AN INVESTIGATION OF MACHINE VISION FOR GAS TUNGSTEN ARC WELDING ROBOTS

A Thesis
Presented in Partial Fulfillment of the Requirements for the Degree Master of Science
by
Russell David Richardson, B.S.E.E.
The Ohio State University
1984
ACKNOWLEDGEMENTS

I would like to thank my advisors, Professor Richard Richardson for this excellent research opportunity, and Professor David Orin for his guidance and consultation on this thesis. I would also like to express my appreciation to Professor Robert Lackey, and professor Dean Davis, for their helpful consultation to my research work.

I wish to thank David Farson, and David Gutow for their many helpful suggestions.

I am indebted to Joe Glandorf, without whom typing of this thesis would have been extremely difficult.

This research was supported by the Center for Welding Research, Ohio State University.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>MACHINE VISION LITERATURE REVIEW</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Introduction</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Machine Vision Fundamentals</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Preprocessing</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Edge Detection</td>
<td>22</td>
</tr>
<tr>
<td>2.5</td>
<td>Boundary Detection</td>
<td>28</td>
</tr>
<tr>
<td>2.6</td>
<td>Conclusion</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>PROBLEM FORMULATION</td>
<td>34</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>34</td>
</tr>
<tr>
<td>3.2</td>
<td>Gas Tungsten Arc Welding</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>Equipment</td>
<td>40</td>
</tr>
<tr>
<td>3.4</td>
<td>Previous Work</td>
<td>54</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusion</td>
<td>61</td>
</tr>
<tr>
<td>4</td>
<td>MEASUREMENT OF TWO-DIMENSIONAL WELD POOL GEOMETRY</td>
<td>64</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>64</td>
</tr>
<tr>
<td>4.2</td>
<td>The Weld Pool Image</td>
<td>65</td>
</tr>
<tr>
<td>4.3</td>
<td>Early Processing</td>
<td>76</td>
</tr>
<tr>
<td>4.4</td>
<td>Edge Detection</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>Boundary Detection</td>
<td>86</td>
</tr>
<tr>
<td>Chapter</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.6</td>
<td>Extraction of Weld Pool Variables</td>
<td>96</td>
</tr>
<tr>
<td>4.7</td>
<td>Conclusion</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>TWO-DIMENSIONAL WELD JOINT DETECTION</td>
<td>103</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>103</td>
</tr>
<tr>
<td>5.2</td>
<td>Coaxial Weld Joint Image</td>
<td>104</td>
</tr>
<tr>
<td>5.3</td>
<td>Edge Detection</td>
<td>109</td>
</tr>
<tr>
<td>5.4</td>
<td>Boundary Detection</td>
<td>112</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>114</td>
</tr>
<tr>
<td>6</td>
<td>SUMMARY AND CONCLUSIONS</td>
<td>116</td>
</tr>
<tr>
<td>6.1</td>
<td>Research Contributions</td>
<td>116</td>
</tr>
<tr>
<td>6.2</td>
<td>Research Extensions</td>
<td>117</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
<td>119</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Robotics definition</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>General process for developing machine vision</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Main processing loop inherent to most machine vision algorithms</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Robotic machine vision system</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Digital image representation</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>General $3 \times 3$ window function</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>Averaging window function used to smooth raw image data</td>
<td>18</td>
</tr>
<tr>
<td>2.7</td>
<td>A binary image and its histogram</td>
<td>20</td>
</tr>
<tr>
<td>2.9</td>
<td>Ideal edge profiles</td>
<td>23</td>
</tr>
<tr>
<td>2.10</td>
<td>Profile of real-world edge</td>
<td>24</td>
</tr>
<tr>
<td>2.11</td>
<td>Classical edge operators</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>Typical Gas Tungsten Arc welding arrangement</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>Three sub-systems used for investigation</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>Photograph of the vision system</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Block diagram of the vision system</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>Digital camera, camera control unit, and T.V. monitor</td>
<td>45</td>
</tr>
<tr>
<td>3.6</td>
<td>Block diagram of the video computer interface</td>
<td>47</td>
</tr>
<tr>
<td>3.7</td>
<td>Photograph of the GTA welding robot</td>
<td>48</td>
</tr>
<tr>
<td>3.8</td>
<td>Block diagram of the robot</td>
<td>49</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>3.9</td>
<td>Research coaxial viewing torch</td>
<td>51</td>
</tr>
<tr>
<td>3.10</td>
<td>Optic details of GTA coaxial viewer</td>
<td>52</td>
</tr>
<tr>
<td>3.11</td>
<td>Practical coaxial GTA viewing torch</td>
<td>53</td>
</tr>
<tr>
<td>3.12</td>
<td>Light intensity distribution across a weld pool at the electrode</td>
<td>55</td>
</tr>
<tr>
<td>3.13</td>
<td>Intensity distribution models</td>
<td>58</td>
</tr>
<tr>
<td>3.14</td>
<td>Ideal weld pool intensity model and its derivative</td>
<td>60</td>
</tr>
<tr>
<td>3.15</td>
<td>Second derivative model of the joint feature</td>
<td>62</td>
</tr>
<tr>
<td>4.1</td>
<td>Photograph of a coaxial GTA weld pool</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>Reflection of light from the viewing area</td>
<td>68</td>
</tr>
<tr>
<td>4.3</td>
<td>Gray-scale representation of a digital GTA weld pool image</td>
<td>69</td>
</tr>
<tr>
<td>4.4</td>
<td>Histogram equalized image of Figure 4.3</td>
<td>70</td>
</tr>
<tr>
<td>4.5</td>
<td>Light intensity contour plot of the image of Figure 4.3</td>
<td>72</td>
</tr>
<tr>
<td>4.6</td>
<td>Gray-level representation of a non-symmetric GTA weld pool image</td>
<td>74</td>
</tr>
<tr>
<td>4.7</td>
<td>Contour plot of the image of Figure 4.6</td>
<td>75</td>
</tr>
<tr>
<td>4.8</td>
<td>Summary of common characteristics of coaxial weld pool image</td>
<td>77</td>
</tr>
<tr>
<td>4.9</td>
<td>Results of smoothing weld pool image</td>
<td>79</td>
</tr>
<tr>
<td>4.10</td>
<td>Contour plots of the images of Figure 4.9</td>
<td>80</td>
</tr>
<tr>
<td>4.11</td>
<td>Histogram of the weld pool image of Figure 4.9</td>
<td>82</td>
</tr>
</tbody>
</table>
LIST OF FIGURES (continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.12</td>
<td>Edge operator developed to locate vertical edges in GTA weld pool images</td>
<td>84</td>
</tr>
<tr>
<td>4.13</td>
<td>Binary image of edge points in a weld pool image</td>
<td>87</td>
</tr>
<tr>
<td>4.14</td>
<td>Decomposition of Figure 4.13 into weld pool and electrode edge points</td>
<td>89</td>
</tr>
<tr>
<td>4.15</td>
<td>Details of radial representation</td>
<td>91</td>
</tr>
<tr>
<td>4.16</td>
<td>Resultant radial representation for the image of Figure 4.13</td>
<td>93</td>
</tr>
<tr>
<td>4.17</td>
<td>Plots of the boundary of Figure 4.16</td>
<td>94</td>
</tr>
<tr>
<td>4.18</td>
<td>Final result of processing the weld pool image of Figure 4.13</td>
<td>97</td>
</tr>
<tr>
<td>4.19</td>
<td>Extraction of pool width</td>
<td>98</td>
</tr>
<tr>
<td>4.20</td>
<td>Extraction of pool area</td>
<td>100</td>
</tr>
<tr>
<td>4.21</td>
<td>Summary of processing algorithms</td>
<td>102</td>
</tr>
<tr>
<td>5.1</td>
<td>Gray-level representation of a coaxial weld joint image</td>
<td>105</td>
</tr>
<tr>
<td>5.2</td>
<td>Intensity contour plot of a joint image</td>
<td>106</td>
</tr>
<tr>
<td>5.3</td>
<td>Plot of a row of a joint image</td>
<td>107</td>
</tr>
<tr>
<td>5.4</td>
<td>Plot of row of a smoothed joint image</td>
<td>108</td>
</tr>
<tr>
<td>5.5</td>
<td>Plots of several rows of a joint image</td>
<td>110</td>
</tr>
<tr>
<td>5.6</td>
<td>Binary images of edge points of the Figure of 5.1</td>
<td>113</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

The modern manufacturing engineer has recognized the need to automate in order to compete in the world market. Because of the microprocessor revolution of the seventies, it has become feasible to dedicate an entire computer to control a single manufacturing task. With the steady increase of technology more and more complex tasks will be controlled by microprocessors.

One of the most active areas of research in the automation field is "robotics". "Robotics" may be shown as a subdivision of the science of "automation" as given in Figure 1.1. Figure 1.1 emphasizes the key difference in a robot and an automated industrial machine; robots are programmable. This means that one robot is flexible enough to perform a number of different tasks.

Robots have been used since the sixties to automate parts handling and other tasks requiring tool movement. But it wasn't until the advent of the microprocessor that it became feasible to integrate sensor feedback within the robot itself. With added sensing a number of new applications for robots emerged, for example, arc welding.
Figure 1.1 Robotics is a subdivision of the science of automation.
Though severely limited in application, today's industrial robots do their jobs with precision at rates unattainable by their human counterparts. But at best these tireless robots use limited sensor technology. The next generation robot promises to be more sensory based and perhaps mobile. One step in increasing a robot's sensing is the addition of machine vision. With the addition of vision it has been estimated that as many as half of the 600,000 routine factory jobs in the United States could be performed by industrial robots [2].

Of the estimated 8000 robots in U.S. factories, only 800 have any type of vision [2]. Those that do have vision are expensive and reliability is low. One reason for this is the stage of development of machine vision; it is still an embryonic technology [3]. Since the field is so young, the process of vision is not well understood. It is not only vision understanding that needs to be developed, vision is pushing the resources of today's computers to the limit. To use sophisticated machine vision, computer speed needs to increase by a factor of "100 to 1000" beyond today's capabilities [4]. The lack of computer speed and image understanding is the main detriment in robotic machine vision.

This thesis describes one way to develop machine vision for a specific manufacturing robot, the industrial gas tungsten arc (GTA) welding robot. Vision (either by human or machine) is
especially important in welding because it can be made non-invasive; the welding environment is far too harsh for most other types of sensors. This process has been chosen since it is probably one of the easiest for which to develop a visual control. Arc welding mechanization has been successful for years, and the GTA process is the easiest of the various arc welding processes to sense visually. This process uses a non-consummable electrode, eliminating the spatter and smoke of other forms of arc welding. Also, since the arc is very stable, there is less erratic behavior than other types of welding [5].

This investigation of machine vision for arc welding robots is made possible by a recently developed weld vision system [6]. Part of the success of this vision system has been the use of the so-called coaxial weld viewing concept [7]. This viewing approach utilizes an optical viewing arrangement which aligns the optical axis with the welding electrode. A number of advantages result including masking of the bright arc by the electrode shadow and a clear view of the joint, electrode, and the entire periphery of the weld pool.

The specific objective of this work has been to develop a method for the two-dimensional image analysis of the coaxial weld pool image. This includes the identification of the pool edge around the entire periphery of the weld pool, the joint between the weldments being welded, and the electrode. With a knowledge
of the location of the weld pool and joint, the robot can be made to seam track with pool centering. That is, the robot will be able to guide its welding torch along the joint between the pieces being welded keeping the molten weld pool centered on the joint. Since the entire shape of the weld pool is calculated, the area and width of the pool can be found. This information is important for weld quality assurance. It also provides an indirect indication of weld penetration which is often used as one gauge of weld quality [8].

The work has proceeded in three steps. First, available literature on machine vision, along with its applications to arc welding, was researched. Chapters 2 and 3 summarize the results. Second, weld pool images were studied off-line (on a PDF-11) to characterize the image and to test recognition algorithms. Chapter 4 details this work. Third, weld joint images were analyzed to locate the joint position and width in real-time. This is discussed in Chapter 5. Chapter 6 provides some conclusions and possible extensions of the work.
CHAPTER 2
MACHINE VISION LITERATURE REVIEW

2.1 Introduction

Machine vision is the ability for a machine to construct explicit, useful descriptions of physical objects from digital images [9]. Descriptions or representations are the first step for recognizing, classifying, and "thinking" about the objects. Machine vision differs from image processing, which tries to improve pictorial information for human interpretation by image-to-image transformations. Machine vision instead is a collection of techniques including not only image processing (transforming, encoding, and transmitting images), but also, artificial intelligence (ability of computers to reason as biological organisms), and statistical pattern classification (statistical decision theory applied to general patterns). Unlike other techniques, machine vision's goal is to allow autonomous machine perception by understanding an image from the processing of scene data.
Figure 2.1 illustrates the general process for developing machine vision [10]. It is a three step process. The first step is to choose an imaging system and a suitable evaluation criteria for the application of machine vision. The second step is to compile a useful list of a priori knowledge and use the list to derive the task's objectives and constraints. The third step is to write, test, and modify algorithms. The last step is repeated until the machine performs within the limits set by the evaluation criteria.

One important element in machine vision is that the algorithms must operate in real time. That is, the main processing loop time must be smaller than the largest time constant of the machine that is being controlled. This restriction severely limits the type of algorithms that can be used in machine vision. It has also lead to the specific structure of the main processing loop given in Figure 2.2 [10].

The main loop of Figure 2.2 is inherent in most machine vision applications. The first step, preprocessing, is low-level processing that is implemented in hardware to decrease computer program execution time. This will be discussed in Section 2.3. Next is low-level processing which transforms the image into an intermediate representation that is more easily understood than the gray-level intensity function of an image. This will be discussed in Section 2.4 with an emphasis on edge detection.
Figure 2.1 General process for developing machine vision[10].
Figure 2.2. Main processing loop inherent to most machine vision algorithms [10].
Section 2.5 which deals with boundary detection, discusses the processing of the intermediate representation to derive a representation or description of the image. The last section, 2.6, discusses decision theory as it applies to machine vision.

2.2 Machine Vision Fundamentals

A typical system used for robotic machine vision is illustrated in Figure 2.3. The purpose of the system is to provide a flow of information from object to robot. The objects in the scene reflect light, forming an image that passes through filters to a sensor. The sensor digitizes the image and the results are preprocessed. The preprocessed image is gathered in computer memory and the machine vision algorithm goes through one iteration of the main loop feeding the results to the robot controller, display, and/or storage device. Details are discussed in the following paragraphs.

The term image refers to a two-dimensional light intensity function, denoted by \( f(x,y) \), where the value of \( f \) at spacial coordinates \((x,y)\) gives the amplitude of the intensity of the image at that point [11]. The function \( f(x,y) \) can be separated into two components, illuminance denoted by \( i(x,y) \), and reflectance \( r(x,y) \). Illuminance is the amount of light incident on the scene, and reflectance is the amount of light reflected by the objects in the scene. The components combine as a product to
Figure 2.3 Typical system used in robotic machine vision.
form \( f(x,y) \):

\[ f(x,y) = i(x,y) r(x,y) \]  \hspace{1cm} (2-1)

The illuminance has values from 0 to infinity and is a property of the light source. Reflectance has a range from 0 to 1 and is a property of the objects in the scene.

The image is preprocessed in the analog world before being sensed. This processing step is often attenuation using neutral density filters and/or bandpass filters. Neutral density filters are needed if the image is too bright for the sensor, and bandpass filters are used to limit the spectral bandwidth of the image before it is sampled.

The image function \( f(x,y) \) must be sampled spatially, and digitized in intensity, to be suitable for digital processing. Sampling in spacial coordinates \((x,y)\) is completed at the sensor and is called "image sampling". Each sampled intensity is digitized at the digitizer, this process being referred to as "gray-level quantization". At this point the continuous image \( f(x,y) \) is approximated by samples with integer values from 0 to \( G \) arranged in an \( N \times M \) array. The quantity \( G \) is the number of gray-levels that the digitizer can distinguish and \( N \times M \) is the number of spacial samples (called picture elements or pixels) that the sensor produces.

From the preceding paragraph, it is evident that the larger \( G, M, \) and \( N \) the closer the digital image approximates the
continuous image. However, the real-world limits of storage and
time requires careful selection of these parameters. With today's
technology, images for use in machine vision applications are
usually on the order of 512 x 512 x 256 (N x M x G) [11].

Figure 2.4 summarizes the representation that will be used
for digital images. The continuous image \( f(x,y) \) is approximated
by the digital image \( f'(x,y) \). Each square represents a small area
of the image called a pixel. Each pixel has an integer value from
0 to G. The first row is the top of the image and the first
column is the left side of the image. The digital image is
thought of as a 2-D integer array with row limits from 1 to N, and
column limits from 1 to M.

A small portion of the image is called a window. A window
can be considered a subset of the image array. A window function
is a function that operates on windows to produce a single pixel
value that replaces the center pixel in the window. It is
represented as an array, like a window, but the values that fill
the array are called weights which can be real or integer. An
example of the general 3x3 window function \( W \) is given in Figure
2.5. The weights are not numbered like pixels; instead, the
center weight is given the (0,0) coordinate as illustrated. A
window function is designed to perform a particular operation by
properly selecting its size and weights. Like many window
functions the 3x3 window is used for the two dimensional
Figure 2.4 Digital image representation.

\[ f'(x,y) = \approx f(x,y) \]
<table>
<thead>
<tr>
<th>$W(-1,-1)$</th>
<th>$W(-1,0)$</th>
<th>$W(-1,1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W(0,-1)$</td>
<td>$W(0,0)$</td>
<td>$W(0,1)$</td>
</tr>
<tr>
<td>$W(1,-1)$</td>
<td>$W(1,0)$</td>
<td>$W(1,1)$</td>
</tr>
</tbody>
</table>

Figure 2.5 General 3x3 window function with weights $W_{ij}$. 

15
convolution [8]:

\[ B(i,j) = \sum_{a=-1}^{1} \sum_{b=-1}^{1} W(a,b) A(i-a,j-b) \]

where \( 1 < i < N \)

\( 1 < j < M \)

The new image \( B \) is the result of convolving the window \( W \) with the original image \( A \). Windows exist for many different operations including low and high-pass filtering, Laplacian operators, and various feature detectors [12].

2.3 Preprocessing

Once an image has been converted to a digital image, low-level processing begins. It is advantageous to perform this processing as fast as possible to decrease the main loop time. In order to speed up the processing, it may be desirable to perform some or all low-level processing steps in hardware instead of software. That part of the processing that is done in hardware is called preprocessing. Preprocessing is usually done digitally (i.e. on data output from the digitizer) but may also be accomplished with analog processing of signals between the sampler and digitizer.

One common type of preprocessing is image smoothing [11]. Smoothing is necessary to reduce the noise that is introduced into the image by the sensor, digitizer, and transmission channel. The simplest and most easily realizable (in hardware) image smoothing
operation uses an averaging window (Figure 2.6). The purpose of averaging is to deemphasize high frequency components of the image by transforming each pixel value into a new value that is the average of the 9 pixels in its 3x3 neighborhood, i.e.

\[ B(i,j) = \frac{1}{9} \left( \sum_{a=-1}^{1} \sum_{b=-1}^{1} A(i-a,j-b) \right) \] (2-3)

Some processors use a similar smoothing operation with 5x5 and 7x7 windows.

A major disadvantage of a linear smoothing operation, like the 3x3 averaging window, is the degradation of sharp edges. The detection of these features is often the object of later processing, and the smoothing may make these edges impossible to locate. For this reason, other smoothing techniques that preserve edges have been devised. One non-linear smoothing operation that is completed in a preprocessing stage is median filtering. Each of the filtered image pixels are the median or middle value of the sorted pixels in its neighborhood. For example, to calculate the median of a 3x3 window, the nine pixels in the window are sorted by magnitude and the fifth value in the ordered list becomes the new pixel value for the center of the window. Since the median filter preserves edges, it is preferable to an averaging window. However, averaging windows are more common since they are more easily implemented.

Smoothing is not the only low-level processing that is
Figure 2.6 Averaging window function used to smooth raw image data.
completed in the preprocessing stage. Another step commonly performed at the preprocessing level is the calculation of the image's gray-level histogram [13]. The histogram of an image is a function giving the frequency of occurrence of each gray level. Essentially, it is a tabulation of the number of times each possible pixel value has occurred in the image.

The gray levels, \( g \), in an image may be considered random quantities in the interval \((0, G)\) [11]. Therefore, the gray levels can be characterized by their probability density function \( p'(g) \). The image histogram is equivalent to this density function. For example, if an \( 8 \times 8 \) (\( N \times M \)) digital image has \( 16 \) (of \( 64 \)) pixels with a gray level \( g \) of 25, then the probability of a randomly chosen pixel having gray level \( g=25 \) is written \( p'(25) \), and is equivalent to \( 16/64 = 1/4 \), i.e., \( p'(25)=1/4 \). Note that this is independent of the number of gray levels, \( G \).

Many low-level as well as high-level processing steps use this density function as a statistical indication of image properties. One such use is for automatic threshold adoption [14]. Given an image like the one in Figure 2.7 along with its histogram, the problem is to find the gray-level threshold that separates the object and background, thus forming a binary image. To find the threshold, the probability density function is processed to find the local minima between the normal distributions (peaks). This local minima corresponds to the
Figure 2.7 A binary image and its histogram [8].
optimal threshold level.

Another use of histograms is to enhance the image by contrast stretching. If an image has a narrow range of values occupied by the pixels (small dynamic range), the image can be improved by increasing the range of values (increasing dynamic range). This technique is called contrast stretching. One method that maximizes dynamic range is called histogram equalization [10]. This method transforms the image into one with a constant probability density function. The idea is to transform the original gray levels, $g$, to new levels, $s$, in a way that $p'(s)$ is as close to a constant value as possible. This transformation can be accomplished by the relation

$$s_k = \sum_{j=0}^{k} p(r_j).$$

(2-4)

where $s_k$ is the $k$th gray level of the new image and $p(r_j)$ is the probability of occurrence of the $j$th gray level in the original image.

Another important reason for calculating a histogram is for automatic iris control of the vision sensor. If the histogram indicates that most pixels have very high values, it is desirable to close the iris to decrease the amount of light incident on the sensor. Similarly, if the histogram shows that most pixels have a low value, the image is too dark and the iris should be opened.
2.4 Edge Detection

A small area in an image where the local gray levels are changing abruptly is called a local edge [15]. Edge detection is the low-level process of finding the local edges in an image. Its purpose is to reduce the complexity of a digital image to a representation that is more easily understood.

Ideal local edges in digital images have one of three different profiles (Figure 2.9) [9]. The most common is a step edge (Figure 2.9a). A step edge is the boundary between 2 pixels with respective brightness values significantly different. Therefore, a step edge exists between two pixels when one is inside a brighter region than the other. The second most common type of ideal edge is the roof edge (Figure 2.9b). A roof edge exists at the point of change between brightness values steadily increasing and steadily decreasing (local extrema). The third type of ideal edge is the ramp edge shown in Figure 2.9c. Edges in real-world digital images are a combination of the 3 ideal edges with the addition of noise (Figure 2.10).

The classical methods of local edge detection are based on edge operators. An edge operator is a window function with small spacial extent designed particularly to find local edges. Edge operators are used since they do not rely on models of the edge. Since local edges have many profiles, various edge operators have been developed. Figure 2.11 illustrates several classical edge
Figure 2.9 Ideal edge profiles that are commonly encountered in digital images [9].
Figure 2.10 Profile of a real-world edge in an image.
Figure 2.11 Classical edge operators used in machine vision.
operators that are used in machine vision.

To use an edge operator to calculate the location of a local edge, the edge operator is convolved with the digital image to calculate a new image. Edge points will have higher absolute values than non-edge points. Therefore, a threshold is selected to separate the new image's pixels into pixels that are, and are not, edge points. The threshold should be chosen to make the conditional probability of assigning an edge given that there is an edge, equal to the conditional probability of there being a true edge given that an edge is assigned [16]. True edges are defined to be the two pixel wide region in which each pixel neighbors some pixel having a value different from itself. The preceding process will reduce the digital image $f'(x,y)$, with $G$ gray levels to the binary image $b'(x,y)$. The binary image is an intermediate representation of the image where "1" is a local edge pixel and "0" is not a local edge pixels.

Classical edge operators are all based on the principle that edge regions of images have high spacial derivative content. The window function performs either a gradient or Laplacian operation on a local area of the image. A gradient window function will produce high values whenever a step edge is present, and the Laplacian window function will produce high values whenever roof or ramp edges are present. Since these operators perform differentiation, they are noisy and produce false edge points.
One way to improve performance is to smooth the image first with a smoothing window. Another improvement is gained if the window size is increased. Unfortunately, the added smoothing and window size are very costly in both computation speed and edge location resolution.

Modern edge detectors do not calculate the edge locations directly from the pixel values. Instead, they assume that the digital image came from sampling a real-valued function, \( f(x,y) \), defined on the domain of the image which is a bounded and connected subset of the real plane \( \mathbb{R}^2 \). Therefore, a parametric form for \( f(x,y) \) can be assumed [17]. For example, \( f(x,y) \), could be approximated by the third order polynomial

\[
    f(x,y) = k_1 + k_2 x + k_3 y + k_4 x^2 + k_5 xy + k_6 y^2 + k_7 x^3 + k_8 x^2 y + k_9 xy^2 + k_{10} y^3. \tag{2-5}
\]

The parameters \((k_1, k_{10})\) in above example are calculated from the digital image data. The gradient and second directional derivative are then calculated from the polynomial \( f(x,y) \). An edge is said to occur at a pixel if and only if there is some point in the pixel’s area having a negatively sloped zero crossing of the second directional derivative taken in the direction of a non-zero gradient at the pixel's center [16]. The modern edge detector does not find locations of high gradient, but finds locations of spatial gradient maxima. Many parametric forms for
f(x,y) have been developed along with techniques for calculating parameters and direction derivatives [18,19].

Modern edge detection algorithms perform much better than classical ones and should be used whenever possible. However, most machine vision applications are too time critical to apply modern edge detection.

2.5 Boundary Detection

The reduction of the digital image f'(x,y) to the binary image b'(x,y) is the result of edge detection. Processing of the binary image must then be completed to group edge elements into structures better suited to the process of computerized interpretation. Boundary detection is that process of collecting edge elements into a coherent one-dimensional (edge) feature. It is perhaps the most important link between raw image data and its interpretation.

Boundary detection methods are classified by the amount of a priori knowledge incorporated into the grouping operation [9]. If much is known about the boundary, such as its approximate location, type of curvature, and noise processes, then the form of the boundary and its relation to other image structures is highly constrained. In this case, high knowledge detection algorithms are used. Conversely, if little is known about the boundary, than low-knowledge techniques are implemented. Low-knowledge routines
use general world knowledge and heuristics that are true for most images.

A well known high knowledge method for boundary detection is the Hough transform [20]. It detects boundaries whose shapes can be described in an analytical or tabular form. To introduce the method, the problem of detecting straight lines in an image will be discussed. It is assumed that the image has been processed and the binary image \( b'(x,y) \), is available. The Hough technique organizes these points into straight lines and decides which straight line best explains the data.

The general form of a line in the image \( b'(x,y) \), is \( y = mx+c \). The first step in detecting all lines in the image by the Hough transform is to list all the possible values of \( m \) and \( c \) that could describe these lines. To do this, the possible orientation and location of the lines must be known. Next, an array is formed \( A(c,m) \), such that one entry in the array corresponds to each possible combination of \( c \) and \( m \). The array elements are initially set to zero. Then, for each point \((x,y)\) such that \( b'(x,y)=1\), all locations, \((c,m)\), in the parameter array, \( A(c,m) \), for which \( c=-mx+y \) are incremented, i.e.,

\[
A(c,m) = A(c,m) + 1.
\]

(2-6)

The last step is to process the array, \( A(c,m) \), to find the locations of local maxima. Each local maximum in \( A(c,m) \) indicates a line in the image with parameters \((c,m)\).
Many other techniques exist for boundary detection. Graph searching is a high-knowledge method that represents the image of edge elements as a graph [21]. Thus, a boundary is a path through the graph. Dynamic programming is a low-knowledge technique that uses a mathematical formulation of the globally best boundary [22]. Contour following is a popular low-knowledge technique that finds blobs in the image [9]. Contour following works best in images that are noise free.

2.6 Decision Theory

Decision theory appears in machine vision whenever some collection of operations have been applied to the image and a decision is made as a result. As an example, assume that an image contains one of two objects, d1 and d2, and it is necessary to process the image and decide which of the two is present. The image is processed to obtain a measurement vector, \( \mathbf{x} \), which is an indication of the features in the image. One decision rule based only on probabilities is

\[
P(d1|\mathbf{x}) > P(d2|\mathbf{x}) \implies x \epsilon d1 \tag{2-7}
\]

\[
P(d2|\mathbf{x}) > P(d1|\mathbf{x}) \implies x \epsilon d2 \tag{2-8}
\]

which means, if the probability of object one being in the scene given the measurement vector (i.e. \( P(d1|\mathbf{x}) \)) is greater than the probability of object two being in the scene given the measurement vector (i.e. \( P(d2|\mathbf{x}) \)), then decide that object one is in the
scene (i.e. $\bar{x} \in d1$) [23]. This decision rule minimizes error and was first used by Bayes. Bayes also developed a method for calculating the conditional probabilities, namely,

$$P(d1|\bar{x}) = [P(\bar{x}|d1)P(d1)]/P(\bar{x}).$$  \hspace{1cm} (2-9)

There are many other decision rules, such as, Bayes rule for minimum risk, minimax test, Neyman-Pearson rule, to name a few [23].

With every decision rule there is a possibility of error. Measuring this error gives one indication of how well the decision rule works. One way to measure this error is the Bayesian approach [23]. The Bayesian approach considers the average or "Bayesian error" denoted by $e$. Let $R1$ be the region in the domain of $\bar{x}$ for which the measurement vector indicates object $1$ is in the image. Similarly, let $R2$ be the domain of $\bar{x}$ which indicates object 2. Then, when $\bar{x}$ is in $R1$, $P(d1|\bar{x}) > P(d2|\bar{x})$ and when $\bar{x}$ is in $R2$ the $P(d2|\bar{x}) > P(d1|\bar{x})$. The probability of error is

$$e = P(\text{error}) = P(\text{error}|d1)P(d1) + P(\text{error}|d2)P(d2) \hspace{1cm} (2-10)$$

where

$$P(\text{error}|d1) = p(\bar{x}|d1) \hspace{1cm} (2-11)$$

and

$$P(\text{error}|d2) = p(\bar{x}|d2). \hspace{1cm} (2-12)$$

The Bayesian approach does not work well for images containing a large number of possible objects. For example, assume that the image could contain one of a hundred objects
instead of one or two. To decide which object is in the image using a direct Bayesian approach, the decision function, \( P(d_i | \bar{x}) \), must be calculated for all objects, \( d_i \) (i=1 to 100). The highest valued decision function indicates which object is in the image. This process is too time consuming for machine vision so other related methods have been developed.

A popular solution to this problem is using a decision tree [23]. A decision tree has the property that the measurement vector, \( \bar{x} \), is subjected to a hierarchy of decision rules before being assigned to a specific object. For example, assume 1 of 4 objects (small disk, large disk, small square, large square) could appear in the image. The first decision rule in the hierarchy could choose between objects with straight lines and those without. Two decision rules would exist on the second level of the tree; one for each of the choices made at the first level. The rules may be choosing between objects with a small diameter from those with a large diameter, and choosing between objects with long sides from those with short sides. If the hierarchy is well designed, the decision tree is accurate, flexible, and computationally efficient.

2.7 Conclusion

This chapter has reviewed much of the information necessary to understand the problems, as well as some solutions, of machine
vision. Perhaps, the most important thing to realize is how critical computation time is. The processing must work in real-time which severely limits the algorithms that can be used. In the next chapter, the problem of applying machine vision to the control of Gas Tungsten Arc welding robots is detailed.
CHAPTER 3
PROBLEM FORMULATION

3.1 Introduction

Arc welding has not been widely automated with conventional machinery. Richardson [5] states three reasons:

1) A high degree of dexterity of gross motion of the weld torch or gun is required for most applications.

2) A high level of manual interaction is required for proper placement and manipulation of the arc and weld pool.

3) Most arc welding operations are low volume, batch production oriented.

Robotic technologies promise to provide some of the above capabilities. However, even with industrial robots, arc welds of consistent quality are difficult to produce. Inconsistent quality occurs due to varying alignment of parts to be joined, varying
heat sink conditions, subtle variations in the arc behavior and variations in material properties [15]. Since welding productivity can be increased dramatically with the properly controlled robots, much work is being done to develop sensory control (closed-loop) systems to solve robotic arc welding problems.

One approach to sensory control is to emulate the manual arc welder. The manual welder provides two interacting control functions in arc welding. First, by visual observation of the arc, weld pool, and joint, he guides the welding arc and pool relative to the weld joint. Since the center of the pool is guided along the joint, this function is called pool-centered joint tracking. Second, by visually observing the size and shape of the weld pool, the welder controls the arc heat, wire feed rate, and rate of arc travel. This function is called weld pool geometry control. Both control functions require the manual welder to see the arc, weld pool, and joint features in the vicinity of the weld pool. This implies that a sensory control, that emulates the manual arc welder, would involve machine vision technologies.

Arc welding application places different requirements and operational constraints on vision systems than the more common inspection and material handling applications. Therefore, a vision-based sensing and control system that is especially
suitable to arc welding has been developed [6]. Part of the success of this vision system has been the use of so-called coaxial weld viewing. This viewing approach utilizes an optical viewing arrangement which aligns the optical axis with the welding electrode. A number of advantages result including the masking of the bright arc by the electrode shadow and a clear view of the entire periphery of the weld pool. The problem is to autonomously detect the weld pool boundary and the electrode in a digital image produced by the vision system.

This chapter discusses Gas Tungsten Arc welding and its control (Section 3.2). It also discusses the equipment, including the coaxial viewing vision system, used to investigate the application of machine vision to arc welding (Section 3.3). In Section 3.4 previous investigators' work, that used the coaxial viewing concept, is detailed.

3.2 Gas Tungsten Arc Welding

Gas Tungsten Arc (GTA) welding is a fusion process used to join metal [24]. Fusion is the result of melting the metal at the workpiece (weldment) edges by an electric arc between a non-consumable tungsten electrode and the weldment. The tungsten electrode is not consumed because of the high melting point of tungsten. In other types of welding, such as Gas Metal Arc Welding, a metal wire electrode is constantly fed into, and melted.
by, the arc.

A typical GTA welding arrangement is shown in Figure 3.1. The welding power supply is typically a constant current source with an output from as little as 15 amps to as much as 1000 amps. The workpiece is usually at ground potential with the electrode being at negative potential. The electrode and weldment are protected from high temperature oxidation by an inert gas (typically helium or argon) shield. If the joint is wide, extra material may be needed to build up the weld bead. This is accomplished by feeding filler wire directly into the molten weld pool. The entire torch requires water cooling if high currents are used.

Gas Tungsten Arc welding is used whenever precision and quality of welding are of utmost importance. This is because it is the most controllable of the arc welding processes [5]. Since the process does not involve a consumable electrode, it is free of spatter, smoke, and much of the erratic behavior of consumable electrode processes. Some applications include high pressure pipes, turbine engine parts, heat exchangers, transistor cases and batteries.

There are three main areas of GTA welding process control. First is guidance. Guidance refers to the location of the solidified bead relative to the original prepared seam or groove. Ideally, the solidified bead is centered on the seam or groove at
Figure 3.1 Typical Gas Tungsten Arc (GTA) welding arrangement [24].
all points on the weldment. The second area of GTAW process regulation is weld bead penetration control. For good weld quality, the penetration must be kept constant at some predetermined depth. The third area is the control of reinforcement or joint fill. Reinforcement is that part of the weld bead that lies above the plane formed by the original surface of the weldment. Reinforcement should be kept constant during the entire weld at some predetermined set point.

Various researchers have investigated these three areas and a number of techniques for autonomous control have been reported. One class of techniques that has potential to solve the problems of all three areas uses visual imaging.

Visual imaging approaches fall into two classes, utilizing non-structured or structured light. Traditionally nonstructured light systems [25,26] view in advance of the point of welding, allowing for guidance, or directly in front of the electrode [8], allowing for penetration control. Reinforcement control is not possible with nonstructured light since the three-dimensional information needed is not available. Structured light systems [27] utilize a pattern of light projected on the point of welding or in front of the point of welding. Recognition of the reflected pattern of light from the weldment allows for a three-dimensional contour of the joint and/or weld bead. These techniques have reported limited success, but both suffer from being intrusive and
directional.

The problems of intrusion and directionality has led to the development of a concept (termed coaxial viewing) in which the imaging system is integrated, literally, into the weld torch. The integration is such that the point of welding is viewed coaxially with the welding electrode from within the weld torch. Since the imaging system is built into the torch it is non-intrusive. It is also non-directional because the weld area viewed is the same regardless of the direction the torch is moving.

3.3 Equipment

The system used to investigate machine vision applications to GTA welding robots is comprised of three sub-systems: a vision system, a GTA welding robot equipped with a coaxial viewing torch, and a minicomputer. A block diagram illustrating the connections between the three is shown in Figure 3.2.

The vision system [6] is comprised of a solid state video camera, a programmable frame buffer, a microcomputer, a video tape recorder and a T.V. monitor. Figure 3.3 is a photograph of the vision system. The microcomputer and its floppy disks are on the top shelf, the video tape recorder is beneath the computer and the camera control unit is beneath the recorder. The camera head and T.V. monitor are not shown. The vision system collects the image data, and either analyzes it and outputs the results, or stores
Figure 3.2 Block diagram showing interconnection of the three sub-systems used to investigate machine vision for GTA welding robots.
Figure 3.3 Photograph of the vision system [6].
the data.

A block diagram of the vision system is shown in Figure 3.4. The camera is mounted in the coaxial viewing system and senses the welding area. The digital video data from the camera is collected by the Video Computer Interface (VCI), a programmable frame buffer/memory built to collect the high speed (4.5 MHz) video data. The microcomputer accesses the data in the VCI, and either stores it, sends it to the minicomputer, or analyzes it to extract the necessary information, and sends the results to the robot or display.

The vision system utilizes the General Electric TN2500 solid state camera [26]. The camera head is very compact and durable due to the use of a solid state Charge Injected Device (CID) imaging chip. The imaging chip has 244 rows by 248 elements that are spatially fixed. Each element corresponds to one pixel. The camera head shown on the left in Figure 3.5 is connected to the camera control unit via a shielded cable. The closed circuit T.V. monitor (Setchell-Carlson model 10 m 915) is on top of the camera control unit. The camera controller outputs its data in closed-circuit T.V. format for the monitor, and in 8-bit digital parallel format for digital processing. All 244 rows of data are output consecutively (sequentially) every 1/30 of a second.

The camera has an added feature called inject inhibit (IIG) which allows the CID imaging chip to integrate the image for
Figure 3.4 Block diagram of the vision system [6].
Figure 3.5 The GE TN2500 camera and the Setchell-Carlson monitor (the unit under the monitor is the camera control unit)[8].
longer than one frame time. This is analogous to lengthening the exposure time, and is useful for low light level scenes.

The microcomputer is the Aim 65 by Rockwell International [29]. This 6502-based microcomputer includes an 8-bit parallel port to the welding robot and a serial (RS-232) port to the minicomputer.

The programmable frame buffer (VCI) allows for the collection of up to 4064 (about 1/15 frame) bytes of video data from the possible 60,512 (244x248) bytes. The collected data must reside in a rectangular area of the image, but the location and size of the rectangle is fully programmable. A block diagram is given in Figure 3.6. A complete description of the VCI along with the operating software is given in [8].

The robot used is a simple 3-axis rectilinear gantry under control of a Aim 65 based robot controller. Figure 3.7 is a photograph of the robot and 3.8 is a block diagram. The robot controller controls the speed and location of the welding torch as well as the gas flow and welding power supply current output.

The vision system communicates with the robot through a high speed parallel data link. The link is fully isolated at both ends for maximum noise immunity. With the interface all the various robot parameters such as speeds, voltage, and current can be controlled by the vision system.

The robot is equipped with a coaxial viewing GTA weld torch
Figure 3.6 Block diagram of the video computer interface [8].
Figure 3.7 Photograph of the GTA welding robot [8].
Figure 3.6 Block diagram of the robot [8].
developed by Richardson [7]. It represents a significant breakthrough in GTA weld area viewing. A photograph of the viewing arrangement is shown in Figure 3.9 mounted to the welding robot. The optical layout is further detailed in Figure 3.10. This viewing arrangement differs from other methods that view the weld area obliquely by viewing the area from directly above. Since the optical axis is coaxial with the electrode, it is termed coaxial viewing.

The coaxial viewing torch shown in Figure 3.9 has a number of inherent advantages. The bright arc is masked by the electrode and the entire periphery of the weld pool can be viewed. The torch is also non-intrusive and non-directional. Details of coaxial viewing can be found in [7].

The coaxial viewing arrangement in Figure 3.9 is a laboratory apparatus and is not practical for commercial use. Most of the work in this thesis was completed using this arrangement.

Shown in Figure 3.11 is a recently developed commercial torch that uses the coaxial viewing concept. The main difference is the use of fiber optics to guide the image from the torch body to the camera head. Previously, mirrors were used with a right angle torch. Unless stated, the laboratory apparatus was used in all work described in this thesis.

The minicomputer is a PDP 11/34A under the RSX operating system. It is used for off-line analysis of weld pool images. It
Figure 3.9 Photograph of the coaxial viewing GTA weld torch mounted on the robot [7].
Figure 3.10a Line drawing showing component layout of GTA viewer of Figure 3.9.

Figure 3.10b Optics details of GTA coaxial viewer system [7].
Figure 3.11 Photograph of a practical coaxial GTA viewing torch.
can be programmed in Fortran and has a graphics package for plotting 3-D images.

3.4 Previous Work

This section discusses the previous work completed at the Center for Welding Research that pertains to machine vision for GTA welding robots equipped with the laboratory coaxial viewing torch. This work has utilized analysis of the coaxial image to measure and control the width of the weld pool, and to detect a butt-joint feature to provide joint tracking, for full penetration, GTA welding. These efforts have used one-dimensional analysis of the image. That is, a single, appropriate video line in the image was selected for analysis to identify pool and joint features. This work has served to demonstrate the concept of vision-based regulatory control of the weld process. It has also led to understanding of edge features in the weld pool image.

The first investigator was Rao [30]. Rao attempted to process a single video line positioned at the electrode, oriented perpendicular to the direction of travel. His goal was to detect the width of the weld pool, in real time, and display the results on the monitor. He developed the model of the ideal image intensity distribution of the single video line (described above) shown in Figure 3.12.

A ideal weld pool image, Rao concluded, would consist of 4
Figure 3.12 Model for the idealized light intensity distribution across a weld puddle at the electrode (viewed through the GTA welding torch) [30].
regions. The first region is a low intensity disk in the middle of the image that corresponds to the electrode shadow. In the single video line of Figure 3.12 this region is labeled Zone A. The second region is the weld pool (molten metal). This region starts with a high intensity in its middle and diminishes to a slightly higher intensity than the electrode shadow at its edges. This area is labeled Zone B. Zone C in the ideal model corresponds to the edge of the weld pool. This is the region in which the metal changes from molten to solid. It is modeled as a local maximum in Figure 3.12. The last region is the virgin base metal which corresponds to the background of the image. It usually has a light level approximately equal to that of the electrode shadow.

Rao's algorithm was straight-forward, and based on his model of the ideal light intensity distribution of a video line across the electrode. He first smoothed and magnified the data by summing the value of each pixel in the video line with the three consecutive data points on either side, and assigned the results to the smoothed pixel value. Since he assumed 7 bit data, he limited the result to hexadecimal 7F. Next, he median filtered the smoothed data using a 1x3 window. He assumed that the center pixel of the video line was positioned in Zone A. He also assumed that after magnifying the data with the summing operation, the peak value of pixels in Zone B on either side of Zone A would be equal
to the limit 7F. To calculate the weld pool edge, he searched to the left of the center pixel until he reached a pixel with the value 7F. This marked the beginning of Zone B. The search continued to the left until the data changed from steadily decreasing until at least 2 pixels that increased in value. This was considered the left pool edge. A similar search was used to find the right pool edge.

Rao's processing technique worked as long as the light distribution resembled his model. Unfortunately, it proved to be limited to a narrow range of intensity distribution heights and failed entirely when the intensity distribution was asymmetric, such as might be caused by arc blow (the arc is "blown" to one side of the electrode).

The second investigator was Gutow [8]. Gutow also processed a single video line to detect weld pool edges. He went further and actually used his results for closed-loop control of weld pool width. Using the robot and vision system, he processed the single line at the electrode and calculated a pool width. He compared the width to a set point and commanded the robot to increase or decrease arc current to control the weld pool width.

Gutow realized the shortcomings of Rao's approach. He developed a set of models of the intensity distribution of a video line taken across the electrode (Figure 3.13). Gutow then developed an algorithm that would handle as many of the models as
Figure 3.13 Several models of the intensity distribution of a video line taken across the electrode [8].
possible. He claimed success for all of the intensity
distributions except that from a concave weld pool (Figure 3.13
e).

Gutow's algorithm is based on the first derivative of the
single video line's data. He first median filtered the data with
a 3x5 window (Three video lines are gathered and only 1 remains
after this step. Thus the following algorithms were still
one-dimensional.) He then took the first-order central difference
of the resultant data. He assumed that the derivative would have
the form of the model in Figure 3.14. To find the weld pool's
edges, Gutow first identified the location of the right side of
the electrode by locating the maximum positive number of the
derivative data. He then searched right until the second zero
crossing and called that point the right weld edge. The left pool
edge was found in a similar way.

Gutow's procedure worked better than Rao's but still lacked
reliability. From his work he concluded that reliability could be
increased if the entire image was processed using two-dimensional
techniques.

A third researcher, Anderson [31], worked concurrently with
Gutow but investigated joint tracking. Anderson selected a single
video line just in front of the weld pool to process for a joint
feature. He wanted to detect the middle and the width of the
joint. He used his results for closed-loop control of the robot's
Figure 3.14 Ideal weld pool intensity model and its derivative [8].
position.

Anderson's algorithm was based on a second derivative model of the joint feature (Figure 3.15). After smoothing by averaging with a $1 \times 4$ window, he calculated the first-order central second difference of the video data. He then smoothed the derivative data with the $1 \times 4$ averaging window. The location of the maximum positive number was assumed to be the middle of the joint. The locations of the maximum negative number, on each side of the assumed middle of the joint, were selected as the joint's edges. A new middle was calculated from the two edge locations. The new middle value was used to guide the robot along the joint.

Anderson's algorithm worked only if the video line selected was located just in front of the weld pool, and the weldment had a finished (no scratches or marks) surface. He concluded that reliability could be improved with the addition of structured light and/or two-dimensional processing.

3.5 Conclusion

GTA welding robots need closed-loop control of position (for guidance) and weld pool geometry (for penetration and reinforcement control) if consistent good quality welds are to be made. Machine vision promises to offer this ability. The integrated torch viewing system developed by Richardson allows machine vision to be added without being directional or intrusive.
Figure 3.15 Second derivative model of the joint feature [31].
Several investigators have had limited success using the coaxial viewer. Their work has lead to a understanding of edge features in the weld pool images. It has been limited to one-dimensional analysis to extract weld pool width at the electrode and the location of the joint just in front of the weld pool. It was anticipated that two-dimensional processing would increase the reliability of detecting the joint feature and offer additional, and more reliable, information about the weld pool.

The problem is to use the knowledge gained through one-dimensional processing for the development of two-dimensional processing algorithms. The algorithms should locate the entire weld pool boundary and the electrode.
CHAPTER 4
MEASUREMENT OF TWO-DIMENSIONAL WELD POOL GEOMETRY

4.1 Introduction

There is a great deal of useful information in a two-dimensional image of a weld pool viewed with the coaxial viewer described in Chapter 3. This includes geometric variables such as the maximum pool width, pool area, pool symmetry, pool length, and pool leading and trailing edge shapes. It is known that such information provides considerable knowledge about the ultimate weld properties [32-34]. Thus, it is desirable to extract and use this information in closed-loop control of the weld pool geometry. To control weld pool geometry, welding inputs such as arc heat, arc length, rate of arc travel and position of the arc are manipulated. The objective of the work in this chapter was to develop a method for measuring the two-dimensional weld pool geometry given a coaxial weld pool image.

The development of weld pool geometry measurement algorithms took Chapters 2 and 3 as background and proceeded in two steps.
The first step was a characterization of the weld pool image. This work was accomplished off-line on a PDP-11/34A minicomputer. It involved studying two different representations of several weld pool images and identifying a priori knowledge that could be used in analyzing the images. This work is detailed in Section 4.2.

The second step was also done off-line on the PDP-11/34A. This was a development of machine vision algorithms, in the Fortran 4+, that reduced the image to a representation in such a form that weld pool geometry variables could be easily extracted. The representation was a one-dimensional array describing the weld pool edge around the entire periphery of the weld pool. This work is discussed in Sections 4.3, 4.4, and 4.5.

Once the boundary has been identified, weld pool geometric variables may be calculated. Methods for extracting these variables from the boundary representation are given in Section 4.6.

4.2 The Weld Pool Image

The image of a GTA weld pool as seen by the coaxial viewer is shown in Figure 4.1. The weld is a full-penetration, bead-on-plate weld in 17-4 PH stainless steel. In the unique image, the electrode appears as a spot near the center of the weld pool. The electrode spot or shadow is surrounded by a halo of arc light. The weld pool edge can be distinguished by the transition
Figure 4.1 Photograph of a GTA weld pool as seen by the coaxial viewer [7].
between the dark periphery of the molten pool and the brighter reflection from the surrounding base metal. The very distinct appearance of the weld pool and electrode edges is a result of the viewing method and the convex shape of the molten pool (Figure 4.2) [35].

The vision system could acquire and store images of coaxially viewed weld pools. Several typical images, like the one in Figure 4.7, were obtained and stored on the FDP-11/34A. They were arranged as a 64x63 array of integer values. The integer values range from 0 to 255 corresponding to the 8-bit gray level light intensity of individual camera picture elements. Low light levels correspond to low values and bright levels correspond to high values.

Two representations of the images were studied to determine the characteristics of typical weld pool images. One was the digitized image as output to a graphics dot-matrix printer. This representation, called the gray-level representation, is shown for a typical weld pool in Figure 4.3. The image of Figure 4.3 expresses only 64 levels of gray scale and thus detail is limited. To increase detail in the image, histogram equalization was performed (Section 2.3). The same image of Figure 4.3 after equalization is shown in Figure 4.4. All gray-level representations given in this thesis except Figure 4.3 will be histogram equalized before outputting to the printer.
Figure 4.2 Reflection of light from the GTA weld pool viewing area [35].
Figure 4.3 Digitized image of a GTA weld pool as output to a graphics dot-matrix printer.
Figure 4.4  Image of Figure 4.3 after histogram equalization.
The second representation studied was a light intensity contour plot. The contour plot of the same GTA weld pool as in Figure 4.4, with the x and y-axes representing pixel location and the z-axis representing intensity, is shown in Figure 4.5. The contour can be visualized as a volcanic cone surrounded by foothill like features. The electrode shadow, central to the scene in Figures 4.1 and 4.4, appears in Figure 4.5 as a crater-like feature. The bright arc halo surrounding the electrode, and diminishing to the pool edge, forms the smooth cone. Light reflected from the periphery of the shiny weld pool is reflected away from the acceptance angle of the viewing optics and thus appears darker (Figure 4.2). The bright ring on the outside of the weld pool in Figure 4.4 corresponds to the foothills surrounding the cone in Figure 4.5. This feature is caused by the diffuse reflection of arc light off the solid base material surrounding the weld pool.

Images such as the one of Figures 4.4 and 4.5 represented the best case conditions. These were symmetrical weld pools where the electrode shadow blocked the brightest part of the arc. These images are characterized by the electrode shadow being centered in the weld pool in gray-level representations, and the crater being in the middle of the volcanic cone in light intensity contour plots. Worst case conditions of weld pool images were produced by non-symmetrical weld pools. Non-symmetrical weld pools are caused
Figure 4.5 Light intensity contour plot of the GTA weld pool image of Figure 4.3.
by conditions such as arc blow. Figure 4.6 is a gray-level representation, and Figure 4.7 is a light intensity contour plot of a non-symmetrical weld pool. Note that the electrode shadow lies to the lower right of the bright arc in Figure 4.6. The non-symmetrical nature of the weld pool is clearly seen in Figure 4.7 where the volcanic crater lies on the side of the volcanic cone. Images studied were between the two extreme cases of Figures 4.4 and 4.6.

Two features of importance were extracted from weld pool images to calculate pool geometry. The first was the electrode feature. This corresponded to an area of low light surrounded by very bright light, a very distinguishable feature. It can be seen in Figures 4.5 and 4.7 that the electrode outer edge corresponds to the area of maximum change in the intensity function. It should also be noted that most electrode edge points were coincident with the high intensity pixels forming the top of the volcanic feature.

The second feature to be extracted was the periphery of the weld pool. The weld pool periphery in Figures 4.4 and 4.6 corresponds to a relatively dark ring surrounded by a slightly brighter ring. The brighter ring has peak intensities almost coincident with the minimum intensity representation of the pool edge. In Figures 4.5 and 4.7, the weld pool edge is the local minimum between the volcanic cone and the foothill features.
Figure 4.6  Gray-level representation of a non-symmetrical GTA weld pool image.
Figure 4.7 Contour plot of the non-symmetrical weld pool image of Figure 4.6.
Several characteristics which were common to almost all weld pool images are summarized in the following. The weld pool periphery formed a closed boundary about the electrode. Furthermore, this boundary was somewhat elliptical with no sharp discontinuities. The pool edges were local minima in the image intensity function and were almost coincident with the local maxima formed by the reflection of light off the base material. These local maxima had lower intensity than the local maxima formed by the bright arc light. The arc light formed a volcanic feature whose outside edges were concave upward. The edge between the arc light and electrode shadow formed the highest gradient in the image. The electrode shadow light level and background light level were approximately equal. A summary of these characteristics is given in Figure 4.8.

4.3 Early Processing

Early processing refers to processing that was completed, or is usually completed, in the preprocessing stage of machine vision.

The first processing step performed on the image was analog filtering. The bright arc light was too intense for the CID camera so an analog filter was necessary before sampling. A neutral density filter plate that attenuated the image to 1/8 its original light level was adequate. The camera also has an iris
Figure 4.8 Summary of common characteristics of coaxial weld pool images.
that can be used to control image light level. For work in this thesis the neutral density filter was used in combination with a variety of iris settings. The iris settings were selected so the camera was not saturated (light level too high) and the weld pool was clearly visible.

No preprocessors were used so all early digital processing was completed by the minicomputer. This processing consisted of a smoothing operation followed by a histogram calculation. The smoothing operation used was the 3x3 averaging window discussed earlier (Section 2.3). This straight-forward and efficient spatial domain technique utilized a 3x3 neighborhood about each image point. Given the original 64x63 image represented as $f'(x,y)$, the procedure was to generate a smoothed image $h'(x,y)$ using the relation:

$$h'(x,y)=\frac{1}{9} \sum_{S} f'(n,m) \tag{4.1}$$

The variables $x$ and $y$ are pixel coordinates where $x=2,3,...,63$ and $y=2,3,...,62$. The operation is not performed on the outermost rows and columns ($x=1,64$ and $y=1,63$). $S$ is the set of coordinates of points in the 3x3 neighborhood of the point $(x,y)$. A gray-level representation of an image before and after smoothing is shown in Figure 4.9, and contour plots of the same image are shown in Figure 4.10. The smoothing operation effectively reduced the random effects introduced by the sensor and transmission channel.

During the smoothing operation a histogram of the raw image
Figure 4.9 Gray-level representation of a weld pool image (a) before and (b) after smoothing.
Figure 4.10 Contour plots of the images shown in Figure 4.9.
was generated. The histogram of the raw image of Figure 4.10 is given in Figure 4.11. Figure 4.11 is typical of weld pool image histograms. The larger of the two approximately normal distributions corresponds to the background, weld pool, and electrode. The smaller distribution to the right in Figure 4.11 corresponds to the welding arc. Important features of the histogram are summarized in Figure 4.11.

4.4 Edge Detection

Once the image has been smoothed, the next step is to locate points that lie on the electrode and weld pool edges. This step, known as edge detection, transforms the image into an intermediate representation that is more easily processed to locate boundaries than the gray-level representation of the image.

To detect edges in weld pool images, an edge operator like those discussed in Section 2.4 was used. The problem was to choose the correct operator so that both, electrode and weld pool, edges were found but no other edges. The choice of the operator was based on three characteristics of the weld pool image. First, the electrode-arc light boundary formed a local maxima, that is, the electrode edge was coincident with the local maxima in the image intensity function. Second, the weld pool-base material boundary formed local minima that were almost coincident with the local maxima formed by the reflection of light off the base.
Figure 4.11 Histogram of the raw image of Figure 4.9.
material. Third, the distribution of light from the electrode to weld pool edge was concave upward. As a result of the third characteristic, the area of the image between the electrode and weld pool edge could have zero or positive second derivative but not negative. Negative second derivatives correspond to local maxima in the image intensity function. Because of these three characteristics, the objective of the edge detection algorithm was to locate local maxima in the intensity distribution of the image. Local maxima in the image occurred only at the electrode edge and just outside the weld pool edge.

No edge operators that locate local maxima appear in the literature. Therefore, a special operator was developed. Since a negative value of the second derivative of a function corresponds to a point that is a local maximum, the edge operator used was related to the second derivative. Due to the weld pool being defined by a closed boundary, local edges were oriented in all directions. The edge operator was developed to be sensitive in two dimensions. This was accomplished by using two one-dimensional operators. One operator was applied to rows of the image array, and the other to columns.

The edge operator developed to locate vertical edges is shown in Figure 4.12 and is defined by the relation:

\[ g_R(x,y) = \frac{1}{4} h'(x+2,y) - \frac{1}{2} h'(x,y) + \frac{1}{4} h'(x-2,y). \]  

(4-2)

The value of \( g_R(x,y) \) is negative whenever \( h'(x,y) \) is a relative
| $\frac{1}{4}$ | 0 | $-\frac{1}{2}$ | 0 | $\frac{1}{4}$ |

**Figure 4.12** Edge operator developed to locate vertical edges in GTA weld pool images.
maximum. This edge operator operates on the rows of the image, thus the subscript R. Furthermore, $g_R(x,y)$ is suitable for finding approximate local vertical edges but is insensitive to horizontal local edges.

The second edge operator, $g_C(x,y)$, is similar to $g_R(x,y)$ except that it operates on the columns of the image. It is defined by the relation:

$$g_C(x,y) = \frac{1}{4} h'(x,y+2) - \frac{1}{2} h'(x,y) + \frac{1}{4} h'(x,y-2).$$  \tag{4-3}

The two local edge operators were applied at every image point, $(x,y)$, in the 64x63 array of points where $x=3,4,\ldots,62$ and $y=3,4,\ldots,61$. Again, the outer periphery of pixels were neglected ($x=1,2,63,64$ and $y=1,2,62,63$). Note that the value of $g_R(x,y)$ is negative if $(x,y)$ is a vertical local edge point, and $g_C(x,y)$ is negative if $(x,y)$ is a horizontal local edge point, where local edge point refers to a local maximum point.

Since the edge operators were related to the second derivative, any noise left in the image, $h'(x,y)$, was doubly enhanced in $g_R(x,y)$ and $g_C(x,y)$. To eliminate many of the false edge points the following approach was used. The array $g_R(x,y)$ was filtered with a 3x3 median filter. Then the array $g_C(x,y)$ was filtered using the same filter. After filtering, the arrays were combined to form a binary image $g(x,y)$. The image point located at $(x,y)$ in $g(x,y)$ was set equal to 1 wherever the point $(x,y)$ in the filtered arrays, $g'_R(x,y)$ or $g'_C(x,y)$, was less than zero, and
was set to 0 in $g(x,y)$ whenever it was equal to or greater than zero in $g^1_R(x,y)$ or $g^1_C(x,y)$.

Figure 4.13 gives an image of a weld pool and the corresponding binary image formed by the previous procedure. The black pixels correspond to points where the image function $g(x,y)$ equals one, that is, indicate local edge points. The electrode boundary can be seen inside the weld pool boundary.

4.5 Boundary Detection

Boundary detection is the process of composing the many individual local edge elements (corresponding to a 1 in the binary image $g(x,y)$) into a coherent one-dimensional (boundary) feature. The boundaries of interest in $g(x,y)$ were the outer electrode edge and the weld pool edge.

The first step in boundary detection is to use apriori knowledge to decide on a suitable representation for the boundary that is being detected. The electrode boundary was circular, therefore it could be completely described by a center point and a radius. That is, three integer values were needed to represent the electrode to the computer: the electrode radius in pixels, the $x$ position of the electrode center, and the $y$ position of the electrode center. The electrode radius seldom changes and was known, thus only the electrode position in the image needed to be identified.
Figure 4.13  a) Gray-level representation of weld pool.
b) Resultant binary image after edge
detection.
To calculate the electrode position, the local edge points of \( g(x,y) \) were sorted with the aid of the image function \( h'(x,y) \) into local weld pool edge points, \( g_p(x,y) \), and local electrode edge points, \( g_E(x,y) \). This was accomplished by comparing the relative intensities of all pixels in \( h'(x,y) \) at \((x,y)\) wherever \( g(x,y) = 1 \). Since electrode edge points had relatively large intensities in the image \( h'(x,y) \) as compared to weld pool edge points, a threshold \( T1 \) was defined that divided the two sets of local edge points. Thus, the binary image \( g(x,y) \) was decomposed into two disjoint binary images, \( g_p(x,y) \) and \( g_E(x,y) \), i.e.

\[
\begin{align*}
g(x,y) &= g_p(x,y) \cup g_E(x,y) \quad (4-4) \\
g_p(x,y) \cap g_E(x,y) &= \phi \quad (4-5)
\end{align*}
\]

where

\[
g_E(x,y) = 1 \text{ iff } h'(x,y) \geq T1 \text{ and } g(x,y) = 1 \quad (4-6)
\]

and

\[
g_p(x,y) = 1 \text{ iff } h'(x,y) < T1 \text{ and } g(x,y) = 1 \quad (4-7)
\]

The threshold \( T1 \) was selected based on the histogram of the image \( h'(x,y) \). Figure 4.14 illustrates the decomposition of Figure 4.13 into \( g_p(x,y) \) and \( g_E(x,y) \). Note that this method successfully sorted most of the edge points.

The electrode position was calculated directly from \( g_E(x,y) \). The algorithm used was to average all of the \( x \) and \( y \) positions of
Figure 4.14 The decomposition of Figure 4.13b into the weld pool and electrode edge points.
points \((x,y)\) whenever \(g_E(x,y)\) was equal to 1. Hence, the electrode position point \((\bar{x}, \bar{y})\) was found as:

\[
\bar{x} = \frac{1}{n} \sum_{S} x \\
\bar{y} = \frac{1}{n} \sum_{S} y
\]  

(4.8)  

(4.9)

where \(S\) is the set of all \((x,y)\) such that \(g_E(x,y) = 1\) and \(n\) is the number of such points.

Once the electrode point \((\bar{x}, \bar{y})\) had been located, the weld pool boundary may be found. Two properties of the weld pool boundary were used to derive a suitable representation of the weld pool. First, the weld pool edge formed a closed boundary about the electrode position \((\bar{x}, \bar{y})\). Second, the weld pool boundary was continuous and slowly varying. The representation of the boundary was chosen to be an one-dimensional array \(r(n\theta_0)\) with \(n=0,1,\ldots,63\) and \(\theta_0 = 5.625^\circ\). Each of the 64 values for \(r_n(n\theta_0)\) represents a radial distance from the electrode to the boundary at spacings of 5.625\(^\circ\) (Figure 4.15). The image \(g_p(x,y)\) was processed to find these 64 values. The coordinates \((x,y)\) of each point in \(g(x,y)\) with a value of 1 were transformed from cartesian coordinates, \((x,y)\), to polar coordinates, \((r,\theta)\), with the electrode considered the origin of the coordinate system. Thus, for every local weld pool edge point, \((x,y)\), a radial distance, \(r\), and an angle, \(\theta\), was computed as

\[
r = \left[ (x-\bar{x})^2 + (y-\bar{y})^2 \right]^{1/2}
\]

(4-10)

\[
\theta = \tan^{-1}\left[ \frac{(y-\bar{y})}{(x-\bar{x})} \right]
\]

(4-11)
Figure 4.15 Details of the radial representation of the weld pool boundary.
where \( r \) is in pixels and \( \theta \) in degrees.

Since \( r \) for any edge point must be greater than the electrode radius, \( r_m \), the point was disregarded if \( r \) was less than \( r_m \). The local edge points were sorted into groups according to their associated angle, \( \theta \), and median filtered to give the 64 discrete values for \( r(n\theta_0) \). For instance, the value of \( r_0(0\theta_0) \) was the median radius of all edge points whose angle \( \theta \) was from \(-1/2 \theta_0\) to \(+1/2 \theta_0\), similarly \( r_1(1\theta_0) \) was the median \( r \) of all edge points whose angle \( \theta \) was from \(1/2 \theta_0\) to \(1-1/2 \theta_0\). This process was continued for \( n=2,3,\ldots,63 \), resulting in 64 values for \( r_n(n\theta) \).

The weld pool boundary of Figure 4.13 was located using this procedure and representation and is shown in Figure 4.16. Figure 4.17a is the same boundary plotted as \( r(n\theta_0) \) versus \( n \). Note that there are several discontinuities caused by gaps in \( r_n(n\theta_0) \) (\( n=14, 17, 25-27, 39, 48-52 \)) and areas of high slope (\( n=35, 58-61 \)).

The second property of the weld pool boundary required that no discontinuities or areas of high slope exist in the \( r(n\theta_0) \) versus \( n \) plot. Therefore, a second representation was needed to account for this as well as for the closed boundary property. To derive the second representation from the first, two steps were taken. First, zero values were replaced by values interpolated from adjoining non-zero locations. Second, the 64 points were filtered and compressed. Since, the 64 points were periodic in \( \theta \), e.g. \( \theta = \theta + n\pi \), where \( \pi=2\pi \) and \( n \) is any integer, frequency domain
Figure 4.16 Resultant radial representation of the weld pool boundary for the image of Figure 4.13.
Figure 4.17  a) Plot of the boundary of Figure 4.16 as the signal \( r(\theta_0) \).

b) Result of interpolation and low pass filtering the signal of Figure 4.17a
techniques were used. An ideal low pass filter as described in [36] was used with a frequency cutoff at the third harmonic. Briefly the filter worked as follows:

1. The signal, \( r(n\theta) \), was transformed into the frequency domain using a Fast Fourier Transform (FFT).

2. All terms of the Fourier transform of order four or greater were set equal to zero.

The result was a compressed representation of a smooth, continuous, weld pool boundary. The representation, which was equivalent to the first 4 terms of the Fourier expansion of \( r(n\theta_0) \), is called the Fourier descriptor of the weld pool. The descriptor is a total of 7 real numbers \( F_i \), \( i = -3 \) to 3. This technique of describing and compressing the weld pool boundary is known as the angle invariant compression of closed boundaries.

The Fourier descriptor is ideal for storage of multiple pool boundaries because of the small amount of memory required. However, it is difficult to visualize and extract some aspects of weld pool geometry from this representation. Therefore, an inverse FFT was applied to the Fourier descriptor to return the low pass filtered signal, \( r'(n\theta_0) \), to the spacial domain. The result of interpolating zero spots and filtering the signal of Figure 4.17a and inverse transforming back into the spacial domain is shown in Figure 4.17b. Notice that the signal, \( r'(n\theta_0) \), is continuous and slowly varying, as required. A plot of \( r'(n\theta_0) \)
corresponding to radial distance from the the electrode, is given in Figure 4.18, which demonstrates the results of locating the electrode and weld pool boundaries.

4.6 Extraction of Weld Pool Variables

With the boundary of the weld pool specified as \( r'(n\theta_0) \), the result of filtering \( r(n\theta_0) \), the desired weld pool variables may be extracted. Figure 4.19 indicates the relationships necessary to calculate the weld pool width at the electrode, \( w_e \), and the maximum weld pool width, \( w_m \). As illustrated in Figure 4.19, \( w_e \) is found using the relationship

\[
w_e = r'_0(0\theta_0) + r'_3(32\theta_0) .
\]

To find \( w_m \), trigonometric relationships are used. First, \( w_R \), the maximum width of the right side of the weld pool, is found by calculating

\[
w_n = r'(n\theta_0)\cos(n\theta_0)
\]

for \( n = 8, 49, \ldots, 63, 0, 1, \ldots, 15 \) and setting \( w \) equal to the maximum \( w_n \). Similarly, \( w_L \), the maximum width of the left side of the pool, is found by calculating

\[
w_n = -r'(n\theta_0)\cos(n\theta_0)
\]

for \( n = 16, 17, \ldots, 47 \) and setting \( w \) equal to the maximum \( w_n \).

Finally, \( w_m \) is calculated by summing \( w_L \) and \( w_R \), e.g.,

\[
w_m = w_R + w_L .
\]

Given \( w_R \), \( w_L \), and the electrode position \((\bar{x}, \bar{y})\), the location
Figure 4.18 The final result of processing the smoothed image of Figure 4.13 to locate the electrode point and the weld pool boundary.
Figure 4.19  Extraction of the maximum weld pool width and weld pool width at the electrode from the radial representation of the weld pool boundary.
of the weld pool is completely specified in the dimension orthogonal to the direction of travel. This location information is useful in eventual joint tracking where maintaining the weld pool centered on a weld seam feature is important.

The approximate area of the weld pool may also be calculated. The strategy used was to fit a number of triangles into the weld pool area as in Figure 4.20. Summing the individual areas \( A_k \), \( k=0,1,...,31 \), gives an approximation of the weld pool area. Using 32 triangles, each formed as in Figure 4.20, the equation to calculate the approximate area of the weld pool, \( A_{WP} \), is

\[
A_{WP} = \sin \theta_o \sum_{k=0}^{31} (r'(2k\theta_o))^2
\]

(4-16)

where \( \theta_o \) is 5.625 degrees.

4.7 Conclusions

A method of computer analysis of the two-dimensional image of gas-tungsten-arc (GTA) weld pools as acquired through a coaxial viewing system was presented. The method calls for the following inputs:

(1) A two-dimensional image of the weld pool.

(2) A threshold value, \( T_1 \), for dividing local edge points into weld pool and electrode edge points.

(3) The electrode width, \( r_m \).

The procedure produces the following outputs:
Figure 4.20 Extraction of an approximation to the area of the weld pool from the radial representation.
(1) An electrode position \((\bar{x}, \bar{y})\).

(2) The boundary of the weld pool represented as a Fourier descriptor, \(F_i(i=-3 \text{ to } 3)\), comprised of seven real numbers, and a one-dimensional array, \(r'(n\theta_o)\), of 64 real numbers.

The technique was based on locating local maxima in the intensity distribution function with an edge operator, and then using an angle invariant compression technique to produce a Fourier descriptor \(F_i\), \(i=-3 \text{ to } 3\). An inverse FFT applied to the Fourier descriptor produced \(r'(n\theta_o)\). Weld pool geometric variables were then calculated from \(r'(n\theta_o)\).

The algorithm was comprised of a series of transformations. A brief summary is given in Figure 4.21.

Methods for extracting some weld pool geometrical variables were also described. The procedure described in Figure 4.21 was applied to the analysis of a general sample of weld pool images. It was felt that the sample represented worst, best, and typical cases. The algorithm was fairly robust. When it did fail it was generally do to improper separation of electrode and weld pool edge points. The analysis was implemented off-line, in Fortran 4+, on a PDP-11/34A minicomputer.
Figure 4.21 Summary of processing algorithms.
CHAPTER 5
TWO-DIMENSIONAL WELD JOINT DETECTION

5.1 Introduction
The torch position relative to the weld joint is the most important control variable in robotic arc welding. Ideally, the weld pool is centered on the joint for the entire length of the weld. This joint tracking ability is best accomplished if the location of the joint is sensed just ahead of the point of welding in real-time. The objective of the work described in this chapter was to develop a vision algorithm that locates the joint feature in coaxial weld joint images. The algorithm developed operated on the two-dimensional joint image to identify the location and width of the joint. The results of the detection were then displayed on the monitor. The technique was implemented in real-time on the Aim-65 through use of 6502 assembly language.
5.2 Coaxial Weld Joint Image

A gray-level representation of a typical coaxial weld joint image is shown in Figure 5.1. The image is represented as a 16x64 array of 256 gray-level pixels. The direction of weld travel is up in the image. The electrode and weld pool are below the image, out of the field of view. The joint feature corresponds to the dark, vertical stripe in the approximate middle of the image.

The joint feature was assumed to be a straight, approximately vertical, line in all coaxial joint images. The above assumption is valid even for curved joints since the coaxial image contains only a 1/4 inch section of the joint immediately in front of the pool. By orienting the image such that the direction of travel is upwards, the joint appears approximately vertical in the image. This can be accomplished through rotation of the image before sampling. Most robots have a wrist roll axis that will accomplish the rotation. Alternatively, a dove prism can be used to rotate the image before sampling.

The same image as in Figure 5.1 is shown in Figure 5.2 as an intensity contour plot. A plot of a single row of the image of Figures 5.1 and 5.2 is shown in Figure 5.3. Figure 5.4 is the same row after a 3x3 averaging window was used on the image. The averaging window effectively smooths the weld joint image, thus no other early processing was needed.

Figure 5.4 illustrates the characteristics of a joint feature
Figure 5.1 Gray-level representation of a typical coaxial weld joint image.
TABLE 5.2: Interneuron contour plot of the tongue
Figure 5.3 Plot of row 3 of the image of Figure 5.1.
Intensity

Figure 5.4  Plot of row 3 of the image of Figure 5.1 after smoothing with a 3x3 averaging window.
in the coaxial image. The joint feature corresponds to the largest local minimum in the figure. Furthermore, the edges of the joint are coincident with the local maxima on each side of the local minimum.

The light intensity in images of the joint decreased as distance increased from the front of the weld pool (the bottom of the image). This is illustrated in Figure 5.5. The effect on individual rows in the image was in amplitude and not shape. The closer to the welding arc (the light source) the higher the light intensity amplitude. At distances several pixels (20-30) ahead of the weld pool, the amplitude decreased to a point that the edge feature was no longer recognizable.

From the above, three important points concerning coaxial weld joint images were deduced. First, the joint feature corresponded to the largest local minimum in any row of the image. Second, the edges of the joint were coincident with the local maxima on each side of the local minimum. Third, the image gathering window should be located over an area just in front of the weld pool to maximize joint feature contrast.

5.3 Edge Detection

Edges corresponding to the approximate middle, and the sides, of the joint were detected. Each row of the smoothed image, \( h'(x,y) \), was processed for these edges separately. The edge
Figure 5.5 Plots of several rows from a typical weld joint image [31].
detection was based on heuristic algorithms in combination with an edge operator.

The joint feature corresponded to a local minimum (joint middle) and two local maxima (joint sides) in the rows of the image function \( h'(x, y) \). To find these local extrema, an edge operator discussed earlier was used. The image, \( h'(x, y) \), was transformed into \( g_R(x, y) \) by the relation

\[
g_R(x, y) = \frac{1}{4} h'(x+2, y) - \frac{1}{2} h'(x, y) + \frac{1}{4} h'(x-2, y). \quad (5-1)
\]

Negative values of \( g_R(x, y) \) corresponded to local maxima and positive values corresponded to local minima.

From \( g_R(x, y) \) three binary images were produced. These were \( b_j(x, y) \) corresponding to edges related to the middle of the weld joint, \( b_R(x, y) \) corresponding to the right edges of the weld joint, and \( b_L(x, y) \) corresponding to the left edges of the joint. The first, \( b_j(x, y) \), was produced by observing each row of \( g_R(x, y) \) separately. The point, \((x, y)\), associated with the highest positive value of each row of \( g_R(x, y) \) was set equal to 1 in \( b_j(x, y) \); all other points in the row were set equal to zero. Therefore, only one point in each row equaled 1, and that point indicated the position of the approximate middle of the joint in that row.

From the middle position of the joint, the same row of \( g_R(x, y) \) was searched to the left for the first occurrence of a local maximum. That point was set equal to 1 in \( b_L(x, y) \).
Similarly a search to the right for a local maximum indicated the point in the row of $b_R(x,y)$ to set equal to 1. Thus, 3 binary images were produced, each having only 1 pixel equal to 1 in any row of the image. Figure 5.6 shows the 3 binary images produced by the preceding method applied to the image of Figure 5.1

5.4 Boundary Detection

The boundaries associated with the middle, right, and left side of the joint were found from the binary images $b_j(x,y)$, $b_R(x,y)$, and $b_L(x,y)$ respectively. Since the joint was assumed to be vertical, the boundaries were completely described by 3 integers. One indicated the position (column number) of the middle of the joint in the image. The other two indicate the positions (column numbers) of the left and right sides of the joint.

To calculate the column number of the middle of the joint, a Hough transform was applied to the binary image $b_j(x,y)$. The transform was used to locate the best vertical line in the image. Vertical lines have an equation of the form $y=k$ (Note that the y axis is horizontal.) The only parameter in this equation is the constant $k$. If there are $n$ columns in the image, $h'(x,y)$, then only $n$ columns exist in $b_j(x,y)$. Therefore, only $n$ different values of $k$ may exist, each corresponding to 1 column in $b_j(x,y)$. This means that only $n$ possible lines could exist in $b_j(x,y)$,
Figure 5.6 Binary images of edge points showing the location of the (a) joint, (b) left side, (c) right side, of the weld joint image of Figure 5.1.
given that the lines are vertical. The Hough transform calculated which of these \( n \) possible lines was best represented in \( b_j(x,y) \).

The first step of the Hough transform was to create a parameter array \( A(k) \), \( k=1 \) to \( n \). The array was initially set to zero. Than the binary image, \( b_j(x,y) \), was observed. At each point, \((x,y)\), in \( b_j(x,y) \), that equalled 1, the corresponding line going through that point had an equation \( k=y \). Thus, the array, \( A(k) \), was incremented at \( k=y \). That is, for every point, \((x,y)\), in \( b_j(x,y) \), equal to 1, \( A(y) \) was set equal to \( A(y)+1 \). Once all points in \( b_j(x,y) \) had been processed, the \( k \) associated with the highest value in \( A(k) \) indicated the position of the middle of the joint. Similarly a Hough transform was used to find the left edge from \( b_L(x,y) \), and the right edge from \( b_R(x,y) \).

5.5 Conclusion

A method for the real time identification of a weld joint from a coaxial weld joint image was presented. The technique was implemented on the vision and robot system. The algorithm was based on an one-dimensional Hough transform. Hough transforms are more efficient than a least square fit. The speed of the algorithm was proportional to the size of the image. Typical image size was 16x64 pixels which took 0.072 seconds. Though the technique was limited to vertical joints, it can be expanded to joints of all orientations by using the two-dimensional Hough.
transform. Several real-time tests were executed, indicating that the technique was fairly robust.
CHAPTER 6
SUMMARY AND CONCLUSIONS

6.1 Research Contributions

The objective of this thesis was to develop a method of analysis of two-dimensional coaxial GTA weld images. The weld pool boundary, the electrode feature, and the joint boundary were identified. The weld pool and electrode analysis was implemented off-line and the joint analysis was implemented in real-time.

Several weld pool images were analyzed. First, the edges of the weld pool and electrode were located using an edge operator. The edge points where separated into weld pool edge points and electrode edge points by a thresholding operation. Using an angle invariant compression technique, the weld pool boundary was found from the weld pool edge points. The boundary was represented as a Fourier descriptor and a one-dimensional array. Several methods for extracting weld pool geometry variables from the boundary representation were given. The algorithms were tested on a general sample of coaxial images with high success.
Weld joint images were also analyzed. Edges corresponding to the middle and sides of the joint were located with an edge operator. A Hough transform was used to locate the joint boundary. Results were demonstrated in real-time under several different welding conditions.

The research demonstrated that weld pool geometry could be measured from two-dimensional coaxial weld pool images. It also resulted in a real-time joint detection routine.

6.2 Research Extensions

The next major step to be taken is a real-time implementation of the algorithm that locates the weld pool boundary. This would include the building of preprocessors and any special purpose hardware needed for the implementation. The real-time algorithm could then be used to test the generality and reliability of the method. It could also be combined with the joint detection routine to provide the robot with pool centered joint tracking. A quasi real-time algorithm has been implemented but was not discussed in this thesis.

The present algorithm does not integrate its results from image to image. Much work could be done in using the results of the previous image analysis to help in the analysis of the present image. This could greatly improve the efficiency and reliability of the algorithm.
Joint tracking using the two-dimensional Hough transform for locating lines in an image should be investigated. The y-intercept gives an indication of the cross-seam error of the torch position. The slope information may be used to rotate the image so the joint appears approximately vertical in the image.

Another area that needs researched is joint tracking in three dimensions. Structured light would need to be integrated into the present coaxial viewing torch to provide this capability.

Hopefully, research in machine vision for GTA welding robots will continue. Machine vision offers solutions to most GTA welding control problems. It is felt that this thesis represents a valuable step towards a robust, vision based GTA welding robot controller.
REFERENCES

1 Mack Corporation, Robotics... Start Simple, Internal Pub., 1982.

2 Knight-Rider news service, "Robots' Lack of vision prevents their greater use", Columbus Dispatch, Dec. 18, 1983.


4 Rosenfeld, Dr. Azriel, Lecture notes, University of Maryland's computer vision laboratory, 1983.


21 Martelli, A., "Edge detection using heuristic search methods", CGIP 1,2 , August, 1972, 169-182.


