Feature Extraction & Noise Removal: Vital Predecessors to Image Matching

Shaolei Zhang

Supervising Professors: Dr. Andrew Blanchard, Dr. Bruce Gnade, and Dr. Russell Hulse

Spring 2005
# Table of Content

Table of Content .................................................................................................................. i  
List of Figures ..................................................................................................................... ii  
List of Tables ..................................................................................................................... iii  
List of Tables ..................................................................................................................... iii  
1. Introduction................................................................................................................. 1  
2. Feature Extraction....................................................................................................... 1  
   2.1 Image Uniqueness ............................................................................................... 1  
   2.2 Extracting Relevant Information ........................................................................ 2  
   2.3 Computational Method of Feature Extraction .................................................... 3  
   2.4 Benefits of Edge Detection ................................................................................. 4  
   2.5 Less is More ........................................................................................................ 4  
3. Definition of Noise ..................................................................................................... 5  
   3.1 Varying Light Intensity ....................................................................................... 5  
   3.2 Varying Light Orientation ................................................................................... 6  
   3.3 Partner in Crime: The Physical Environment ..................................................... 7  
4. Noise Removal.......................................................................................................... 10  
   4.1 Median Filter ..................................................................................................... 10  
   4.2 Wiener Filter ..................................................................................................... 11  
   4.3 Line Detection ................................................................................................... 12  
   4.4 The Good and the Bad ...................................................................................... 14  
5. Conclusion ................................................................................................................ 16  
6. BIBLIOGRAPHY ..................................................................................................... 17
List of Figures

Figure 1. The Mona Lisa ................................................................. 1
Figure 2. A Patterned Image ............................................................ 1
Figure 3. A Historical Epic ............................................................. 2
Figure 4. Steel Hand ................................................................. 2
Figure 5. In Texas ........................................................................... 2
Figure 6. Drifters of the Sea .......................................................... 3
Figure 7. Edge Detection Operators ................................................. 4
Figure 8. Revisit "In Texas", Part I .................................................... 4
Figure 9. Revisits "In Texas", Part II ............................................... 4
Figure 10. Clocks ............................................................................ 5
Figure 11. Clocks: After Grayscale .................................................. 5
Figure 12. Clocks: After Edge Detection .......................................... 5
Figure 13. UTD Newsletter ............................................................ 5
Figure 14. Plaque ........................................................................... 6
Figure 15. A Crowd of People ........................................................ 7
Figure 16. A Single Person ............................................................ 7
Figure 17. Handicap Sign .............................................................. 7
Figure 18. Handicap Sign: After Edge Detection ............................... 8
Figure 19. Man under Ball of Light ................................................. 8
Figure 20. Man under Ball of Light: After Edge Detection ............... 8
Figure 21. Comparing Enlarged Regions .......................................... 9
Figure 22. After Image Processing .................................................. 9
Figure 23. Man Sitting by a Giant Folder, Part I ................................. 11
Figure 24. Various Windows for Median Filter ................................. 11
Figure 25. Man Sitting by a Giant Folder, Part II ............................... 12
Figure 26. Various Windows for Wiener Filter .................................. 12
Figure 27. Man Sitting by a Giant Folder, Part III .............................. 12
Figure 28. Extracting Straight Lines ............................................... 13
Figure 29. Row by Row ................................................................ 13
Figure 30. Row by Row Again ...................................................... 13
Figure 31. Column by Column ..................................................... 14
Figure 32. Result after Line Detection ............................................. 14
Figure 33. Failure of Line Detection ............................................... 16
List of Tables

Table 1. Causes and Definitions of Noise, part I ............................................................... 9
Table 2. Causes and Definitions of Noise, part II ............................................................. 10
Table 3. Causes and Definitions of Noise, part III ........................................................... 10
Table 4. Causes and Definitions of Noise, part IV ........................................................... 10
Table 5. The Advantages and Disadvantages of the Median Filter ............................... 14
Table 6. The Advantages and Disadvantages of the Wiener Filter ............................... 15
Table 7. The Advantages and Disadvantages of the Line Detection ............................... 15
1. Introduction

“One picture is worth a thousand words”. Although this ancient proverb sounds hackneyed, it is very true. It is much easier for a speaker to convey a story to an audience when an illustration of what is occurring in each junction of the story is presented before the audiences’ eyes, given that, of course, the illustration is relevant and clearly depicts what is happening in the story.

Now let’s image just for a second, that the illustration did not truly depict what the speaker is trying to convey. So in what ways will the illustration distort the story? If the speaker is attempting to convince an audience that a certain specie of mountain lions exits only in the dessert region of South Africa while holding up an figure that displays a mountain lion of that same specie comfortably lying in the tropical jungles of Australia, the audience will undoubtedly be perplexed by the irrelevancy between the speaker and the figure. In a similar fashion, when a raw, unprocessed image undergoes an image matching algorithm, the results could be perplexing.

In this thesis, the concept behind feature extraction of a raw, unprocessed image is discussed in detail. Furthermore, the definition and cause of noise that appears in an image is illustrated through concrete examples. Finally, various methods of noise removal and their advantages and disadvantages are presented.

This thesis is intended to be a basic body of work that could be expanded upon. As such, this thesis is grouped into these three parts:

- The concept of feature extraction
- The definition and cause of noise in an image
- Methods of noise removal and their respective advantages and disadvantages

Each of the three parts can be viewed as a separate subject. On the other hand, the interconnectivity between each part allows the entire paper to be viewed as one subject.

2. Feature Extraction

2.1 Image Uniqueness

Before getting into the details and the various aspect of feature extraction, a few words must be said regarding the concept of image uniqueness.

Every portrait, figure, or painting possesses its own unique characteristics. These characteristics help visitors paint a mental representation of the art work in their mind. For example, when a visitor visits the museum and looks at the Mona Lisa, the visitor will most likely focus closely at the face and upper body of the Mona Lisa while ignoring the landscape behind the Mona Lisa.

![Figure 1. The Mona Lisa](image1.png)

And it would be reasonable to proclaim that after visiting the entire museum, the visitor will most likely remember the face and perhaps the upper body of the Mona Lisa rather than the details of the landscape behind the Mona Lisa since the painting of Mona Lisa is renowned world wide for the elegance of the lady’s smile, not the background landscape.

On the other hand, when a visitor looks at a piece of art work that consists mostly of repetition of the same object, the visitor will most likely notice the pattern of the art work more vividly than the makeup of each individual object since all the objects are identical.

![Figure 2. A Patterned Image](image2.png)
Furthermore, a painting saturated with details such as one that depicts a monumental time in history shown below will have a visitor’s attention focused on every single inch of the art work.

![Figure 3. A Historical Epic](image)

The gloomy smoke circling the sky, the dying general uttering his last few words as he is helped up by his men, and the last desperate plea conveyed by the men wearing the fur hats are just few things that a visitor will notice and react to before a solid mental image of the painting is stored in the visitor’s mind.

### 2.2 Extracting Relevant Information

By taking notice of the unique characteristics of a piece of art work, a visitor can easily remember the art work by referring back to these characteristics stored in the visitor’s mind. At the same time, in a parallel fashion, by recognizing the unique features of an image, one can extract such relevant information to enhance an image matching algorithm, rendering it less complex, yet more robust. Furthermore, in many instances, without an accurate representation of the original image, even the most capable image matching algorithm would fail to work properly. Therefore, the feature extraction process is an extremely important initial step in creating an efficient and reliable image matching system.

Recognizing what features to extract from an image is crucial. For instance, this image taken from the UTD art gallery barn features a very noticeable object located in the center of the art work.

![Figure 4. Steel Hand](image)

For this particular image, the most striking feature would be the outline of the hand. Overall, this image contains fairly little detail with the exception of the human hand. And since no other piece of art work inside the UTD art gallery barn resembles anything like a human hand, extracting the outline of what appears to be a human hand from this piece of art work would be ideal.

On the other hand, here is an image of another piece of art work taken from the UTD art gallery barn. Let us name this image, “In Texas”.

![Figure 5. In Texas](image)

Unlike the piece of art work that resembles a human hand, “In Texas” contains more than just one obvious detailed object. Several artful writings, a mechanical cow and horse, a glowing neon light, and a Texas state sign are just few very detailed items that make up this particular piece of art work. Simply extracting the outline of only one of these items would be unsatisfactory. One must extract the outlines of all the items included within the art work in order to generate the best and fullest representation of this particular piece of art work.
2.3 **Computational Method of Feature Extraction**

Since this approach of feature extraction techniques rely heavily on detecting the outlines of objects within an image, the most suitable computational method that should be used to implement this feature extraction techniques would be edge detection.

Since edge detection was developed many years ago, there are hundreds of different algorithms that can be used to detect outlines or boundaries of an image. All of them share a few fundamental concepts. First of all, edge spots on images can be considered as pixel spots of sudden gray level changes. Therefore, an image must be converted into gray scale in order for edge detection to work. For example, it is understandable to classify edge spots in a binary image as black pixels that contain at least one white pixel in its neighborhood. For a continuous image, \( f(x, y) \), its derivative presumes a local maximum exists in the direction of the edge point. As a result, one edge detection method is to determine the gradient of function \( f \), along, \( r \), in the direction of \( \theta \) in the following manner:

\[
\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial r} = f_r \cos \theta + f_y \sin \theta
\]

When \( \left( \frac{\partial f}{\partial r} \right) \left( \frac{\partial f}{\partial r} \right)^T \) is equal to zero, the maximum value of \( \frac{\partial f}{\partial r} \) is attained, giving this:

\[-f_r \sin \theta_k + f_y \cos \theta_k = 0\]

\( \Rightarrow \theta_k = \tan^{-1} \left( \frac{f_y}{f_r} \right) \)

Where: \( \left( \frac{\partial f}{\partial r} \right)_{\text{max}} = \sqrt{(f_r)^2 + (f_y)^2} \)

And \( \theta_k \) is the direction of the edge spot.

Based on these equations, several different gradient operators can be used to detect edges. While some of the most popular gradients include Roberts, Prewitt, Sobel, and Canny, the gradient that works the best is the Canny operator. Unlike the other operators, the Canny operator processes data in a multi-stage fashion. First, the grayscale image is evened out by Gaussian convolution. Second, a basic first order derivative operator is applied to the smoothed image. As the edges produce crests in the gradient magnitude, the algorithm traces along these crests and convert all pixels that are not aligned on the crest to zero, giving the output a look of a thin line. This process is known as non-maximal suppression. It is governed by two thresholds, \( A \) and \( B \), where \( A > B \). The non-maximal suppression process can only initiate at a point on the crest that is larger than \( A \). The process then persists in the direction out from that point until the maximum of the crest drops below \( B \). Therefore, the two thresholds guarantees that noisy edges are not fragmented into smaller pieces. In addition, the two thresholds allows the operator to detect true weak more efficiently than the other operators and is less likely than the others to be deceived by noise.

Here is a grayscale version of an image taken at the UTD art gallery barn.

**Figure 6. Drifters of the Sea**

And here is how the image would after undergoing edge detection using MATLAB software.
The edge detection algorithm utilizing the Sobel, Roberts, and Prewitt operators did not identify all possible boundaries exited in the original image. Only the edge detection algorithm utilizing the Canny operator was able to identify majority of the boundaries exited in the original image.

2.4 Benefits of Edge Detection

When an image undergoes edge detection, it loses a large amount of information. First, the image is converted to grayscale and the colors are thrown away. Then edge detection converts the grayscale image into a binary image that consists only of white and black pixels, white being one, black being zero. So how can anyone be possibly pleased when a substantial amount of information that makes up the original image is lost?

Losing information of an image is actually beneficial if the information that is lost is not essential to detecting the features of an image. For instance, let’s revisits the art piece, “In Texas”.

When the original image that still contains the color spectrum is converted into a grayscale image, the size of the image is reduced from one hundred sixty four kilobytes to thirty four kilobytes, an eighty percent decrease. However, since the feature extraction process relies a great deal on extracting the outlines and boundaries of an image, losing the colors of an image is trivial. Next the image undergoes edge detection using the Canny operator, once gain, done using MATLAB software.

2.5 Less is More

One can reiterate the message an orator is attempting to convey by summarizing his speech into a few sentences. In the same fashion, one can reconstruct an image by picking and choosing the unique features of an image that best represents the image.

By recognizing what distinctive features to extract from one or a group of images, one can reduce the complexity of the image without losing vital information required for image matching. Moreover, discarding unnecessary information and reducing the size of an image in terms of memory allows an image matching algorithm to process the image much more efficiently.
3. Definition of Noise

3.1 Varying Light Intensity

When a snapshot is taken using a standard digital camera, the image that is conceived can be altered by various factors. For instance, let’s take a close look at these two images.

![Figure 10. Clocks](image)

The image of a clock on the left is taken by a digital camera during the afternoon on a particular day with the lights turned on. The image of the same clock on the right is taken by the same digital camera with the lights turned off. Upon a cursory glance, the two images appears to be identical with the exception of a difference in coloration, the image on the left possessing a tint of tan while the image on the right possessing a shade of light gray. However, when feature extraction process is applied to the two images utilizing edge detection, one would expect the edge detected image on the left to appear identical to the edge detected image on the right since information regarding the color spectrum is thrown away, leaving us with nearly identical grayscale images.

![Figure 11. Clocks: After Grayscale](image)

Moreover, one would expect the edge detected counterparts of both images to be nearly identical since their grayscale counterparts are nearly identical. However, contrary to this prediction, here are the results of the edge detected images using the Canny operator.

![Figure 12. Clocks: After Edge Detection](image)

The edge detected image on the right hand side contains largely the outlines of the clock and nothing else. On the other hand, the edge detected image on the left hand side possesses small groupings of clutters in addition to the outlines of the clock.

![Figure 13. UTD Newsletter](image)

From the example above, a critical weakness of the edge detection algorithm is revealed. At different levels of light intensity, the edge detection algorithm is capable of detecting the outlines or boundaries of the object of one’s interest as well as undesirable clutters that appears in a random fashion. To further illustrate this unwelcoming affect, let us take a look at this set of images taken using a digital camera at different times during the day along with its edge detected counterparts.

Unlike the previous example, the coloration of each image above does not vary. However, one can easily observe that the level of light intensity for each image does vary. While the first image is exposed to the most amount of light, the third image is exposed to the least amount of light. By coincidence, the clutters just happened to increase as the light intensity decreases in this particular
example. However, further experiments have proved that this is not always the case. Nevertheless, a grave problem regarding feature extraction has emerged: as light intensity varies, the arrangement of undesirable clutters also varies. Even though these three images above are essentially identical, no image matching algorithm will be able to correctly match any two of these three images because each of these image’s edge detected counterpart contains a different amount of clutter as well as a different arrangement of clutters.

Through feature extraction, an edge detected image is supposed to represent an accurate portrayal of the original image. But with the existence of clutters that vary with light intensity, an edge detected image now ends up distorting the representation of the original image. For all three images above for instance, instead of representing the UTD newsletter shaped like a rectangular box, the edge detected counterpart now appears to represent a man who possesses a rectangular face and a bizarre and puffy hairdo.

When jitters began to smother human voices during a telephone conversation, spoken messages can be difficult to convey. In a similar fashion, when clutters began to alter the representation of an image in the process of feature extraction, image matching can be very difficult to implement. Finally, since these clutters symbolize a disturbance that obscures the clarity of an entity, they shall be referred to as noises.

3.2 Varying Light Orientation

As it turns out, a varying light intensity is not the only problem that could immobilize the feature extraction process and complicates an image matching algorithm.

For instance, let’s take a look at another set of images taken using a digital camera and its edge detected counterparts. This set of images, however, is taken in such a way that the orientation of the light source strikes the object of our interest differently for each image.

The light source strikes the object in the first image at a ninety degree angle toward the top side. On the other hand, the light source strikes the object in the second image at a zero degree angle towards the right side. Finally, the light source strikes the object in the third image at a one hundred and eighty degree angle towards the left side. Because the light source strikes the object at a different point of view for each image, its edge detected counterpart therefore contains a unique amount of noise that possesses a distinctive arrangement.

For the first image, the edge detected counterpart contains an arrangement of noise that is located around the top of the object. By looking closely at the original image and its edge detected counterpart, one can draw the conclusion that the configuration of the noise is shaped similar to the outline of the light source.

For the second image, the edge detected counterpart contains an arrangement of noise that is located on the right side of the object. Once again, by correlating the original image and its edge detected counterpart, one can conclude that the configuration of the noise is shaped similar to the outline of the light source. In this case, the light source resembles a large oval that is center on the right side of the object. Similarly, the edge detected counterpart contains noises that are arranged in an oval fashion that is also centered on the right side the object.

Finally, for the third image, the edge detected counterpart contains an arrangement of noise that is located on the left side of the object. Yet again, the edge detected counterpart contains noise that
resembles the shape of the outline of the light source in the original image.

In the previous section, a light source that does not vary in the orientation in which it strikes an object of interest, but does vary in intensity, created various arrangements of noise in an edge detected image. On the other hand, a light source that does vary in the orientation in which it strikes an object of interest, but does not vary in intensity, produces the same results.

So far, two major problems involving the light source have been discovered that could impede the feature extraction process. Before any solution can be introduce to solve these problems, however, one more issue needs to be discussed.

### 3.3 Partner in Crime: The Physical Environment

When a certain person is attempting to identify a family member, a friend, a lover, or just anyone from a crowd of people, he or she might take a few seconds or more before finally spotting that certain someone. For instance, how long would an average person take to identify the lady wearing an inner red shirt in the image below?

![Figure 15. A Crowd of People](image15.jpg)

On the other hand, how long would it take an average person to identify the same lady in the next image?

![Figure 16. A Single Person](image16.jpg)

For the first image, if someone started to look for the lady wearing an inner red shirt from left to right, he or she would have taken more than just a second. For the second image, however, the same process would have taken no time. On the other hand, if someone started to look for the lady wearing an inner red shirt from right to left; it wouldn’t have mattered which image the person is looking at since the lady is on the far right in both images.

This simple example might seem silly and obvious. However, it illustrates a very important point. It is fairly easy for someone to identify a certain person or object in an image that contains only that person or object. In contrast, it is not as easy for someone to identify a certain person or object in an image that contains not only the object or person of interest, but additional objects or people.

Now let’s take a look at this image taken in the Electrical Engineering building in UTD.

![Figure 17. Handicap Sign](image17.jpg)

Let’s assume that the object of interest in this particular image is the handicap sign indicated by the red arrow. If someone wishes to runs this image through an image matching algorithm,
comparing it with a set of database images, how well will the algorithm recognize that this input image represents a handicap sign? Before attempting to answer this question, let's take a closer look at the edge detected counterpart of this image.

Figure 18. Handicap Sign: After Edge Detection

It appears that edge detection not only identified the outlines and boundaries of the handicap sign, but also every other objects that possess an outline or boundary as well. But this is expected since the edge detection algorithm is not built to distinguish between different objects within an image. Its only job is to extract outlines or boundaries that exist within an image. It is a human being’s responsibility to interpret if the extracted outlines or boundaries represent something comprehensible. Therefore, in this particular example, every outlines and boundaries that were identified by the edge detection algorithm that does not resemble the handicap sign is regarded as noise. Although the term noise used in this section is defined differently than the previously section, it essentially conveys the same meaning. The existence of clutters caused by a varying light intensity or orientation defeats the purpose of feature extraction and complicates the implementation of an image matching algorithm. Likewise, blindly identifying all outlines and boundaries within an image without properly recognizing the ones that actually represents the object or objects of interests, feature extraction is once again rendered useless, causing the implementation of an image matching algorithm to be cumbersome or even impossible. For that reason, one must be aware of the physical environment that surrounds the object or objects of interesting within an image.

Now let’s take a close look at another image shown below.

Figure 19. Man under Ball of Light

To a pair of naked eyes, this image taken at the UTD art gallery barn appears to be free of all the problems that were discussed in the previous example. First of all, unlike the previous image, there are no physical distractions such as a door, a window, or a fire alarm surrounding the object of interest in this image. Second, the light source striking the artwork is not oriented in any awkward angles that would cause an unusual arrangement of noises. Third, the environment in which the artwork is kept appears to be free of any variation of light intensity. Hence, one would expect the feature extraction of this image to perfectly represent the original image. Well, here is the resulting image after edge detection.

Figure 20. Man under Ball of Light: After Edge Detection

Contrary to one’s expectations, the edge detected image above still contains an amount of noise large enough to distort a fair representation of the original image. But how is this possible if the original image appears to be free of all the problems which were discussed in the previous example that would cause noise to materialize in an
edge detected image? By carefully analyzing the original image and its edge detected counterpart, the answer to this question becomes obvious.

Let’s first pick a region on the edge detected image that contains a large amount of noise. Then, let’s find the same region on the original image and enlarge both regions and see if there is any correlation between the two.

Figure 21. Comparing Enlarged Regions

By looking at the two enlarge areas in the figure above; one still can not clearly determine whether a correlation exits between the two regions. As a result, let’s manipulate the enlarged region of the original image by performing a few image processing procedures. First, the region is brightened to allow a more visible texture. Second, the hue and saturation of the region is altered to a point where one can easily distinguish any rough or uneven spots within an image. The result of the processed region is shown in the figure below along with its edge detected counterpart.

Figure 22. After Image Processing

After implementing these image processing procedures, one can now clearly see the rough peaks and valleys scattered within the enlarged region of the original image. And by comparing them with the noises that are scattered within the edge detected counterpart, one can conclude that these microscopic peaks and valleys are the cause of the noises that exist within the edge detected counterpart.

Furthermore, these rough peaks and valleys are actually microscopic edges and protrusions that are physically embedded on the walls of the UTD art gallery barn. But since they are essentially invisible to the human eyes, one can easily overlook their existence. On the other hand, the edge detection algorithm does not fail to notice them.

In summary, noises materialize within an image due to various factors. Here is a set of tables that provides a summary of each case.

<table>
<thead>
<tr>
<th>Cause of Noise</th>
<th>Definition of Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Varying Light Intensity</td>
<td>When the intensity of the light source that strikes the object or objects of interest varies, noises will exist corresponding to the level of light intensity.</td>
</tr>
</tbody>
</table>

Table 1. Causes and Definitions of Noise, part I
4.4.4. Median Filter

In a median filter, each output pixel is set to a median of the pixel values in the neighborhood of the corresponding input pixel instead of a mean because the median is much less likely to be influenced by outliers than the mean. Therefore, median filtering is better equipped to remove outliers without diminishing the sharpness of the image. During the filtering process, the input pixel is replaced by the median of the pixels contained in a window around the pixel in the following fashion:

\[
v(m, n) = \text{median}\{y(m - k, n - l), (k, l) \in W\}
\]

The variable \( W \) in this equation represents an appropriate window. \(^x\) The algorithm for median filtering requires organizing the pixel values in the window in increasing or decreasing order and choosing the median value. Normally the window is chosen so that the number of pixels in the window is odd. If the number of pixels in the window is even, the median is calculated to be the mean of the two middle values. Common windows are three by three, five by five, and seven by seven.

The median filter possesses several important properties. First, it is a nonlinear filter. \(^x\) Therefore, for two functions \( x(m) \) and \( y(m) \):

\[
\text{median}\{x(m) + y(m)\} \neq \text{median}\{x(m)\} + \text{median}\{y(m)\}
\]

Second, the median filter is best used for isolated lines or pixels while the number of noise pixels in the window is more than or half the number of pixels in the window. Third, the median filter is best useful for removing isolated lines or pixels while safeguarding spatial resolution. \(^x\)
Let’s use this image taken at the UTD art gallery barn to conduct a simple test.

![Original image](image)

**Figure 23. Man Sitting by a Giant Folder, Part I**

Here is a set of edge detected counterparts of the above image that have been processed through the median filter under various window parameters using MATLAB software.

![Edge detected images](image)

**Figure 24. Various Windows for Median Filter**

As one might notice by now after looking at the figure above, determining the most appropriate window parameter for the median filter allows the edge detection algorithm to extract the features that best represents the original image. In this particular example, the eight by eight windows appears to remove just the right amount of noise and still preserves the essential features that makes up the original image. An imbalance toward either end would results in an unsatisfactory edge detected image.

### 4.2 Wiener Filter

While a median filter tailors itself to the local median, the wiener filter filters an image in an adaptive fashion, tailoring itself to the local image variance. If a particular area on an image possesses a high variance, the filter performs a modest smoothing. On the other hand, if a particular area on an image possesses a low variance, the filter performs a more rigorous smoothing. This approach frequently generates better results than median filtering because the wiener filter is more selective than the median filter, preserving edges and other high frequency parts of an image, making it very versatile. The wiener filter has the following implementation:

\[
G(u, v) = \frac{H^\star (u, v)P_s (u, v)}{|H(u, v)|^2 + P_s (u, v) + P_n (u, v)}
\]

The term \( \frac{P_n}{P_s} \) can be understood to be the reciprocal of the signal to noise ratio. If the signal is very strong compared to the noise, \( \frac{P_n}{P_s} \) becomes zero. Contrary, if the signal is very weak compared to the noise, \( \frac{P_n}{P_s} \) becomes infinity. The ratio \( \frac{P_n}{P_s} \) is also known as the noise variance of an image.
To put the wiener filter to the test, let’s use the same image from the previous section.

Figure 25. Man Sitting by a Giant Folder, Part II

Now let’s take a look at a set of edge detected counterparts of the above image that have been processed through the wiener filter under various window parameters using MATLAB software.

Figure 26. Various Windows for Wiener Filter

Similar to the median filter, the eight by eight window appears to extract the most appropriate features. The edge detected images having window parameters lower than eight by eight still contain too much noise. On the other hand, the edge detected images having window parameters higher than eight by eight possess a distorted representation of the original image. Once again, balance is the key.

In addition, the median filter appears to work as well as the wiener filter based upon the result of these experiments. Perhaps the wiener filter does work better than the median filter on many applications. But for the purpose of noise removal, one can conclude that both filters work equally well.

4.3 Line Detection

Finally, there is yet one more method of noise removal. This method does not require the deciphering of complicated filter equations. It is simply a brute force technique that actually works as well, if not better than filters themselves under the right circumstances.

Once again, let’s use the same image along with its edge detected counterpart.

Figure 27. Man Sitting by a Giant Folder, Part III

This particular image, like so many other images taken at the UTD art gallery barn, contains a piece of art work that possesses a rectangular outline consists of four straight edges. If one was assigned the task of performing feature extraction on all the images taken in the UTD art gallery barn, then it would be advantageous for one to somehow utilize these straight edges.
If there is a way to preserve the details of the actual art work situated inside the four straight edges and remove all the noise located outside the four edges, the resulting feature extraction would perfectly represent the original image.

Well, this is exactly what the new technique will accomplish. First, let’s take notice that the edge detected image is a binary image, the white pixel values being one and the black pixel values being zero. Second, let’s attempt to first locate the longest horizontal edges in the image by searching through the entire image row by row, summing up the total number of ones on each row. At the same time, if row $x(n+1)$ contains a sum greater than row $x(n)$, the value of $n+1$ is saved into a temporary variable. Next, if row $x(n+2)$ contains a sum greater than row $x(n+1)$, the value of $x+2$ is then saved into the temporary variable, overriding the previous value. This process will iterate until all the rows have been searched. At the end of this iteration, the value in the temporary variable will contain the row in which the longest horizontal straight edge exits. This value, however, only indicates one of the longest horizontal edges in the image. And it could either be the top or the bottom edge, indicated by the figure shown below.

In order to obtain the second longest horizontal edge, the image is first divided into an upper and a lower half. Then, if the first longest horizontal edge was found in the upper half of the image, the iteration will proceed to find the longest horizontal edge in the lower half of the image. On the other hand, if the first horizontal edge was found in the lower half of the image, the iteration will proceed to find the longest horizontal edge in the upper half of the image. The process is demonstrated in the figure below.
Similarly, to find the two longest vertical straight edges, the iteration searches through the entire image column by column in the similar fashion. The process is demonstrated in the figure below.

**Figure 31. Column by Column**

Finally, when all four straight edges are obtained, a program can be easily written to convert all pixels outside of the four straight edges to black, or the value of zero. The final result for this particular image is shown below.

**Figure 32. Result after Line Detection**

Since this technique relies solely on locating the longest edges or lines in an image, it shall be known as line detection.

### 4.4 The Good and the Bad

So far, three solutions have been presented to solve the problem of noise removal. Let’s discuss the advantage and disadvantages of each solution in a set of tables below.

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages/Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td></td>
</tr>
<tr>
<td>▪ The median filter is capable of removing a large amount of noise.</td>
<td></td>
</tr>
<tr>
<td>▪ It relies on the median of the neighboring pixels, making it better equipped to remove outliers without diminishing the sharpness of the image.</td>
<td></td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td></td>
</tr>
<tr>
<td>▪ The amount of noise the median filter removes corresponds to a chosen window. If the dimension of the window is chosen too large, the median filter could remove valuable details of the image in addition to noise. If the dimension of the window is chosen too small, however, the median filter might not remove enough noise.</td>
<td></td>
</tr>
<tr>
<td>▪ Tedious tests must be done in order to choose the appropriate window for a particular image. In addition, if more than one image is required to conduct noise removal, more tests is needed to choose the appropriate window for a set of images.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5. The Advantages and Disadvantages of the Median Filter**
<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages/Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The wiener filter is also capable of removing a large amount of noise.</td>
</tr>
<tr>
<td></td>
<td>• On the other hand, instead of relying on the median of the neighboring pixels, it tailors itself to the local variance of the neighboring pixels. It is therefore much more versatile than the median filter in that it is more selective, preserving the edges and other high frequency parts of an image.</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Like the median filter, the amount of noise the median filter removes corresponds to a chosen window. A perfect balance must be struck to keep the filter from removing too much valuable information and not removing enough noise.</td>
</tr>
<tr>
<td></td>
<td>• Again, like the median filter, tedious tests must be done in order to choose the appropriate window for a particular image. In addition, if more than one image is required to conduct noise removal, more tests is needed to choose the appropriate window for a set of images.</td>
</tr>
</tbody>
</table>

Table 6. The Advantages and Disadvantages of the Wiener Filter

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages/Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Unlike the median or the wiener filter, the line detection method does not require a window to determine an amount of noise to be removed. It attempts to locate the longest horizontal and vertical straight lines in an image and preserves only the details located inside those lines.</td>
</tr>
<tr>
<td></td>
<td>• The line detection method does not need to strike a balance between removing too much detail or not removing enough noise since it always preserves the details inside the four longest lines and remove the noises outside of the four longest lines.</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Since the line detection method relies solely on detecting the longest straight lines that forms a square or rectangle, it will fail if the object or objects of interest do not possess straight lines that form a square or rectangle. In addition, if there exists a straight line that is longer than the straight lines that form a rectangle, the algorithm will fail as well.</td>
</tr>
<tr>
<td></td>
<td>• The method will also fail if the image of an object is taken in a rotated fashion shown in the figure below.</td>
</tr>
</tbody>
</table>

Table 7. The Advantages and Disadvantages of the Line Detection
The line detection method will not be able to locate the two blue lines in the figure above because the image is taken at a rotated angle. Since the line detection method searches the image row by row and column by column, it requires the longest horizontal lines in an image to be perfectly horizontal and the longest vertical lines to be perfectly vertical.

5. Conclusion

The process in which a raw image is processed through feature extraction using edge detection discussed in this paper is still fairly primitive. In addition, the various methods of noise removal discussed in this paper are also fairly unrefined. However, now that the reasons for feature extraction is clearly identified and the various causes for noise to appear in an image is clearly justified, future techniques can be improvised to improved the methods described in this paper. For instance, various algorithms mentioned in this paper can be enhanced in the following manner:

- Develop a feature extraction algorithm that relies on more than just the edges or boundaries of the original image.

- Develop a feature extraction algorithm that can fill holes and gaps in an image. More specifically, develop an algorithm that dumps ones into areas in an image where the edges form a close loop while dump zeroes everywhere else.

- Conduct further testing to find the most appropriate window parameters for the median or wiener filter to be implemented on a set of images.

- Develop a filter or algorithm that does not need to strike a balance between removing too much detail and not removing enough noise in an image.

- Develop an improved line detection algorithm that identifies not only the longest horizontal and vertical straight lines, but also the longest straight lines that orient in any angle.

Feature extraction and noise removal is a critical step prior to image matching. If the feature extraction process does not create an accurate representation of the original image due to the existence of noise, even the most robust image matching algorithm would fail to work properly. As much progress as has been made, however, in this paper as well as in current researches in the last fifty years, much more distance must be covered. The work will not be complete until a perfect feature extraction system is created possessing the ability to remove undesirable noises in addition to preserving valuable and relevant details within an image.
BIBLIOGRAPHY


