

Computer Vision Algorithm for Surface Defects Identification in TIG Welded Joints

Sunil Kumar^{1,*}, Adarsh Kumar²

Abstract

Quality monitoring of welded joints is still quite a tedious job for many industries. The surface defects on welded joints arises from the improper selection of various input parameters which further result bad quality weld. This paper highlight the application of a computer vision algorithm called discrete Fourier transformation approach for identification of surface defects present on TIG welded joints.

Keywords: Machine vision, surface defects, TIG welding, Fourier transformation

INTRODUCTION

Computer Vision has been one of the most popular and profitable industries over the past decade, especially in the manufacturing industry. Technology is making a huge impact in all stages of the manufacturing process, from using computer-assisted computer archive to modern robotics in R&D labs [1, 2]. These technological advances in the manufacturing sector have helped reduce production defects, improve product quality, increase flexibility, reduce time and cost, and achieve higher productivity.

Artificial Intelligence, In-Depth Learning, Machine Learning, Industry 4.0, Cloud Computing, Internet of Things (IoT), and other advanced technologies present major challenges for users and developers of visualization when choosing the right application for their applications.

In the worst case of mechanical malfunction or mechanical failure, and the work stops. Any material-based business should consider storing the necessary equipment or equipment. Adjustment Forecasting is defined as the process of using learning equipment and IoT to assess where asset management is required. This allows the manufacturer to use the working time of the equipment and reduce the efficiency as shown in the Figure 1. For example, an industrial retailer develops a software system that collects images from 7000 robots connected to robots in 38 factories. Along with metadata, this image collection can detect and prevent 72 items failures.

*Author for Correspondence

Sunil Kumar

¹Research Scholar Department of Mechanical Engineering, Bansal Institute of Engineering and Technology, Lucknow India

²Assistant Professor, Department of Mechanical Engineering, Bansal Institute Of Engineering and Technology, Lucknow India

Received Date: March 09, 2021

Accepted Date: April 20, 2021

Published Date: April 29, 2021

Citation: Sunil Kumar, Adarsh Kumar. Computer Vision Algorithm for Surface Defects Identification in TIG Welded Joints. Research & Reviews: Journal of Computational Biology. 2021; 10(1): 14–23p.

As manufacturing companies deal with the production of large quantities, larger or heavier machinery should be used. These pieces of equipment should be inspected regularly to avoid the downtime of the equipment. Manual testing of each machine in the manufacturing industry is not only time consuming but also costly and prone to errors. Therefore, practitioners traditionally rely solely on functional care. The idea was to repair the equipment only after it was broken or damaged. However, replacing equipment using this concept can have negative impacts on employee productivity, production quality, and cost. The Wall Street Journal estimates that \$ 50 billion is spent annually at random.



Figure 1. Predictive maintenance.

On the other hand, what would happen if manufacturing companies had the opportunity to anticipate the state of the operating system and take corrective measures to avoid possible deterioration? By incorporating that amazing approach, manufacturers can not only reduce cases of mechanical failure but also improve the productivity of their employees and improve cost savings. Machine viewing systems can effectively perform these functions and ensure minimal downtime.

Robots have been part of the manufacturing industry for some time now. Now consider empowering these robots with AI and machine vision. Such robots can easily capture images of all machines. Photos and more information about the equipment will be sent to the cloud, where analysis and performance is performed. The data, after calculation, will help producers gain a reasonable understanding of whether those specific machines are going to fail or not. At improved customer satisfaction levels, manufacturers must provide the highest quality products to their customers. If not done correctly, producers will see that their customers, revenue, and profit margins are reduced. Product testing and quality control, therefore, are important production functions that cannot be ignored. Doing these tasks yourself can be frustrating, time-consuming, and repetitive. Otherwise, manufacturers cannot guarantee 100 percent accurate results due to human involvement. In addition, sometimes a deformity is not visible to the human eye. However, with machine vision systems, all of these problems can be eliminated. Manufacturing companies can make and adjust these processes in a very short time.

With machine vision systems, industrial manufacturers can detect defects, cracks, or any defects in the body product. Alternatively, these systems can easily check the precise and accurate measurements of materials or components used while the product is being assembled. Machine viewing programs will take pictures of products or goods. These images will then be sent to a computer for processing. Details of any disabilities or errors will be sent to the affected staff. This is how manufacturers can implement automated product testing systems for automatic product testing and accurate quality control for customer satisfaction. Machine viewing systems, with real-time image processing capabilities, can collect images of a productive plant. Photos will also capture employees and their actions. The data is then entered into a computer program when it is analyzed and processed [3, 4]. The result will contain important information on whether employees are in a dangerous and life-threatening situation. Machine detection systems will inform affected officials of the hazardous site environment, allowing them to take the necessary steps in a timely manner. Therefore, machine vision systems can benefit manufacturing companies by detecting natural hazards or hazards in a timely manner, preventing future hazards.

The manufacturing industry has been at the forefront of adopting new technologies. Using emerging technologies in production, manufacturers create and modify a number of functions. Similarly, machine vision systems in production solve the major challenges companies face,

providing them with benefits such as higher yields, higher production quality, and lower financial costs. A tireless and highly productive machine vision system can assist manufacturers in producing the highest volume of goods as required by the customer at no additional cost. Machine vision is no longer just a good thing to have—it is a need for the manufacturing sector. Applying a concept test to a production or packaging line is a well-established practice. Systems ranging from single-point and self-contained smart-point cameras do the job of testing and delivering a pass / fail result in the control system, to PC-based systems that can include multiple cameras and / or multiple test channels. Viewing programs may be redesigned from existing lines or designed for new ones. Opinion tests can also be used in conjunction with methods to control mathematical processes not only to test critical measurements but also to analyze styles in these measurements. In this way, interventions can be made to correct the process before producing an intolerable product. This is probably the closest predecessor to Industry 4.0 requirements.

Industrial robots are already widely used and with the advent of interactive robots and rapid advances in the use of 3D images, they are widely used in combination, especially vision-guided robots. The vision system identifies the exact location of the object and these links are transmitted to the robot. The rapid development of robotic consultation sites makes this process much easier. One of the most popular features of the 3D robots view is the use of pick and location [5, 6].

The discovery of small, embedded processing boards, often based on ARM design, provides great strength in developing vision systems embedded in other equipment and production processes. Many of the leading image processing libraries and tool tools can now be sent to these platforms, providing a wide range of vision solutions in this way. Combining these processing capabilities with low-cost cameras, including board-level cameras, means that viewing systems can be integrated into a variety of products and processes with high cost costs.

EXPERIMENTAL PROCEDURE

The TIG welded image shown in Figure 2 is firstly imported to the Google Colaboratory platform and is further subjected to the computer vision algorithm developed in Python Programming.



Figure 2. TIG welded joint.

With the help of Fourier transformation algorithm, the image is further decomposed into sine and cosine components. For a square image of size $N \times N$, the two-dimensional DFT is given by:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-i2\pi(\frac{ki}{N} + \frac{lj}{N})} \quad (1)$$

where $f(a,b)$ is the image in the spatial domain and the exponential term is the basis function corresponding to each point $F(k,l)$ in the Fourier space. The equation can be interpreted as: the value of each point $F(k,l)$ is obtained by multiplying the spatial image with the corresponding base function and summing the result.

The basic functions are sine and cosine waves with increasing frequencies, i.e. $F(0,0)$ represents the DC-component of the image, which corresponds to the average brightness, and $F(N-1,N-1)$ represents the highest frequency.

In a similar way, the Fourier image can be re-transformed to the spatial domain. The inverse Fourier transform is given by:

$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) e^{i2\pi(\frac{ka}{N} + \frac{lb}{N})} \quad (2)$$

RESULTS AND DISCUSSIONS

Firstly, important python programming libraries are imported by using following code:

```
!pip install -U -q Pydrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

The Google Drive on which image is uploaded is further linked to the simulation environment by using following code:

```
auth.authenticate_user()
gauth=GoogleAuth()
gauth.credentials=GoogleCredentials.get_application_default()
drive=GoogleDrive(gauth)
```

The NumPy and related machine vision algorithms are imported by using following code:

```
import cv2
import numpy as np
from math import exp
```

For the given welded image, the processed image is obtained with the help of following code:

```
plt.figure(figsize=(6.4*5, 4.8*5), constrained_layout=False)
```

```
img_c1 = cv2.imread("surface.jpg", 0)
img_c2 = np.fft.fft2(img_c1)
img_c3 = np.fft.fftshift(img_c2)
img_c4 = np.fft.ifftshift(img_c3)
img_c5 = np.fft.ifft2(img_c4)
```

```
plt.subplot(151), plt.imshow(img_c1, "gray"), plt.title("Original Image")
plt.subplot(152), plt.imshow(np.log(1+np.abs(img_c2)), "gray"), plt.title("Spectrum")
plt.subplot(153), plt.imshow(np.log(1+np.abs(img_c3)), "gray"), plt.title("Centered Spectrum")
plt.subplot(154), plt.imshow(np.log(1+np.abs(img_c4)), "gray"), plt.title("Decentralized")
plt.subplot(155), plt.imshow(np.abs(img_c5), "gray"), plt.title("Processed Image")
```

The output image obtained is shown in Figure 3.

To obtain the spectrum and phase of the processed image the following Python code is used:

```
plt.figure(figsize=(6.4*5, 4.8*5), constrained_layout=False)
```

```
img = cv2.imread("surface.jpg", 0)
```

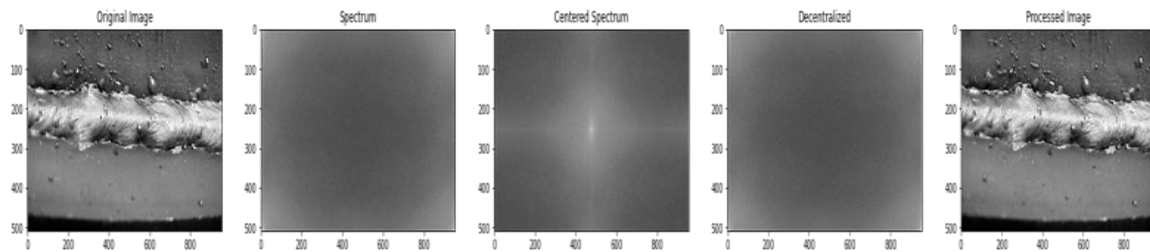


Figure 3. Conversion of original image to processed image.

```
original = np.fft.fft2(img)
plt.subplot(131), plt.imshow(np.log(np.abs(original)), "gray"), plt.title("Spectrum")

plt.subplot(132), plt.imshow(np.angle(original), "gray"), plt.title("Phase Angle")
plt.show()
```

The output obtained is shown in Figure 4.

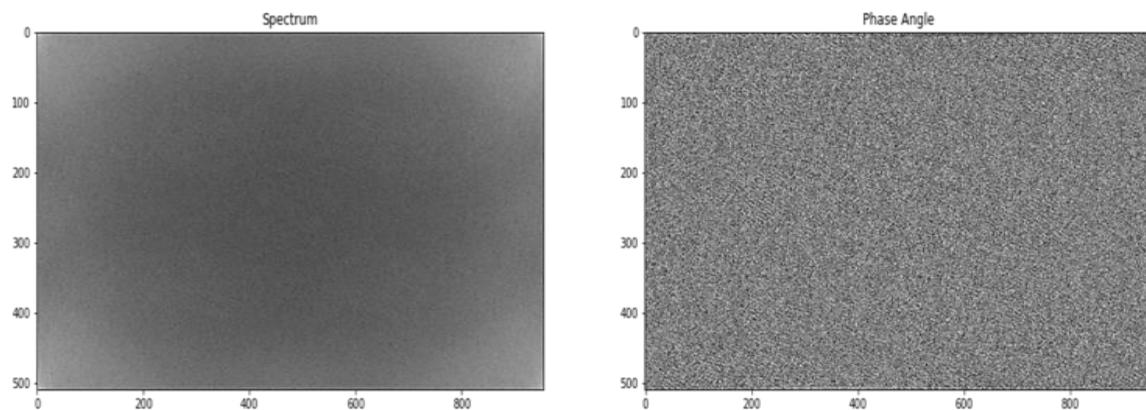


Figure 4. Spectrum and phase angle of the processed image.

For subjecting, the given welded image to various Machine Vision filters for extracting useful information following code is used:

```
def distance(point1,point2):
    return np.sqrt((point1[0]-point2[0])**2 + (point1[1]-point2[1])**2)

def idealFilterLP(D0,imgShape):
    base = np.zeros(imgShape[:2])
    rows, cols = imgShape[:2]
    center = (rows/2,cols/2)
    for x in range(cols):
        for y in range(rows):
            if distance((y,x),center) < D0:
                base[y,x] = 1
    return base

def idealFilterHP(D0,imgShape):
    base = np.ones(imgShape[:2])
    rows, cols = imgShape[:2]
    center = (rows/2,cols/2)
    for x in range(cols):
```

```

for y in range(rows):
if distance((y,x),center) < D0:
base[y,x] = 0
return base

defbutterworthLP(D0,imgShape,n):
base = np.zeros(imgShape[:2])
rows, cols = imgShape[:2]
center = (rows/2,cols/2)
for x in range(cols):
for y in range(rows):
base[y,x] = 1/(1+(distance((y,x),center)/D0)**(2*n))
return base

defbutterworthHP(D0,imgShape,n):
base = np.zeros(imgShape[:2])
rows, cols = imgShape[:2]
center = (rows/2,cols/2)
for x in range(cols):
for y in range(rows):
base[y,x] = 1-1/(1+(distance((y,x),center)/D0)**(2*n))
return base

defgaussianLP(D0,imgShape):
base = np.zeros(imgShape[:2])
rows, cols = imgShape[:2]
center = (rows/2,cols/2)
for x in range(cols):
for y in range(rows):
base[y,x] = exp(((distance((y,x),center)**2)/(2*(D0**2))))
return base

defgaussianHP(D0,imgShape):
base = np.zeros(imgShape[:2])
rows, cols = imgShape[:2]
center = (rows/2,cols/2)
for x in range(cols):
for y in range(rows):
base[y,x] = 1-exp(((distance((y,x),center)**2)/(2*(D0**2))))
return base

```

The original image is again subjected to processing with the help of following code:

```

plt.figure(figsize=(6.4*5, 4.8*5), constrained_layout=False)

img = cv2.imread("surface.jpg", 0)
plt.subplot(161), plt.imshow(img, "gray"), plt.title("Original Image")

original = np.fft.fft2(img)
plt.subplot(162), plt.imshow(np.log(1+np.abs(original)), "gray"), plt.title("Spectrum")

center = np.fft.fftshift(original)

```

```
plt.subplot(163), plt.imshow(np.log(1+np.abs(center)), "gray"), plt.title("Centered Spectrum")
```

```
LowPassCenter = center * idealFilterLP(50,img.shape)
plt.subplot(164), plt.imshow(np.log(1+np.abs(LowPassCenter)), "gray"), plt.title("Centered Spectrum
multiply Low Pass Filter")
```

```
LowPass = np.fft.ifftshift(LowPassCenter)
plt.subplot(165), plt.imshow(np.log(1+np.abs(LowPass)), "gray"), plt.title("Decentralize")
```

```
inverse_LowPass = np.fft.ifft2(LowPass)
plt.subplot(166), plt.imshow(np.abs(inverse_LowPass), "gray"), plt.title("Processed Image")
```

```
plt.show()
```

The output obtained is shown in Figure 5.

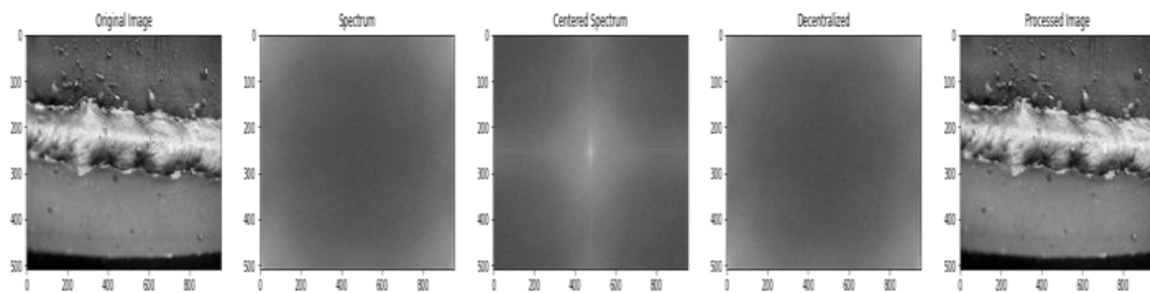


Figure 5. Processed image.

The image is further subjected to low-pass filter with the help of following code:

```
img = cv2.imread("surface.jpg", 0)
original = np.fft.fft2(img)
center = np.fft.fftshift(original)
```

```
plt.figure(figsize=(6.4*5, 4.8*5), constrained_layout=False)
```

```
plt.subplot(151), plt.imshow(np.log(1+np.abs(center)), "gray"), plt.title("Spectrum")
```

```
LowPass = idealFilterLP(50,img.shape)
plt.subplot(152), plt.imshow(np.abs(LowPass), "gray"), plt.title("Low Pass Filter")
```

```
LowPassCenter = center * idealFilterLP(50,img.shape)
plt.subplot(153), plt.imshow(np.log(1+np.abs(LowPassCenter)), "gray"), plt.title("Centered Spectrum
multiply Low Pass Filter")
```

```
LowPass = np.fft.ifftshift(LowPassCenter)
plt.subplot(154), plt.imshow(np.log(1+np.abs(LowPass)), "gray"), plt.title("Decentralize")
```

```
inverse_LowPass = np.fft.ifft2(LowPass)
plt.subplot(155), plt.imshow(np.abs(inverse_LowPass), "gray"), plt.title("Processed Image")
```

```
plt.show()
```

The output obtained is shown in Figure 6.

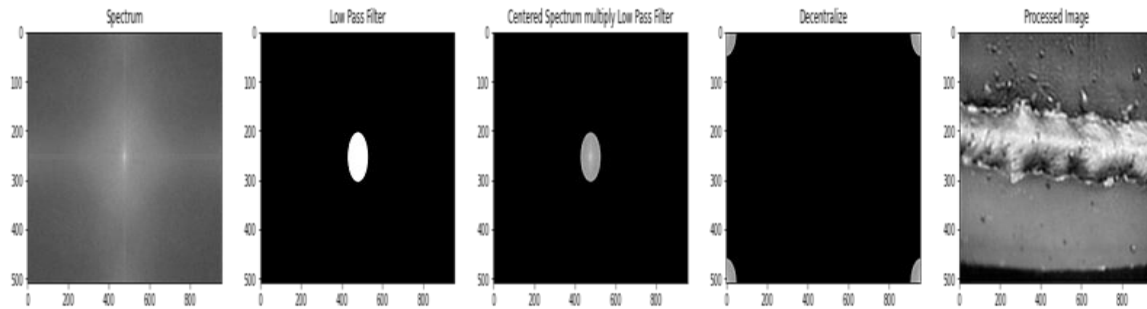


Figure 6. Subjection of image to low-pass filter.

Secondly, the image is further subjected to High Pass filter with the help of following code:

```
img = cv2.imread("surface.jpg", 0)
original = np.fft.fft2(img)
center = np.fft.fftshift(original)

plt.figure(figsize=(6.4*5, 4.8*5), constrained_layout=False)

plt.subplot(151), plt.imshow(np.log(1+np.abs(center)), "gray"), plt.title("Spectrum")

HighPass = idealFilterHP(50,img.shape)
plt.subplot(152), plt.imshow(np.abs(HighPass), "gray"), plt.title("High Pass Filter")

HighPassCenter = center * idealFilterHP(50,img.shape)
plt.subplot(153), plt.imshow(np.log(1+np.abs(HighPassCenter)), "gray"), plt.title("Centered Spectrum
multiply High Pass Filter")

HighPass = np.fft.ifftshift(HighPassCenter)
plt.subplot(154), plt.imshow(np.log(1+np.abs(HighPass)), "gray"), plt.title("Decentralize")

inverse_HighPass = np.fft.ifft2(HighPass)
plt.subplot(155), plt.imshow(np.abs(inverse_HighPass), "gray"), plt.title("Processed Image")

plt.show()
```

Weld subjected to high pass filter can be obtained by the following code:

```
img = cv2.imread("surface.jpg", 0)
original = np.fft.fft2(img)
center = np.fft.fftshift(original)

plt.figure(figsize=(6.4*5, 4.8*5), constrained_layout=False)

HighPassCenter = center * idealFilterHP(50,img.shape)
HighPass = np.fft.ifftshift(HighPassCenter)
inverse_HighPass = np.fft.ifft2(HighPass)
plt.subplot(131), plt.imshow(np.abs(inverse_HighPass), "gray"), plt.title("Ideal High Pass")

HighPassCenter = center * butterworthHP(50,img.shape,10)
HighPass = np.fft.ifftshift(HighPassCenter)
inverse_HighPass = np.fft.ifft2(HighPass)
plt.subplot(132), plt.imshow(np.abs(inverse_HighPass), "gray"), plt.title("Butterworth High Pass
(n=10)")
```

```
HighPassCenter = center * gaussianHP(50,img.shape)
HighPass = np.fft.ifftshift(HighPassCenter)
inverse_HighPass = np.fft.ifft2(HighPass)
plt.subplot(133), plt.imshow(np.abs(inverse_HighPass), "gray"), plt.title("Gaussian High Pass")

plt.show()
```

The obtained output is shown in Figure 7.

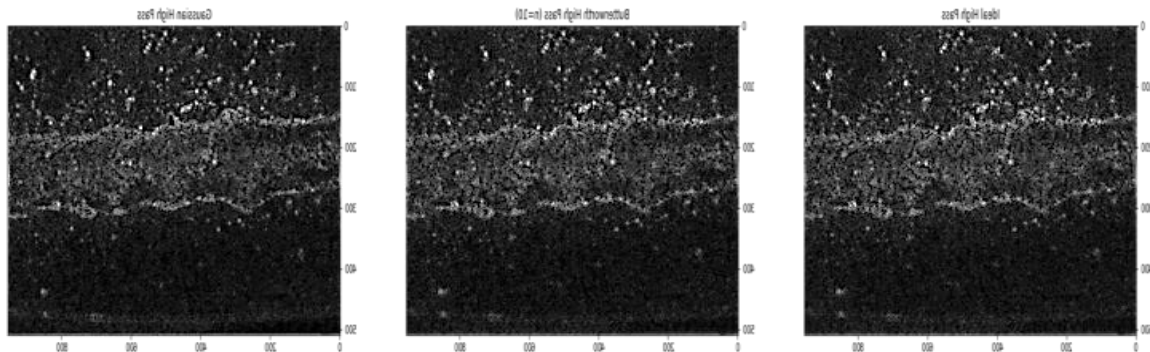


Figure 7. Weld image subjected to high pass filter.

The image subjected to Gaussian low pass can be visualized by following code:

```
img = cv2.imread("surface.jpg", 0)
original = np.fft.fft2(img)
center = np.fft.fftshift(original)

plt.figure(figsize=(6.4*5, 4.8*5), constrained_layout=False)

plt.subplot(131), plt.imshow(img, "gray"), plt.title("Original Image")
LowPassCenter = center * gaussianLP(50,img.shape)
LowPass = np.fft.ifftshift(LowPassCenter)
inverse_LowPass = np.fft.ifft2(LowPass)
plt.subplot(132), plt.imshow(np.abs(inverse_LowPass), "gray"), plt.title("Gaussian Low Pass")
HighPassCenter = center * gaussianHP(50,img.shape)
HighPass = np.fft.ifftshift(HighPassCenter)
inverse_HighPass = np.fft.ifft2(HighPass)
plt.subplot(133), plt.imshow(np.abs(inverse_HighPass), "gray"), plt.title("Gaussian High Pass")
plt.show()
```

The output of the code is shown in Figure 8.

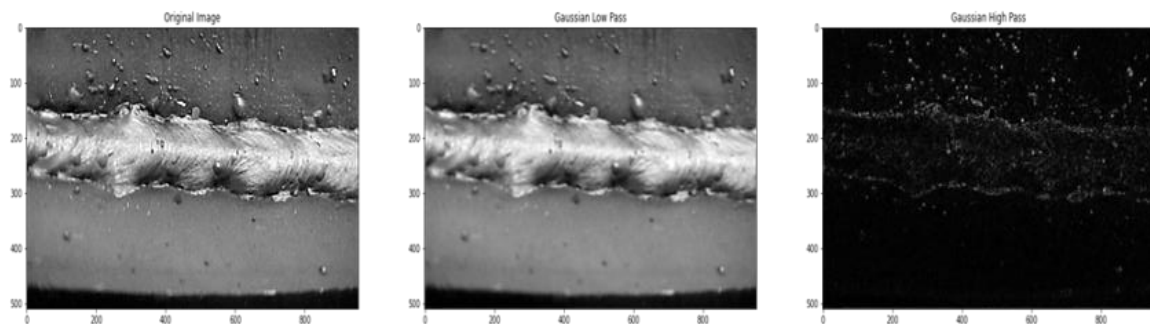


Figure 8. Image obtained after Gaussian low-pass filter subsection.

CONCLUSION

The study suggests a novel approach to land-edge in the Tungsten Inert Gas welded joint using a computer viewing process. It is evident that the Gaussian filter works better than the ideal filter and the Butterworth filter because it results in less distortion and more smoothness in the image created. In addition, it is clear that a high pass filter has a better advantage than other filters by getting divine edges. In future research, the Fourier transformation could play a key role in image processing by providing additional flexibility.

Acknowledgment

Author would like to thanks his professor and parents for the successful completion of this research work.

REFERENCES

1. Jain R, Kasturi R, Schunck BG. Machine vision. Vol. 5. New York: McGraw-Hill; 1995.
2. Davies ER. Machine vision: theory, algorithms, practicalities. Elsevier; 2004.
3. Chen YR, Chao K, Kim MS. Machine vision technology for agricultural applications. Comput Electron Agric. 2002;36(2–3):173–91. doi: 10.1016/S0168–1699(02)00100-X.
4. Oren M, Nayar SK. Generalization of the Lambertian model and implications for machine vision. Int J Comput Vision. 1995;14(3):227–51. doi: 10.1007/BF01679684.
5. Pérez L, Rodríguez Í, Rodríguez N, Usamentiaga R, García DF. Robot guidance using machine vision techniques in industrial environments: A comparative review. Sensors (Basel). 2016;16(3):335. doi: 10.3390/s16030335, PMID 26959030.
6. Vernon D. Machine vision-Automated visual inspection and robot vision. NASA STI/Recon Tech Rep A. 1991;92:40499.