Design Project Report

On

Image Processing on Wire-Arc Additive Manufacturing

BY

J Chaitanya Sree Krishna (2021A4PS2273H)

Under the supervision of

Prof. Kundan Kumar

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF ME F376: DESIGN PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

HYDERABAD CAMPUS

Table of Contents

1. ABSTRACT

The purpose of this research is to get a thorough understanding of wire arc additive manufacturing (WAAM), including its concepts, applications, benefits, and challenges.Due to its ability to facilitate real-time monitoring, fault detection, and process optimisation, image processing is essential in tackling issues. Image processing techniques make it easier to identify and mitigate flaws like cracks, voids, and spatter by thoroughly analysing process parameters, melt pool dynamics, and surface morphology. This helps to maintain quality standards and minimise the need for post-processing. Moreover, image processing helps optimise process parameters to maximise efficiency and reduce material waste by examining heat gradients, material flow patterns, and bead geometry. WAAM is moving towards intelligent manufacturing systems that can self-correct and continuously improve by integrating artificial intelligence and machine learning algorithms with image processing to provide predictive maintenance and automated decision-making. To fully realise the potential of WAAM, future research should concentrate on improving image processing methods and incorporating advanced analytics.

2. Introduction to WAAM

In Wire Arc Additive Manufacturing, an electric arc is used to deposit filler wire for manufacturing. Talking about additive manufacturing, minimizing production time and material consumption is the major objective of Additive Manufacturing (AM) procedures. On the other hand, solving performance and structural quality issues is the key difficulty of AM technology.

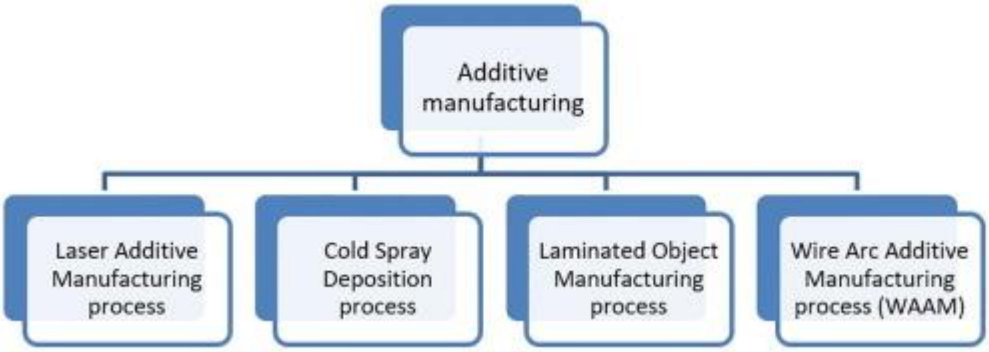


Fig1. Categorization of some important AM processes

(Paper 2)

2.1 Process of WAAM:

- Wire as feedstock
- Electric Arc as heatsource (similar to ones used in welding GMAW,GTAW,PAW)
- Material deposition layer by layer

2.2 Process Parameters:

- Wire feed rate
- Travel speed
- Current and velocity

Benefits:

- Capital cost
- Open architecture
- Part size
- Complex geometries

Cons:

- Deformation and Residual stresses
- Porosity
- Cracks
- Oxidation
- Surface finish

2.3 Image processing for height tracking (Paper1)

- Image analysis - detection

- Removing noise - using gaussian filter
- Top edges detection - Sobel edge detector
- Threshold segmentation - Otsu
- Edge line detection
- Hough Transform : The Hough transform is a technique used in image processing and computer vision to detect simple shapes in images. It works by converting the image space into a parameter space, where the shapes can be detected by identifying patterns. The Hough transform can detect shapes like lines, circles, and ellipses. It can also detect other structures if their parametric equation is known. The Hough transform can provide robust detection under noise and partial occlusion.

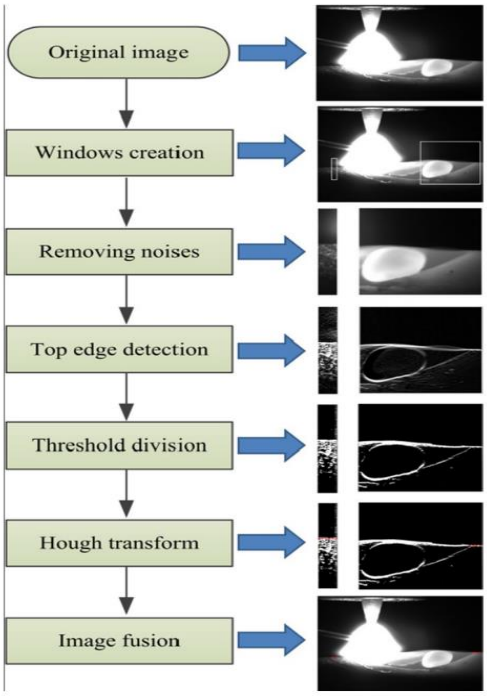


Fig. 4 Flow chart of image processing procedures for deposition height measurements

3. Introduction to Image processing

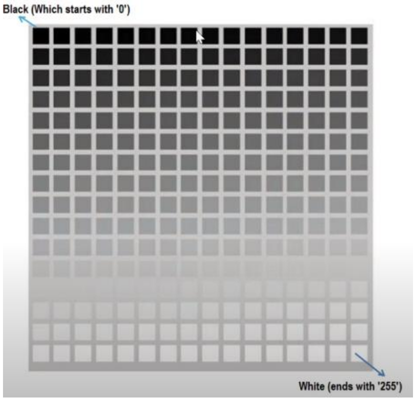
An image is a binary representation of some visual information. In image processing, the image is fed as input and the system interprets and understands the content to take further actions. Image contains an array of multiple pixels. Pixels are the fundamental components of an image.

3.1Types of images:

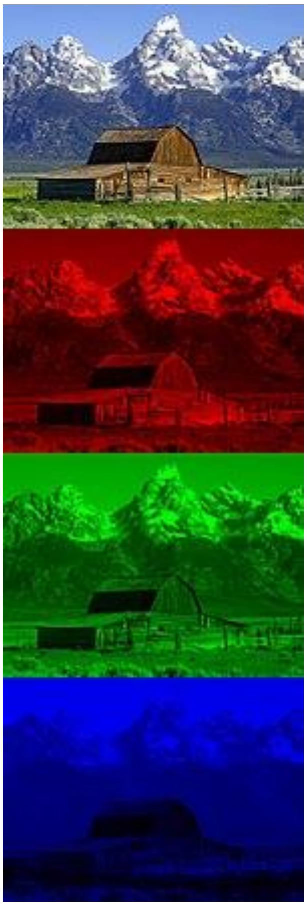
1. Binary images: represented as 0s and 1s. Black is represented by 0 and white is represented as 1s. In a binary image, each pixel needs one bit of storage.

2. Black and white images: the difference between binary images and black and white images is that each pixel in B/W images needs 8 bit of storage instead of 1 bit. These images are smoother and high quality.

3. Grey scale: The range of pixels vary from 0-255, 0 being pure black and 255 being pure white. These images follow 8-bit format.

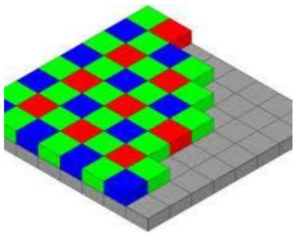
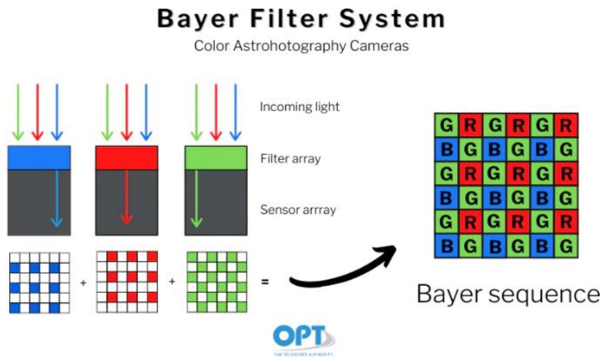


4. <u>Color Image</u>: Each pixel has some color information and can have any of the 3 channels. The 3 channels include, Red (R), Green (G) and Blue (B) . Each channel requires 8 bits of storage, hence each pixel takes 24 bits of storage.



3.2 Image Formation:

1. Light enters the camera lens and falls onto the sensor.
2. The Bayer filter mosaic sits in front of the sensor, and each filter only allows a specific color of light to pass through.
3. The Bayer filter mosaic: A small grid of color filters covering the light-sensitive pixels of a digital camera sensor is called a Bayer filter mosaic, sometimes referred to as a color filter array (CFA). Only a certain color of light may pass through each filter, and a full-color image is subsequently produced using the data that is obtained.

.

4. The light intensity that goes through each filter is recorded by the sensor.
5. The image data that the sensor records is not a full-color image because each pixel can only record one color.
6. The missing color information for each pixel is then interpolated using a technique known as demosaicing, resulting in a full-color image.

Bits Per Pixel: Number of bits of information stored in each pixel of a given image. More the number of bits, more the image is better represented. Hence, more the bits per pixel, better the image quality but also more the storage.

Number of colors = 2BPP

Intensity of an image: An image's intensity is defined as each pixel's brightness or darkness. The basic characteristic of a picture that controls how light or dark it appears to the human eye is this one.

- Each pixel in an image has an intensity that is represented by a number. This value normally spans from 0 (black) to 255 (white) in grayscale images, with intermediate values denoting various shades of grey. Every pixel in a colour image contains several intensity values, one for each colour channel (red, green, and blue, for example).

There are several statistical measures that can be used to characterise an image's overall intensity, including:

- The average intensity of each pixel in the image is termed the mean intensity.
- When all pixel intensities are sorted in either ascending or descending order, the middle value is known as the median intensity.
- A histogram is a visual depiction of the distribution of the image's pixel intensities.

Contrast: The difference in colour or luminance between different areas of an image is referred to as its contrast. It basically controls how distinct and observable details and objects appear in the picture.
Image contrast can be measured in a variety of ways, from straightforward computations to intricate algorithms. Typical techniques include the following:

- Measures the intensity difference between adjacent pixels, called local contrast.
- The general distribution of intensities throughout the entire image is measured by global contrast.
- Determines the contrast ratio by dividing the image's greatest and lowest intensity values by their ratio.

3.3 Types of color models:

1. Red, Green, and Blue (RGB):
   - This is a color model that is widely utilised in digital cameras and electronic displays. It depicts colors as mixtures of the intensities of blue, green, and red.
   - Use: The majority of electronic displays and digital cameras use RGB as their default color model, which is widely used in digital imaging.

2. Cyan, Magenta, Yellow, Key/Black, or CMY (K):
    - In color printing, the subtractive color model known as CMY is employed. Different amounts of light-absorbing cyan, magenta, and yellow ink are subtracted from a white background to create the colors. A black component is included in CMYK for improved color reproduction.
    - Usage: Frequently utilized in color printing operations, including commercial printing equipment and color printers.

3.5 Steps in Image Processing:

1. Image acquisition: It is the first and foremost stage. Utilizing tools like scanners or cameras, get the image. Image is prepared for pre-processing.
2. Image enhancement: To get the picture ready for additional processing, this step entails a number of procedures. Typical pre-processing duties consist of:
    - Removing unwanted noise that was introduced during transmission or capture is known as noise reduction.
    - Format conversion: changing the image's color space or resizing it to make it acceptable for processing.
    - Resolving any distortions or misalignments in the image is known as geometric correction.

3. Image Enhancement: This is one of the most crucial steps. The objective of this stage is to enhance the image's visual quality for enhanced human perception or more machine analysis. Typical methods for improvement consist of:
    - Adjusting the contrast will highlight elements by making the contrast between light and dark areas larger.
    - Brightness adjustment: modifying the image's overall level of lightness or darkness.
    - Color correction: bringing color channels into balance or removing color casts.
    - Sharpening: improving details and edges to produce a picture with more clarity.

4. Image Restoration: The goal of this stage is to restore the original image from any deterioration that might have happened during storage or capture. Restoration methods deal with things like:
    - Blurring: eliminating blur brought on by motion, shakes, or insufficient focus.
    - Removing more intricate noise patterns by going beyond simple pre-processing techniques is known as noise reduction.
    - Eliminating artifacts such as dust accumulations, scratches, and compression artifacts.

5. Color image processing: In this step, the color information(RGB) could be used to extract and understand the features from the image. Multiple color conversion models can be used to facilitate this process.

6. Image Segmentation:
    - Based on specific features, divide the image into sections or segments that have meaning.
    - Techniques including edge detection, clustering, and thresholding are frequently used.

    This can be done by two means; Boundary representation and Regional representation. Boundary focuses on external shapes,i.e, edges,corners,etc, whereas regional focuses on internal properties such as texture, shape, etc.

7. Object Detection: Last and final step which includes
    - Identifying and locating specific objects within the image.
    - Classifying the identified objects into predefined categories.
    - Identifying recurring patterns or shapes within the image.
    - Extracting measurements like size, shape, or distance from the image.

4. Image Processing

4.1 Extraction of Images

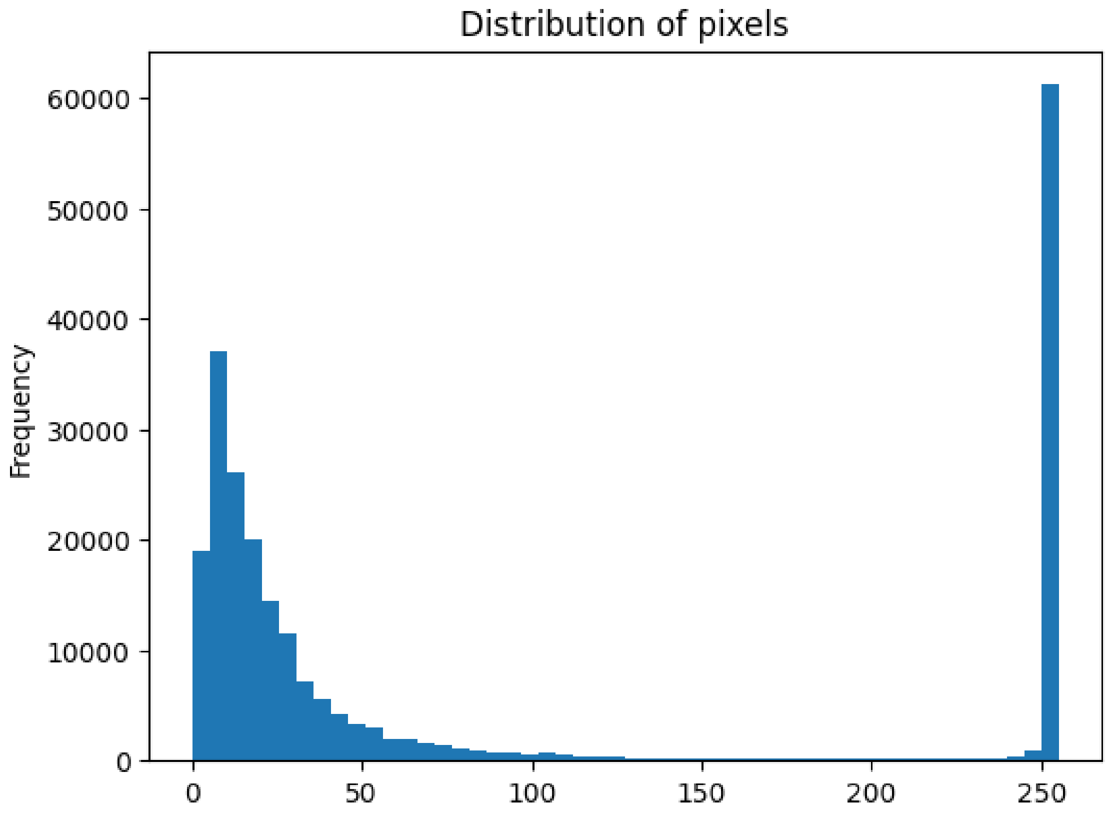
The pictures were captured at various intervals throughout the video, which was recorded on a smartphone documenting the metal deposition process.

Below are the images snapshot which were used for the image processing:

1.  2.  3. 

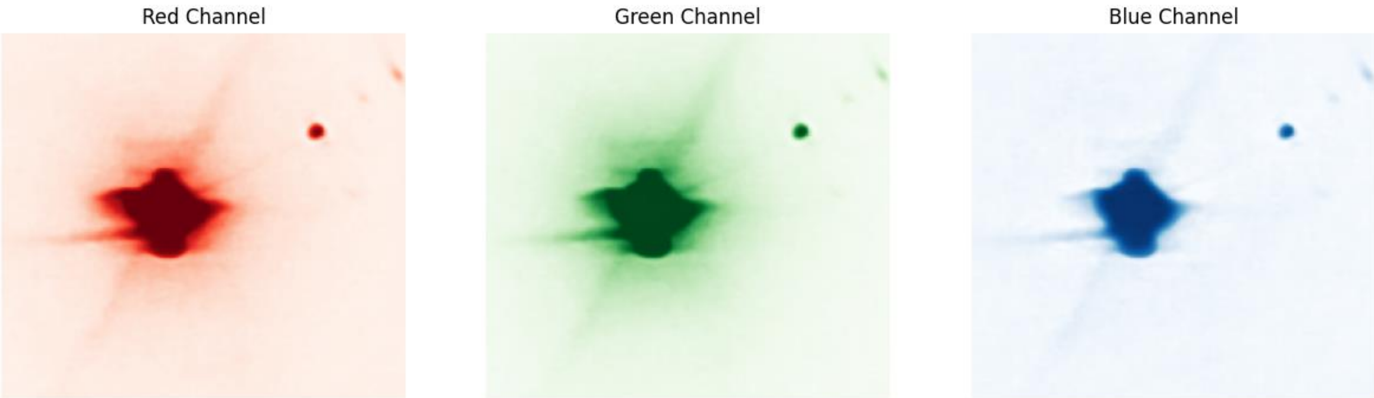
4.2 Distribution of Pixels

We've extracted the pixel distribution from the first image, using OpenCV to convert the image to grayscale and then applying a histogram function to calculate the distribution of pixel intensities.



4.3 Analysing RGB Colour Channels

The next step involves examining the RGB color channels to understand the distribution of red, green, and blue within the image. Using OpenCV, we can split the image into its individual RGB components and then analyze the histograms for each channel to identify color patterns and variations.



4.4 Edge Detection (Paper 4)

To enhance the image for further analysis, we applied a Gaussian blur followed by Canny edge detection.

- Gaussian Blur: This technique smooths the image by reducing noise and detail through a weighted average of surrounding pixels, where the weights are determined by a Gaussian function. It helps in minimizing sharp transitions and artifacts that could affect subsequent operations.
- Canny Edge Detection: After blurring, we applied Canny edge detection to identify and highlight the edges within the image. This method uses gradients to detect abrupt changes in pixel intensity, thereby emphasizing the contours and edges, which can be useful for tasks like object detection or boundary delineation.

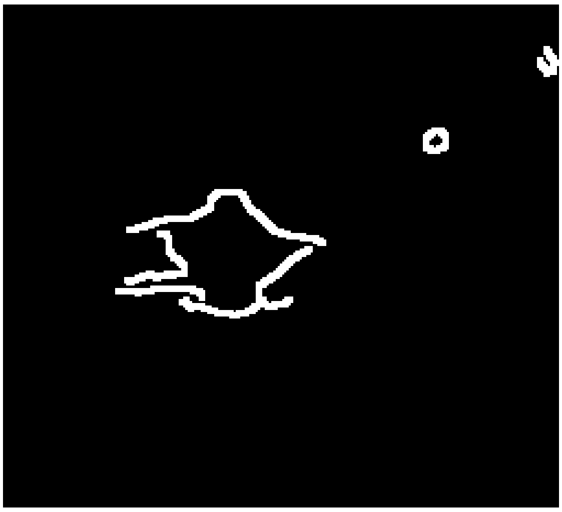


Fig 4.5.1. Edge detected shape

Fig 4.5.2. "Thresholded Image with Highlighted Edges"

4.5 Analysing Images with Edge Detected Images

- Comparing the original image with its edge-detected equivalent in the context of image analysis provides crucial information about the underlying details and structure. Many details are present in the original image, such as colour, texture, and delicate gradients. These characteristics are frequently intricate, making it difficult to examine them for particular structures or patterns.

  The goal of edge detection, on the other hand, is to pinpoint the boundaries and contours inside the image by detecting noticeable changes in pixel intensity. We can make out more details in the image by using Gaussian blur to minimise noise and then Canny edge detection to separate edges.

- It is possible to isolate particular characteristics by comparing the edge-detected version of the image with the original. With its emphasis on boundaries, the edge-detected image makes it easier for humans to trace item outlines, recognise important structures, and clearly analyse the composition. When the emphasis is on forms and transitions rather than on intricate details of colour and texture, this technique is quite helpful. We obtain a thorough grasp of the image's content by analysing both the original and edge-detected images. We can then use this understanding to perform a variety of analytical tasks, such as object detection and image segmentation.

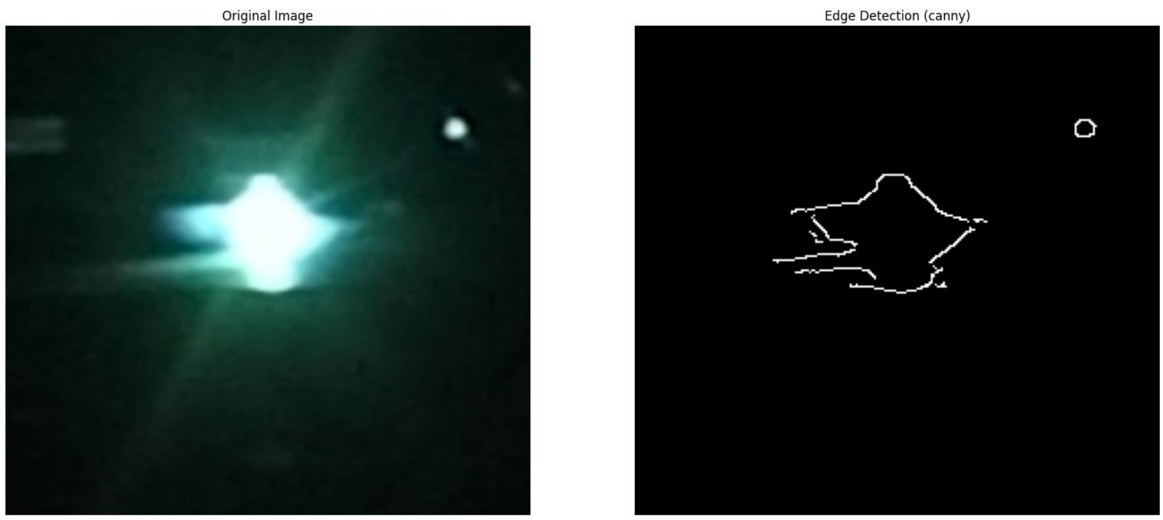


Fig. Comparison between original image

- The Fig. 4.5.3 is a side-by-side comparison of two grayscale images, with highlighted keypoints and lines indicating "good matches" between corresponding features in the images. This depiction shows where the images share similarities and can be used to gauge their alignment or detect common elements.

- These visualizations are useful for applications like image stitching, where the goal is to combine multiple images into a coherent panorama, or for tasks requiring image comparison, such as detecting changes or analyzing structural similarities.

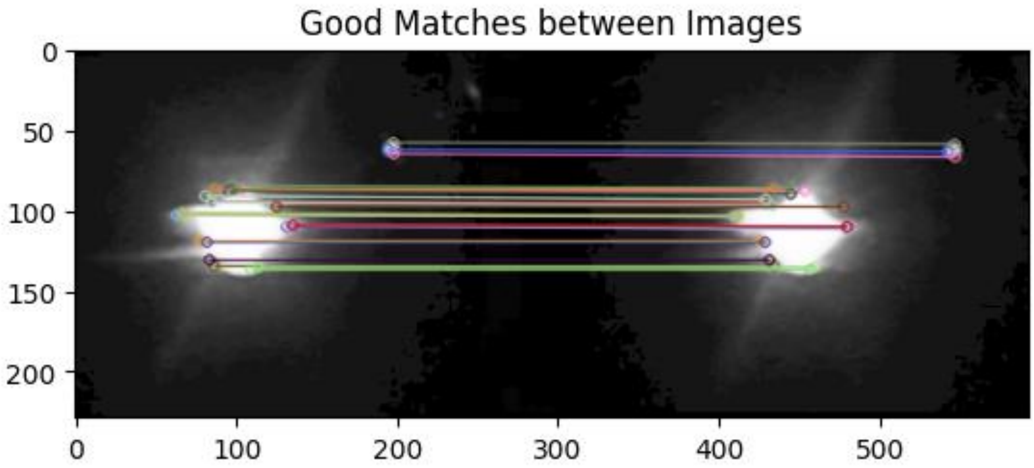


Fig. 4.5.3

4.6 Ellipse Detection and Analysis from Contour Extraction

- In this process, we began by loading an image to analyze its shapes and contours. If the image loaded successfully, we converted it to grayscale and applied Gaussian blur to smooth out the noise. This preparation step helps in reducing spurious edges that might lead to inaccurate contour detection.

- Next, Canny edge detection was employed to highlight the edges in the image, followed by dilation to enhance the edge boundaries. Using these processed edges, contours were found using the OpenCV method `cv2.findContours`. Once the contours were extracted, we looped through each one to determine if it was suitable for ellipse fitting—this was achieved by checking that the contour contained at least five points.

- For each valid contour, we fitted an ellipse and drew it onto a copy of the original image. By calculating ellipse parameters, we extracted the center coordinates, the major and minor axes lengths, and the orientation angle. This information was stored in a list for further reference and analysis.

- The output (fig 4.6.1) included two visualizations. The first subplot displayed the original image, providing context for where the ellipses were drawn. The second subplot showcased the result after fitting ellipses on the detected contours, highlighting each ellipse with a green outline. This visualization helped identify prominent shapes in the image and allowed for further geometric analysis.

- Additionally, we printed key metrics for each detected ellipse, including its center, major and minor axes, and orientation in degrees. This data can be useful for understanding the distribution and characteristics of elliptical shapes within the image, which has applications in various fields such as materials analysis, object detection, and pattern recognition.

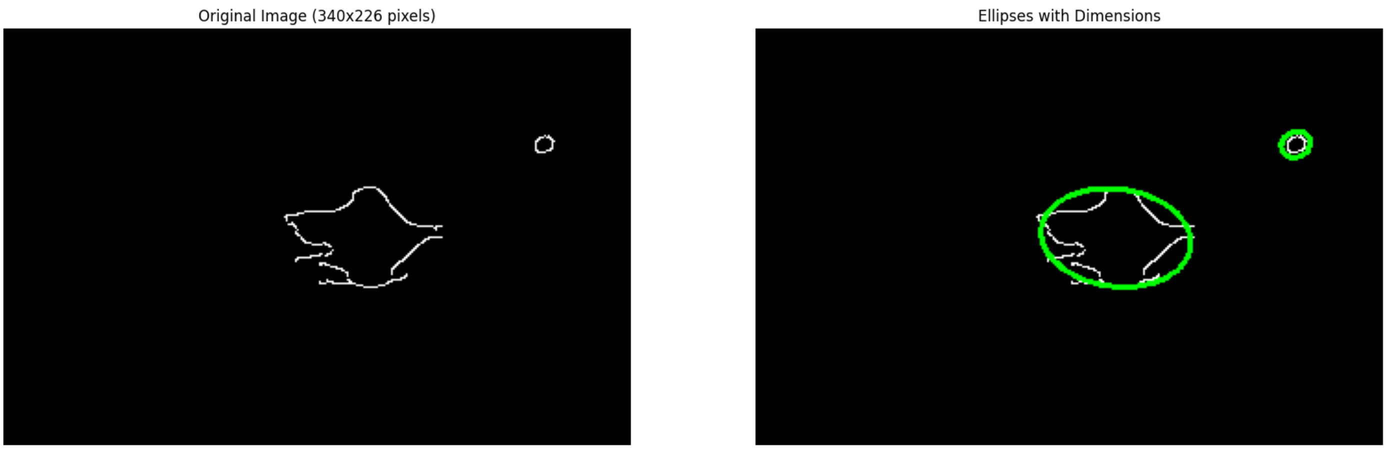


Fig. 4.6.1

Dimensions of the ellipse created.

Image dimensions: 340x226 pixels

Ellipse 1:
Center: (194.8719024658203, 113.22708129882812)
Major Axis: 81.44
Minor Axis: 52.79
Orientation: 97.14 degrees
Ellipse 2:
Center: (292.2979431152344, 62.86743927001953)
Major Axis: 16.54
Minor Axis: 14.00
Orientation: 64.01 degrees

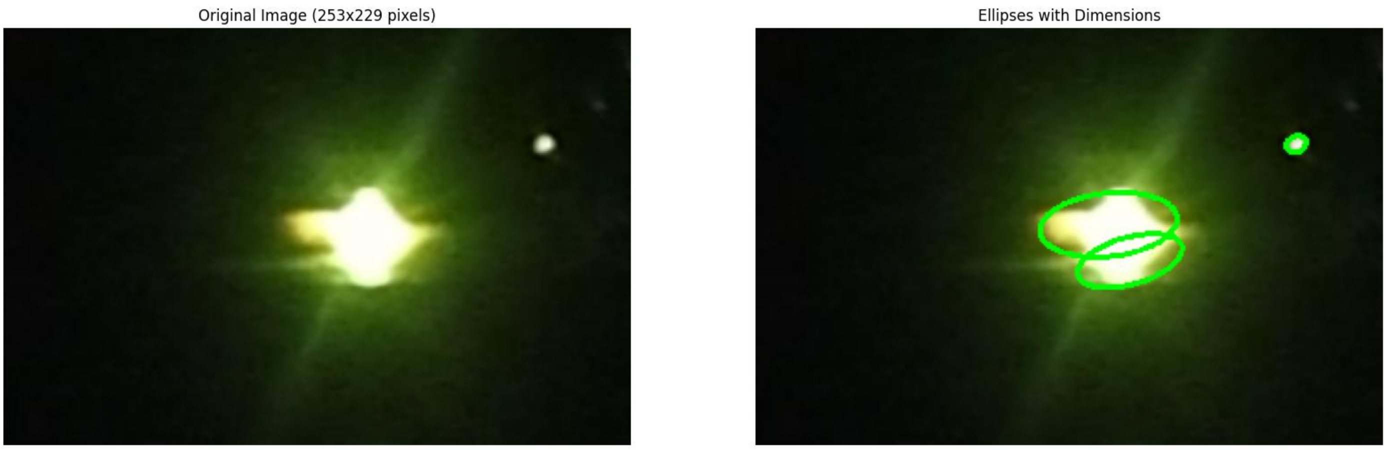


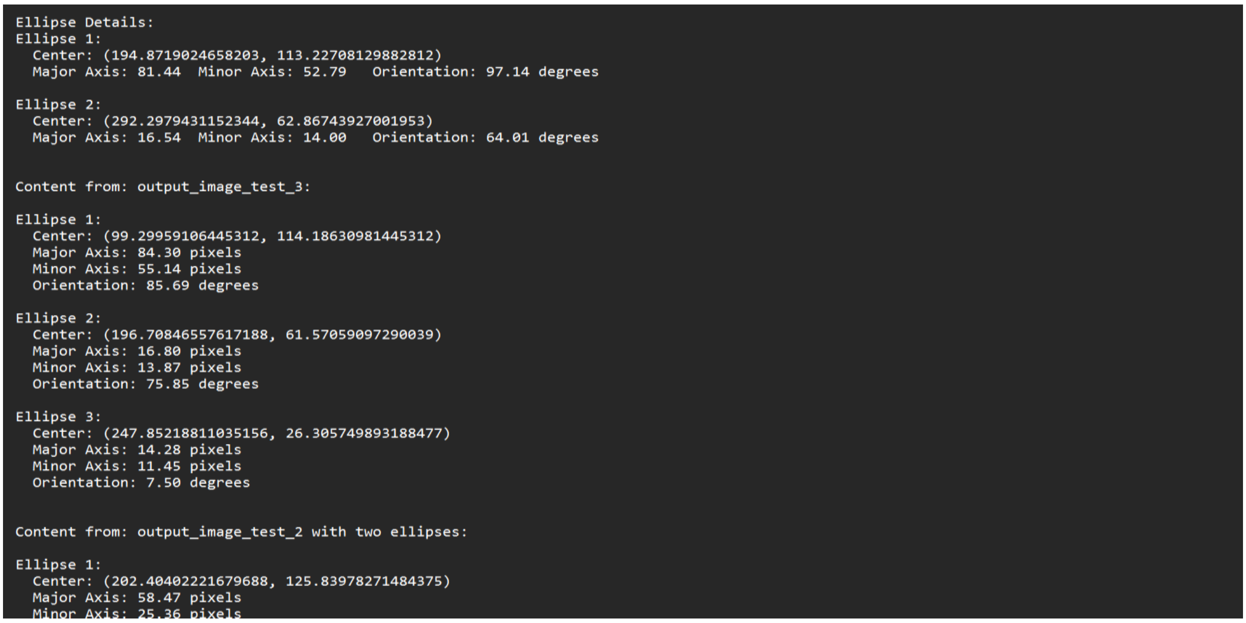
Fig. 4.6.2 Using two ellipses to get dimensions of our edge detected shape.

```
Image dimensions: 253x229 pixels
 Ellipse 1:
    Center: (202.40402221679688, 125.83978271484375)
    Major Axis: 58.47
    Minor Axis: 25.36
    Orientation: 74.50 degrees
Ellipse 2:
    Center: (190.76400756835938, 106.391357421875)
    Major Axis: 74.50
    Minor Axis: 34.33
    Orientation: 83.84 degrees
Ellipse 3:
    Center: (292.4830322265625, 62.72209548950195)
    Major Axis: 11.39
    Minor Axis: 8.67
    Orientation: 64.88 degrees
```

4.7 Creating a Text File data to store Ellipse data

A text file named "ellipse_dimensions.txt" is produced in order to store ellipse data that has been retrieved from a picture. Each detected ellipse's index, centre coordinates, major and minor axis lengths, and orientation angle in degrees are all included in this file. The data for the ellipses is arranged in a readable manner and is presented in a structured style.

You can download this text file separately, which is a helpful resource for additional analysis, calculations, or project integration. This kind of flexible and convenient data storage for ellipse-related files lets you manipulate the information without being dependent on the source image or graphical outputs.



```
Ellipse Details:
Ellipse 1:
  Center: (194.8719024658203, 113.22708129882812)
  Major Axis: 81.44  Minor Axis: 52.79   Orientation: 97.14 degrees

Ellipse 2:
  Center: (292.2979431152344, 62.86743927001953)
  Major Axis: 16.54  Minor Axis: 14.00   Orientation: 64.01 degrees


Content from: output_image_test_3:

Ellipse 1:
  Center: (99.29959106445312, 114.18630981445312)
  Major Axis: 84.30 pixels
  Minor Axis: 55.14 pixels
  Orientation: 85.69 degrees

Ellipse 2:
  Center: (196.70846557617188, 61.57059097290039)
  Major Axis: 16.80 pixels
  Minor Axis: 13.87 pixels
  Orientation: 75.85 degrees

Ellipse 3:
  Center: (247.85218811035156, 26.305749893188477)
  Major Axis: 14.28 pixels
  Minor Axis: 11.45 pixels
  Orientation: 7.50 degrees


Content from: output_image_test_2 with two ellipses:

Ellipse 1:
  Center: (202.40402221679688, 125.83978271484375)
  Major Axis: 58.47 pixels
  Minor Axis: 25.36 pixels
```

Fig. 4.7.1 Text File showing dimensions of ellipses taken

References:

1. Paper 1
2. Paper 2
3. Paper3
4. https://youtu.be/kSqxn6zGE0c?si=Q8kPF0ujjPa2_hsw
5. Session - 1: What is an image?What is image processing?
6. Wire and Arc Additive Manufacture (WAAM) of carbon steel cylinder
7. Wire Arc Additive Manufacturing (WAAM) l Mechanical Engineering
8. Paper 4

Python notebook:
Google Colab Notebook