DOCUVISION III ™
IMAGE ENHANCEMENT PRIMER

Prepared By:

Litton Industrial Automation Systems, Inc.
Integrated Automation Division
1301 Harbor Bay Parkway
P.O. Box 4004
Alameda, California 94501-0404

Document Number: D008018, Revision 4

Table of Contents

List of Figures

1. INTRODUCTION

There are three primary objectives of the scanning input process of an image-based system. They are:

- Cost of capture must be minimized
- Integrity of the document must be maintained
- Image quality must meet organizational needs

The best way to contain scanning costs is to minimize the human involvement during the capture and quality review process. Each input activity obviously has significant cost implications when multiplied by hundreds of thousands of documents being captured.

The integrity issue is just as important as costs. If an operator is given the ability to “clean up” an image, provisions should be made to have a knowledgeable person verify that the contents of the document images have not been significantly modified. Not only does this add expense to the process, but it will also lower scanning throughput rates.

Litton/Integrated Automation has addressed these objectives by developing a suite of sophisticated image enhancement algorithms which automatically, without operator intervention, generate a good quality image even when the original document or microfilm is of marginal quality.

This document addresses the key features of this proprietary image enhancement module.

2. PROCESS OVERVIEW

Automated image enhancement is done with computer hardware or software that can analyze a two-dimensional, grey scale matrix of pixels (or dots), determine which pixels are significant, and convert the matrix to a binary (black and white) matrix, ready for further processing.

Figure 2-1 shows an illustration of image enhancement. While this is a simplified representation of image enhancement, it does demonstrate two features of most image enhancement algorithms. The “grey” background has been been eliminated and the fuzzy edges of the object have been solidified.

As scanned **As Enhanced**

Figure 2-1 An illustration of image enhancement

3. THRESHOLDING

The most common image enhancing algorithm is thresholding as shown in Figure 3-1. Whenever the input signal is below some threshold value, the resulting signal is off, or white. When the input signal is higher, the resulting signal is on, or black.

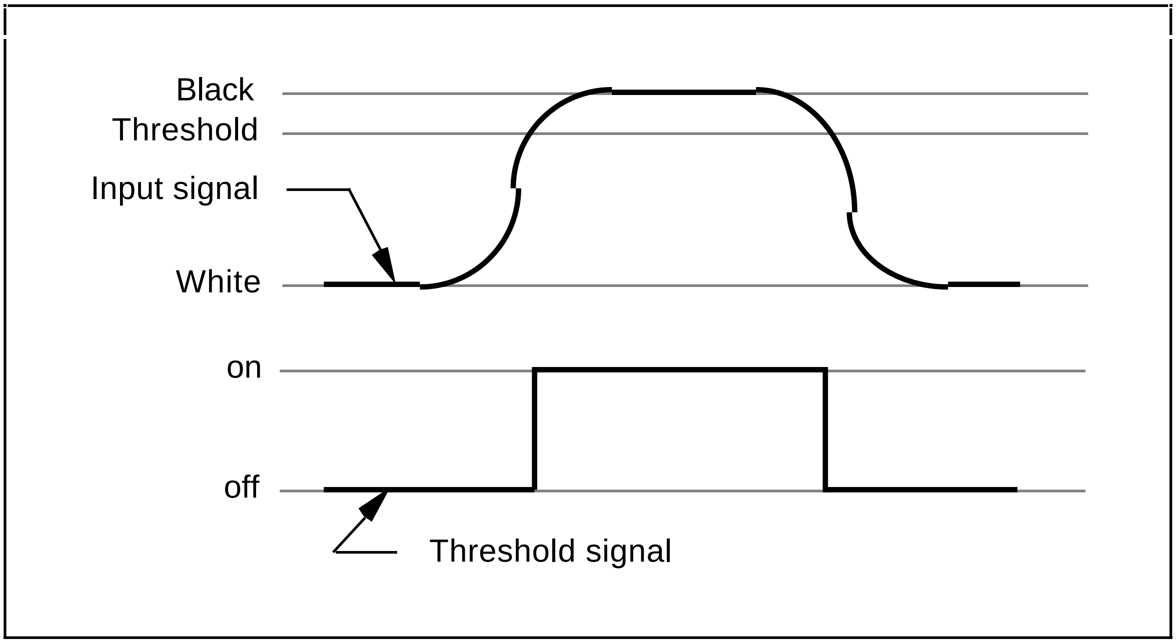

Figure 3-1 Typical threshold process

If the threshold value is set high, as shown in the figure above, only areas that are dark grey and black are saved. All other values of grey to white become white. As the threshold setting is lowered, more areas will become black until the everything is black.

At first glance, this sounds feasible, particularly when the ability is included to adjust the threshold value either statically before scanning a document or dynamically during the scanning process. In either condition, more or less “detail” can be saved, making for darker or lighter resulting documents.

Unfortunately, thresholding, as described above, will miss significant detail if the image contrast is low. Figure 3-2 shows a typical input signal where there is important detail within both the light and dark areas. Either the data in the dark areas or in the light areas will be captured with proper adjustment of the threshold value, but not both. Some data will be lost if only thresholding is used. Even dynamic thresholding cannot adjust to *both* conditions if they occur near each other as is frequently seen on engineering drawings.

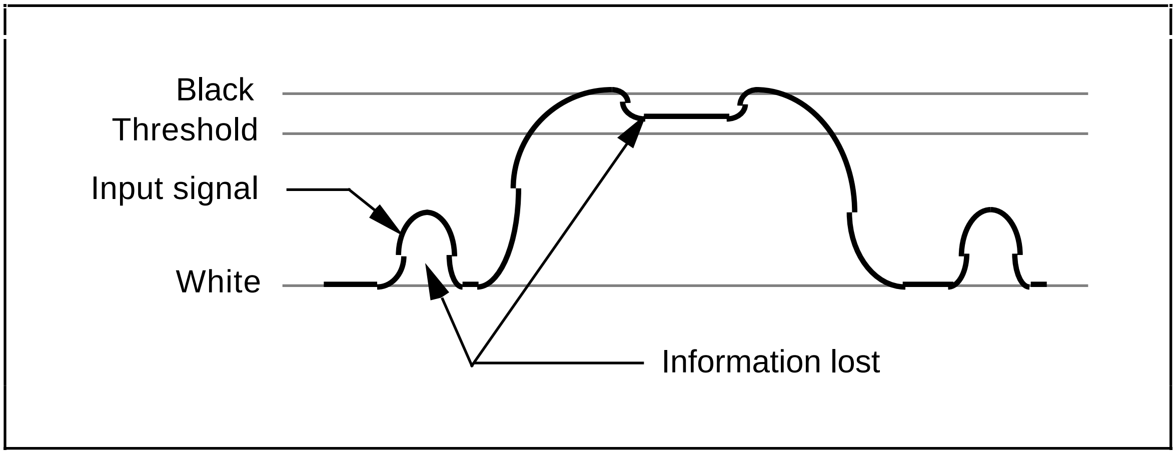

Figure 3-2 Loss of low contrast features

4. LITTON/IA IMAGE ALGORITHM

In order to address the loss of significant data in low contrast areas and other image enhancement problems, Litton/Integrated Automation has developed a sophisticated suite of proprietary programs that work in concert to produce an enhanced image.

For example, one portion of the image enhancement module uses a combination of three operators to determine the action to be taken to improve the image. Depending upon the relative values of these operators, different enhancement routines are used. The three operators are the following:

- Edge operator – identifies the region that contains an edge, or color change
- Level operator – identifies color based upon a threshold value
- Point operator – identifies the specific point for color change in the edge region

These operators are discussed further below. They are applied using the following logic equation:

If "**edge**"
  then "**point**"
else "**level**"

In other words, if the pixel being tested is in an edge region, as determined by the *edge operator*, the *point operator* is used. If the pixel is not in an edge region, the *level operator* is used.

This image enhancement suite is designed for real time image processing at a relatively high rate of data acquisition of over 2 megabits per second, or the equivalent of scanning 180 ANSI E-size drawings per hour.

4.1. EDGE IDENTIFICATION

Edge regions are identified by the edge operator. This operator uses the first derivative of the input signal to determine if the pixel being tested is in an edge region. If the absolute value of the first derivative is large, the pixel must be located in an edge region and the level operator is turned off. (See below for the level operator's functions.) If the absolute value is small, the pixel under test must be either in a solid black or solid white area and the level operator is turned on. See the edge operator output in Figure 4.1-1.

4.2. LEVEL OPERATOR

The level (or background) operator compares the value of the pixel being tested to a fixed threshold value and generates an "on" or "off" report based upon whether the pixel is greater or less than the threshold value. This operator is turned off when the testing is in an edge region. See edge operator above for how edges are determined. See Figure 4.2-1 for pictorial view of the level operator.

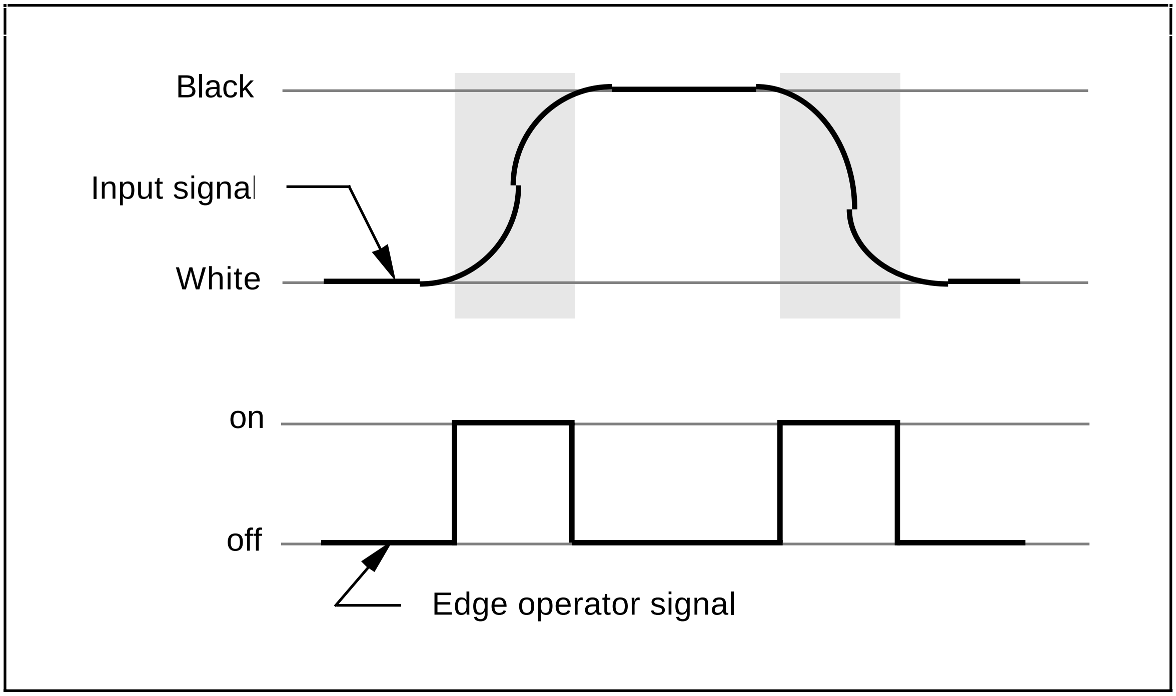

Figure 4.1-1 Edge operator domains

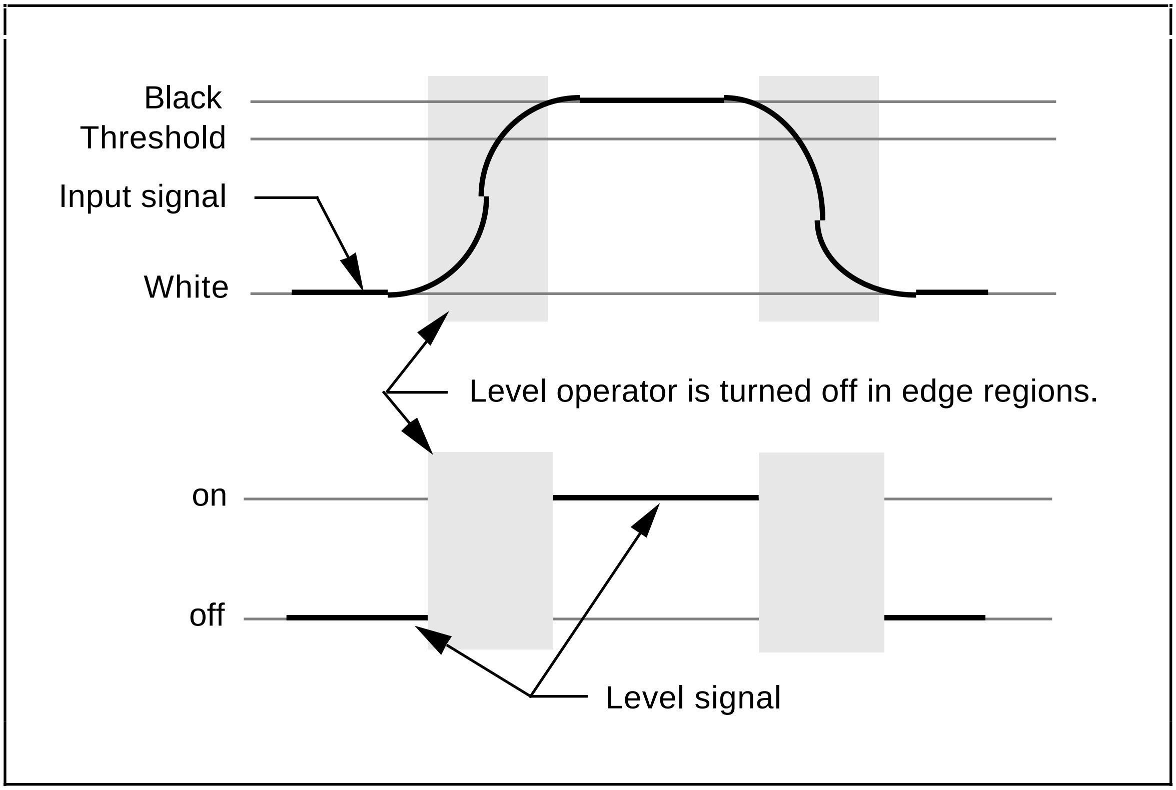

Figure 4.2-1 Level operator with a high threshold value

Thus, the threshold value can be set relatively high to discriminate against noise and speckle in the input signal without sacrificing the sensitivity needed to resolve detail in the edge regions.

4.3. POINT IDENTIFICATION

The point operator is used to identify the precise edge of an object. The operator uses the sign change of the second derivative of the input signal to determine the point where the change of grey is at its highest rate. Pixels before this point are turned white (or black) and

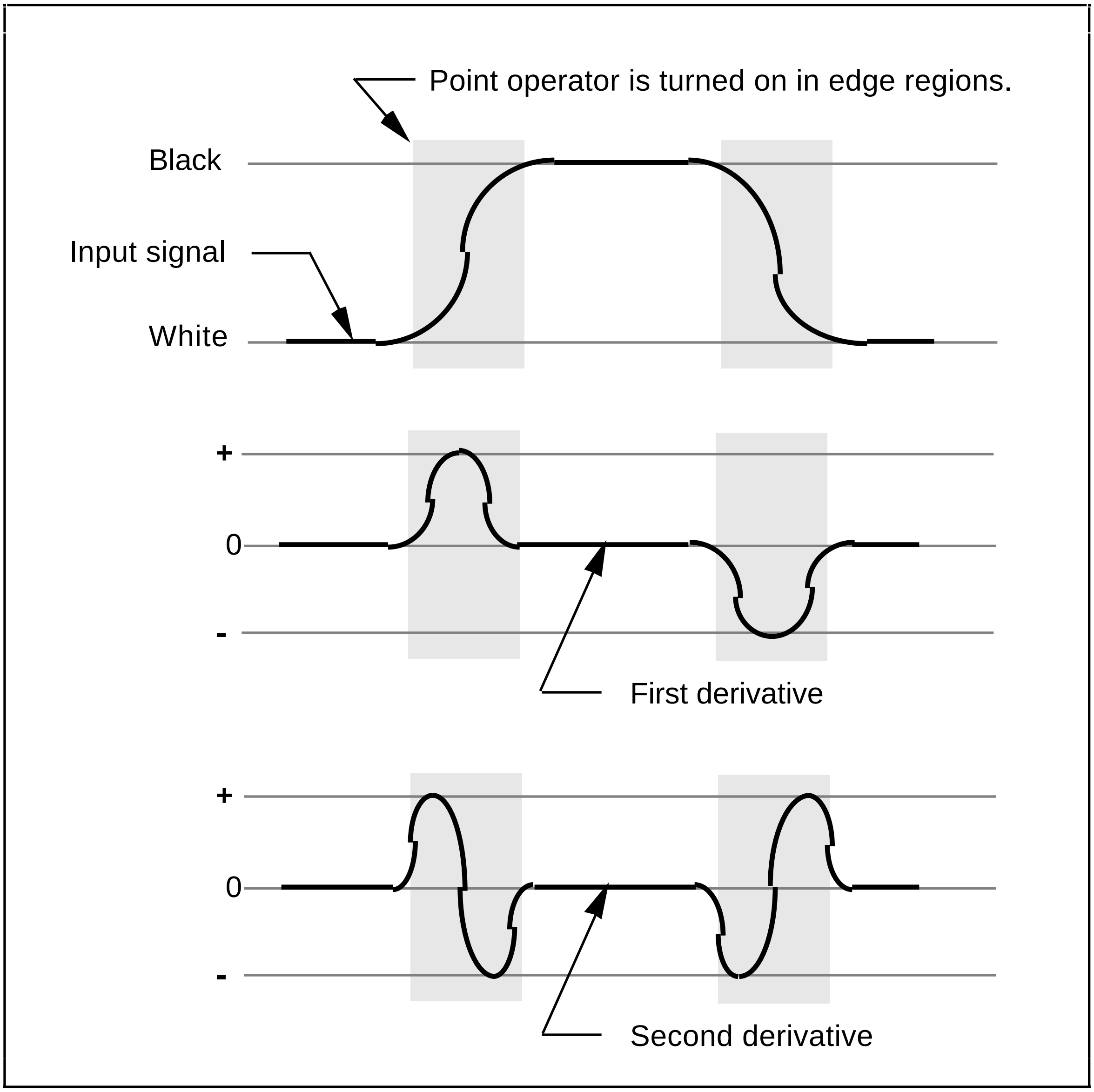

Figure 4.3-1 Input signal derivatives used for point identification

pixels after this point are turned the opposite color. The point operator only functions in the edge regions as defined by the edge operator noted above. See Figure 4.3-1 for a diagram of these functions.

5. RESULTS AND RECOMMENDATION

The discussion and diagrams above use a one dimensional matrix and the analysis of a single line of pixels. In actual practice, the analysis is applied to a two-dimensional matrix representing both length and wide of the scanned document.

The analysis algorithm is further refined by dynamically choosing the neighborhood that is used in the above calculations. The program can switch from a tight 3 x 3 pixel neighborhood to a sparse 7 x 7 or 9 x 9 neighborhood. The optimum size of the neighborhood depends upon the blur characteristics of the image being enhanced and the minimum feature size which needs to be resolved. The program automatically selects the appropriate neighborhood size.

Based upon the extensive and sophisticated image analysis that Litton/IA has incorporated in its image enhancement module, the results of this process are usually excellent and frequently dramatic.

It is because of these elegant tools that we usually recommend no additional work be performed by users to "clean-up" scanned images during quality review and before releasing. Not only does this work add a major cost and delay to the project, but it introduces the possibility that significant data might be erroneously modified.