories. In Sections 2.1 to 2.5, these catagories are discussed. Section 2.6 gives a review of current edge evaluation schemes.

2.1 Gradient and Template Filters

Since edges correspond to places where intensity changes abruptly, an obvious way to locate them is to look at places with a large intensity gradient. The gradient can be obtained using a spatial differentiation operation on the input image. The results are then compared with a given threshold to generate the edge map. If the magnitude of the gradient is larger than the threshold, then an edge point is declared [14, 22, 31]. Different ways to approximate gradients in different directions lead to many edge operators.

Suppose the input image I is convolved with a set of n linear operators $H_i (i = 1, \ldots, n)$. Each operator approximates the gradient along a specific direction. An operator in edge detection is often called a *kernel* or a *mask*, and the spatial extent, i.e., the size, of an operator is called its *support*. The gradient measure at a given pixel in an image is obtained by centering the mask on top of the image at the given pixel and multiplying the pixel values with the corresponding mask values on a pixel-by-pixel basis. The sum of the products is the edge strength at the given pixel. After convolving, a set of feature-enhanced functions $G_i(j,k)$ is obtained:

$$G_i(j,k) = I(j,k) * H_i(j,k) \qquad \text{for } i = 1, \ldots, n. \tag{2.1}$$

An enhanced image A can be obtained by using an appropriate norm on G_i. An edge is said to be present at (j,k) if $A(j,k) > T$, where T is a preselected threshold.

Figure 2.1 shows examples of the gradient edge detectors, the Roberts [30], the Prewitt [29] and the Sobel [10] operators, where two orthogonal gradient functions are used. The edge enhanced array can be generated using the Euclidean norm. The edge orientation $\theta(j,k)$ at (j,k) for the Prewitt and the Sobel operators can be determined by

$$\theta(j,k) = tan^{-1}[G_2(j,k)/G_1(j,k)]. \tag{2.2}$$

Masks can also be considered as matched filters (templates) that are designed to extract the desired edge features. The presence of a feature is indicated by a large output value when the image is convolved with the appropriate template. Typical

$$H_1 = \begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix} \quad H_2 = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$$

(a) Roberts

$$H_1 = \begin{matrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{matrix} \quad H_2 = \begin{matrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{matrix}$$

(b) Prewitt (c=1) and Sobel (c=2)

Figure 2.1: Gradient operators.

$$H_1 = \begin{matrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{matrix} \quad \begin{matrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{matrix} \quad \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \quad \begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

$$H_2 = \begin{matrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{matrix} \quad \begin{matrix} 3 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{matrix} \quad \begin{matrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{matrix} \quad \begin{matrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{matrix}$$

(a) Compass (b) Kirsck (c) 3-level (d) 5-level

Figure 2.2: Template matching operators.

examples of this kind of edge detector are the compass gradient, Kirsck and other 3- and 5- level operators [31, 22, 14]. Figure 2.2 gives the first two templates of the eight possible orientation templates for each of the above operators. Each template is designed to measure an ideal step edge at a specific orientation. The edge enhanced array is computed using the maximum norm. The edge orientation $\theta(j, k)$ is determined by the direction of the template that gives the maximum response.

These two groups of edge detectors are simple, efficient and easy to implement. Unfortunately, they are very sensitive to noise and tend to create spurious edges in a noisy environment. One way to increase noise immunity is to employ a large operator, which can suppress noise, but is likely to produce inaccurate edges and corners.

2.2 Gaussian-Like Filters

2.2.1 Laplacian of the Gaussian

Most edge detection schemes begin by applying small differential operators to an image, followed by a detection operation to locate small edge segments. Directional operators must be applied at different directions to detect edges at these orientations in the image. A sharp change in the intensity will give rise to a peak in the output of the first derivative operator or a zero crossing in the output of the second derivative. Hence, the initial differentiation is generally followed by peak detection or zero crossing detection, and thresholding is then applied to extract boundaries and to remove edge segments that arise from noise in the imaging process.

Marr and Hildreth [23] propose a two-dimensional symmetric Gaussian function,

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.3}$$

as a filter in edge detection, where (x, y) are coordinates relative to the center of the mask. Then, the zero crossings of the rotationally invariant second derivative operator, namely, the Laplacian operator ∇^2, are used to detect edges.

The Gaussian function satisfies the criterion of minimizing the product of the spatial variance and the frequency variance. The convolution operator has the following property:

$$\nabla^2(G(x,y) * I(x,y)) \equiv (\nabla^2 G)(x,y) * I(x,y) \tag{2.4}$$

where $I(x, y)$ is the input image. Combining the Laplacian operator with the Gaussian function, the filtering and the differentiation processes are merged into one step. Hence the final edge detector is the Laplacian of the Gaussian (**LoG**) point-spread function:

$$(\nabla^2 G)(x,y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{2.5}$$

Edges are located at the zero crossings of the output of the **LoG** operator. The size of the operator is defined by w, which varies linearly with the spatial constant σ of the underlying Gaussian as $w = 2\sqrt{2}\sigma$.

LoG is important because of its computational efficiency and its ability to locate edges at different scales determined by the standard deviations of the Gaussian filters. **LoG** is accurate only when a set of assumptions are met. These assumptions are:

- Edges have infinite length.
- Edges are straight.
- The illumination varies linearly along the edge.

Under the above assumptions, it has been proven that the zero crossings of **LoG** are at the exact location of the original edges [3]. In the case of a right-angle corner, the contour found by **LoG** goes through the vertex of the corner but deviates from the sides of the corner. The maximum displacement of the contour from the true edge is proportional to σ, where σ is the standard deviation of the Gaussian filter [3]. The behaviour of **LoG** in the case of an infinite corner with an angle θ, a corner subtended by finite edges and circular edges, and the effect of nonlinear intensity on edge detection by **LoG** have been studied [3]. It was found that when there is nonlinear illumination, displacements of the contours and spurious double contours occur.

Ulupinar and Medioni [36] propose techniques to correct some of the problems associated with the **LoG** filter. A second derivative operator generates a zero crossing whenever the input function has an inflection point. But this inflection point may not always correspond to edges [36]. A zero crossing corresponds to an edge iff $f^{'}f^{'''} < 0$. $f^{'}$ can be obtained by convolving the input with the first derivative of the Gaussian. Only the sign of $f^{'''}$ is needed. At each zero crossing, the above test can be applied to eliminate faulty edges.

The displacement problem can be corrected by several techniques [36]. The first one is to convolve the input signal with two operators of different σ values. In the case of an ideal step edge, the zero crossing of the **LoG** is located at x_z given by:

$$x_z = t + c\sigma^2 \tag{2.6}$$

where t is the true location of the step edge and c, the ratio of slope difference on each side of the edge to the height of the edge. t and c are two unknowns. By convolving with two operators with different values of of σ, two equations are obtained. Therefore, the true location of the step edge can be found. The main drawback of this method is that the zero crossings in the convolved image with a small value of σ may correspond to one or no zero crossing in the convolved image with a larger value of σ. Also, Equation (2.6) is applicable only to step edges.

The second method is to use the locations of the maximum (x_p), the minimum (x_n) and the zero crossing (x_z) of the convolution output to correct the displacement. The true edge is located at $t = x_p + x_n - x_z$. Another technique uses the values of the convolution output at different locations *around* the zero crossing x_z. Suppose m_1 and m_2 are the values of the convolution output at $x_z + h$ and $x_z - h$, respectively, where h is a parameter. The location of the true edge is at [36]:

$$t = x_z + \frac{\sigma^2}{2h} log \left| \frac{m_1}{m_2} \right| \tag{2.7}$$

In this method, the parameter h is quite critical. Because of the interaction between edges, an improper value of h can lead to poor accuracy.

The use of the Gaussian filter and the zero crossing as an indication of an edge is an important concept in the theory of edge detection. There are still a few questions associated with the **LoG** operator. Experiments show that for images with a large signal-to-noise ratio (SNR), the **LoG** operator of any given size is capable of detecting edges [17]. In the presence of noise, however, a small size operator is unable to detect high contrast edges. Large operators yield more reliable detection of the intensity changes. The integration of zero crossing descriptions from operators with different sizes remains an open research question. Hildreth [17] suggests that the spatial coincidence of zero crossings from nearby operators of different sizes may be used to assert the presence of an edge. Unfortunately, zero crossings do not necessarily appear in operators with different σ.

Another problem is that the assumption that the change in intensity near the line of zero crossing is locally linear is often false, in which case the zero crossings are displaced from the true edges. Directional operators can achieve better detection and localization performance than rotationally invariant operators, such as **LoG**, especially around corners [35]. An intuitive reason is that a directional operator takes averaging only along the edge direction, rather than across it. So a directional operator effectively reduces noise without blurring the edge as severely as a circularly symmetric two-dimensional operator which smoothes the image along the edge direction as well as across it.

2.2.2 Optimal Filter

Canny [5] proposes an optimal detector that optimizes the following performance and constraint criteria:

- Good detection: There should be a low probability of missing real edge points and a low probability of falsely marking non-edge points.
- Good localization: The distance between the points marked as edge points by the detector and the true edge should be minimized.
- A single edge should yield only one response.

Using variational techniques with the step edge as input, the optimal operator in the one-dimensional case which optimizes the above criteria is a linear combination of four exponentials [5]. The shape of the function can vary from a smooth difference of square pulses to the first derivative of the Gaussian. The one that yields the best performance is the one with a shape similar to the first derivative of the Gaussian. Canny chooses the one-dimensional first derivative of the Gaussian, $G^{'}$, as his optimal edge detector. Two dimensional extension is obtained by rotating the one-dimensional operator in different directions.

Canny also proposes two useful post processing methods, called *feature synthesis* and *hysteresis thresholding*. Feature synthesis is a heuristic fine-to-coarse method to integrate results from different scaling operators. It begins by marking all edges from the smallest operator which has sufficient output signal-to-noise ratio. Let the smallest operator be denoted by G'_{σ_1}, where σ_1 is its standard deviation, and the resultant edge map $E_1 = G'_{\sigma_1} * I$, where I is the input image. The synthesized output, E_{2syn}, is generated by convolving E_1 with G'_{σ_2}, i.e., $E_{2syn} = E_1 * G'_{\sigma_2}$, where σ_2 is larger than σ_1.Then the actual operator's output, $E_2 = G'_{\sigma_2} * I$, is compared to the synthetic one, E_{2syn}. Additional edges are marked if the output from the larger operator has a significantly greater response than what the synthetic output predicts. In this way, edges that exist in the large spatial scale and are superimposed on several sharp edges can be distinguished.

Hysteresis thresholding is used to reduce streaking of an edge contour caused by the fluctuation of the operator's output above and below the threshold along the length of the contour. Hysteresis thresholding is a two-stage thresholding scheme. For a single thresholding scheme, it is difficult to set a threshold so that the probability of marking noise as edges is low while retaining high sensitivity to true edges. With a two-stage thresholding scheme, the group of pixels that are above the high threshold together with the group of pixels that connect to them and are above the low threshold are marked as edges.

With these postprocessing steps, the detector yields better performance than the simple **LoG**. On the other hand, if an edge in not an ideal step function or not well isolated, the assumptions used by Canny to derive his optimal detector may no longer be true. The optimality of the resulting operator is, therefore, unsubstantiated.

2.3 Regularization and Residual Analysis

Regularization provides a general framework to transfer ill-posed problems into well-posed problems by restricting the class of admissible solutions using constraints such as smoothness. A problem is said to be well-posed if a unique solution exists and is continuously dependent on the data. Since edge detection violates the continuity requirement of the well-posed problem, regularization theory can be used to transfer it into a well-posed problem [13]. This is achieved by imposing constraints on the solution in the variational form such as a cost functional. In regularization theory, smoothness can be imposed on the function f as follows. Find the function f that minimizes:

$$E(f) = \beta \parallel Pf \parallel_1^2 + \parallel Af - d \parallel_2^2 \tag{2.8}$$

where A and P are linear operators, d represents the data, $\parallel \bullet \parallel_1$ and $\parallel \bullet \parallel_2$ denote suitable norms. P is chosen so that a smaller $\parallel Pf \parallel_1$ implies a smoother solution. β, the regularization parameter, controls the compromise between the smoothness of the solution, the first term, and the closeness to the data, the second term [20].

2.3.1 Iteratively Refined Regularization

By choosing stabilizers and the usual quadratic norm, the minimization problem in the one-dimensional case becomes:

$$E(f) = \int (f-d)^2 dx + \lambda \int f_x^2 dx \tag{2.9}$$

where $E(f)$ is the functional to be minimized. The main problem of the standard regularization is the global assumption of smoothness by disregarding the discontinuities in the scene characteristics that are especially important for detecting edges. This translates to finding a suitable value of λ for the whole image. An alternative approach proposed by Gokmen and Li [12] is to find local solutions of λ. The minimization problem is reformulated as follows:

$$E(\lambda, f) = \int_\Omega (f-d)^2 dx + \int_\Omega \lambda(x) f_x^2 dx \tag{2.10}$$

In this formulation, the parameter λ of the standard regularization is treated as a variable. The compromise between the closeness to the data and the smoothness of the solution is spatially determined by the value of $\lambda(x)$ at each point x. Hence, a large $\lambda(x)$ enforces the smoothness constraint resembling the convolution with a large filter and a small $\lambda(x)$ forces the solution to follow the image intensity data $d(x)$ by weakening the connection between successive pixels. Assuming the image is corrupted by zero-mean Gaussian noise, the residue $e = f - d$ can be due to either noise or discontinuity. If the error distribution around a point appears to be random, then it is assumed to be due to noise. If there exists a systematic trend, it is most likely due to discontinuity. A control signal, c, computed from the residue, is used to determine λ at each pixel location [12]. Then a new solution f and a new value of λ can be obtained. $\lambda(x)$ is refined iteratively until there is no change. The extension of this approach to 2-D algorithms is discussed in [13]. Iteratively refined regularization is a parallel algorithm and it can extract multisize features in the image during the iteration procedure.

2.3.2 Residual Analysis

As the stabilizer in standard regularization cannot deal with discontinuity problems in early vision, a class of nonstandard stabilizers are proposed. A convenient way to control the continuity properties of a generalized spline stabilizer of order m is to blend it with generalized splines of lower order. These stabilizers are generally nonquadratic and define a class of nonlinear and spatially noninvariant regularization methods.

Grimson and Pavlidis [15] propose using residual analysis instead of the differentiation operation for detecting discontinuities. Residual analysis compares the initial data $f(x,y)$ with their smoothed version $g(x,y)$. The residual is: $e = f - g$. The results from analyzing several functions f reveal that discontinuity in f will manifest itself as a change in the sign (i.e., zero crossing) of the residuals. The smoothed output g from the image data f can be obtained by computing a planar approximation to the image data f at each point. In the case of a step edge, supposing that $f(t)$ is an ideal step function at 0 of size S and $g(t)$ is the straight line minimizing the maximum pointwise error over a *small* interval $[-\delta, \delta]$, it can be derived that the residue is $e = \frac{S}{4}t$. It can be seen that the edge lies at the zero crossing of the residuals. The residuals on either side of the edge have a value proportional to the step size [15]. It is also claimed that the proposed method seems to detect discontinuities correctly as long as their size is large compared to the noise's amplitude. The operational advantage of this method is that there is no need to set a threshold when deciding about discontinuities [15]. The weakness is that there is no quantitative analysis of the reliability of the zero crossings of residuals in the presence of noise.

Lee and Pavlidis [20] propose another technique to apply more general smoothing splines, which incorporate the constraints from discontinuities in an explicit way. They define the discrete regularization as the set of z_i $(i = 0, \ldots, n)$ that minimizes:

$$E = \sum_{i=0}^{n} \xi_i (z_{i+1} - 2z_i + z_{i-1})^2 + \beta \sum_{i=1}^{n} (z_i - f_i)^2 \tag{2.11}$$

where $x_{i+1} - x_i = 1$, assuming data points are equally spaced. If one of the ξ_i's is set to be zero, sharp corners are preserved. If a pair of ξ_i and ξ_{i+1} are set to be zero, then there will be a discontinuity at x_{i+1}. The experimental results are in one dimension [20]. The extension to 2-D still needs to be studied.

Chen et al. [8] present a formal analysis justifying the claim that the residuals contain the nonpredictable part of the data. A common approach in residual analysis is to compute an error norm between the original data and the smoothed version and see if it is small. It is difficult to decide what is *small*. So an alternative way is to consider the distribution of the residuals. The autocorrelation function of the residuals $r(t)$ is defined as:

$$A_{rr}(t) = \int r(z) r(z+t) dz. \tag{2.12}$$

If the residuals are white noise, then $A_{rr}(t)$ is identically zero for nonzero t, while $A_{rr}(0)$ is equal to the noise power. Therefore for nonzero t_0, the ratio

$$R(t_0) = A_{rr}(t_0)/A_{rr}(0) \tag{2.13}$$

should be small if the residuals are due to noise. A threshold on the error norm depends on the amplitude of the noise, while a threshold on the ratio R depends only on the form of its autocorrelation function and is easier to select. These arguments lead to a new edge detector [8], the results of which have been compared with that from Canny's edge detector. Qualitatively, the new detector seems to perform better on sharp corners. But the processing time for the autocorrelation computation appears excessive and there are no quantitative evaluations.

2.4 Surface Fitting Methods

An image is a two-dimensional array of discrete intensities that can be considered as the samples of the underlying three-dimensional image surface contaminated by noise. Therefore, a natural way for edge detection is to fit a surface by a set of basis functions. The edge properties, such as the edge position, orientation and contrast, are then estimated from the best-fitted surface characterized by the parameters of the basis functions. The fitted surface possesses the same degree of smoothness as its basis functions. Hence, the surface fitting approach offers an effective way to suppress noise.

The basis functions are very important in surface fitting schemes. When the basis is inadequate, it cannot give a good description of the features of the surface. On the other hand, if the basis is not minimal, it will be noise-sensitive [27]. Consider the hypothetical situation where noisy samples of y, which is a quadratic function of x, are given. The goal is to determine a description of the underlying curve. Obviously, fitting a straight line will not be adequate. Moreover, fitting a polynomial with a higher order than a quadratic is going to have nonzero coefficients for x^i, $i > 2$, owing to the noisy nature of the data. Hence, even though polynomials of order greater than 2 give a smaller least-sum-of-squares error than a polynomial of order 2, it is still desirable to fit a quadratic. This is a simple illustration of the importance of the basis functions.

It is believed that the surface fitting approach is an effective way to detect edges, since it can detect local features as well as suppress the noise. The following sections discuss in detail three surface fitting schemes that form the basis of this proposed research.

2.4.1 Polynomial Fitting

Haralick [16] proposes that the underlying gray tone intensity function f takes the parametric form of a polynomial in the row and column coordinates. The polynomials used are the discrete Chebyshev polynomials, denoted in the one-dimensional case by $\{P_i(r), i = 0, 1, \ldots, N-1\}$, where $r \in R$, and R is the symmetric pixel index set, that is, if $r \in R$, then $-r \in R$. They are orthogonal with respect to the discrete inner product over the operator's support. Two-dimensional discrete orthogonal polynomials can be created from two sets of one-dimensional discrete orthogonal polynomials by taking Cartesian products [16]. One and two-dimensional examples are shown as follows:

Index Set	Discrete Orthogonal Polynomial Set
$\{-1, 0, 1\}$	$\{1, r, r^2 - 2/3\}$
$\{-1, 0, 1\} \times \{-1, 0, 1\}$	$\{1, r, c, r^2 - 2/3,$
	$rc, c^2 - 2/3, r(c^2 - 2/3),$
	$c(r^2 - 2/3), (r^2 - 2/3)(c^2 - 2/3)\}$

In the one-dimensional case, the approximate fitting problem is to determine $a_0, \ldots, a_k, k \leq N-1$, such that

$$e^2 = \sum_{r \in R} [d(r) - \sum_{n=0}^{k} a_n P_n(r)]^2 \tag{2.14}$$

is minimized, where $d(r)$ is the observed data value at $r \in R$. By setting to zero the partial derivatives of e^2 with respect to each of the a_i's, the solution is obtained:

$$a_m = \sum_{r \in R} P_m(r) d(r). \tag{2.15}$$

It can be seen that a_m is a linear combination of the data values. The coefficients of P_m are easy to compute and can be used as a template to compute a_m. Templates for different windows sizes are given in [16].

Once the fitting coefficients a_i, $i = 0, \ldots, k$, have been computed, the estimated polynomial $Q(r)$ is given by

$$Q(r) = \sum_{i=0}^{K} a_i P_i(r). \tag{2.16}$$

In the two-dimensional case, only a cubic polynomial in r and c is used. Other terms with higher degrees than three are truncated. $Q(r)$ is rewritten to be f:

$$f(r,c) = k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 + k_7 r^3 + k_8 r^2 c + k_9 rc^2 + k_{10} c^3. \tag{2.17}$$

The gradient of f and the gradient direction angle α at the center of the neighbourhood used to estimate f can be computed. The second directional derivative can then be derived:

$$f_\alpha''(\rho) = A_\alpha \rho + B_\alpha \tag{2.18}$$

where α is the gradient direction, $\rho = r^2 + c^2$, along the gradient direction, and A_α and B_α are coefficients determined by α and $k_i, i = 1, \ldots, 10$. If for some ρ, $|\rho| < \rho_0$, where ρ_0 is slightly smaller than the length of the side of a pixel, $f_\alpha'''(\rho) < 0, f_\alpha''(\rho) = 0$ and $f_\alpha'(\rho) \neq 0$, then the center pixel of the neighbourhood is marked as an edge pixel. In other words, an edge is located at the zero crossing of the second directional derivative of the fitted image surface if the first derivative is nonzero and the third derivative is negative. Haralick has also discussed the statistical analysis of this technique illustrating how to determine confidence intervals for the direction of the gradient, and how this interval determines a confidence interval for the placement of the zero crossings.

It should be noted that the kernel used to estimate the derivative depends on the neighbourhood size, the order of the fit, and the basis functions used for the fit. In Haralick's approach, only polynomials up to degree three are chosen as basis functions. In fact, the number of discrete polynomials over a given support is equal to the number of pixels in that support. Higher order polynomials are not preferred because they have extra wiggles and perform badly in LSS fitting. This again reinforces the statement that in surface fitting, the basis function space and the function set within this space are key factors in these kinds of approaches. The drawback of the Haralick operator is that its performance at the saddle points (corners) is poor because the gradient is zero there.

2.4.2 *Tanh* Function Fitting Approach

Nalwa and Binford [27] propose an edge detection scheme that uses the *tanh* function as the basis function for surface fitting. This approach aims to detect *edgels*: short linear segments characterized by their positions and orientations. The *tanh* function defined as follows has a step-like shape:

$$Q(z) = s \tanh\,(f(z+p)) + k \tag{2.19}$$

where s, p, and k are the parameters and f is a constant determined by the blur of the imaging system. s represents the edge contrast, p, the displacement of the zero crossing, and k, the average of intensities within the support of the operator. The maximum error in approximating an ideal step-edgel by the *tanh* function is less than 1 percent of the step-size [27]. For edges that have large deviation from an ideal step-edgel, this basis will not be adequate and other basis functions must be added.

In this *tanh* fitting scheme, the image surface within the support is considered to be a one-dimensional surface. This results in effective noise reduction without blurring the edges as severely as with circularly symmetric operators. The noise in an image is reduced by averaging data in a direction parallel to the edge if the direction of the edge could be reliably found. This relies on the fact that the window, i.e., the kernel of the operator, is small enough for the edge-segment in it to be modeled as an edgel. The above smoothing is achieved by fitting each window with a one-dimensional oriented surface, i.e., a surface that is constant in one direction.

The steps involved in this algorithm are summarized as follows:

1. *Planar-Fit:*

   Perform a least-sum-of-squares planar-fit to the window:

   $$\xi_p = \sum_{x,y} (I_{(x,y)} - (a_0 + a_1 x + a_2 y))^2 \tag{2.20}$$

   where, (x, y) are pixel coordinates in the window. Minimize ξ_p w.r.t. a_0, a_1 and a_2.

   The initial estimate for the direction of variation in the window is given by the gradient of planar-fit:

   $$\theta_0 = tan^{-1}(a_2/a_1). \tag{2.21}$$

2. *1-D Cubic-Fit:*

   Refine the estimate of the direction of variation by fitting a 1-D cubic surface with the least-sum-of-squares error criterion.

   $$\xi_c = \sum_{x,y} (I_{(x,y)} - (a_0 + a_1 z + a_2 z^2 + a_3 z^3))^2 \tag{2.22}$$

   where

   $$z = x\ cos(\theta) + y\ sin(\theta) \tag{2.23}$$

   is the displacement from the origin along the direction of variation.

   For a fixed θ, ξ_c is minimized w.r.t. a_0, a_1, a_2 and a_3. The angle θ is refined to the nearest 5° using Newton's method. It is the final estimate of the direction of variation in the window. This angle is used in the following surface fitting. The direction of the edgel is orthogonal to θ.

3. *1-D Tanh-Fit Along θ:*

   Find the least-sum-of-squares 1-D $tanh$ surface oriented in the direction θ:

$$\xi_t = \sum_{x,y} (I_{(x,y)} - (s\ tanh(f[z+p]) + k))^2 \tag{2.24}$$

   Minimize ξ_t w.r.t. s, p and k. Since ξ_t is nonlinear in p, p is refined iteratively using Newton method with the initial value set to be zero. For each p, ξ_t is minimized w.r.t. s and k. f is the scaling constant chosen to be $f = 0.85/\sigma_{blur}$ [27].

4. *1-D Quadratic-Fit Along θ:*

   Find the least-sum-of-squares 1-D quadratic fit oriented in the direction θ:

$$\xi_q = \sum_{x,y} (I_{(x,y)} - (a_0 + a_1 z + a_2 z^2))^2 \tag{2.25}$$

   Minimize ξ_q w.r.t. a_0, a_1 and a_2. ξ_q is linear in its parameters.

5. *Comparison:*

   If the least-sum-of-squares error in quadratic-fit is less than that in $tanh$-fit, the absence of an edgel is declared. Otherwise, an edgel exists and its position is given by the displacement p with a step size of $2s$. In practice, thresholds are needed in p and s.

There are several advantages to choosing $tanh$ as the basis function. First, the basis requires the specification of only three parameters, i.e., p, s, and k, to determine the position and the upper and lower intensities of the step-edge, respectively. It follows that smaller windows than previously discussed approaches can be used. This implies that the $tanh$ fitting approach offers better localization capabilities and improved performance on high-curvature edges. Secondly, in determining the edges, 5 × 5 window:

211	139	99	29	20
201	159	90	76	25
191	219	97	122	26
202	214	162	123	39
202	206	180	109	100

Fitting Surface	θ	Fitting Equation	Fitting Error
Planar	20.80	$I(x,y) = 129.64 - 41.44x - 15.74y$	13078.70
Cubic	22.12	$I(x,y) = 138.60 - 59.33z - 4.48z^2 + 3.62z^3$	8307.58
$Tanh$	22.12	$I(x,y) = 129.64 - 74.57tanh(1.42(z - 0.001))$	14466.69
Quadratic	22.12	$I(x,y) = 138.84 - 44.32z - 4.60z^2$	10978.41

Since $tanh$ fit error is larger than quadratic fit, the absence of edgel is declared. Actually, the center of the window is a very good candidate for an edge point.

Figure 2.3: An example in the $tanh$ fitting approach.

no derivative estimation is involved, and hence, the influence of noise is reduced. Lastly, edge properties can be directly estimated from the fitting parameters.

There are also problems associated with the $tanh$ fitting approach. The basis has a highly constrained nature and is not adequate to detect other kinds of edges such as roof edges. It is also nonlinear in its parameters. Fitting a nonlinear basis requires an iterative process and is therefore computationally more expensive than fitting a linear basis. In practice, Newton method is used to do various fittings and refinements. Whether it will converge or not is not rigorously proven. Another problem is that whether the error of the $tanh$ fit can be compared with the error of the quadratic fit to declare the presence of an edgel in a noisy environment is debatable. An example is given in Figure 2.3 where the center pixel is a good edge candidate. The edge direction is 135 degrees. The figure shows that even when the quadratic fit has a smaller fitting error, there may exist an edgel in the fitted window.

2.4.3 Regularized Cubic B-Spline Fitting (RCBS)

Chen and Yang [6] propose a new edge detection method, called fitting with regularization, which aims to achieve better noise immunity and to preserve accurate localization. In this approach, the basis function space is a linear space and is spanned by a set of cubic B-splines (CBS). CBS are more appropriate for surface fitting because of their flexibility and capability to model real image surfaces. Compared with simple polynomials of high degree, splines have the same flexibility but with less risk of severe oscillations. The highest degree of the proposed basis function space is three. When the support of the operator increases, the dimension of the basis function space increases accordingly; i.e., the number of nonzero cubic splines over the support increases. But the highest degree of the basis function space remains unchanged. This can avoid the oscillation problems common to polynomial fitting, in which the highest degree of the basis function space increases as the support of the operator increases. CBS also have the advantage of computational feasibility, since the highest degree of CBS is 3 in the one-dimensional case and 6 in the two-dimensional case.

Suppose $\Psi = \{\psi_i(x) \mid i = 1, \ ..., \ N\}$ represents all the nonzero cubic B-splines over the interval $[0, I]$, in which $N = I + 3$. The basis function set used for fitting is $\{e_1, \ e_2, \ ..., \ e_V\}$, $1 \leq V \leq N$. Thus the fitting can be written as:

$$S(x) = \sum_{i=1}^{V} c_i e_i(x), \qquad 1 \leq V \leq N. \tag{2.26}$$

But in a noisy environment, the fitting results may still follow the raw image data too closely, and hence, noise is not sufficiently suppressed. So the regularization technique is added and the regularized cubic B-spline fitting (RCBS) approach is proposed. The problem of fitting an input function f in the basis function space of RCBS becomes one of determining the coefficients of the fitting function $S(x)$ in Equation (2.26), so that the following function is minimized:

$$e^2 = \sum_{j=1}^{M}(f(x_j) - \sum_{i=1}^{V} c_i e_i(x))^2 + B\sum_{j=1}^{M}(\sum_{i=1}^{V} c_i e_i^{''}(x_j))^2 \tag{2.27}$$

where M is the number of data points, and B is the regularization factor that controls the degree of smoothness imposed on the solution.

This edge detection scheme uses a one-dimensional surface model. Two-dimensional image data are reduced into one dimension by projecting the data in the direction of least local intensity change, i.e., the direction perpendicular to the direction of maximum local gradient as shown in Figure 2.4. The horizontal-vertical grid is the image grid, while the slanted grid in this case is the operator grid. The operator grid is rotated according to the gradient direction. In cases when the nodes of the operator grid do not coincide with the nodes of the image grid, but lie in the holes of the image grid, a bilinear interpolation is done to get the data value at each operator grid node. Then equal weight averaging of operator grid nodes is used to compute each data point. Masks for different projection size and direction are generated beforehand to speed up processing.

In this approach, the center pixel of the support is declared to be an edge pixel if the zero crossing of the second derivative is within the maximum allowable displacement, the third derivative at the zero crossing is nonzero, and the magnitude of the first derivative is larger than a pre-selected threshold. Experiments in [6] show that an operator of size 7×5 generally gives good performance. Compared with the Sobel operator and Haralick's second directional derivative operator, the RCBS approach performs best on synthetic images with high noise level. On real images with low noise levels, the RCBS detector produces similar outputs to the Haralick operator.

In the proposed RCBS scheme, the regularization parameter B is a noise sensitive parameter. A larger value of B has a stronger smoothing effect, thus providing better noise immunity. Yet thicker edges are produced. A smaller value of B produces

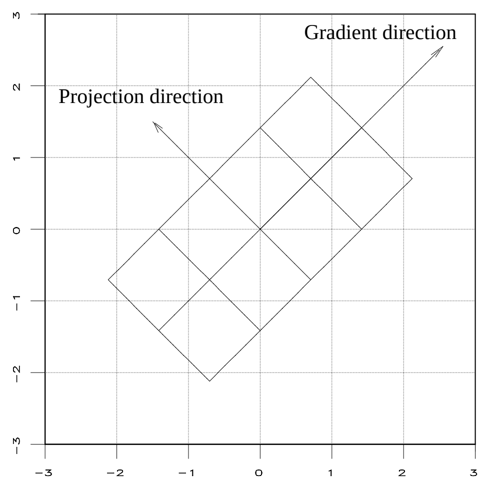


Figure 2.4: Projecting two-dimensional data into one dimension.

thinner edges, but is sensitive to noise. In the paper [6], a performance evaluation model is proposed and is used to choose the optimal value of B. Unfortunately, the outputs using this optimal B do not yield the best visual result. The global assumption of smoothness is also a problem.

2.5 Robust Methods

2.5.1 Robust Regression Techniques

The well-known *least-sum-of-squares* (LSS) method

$$\text{Min} \sum_{i=1}^{n} r_i^2 \tag{2.28}$$

is the cornerstone of classical statistics and has been widely used in many computer vision algorithms. The basic idea of LSS is to optimize the fit by minimizing the sum-of-squares of the residuals, r_i's, where r_i is the deviation of the observation value

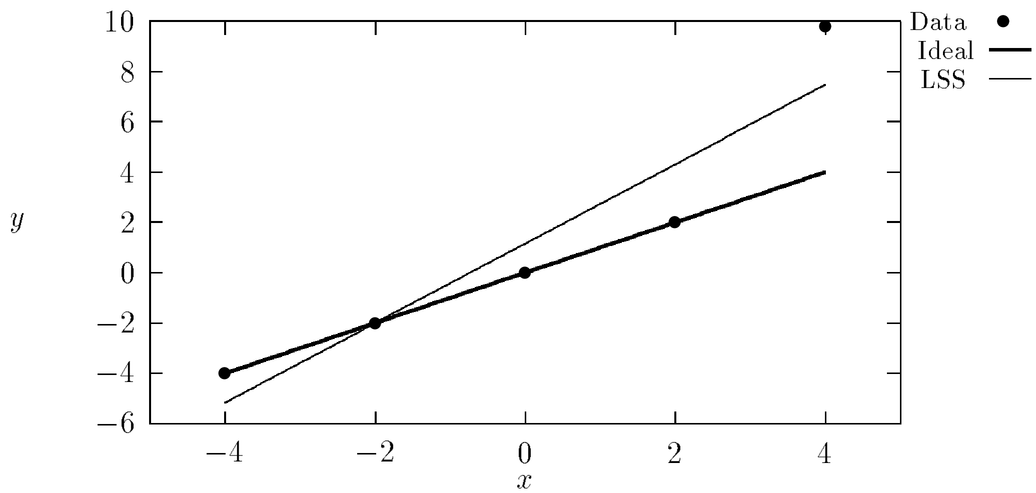


Figure 2.5: Behaviour of LSS fit in the presence of outliers.

from the fitted value. It is proven that if the error has a Gaussian distribution, the LSS fit is optimal [32]. As stated in the introduction, an image is contaminated not only by Gaussian noise but also by outliers, which are data that do not conform to any known models. The presence of outliers may severely ruin the estimates calculated by LSS algorithms [33]. Figure 2.5 is an illustration of this case, where one outlier in the y-direction can make the LSS fit far from the "ideal" line by an arbitrary amount.

Robust statistics offers methods for extracting measurements from corrupted data. The basic robust estimators are classified as M-, R-, and L-estimators [24].

- M-estimators:

$$\text{Min} \sum_i \rho(r_i) \tag{2.29}$$

  where $\rho(r_i)$ is a symmetric, positive-definite function with a unique minimum at $r_i = 0$. For the LSS method, $\rho(r_i) = r_i^2$. Several ρ functions have been proposed to reduce the influence of large residual values on the estimated fit.

- R-estimators: These estimators are based on the ordering of residuals.

$$\text{Min} \ \sum_i a_n(R_i) r_i \tag{2.30}$$

  where R_i is the location of the ith residual in the ordered list. The score function a_n must be monotonic and $\sum_i a_n(R_i) = 0$.

- L-estimators: These estimators employ linear combinations of order statistics. The median and α-trimmed-mean based methods belong to this class [24].

The breakdown point of a regression method is the smallest amount of outlier contamination that can bias the value of the estimate to an arbitrarily chosen value. (A formal definition of breakdown point is given in Section 3.2.1.) 50% is the best attainable breakdown point, since for larger amounts of contamination, it becomes impossible to distinguish between the "good" and the "bad" parts of the sample. The M-, R-, and L-estimators have breakdown points that are less than or equal to $1/(p+1)$, where p is the number of parameters in the regression [24].

Some robust estimators with 50% breakdown point have been proposed. One of them is the *repeated-median* (RM) method for solving multidimensional regression problems. RM has been successfully applied to problems with a small number of parameters. The drawback of RM is that the resultant estimator is dependent on linear transformations such as scaling and coordinate transformation. Another important robust estimator is the *least-median-of-squares* (LMS) estimator, given by

$$\text{Min Med} \ \ r_i^2. \tag{2.31}$$

LMS also has a 50% breakdown point and is quite robust to noise. It is discussed in detail in Chapter 3 in this thesis.

2.5.2 Robust Edge Detectors

In recent years, there has been a substantial effort in incorporating robust statistics in computer vision algorithms. Shunck et al. [34] present a two-stage algorithm for visual surface reconstruction from scattered data while preserving discontinuities. The first stage uses robust regression (the moving least median of squares (MLMS) method) to remove outliers. The second stage uses weighted bicubic splines to approximate the optimal function of an adaptive regularization problem. The weighted bicubic splines are able to fit discontinuity with fidelity. Experimental results show that the algorithm has removed outliers successfully [34]. Formulating computer vision algorithms using robust methods provides immunity to outliers and preserves discontinuities. The robust MLMS reconstruction algorithm trades off the statistical efficiency of least squares regression (for normally distributed noise) for resistance to outliers.

In edge detection, Mintz [26] proposes a robust consensus based detection scheme. The conventional fitting methods are sensitive to noise and the order and the type of function that is fitted to the data. The new paradigm is based on robust estimation of fits to local windows and voting to detect edge pixels. Using the shift invariant property of robust estimators, the correct fit should be recovered in many overlapping windows that contain the same discontinuity. In this approach, robust fits are computed at each pixel of the image using $n \times n$ windows. Pixels in each window are classified into inliers and outliers. The robust fit is also used to declare the existence of an edge in the window. All pixels that are on the boundary of the step edge are marked. For a given image pixel P, if the number of the accumulated votes from $m \times m$ different windows that contain P has more than $\lfloor \frac{m^2}{2} \rfloor$ positive votes, then P is declared to be an edge pixel.

The models used to fit the data are only planar and constant fits, and a *consensus by decomposition* (CBD) robust estimator is used to do the fitting. Planar fit does result in detecting false edges. It is claimed in the paper [26] that when the consensus scheme is used, the false edge will not appear in most overlapping windows at exactly the same place. However, this argument is based mostly on experiments and not on formal analysis. The CBD estimator introduced by Meer, et al. is also based on consensus seeking, trying to eliminate the outliers. The optimization criterion of CBD is not that of least-median-of-squares residuals. Though CBD achieves speed up relative to the LMS estimator [25], its breakdown point is influenced by the reliability of the consensus process, i.e., by the signal-to-noise ratio (SNR). This is not desirable for edge detection.

The window size used to do the fit is set to be 9×9, which is considered to give good results [26]. This is questionable, since for such a large window the underlying step edge model may not be true. The size of the consensus set (m^2) depends on the support needed for decision making. m is a noise sensitive parameter and needs to be chosen such that the overlapping windows contain *mostly* the same discontinuity and allow enough values for a good consensus based decision.

2.6 Performance Evaluation Schemes

As discussed in Section 1.2, although qualitative analysis based on human evaluation is widely used, a quantitative approach is important to evaluate the performance of edge detectors. It should be noted here that quantitative evaluation is different from post processing. It takes the edge map (with gradient map in most cases) as input and computes the performance measurements. The goals of quantitative evaluation are summarized as follows:

- It provides a way to quantitatively study the performance of an edge detector with respect to its parameters. Parameter settings that give the best performance measurements can be determined. The behaviour of the detector under different noise levels can also be studied.

- It facilitates the comparison of different edge detectors.
- The performance measurements can give an indication of the quality of the original image.

To achieve the above goals, the evaluation model has to be carefully chosen. Several models have been proposed. Section 2.6.1 reviews models that require *a priori* knowledge of the location of the actual edge. Section 2.6.2 discusses evaluation models based solely upon the goodness of edge formation. The criteria of a good evaluation model are:

- Better evaluation measurements computed using the model conform to a human's perception of a better image.
- The model is supposed to be edge detection scheme independent and thus can make comparison more feasible (without the need to repeat the time-consuming implementation of various edge detection schemes).
- The model is applicable with or without *a priori* knowledge of the location of the true edges.

2.6.1 Ideal Edge Map Based Evaluation Models

One of the earliest evaluation schemes, proposed by Fram and Deutsch [11], aims to study the effect of noise on various edge detectors. Synthetic images composed of three vertical panels are used and the exact locations of edge pixels are known. The output is evaluated according to two measures. The first measure, P_1, measures the fraction of the detected edge pixels that are actually edge points. The second measure, P_2, computes the fraction of the vertical extent of an edge that is covered by detected edge pixels. These measures have very limited applications, since they can be used on vertical edges only. Also, different edge maps can give the same measure P_2.

Abdou and Pratt [1] propose a model that expresses the performance of an edge detector as a function of the number of ideal edges, I_I, the number of true edges, I_A, and the error distance between the actual edge and the ideal edge, $d(i)$:

$$F = \frac{\sum_{i=1}^{I_A} \frac{1}{1+\alpha(d(i))^2}}{\max\{I_A, I_I\}} \tag{2.32}$$

where α is a scaling factor and is set to be 1/9 in their experiments. The denominator in this function is a normalization factor which penalizes F when too few or too many edge points are detected, while the numerator measures the displacement of each detected edge pixel from its ideal position. Again, with modifications, the figure of merit method is implemented only for vertical and diagonal edges, which is far from enough in practical applications.

In [37], Venkatesh and Kitchen develop an edge-evaluation method that comprehensively combines a set of error components. The error set contains:

1. **False negatives** (FN): the edge detector fails to detect a true edge.
2. **False positives** (FP): the detector reports an edge where there should not be one.
3. **Multiple detections** (MD): the edge detector detects several edges instead of a one-pixel wide edge.
4. **Localization error** (LOC): this is the error between the position of the edge as reported by the edge detector and the true edge position.

The proposed failure measure (FM) reflects the total *error cost* and is defined as

$$FM = \alpha FP + \beta FN + \gamma MD + \delta LOC \tag{2.33}$$

where the cost factors satisfy $\alpha + \beta + \gamma + \delta = 1$. While combining different error components is a good idea in evaluation, the definitions of FN, FP, MD and LOC are constrained to vertical edges (by extension, oriented edges), which limits the applicability of this approach [37]. A more accurate examination of the nature of edge pixels given in [6] leads to a new evaluation model.

The new model called false-detected to correct-detected ratio (FCR) is recently proposed by Chen and Yang [6]. A labeling technique is used in this scheme. Each pixel is either given a single label or is not labeled. Labeled pixels are either in the actual detected edge map (DEM) or in the ideal edge map (IEM). There are five different labels, namely,

1. **True-positive** (tp): the pixel is in both DEM and IEM,

2. **True-approximate** (ta): the pixel is in DEM (but not in IEM). The pixel can be considered as an approximation of a true edge pixel in IEM. The discipline used here is that it is within one pixel distance along the gradient direction of the edge pixel in IEM,

3. **True-missing** (tm): the pixel is in IEM, but not in DEM, and there is no approximation of it.

4. **Neutral-extra** (ne): the pixel is in DEM, but is labeled neither as tp nor ta, but its one pixel distance neighbour is in IEM. These kinds of edge points make the boundaries thicker.

5. **False-positive** (fp): the pixel is in DEM, but is not labeled and has none of the above labels. These are falsely detected edges.

Let N_i denote the number of labels of type i, thus

$$N_t = N_{tp} + N_{ta} + N_{tm}. \tag{2.34}$$

$$N_f = N_{fp} + N_{tm}. \tag{2.35}$$

$$N_c = N_{tp} + a_{ta}N_{ta} + a_{ne}N_{ne}. \tag{2.36}$$

where N_t is the number of true edges in IEM, N_f, the false-detected function and N_c, the correct-detected function. a_{ta} and a_{ne} are weight coefficients ranging from -1 to 1. In the paper [6], it is considered that neutral-extra edges cause no harm and Equation (2.36) can be simplified to:

$$N_c = N_{tp} + N_{ta}. \tag{2.37}$$

Three measurements are introduced:

$$FIR = \frac{N_f}{N_t}. \tag{2.38}$$

$$CIR = \frac{N_c}{N_t}. \tag{2.39}$$

$$FCR = \frac{FIR}{CIR} = \frac{N_f}{N_c}. \tag{2.40}$$

Compared with FIR or CIR, FCR is a more comprehensive performance measurement and is used in [6]. This model is applied to synthetic checkerboard and crossboard images to choose optimal parameter settings and to compare the performance of different edge detectors. Using this model, the RCBS approach outperforms the Sobel operator and Haralick's second directional derivative operator.

Constrained by its structure, FCR cannot discriminate thick edges. The model can be further explored. Equation (2.37) can be rewritten as:

$$N_c = (N_{tp} + n) + (N_{ta} - n) = N'_{tp} + N'_{ta}. \tag{2.41}$$

It can be easily seen that an edge map with N_{tp}, N_{ta} edge pixels have the same

FCR value as an edge map with N'_{tp}, N'_{ta} edge pixels, provided both edge maps have the same N_f edge pixels. Intuitively, the second edge map should have a better measurement value because it has a higher number of true-positive edges.

All of these performance evaluation models require *a priori* knowledge of the location of the true edge, since they are based on the discrepancy between the detected edge pixels and the ideal position of the edge. For synthetic images, this can be done. But there are questions when these approaches are applied to real images. The determination of edges in real images is very much a subjective decision of a human observer. The above techniques are not applicable to images for which the true edge locations are unknown. In the next section, evaluation models which do not require *a priori* knowledge of the actual location of the edge are discussed.

2.6.2 Edge Formation Based Evaluation Models

Bryant and Bouldin [4] propose a technique called *relative grading*. Omitting the details, this approach is based on comparing the output of a number of detectors and rating each detector by how often it agrees with the consensus of the other detectors in deciding whether an edge exists at each pixel. They compared different edge detectors and quantified the improvements in edge output achieved by edge-linking and edge-thinning. The weakness of this approach is that its results highly depend on the details of the consensus determination used and on the particular mix of operators chosen for comparison. Moreover, it is completely oblivious to detection errors made by all detectors and may penalize a good detector that does not make an error made by a majority of bad detectors. This approach is consensus based and does not actually examine the edge formation. The performance of this technique is questionable.

Kitchen and Rosenfeld [18] propose an edge evaluation model that uses local edge coherence (LEC) to account for both the edge position and the edge direction information produced by an edge operator. Special attention is paid to the form of the detected edge. The criteria for a well-formed edge are continuity and thinness. Essentially, a 3×3 neighbourhood of an edge pixel is examined. The continuity and thinness scores of each edge neighbourhood are computed with a range between 0 and 1. The overall ratings (C for continuity, T for thinness) are averaged over all the edge neighbourhoods in the image. A linear combination of the two ratings is chosen to be the final measure E given by:

$$E = \gamma C + (1 - \gamma) T \tag{2.42}$$

where γ is the weighting factor with a value between 0 and 1. For each neighbourhood, the continuity score measures how well the edge gradient direction at the neighbour agrees with that at the center and how close a neighbour is to the expected direction of continuation of the edge. Thinness score is computed as the fraction of the remaining pixels (not along the edge direction) that are non-edge pixels. This evaluation scheme has been studied in this thesis and experimental results suggest that this scheme performs well on synthetic images. Compared with the FCR model, this scheme tends to choose large thresholds, since it penalizes thick edges. The effect of γ and the variation of the measure E w.r.t threshold are shown in Figure 2.6. It can be seen that there is a certain threshold in Figure 2.6 at which the evaluation measure E does not change much w.r.t. different γ. This is an undesirable "feature" of the LEC evaluation scheme, since intuitively by changing the weight between continuity and thinness, the final measure should be expected to change accordingly. In the experiments, γ is set to be 0.8, which is considered to be a reasonable combination factor between continuity and thinness.

Figure 2.7 shows the comparison of the FCR and the LEC models. (This experiment is done on the crossboard image with an edge contrast of 100. The standard deviation of the Gaussian noise in the image is 50.) It can be seen that at certain

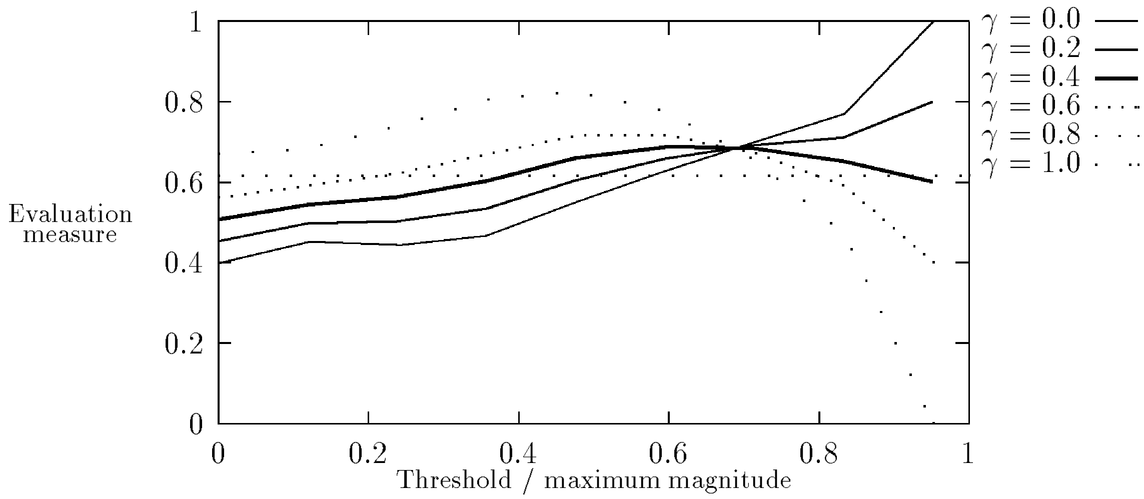


Figure 2.6: The variation of the LEC evaluation measure w.r.t. threshold with various γ on synthetic image.

ranges of threshold, better FCR values correspond to better LEC values, demonstrating that the two models agree with each other. (It should be noted that a smaller FCR value and a larger LEC value correspond to a better performance.) Between threshold values of 16 and 24, the two models do not conform. A possible reason may be that LEC penalizes thick edges. So in that range of threshold, increasing the threshold produces thinner edges, which corresponds to a better LEC value. On the other hand, the FCR model does not penalize thick edges and its edge map is compared with the *ideal* one, which itself may contain thick edges too. It can also be seen that the LEC value does not change as *sharply* as the FCR value with respect to the threshold. This is a weakness of the LEC model.

The initial goal of exploring the LEC scheme is to apply it to real images, since it does not require *a priori* knowledge of the edge locations. But when actually applied to real images, LEC does not perform as well as originally expected. Figure 2.8 shows two edge maps (generated from the output of the *tanh* operator) obtained by using a

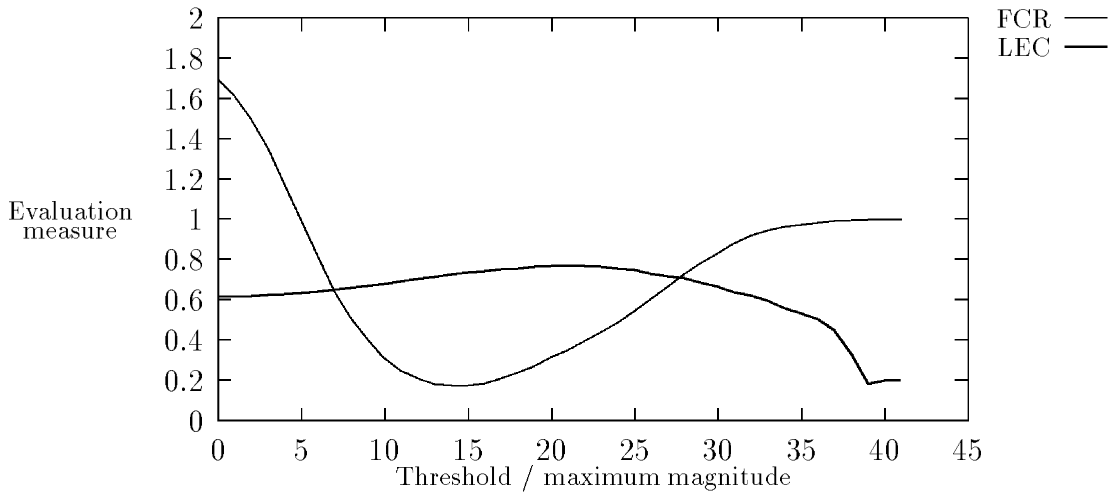


Figure 2.7: Comparison of the FCR and the LEC models.

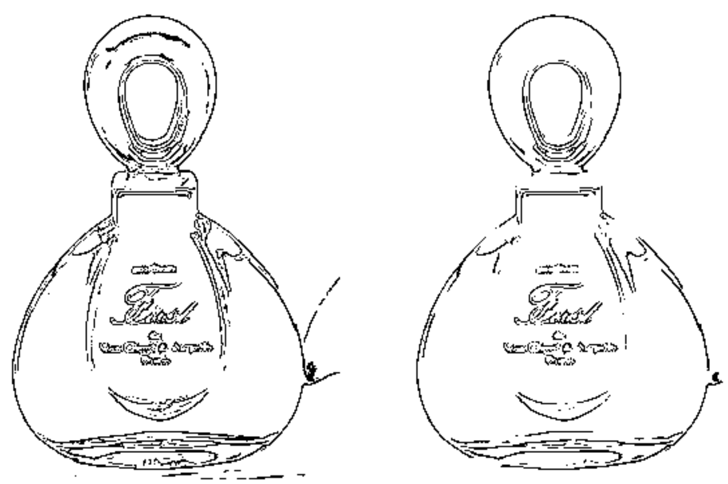


Figure 2.8: Edge maps ($tanh$): human (left) and LEC (right) selected thresholds.

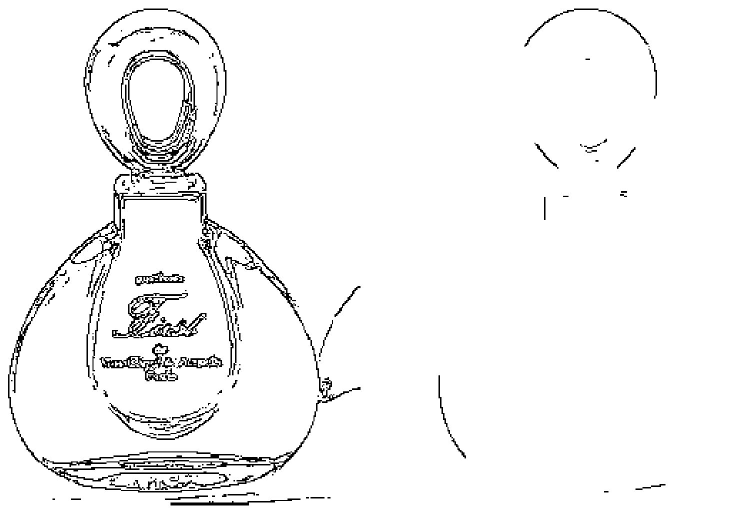


Figure 2.9: Edge maps (RCBS): human (left) and LEC (right) selected thresholds.

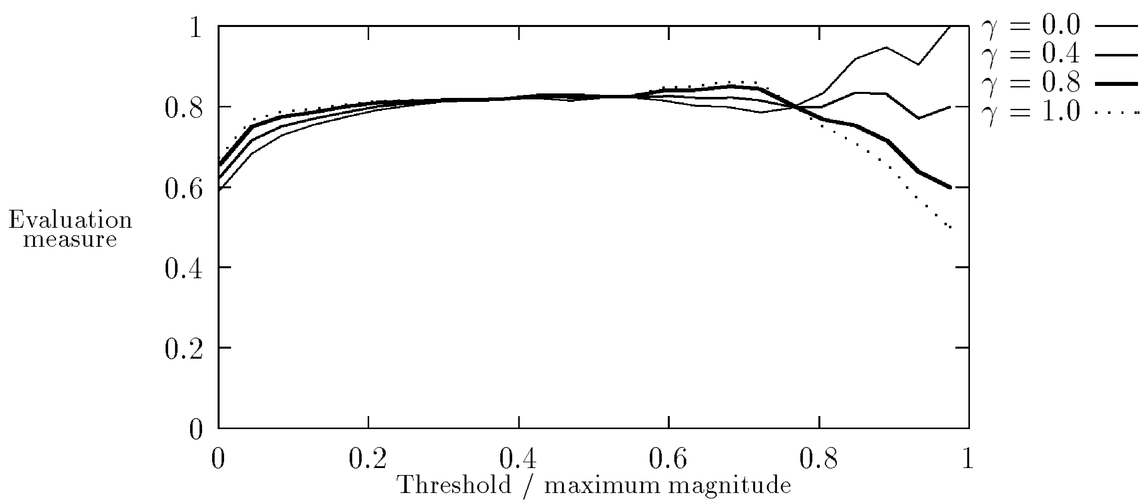


Figure 2.10: The variation of the LEC evaluation measure w.r.t. threshold with various γ on a real image.

threshold value chosen by a human and a threshold that gives the best LEC measure. Figure 2.9 shows another example, in which the gradient output of the RCBS operator is thresholded. It can be seen that in real images, the original goal of LEC is not achieved. Figure 2.10 shows the evaluation measure with respect to threshold. As can be seen visually, the quality of the edge map changes a lot using different thresholds, yet the LEC measure does not change much. This shows that the LEC evaluation model is not sensitive enough to capture the characteristics of well-formed edges in real images.

2.7 Summary

In this chapter, major approaches to edge detection are reviewed. Edge detection schemes are classified into different groups. Gradient and template filters are the earliest developed edge detectors. They are easy to implement, but do not work well in a noisy environment. The Gaussian-like filters blur the image first and then use the zero crossings of the second derivative or the local maxima of the first derivative as edge pixels. There are also regularization and surface fitting approaches that aim to achieve better noise immunity and accurate localization. Least-sum-of-squares (LSS) fitting is generally used, which is known not to be robust in the presence of outliers. A new consensus based robust edge detector is also reviewed. The weakness of this edge detector is that its breakdown point is influenced by the signal-to-noise ratio and the consensus set is a noise sensitive parameter.

Several edge evaluation models are also studied. Particularly, the FCR and the LEC models are investigated in more detail and a comparison of their performance is given. For synthetic images, both the FCR and the LEC models give similar performance. For real images, the FCR model does not apply. The LEC model does not perform well.

Chapter 3

Robust Edge Detection

This chapter presents two new robust edge detectors proposed in this thesis work. First, the motivation for using a robust fitting technique is discussed. It is followed by a brief review of the *Least-Median-of-Squares* (LMS) estimator and the random sampling technique employed to obtain the LMS solutions. An experimental study of the performance of random sampling is presented in Section 3.3. LMS employing random sampling is then incorporated in the $tanh$ edge detector in Section 3.4. This section also presents a new robust edge detector using the robust location estimator. Quantitative performance comparisons of different detectors are given in Section 3.5 where the FCR evaluation model is used. The summary of the chapter is given in Section 3.6.

3.1 Robust Fitting Technique

As discussed in Chapter 2, the surface fitting approach has been considered to be an effective way to detect edges. It can detect local features as well as suppress the noise. Most of the previous experiments in published work were done assuming that the noise has a Gaussian distribution, which the surface fitting approach can effectively suppress using the well-known LSS technique. In the presence of outliers, however, the performance of this kind of detector degrades significantly. When there are a large number of outliers, the true edges are hardly detected.

Robust statistics provide methods for extracting measurements from unco-operative data and can be used in computer vision algorithms to make the algorithms more robust to various kinds of noise. While robust techniques have attracted much attention in recent years, there has not been much concrete work done in making edge detectors more robust. This thesis work explores the application of robust statistics to edge detection. The *least-median-of-squares* (LMS) estimator in robust statistics has a breakdown point of 50%, which means it allows up to 50% of the data points to be corrupted.

The present objective is to make edge detectors robust to outliers using the LMS technique and to evaluate the performance of edge detectors on both synthetic and real images.

Section 3.2 gives a brief review of the LMS estimator. This estimator is proposed to be incorporated in edge detection to make detectors robust to outliers. By using the LMS robust fitting technique, outliers can be recognized and can then be discarded when doing further LSS fitting. Thus the accuracy of LSS fitting will not be greatly influenced under the existence of outliers. This is the idea behind the robust $tanh$ fitting approach which is discussed in Section 3.4.2. The motivation to develop a fast edge detector with high breakdown point leads to a new operator which is based on LMS estimation with only one coefficient to be determined. This new operator is presented in Section 3.4.3. These two robust edge detectors together with three other surface fitting approaches, namely, Haralick's second directional derivative, regularized cubic B-Spline (RCBS) and the $tanh$ operator, are compared quantitatively in Section 3.5 using the false-detected to correct-detected ratio (FCR) model.

3.2 Least-Median-of-Squares Estimator

3.2.1 Properties of LMS Estimator

In order to have a better understanding of the LMS estimator, this section presents its properties more formally. The mathematical details are not shown here, since the application of robust regression is the primary interest. (Most of the proofs can be found in [32].)

The classical linear regression problem is to find the coefficients $\hat{\theta} = [\theta_1, \ldots, \theta_p]^T$ in the following model:

$$y_i = x_{i_1}\theta_1 + \ldots + x_{i_p}\theta_p + e_i \qquad \text{for } i = 1, \ldots, n \tag{3.1}$$

where n is the sample size. $x_{i_1}, \ldots, x_{i_p}$ are called *explanatory variables*, y_i is the *response variable*, and e_i, the error term. In classical theory, e_i is assumed to be normally distributed with mean zero and unknown standard deviation σ. In robust statistics, there is no underlying model for the error (noise) e_i. It may contain well-behaved Gaussian noise as well as outliers.

Definition 1. *Breakdown point*

Take any sample of n data points,

$$Z = (x_{11}, \ldots, x_{1p}, y_1), \ldots, (x_{n1}, \ldots, x_{np}, y_n) \tag{3.2}$$

and let T be a regression estimator. The regression coefficients obtained when applying T to sample Z are denoted by:

$$T(Z) = \hat{\theta}. \tag{3.3}$$

Consider all possible corrupted samples Z' that are obtained by replacing any m of the original data points by arbitrary values. bias$(m; T, Z)$ denotes the maximum bias that can be caused by such a contamination:

$$\text{bias}(m; T, Z) = \sup_{Z'} \parallel T(Z') - T(Z) \parallel . \tag{3.4}$$

If bias$(m; T, Z)$ is infinite, this means that m outliers can have an arbitrarily large effect on T, which is expressed by saying that the estimator "breaks down". Therefore, the *breakdown point* of the estimator T at sample Z is defined as

$$\epsilon_n^*(T, Z) = \min \{\frac{m}{n}; \text{bias}(m; T, Z) \text{ is infinite}\}. \tag{3.5}$$

In other words, the breakdown point is the smallest fraction of contamination that can cause the estimator T to take on values arbitrarily far from $T(Z)$.

Definition 2. *Scale equivariant*

An estimator T is said to be *scale equivariant*, if

$$T((\mathbf{x_i}, cy_i); i = 1, \ \dots \ , n) = cT((\mathbf{x_i}, y_i); i = 1, \ \dots \ , n) \tag{3.6}$$

for any constant c. It implies that the fit is essentially independent of the choice of measurement unit for the response variable y.

Definition 3. *Affine equivariant*

An estimator T is said to be *affine equivariant* if

$$T((\mathbf{x_i A}, y_i); i = 1, \ \dots \ , n) = \mathbf{A^{-1}} T((\mathbf{x_i}, y_i); i = 1, \ \dots \ , n) \tag{3.7}$$

for any nonsingular square matrix $\mathbf{A}$. In other words, affine equivariance means that a linear transformation of $\mathbf{x_i}$ should transform the estimator T accordingly, because $\hat{y}_i = \mathbf{x_i} T = (\mathbf{x_i A})(\mathbf{A^{-1}} T)$. This allows the use of another coordinate system for the explanatory variables, without affecting the estimated $\hat{y}_i$.

The conventional *least-sum-of-squares* (LSS) estimator, given by

$$\text{Minimize}_{\hat{\theta}} \sum r_i^2 \tag{3.8}$$

has a breakdown point equal to

$$\epsilon_n^*(T, Z) = 1/n \tag{3.9}$$

which tends to zero for increasing sample size n. So it can be said that LSS approaches a breakdown point of 0% when $n \rightarrow \infty$.

The *Least-Median-of-Squares* (LMS) estimator is given by

$$\text{Minimize}_{\hat{\theta}} \text{ Med } r_i^2, \tag{3.10}$$

which replaces the sum in LSS by a median. The LMS estimator has the following properties [32]:

Theorem 1. There always exists a solution to (3.10) with respect to the linear regression model (3.1).

Theorem 2. The LMS estimator (3.10) is scale equivariant and affine equivariant with respect to the linear regression model (3.1).

Theorem 3. If $p > 1$ and the observations are in general position, then the breakdown point of the LMS method (3.10) with respect to the linear regression model (3.1) is

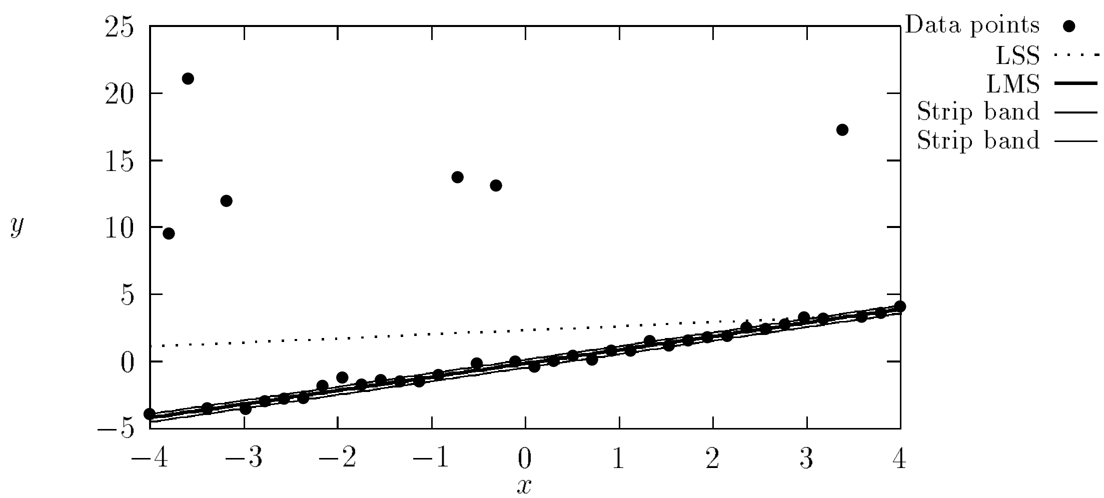
$$([n/2] - p + 2)/n \tag{3.11}$$

which tends to 50% as $n \rightarrow \infty$. This is the highest breakdown point that can be attained.

To get a geometrical view of the LMS estimator, consider the LMS solution for simple regression with intercept, given by

$$\text{Minimize}_{\hat{\theta}_1,\hat{\theta}_2}\ \text{Med}_i\ (y_i - \hat{\theta}_1 x_i - \hat{\theta}_2)^2. \tag{3.12}$$

This is to fit a line in the (x, y) coordinate system. Geometrically, LMS corresponds to *finding the narrowest strip covering half of the observations.* (To be precise, by "half" it means $[n/2]+1$, where $[n/2]$ denotes the integer part of $n/2$. Moreover, the thickness of this strip is measured in the vertical direction.) The LMS solution is the straight line that lies exactly at the middle of this band (Figure 3.1).



The dotted line corresponds to the LSS fit. The solid LMS line is surrounded by the narrowest strip containing half of the data points.

Figure 3.1: LMS and LSS solution for simple regression with intercept.

3.2.2 Location Estimator

While LMS has many good properties, the computation of the regression coefficients is not obvious at all. Only for the location estimator, where $p = 1$ and the underlying model is given by

$$Y_i = \theta + e_i, \tag{3.13}$$

is there a simple way to compute the LMS solution. Suppose $\{y_1, y_2, \ \ldots, y_n\}$ are observed data points and $y_{1:n} \leq y_{2:n} \leq \ \ldots \leq y_{n:n}$ are the ordered observations. To compute the LMS solution in the location case, one only has to determine the *shortest half* of the sample. This is done by finding the smallest of the differences

$$y_{h+i-1:n} - y_{i:n} = \text{Min}\ \{y_{h:n} - y_{1:n},\ y_{h+1:n} - y_{2:n},\ \ldots, , y_{n:n} - y_{n-h+1:n}\}, \tag{3.14}$$

where $h = [n/2] + 1$. The corresponding subsample $y_{i:n}, \ \ldots, y_{h+i-1:n}$ containing h observations is then called the shortest half because of all the possible subsets with h elements it possesses the shortest range. The LMS solution simply equals the midpoint of this shortest interval [32]. To illustrate the LMS estimate in the location case, consider the following sample data points:

$$90,\ 93,\ 86,\ 92,\ 95,\ 83,\ 75,\ 40,\ 88,\ 80.$$

Order the observations as in Figure 3.2. The horizontal lines indicate the halves of the sample. Consider the differences defined in (3.14), yielding the values 48, 15, 12, 10 and 9. The smallest of them corresponds to the distance between 86 and 95. This is the lower and upper bound of the strip (band). Finally, the midpoint of this interval, namely, 90.5, is the LMS estimate of location. Notice that the presence of the aberrant value 40 does not affect this result.

The complexity of finding the location estimate depends on the algorithm used

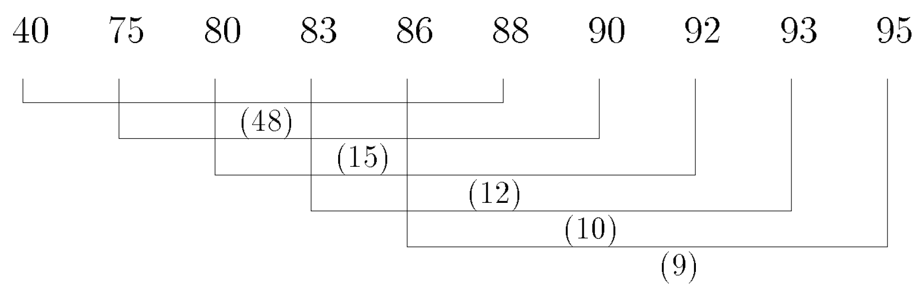


Figure 3.2: Finding the shortest half of the sample.

to sort the data points. If bubble sort is used, the complexity will be $O(n^2)$. A more efficient algorithm can also be used. Except for the location estimator, there is no straightforward formula for the LMS solution, unlike for the LSS estimator. The following section describes the random sampling technique used in this research work to find an approximate LMS solution.

3.2.3 Random Sampling Technique

Suppose there are p coefficients to be determined and n data observations available, as denoted in Equation (3.1). The random sampling technique [32] proceeds by repeatedly drawing subsamples of p different observations. For such a subsample, indexed by $\mathbf{J} = \{(i_0, \ \ldots \ , i_{p-1})\}$, one can determine the regression surface through the p points and denote the corresponding vector of coefficients by $\theta_{\mathbf{J}}$. This will be a solution of a system of p linear equations in p unknowns. Such a solution $\theta_{\mathbf{J}}$ is called a trial estimate. For each $\theta_{\mathbf{J}}$, the corresponding LMS objective function with respect to the whole data set is also calculated:

$$\text{med}_{i=1, \ \ldots \ ,n}(y_i - \mathbf{x_i}\theta_{\mathbf{J}})^2 \tag{3.15}$$

For all trial estimates, the one that yields the smallest value of the objective function

p	5%	10%	20%	25%	30%	40%	50%
1	1	2	2	3	3	4	5
2	2	2	3	4	5	7	11
3	3	3	5	6	8	13	23
4	2	3	6	8	11	22	47
5	3	4	8	12	17	38	95
6	3	4	10	16	24	63	191
7	3	5	13	21	35	106	382
8	3	6	17	29	51	177	766
9	4	7	21	36	73	296	1533

Table 3.1: The number of random samples m, determined by p and ϵ.

(3.15) will be considered to be an LMS solution of Equation (3.1).

The problem is how many subsamples $J_1, J_2, \ldots, J_m$ should be considered. In principle, one could repeat the procedure for all possible subsamples of size p, of which there are C_n^p. It is well-known that C_n^p increases very fast with n and p. A feasible way is to perform a certain number of random selections such that the probability that at least one of the m subsamples is "good" is almost 1. A subsample is "good" if it consists of p good observations of the sample. The sample may contain up to a fraction ϵ of bad observations. The expression for this probability, assuming that n/p is large, is

$$1 - \{1 - (1 - \epsilon)^p\}^m. \tag{3.16}$$

By requiring that this probability must be near 1 (say, at least 0.95 or more), one can determine m for given values of p and ϵ. This has been done in Table 3.1 for a percentage of contamination varying between 5% and 50%. For fixed p, the number of subsamples increases with ϵ in order to maintain the same probability (3.16).

In this research work, the value of m for small p is taken systematically larger than the value needed in formula (3.16) in order to increase the expected number of good subsamples. Therefore, the objective function (3.15) may become even smaller. m is computed by

$$m = \begin{cases} C_n^p, & \text{if } C_n^p < 500 \\ 500, & \text{otherwise.} \end{cases} \tag{3.17}$$

3.3 Performance of Random Sampling

3.3.1 Models and Parameter Settings

Since low order polynomials are often used as basic fitting functions in edge detection, the LMS estimator employing the random sampling technique is first applied to fit polynomials of different order. The underlying model is written as

$$y = a_{p-1}x^{p-1} + a_{p-2}x^{p-2} + \ldots + a_0 + e \tag{3.18}$$

where the error term e contains both Gaussian noise of zero mean with standard deviation of σ and error due to outliers. It is noteworthy that this is not a linear regression with respect to x, except for $p \leq 2$. When the LMS method is used, the estimator is scale equivariant, but not affine equivariant.

For the LSS method, to minimize the objective function (3.8), partial derivatives with respect to a_i are set to zero. Supposing $k = p - 1$, a_i's are obtained by solving the following equations:

$$\begin{cases} \sum x_i^{2k} a_k & + & \sum x_i^{2k-1} a_{k-1} & + & \ldots & + & \sum x_i^k a_0 & = & \sum y_i x_i^k \\ \sum x_i^{2k-1} a_k & + & \sum x_i^{2k-2} a_{k-1} & + & \ldots & + & \sum x_i^{k-1} a_0 & = & \sum y_i x_i^{k-1} \\ \vdots \\ \sum x_i^k a_k & + & \sum x_i^{k-1} a_{k-1} & + & \ldots & + & n \cdot a_0 & = & \sum y_i \end{cases} \tag{3.19}$$

For the LMS estimator, when a subsample of p data observations is chosen, the trial estimate of a_i's is the solution of

$$\begin{cases} x_{i_0}^k a_k & + & x_{i_0}^{k-1} a_{k-1} & + & \dots & + & a_0 & = & y_{i_0} \\ x_{i_1}^k a_k & + & x_{i_1}^{k-1} a_{k-1} & + & \dots & + & a_0 & = & y_{i_1} \\ \vdots & & & & & & & & \\ x_{i_{p-1}}^k a_k & + & x_{i_{p-1}}^{k-1} a_{k-1} & + & \dots & + & a_0 & = & y_{i_{p-1}} \end{cases} \tag{3.20}$$

The sampling is done repeatedly and the final result is the trial estimate with which the objective function (3.10) is minimized.

Let $\hat{a}_i,\ i = 0,\ \dots,\ p-1$, denote the estimated coefficients. At each data point $x_i,\ i = 1,\ \dots,\ n$, the true value y_i^t is computed using the underlying model:

$$y_i^t = a_{p-1} x_i^{p-1} + a_{p-2} x_i^{p-2} + \dots + a_0. \tag{3.21}$$

The estimated value $\hat{y}_i$ is obtained by:

$$\hat{y}_i = \hat{a_{p-1}} x_i^{p-1} + \hat{a_{p-2}} x_i^{p-2} + \dots + \hat{a}_0. \tag{3.22}$$

The evaluation measurement is the maximum difference between the true value and the estimated value, which is:

$$\mathrm{Err} = \mathrm{Max}_{i=1,\ \dots\ ,n} \mid \hat{y}_i - y_i^t \mid . \tag{3.23}$$

By using the LSS and LMS methods, two error measurements can be obtained. Smaller Err shows a better estimation of the coefficients.

The parameters involved in the experiments and their settings are summarized as follows.

1. p: the number of coefficients. $2 \le p \le 4$. (Polynomials up to degree three are considered here.)

2. n: the number of data points. $p \leq n$. The number of data points should not be less than the number of coefficients.

3. ϵ: the fraction of outliers contained in the sample data. $0\% \leq \epsilon \leq 60\%$. Up to 60% fraction of outliers are tested.

4. σ: the standard deviation of Gaussian noise. $0.0 \leq \sigma \leq 2.0$.

5. m: the number of subsamples when employing the random sampling technique to get the LMS solution. m is chosen using Equation (3.17).

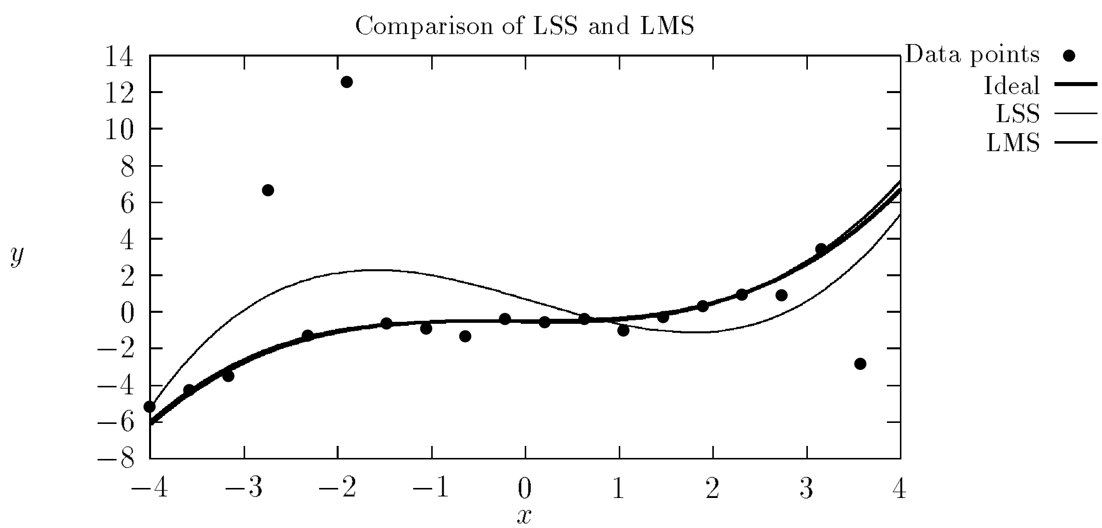
In this work, the data points are generated by adding Gaussian noise on the *clean* data. Then a fraction ϵ of the data points are randomly chosen to be outliers. The y value of outliers is set to be a randomly generated number scaled by a factor of 3. The experimental results are shown and analyzed in the following section.

3.3.2 Experimental Results

This section gives an analysis of the performance of LSS and LMS on fitting a polynomial of degree three. The experimental results of other orders of polynomials can be found in Appendix A. Similar observations can be obtained.

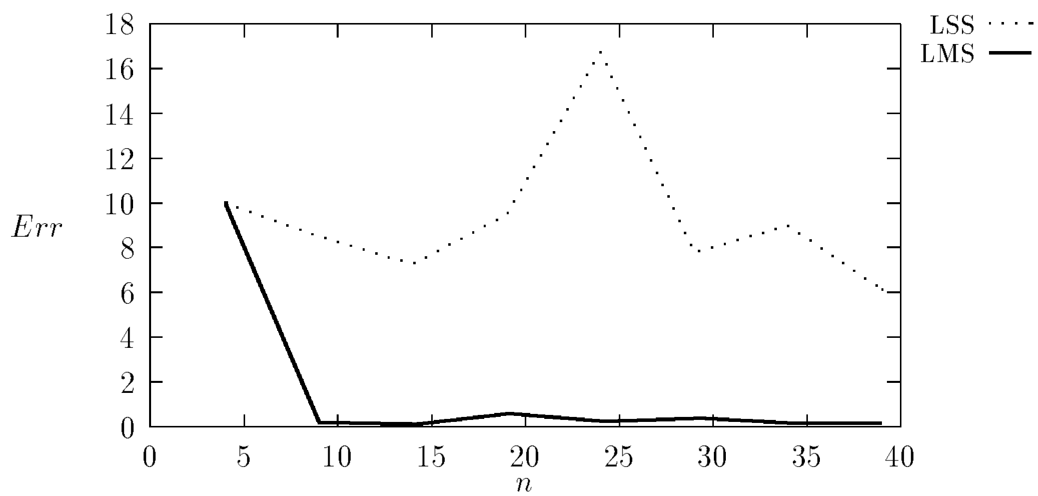
The underlying polynomial is chosen to be $y = 0.1x^3 + 0.05x^2 - 0.5$. The fitted curves are shown in Figure 3.3, where the Gaussian noise level of y is 0.3, the fraction of outliers in y is 20% and there are 20 data points. It can be clearly seen that LMS follows the ideal curve closely, whereas LSS deviates far from the ideal one. Figures 3.4, 3.5 and 3.6 show the quantitative analysis. When analyzing the figures, it should be noted that the Gaussian noise and the outliers are all chosen randomly. Each run of the experiment may not have exactly the same results, but the systematic trend is the same.

In Figure 3.4, when n is equal to p, there are just enough data points to do the fitting. The LMS degenerates to LSS and the error measurements are the same. When



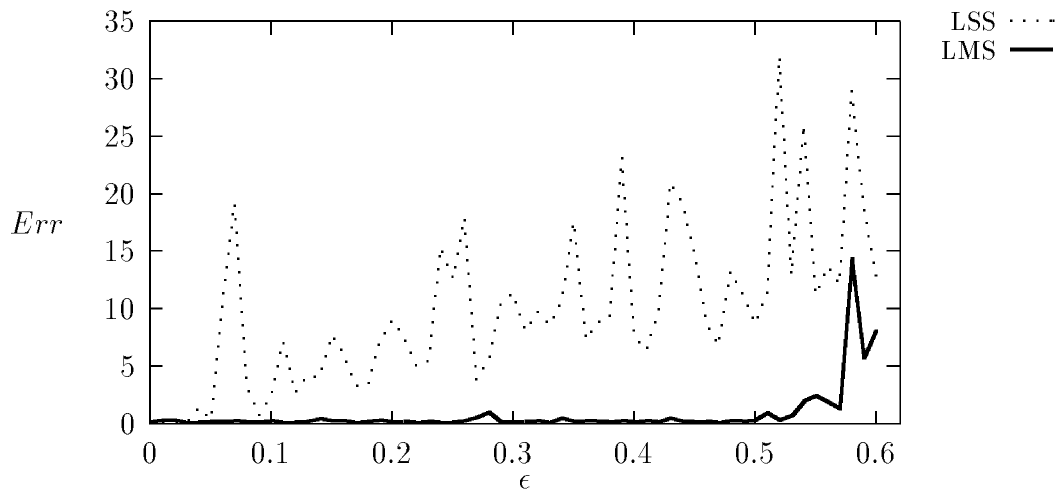
$y = 0.1x^3 + 0.05x^2 - 0.5,\ n = 20,\ \sigma = 0.3,\ \epsilon = 20\%$

Figure 3.3: Fitted curves using LSS and LMS ($p = 4$).



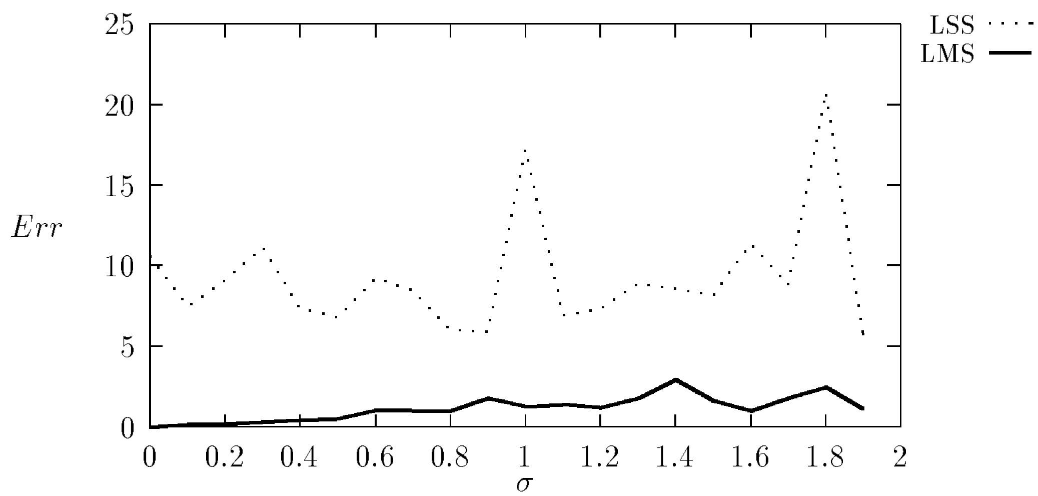
$y = 0.1x^3 + 0.05x^2 - 0.5,\ \sigma = 0.1,\ \epsilon = 40\%$

Figure 3.4: Comparison of LSS and LMS with Err vs. n ($p = 4$).



$$y = 0.1x^3 + 0.05x^2 - 0.5,\ n = 30,\ \sigma = 0.1$$

Figure 3.5: Comparison of LSS and LMS with Err vs. ϵ ($p = 4$).



$$y = 0.1x^3 + 0.05x^2 - 0.5,\ n = 30,\ \epsilon = 30\%$$

Figure 3.6: Comparison of LSS and LMS with Err vs. σ ($p = 4$).

n increases, LMS shows its advantage. Through random sampling, it can eliminate the outliers and find the fit using the "good" data points. The error decreases when n is larger than p and the error of LMS is significantly less than that of LSS. One may suspect the reliability of LMS when n is large, since only m (defined by (3.17)) subsamples are chosen to compute the trial estimate. Figure 3.4 shows that random sampling actually works very well even if n is large. In this example, $C_{20}^4 = 4845$ and the fraction of outliers is 40%. Though random sampling is only done 500 times, the resultant error is still very small.

Figure 3.5 shows the relationship between the error measure and the fraction of outliers ϵ. When there are no outliers, the performance of LSS and LMS is about the same. As the fraction of outliers increases, the error of LSS increases. (One should pay more attention to the general upward trend of the curve. The "bumps" in the figures are mostly due to the random selection of outliers across different runs.) For LMS, as long as the fraction of outliers is less than 50%, the error does not have a tendency to increase with ϵ and is always within a small range. When $\epsilon > 50\%$, the LMS error obviously grows, which reinforces the statement that LMS has a breakdown point of 50% as discussed above.

In Figure 3.6, the Gaussian noise level is varied while keeping the outlier fraction at 30%. When there is no Gaussian noise, the LMS error is zero. Meanwhile, LSS has a high error value, because the presence of outliers ruins the LSS estimate. When the Gaussian noise level increases, the LMS error also increases. This is because the so-called "good" subsamples themselves are not *clean* data points, so the exact fit of these data points results in errors. As already stated, when the error has a Gaussian distribution, LSS is optimal. This leads to the idea of doing a LSS fit on the "good" data samples after recognizing outliers using the LMS method. This is studied in detail in Section 3.4.2.

3.4 Robust Edge Detection

3.4.1 Apply LMS Technique in Edge Detection

It has been shown that LMS fitting performs much better than LSS fitting in the presence of outliers. How to incorporate the LMS fitting technique into conventional surface fitting approaches is another issue. Three surface fitting approaches are considered here.

1. **Haralick's second directional derivative operator:**
   Due to the structure of the algorithm, the polynomial fitting is done on the 2-dimensional symmetric row-col index set. In order to do the fitting, all data points in the window need to be present. The idea behind LMS fitting is to recognize and then discard the outliers. Hence, LMS fitting is not directly applicable to the Haralick operator.

2. **Regularized Cubic B-Spline (RCBS) operator:**
   In the RCBS operator, there is more than one fitting coefficient to be computed. Thus LMS employing the random sampling technique may be applicable. The basic requirement of using random sampling is that the number of data points is more than the number of fitting coefficients. Otherwise, random sampling will not be effective. There are two possible ways to incorporate LMS in the RCBS operator. The first is to use scaled mapping (SM) to provide more data points [6]. The LMS method can then be applied to the 1-D cubic B-spline fitting. However, the RCBS model is not a linear regression model but a rather complicated nonlinear one. The maximum allowed fraction of outliers (ϵ) can be intuitively computed by:

$$\epsilon = \frac{M - V}{M} \tag{3.24}$$

   where M is the number of data points, and V, the number of coefficients.

IPV	M	ϵ
427	9	22.2%
629	13	30.7%
82b	17	35.3%

* $M = P\ I + 1$
* b in 82b stands for 11 in hexadecimal.

Table 3.2: Maximum fraction of outliers in SM in the RCBS operator.

Equation (3.24) is based on the idea of random sampling. To guarantee that at least one subsample contains "good" data points, there can be at most $M - V$ outliers.

Table 3.2 gives the maximum fraction of outliers for different parameter settings in SM in the RCBS operator, where IPV represents parameters I, P and V. (P is a parameter in SM, where each subinterval is mapped to a small window with P data points in it.) It can be seen that ϵ is only 35.3% even when using a very large operator of size 17. In fact, when such a large operator is used, the performance of the RCBS operator already degrades. Furthermore, the M data points are resampled by projecting the 2-D data along the direction perpendicular to the gradient direction. The effect of the outliers on the resampled data is hard to measure.

An alternative way of robustifying the RCBS operator is to use the LMS method to estimate the direction of the gradient and to recognize outliers. When doing the 1-D projection, the outliers are ignored. This is supposed to make the resampled data points robust to outliers. But, in fact, when projecting two-dimensional data into one dimension to get the resampled data, there may be missing points because of the existence of outliers. When the nodes of the operator grid do not coincide with the nodes of the image grid, the original bilinear interpolation cannot be performed. Though the weight of each non-outlier image grid node can be reassigned according to its distance to the operator grid, the result does not turn out to be satisfactory. Possible reasons may lie in the resampling procedure.

3. $Tanh$ **operator:**

   The steps involved in the $tanh$ operator are summarized in Section 2.4.2. Basically, it first tries to find a reliable estimate of the gradient direction. Then the data points are fitted with a 1-D $tanh$ surface along that direction. Experiments show that the accuracy of the $tanh$ operator depends highly on the estimate of gradient direction. Both the planar-fit and the cubic-fit are necessary to obtain the direction θ. Experiments have shown that if the refinement process (cubic-fit) is omitted or the planar fit is replaced by a horizontal and vertical mask to give an initial estimate of θ, the performance of the $tanh$ operator greatly degrades. This shows that the gradient direction estimation is a critical procedure in the $tanh$ operator. To make the $tanh$ operator robust to outliers, an essential step is to make the estimation of θ robust to outliers.

   Recall that the first step in the $tanh$ operator is to do the following least-sum-of-squares planar-fit:

$$\text{Min } \xi_p = \text{Min } \sum_{x,y} (I_{(x,y)} - (a_0 + a_1 x + a_2 y))^2 \tag{3.25}$$

   which will inevitably be sensitive to outliers. In the proposed robust approach, the "sum" operator is replaced by the median operator and LMS employing random sampling technique is used to do the fitting. Outliers can be recognized by the robust fitting. Then the LSS planar-fit is done on the rest of the data points, which are considered to be "good" samples. This new robust version of the $tanh$ operator performs very well in the presence of outliers compared with other surface fitting approaches. The following section presents a more detailed discussion of this robust *tanh* operator.

3.4.2 Robust *Tanh* Fitting Approach

There are two major differences between the new robust *tanh* operator and the original *tanh* operator. First, the robust version uses LMS to do the planar fit and estimate the gradient direction instead of using the LSS planar fit. Second, in the new version, after the LMS fit, data points are classified and labeled as either good data points or outliers. Further fittings are done on good data points only. The algorithm is summarized as follows.

1. *LMS Planar-Fit:*

$$\text{Minimize}_{a_0,a_1,a_2}\ \text{Med}_{x,y}\ (I_{(x,y)} - (a_0 + a_1 x + a_2 y))^2 \tag{3.26}$$

   The random sampling technique is used to get the LMS solution. This is a linear regression model with three unknown coefficients ($p = 3$). The window size used in the *tanh* operator is 5×5. There are 25 data points altogether ($n = 25$). It should be noted that the number of data points in this case is not sufficiently large, so the breakdown point of LMS will not be 50%. The actual breakdown point is 44%, which can be obtained by using Equation (3.11).

   The number of subsamples (m) is determined by Equation (3.17). After random sampling, m trial estimates $\{a_i^{(j)},\ \text{for } i = 0,\ 1,\ 2\ \text{ and }\ j = 1,\ \ldots,\ m\}$ have been computed. A new estimate can be obtained by:

$$a_i^{(new)} = \text{Med}_j\ a_i^{(j)}, \qquad \text{for } i = 0,\ 1,\ 2. \tag{3.27}$$

The residue at each data point is computed by:

$$r_k^2 = (I_k - (a_0 + a_1 x + a_2 y))^2, \qquad \text{for } k = 1, \ \ldots, \ n. \tag{3.28}$$

A robust estimate of the dispersion of the residuals is given by

$$\hat{\sigma}^2 = \left(1.4826 \left(1 + \frac{5}{n-p}\right)\right)^2 \ \mathrm{Med}_k \ r_k^2 \tag{3.29}$$

[32]. In the LMS planar fit, $n = 25, \ p = 3$, thus,

$$\hat{\sigma}^2 \approx 4.0 \ \mathrm{Med}_k \ r_k^2 \tag{3.30}$$

In the experiments, if $r_i^2 \ > \ \hat{\sigma}^2$, then the data point i is labeled as an outlier. All the following fits are similar to the conventional fits, except that they are done only on data points that are not outliers.

2. *LSS Planar-Fit:*

   The objective function is (3.25). The initial estimate is obtained by Equation (2.21). Since the outliers have been discarded, the initial estimate is more reliable.

3. *1-D LSS Cubic-Fit:*

   Cubic fit (2.22) aims to refine the gradient direction θ by using Newton method.

4. *1-D LSS Tanh-Fit Along θ:*

   Tanh fit (2.24) is obtained by iteratively refining the displacement p using Newton method.

As shown in Section 2.4.2, the comparison of the error of the *tanh* fit with that of the quadratic fit is considered to be inappropriate in this work. The final edge map is obtained by thresholding on the contrast.

There are several implementation and computational aspects that need to be addressed.

- The relative variance V of the intensity levels in the fitted window is first computed. Let $\bar{I}$ denote the average intensity in the window. V is obtained by:

$$V = \frac{\sqrt{\frac{1}{n}\sum_{x,y}\left(I_{(x,y)} - \bar{I}\right)^2}}{\bar{I}}, \quad \text{if } \bar{I} \neq 0. \tag{3.31}$$

  ($\bar{I} = 0$, if and only if $I_{(x,y)} = 0$, for all of the pixels in the window. In this case, the center pixel is definitely not an edge point.) If the variance is less than a pre-selected threshold, the center pixel is declared as a non-edge pixel. This serves as a pre-processing step aiming to speed up the detection procedure. The threshold is not sensitive and is chosen to be 10% in this thesis.

- When using Newton method, the convergence is not guaranteed. A limit on the number of iterations is used as a guard. In this thesis, a limit of 10 iterations is used.

- When doing the $tanh$ fitting, the absolute value of the displacement p is controlled to be less than a threshold, which is chosen to be 0.71 ($\sqrt{2}/2$) (half of the diagonal of a unit square). When p is larger than the threshold in the refinement process, Newton method is stopped and the center pixel is declared as a non-edge pixel. An intuitive reason for this is that with larger p, the actual edge pixel will lie in the center of another window. The pixel will be considered when the window moves there. This is also crucial in the implementation, since with large p the coefficient matrix of the fitting equations may become nearly singular and the execution is abandoned.

- The resultant robust algorithm turns out to be computationally excessive. For each data point, 500 planar-fits are computed in order to get the LMS solution.

Depending on the image, it may take more than one hour to process an image of size 128×128 on a Sun SPARC 10 workstation. This is a tradeoff between efficiency and the robustness of an algorithm.

3.4.3 New Robust Edge Detector

As stated in the previous section, the robust $tanh$ fitting approach involves the computation of the LMS solution of the planar fit which has three parameters to be determined. The random sampling technique is used to obtain the approximate LMS solution, which is inevitably a computationally complex procedure. A new objective is to develop a fast robust edge detector that still has a high breakdown point.

The *Least-Median-Square* location estimator deserves more attention since its solution is easy to compute (Section 3.2.2). Consider the step edge model which is the most dominant edge type. An ideal step edge of $45°$ in a 5×5 window is shown in Figure 3.7. The edge direction and the gradient direction are perpendicular to each other. The data points in the window are divided into two groups along the edge direction. Data points in the same group have the same gray level. This leads to the idea of using the LMS location estimator in each group. Now consider a step edge corrupted by noise. The gradient direction (thus the edge direction) can first be estimated using a pair of horizontal and vertical masks. Then the data points are classified to be in the left or right group depending on their locations w.r.t. to the edge direction. For each group, the LMS location estimator is applied to determine the band of the shortest half. Data points outside the band are labeled as outliers. After all the data points are labeled, the conventional $tanh$ fitting is done on non-outlier data points. The output of the $tanh$ fitting is the estimated edge contrast. The final edge map is obtained by thresholding the resultant contrast map. The following is a detailed discussion of this new robust edge detector.

edge direction

200	200	200	200	50
200	200	200	50	50
200	200	50	50	50
200	50	50	50	50
50	50	50	50	50

gradient direction

Figure 3.7: Ideal step edge of 45° in a 5×5 window.

1. Estimation of gradient direction:

   The gradient direction can be first estimated by using masks. (The edge direction is perpendicular to the gradient direction.) Figure 3.8 shows two sets of masks of size 5 and 3. The image is convolved with the horizontal and vertical mask in one set. If H_{grad} and V_{grad} denote the output of the convolution, respectively, the gradient direction is determined by

$$\theta_{grad} = tanh^{-1} \frac{V_{grad}}{H_{grad}}. \tag{3.32}$$

The size of the mask should not be too large, since the presence of an edge is a local feature. On the other hand, it should not be too small, since a very small window will be sensitive to noise. The effect of the size of the mask is studied in the next section. It is shown that a mask of size 5 performs better than that of size 3. In the $tanh$ fitting approach, there is a cubic fitting step to refine the estimate of gradient direction. Whether this step is necessary or not in this new robust edge detector is also studied in the next section.

$$H = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad V = \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$$

(a) Horizontal and vertical masks of size 3.

$$H = \begin{matrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{matrix} \qquad V = \begin{matrix} 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 2 & 2 \end{matrix}$$

(a) Horizontal and vertical masks of size 5.

Figure 3.8: Masks used to estimate the gradient direction.

2. Data Points Labeling:

   After the edge direction has been estimated, the data points can be divided into two groups.

$$Right = \{(x, y) | y > tan\ \theta_{edge}\ *\ x\} \tag{3.33}$$

$$Left = \{(x, y) | y \leq tan\ \theta_{edge}\ *\ x\} \tag{3.34}$$

   The LMS location estimator is then applied to these two groups. The result of the location estimator is a band covering half of the data points in the group. A point is labeled as "good" if it is in the band; otherwise, it is labeled as an outlier.

3. $Tanh$ Fitting:

   Though the middle of the band of each data group can be taken as the estimation of the gray level on the two sides of a potential edge, it is considered better to do a $tanh$ fitting on the good data points. First, the data points are often corrupted by Gaussian noise. The LSS $tanh$ fitting can suppress Gaussian noise, fit the

213	204	221	...	91	86	95
247	194	163	...	98	80	100
...	...	...	...	...	...	...
212	198	178	...	108	89	98
189	211	203	...	107	79	86

(a) Original image

213	213	213	204	221	...	91	86	95	95	95
213	213	213	204	221	...	91	86	95	95	95
213	213	213	204	221	...	91	86	95	95	95
247	247	247	194	163	...	98	80	100	100	100
...	...	...	...	...	...	...	...	...	...	...
212	212	212	198	178	...	108	89	98	98	98
189	189	189	211	203	...	107	79	86	86	86
189	189	189	211	203	...	107	79	86	86	86
189	189	189	211	203	...	107	79	86	86	86

(b) Padded image

Figure 3.9: Padding an image (Window size = 5, RADIUS = 2).

data better and result in a more accurate estimation of the contrast of an edge. Second, in the $tanh$ fitting, the absolute value of displacement p is controlled by a threshold ($\sqrt{2}/2$). If p is larger than the threshold in the refinement process of doing the $tanh$ fitting, the center pixel of the window is not considered to be an edge point candidate. This will help to eliminate thick edges.

The above steps are applied to each image pixel, and the window size to do the labeling and fitting is taken to be 5. In order to examine the border pixels, the image is padded first as shown in Figure 3.9.

After the $tanh$ fitting, each pixel is either declared as a non-edge pixel or is associated with the estimated contrast of an edge. The resultant file is called a contrast map. The edge map is obtained by applying the thresholding technique on the contrast map. In the next section, the performance of this new edge detector is studied. The quantitative comparison of several detectors is also given.

3.5 Performance Evaluation

3.5.1 Synthetic Images

Two new robust edge detectors using LMS techniques are proposed in the previous sections. To study the performance of edge detectors, they are first used on synthetic images, in which all the true edges and their locations are exactly known. Moreover, the noise level of synthetic images can be controlled. The robustness of an edge detection scheme against noise is one of the key performance measurements to judge the goodness of the scheme.

In this thesis work, three sets of well structured input images, namely, checkerboard, crossboard and ring images are used. They are chosen because they represent edges that appear in various orientations. Each set has a 128×128 noiseless image. The bright area has an intensity level of 200 and the dark area, an intensity of 100, i.e., the **edge contrast** is 100. The rest of the input images in each set are derived from the noiseless image by adding the following two kinds of noises.

1. **Gaussian noise:**

   Independently and identically distributed Gaussian noise of zero mean is added to each pixel in the noiseless image. Its distribution function conforms to the following model:

$$G = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, \tag{3.35}$$

   where, σ is the standard deviation which represents the Gaussian noise level.

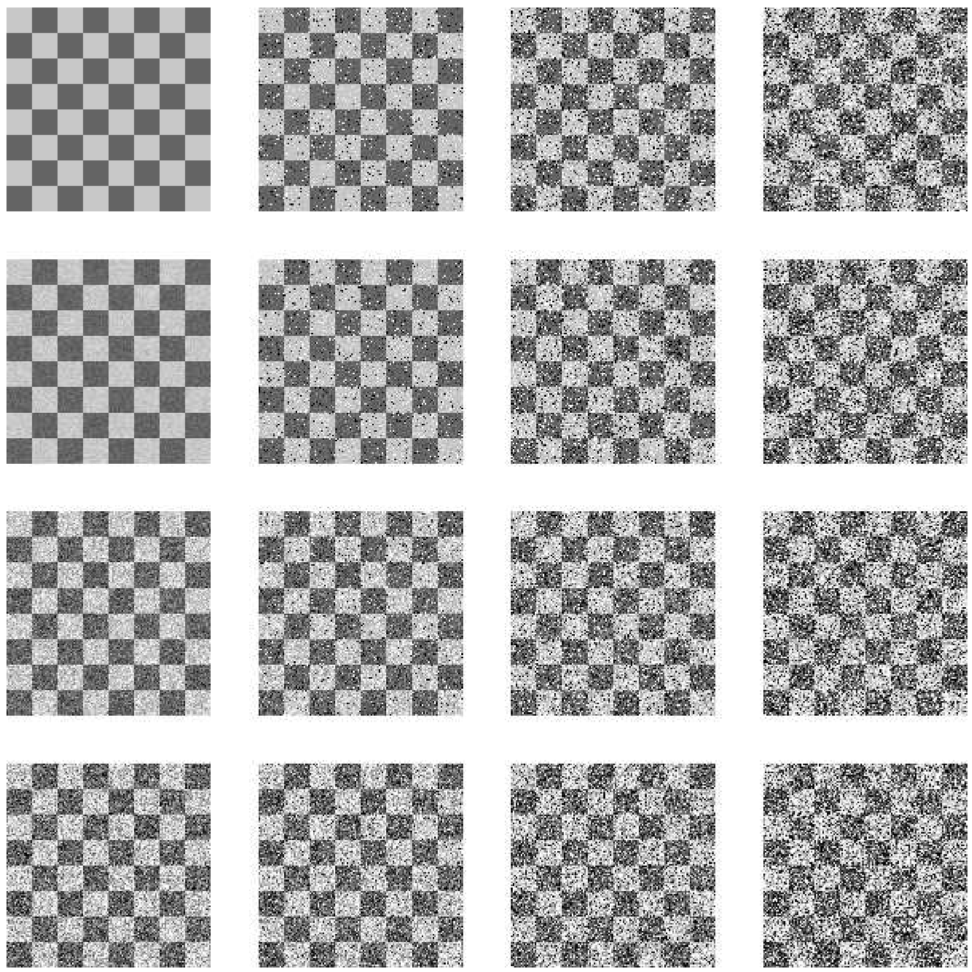
2. **Outliers:**

   Outliers are data points that do not conform to any known models. Normally outliers take the form of having extreme intensity values. Outlier percentage ϵ denotes the outlier level. In this work, outliers are generated by first randomly choosing the ϵ portion of all pixels to be outliers and then adding to the pixel value a randomly generated intensity value in the range from -255 to 255. (255 is the maximum gray pixel level.) For any pixel value less than zero, it is reset to be zero. For any pixel value larger than 255, it is reassigned the maximum gray level 255. In the resultant image, there are pixels having extreme intensity values compared with its neighbourhood. Visually these data points appear to be a black spot in a white area and vice versa.

In the experiments, there are 17 Gaussian noise levels in each input set of images, ranging from $\sigma = 0$ to $\sigma = 80$ in step of 5. At each Gaussian noise level there are 13 outlier levels, ranging from $\epsilon = 0\%$ to $\epsilon = 60\%$ in step of 5%. Hence, in each set of synthetic images, there are $17 \times 13 = 221$ images. Figures 3.10, 3.11 and 3.12 each show sample images in each set. Figure 3.13 shows the ideal edge maps of each synthetic image type. The performance of an edge operator is measured by comparing the edge map detected by the operator with the corresponding ideal edge map. The choice of the evaluation model and the experimental procedure are presented in the next section.

3.5.2 Evaluation Model and Experimental Procedure

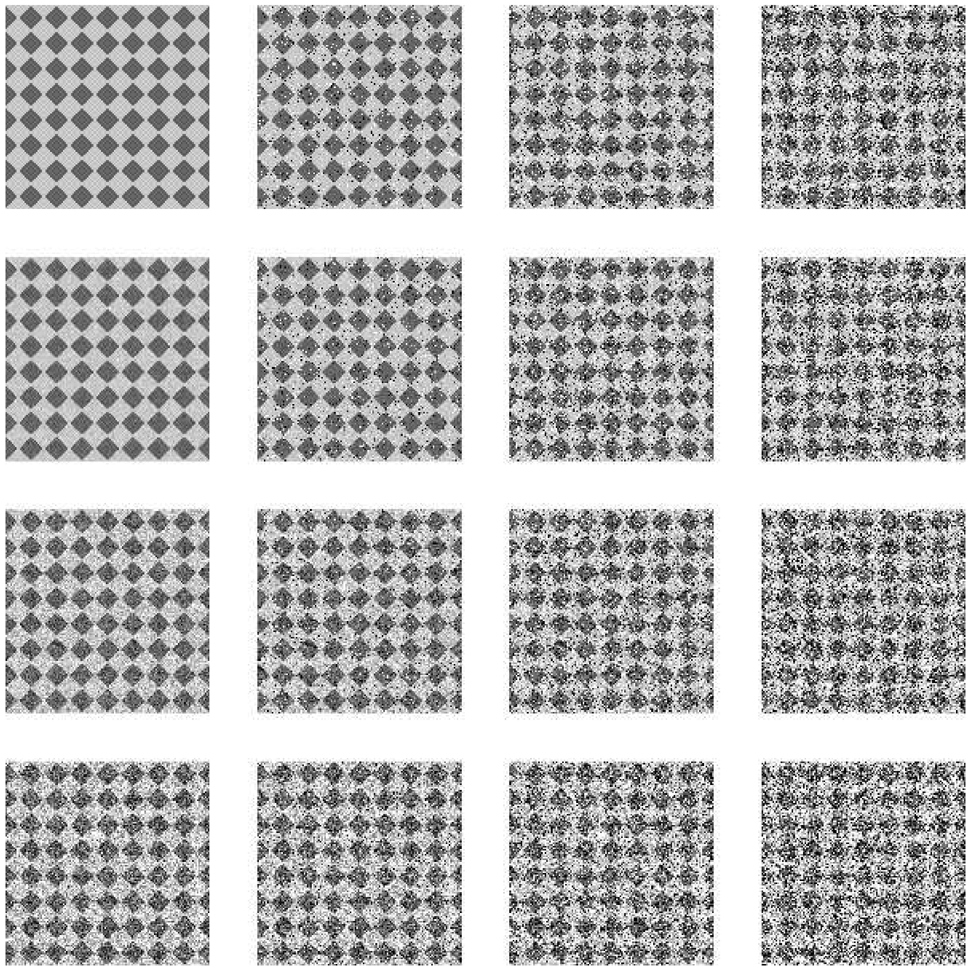
In order to evaluate the experimental results scientifically and statistically, quantitative methods must be used. The FCR model has been shown to be a good evaluation model on synthetic images. It is based on the ideal edge map (Section 2.6.1). In this model, the actual edge map is compared with the ideal one with true edge locations. Thus, the false-detected to correct-detected ratio is a good measurement to show the



Each row has the same Gaussian noise.
Each column has the same outlier percentage.

(1) noiseless	(2) $\sigma = 0, \epsilon = 10\%$	(3) $\sigma = 0, \epsilon = 30\%$	(4) $\sigma = 0, \epsilon = 50\%$
(5) $\sigma = 10, \epsilon = 0\%$	(6) $\sigma = 10, \epsilon = 10\%$	(7) $\sigma = 10, \epsilon = 30\%$	(8) $\sigma = 10, \epsilon = 50\%$
(9) $\sigma = 30, \epsilon = 0\%$	(10) $\sigma = 30, \epsilon = 10\%$	(11) $\sigma = 30, \epsilon = 30\%$	(12) $\sigma = 30, \epsilon = 50\%$
(13) $\sigma = 50, \epsilon = 0\%$	(14) $\sigma = 50, \epsilon = 10\%$	(15) $\sigma = 50, \epsilon = 30\%$	(16) $\sigma = 50, \epsilon = 50\%$

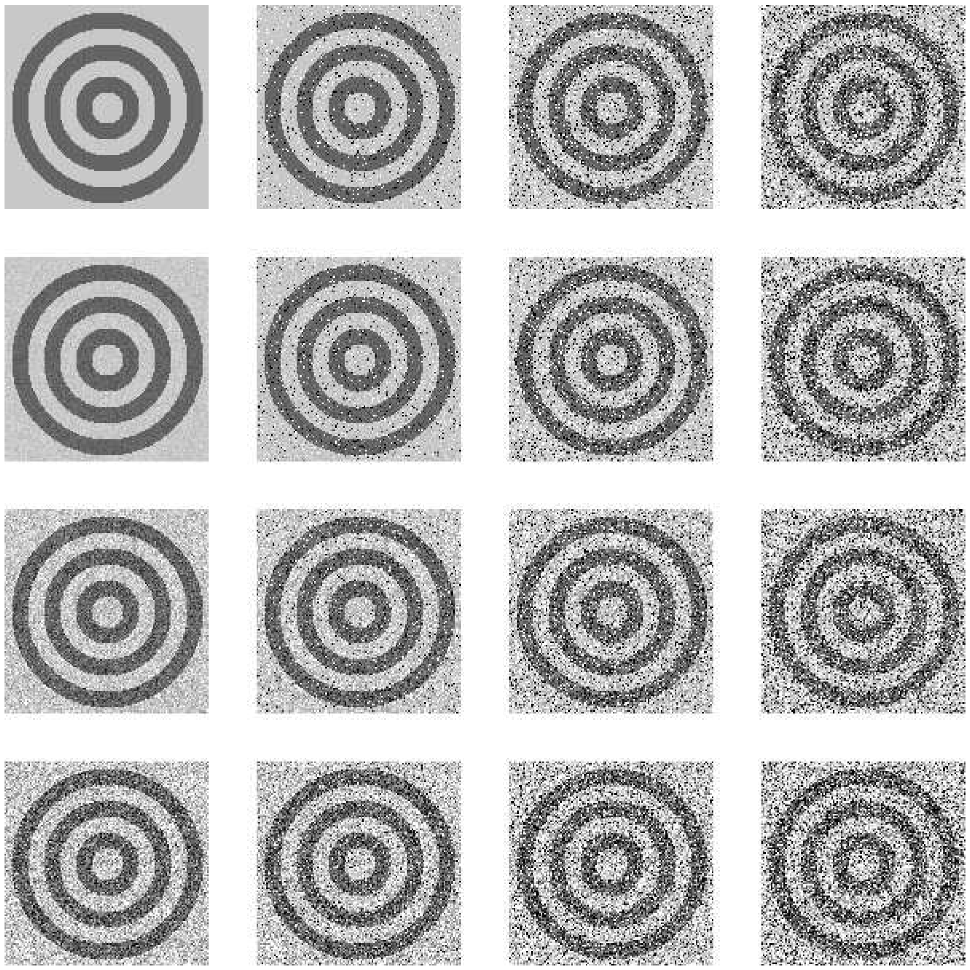
Figure 3.10: Checkerboard input images with size 128×128.



Each row has the same Gaussian noise.
Each column has the same outlier percentage.

(1) noiseless	(2) $\sigma = 0, \epsilon = 10\%$	(3) $\sigma = 0, \epsilon = 30\%$	(4) $\sigma = 0, \epsilon = 50\%$
(5) $\sigma = 10, \epsilon = 0\%$	(6) $\sigma = 10, \epsilon = 10\%$	(7) $\sigma = 10, \epsilon = 30\%$	(8) $\sigma = 10, \epsilon = 50\%$
(9) $\sigma = 30, \epsilon = 0\%$	(10) $\sigma = 30, \epsilon = 10\%$	(11) $\sigma = 30, \epsilon = 30\%$	(12) $\sigma = 30, \epsilon = 50\%$
(13) $\sigma = 50, \epsilon = 0\%$	(14) $\sigma = 50, \epsilon = 10\%$	(15) $\sigma = 50, \epsilon = 30\%$	(16) $\sigma = 50, \epsilon = 50\%$

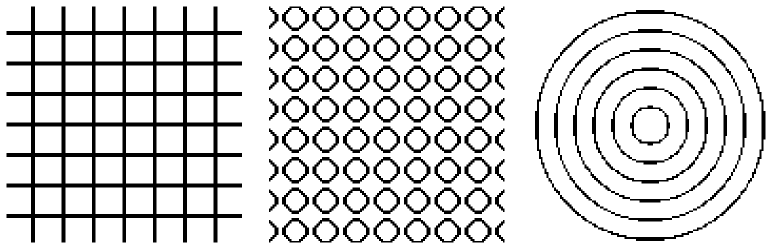
Figure 3.11: Crossboard input images with size 128×128.



Each row has the same Gaussian noise.
Each column has the same outlier percentage.

(1) noiseless	(2) $\sigma = 0, \epsilon = 10\%$	(3) $\sigma = 0, \epsilon = 30\%$	(4) $\sigma = 0, \epsilon = 50\%$
(5) $\sigma = 10, \epsilon = 0\%$	(6) $\sigma = 10, \epsilon = 10\%$	(7) $\sigma = 10, \epsilon = 30\%$	(8) $\sigma = 10, \epsilon = 50\%$
(9) $\sigma = 30, \epsilon = 0\%$	(10) $\sigma = 30, \epsilon = 10\%$	(11) $\sigma = 30, \epsilon = 30\%$	(12) $\sigma = 30, \epsilon = 50\%$
(13) $\sigma = 50, \epsilon = 0\%$	(14) $\sigma = 50, \epsilon = 10\%$	(15) $\sigma = 50, \epsilon = 30\%$	(16) $\sigma = 50, \epsilon = 50\%$

Figure 3.12: Ring input images with size 128×128.



(a)checkerboard (b) crossboard (c) ring

Figure 3.13: Ideal edge maps for synthetic images.

performance of a detector. It should be noted that the detecting ability of an operator should be measured separately from the effect of the postprocessing procedure. Up to this point, no postprocessing is involved. The FCR model is chosen to measure the performance of an edge detector on synthetic images.

In this thesis work, five edge detectors are evaluated and compared. Their parameter settings are listed below:

- **the Haralick operator:** window size is 7,
- **the RCBS operator:** IPV = 617, B = 1.8, projection size is 5,
- **the** $tanh$ **operator:** window size is 5,
- **the robust** $tanh$ **operator:** window size is 5.
- **the new robust edge operator:** window size is 5,

The raw output of the Haralick operator and RCBS is a gradient map. The raw output of the other operators is a contrast map. The final edge map is obtained by applying a threshold to the gradient/contrast map. The optimal threshold is chosen to be the one that gives the best performance measurement (the minimum value of

FCR). The minimum FCR value is defined as the performance index of an operator and is used in the evaluation and comparison procedures in this chapter.

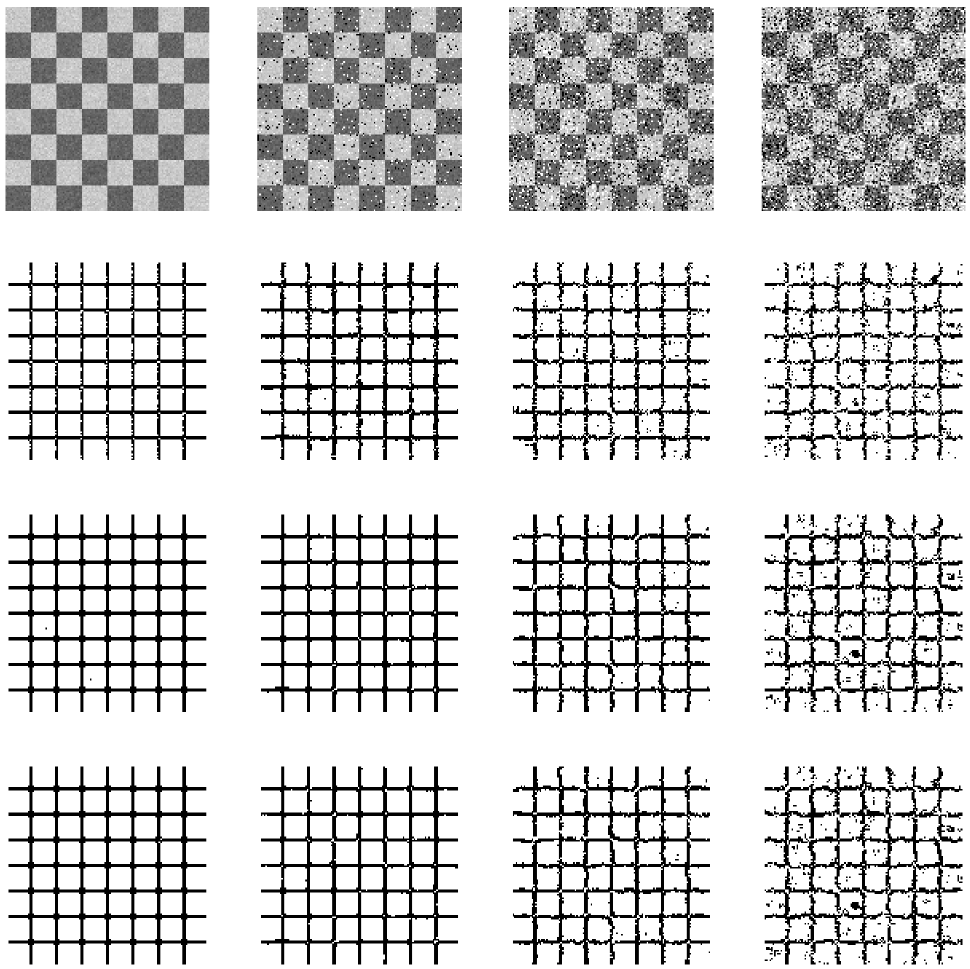
All of the experiments are done on a Sun SPARC 10 workstation. For a 128×128 image, Haralick's operator, RCBS and the new robust operator each take less than one minute of processing time. The computation time for the $tanh$ operator and the robust $tanh$ operator is about 2 minutes and 1 hour, respectively.

3.5.3 Factors in the New Robust Edge Detector

As mentioned in Section 3.4.3, there are two factors in the gradient estimation in the new robust edge detector. One is the size of the mask to do the convolution, the other one is the cubic refinement procedure. Figure 3.14 shows the edge maps generated by the new edge detector using masks of different sizes on checkerboard images. The optimal threshold is chosen using the FCR evaluation model. The Gaussian noise in the figure is $\sigma = 10$. The outlier percentage ranges from 0% to 50%. The first row is the original image. The second row is obtained by using a mask of size 3, and the third row, a mask of size 5. In the second and third rows, no cubic refinement procedure is involved. It is shown from the figure that the mask of size 5 produces better visual results. This is because the mask of size 3 is too small and the estimation of gradient direction can be easily biased by the noise.

The fourth row in Figure 3.14 is obtained by using the mask of size 5 and the cubic refinement procedure to refine the gradient direction. Comparing the third and the fourth rows, there is not much difference between the pairs of edge maps. As was discussed earlier, although the gradient direction may not be very accurate and the data points may be falsely put in another group, the LMS location estimator is robust enough to obtain good reliable results since it allows 50% of data points to be outliers. This indeed reinstates the robustness of the new edge detector.

From the discussion in this section, using a mask of size 5 without the refinement



First row: original images, $\sigma = 10, \epsilon = 0\%, 10\%, 30\%, 50\%$.
Second row: edge outputs of the new detector, mask size is 3, no refinement.
Third row: edge outputs of the new detector, mask size is 5, no refinement.
Second row: edge outputs of the new detector, mask size is 5, with refinement.

Figure 3.14: The effect of the size of the mask and refinement procedure.

procedure in the new edge detection scheme produces good results. This setting is used in all the experiments in the rest of this thesis.

3.5.4 Qualitative Comparison on Synthetic Images

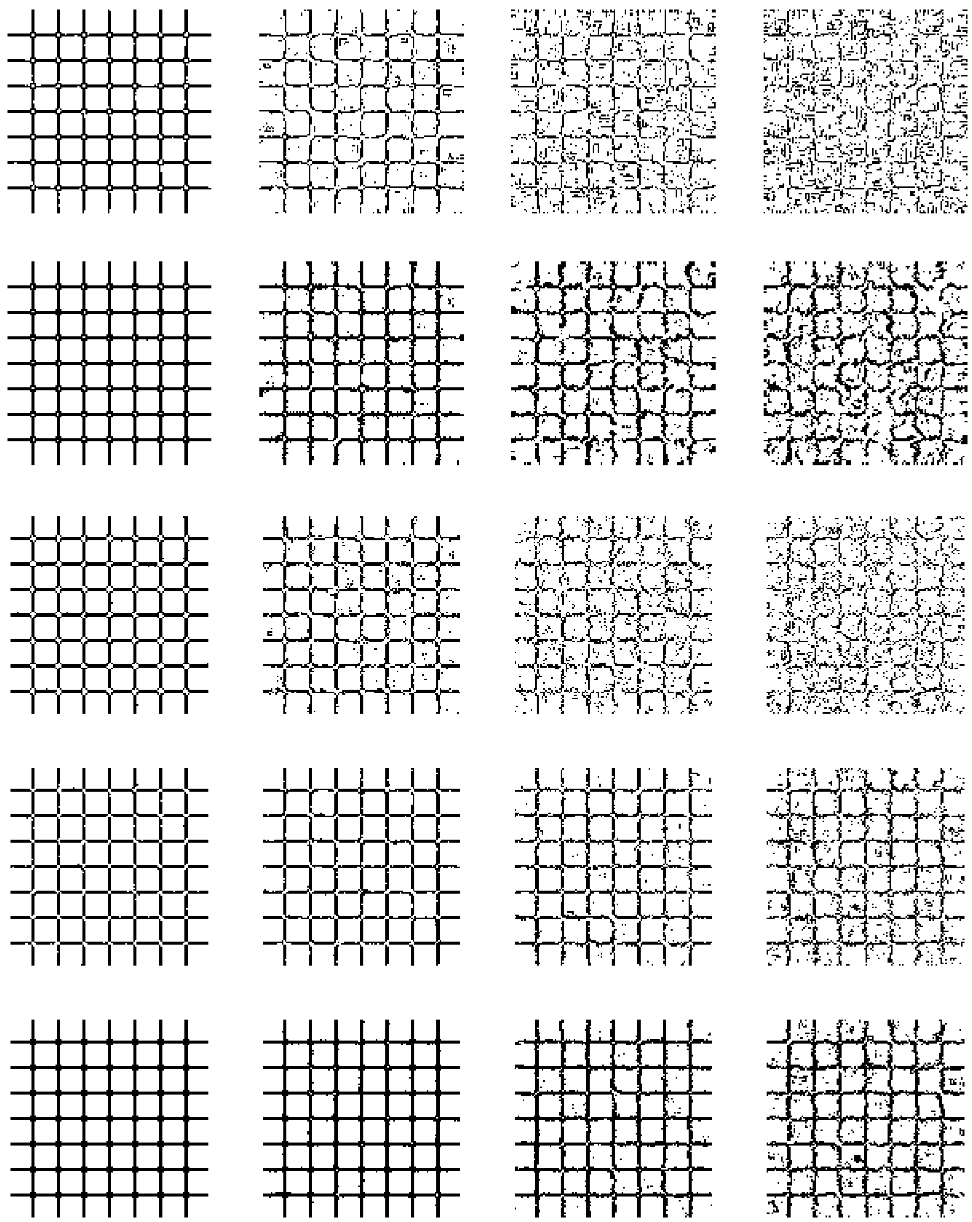
In this section, the performance of the Haralick operator, RCBS, the $tanh$ operator, the robust $tanh$ operator and the new robust edge detector are compared qualitatively. Figures 3.15, 3.16 and 3.17 present the typical results of the above operators on three types of synthetic input images. In all these figures, the Gaussian noise level of the input images is $\sigma = 10$. In each type of input image, there are four images with outlier percentage of 0%, 10%, 30% and 50% from left to right in each row. The original image is shown in the second row of Figure 3.10. As stated in earlier sections, the selected threshold used to get these edge maps is the one which gives the best FCR measurement. In Figures 3.15, 3.16 and 3.17 from the top row to the bottom row are the outputs of the Haralick operator, RCBS, the $tanh$ operator, the robust $tanh$ operator and the new robust operator, respectively. Each column is the outputs of different detectors on the same image.

It is shown in the figures that when there are no outliers (the first column) the five operators have similar results. When the outlier level increases, however, differences appear. The Haralick operator gives poor results in the presence of outliers, since outliers have a very large impact on the polynomial fitting results and lead to inaccurate detection of true edges. RCBS and the $tanh$ operator perform better than Haralick's operator. RCBS tends to produce thick edges when compared with the $tanh$ operator. When the outlier percentage is high, the performance of these two operators degrades significantly. There are lots of false-detected edges and many undetected true edges. The last two rows are the outputs of the two new robust edge detectors. As can be seen, these two operators give better visual results in the presence of outliers. Even when the outlier level is as high as 50%, true edges are still clearly visible. This shows that the robust detectors can effectively suppress noise caused by outliers. A closer look at the two robust operators shows that the robust $tanh$ operator tends to produce "weak" (less continuous) edges when compared with the new robust edge detector. The edge maps generated by using the new robust detector appear to be "darker" because many detected edges are two-pixel wide. In terms of visual effects, the new robust edge detector is more preferable than the robust $tanh$ operator when the noise level is high. Moreover, if the computation time is considered as a factor, the new detector is even more preferable since it takes less than one minute to process an image of size 128×128 while the robust $tanh$ operator takes about an hour on a Sun SPARC 10 workstation.

Experiments are done to compare the performance of the five operators on images with other Gaussian and outlier levels. Appendix B shows samples of other image outputs. The results show that the two robust edge detectors outperform the other three surface fitting approaches. The new robust edge detector gives better visual results compared with the robust $tanh$ operator. To set a common ground for evaluation and comparison, the next section gives the quantitative analysis of the performance of the operators.

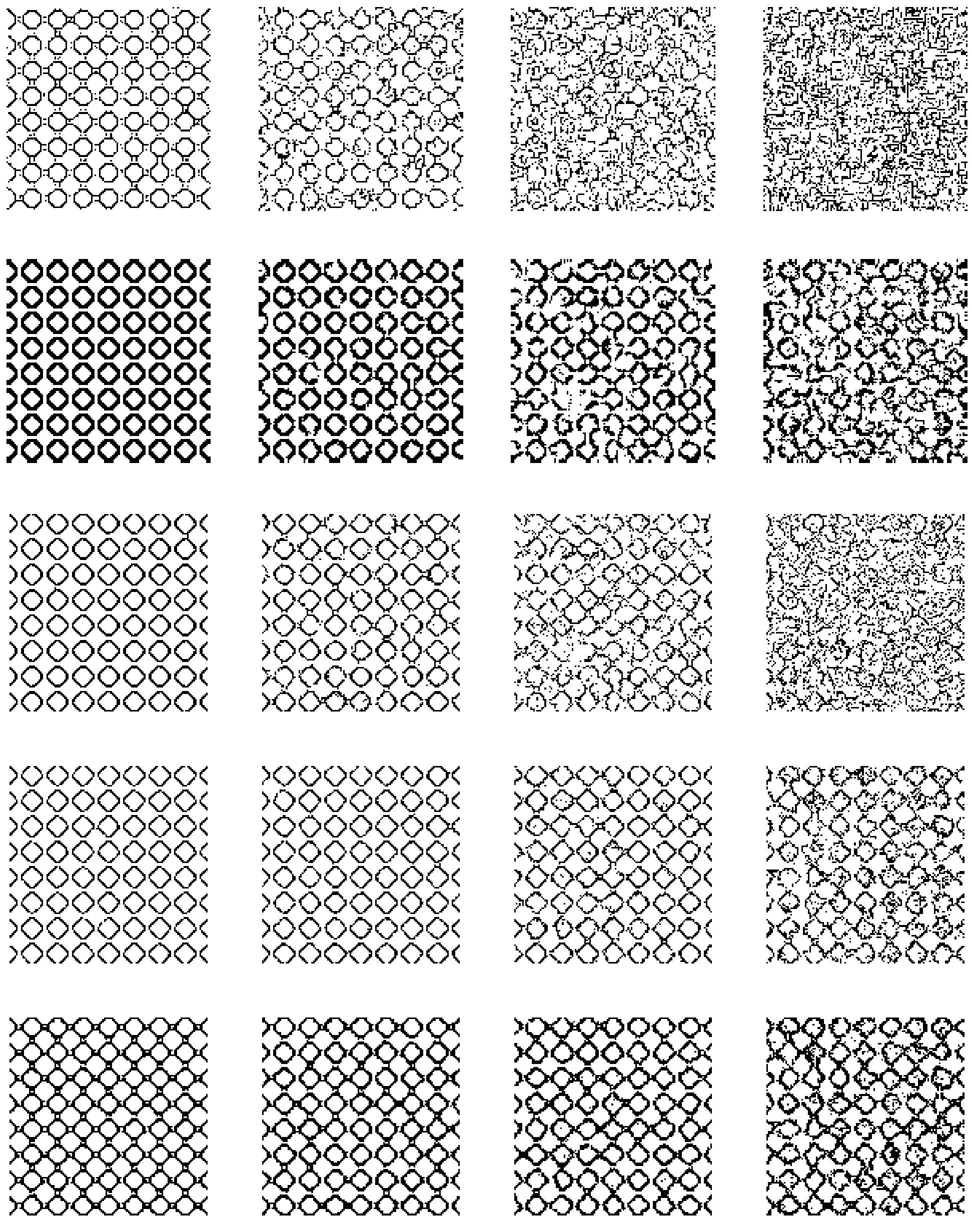
3.5.5 Quantitative Comparison on Synthetic Images

Two sets of experiments are done to quantitatively compare the performance of edge detectors. First, the Gaussian noise level is fixed and the changing parameter is the outlier percentage. Figure 3.18 shows the best measurement curves of checkerboard images. In this figure, the Gaussian noise level of input images is $\sigma = 10$. The outlier percentage ranges from 0% to 60% in steps of 5%. (In all of the figures in this thesis, $DDER$ represents Haralick's operator, and $tanh2$, the robust $tanh$ operator.) The observed rating from best to worst is: the new edge detector, RCBS, the $tanh$ and the Haralick operator. The new robust edge detector gives better performance



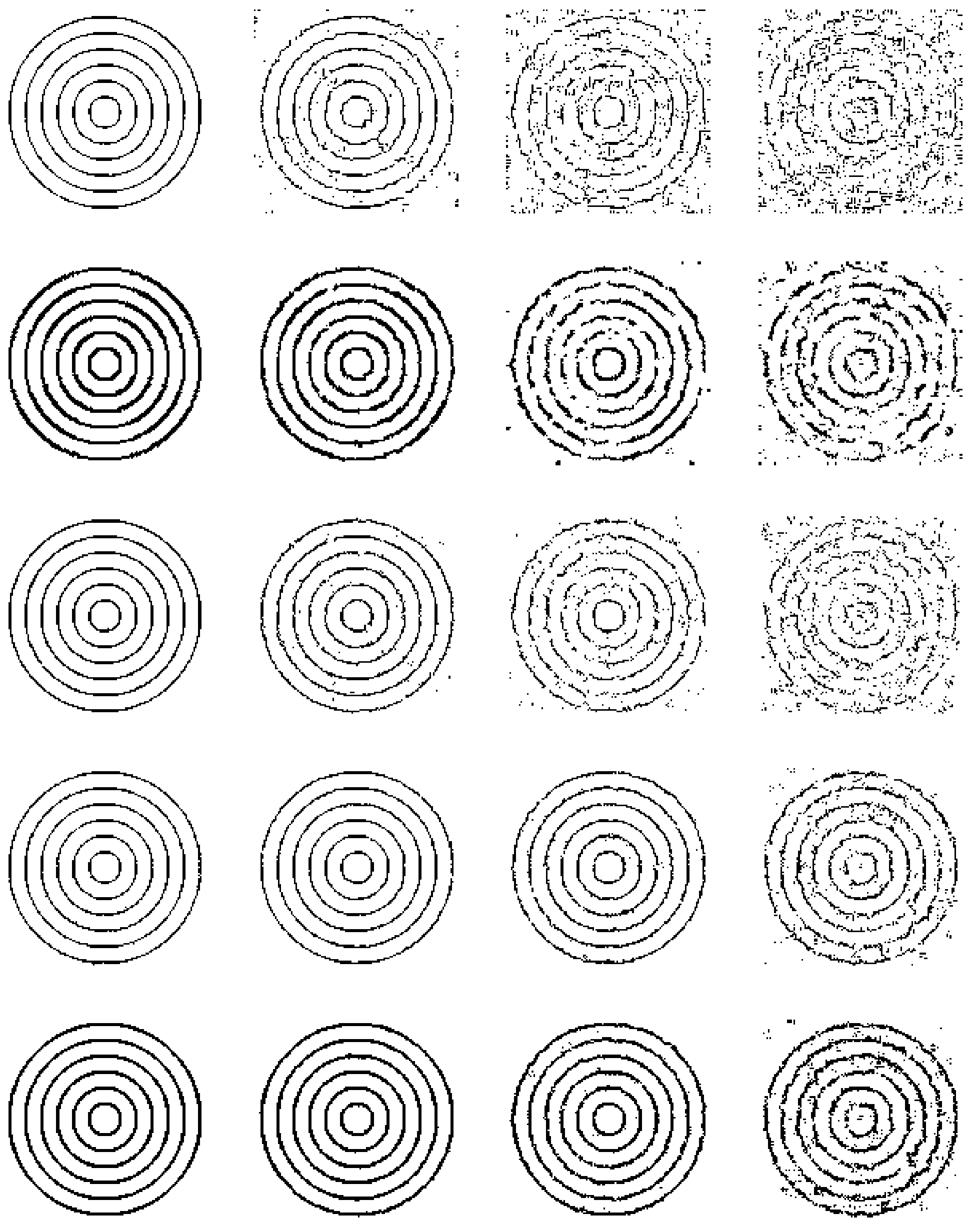
First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the *tanh* operator.
Fourth row: outputs of the robust *tanh* operator.
Fifth row: outputs of the new edge detector.

Figure 3.15: Comparison of checkerboard outputs (FCR model), $\sigma = 10$.



First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the *tanh* operator.
Fourth row: outputs of the robust *tanh* operator.
Fifth row: outputs of the new edge detector.

Figure 3.16: Comparison of crossboard outputs (FCR model), $\sigma = 10$.



First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the $tanh$ operator.
Fourth row: outputs of the robust $tanh$ operator.
Fifth row: outputs of the new edge detector.

Figure 3.17: Comparison of ring outputs (FCR model), $\sigma = 10$.

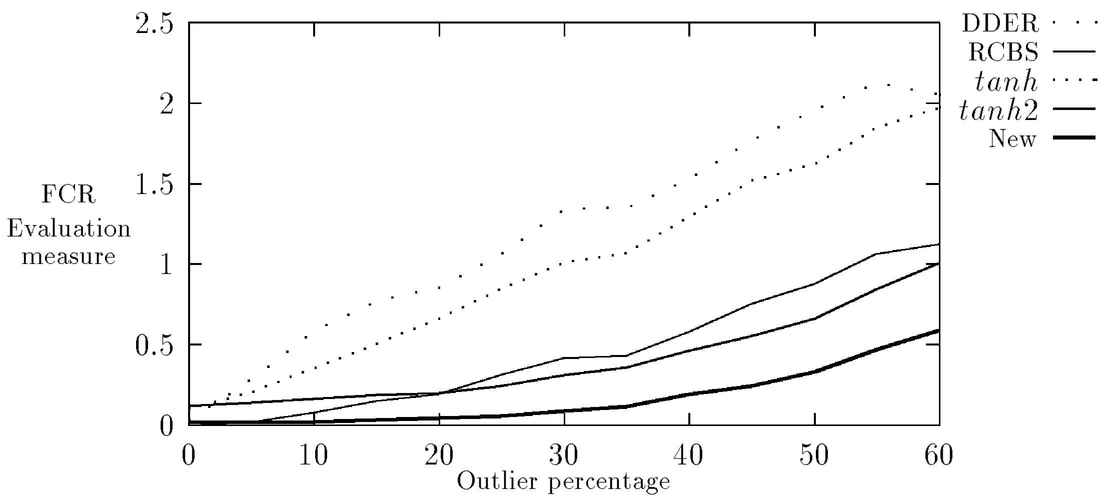


Figure 3.18: FCR measure w.r.t. outlier level on checkerboard ($\sigma = 10$).

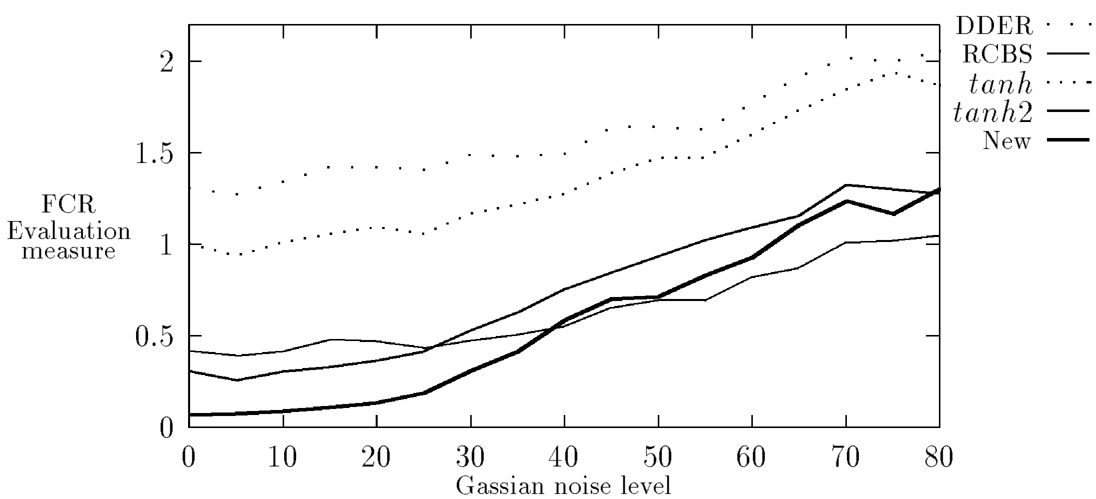


Figure 3.19: FCR measure w.r.t. Gaussian noise on checkerboard ($\epsilon = 30$).

than the robust $tanh$ operator. When the outlier percentage is low, RCBS is a little bit better than the robust $tanh$ operator. When the outlier percentage is high, the robust $tanh$ operator performs better than RCBS. This shows that RCBS has a strong ability to suppress noise, especially when the Gaussian noise is the major source of noise. But as the outlier percentage increases, the performance of RCBS degrades and the robust $tanh$ operator shows its advantage. The second set of experiments shows this more clearly.

In the second set of experiments, the outlier percentage of all input images is 30%. The Gaussian noise level ranges from 0 to 80 in step of 5. Figure 3.19 presents the performance curves. The two robust edge detectors and the RCBS operator perform better than the Haralick operator and the $tanh$ operator. The new robust edge detector systematically performs better than the robust $tanh$ operator. Comparing the two robust operators and the RCBS operator, when the Gaussian noise is low (thus outliers are the major source of noise), the two robust detectors perform better. When the Gaussian noise becomes the major source of noise, RCBS is better. The crossing over point in the figure is where the Gaussian noise level is about the same as the outlier level. (More experiment results at other noise levels can be found in Appendix C. Because the robust $tanh$ operator is computationally excessive, they are only applied on samples of crossboard and ring images. Thus their performance curves are not shown from Figure C.3 to Figure C.10.)

The objective of developing robust edge detectors is to deal with noise which does not conform to the Gaussian model. Experiments show that the proposed robust detectors do perform better than other conventional surface fitting approaches in this case. Furthermore, the new edge detector based on the LMS location estimator performs better than the robust $tanh$ operator.

3.5.6 Qualitative Comparison on Real Image

Figure 3.20 presents the results of applying the five edge detectors discussed above to real images. Figure 3.20(a) is the original image. Though the noise level in this image is not known, it is perceived that there are almost no outliers in the image and the Gaussian noise level is not high. Figure 3.20(b) is obtained by adding outliers ($\epsilon = 5\%$) on the image shown in Figure 3.20(a). Since the FCR model is not applicable on real images, the threshold is chosen manually. The selected threshold is the one that yields the best visual result. The edge maps in each row are obtained using the same edge detector and correspond to the same image in the same column. It is clearly shown that the results of the Haralick operator, RCBS and the $tanh$ operator are very sensitive to outliers. Even when the outlier level is not high, the performance of these three operators degrades significantly. The robust $tanh$ operator and the new robust operator, on the other hand, show their strong ability in suppressing noise.

The manual selection of threshold is a time-consuming procedure which involves human interactions. Furthermore, there is no clearly-stated standard in the human evaluation of an edge map. Thus, a performance evaluation model which can be applied to real images is highly desirable, the details of which are given in the next chapter.

3.6 Summary

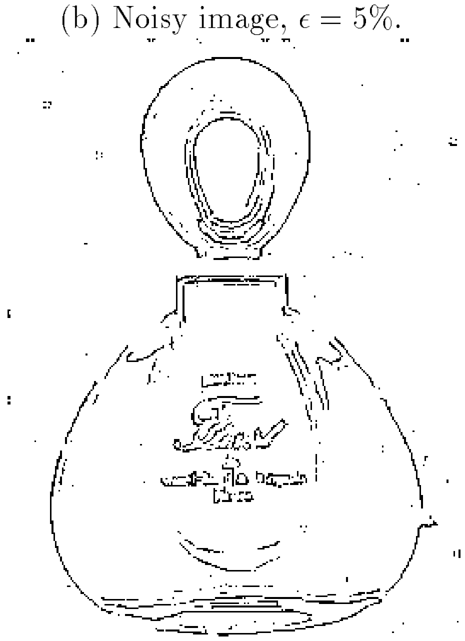
Least-median-of-squares (LMS) is an important estimator in robust statistics. This chapter gives a review of the properties of this estimator and studies its application in edge detection. The random sampling technique is employed to obtain the LMS solution when there is more than one coefficient. The performance of the LMS estimator and random sampling is given. The LMS technique is incorporated in the $tanh$ operator to make the operator robust to outliers. A new fast robust edge detector based on the LMS location estimator is also proposed. This detector has a high breakdown point of 50%. Qualitative and quantitative comparison between the two new robust detectors, the Haralick operator, RCBS, and the $tanh$ operator on synthetic images is presented. Experiments show that the proposed detectors perform very well especially when the noise does not conform to the Gaussian model.

The quantitative analysis in this chapter is based on the FCR model. This model can only be used on synthetic images. Thus, the choice of optimal threshold and quantitative analysis on real images still remain a problem. The next chapter aims to develop an evaluation scheme that can be used on both real and synthetic images.

(a) Original image.

(b) Noisy image, $\epsilon = 5\%$.

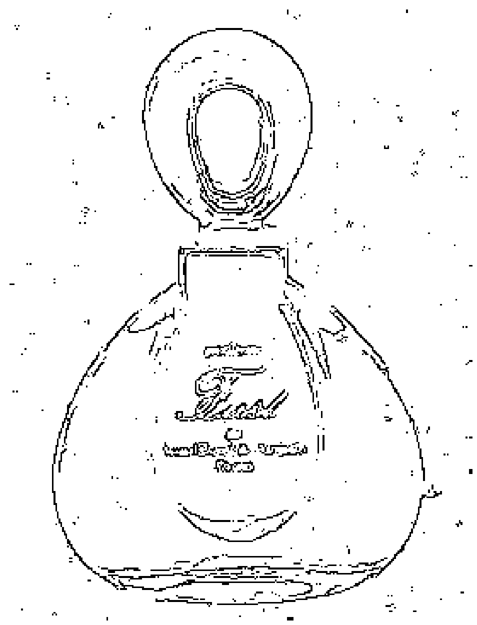
(c) the Haralick operator, $T = 8$.



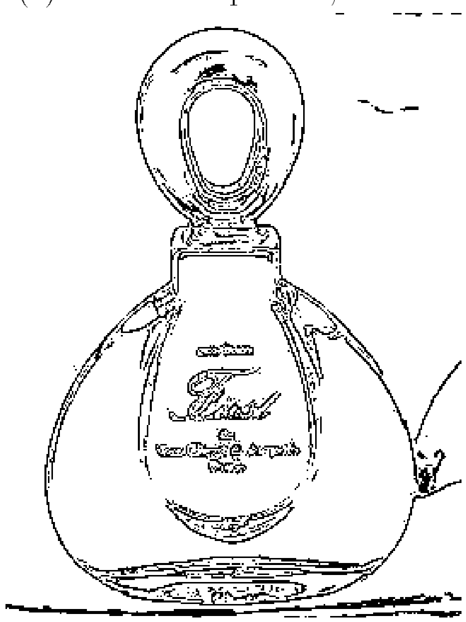
(d) the Haralick operator, $T = 25$.

(Continued on the next page.)

(e) the RCBS operator, $T = 12$.



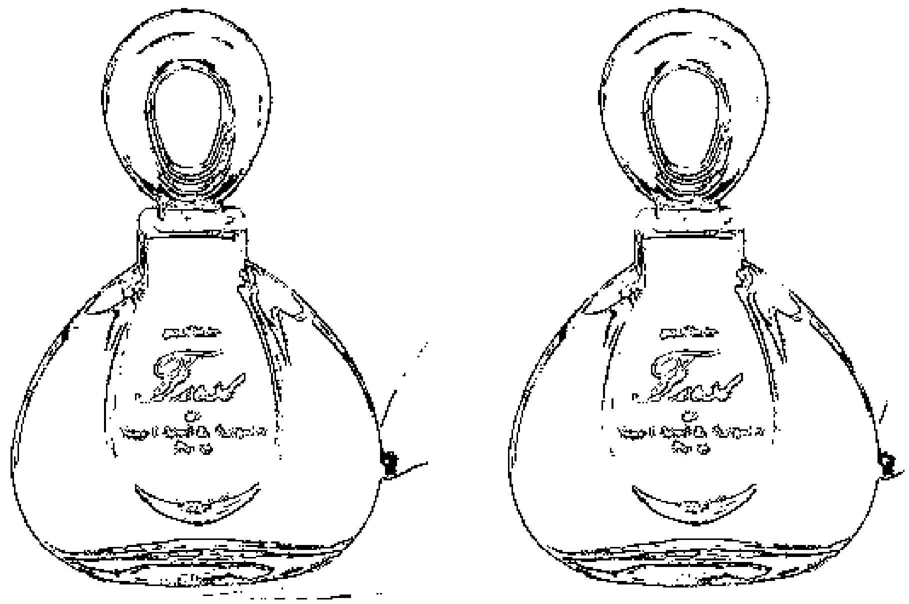
(f) the RCBS operator, $T = 26$.



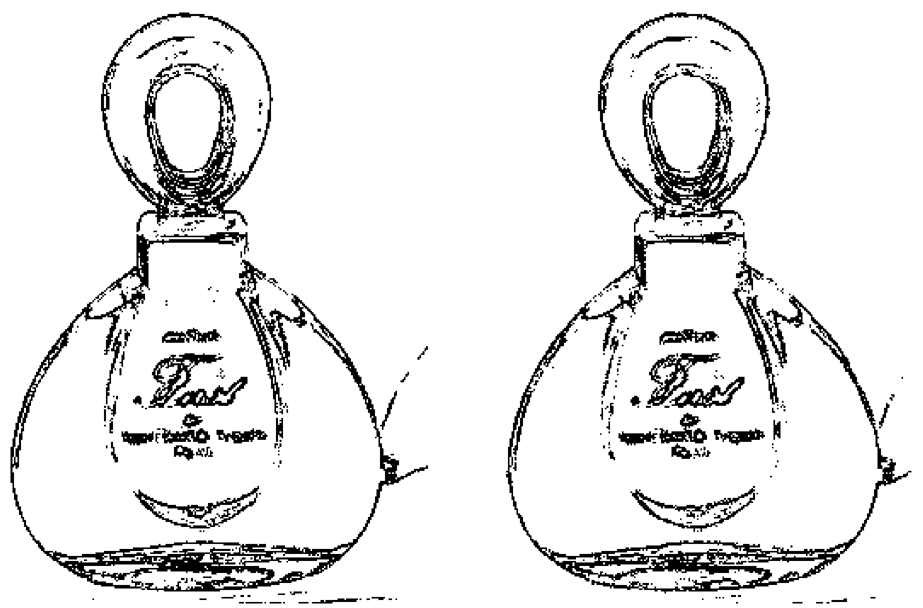
(g) the $tanh$ operator, $T = 14$.

(h) the $tanh$ operator, $T = 48$.

(Continued on the next page.)



(i) the robust $tanh$ operator, $T = 25$. (j) the robust $tanh$ operator, $T = 32$.



(k) the new robust operator, $T = 30$. (l) the new robust operator, $T = 35$.

Figure 3.20: Outputs of edge detectors on bottle image using manually chosen threshold.

Chapter 4

Performance Evaluation

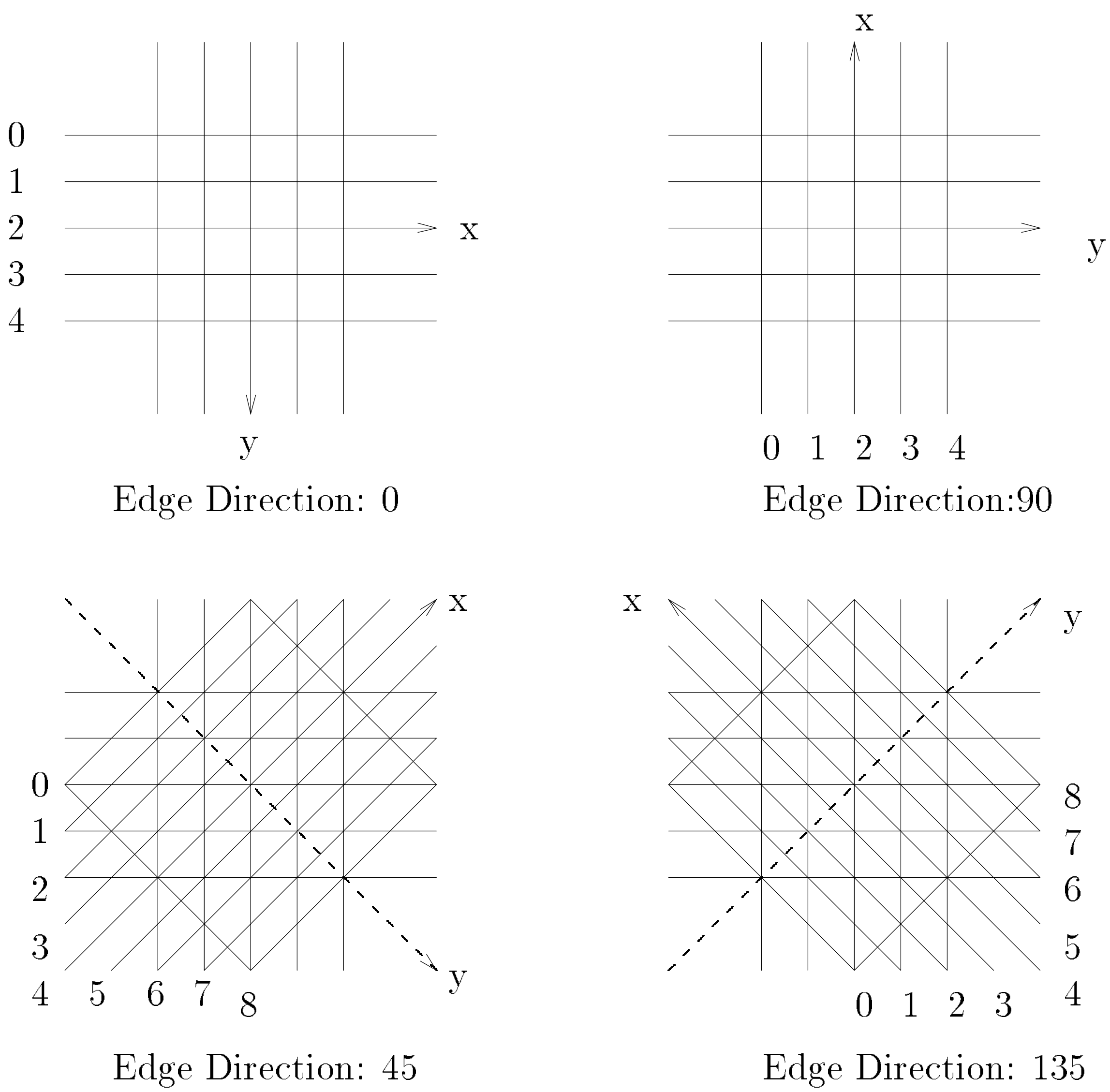
In this chapter, a novel quantitative edge performance evaluation model is proposed. This new model is based on the formation of edges and is to be used on both synthetic and real images. Section 4.1 presents the new evaluation model. Section 4.2 and 4.3 study the behaviour of the new model and an edge refinement procedure using such a model is proposed. Section 4.4 and 4.5 study the performance of edge detectors using this new model on synthetic and real images, respectively. Finally, Section 4.6 summarizes this chapter.

4.1 New Evaluation Scheme

In edge detection, quantitative performance evaluation and comparison is not a common practice. Though some evaluation models have been proposed (Section 2.6), most of them have highly restricted structures and are not quite general. The FCR model which is based on ideal edge maps is shown to be a good evaluation model on synthetic images. Unfortunately, it cannot be applied to real images where the true edge locations are not known. The local edge coherence (LEC) scheme is based on edge formation, yet it fails to capture the global characteristics of edges by examining only 3×3 neighbourhoods. Furthermore, it does not perform well on real images (Figure 2.9). To get an edge map, the raw output of the detector has to be thresholded manually, which is often based on human justification. The development of the new edge evaluation scheme aims to automate this process. The new model is a portable and operator independent quantitative evaluation model. This makes the quantitative analysis more feasible and general. The evaluation scheme does not have to be a part of an edge detection scheme. It takes the raw output of any detectors as input and performs the analysis. No human interaction is involved in deriving the final edge map from the original input. The following subsections present the new evaluation model in detail.

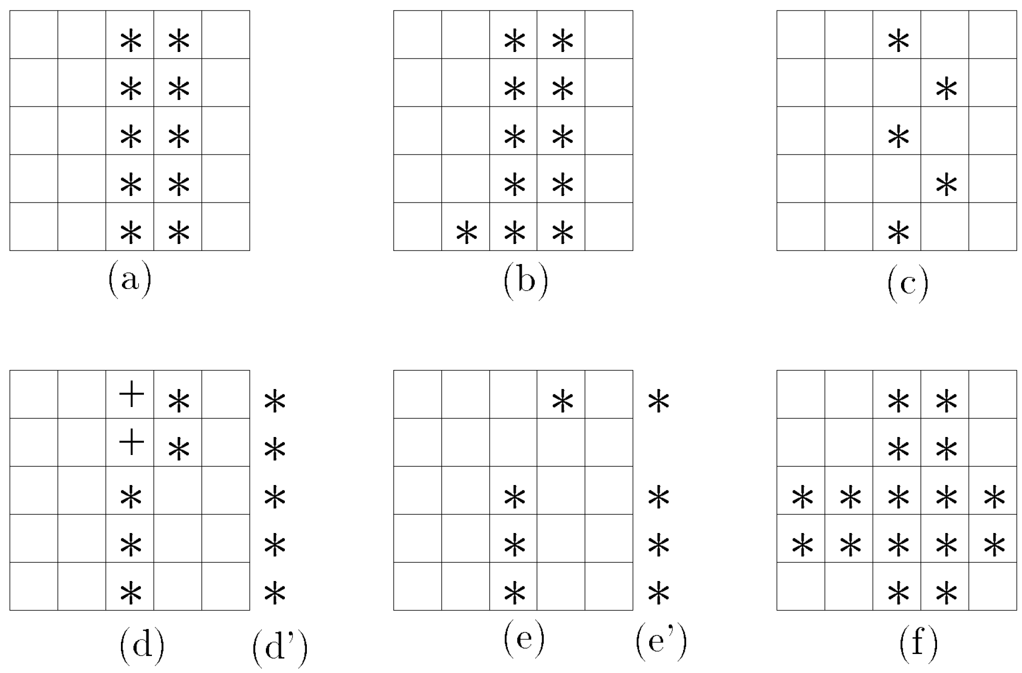
4.1.1 Evaluation Window

The new evaluation model is based on edge formation and does not require *a priori* knowledge of the true locations of edges. Each edge pixel is examined by centering at a 5×5 window, which is called the evaluation window. A 3×3 window is not considered large enough to estimate the trend of the edge. Each edge pixel is associated with its gradient direction (therefore, edge direction). In the experiments in this thesis work, there are four possible directions: $0°$, $45°$, $90°$ and $135°$. Depending on the direction of the edge, the evaluation window is rotated (if necessary) so that the edge lies horizontally in the evaluation window. Figure 4.1 shows four possible cases. The motivation for the rotation is to capture enough data points to do the evaluation in the case of slanted edges. Scanlines are lines in the evaluation window parallel to the edge direction. When the window is horizontal (or vertical), there are 5 scanlines. There are 9 scanlines when the window is rotated by 45 or 135 degrees. The indices of the scanlines in each case are shown in Figure 4.1.



Note: edge direction is along the x-axis. Integer numbers are scanline indexes.

Figure 4.1: Rotation of the evaluation window.



Note: Pixels marked with * are detected edge pixels.

Figure 4.2: Sample forms of edges in edge map.

4.1.2 Measurement Components

Local Components:

Figure 4.2 shows sample edge neighbourhoods in an edge map. A closer look at these edge neighbourhoods leads to the consideration of the following measures.

- **Thinness (T):**

  Edges are supposed to be thin. It is desirable that thick edges are penalized. In the LEC evaluation scheme, the size of the evaluation window is 3×3. An ideal edge is considered to be one pixel wide and the edge pixels in the window are all along the direction of the center pixel (edge direction). So, ideally, there should be three edge pixels along the edge direction, which form a straight line. The thinness measure for the evaluation window is computed as the fraction of the remaining six pixels of the window that are non-edge pixels. There are several problems with this evaluation scheme. First, the window size is too small to capture the edge characteristics. Second, an ideal edge may not form a straight line because of discretization. Finally, well-formed intersecting edges are penalized, where they should not be. Based on these observations, thinness is defined in a different way in the new evaluation scheme.

A 5×5 evaluation window is used. To get the width of the edge, the robust LMS location estimator provides a useful tool. Using LMS, the narrowest strip (band) covering half of the edge points can be found. This greatly helps to eliminate the influence of edge pixels which belong to adjacent edges but do appear in the evaluation window. Pixels outside the band are considered to be "outliers" in the edge neighbourhood. The width of the band is the number of scanlines inside the band. It should be noted that the result of the LMS location estimation depends on the position of the data points in the array being sorted. For example, in Figure 4.2 (b), the direct result of the LMS solution is a band covering the second and the third columns. The width of the band is two. As is obvious, a new band formed by the third and the fourth columns covers even more data points than the second and third columns. It also has a width of two. The direct LMS solution may be either of these two solutions depending on the positions of data points in the sorting procedure. To deal with this, the solution of the LMS estimation is modified by examining one more scanline above and one below the band. Let E_i denote the number of edge pixels in scanline i, N_i, the number of all pixels in scanline i, Max_index, the maximum index of the scanline, i_{low}, the index of the lower bound scanline of the band and i_{up}, the index of the upper bound scanline of the band. The new **modified band** is formed by:

If $i_{low} > 0$, then /* one more lower scanline exists */

If $E_{i_{low}-1} > E_{i_{up}}$, then

Begin /* band moving one scanline down */

$i_{low} = i_{low} - 1;$

$i_{up} = i_{up} - 1;$

End

If $i_{up} < Max\_index$, then /* one more upper scanline exists */

If $E_{i_{up}+1} > E_{i_{low}}$, then

Begin /* band moving one scanline up */

$i_{low} = i_{low} + 1;$

$i_{up} = i_{up} + 1;$

End

The **expanded band** is defined to be the band including one more scanline above and one more scanline below the original band, if the scanline exists and there is at least one edge pixel in the scanline. Let j_{low} and j_{up} denote the index of the lower and upper bounds of the expanded band.

The width of the edge is defined to be:

$Width = i_{up} - i_{low} + 1;$ /* width of the modified band */

If $i_{low} \neq j_{low}$ and $E_{j_{low}} > N_{j_{low}}/2$, then

$Width = Width + 1;$

If $i_{up} \neq j_{up}$ and $E_{j_{up}} > N_{j_{up}}/2$, then

$Width = Width + 1;$

(Note: the width of the edge is increased if the edge pixels in the scanline adjacent to the expanded band is more than half of the pixels in that scanline.)

Ideally edges should be one pixel wide. Practically, edges of two pixels wide are considered to be acceptable in this thesis. The thinness measure is computed by:

$$T = \begin{cases} 0.0, & \text{if } Width \leq 2, \\ Width/(Max\_index + 1), & \text{otherwise.} \end{cases} \tag{4.1}$$

$Max\_index + 1$ is the total number of scanlines.

(All the measurement components are designed to give better visual results when the measure is small. Also the measure ranges from 0 to 1.)

- **Roughness (R):**

  Edges are supposed to be smooth. Zig-zag edges are not desirable. The roughness measure tends to evaluate the zig-zaggedness of edges. This measure has not been studied in any other evaluation models. It is an original contribution of this thesis to propose it as an important measurement. In the FCR model, the numbers of true-positive, true-approximate and false-positive pixels are counted. The edge formation is not considered. For example, compare Figure 4.2(a) and (c). Clearly (a) is much more preferable than (c) in the sense that the edge is smoother. In one scanline, if a pixel is an edge pixel, but the next pixel adjacent to it is not (or vice versa), this is counted as one *jump*. For example, in Figure 4.2(c), there are in total 4 jumps in the third column from row i to row $i+1$, $i = 1, \ldots, 4$. Let J_i denote the number of jumps in scanline i and N_i, the number of all pixels in scanline i. The roughness measure R is computed by:

$$R = \frac{\sum_{i \in \text{expanded band}} (J_i/(N_i - 1))}{j_{up} - j_{low} + 1}, \tag{4.2}$$

  where $N_i - 1$ is the maximum possible number of jumps in one scanline, and $j_{up} - j_{low} + 1$, the total number of scanlines in the expanded band.

  For a perfect edge as in Figure 4.2 (a), $N_2 = N_3 = 5$, $J_2 = J_3 = 0$, therefore,

$R = 0.0$. For extremely zig-zag edges as in Figure 4.2 (c), $N_2 = N_3 = 5$, $J_2 = J_3 = 4$, $j_{low} = 2$, $j_{up} = 3$, thus, $R = 1.0$.

- **Continuity (C):**

  Ideally an edge should be continuous between its two end points. In the actual detected edge map, this is not always the case. The C measurement measures the continuity of the edge in the evaluation window. It should be noted that because of discretization an edge may not be "straight" in the evaluation window. It may be shifted by one pixel as shown in Figure 4.2 (d). The edge in Figure 4.2(d) is still considered to be continuous. In the LEC evaluation scheme, the continuity score measures how well the edge direction at the neighbour agrees with that at the center and how close a neighbour is to the expected direction of the edge. In Figure 4.2(d), the edge direction of the center pixel is 90 degree. The two pixels marked with "+" are along the expected direction of the edge. However, because of discretization, the actual detected edge pixels are shifted one pixel to the right. In the LEC scheme, this case is penalized because the top two edge pixels are not considered to be close to the expected direction of the edge. The LEC scheme does not take into account the discretization problem. Also the edge detection at each pixel may not be accurate. The continuity score can be easily biased by the error in the direction estimation.

  In the new evaluation scheme, to evaluate the continuity of an edge, the scanlines in the modified band are "squeezed" into one line. Figures 4.2(d′) and (e′) show the squeezed line obtained from Figure 4.2(d) and (e), respectively. A continuous segment in the squeezed line is called an *edge segment*. The number of edge segments in the squeezed line is denoted by N_{seg}. For example, in Figure 4.2(d′), there is only one edge segment, $N_{seg} = 1$. In Figure 4.2(e′), there are two edge segments, $N_{seg} = 2$. This procedure aims to solve the discretization problem.

The ideal length, L_{ideal}, of an edge is not the window size, since an edge may terminate inside the window. L_{ideal} should also be larger than half of the window size, otherwise it is too short to be considered as an edge. Let $Radius$ denote the integer part of half of the window size, and L_t, the longest distance between two edge pixels in the squeezed line. (The two edge pixels need not be connected.) In the case of a 5×5 evaluation window, $Radius = 2$. In both Figure 4.2(d′) and Figure 4.2(e′), $L_t = 4$. The ideal length of the edge is:

$$L_{ideal} = Max\ \{L_t, Radius\}. \tag{4.3}$$

The ideal length in both Figure 4.2(d′) and Figure 4.2(e′) is 4. Let L_i denote the length of edge segment i (after squeezing), the continuity measure C is computed by:

$$C = 1.0 - \frac{\mathrm{Max}_i\ L_i}{L_{ideal} * N_{seg}}, \tag{4.4}$$

In Equation 4.4, more edge segments are penalized by putting N_{seg} at the denominator which yields larger C. $Max_i\ L_i$ is the longest edge segment in the squeezed line. If the number of edge segments is the same, C favours longer edge segments. For example, in Figure 4.2(d′), $N_{seg} = 1$, $L_1 = 4$, $L_{ideal} = 4$, therefore $C = 0.0$, which means the edge in Figure 4.2(d) is considered to be continuous. In Figure 4.2(e′), $N_{seg} = 2$, $L_1 = 2$, $L_2 = 0$, $L_{ideal} = 4$, thus $C = 0.75$. This measure penalizes discontinuous edges.

Global Component:

- **Edge density (E):**

  It is noticed that after an edge detection operation, when the threshold is set very high, the resultant edge map may have very few edge pixels. Yet the local measures can still be very good. This is not desirable. Now the ratio of edge pixels w.r.t. the total number of pixels becomes another concern. So another component called **edge density** (E) is added to penalize too few or too many edges.

$$Ratio = \frac{\text{the number of edge pixels}}{\text{the total number of pixels in the image}} \tag{4.5}$$

$$E = \begin{cases} 1.0 - Ratio, & \text{if } 0\% \leq Ratio < 1\%, \\ 0.0, & \text{if } 1\% \leq Ratio < 50\%, \\ Ratio, & \text{if } 50\% \leq Ratio \leq 100\%. \end{cases} \tag{4.6}$$

Here, 50% is chosen to be the highest reasonable ratio, since, ideally, there should not be more than 50% edge pixels in an edge map. The lowest reasonable ratio is chosen to be 1%. The choice of this ratio is not critical as long as it is reasonable. Experiments show that the extreme case with very few edge pixels and very good local measurements normally happens when very large threshold is used in the raw output. In this case the ratio has already dropped to as low as 0.1%. Thus the choice of the lowest ratio is not sensitive.

Aggregate Edge Quality:

The local measurements, namely, continuity, roughness, and thinness, are computed for each edge neighbourhood in the edge map. The overall measurements (denoted by $\bar{T}$, $\bar{R}$ and $\bar{C}$) are obtained by averaging over all edge neighbourhoods in the edge map. The final aggregate edge quality measure, M, is a weighted combination of all of the components discussed above.

$$M = W_R * \bar{R} + W_C * \bar{C} + W_T * \bar{T} + W_E * E, \tag{4.7}$$

where, W_R, W_C, W_T and W_E are the weights of the corresponding components. The choice of these weights is given in Section 4.2.

4.1.3 Intersecting Edges

The previous discussion is based on the assumption that there is only one edge in the evaluation window. In an actual edge map, there are cases when edges intersect in the window as shown in Figure 4.2 (f). If intersecting edges are not considered, edges like that shown in Figure 4.2 (f) will be penalized greatly, since edge points along the other edge direction increases the roughness measure of the window. In this thesis, this problem is addressed as follows.

In order to remove the influence of intersecting edges on the local measurements, the new evaluation scheme first identifies the major edge directions in the evaluation window. Different directions are considered separately. For each direction, the local window measurements in Section 4.1.2 are computed by rotating the evaluation window along that direction. Edge pixels of other directions are not counted when fitting the strip and computing the roughness. This will effectively reduce the influence of intersecting edges. Suppose R_i, C_i and T_i are measurement values obtained when evaluation window is along direction i, $i = 1, \ \ldots, \ N_d$. The resultant window measurements are:

$$R = \frac{1}{N_d} \sum R_i \tag{4.8}$$

$$C = \frac{1}{N_d} \sum C_i \tag{4.9}$$

$$T = \frac{1}{N_d} \sum T_i. \tag{4.10}$$

In this thesis, there are four possible edge directions (0, 45, 90 and 135 degrees). Only two major directions are considered. Let D_i denote the number of edge pixels with edge direction i, $i \in \{0, 45, 90, 135\}$. $\{D_i\}$ is sorted in ascending order, $D_{i_0} \leq D_{i_1} \leq D_{i_2} \leq D_{i_3}$. Direction i is not considered to be a major direction if D_i is not more than half of the window size,

```
If D_{i_3} ≤ Radius, then
    Direction_1 = -1;    /* major direction does not exist */
else
    Begin
        Direction_1 = i_3;
        If D_{i_2} > Radius, then
            Direction_2 = i_2;
        else
            Direction_2 = -1;    /* does not exist */
    End,
```

where, Direction_1 is the major edge direction, Direction_2 is the second major edge direction.

If the major directions exist, the local window measurements are computed. The final local window measurements are:

$$R = (R_1 + R_2)/2 \tag{4.11}$$

$$C = (C_1 + C_2)/2 \tag{4.12}$$

$$T = (T_1 + T_2)/2 \tag{4.13}$$

In Figure 4.2 (f), there are two major directions: horizontal and vertical. The local measurements are:

$$R_1 = R_2 = 0.0; \quad C1 = C_2 = 0.0; \quad T_1 = T_2 = 0.0;$$

Thus,

$$R = 0.0; \quad C = 0.0; \quad T = 0.0,$$

which shows that the edge in the window is considered to be perfect.

4.1.4 Performance Evaluation Algorithm

After all of the previous discussions, the new performance evaluation algorithm is summarized in this section.

Step 1. Adjust edge direction. The gradient direction (therefore the edge direction) is associated with each edge pixel. To reduce the influence of the error in the edge direction derived by the detector, an adjustment is done before the following evaluation steps. The edge direction of each edge pixel is set to be along the major edge direction in its 3×3 neighbourhood.

Step 2. For each edge neighbourhood, find the major and second major directions in the window, if they exist.

Step 3. If the major direction does not exist, which means the edge pixels in the window are sparse, the center pixel is not considered as potential edge point and assign the local window measurements, namely, R, C and T, for the center pixel to be the worst case (1.0). Go to Step 5.

Step 4. Compute window measurements, namely, roughness R, continuity C and thinness T.

Step 5. If all of the edge neighbourhoods have been examined, go to Step 6. Otherwise, go to Step 2.

Step 6. Compute the aggregate edge quality measure M using Equation (4.7).

4.2 Measurement Curve

The raw output of an edge detector needs to be thresholded to generate the edge map. By using the new evaluation model, the optimal threshold can be chosen to be the one that gives the best performance measurement. Figure 4.3 shows the measurement curve with respect to threshold. The input to the evaluation model is the raw output of the new robust edge detector applied to checkerboard images with

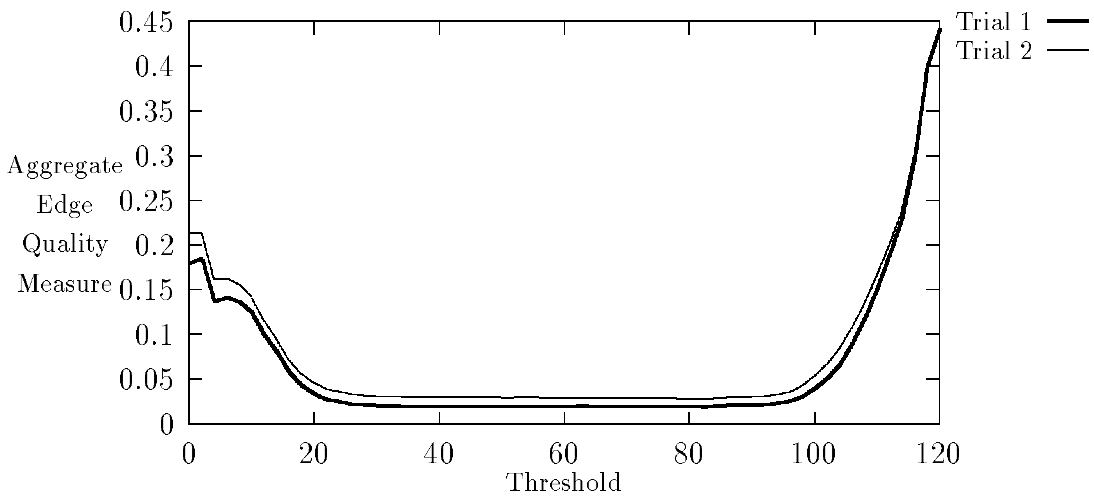


Figure 4.3: The variation of the new measure w.r.t. threshold with various weights.

Gaussian noise level $\sigma = 10$ and outlier percentage 10%. Figure 4.4 shows the original image and the edge map using the threshold obtained by the new model.

In the new model, roughness and continuity are assigned more weight since they are considered to be more important than the other two components, The weights used in the trial tests are:

Trial No.	W_R	W_C	W_T	W_E	Best Threshold
1	0.2	0.6	0.1	0.1	82
2	0.4	0.4	0.1	0.1	82

Table 4.1: Weights used in the new evaluation model.

There are two observations from Figure 4.3. First, the weights used are not too influential. The shape of the curves are similar and the resultant best thresholds are the same. Thus in the rest of the thesis, the weights in the first trial are used.

Second, Figure 4.3 shows that in a certain range of threshold the measurement curve is rather *flat*. The edge quality measure does not have much variations. By

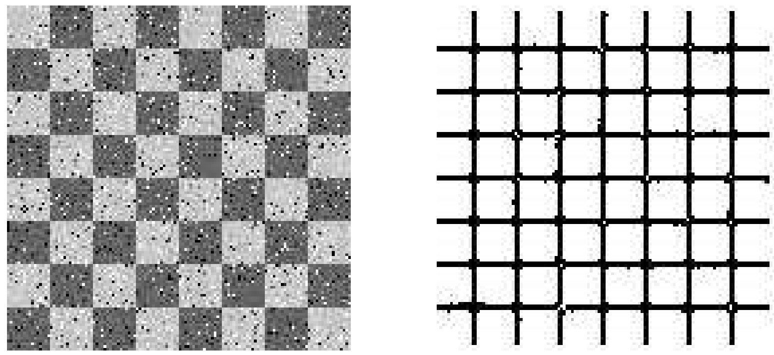


Figure 4.4: Edge map obtained by using threshold which yields the best aggregate edge quality measure.

examining the contrast map obtained by the new robust edge detector, it is discovered that applying a global threshold on the whole image is not a good way to obtain an edge map. There are two main reasons for this.

1. As shown in Figure 4.5, when applying a larger threshold, more pixels are declared as non-edge pixels. Yet the local measures are not any better, since a large threshold also removes good edge point candidates. Figure 4.5 (c) shows the ideal edge map. It is not possible to get Figure 4.5 (c) from Figure 4.5 (a) by using a pre-selected threshold.

2. Edges are local features. When applying a threshold on the whole image, low contrast edges in some areas will be completely invisible. This happens more often in real images because, unlike in synthetic images where edge contrast is fixed (100, in the experiments), real images have edges of various contrast. It is desirable to retain low contrast edges but to eliminate false edges produced by noise.

The above discussion leads to the development of a novel refinement/evaluation scheme which is based on the new evaluation model discussed in Section 4.1. The revised scheme is presented in the next section.

		96	92	
		88	86	
		99	85	
	89	91	98	
		94	97	

(a)

		*	*	
		*		
		*	*	
		*	*	

(b)

		*	*	
		*	*	
		*	*	
		*	*	
		*	*	

(c)

(a) Contrast values at each edge candidate.
(b) Edge pixels after applying threshold 90.
(c) Ideal case.

Figure 4.5: Relations between edge candidates.

4.3 New Refinement/Evaluation (RE) Scheme

The new evaluation scheme discussed in Section 4.1 is revised in this section and becomes a novel refinement/evaluation procedure. Refinement of the edge map is performed while the edge map is evaluated. It takes a raw edge map as input and refines it to get the final edge map. Then the evaluation scheme in Section 4.1 is used to compute the measurement on the final edge map.

The idea behind the refinement/evaluation scheme is to refine the edge map to obtain a better edge quality measure. In this scheme, the edge pixels in the raw edge map are called *edge candidates*. In an evaluation window, the candidate whose existence as an edge pixel will improve the performance measure is given a favourable vote; otherwise, it is given an unfavourable vote. Let N_{uv_i} denote the number of unfavourable votes and N_{fv_i}, the number of favourable votes. N_{uv_i} and N_{fv_i} are computed accumulatively when traversing all of the edge neighbourhoods in an edge map. After all of the edge neighbourhoods are examined, each candidate i is associated with the following ratio:

$$r_i = \frac{N_{uv_i}}{(N_{fv_i} + N_{uv_i})}, \tag{4.14}$$

where $N_{fv_i} + N_{uv_i}$ is the total number of votes. For each candidate, if the unfavourable votes are more than half of the total votes, the candidate is declared as a non-edge pixel. After this process, a refined edge map is obtained.

The refinement/evaluation is done in a similar way as in Section 4.1. The procedure is summarized in detail as follows:

Step 1. Adjust edge direction of each edge candidate as in Section 4.1.4.

Step 2. In each edge candidate's neighbourhood, find the major and second major directions in the neighbourhood, if they exist.

Step 3. If the major direction does not exist, each edge candidate i in this evaluation window is given an unfavourable vote.

$$N_{uv_i} = N_{uv_i} + 1.$$

Go to Step 5.

Step 4. Rotate the evaluation window so that the side of the window is parallel to one major direction. (If two directions exist, this step is done twice.) Do the robust location fit as in Section 4.1.2.

- For each edge candidate that is in the modified band, it is given a favourable vote if the continuity measure in the window is less than a threshold, otherwise, it is given an unfavourable vote. The continuity measure ranges from 0.0 to 1.0. The threshold is chosen to be 0.4, which is chosen arbitrarily.

- For a candidate that is not in the modified band but in the expanded band, there are two cases. The first case is that the other major direction exists and the edge direction of the candidate is the same as the other major direction. In this case, the candidate is given a favourable vote. Otherwise, the candidate only makes the edge more zig-zagged and its existence will make the roughness measure worse. Thus it is given an unfavourable vote.

- For a candidate that is not in the expanded band, no vote is given since it does not affect the performance measure. These candidates normally belong to other nearby edges and will be examined sooner or later.

Step 5. If all of the edge neighbourhoods have been traversed, go to Step 6. Otherwise, go to Step 2.

Step 6. The final edge map is obtained by thresholding the unfavourable vote ratio r in Equation (4.14). In this thesis, 50% is chosen to be the threshold, which means those candidates with vote ratio $r > 50\%$ are declared as non-edge pixels. Other candidates remain as edge pixels.

Step 7. The evaluation scheme in Section 4.1.4 is used to get the quantitative performance measure of the final edge map.

From the above steps, it can be seen that the quality of the refinement procedure is related to the estimation of the edge direction at each candidate. If the edge direction can be estimated more accurately, the refinement procedure can be repeatedly applied until there are no more changes. In this thesis, the refinement procedure is applied to the raw edge map only once.

It should be noted that the refinement procedure decreases the number of edge pixels in the raw edge map. Therefore, the preferable raw edge maps are those that have relatively more edge pixels. The following section shows the experimental results of using this new refinement/evaluation (RE) scheme on both synthetic images and real images.

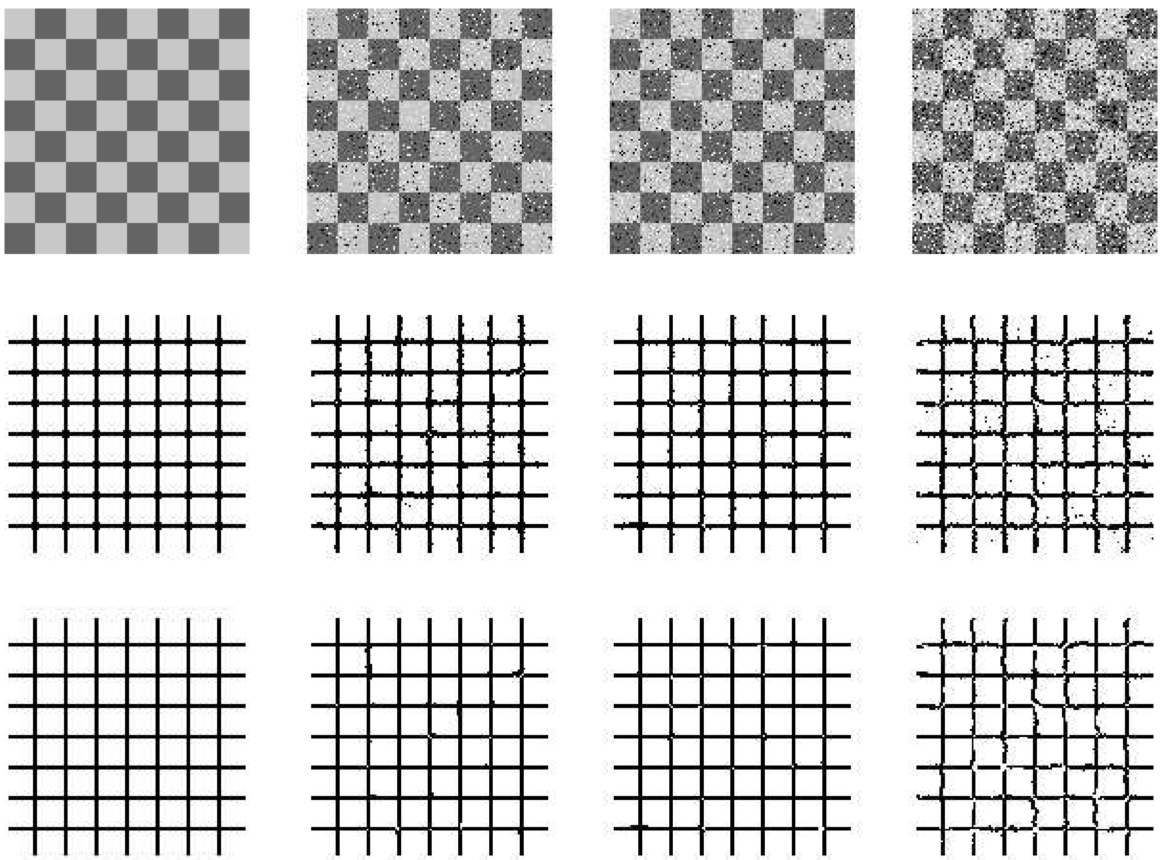
4.4 On Synthetic Images

4.4.1 Comparison of the FCR Model and the New RE Scheme

To study the effect of the refinement procedure, the new refinement/evaluation procedure is applied to the edge maps generated by the FCR model. Figures 4.6, D.1 and D.2 show the experimental results. The new robust edge detector is used in this set of experiments. The first row of these figures is the original image. The second row is the edge map generated by thresholding the contrast map using the FCR model. The third row is the edge map obtained by applying the new RE scheme on the second row. The effect of the refinement procedure can be easily seen. Notice the first column in Figure 4.6. The original image is noiseless. The edge map obtained by the FCR model has extra edge pixels at the corners. After the refinement procedure, the resultant edge map in the third row is visually perfect. Images in other columns in Figure 4.6 have different levels of noises. The comparison between the second and third row reveals that the new refinement scheme has a strong ability to remove edge pixels that form zig-zaggedness in the edge map. The thick and zig-zagged edges in the edge maps are refined to be thinner, smoother and cleaner. Comparison of the FCR model and the RE model on the crossboard and ring input images can be found in Appendix D. The next section presents a quantitative comparison of the performance of different edge detection schemes using the measurement obtained by the new refinement/evaluation model.

4.4.2 Quantitative Comparison of Different Edge Detection Schemes

This section presents the quantitative performance comparison between the Haralick operator, RCBS, the $tanh$ operator, the robust $tanh$ operator and the new robust edge detector, all using the new refinement/evaluation model proposed in Section 4.3.



(1) noiseless (2) $\sigma =\ 0,\ \epsilon = 10\%$ (3) $\sigma = 10,\ \epsilon = 10\%$ (4) $\sigma = 10,\ \epsilon = 30\%$

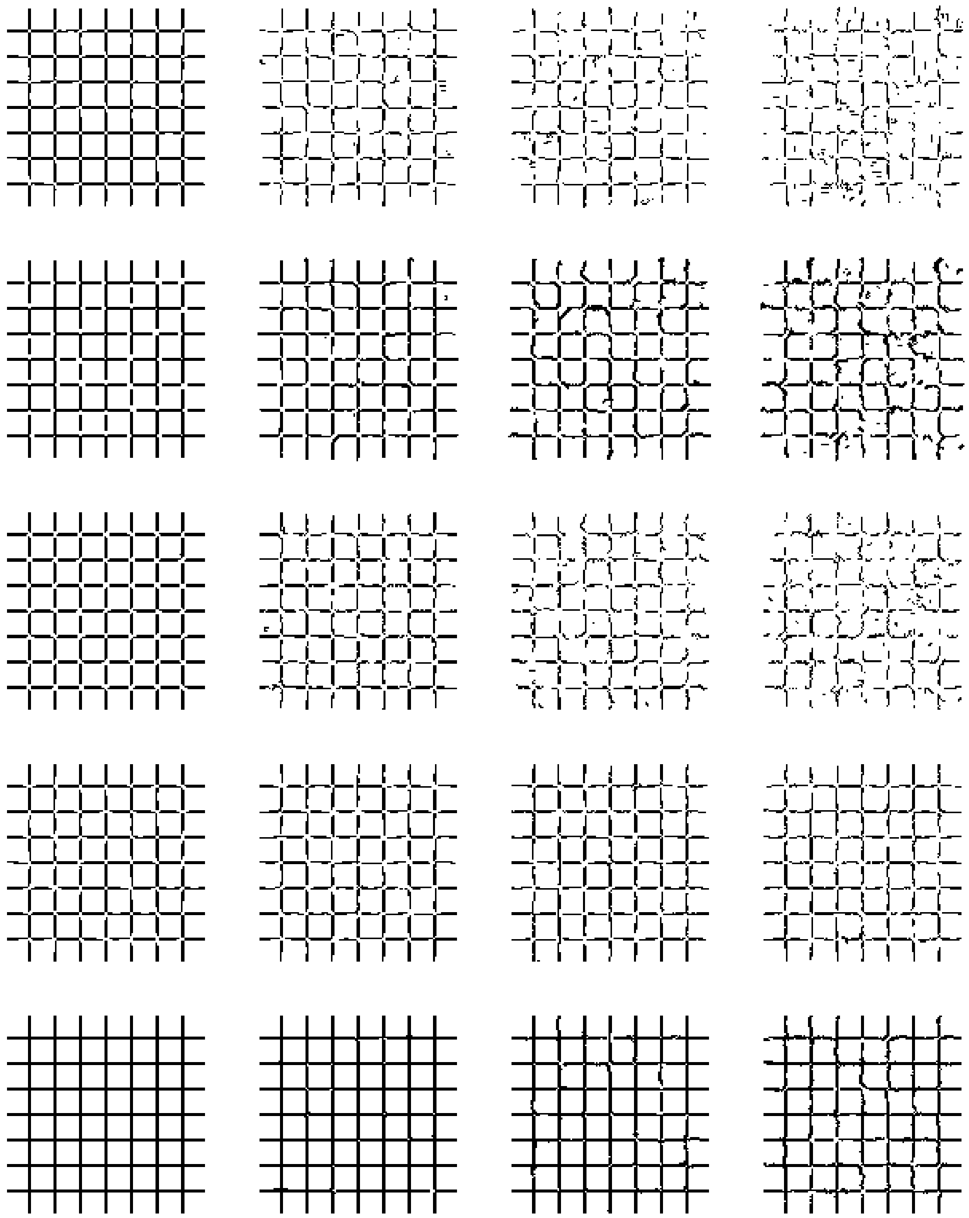
Figure 4.6: Comparison of the FCR model and the RE model on checkerboard.

Sample edge maps are shown in Figure 4.7. Figure 4.8 shows the measurement curves with respect to the outlier levels. The underlying Gaussian noise level is $\sigma = 10$. The observations are similar to that obtained in Section 3.5.5. The RCBS operator, the robust $tanh$ operator and the new robust edge detectors perform better than the Haralick operator and the $tanh$ operator. The new robust operator performs systematically better than RCBS and the robust $tanh$ operator. The curve for the robust $tanh$ operator is not well separated from that of the RCBS operator. This is because the robust $tanh$ operator tends to create "weak" non-continuous edges as mentioned in Section 3.5.5. After the refinement, many of these edge pixels are removed. This can also be observed from Figure 4.7. The new robust edge detector does not suffer from this problem. It generates well-connected edges and performs the best when compared with other detectors.

Figure 4.9 shows the measurement curves with respect to different Gaussian noise levels. The underlying outlier level is $\epsilon = 30\%$. Conclusions are similar to that in Section 3.5.5. The new robust edge detector systematically performs better than the robust $tanh$ operator. When the outlier is the major kind of noise, they perform better than the RCBS operator. When Gaussian noise is dominant, RCBS performs better. More experimental results using the crossboard and ring input images can be found in Appendix E.

4.5 On Real Images

The performance of the new refinement/evaluation model on synthetic images shows that the RE model is appropriate. The motivation for developing this new model is to use it on a real images. This section presents the work done in applying the new RE model on real images.



(a)$\epsilon = 0\%$ (b) $\epsilon = 10\%$ (c) $\epsilon = 20\%$ (d) $\epsilon = 30\%$

First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the *tanh* operator.
Fourth row: outputs of the robust *tanh* operator.
Fifth row: outputs of the new edge detector.

Figure 4.7: Comparison of checkerboard outputs (new model), $\sigma = 10$.

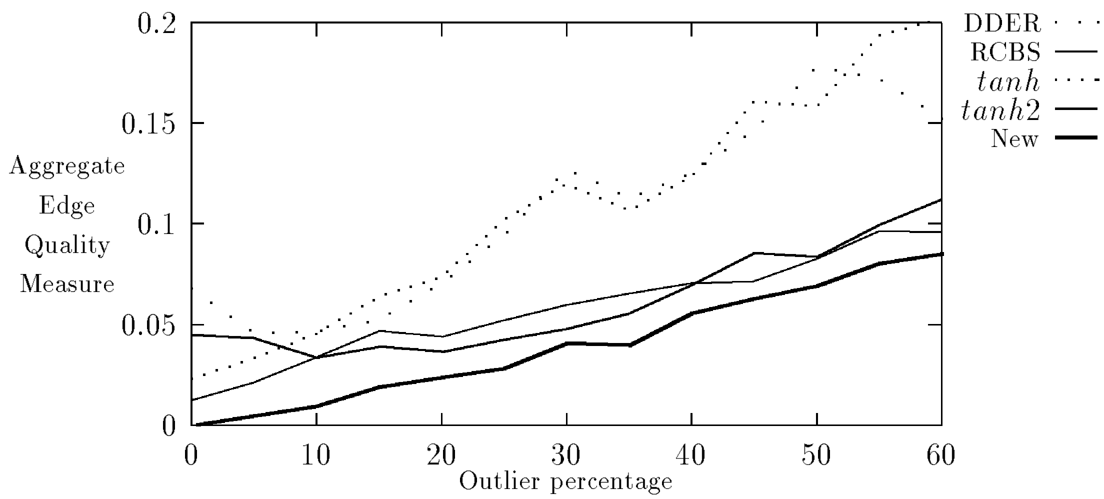


Figure 4.8: New measure w.r.t. outlier level on checkerboard ($\sigma = 10$).

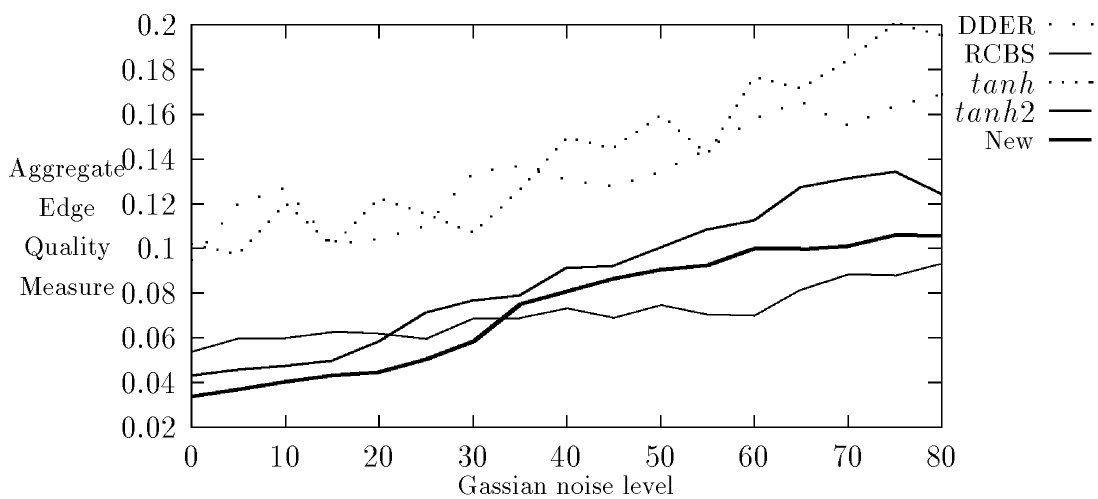


Figure 4.9: New measure w.r.t. Gaussian noise on checkerboard ($\epsilon = 30$).

4.5.1 Experimental Procedure

The Haralick operator, RCBS, the *tanh* operator and the new robust edge detector are applied to real images in this section. (The robust *tanh* operator is not used here because of its excessive computational time.) Their outputs are the gradient or contrast maps. Raw edge maps are required to apply the new refinement/evaluation scheme. Since the RE scheme only removes edge pixels, edge maps with more edge pixels are preferable. A relatively small threshold is used on the raw output of edge detectors to get the raw edge map. It is desirable to select the initial threshold T without human interaction, the procedure of which is described in the following.

Let I denote the set of all pixels in the image, V_i, the value of pixel i in the raw output, E_t, the set of edge pixels after applying threshold t and N_s, the number of pixels in set s. Thus,

$$E_t = \{i \mid V_i > t,\ i \in I\}. \tag{4.15}$$

The ratio, r, of edge pixels, N_{E_t}, with the total number of pixels, N_I, in an image is defined to be:

$$r(t) = N_{E_t} \ / \ N_I \tag{4.16}$$

$r(t)$ is obviously decreasing w.r.t. increasing t. Figure 4.10 shows the ratio curves versus threshold using different operators on the bottle image shown in Figure 4.11(1). It is noticed that the curve decreases sharply at first and then decreases more slowly. As t increases, the initial threshold, T, is selected such that $r(T-1) - r(T) < dr$, where dr is a pre-selected threshold on the difference of ratio. Though dr is a new threshold, experiments show that the results are not sensitive to the value selected. The value at each pixel in a gradient map is the estimated value of the first derivative. In a contrast map, the value at each pixel is the estimated edge contrast value. Thus, different thresholds dr are assigned to contrast maps and gradient maps. For contrast maps (output of the *tanh* operator and the new robust operator), dr is always set to

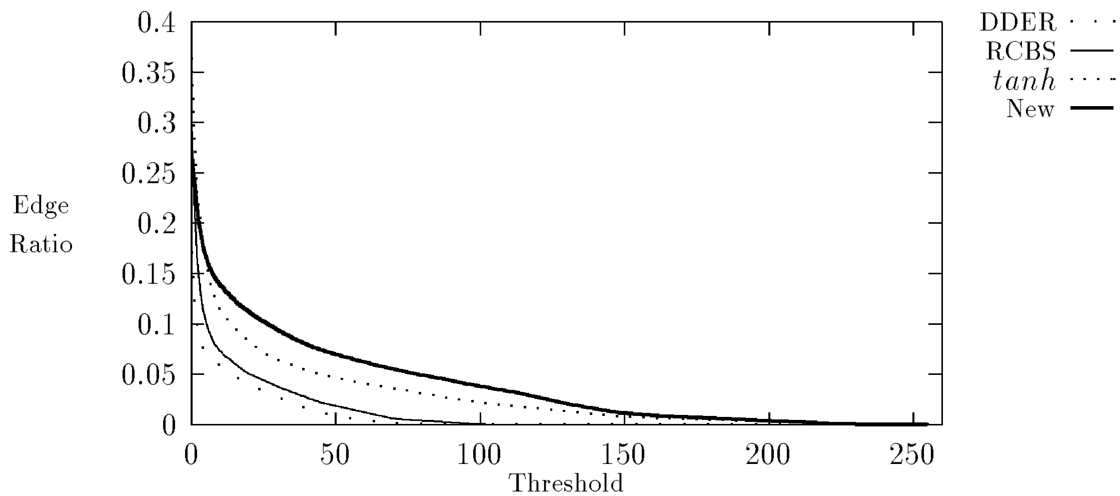
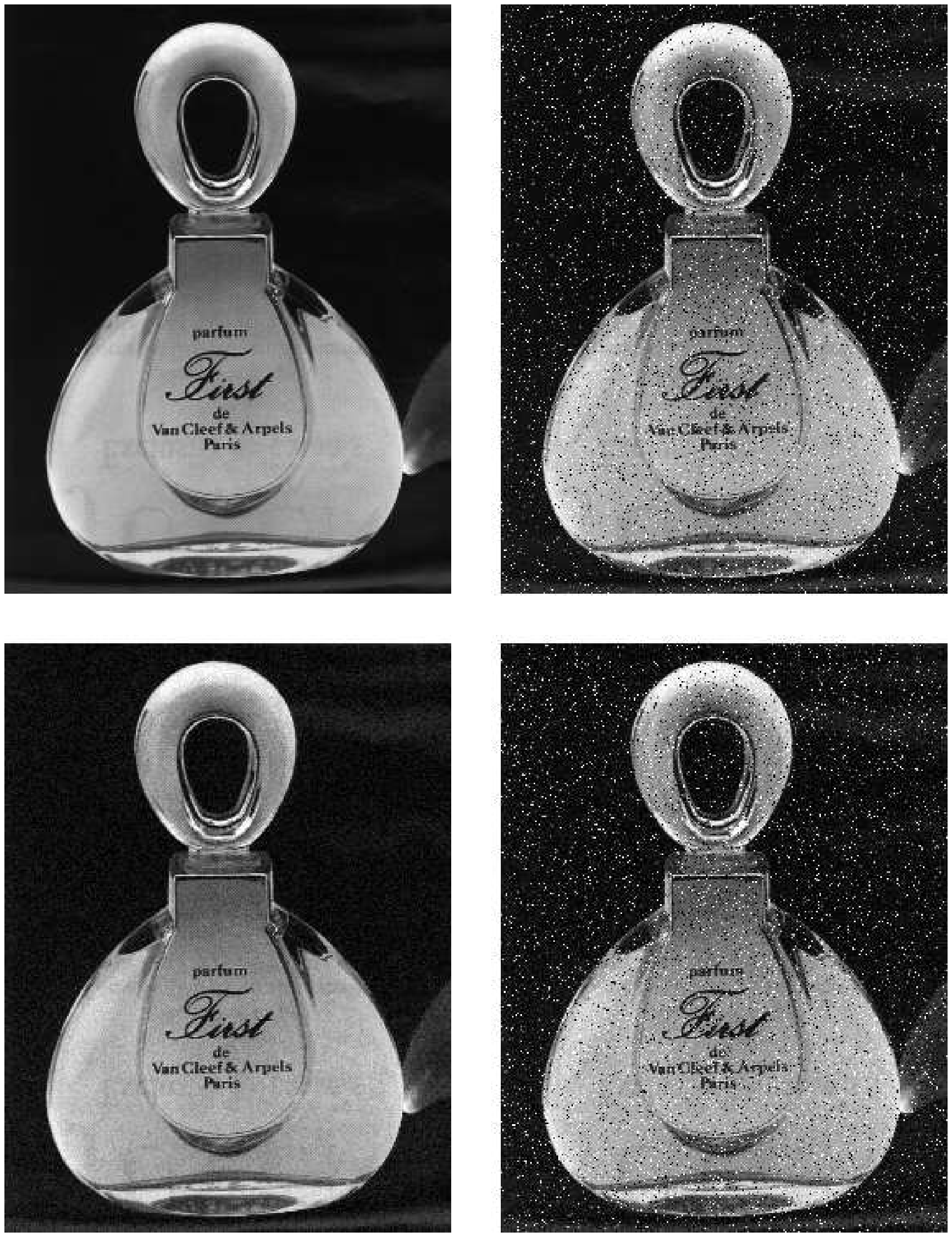


Figure 4.10: The ratio of edge pixels in an image v.s. threshold.

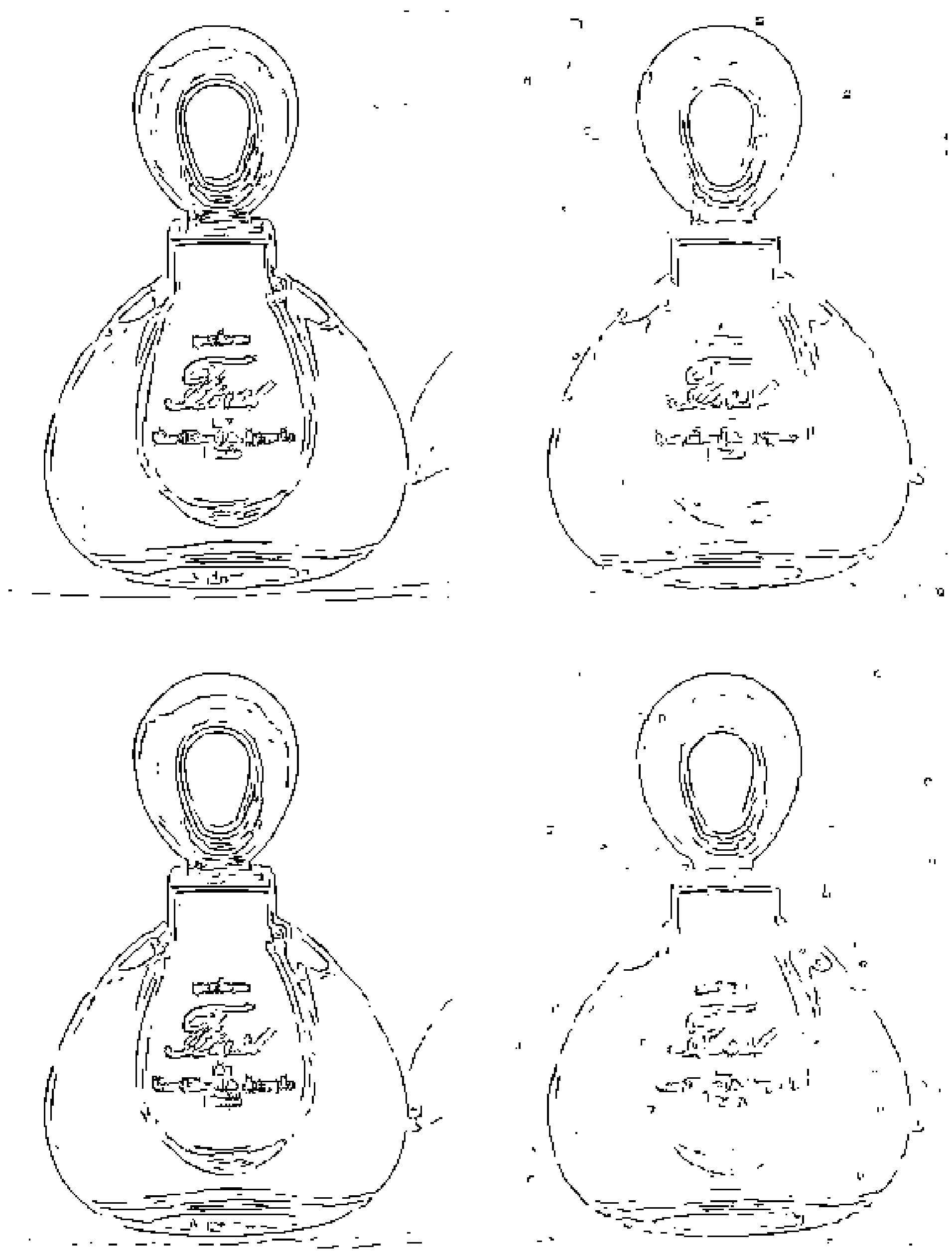
be 0.2%. For gradient maps (output of the Haralick operator and the RCBS operator), $dr = 0.5\%$. This threshold is image independent.

To evaluate the performance of edge detectors on real images w.r.t. noise, different levels of the Gaussian and outlier noise are added to the bottle image, just like in synthetic images. Noise levels are the same as that described in Section 3.5.1. There are in total 221 images. Sample original images are shown in Figure 4.11. (Experimental results on another real image can be found in Appendix F). In this figure, no artificial Gaussian noise is added in the fist row. Gaussian noise level in the second row is $\sigma = 10$. The artificial outlier level is 0% for the left column and 10% for the right column. The Haralick operator, RCBS, the $tanh$ operator and the new robust edge detector are used on all real images (after adding noise). The final edge maps are obtained by using the new refinement/evaluation scheme.



(1) No additional noise (2) $\sigma = \ 0, \epsilon = 10\%$
(3) $\sigma = 10, \epsilon = 0\%$ (4) $\sigma = 10, \epsilon = 10\%$

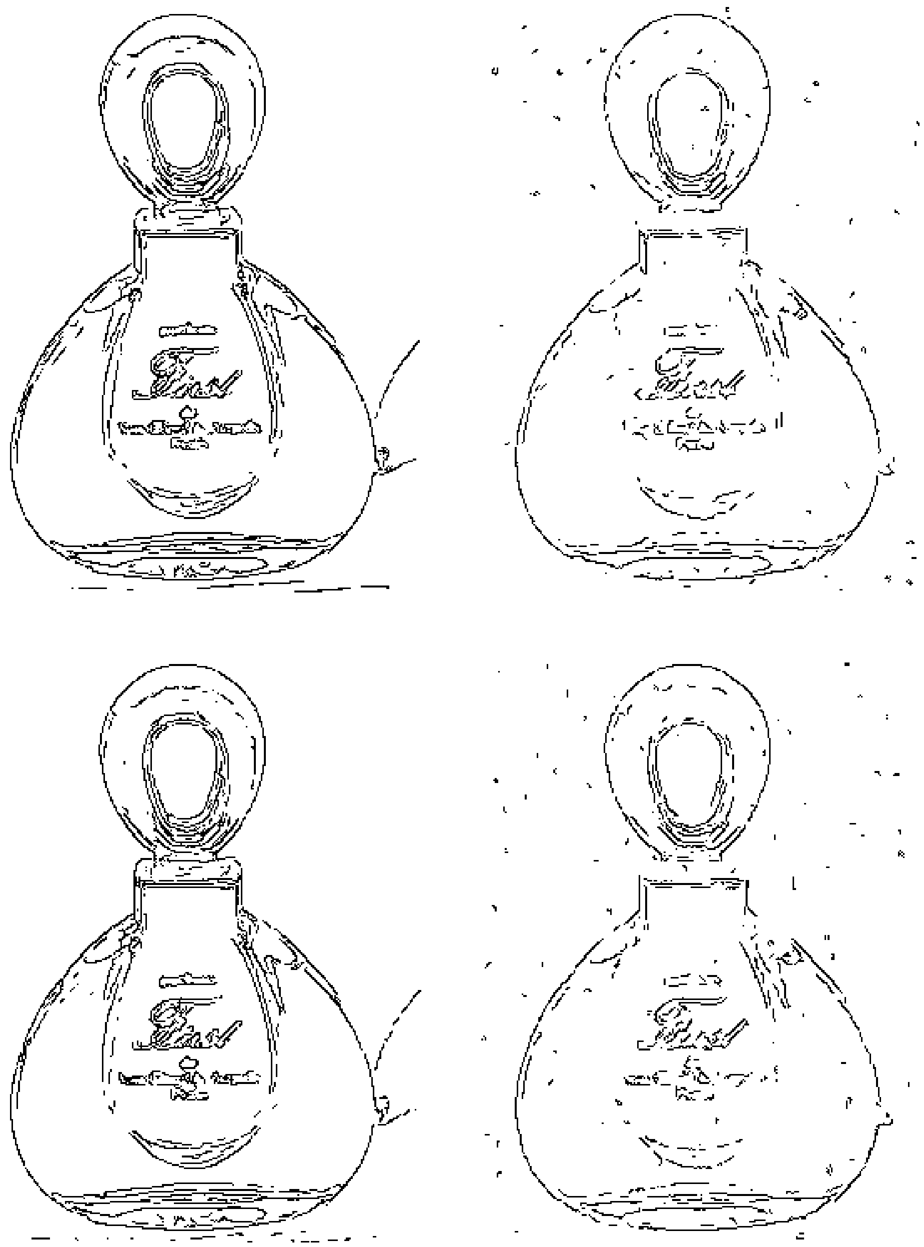
Figure 4.11: Real images (bottle).



Corresponding new edge quality measure:

(1)	0.0700	(2)	0.0990
(3)	0.0703	(4)	0.1033

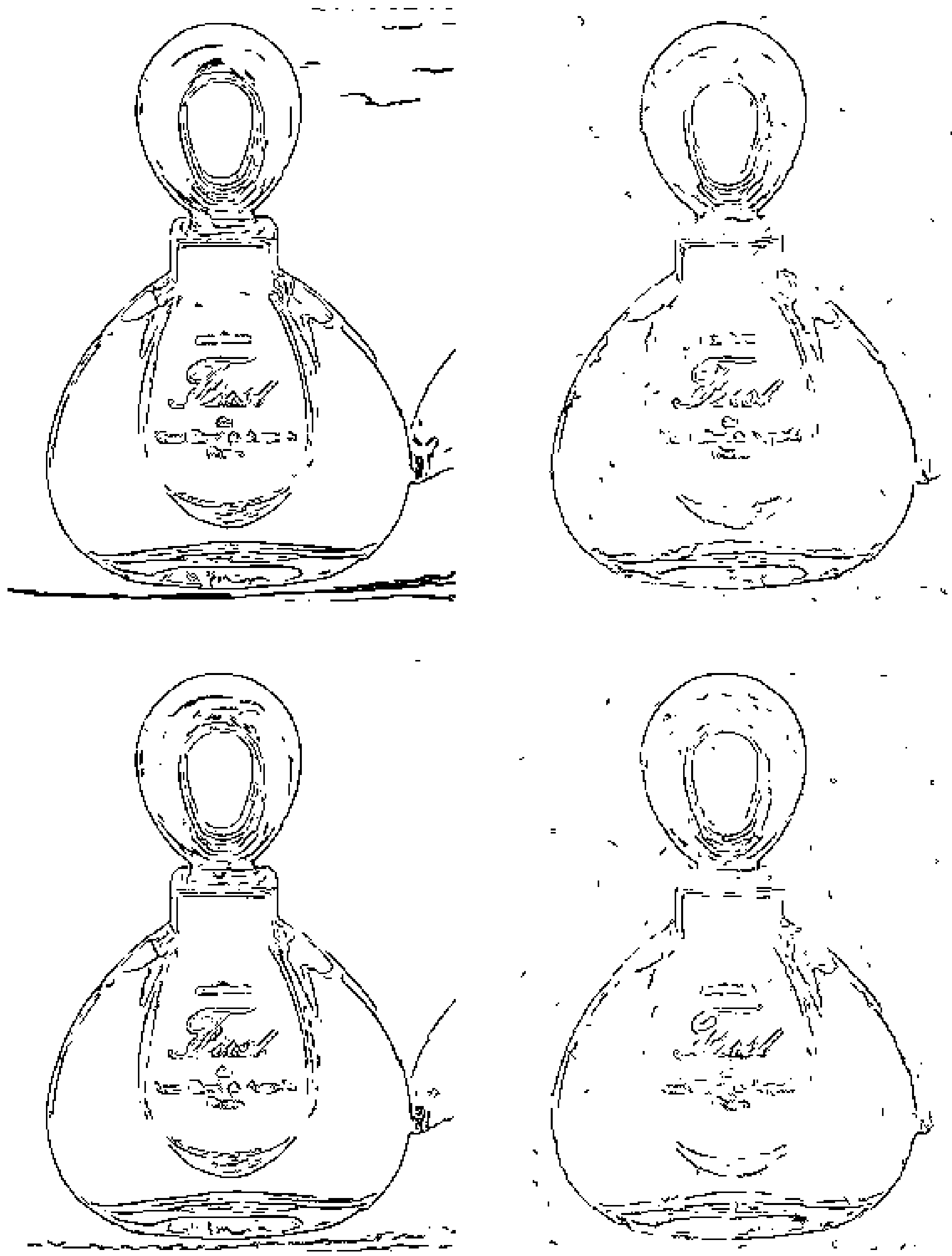
Figure 4.12: Outputs of the Haralick's operator on bottle image.



Corresponding new edge quality measure:

(1)	0.0668	(2)	0.0821
(3)	0.0716	(4)	0.0946

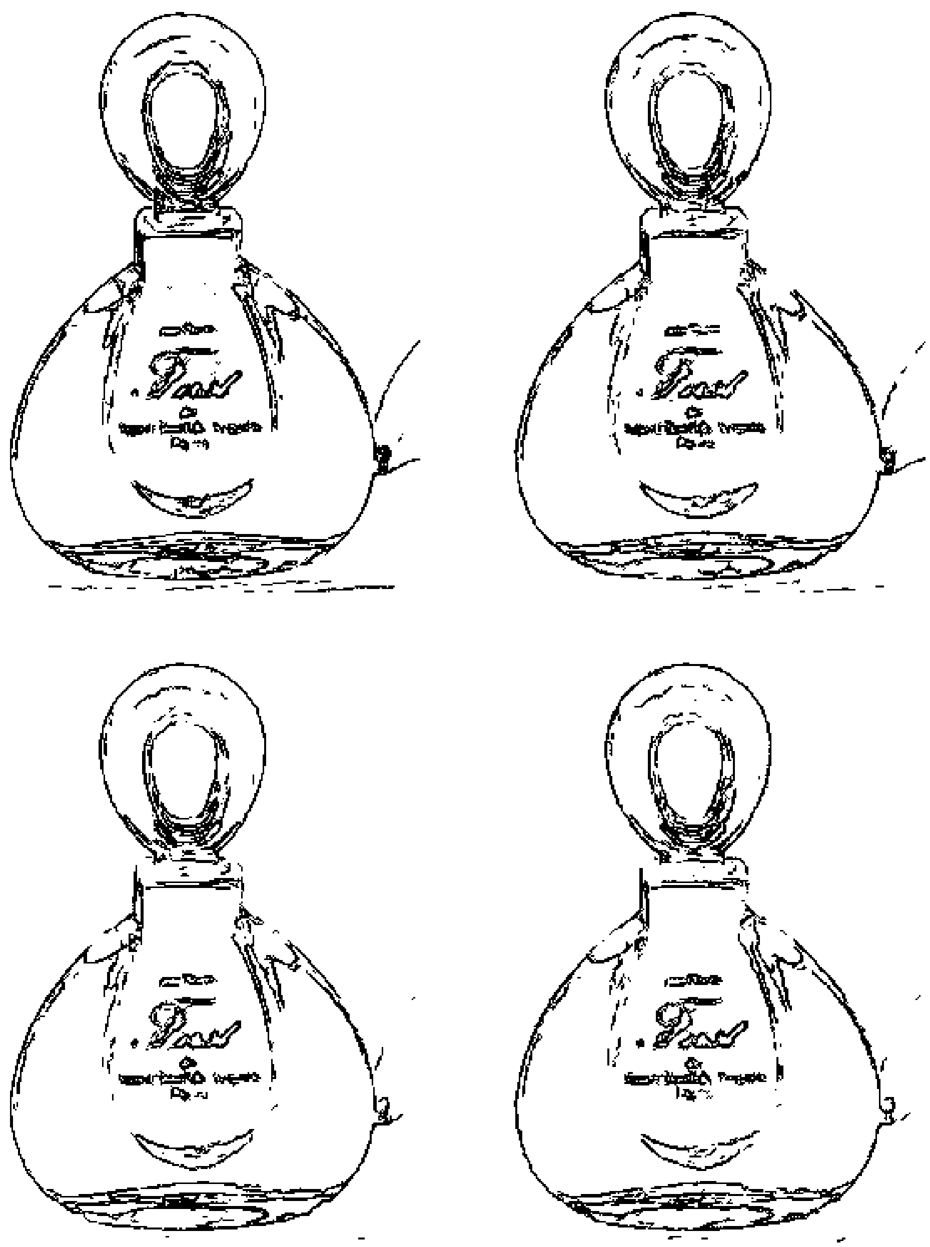
Figure 4.13: Outputs of the RCBS operator on bottle image.



Corresponding new edge quality measure:

(1)	0.0626	(2)	0.0868
(3)	0.0631	(4)	0.0939

Figure 4.14: Outputs of the *tanh* operator on bottle image.



Corresponding new edge quality measure:

(1)	0.0639	(2)	0.0710
(3)	0.0700	(4)	0.0721

Figure 4.15: Outputs of the new operator on bottle image.

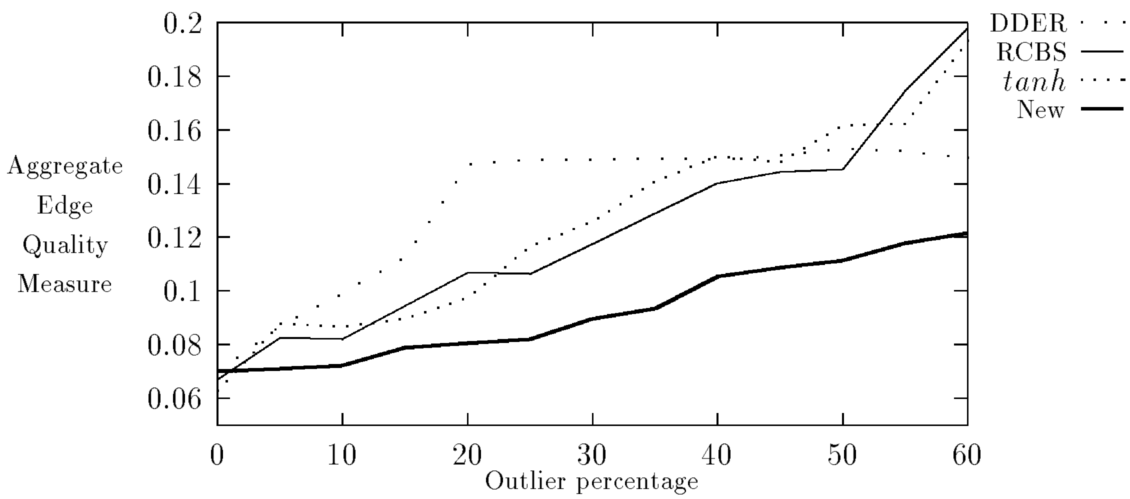


Figure 4.16: New measure w.r.t. outlier level on bottle image ($\sigma = 0$).

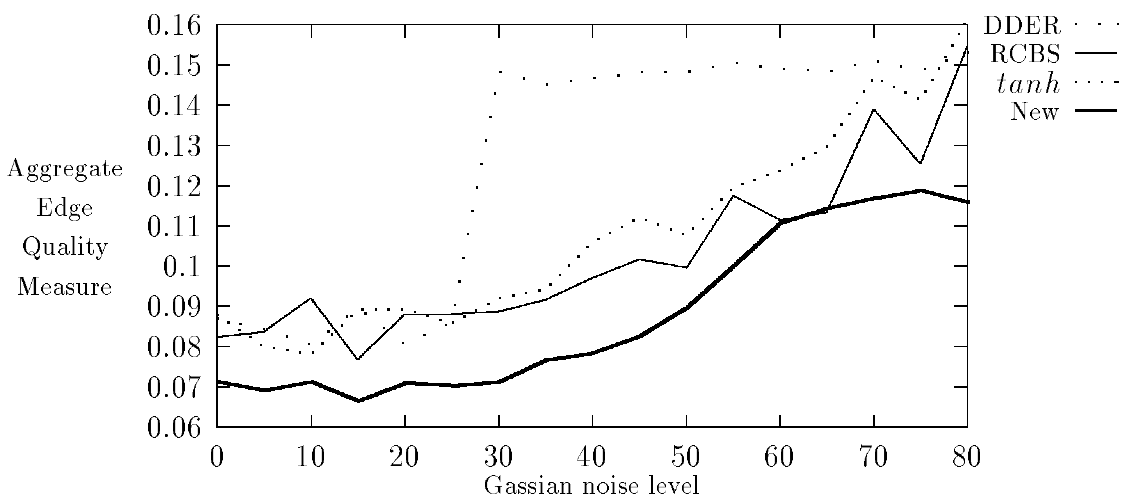


Figure 4.17: New measure w.r.t. Gaussian noise on bottle image ($\epsilon = 5$).

4.5.2 Comparison of Edge Maps

Figures 4.12, 4.13, 4.14 and 4.15 show the edge maps corresponding to the images shown in Figure 4.11. All of the edge maps are generated without human interactions. It is shown that when there are outliers, the new robust edge detector obviously demonstrates its ability in dealing with outliers and performs best when compared with all the other operators. If only Gaussian noise is present (edge maps in left column), RCBS and the $tanh$ operator perform better.

Comparing the top two edge maps in Figure 4.15 with edge maps in Figure 3.20(k) and (l), the effect of the refinement procedure can be clearly seen.

4.5.3 Quantitative Comparison of Edge Detection Schemes

Figure 4.16 shows the measurement curves with respect to outlier percentage. In this figure, no Gaussian noise is added. It can be seen that when the outlier level is 0%, RCBS and the $tanh$ operator perform slightly better than the new robust detector. But when outliers exist, the new detector clearly performs better than other operators. RCBS and the $tanh$ operator perform roughly the same or better than the Haralick operator. The result also conforms with human visual justification described in the previous section.

Figure 4.17 shows the measurement curves with respect to Gaussian noise. The underlying outlier percentage is 5%. The new edge detector demonstrates its strong ability to suppress noise and performs better than RCBS and the $tanh$ operator. RCBS and the $tanh$ operator have similar performance.

4.6 Summary

In this chapter, a new quantitative performance evaluation model is proposed. The proposed new model is portable and is edge detection scheme independent. Its intention is to provide a common framework for quantitative performance evaluation and comparison between different edge detection schemes on both synthetic and real images. More importantly, it is the first effort towards the scientific evaluation of edge detection on real images.

The new model is compared with the FCR model on synthetic images. Experiments show that the new model can be used to refine edge maps to be thinner, smoother and cleaner. With refinements, quantitative performance comparison between the Haralick operator, RCBS, the *tanh* operator and the new edge detector is conducted on both synthetic images and real images. It is concluded that the new edge detector has a strong ability to suppress noise and performs the best in the presence of outliers.

Chapter 5

Conclusions and Future Work

In this thesis, a survey on major contemporary edge detection approaches and evaluation schemes was conducted. The major problem in edge detection is that images are often corrupted by various kinds of noise. Most existing edge detection schemes are developed to deal with images corrupted by Gaussian noise. They have poor performance when the noise does not conform to the Gaussian model.

Robust statistics provides methods for dealing with unco-operative data. In this thesis, the *least-median-of-squares* (LMS) estimator in robust statistics is proposed to be incorporated in edge detection schemes. The solution of the LMS location estimator is the narrowest strip covering half of the data points and is easy to compute. When there is more than one regression coefficient to be determined, the random sampling technique is used to get the approximate LMS solution. Experiments show that LMS fitting employing random sampling technique performs much better than conventional LSS fitting in the presence of outliers. LMS technique is proposed to be added to the $tanh$ fitting operator. The resultant robust $tanh$ operator is shown to have better noise immunity. Another independent new robust edge detector is also proposed, which is based on the LMS location estimator. This new edge detector is fast and has a high breakdown point of 50%. The performance of the proposed edge detectors on synthetic images is studied using the FCR evaluation model. Compared with the Haralick operator, RCBS and the *tanh* operator, the two new robust detectors achieve better noise immunity in the presence of outliers. The new robust detector performs better than the robust *tanh* operator. When the Gaussian noise level is high and is the major source of noise, RCBS performs better than the new robust detectors, which shows that RCBS has a strong capability to suppress Gaussian noise.

The FCR model can only be used on synthetic images where true edge locations are all known. Work has been done to develop a new evaluation scheme to evaluate the performance of edge detectors on real images. A new quantitative evaluation model is proposed, which contains a refinement step to get a final edge map from a raw edge map. The performance measure has four components, namely, roughness, thinness, continuity and the edge density. The computation of the measurement involves LMS location estimation. This new model can be used on both synthetic images and real images. On synthetic images, the edge maps generated by the new model are smoother, thinner and cleaner. On real images, the edge maps can be obtained using the new model without human interaction. The resultant edge map has about the same visual effect when compared with the edge map with threshold manually chosen. Quantitative performance comparison between different edge detection schemes is conducted. Experiments are done on three types of synthetic images and also on real images. Each set of synthetic image has 221 images with a wide range of noise levels. Gaussian noise and outliers are also added to real images. The experimental results based on these numerous tests are considered to be reliable and accurate. These results lead to the conclusion that the new robust edge detector, with refinement, has a strong ability to suppress noise and performs the best in the presence of outliers compared with other detectors.

Below is a list of contributions from this thesis work:

- A general technique of using LMS employing random sampling to robustify edge detection algorithms is proposed. This technique is applied to the $tanh$ operator and the resultant operator is more robust in the presence of outliers.

- A novel robust edge detector based on LMS location estimator is proposed. This detector has a strong ability to suppress noise as well as producing continuous edges. It is fast and has a breakdown point of 50%.

- A new quantitative evaluation model is proposed. It is also based on LMS location estimator in robust statistics. An edge refinement procedure is part of this quantitative evaluation model which yields edge maps with better visual results. One most important contribution is that this evaluation model can be used not only on synthetic images but also on real images. The edge maps are generated without human interactions. And the evaluation measure can be used to quantitatively compare the performance of edge detectors.

This thesis work explores the application of robust statistics in edge detection and evaluation. The results are encouraging. Yet there is still much work to be done. Images containing more types of edges like roof edges can be constructed to test the newly proposed edge detectors. The new detector may be modified to improve its ability to detect finer edges and adjacent edges in real images. Suitable edge linking and thinning techniques can be further explored. In the new evaluation model, the relative importance of different components can be studied. The edge direction at each pixel can be estimated more accurately so that the new refinement/evaluation scheme can be applied repeatedly to obtain better results. Finally, work can be done to modify the new evaluation model and make it able to take the gradient/contrast map as input. Therefore the choice of an initial threshold to obtain the raw edge map can be removed.

Bibliography

[1] I. E. Abdou and W. K. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proc. of the IEEE*, 67(5):753–763, 1979.

[2] F. Bergholm. Edge focusing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(6):726–741, 1987.

[3] V. Berzins. Accuracy of Laplacian edge detectors. *Computer Vision, Graphics, and Image Processing*, 27:195–210, 1984.

[4] D. J. Bryant and D. W. Bouldin. Evaluation of edge operators using relative and absolute grading. In *Proc. IEEE Comp. Soc. Conf. on Pattern Recognition and Image Processing*, pages 138–145, Chicago, Aug. 1979.

[5] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[6] G. Chen and Y. H. Yang. *Edge Detection by Regularized Cubic B-Spline Fitting.* Research Report, #91-8, Computer Vision Lab, Dept. of Computational Science, University of Saskatchewan, 1991.

[7] J. S. Chen and G. Medioni. Detection, localization, and estimation of edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(2):191–198, 1989.

[8] M. H. Chen, D. Lee, and T. Pavlidis. Residual analysis for feature detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(1):30–39, 1991.

[9] L. S. Davis. A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4:248–270, 1975.

[10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[11] J. R. Fram and E. S. Deutsch. On the quantitative evaluation of edge detection schemes and their comparison with human performance. *IEEE Trans. on Computers*, C-24(6):616–628, 1975.

[12] M. Gokmen and C. C. Li. Edge detection using iteratively refined regularization. In *Proc. 10th International Conference on Pattern Recognition*, pages 690–693, Atlantic City, June 1990.

[13] M. Gokmen and C. C. Li. Edge detection using refined regularization. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 215–221, Hawaii, 1991.

[14] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Inc., Reading, Mass, 1977.

[15] W. E. L. Grimson and T. Pavlidis. Discontinuity detection for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 30:316–330, 1985.

[16] R. M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):58–68, 1984.

[17] E. C. Hildreth. The detection of intensity changes by computer and biological vision systems. *Computer Vision, Graphics, and Image Processing*, 22:1–27, 1983.

[18] L. Kitchen and Z. Rosenfeld. Edge evaluation using local edge coherence. *IEEE Trans. on Systems, Man, and Cybernetics*, 11(9):597–605, 1981.

[19] D. Lee. Edge detection, classification, and measurement. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2–10, San Deigo, California, 1989.

[20] D. Lee and T. Pavlidis. One-dimensional regularization with discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(6):822–829, 1988.

[21] D. Lee, T. Pavlidis, and K. Huang. Edge detection through residual analysis. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 215–222, 1988.

[22] M. D. Levine. *Vision in Man and Machine*. McGraw-Hill, New York, 1985.

[23] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Royal Society London, ser. B*, 207:187–217, 1980.

[24] P. Meer, D. Mintz, and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.

[25] P. Meer, D. Mintz, and A. Rosenfeld. Analysis of the least median of squares estimator for computer vision applications. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 621–623, Champaign, Illinois, June 1992.

[26] D. Mintz. Robust consensus based edge detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 651–653, Champaign, Illinois, June 1992.

[27] V. S. Nalwa and T. O. Binford. On detecting edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):699–714, 1986.

[28] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[29] J. M. S. Prewitt. Object enhancement and extraction. In *Picture Processing and Psychopictorics, B. S. Lipkin and A. Rosenfeld*, Academic Press, New York, 1970.

[30] L. G. Roberts. Machine perception of three-dimensional solids. In *Optical and Electro-Optical Information Processing, J.T. Tippett et al.,Eds.*, pages 159–197, The MIT Press, 1965.

[31] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*. Academic Press, New York, 1982.

[32] P. J. Rousseeuw and A. M. Leroy. *Robust Regression & Outlier Detection*. John Wiley & Sons, New York, 1987.

[33] B. G. Schunck. *Robust Computational Vision*. Technical Report, CSE-TR-60-90, Dept. of Electrical Engineering and Computer Science, University of Michigan, 1990.

[34] S. S. Sinha and B. G. Schunck. A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(1):36–55, 1992.

[35] V. Torre and T. A. Poggio. On edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(2):147–163, 1986.

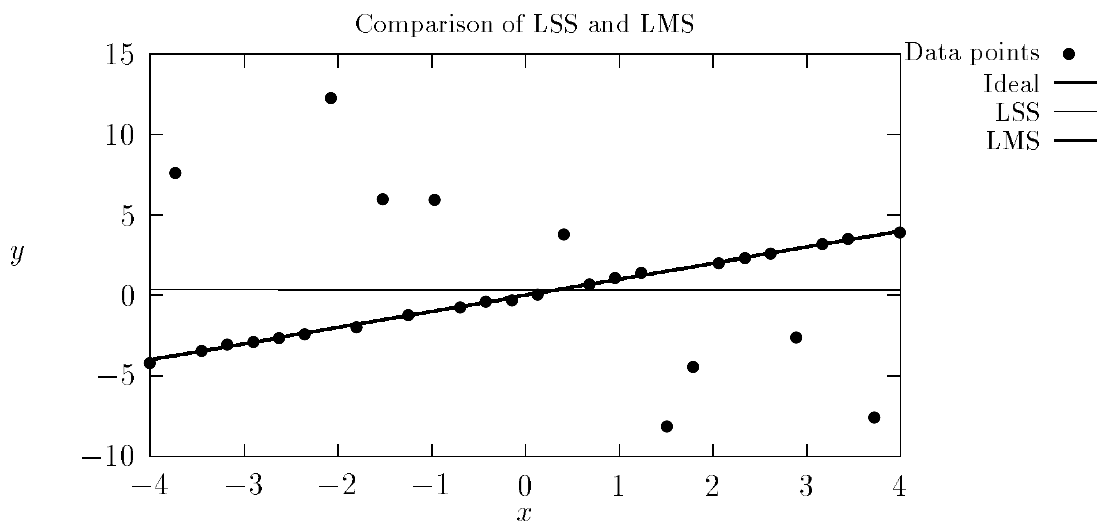
[36] F. Ulupinar and G. Medioni. Refining edges detected by a LoG operator. *Computer Vision, Graphics, and Image Processing*, 51:275–298, 1990.

[37] S. Venkatesh and L. J. Kitchen. Edge evaluation using necessary components. *Computer Vision, Graphics, and Image Processing*, 54(1):23–30, 1992.

Appendix A

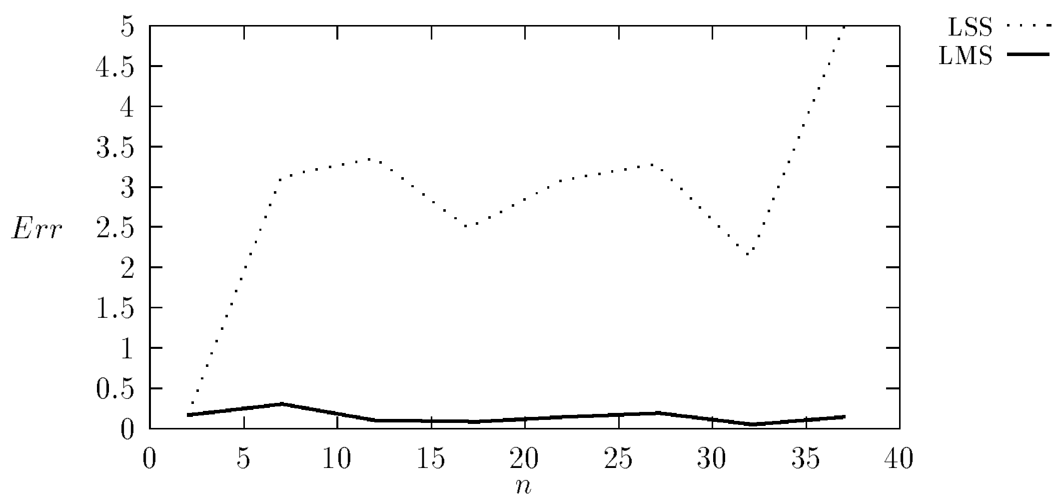
Performance of LSS and LMS

This appendix shows the experimental results of comparing the performance of the LSS and the LMS estimator when applied on polynomials of different order. The random sampling technique is used to get the LMS solution. The results show that LMS performs much better than LSS in the presence of outliers.



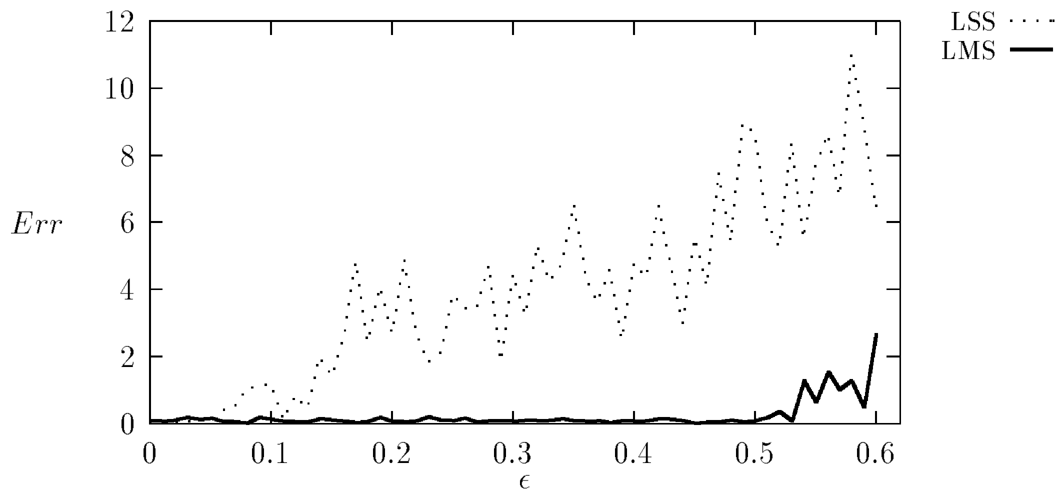
$y = 1.0x,\ n = 30,\ \sigma = 0.2,\ \epsilon = 30\%$

Figure A.1: Fitted curves using LSS and LMS ($p = 2$).



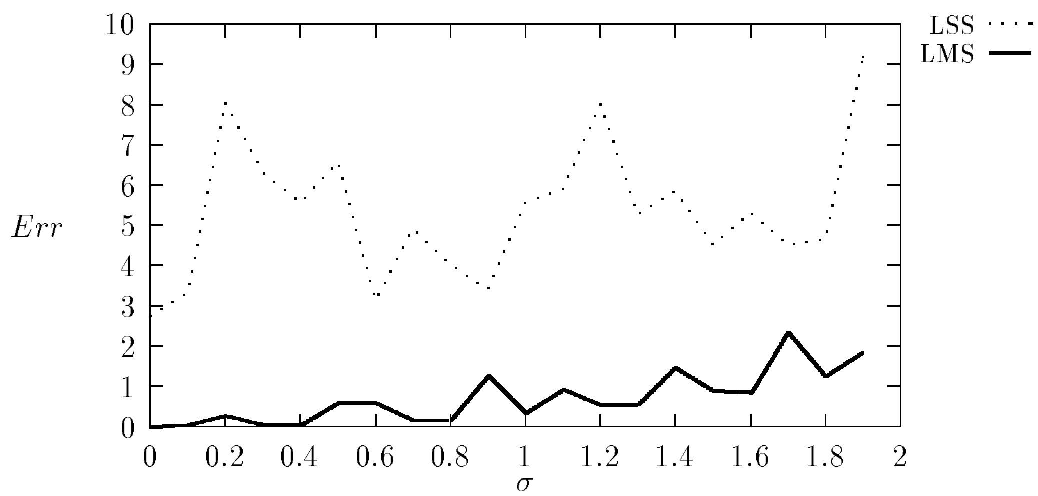
$y = 1.0x,\ \sigma = 0.1,\ \epsilon = 40\%$

Figure A.2: Comparison of LSS and LMS with Err vs. n ($p = 2$).



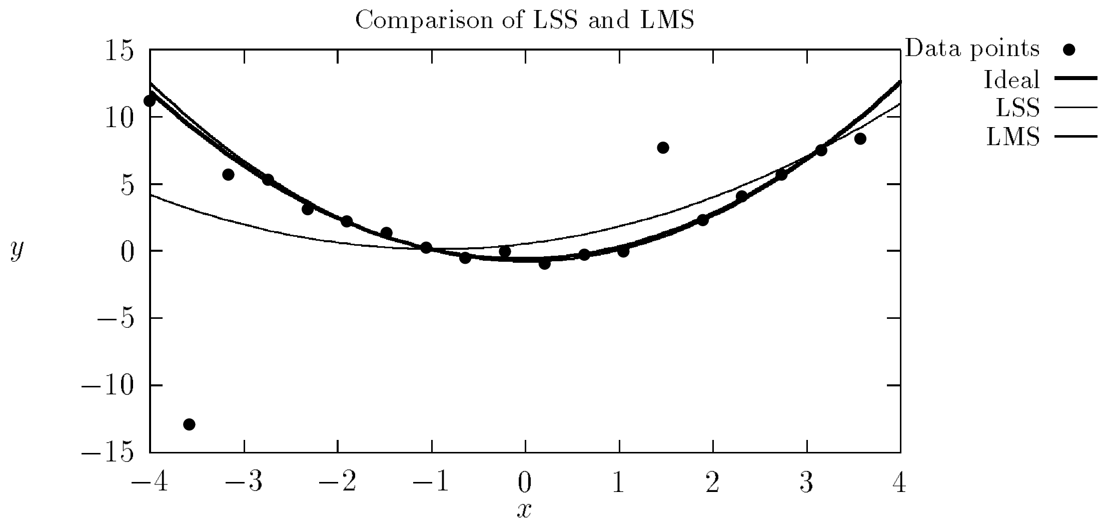
$y = 1.0x, \; n = 30, \; \sigma = 0.1$

Figure A.3: Comparison of LSS and LMS with Err vs. ϵ $(p = 2)$.



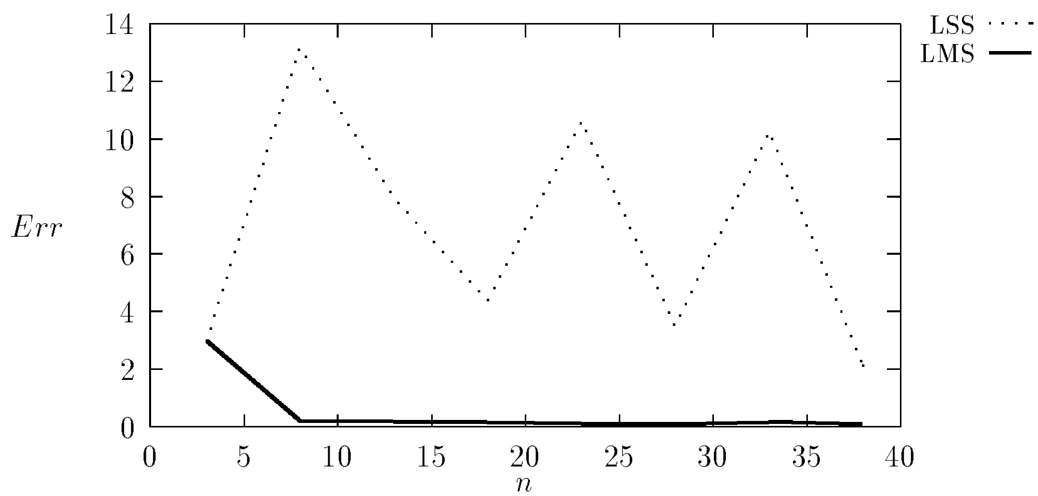
$y = 1.0x, \; n = 30, \; \epsilon = 30\%$

Figure A.4: Comparison of LSS and LMS with Err vs. σ $(p = 2)$.



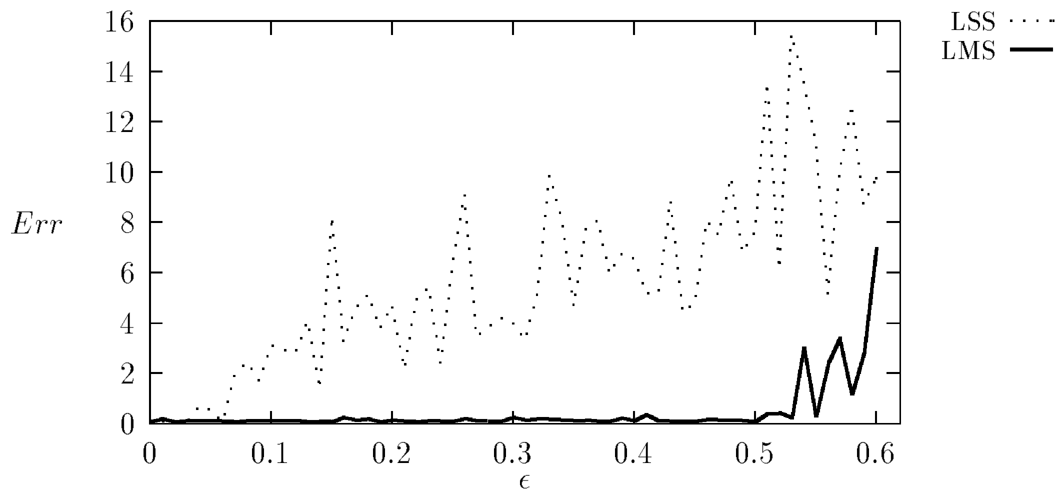
$$y = 0.8x^2 + 0.1x - 0.6,\ n = 20,\ \sigma = 0.3,\ \epsilon = 20\%$$

Figure A.5: Fitted curves using LSS and LMS ($p = 3$).



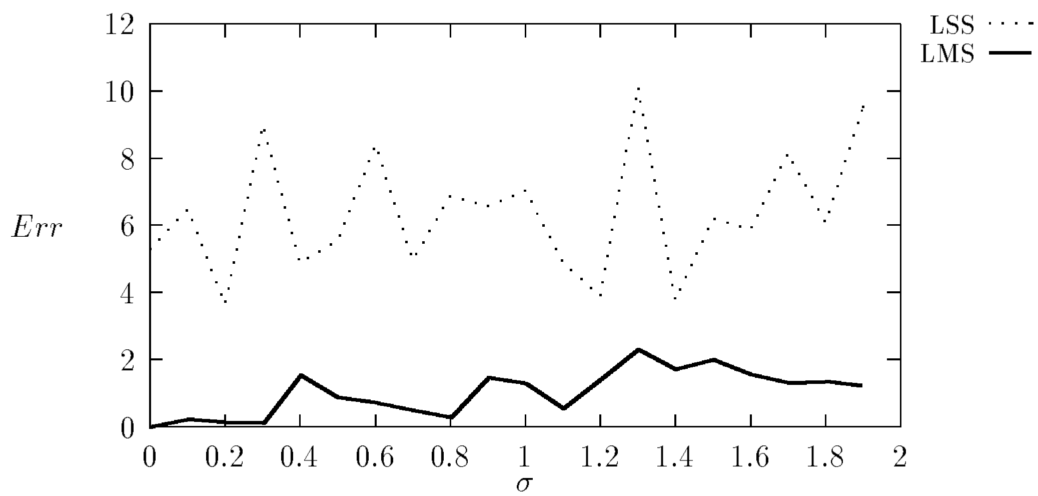
$$y = 0.8x^2 + 0.1x - 0.6,\ \sigma = 0.1,\ \epsilon = 40\%$$

Figure A.6: Comparison of LSS and LMS with Err vs. n ($p = 3$).



$y = 0.8x^2 + 0.1x - 0.6,\ n = 30,\ \sigma = 0.1$

Figure A.7: Comparison of LSS and LMS with Err vs. ϵ $(p = 3)$.



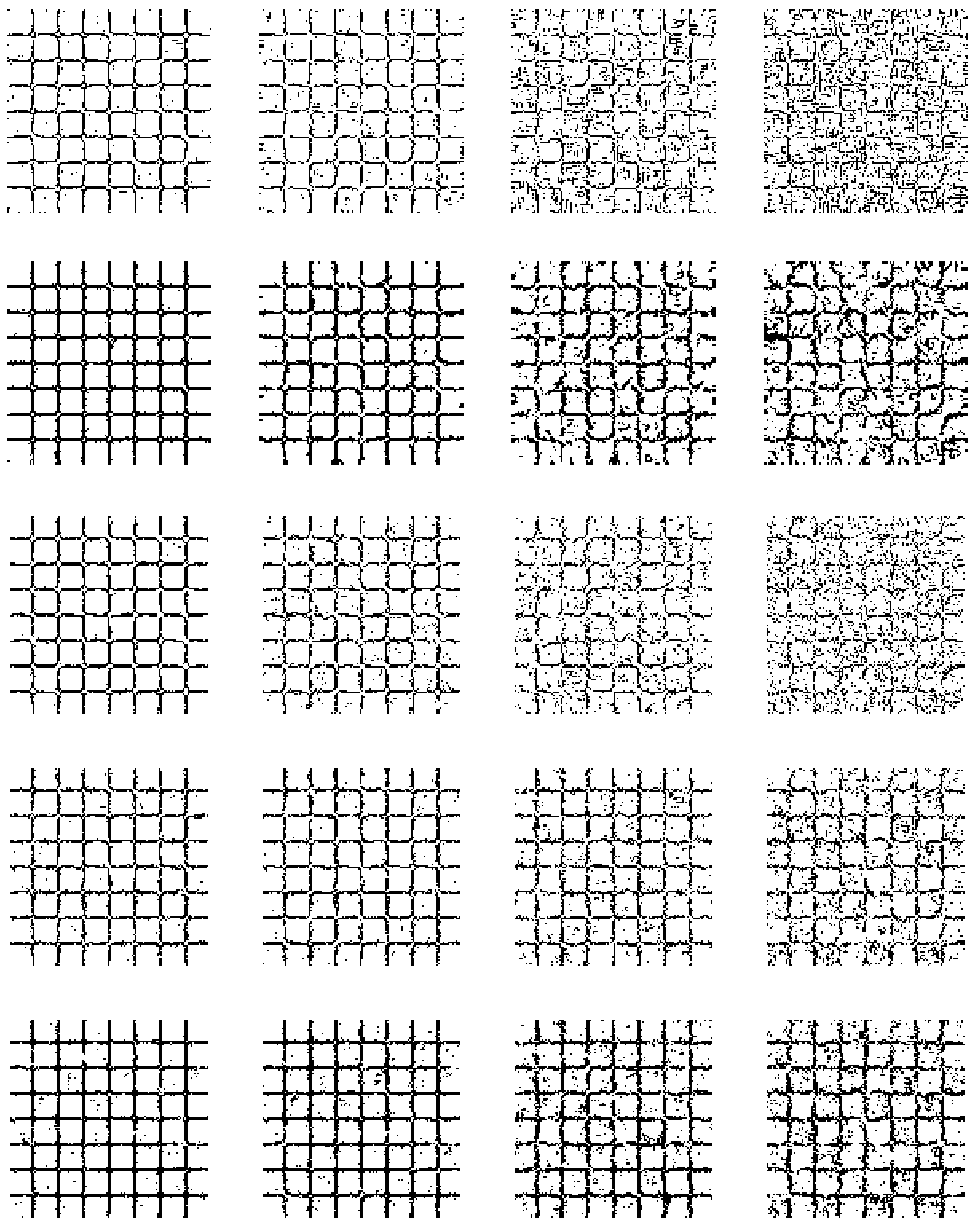
$y = 0.8x^2 + 0.1x - 0.6,\ n = 30,\ \epsilon = 30\%$

Figure A.8: Comparison of LSS and LMS with Err vs. σ $(p = 3)$.

Appendix B

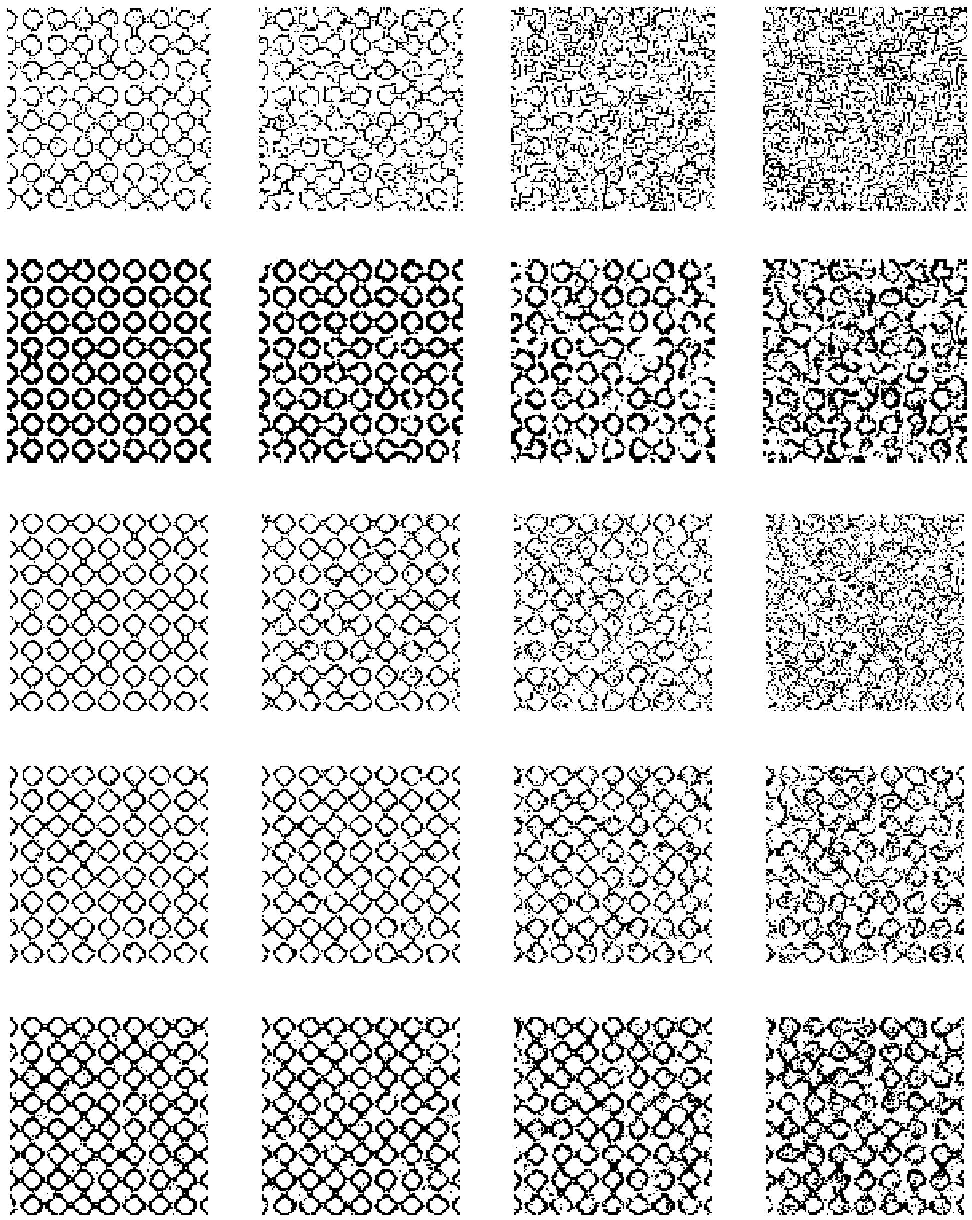
Comparison of Image Outputs Using the FCR model

This appendix shows the edge outputs of different edge detectors (using the FCR model to choose the optimal threshold).



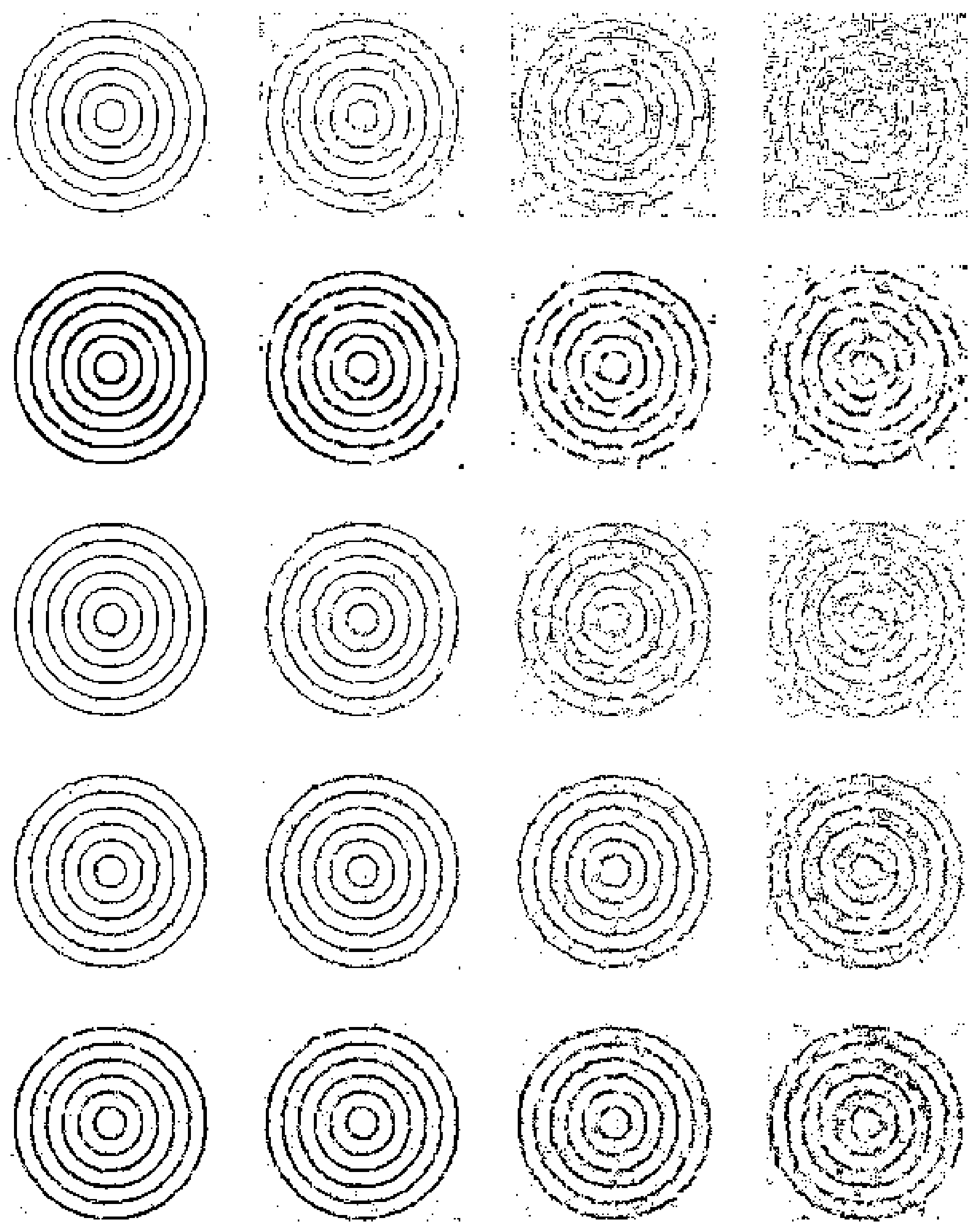
First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the $tanh$ operator.
Fourth row: outputs of the robust $tanh$ operator.
Fifth row: outputs of the new edge detector.

Figure B.1: Comparison of checkerboard outputs (FCR model), $\sigma = 30$.



First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the *tanh* operator.
Fourth row: outputs of the robust *tanh* operator.
Fifth row: outputs of the new edge detector.

Figure B.2: Comparison of crossboard outputs (FCR model), $\sigma = 30$.



First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the *tanh* operator.
Fourth row: outputs of the robust *tanh* operator.
Fifth row: outputs of the new edge detector.

Figure B.3: Comparison of ring outputs (FCR model), $\sigma = 30$.

Appendix C

Quantitative Comparison Using the FCR Model

This appendix presents the experimental results of quantitative comparison of different detectors. The FCR model is used to choose the optimal threshold for each detector.

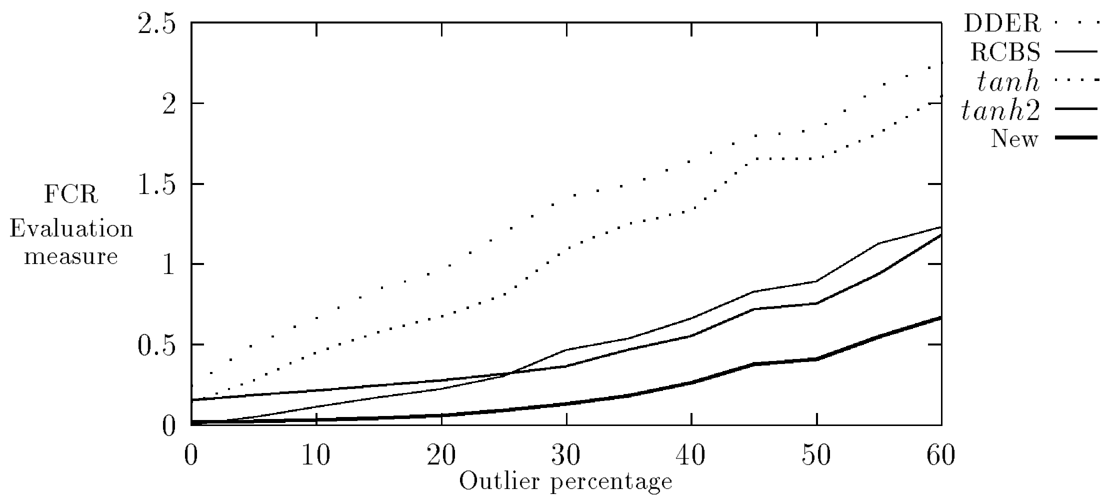


Figure C.1: FCR measure w.r.t. outlier level on checkerboard ($\sigma = 20$).

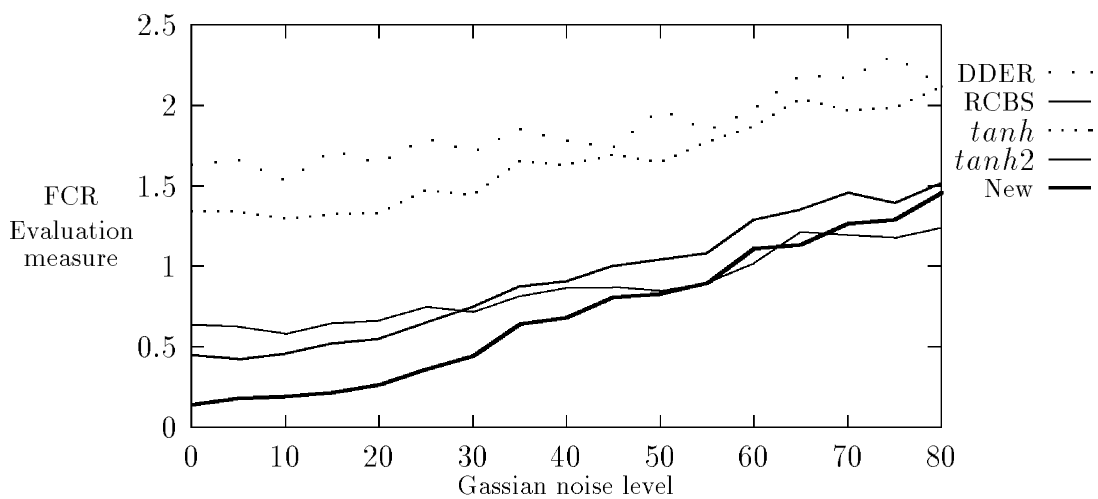


Figure C.2: FCR measure w.r.t. Gaussian noise on checkerboard ($\epsilon = 40$).

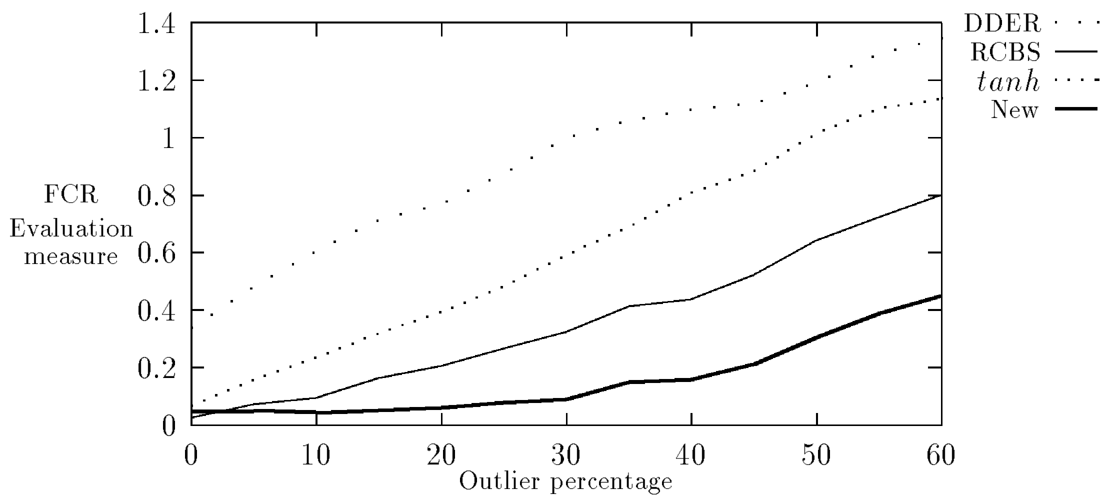


Figure C.3: FCR measure w.r.t. outlier level on crossboard ($\sigma = 10$).

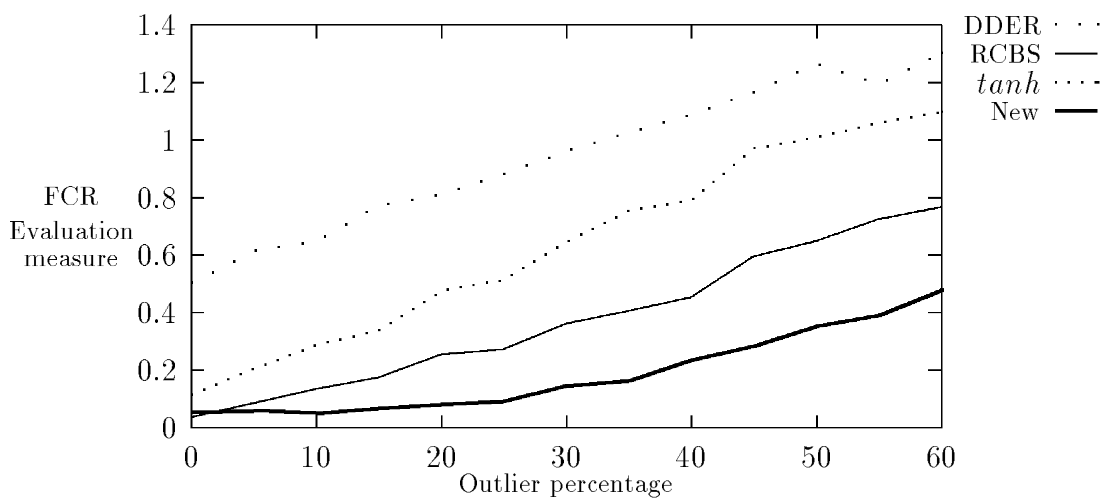


Figure C.4: FCR measure w.r.t. outlier level on crossboard ($\sigma = 20$).

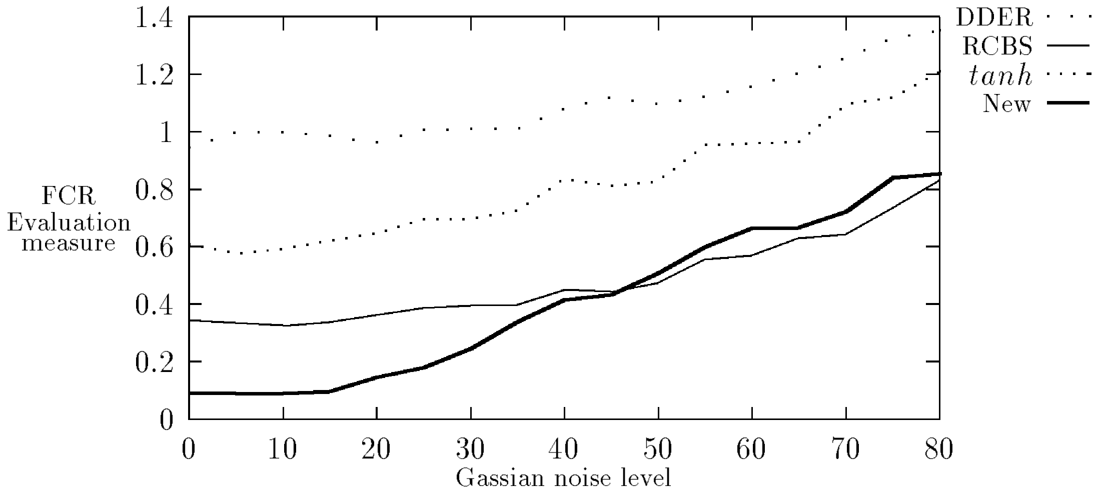


Figure C.5: FCR measure w.r.t. Gaussian noise on crossboard ($\epsilon = 30$).

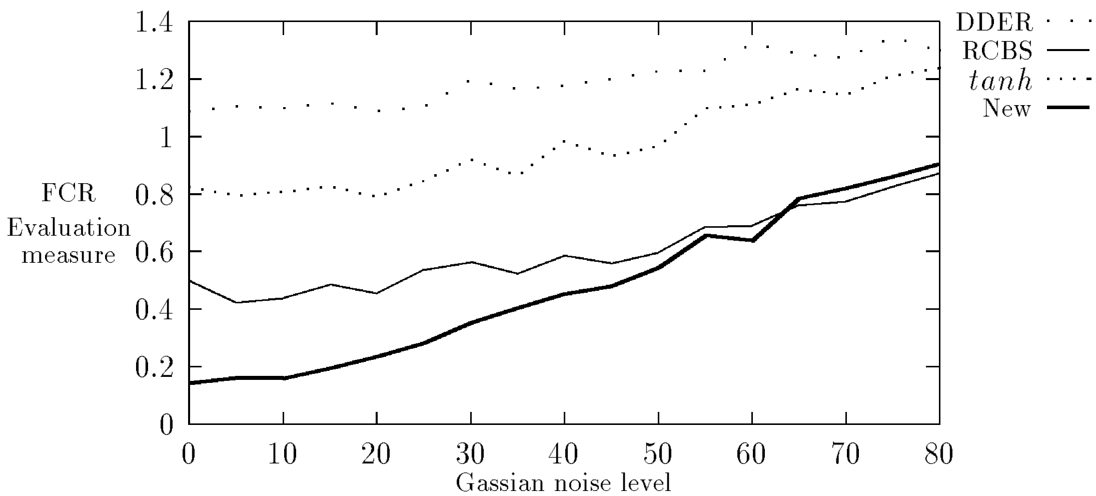


Figure C.6: FCR measure w.r.t. Gaussian noise on crossboard ($\epsilon = 40$).

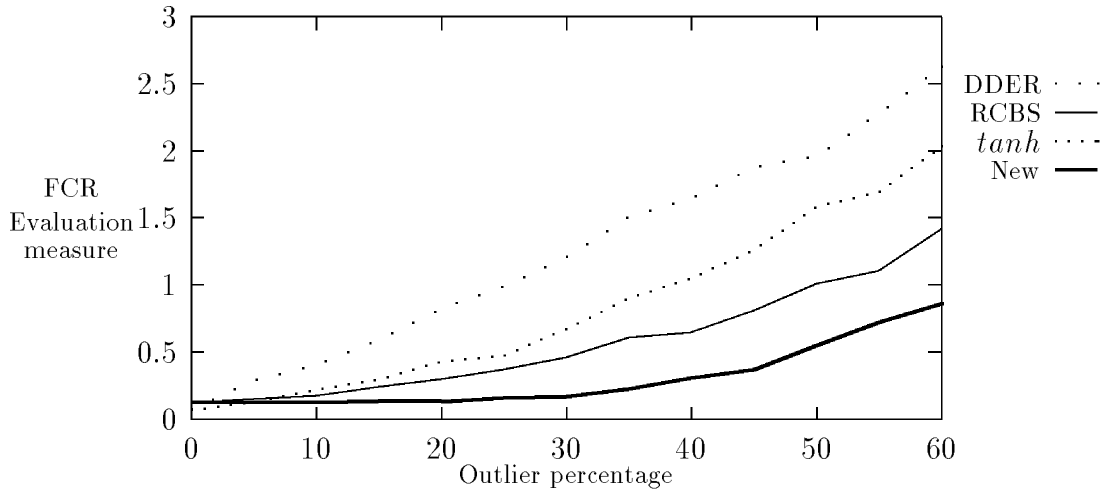


Figure C.7: FCR measure w.r.t. outlier level on rings ($\sigma = 10$).

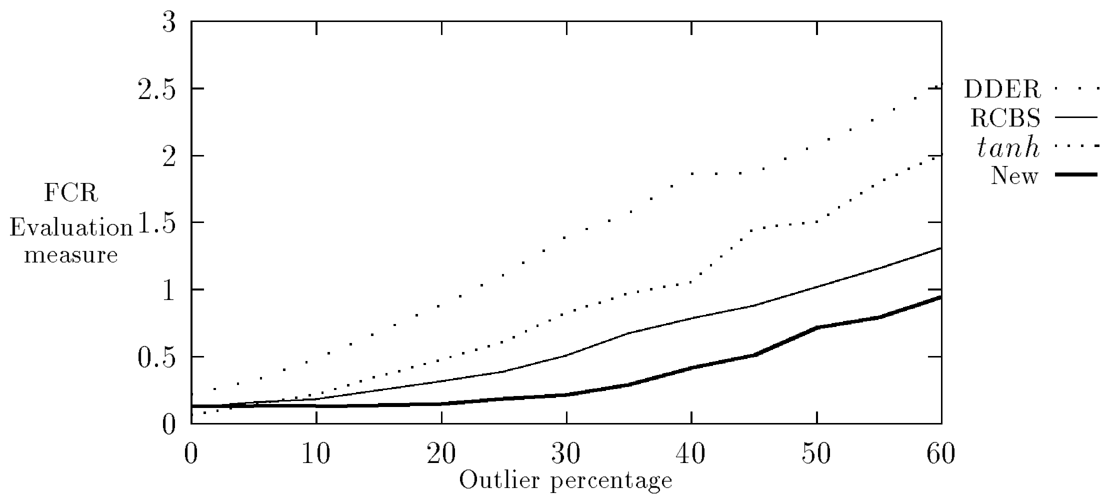


Figure C.8: FCR measure w.r.t. outlier level on rings ($\sigma = 20$).

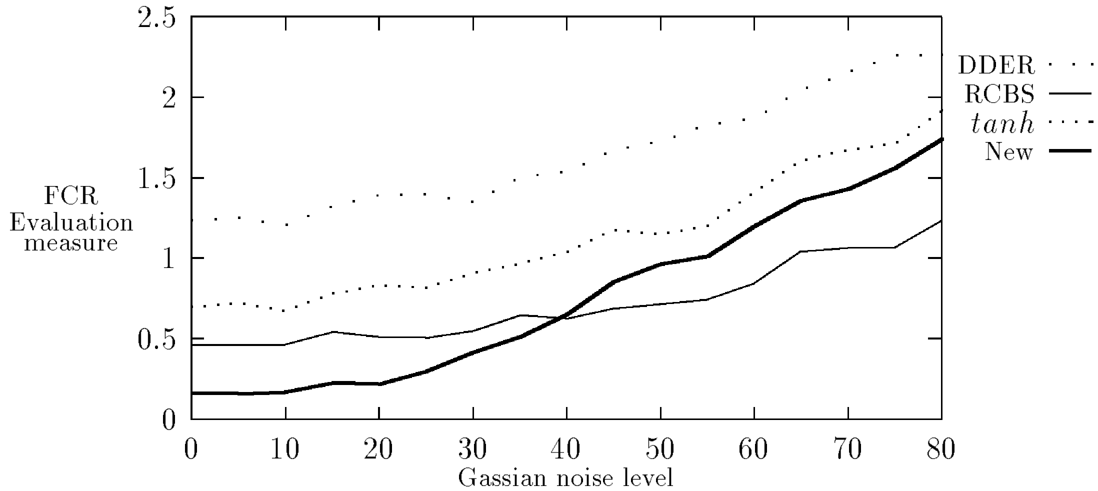


Figure C.9: FCR measure w.r.t. Gaussian noise on ring ($\epsilon = 30$).

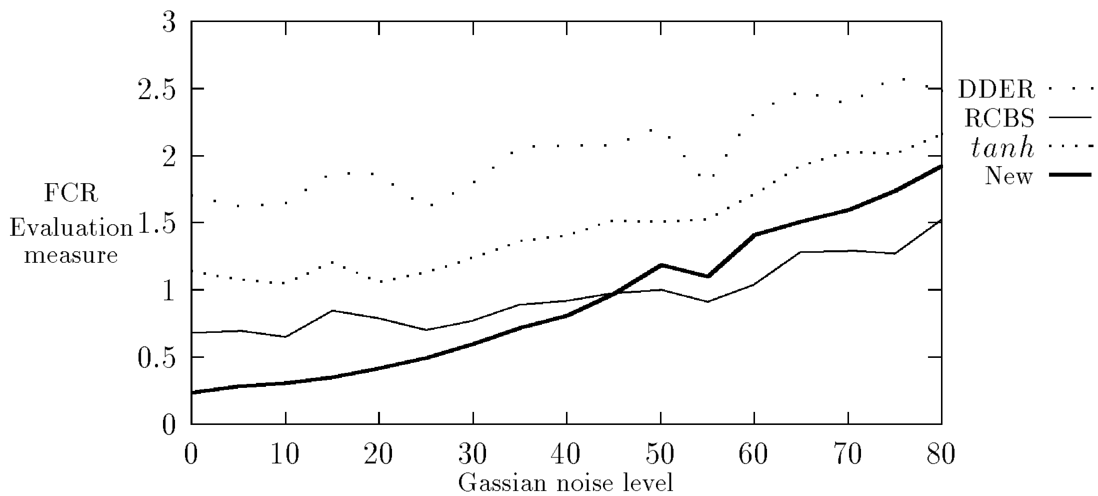
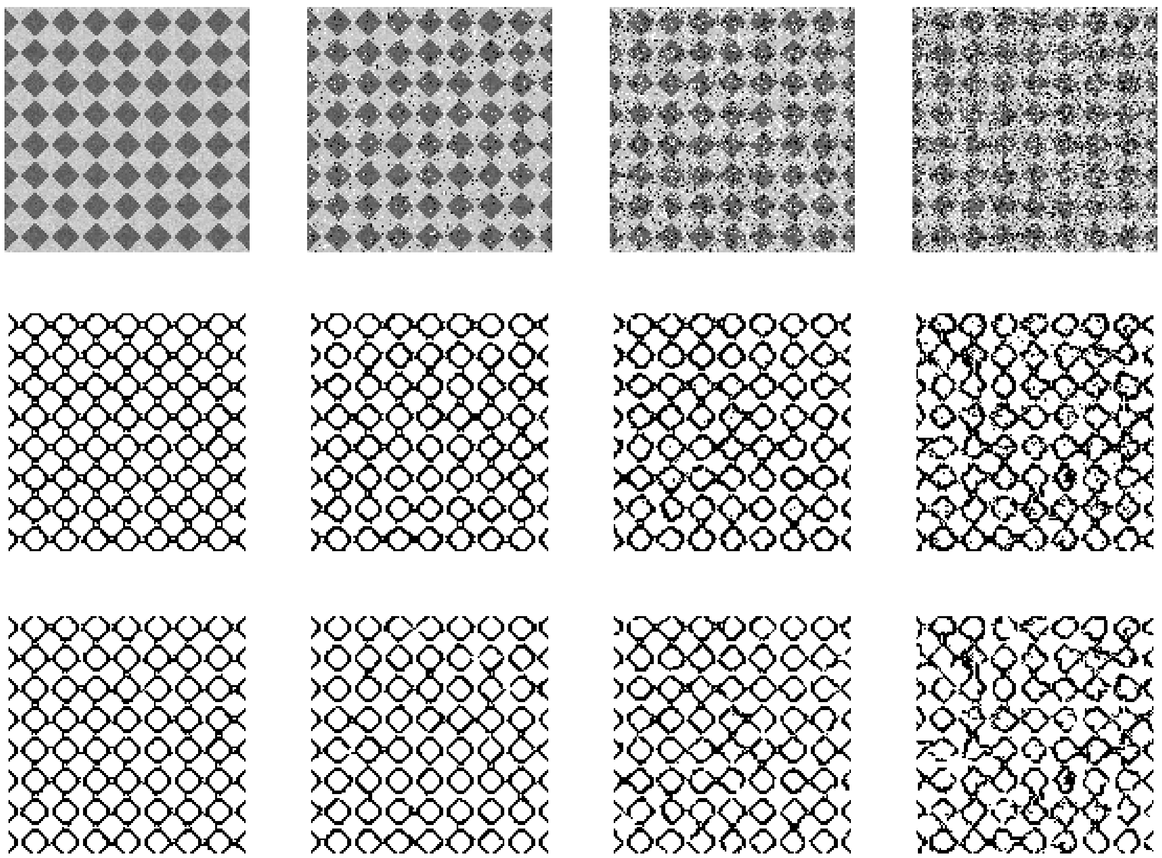


Figure C.10: FCR measure w.r.t. Gaussian noise on rings ($\epsilon = 40$).

Appendix D

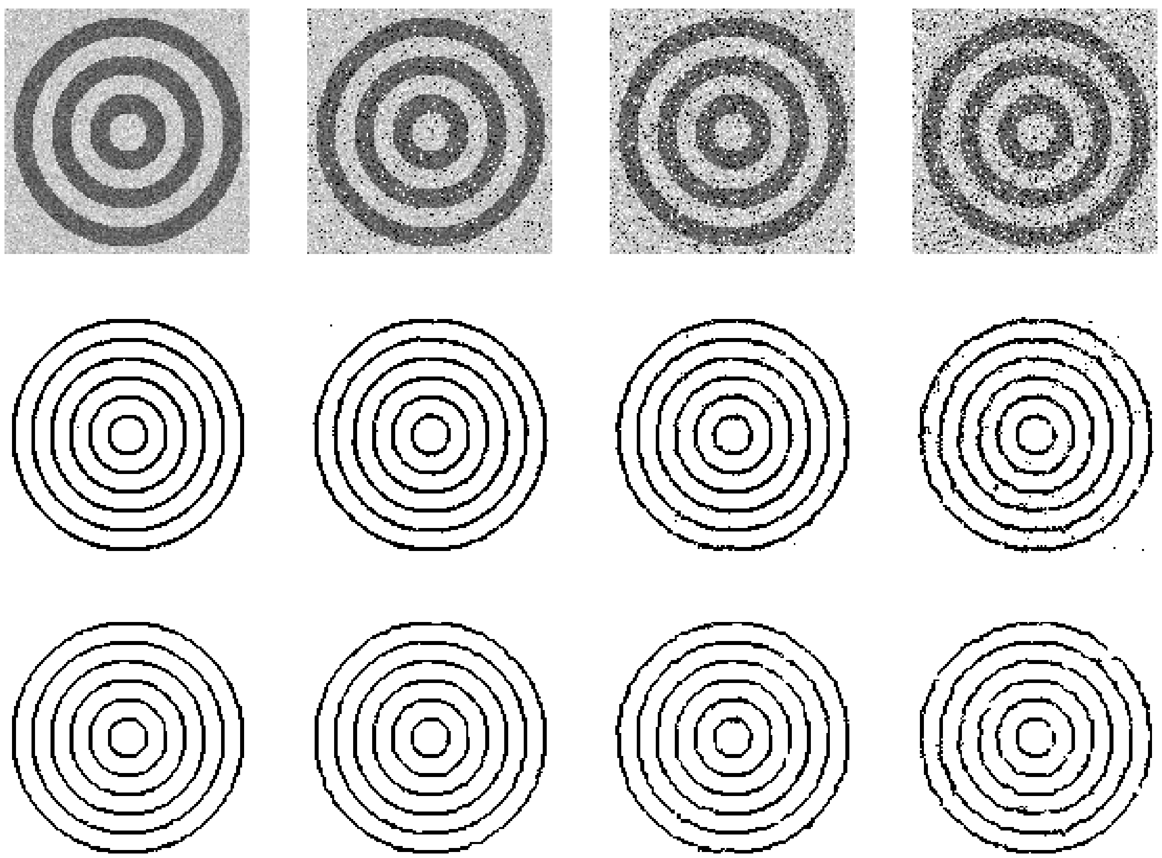
Comparison of the FCR Model and the RE Model

This appendix shows the comparison of the FCR model and the RE model on crossboard and ring images.



(1) $\sigma = 10,\ \epsilon = 0\%$ (2) $\sigma = \ 0,\ \epsilon = 10\%$ (3) $\sigma = 10,\ \epsilon = 30\%$ (4) $\sigma = 10,\ \epsilon = 50\%$

Figure D.1: Comparison of the FCR model and the RE model on crossboard.



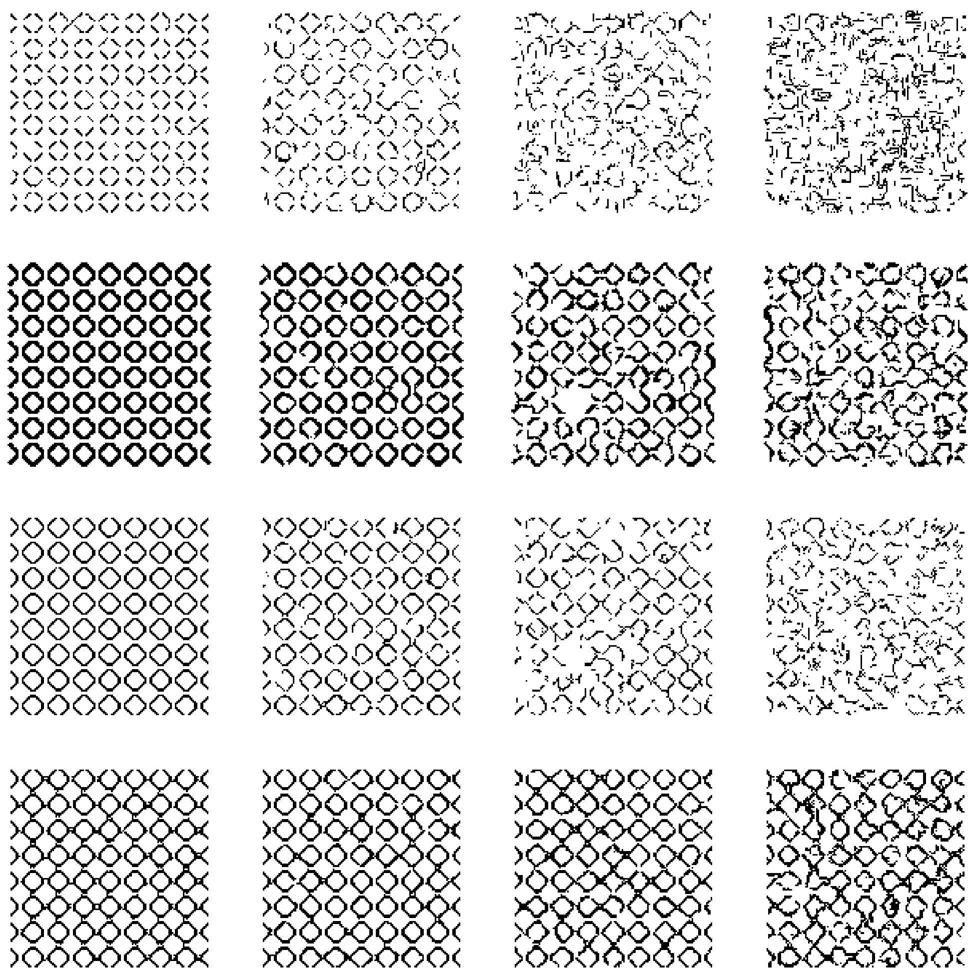
(1) $\sigma = 20,\ \epsilon = 0\%$ (2) $\sigma = 20,\ \epsilon = 10\%$ (3) $\sigma = 20,\ \epsilon = 20\%$ (4) $\sigma = 20,\ \epsilon = 30\%$

Figure D.2: Comparison of the FCR model and the RE model on ring images.

Appendix E

Performance Comparison Using the New RE Model

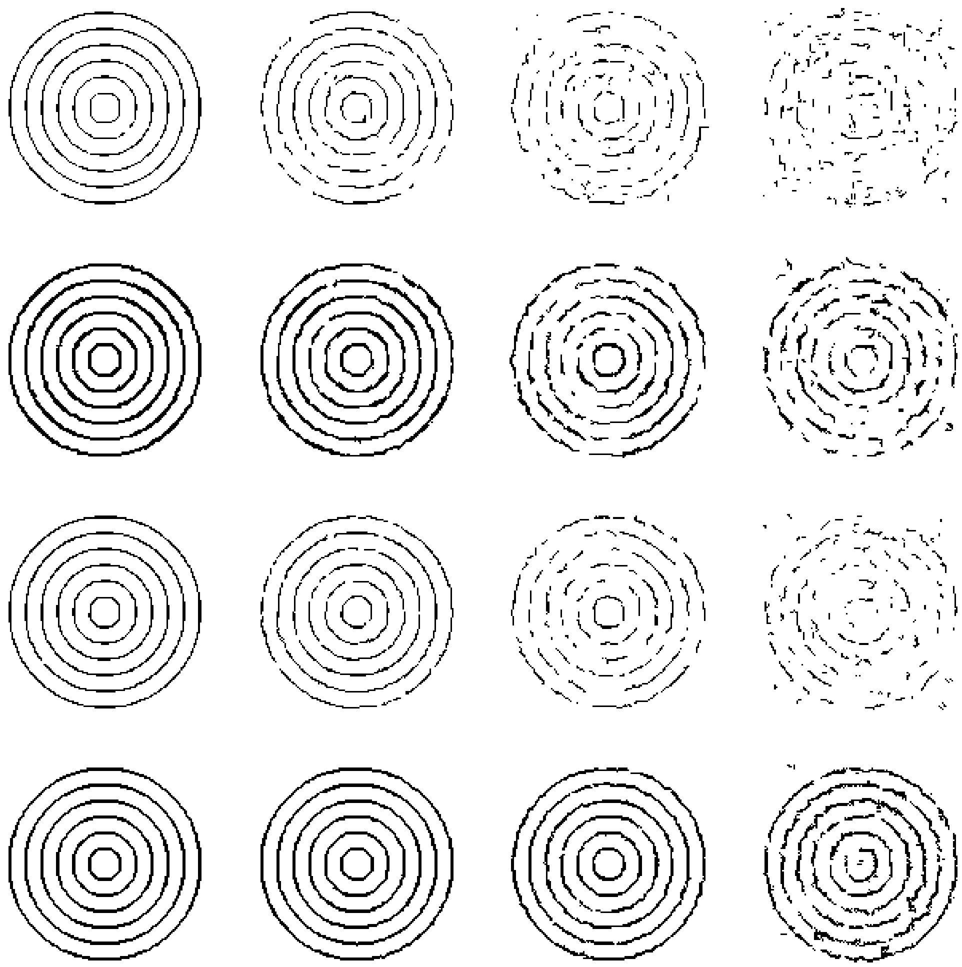
This appendix shows the qualitative and quantitative comparison of the Haralick operator, RCBS, the *tanh* operator and the new robust operator using the new refinement/evaluation model to generate the final edge map.



(a)$\epsilon = 0\%$ (b) $\epsilon = 10\%$ (c) $\epsilon = 30\%$ (d) $\epsilon = 50\%$

First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the $tanh$ operator.
Fourth row: outputs of the new edge detector.

Figure E.1: Comparison of crossboard outputs (new model), $\sigma = 10$.



(a)$\epsilon = 0\%$ (b) $\epsilon = 10\%$ (c) $\epsilon = 30\%$ (d) $\epsilon = 50\%$

First row: outputs of the Haralick operator.
Second row: outputs of the RCBS operator.
Third row: outputs of the *tanh* operator.
Fourth row: outputs of the new edge detector.

Figure E.2: Comparison of ring outputs (new model), $\sigma = 10$.

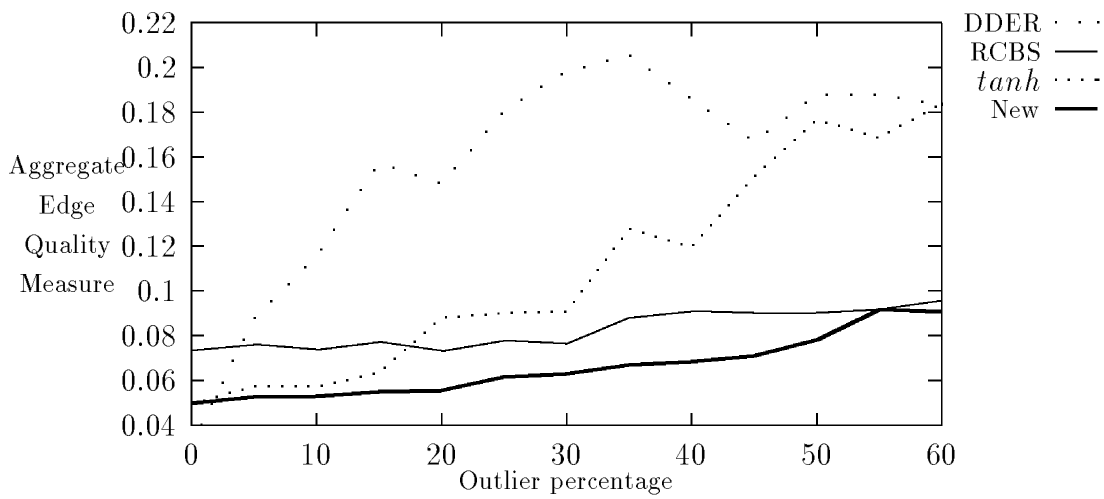


Figure E.3: New measure w.r.t. outlier level on crossboard ($\sigma = 10$).

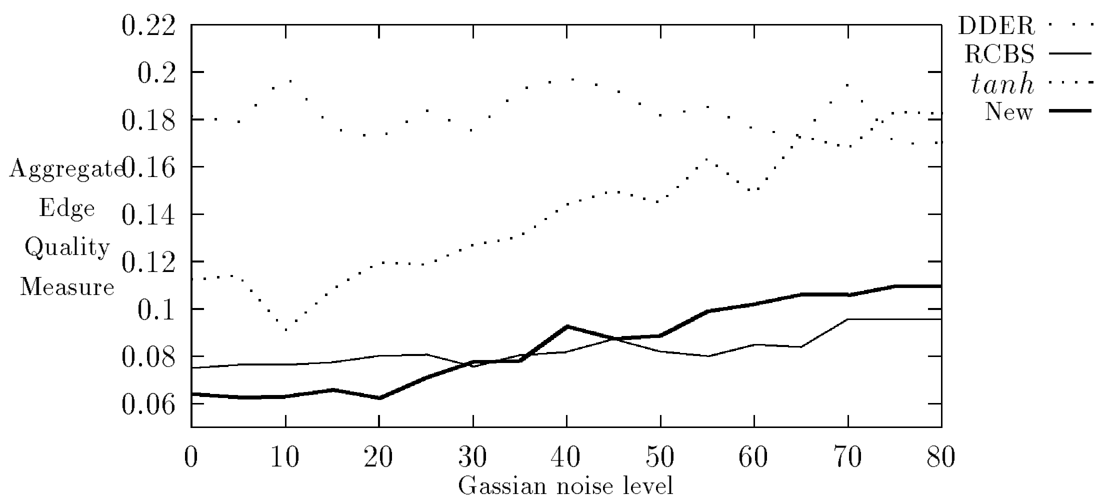


Figure E.4: New measure w.r.t. Gaussian noise on crossboard ($\epsilon = 30$).

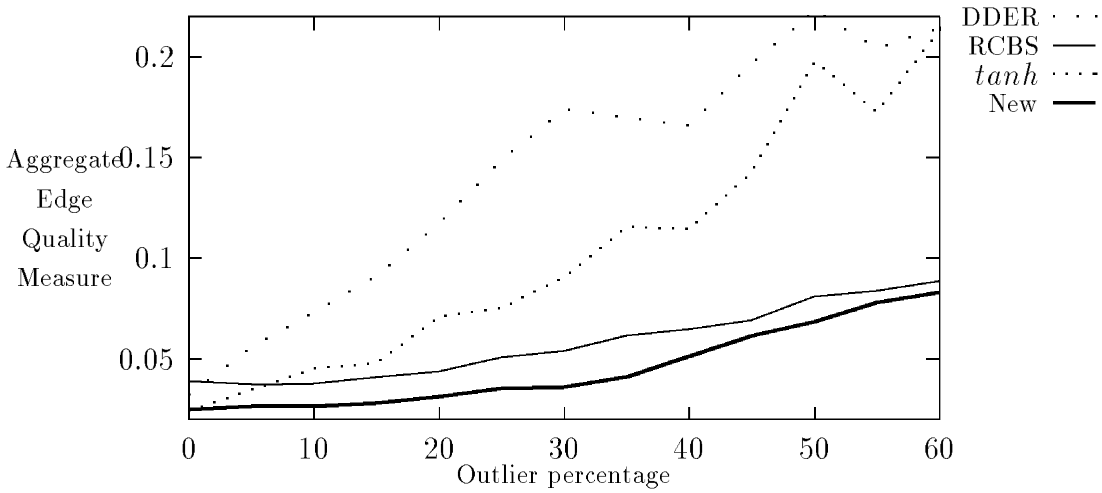


Figure E.5: New measure w.r.t. outlier level on ring images ($\sigma = 10$).

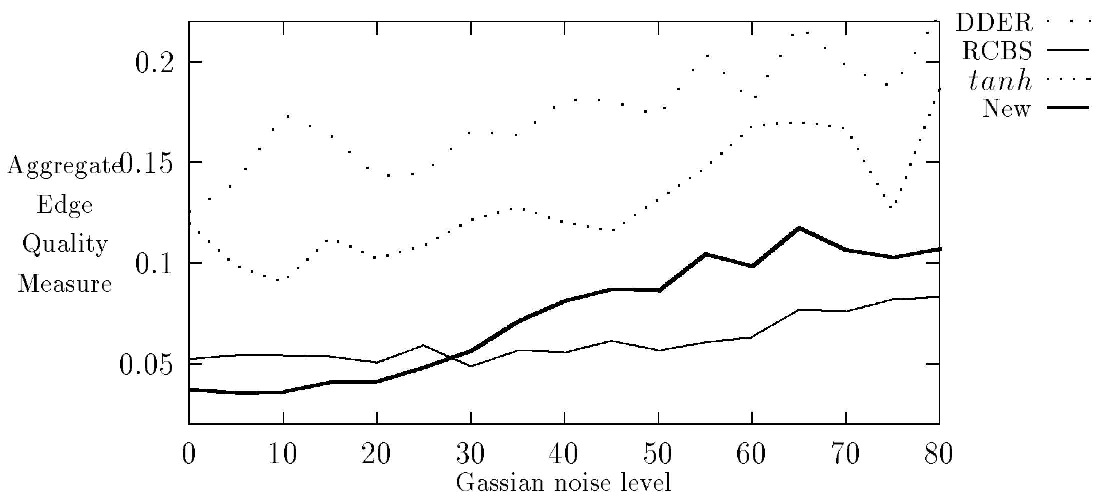
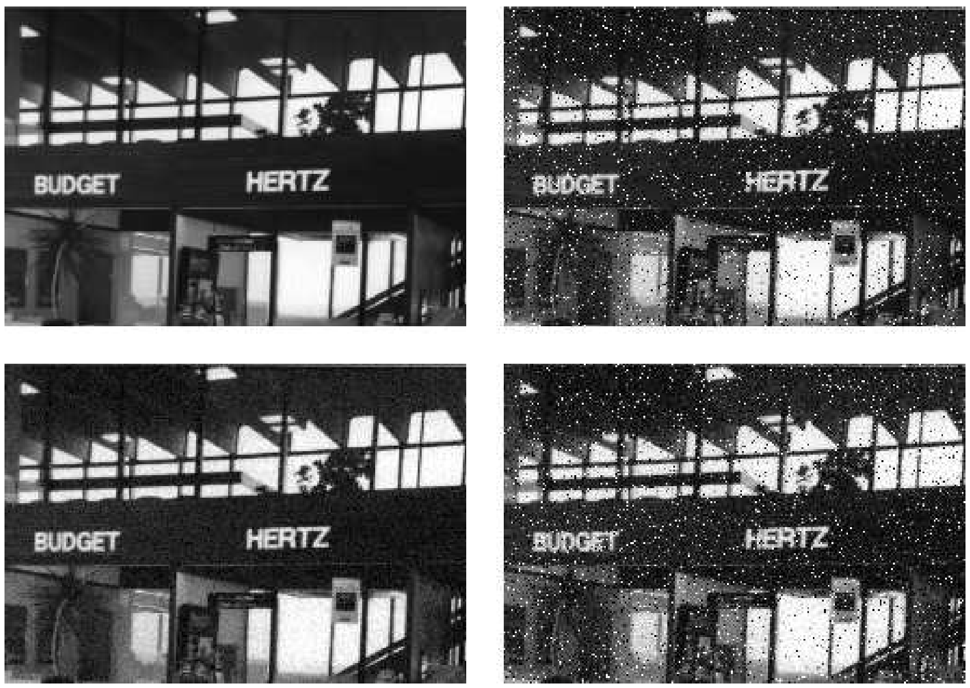


Figure E.6: New measure w.r.t. Gaussian noise on ring images ($\epsilon = 30$).

Appendix F

Comparison of Different Detectors on Real Image

The performance of the Haralick operator, RCBS, the *tanh* operator and the new robust operator on another sample real image is shown in this appendix. (The original picture is taken at Saskatoon airport.)



(1) No additional noise (2) $\sigma = \ 0, \epsilon = 10\%$
(3) $\sigma = 10, \epsilon = 0\%$ (4) $\sigma = 10, \epsilon = 10\%$

Figure F.1: Sample real images.

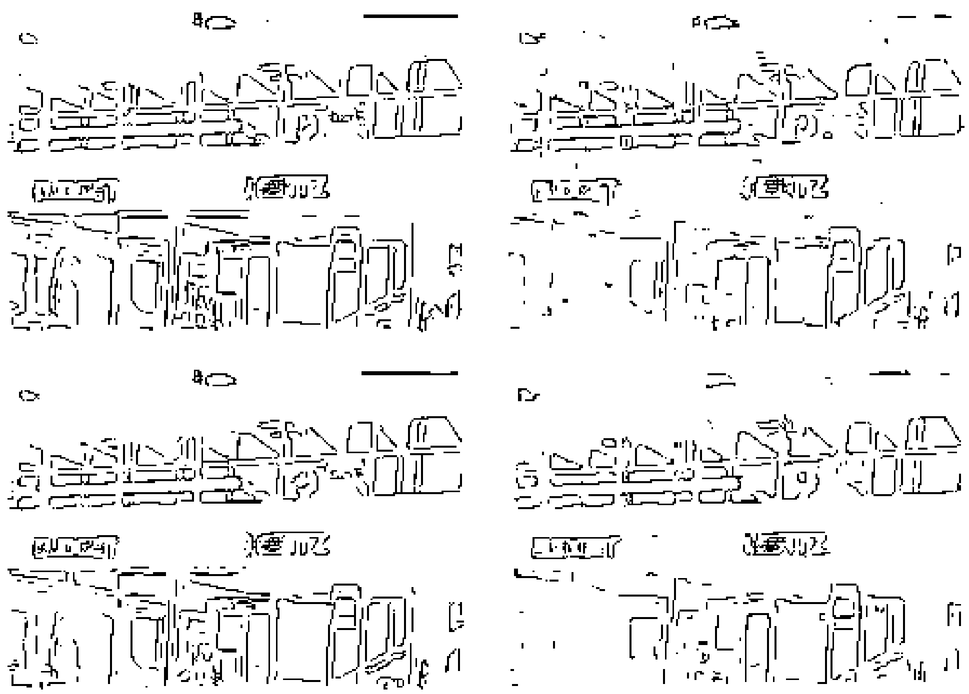


Figure F.2: Outputs of the Haralick's operator on a sample real image.

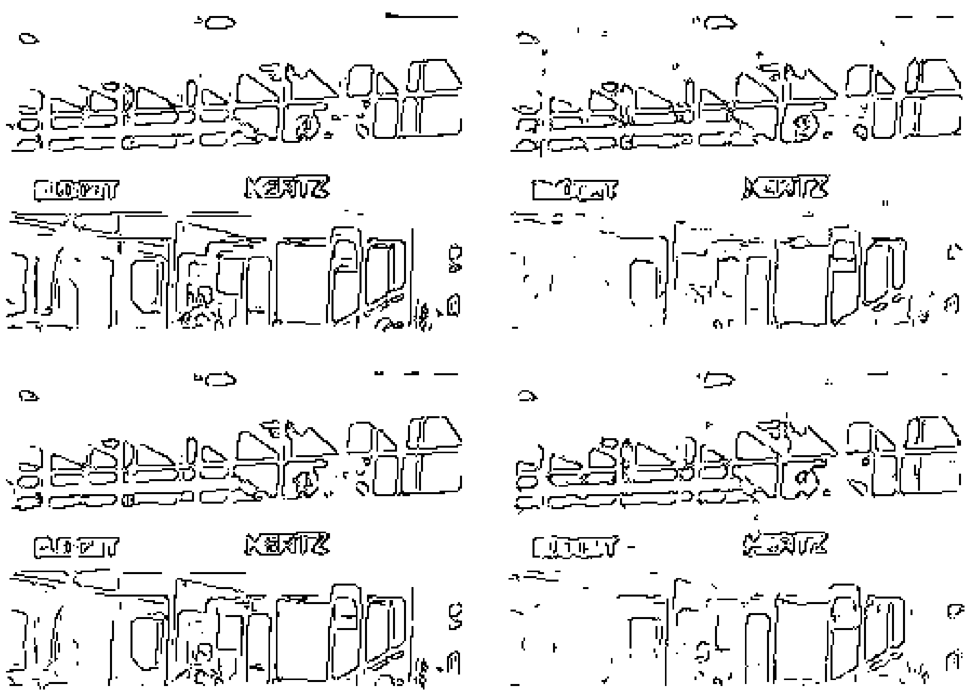


Figure F.3: Outputs of the RCBS operator on a sample real image.

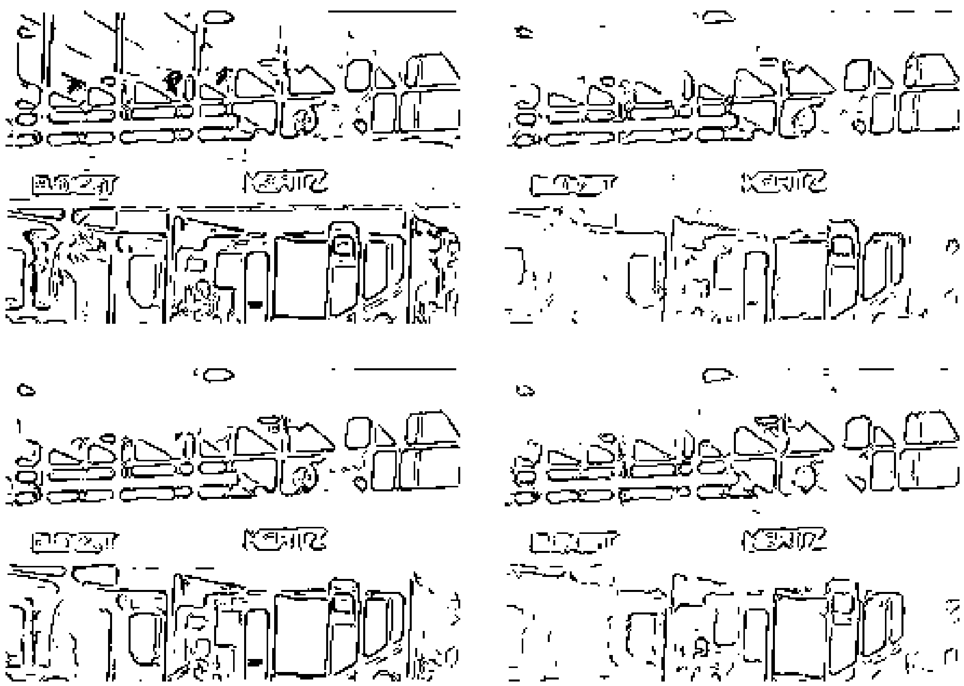


Figure F.4: Outputs of the *tanh* operator on a sample real image.

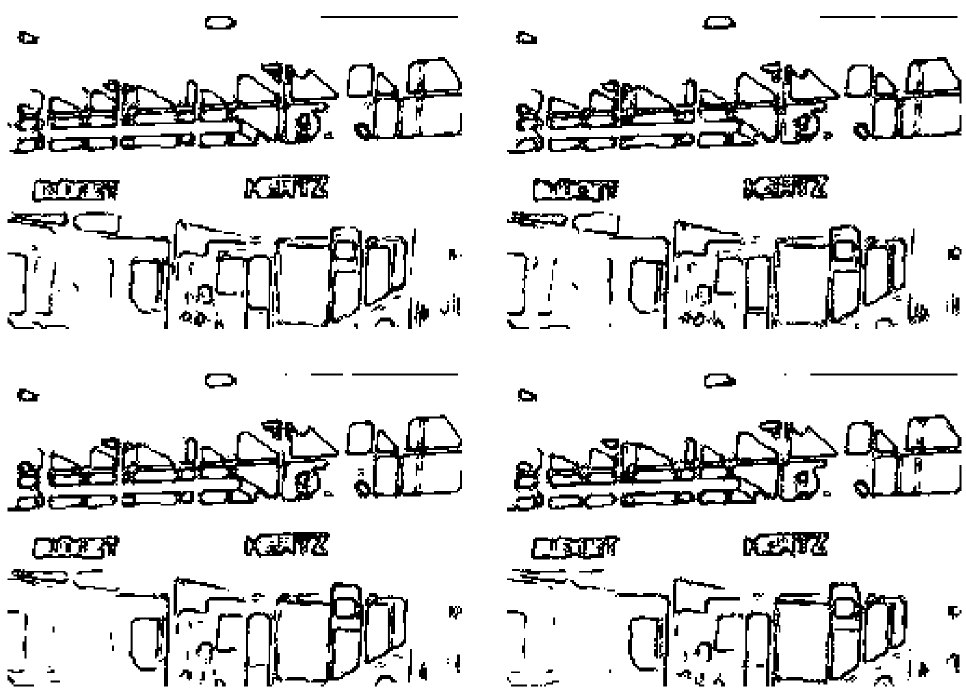


Figure F.5: Outputs of the new operator on a sample real image.

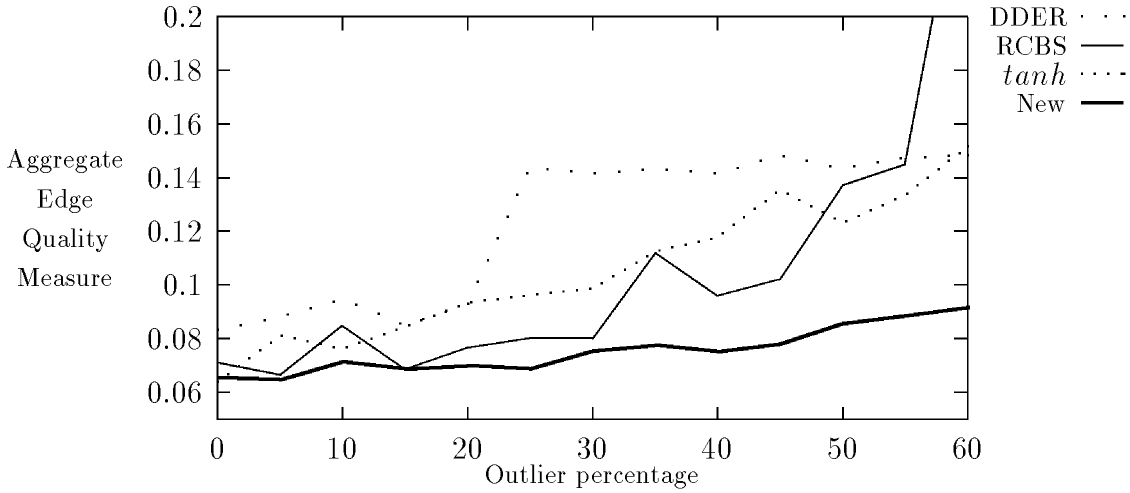


Figure F.6: New measure w.r.t. outlier level on the sample real image ($\sigma = 0$).

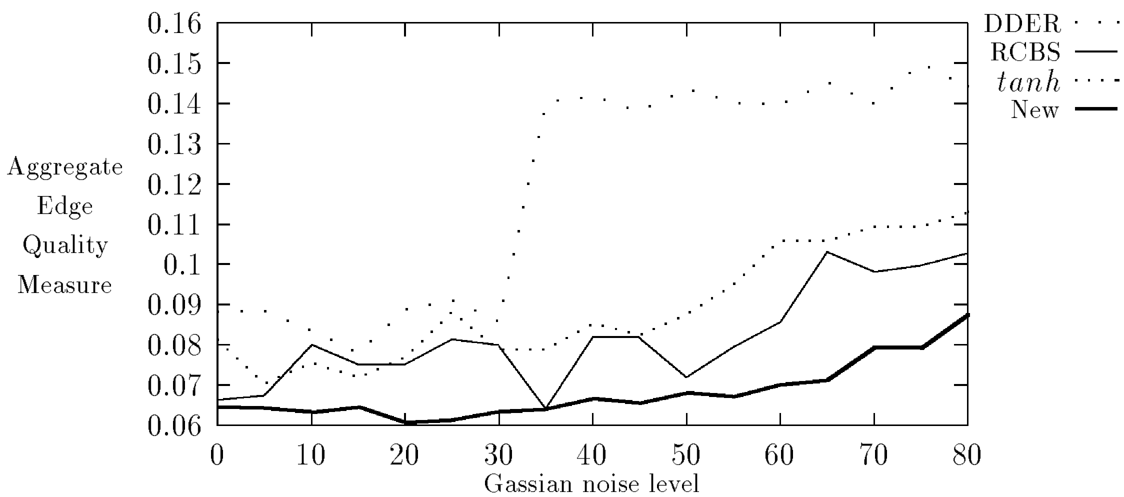


Figure F.7: New measure w.r.t. Gaussian noise on the sample real image ($\epsilon = 5$).