明志科技大學
MING CHI UNIVERSITY OF TECHNOLOGY

生醫暨醫療產品研發博士學程

DIP of the temperature measurement device prototype and an AOI system

指導老師:張創然老師
博一甲 廖浩延 Hao-Yen Liao
e-mail: howard31418@mail.mcut.edu.tw
886-2-29089899 ext. 4580

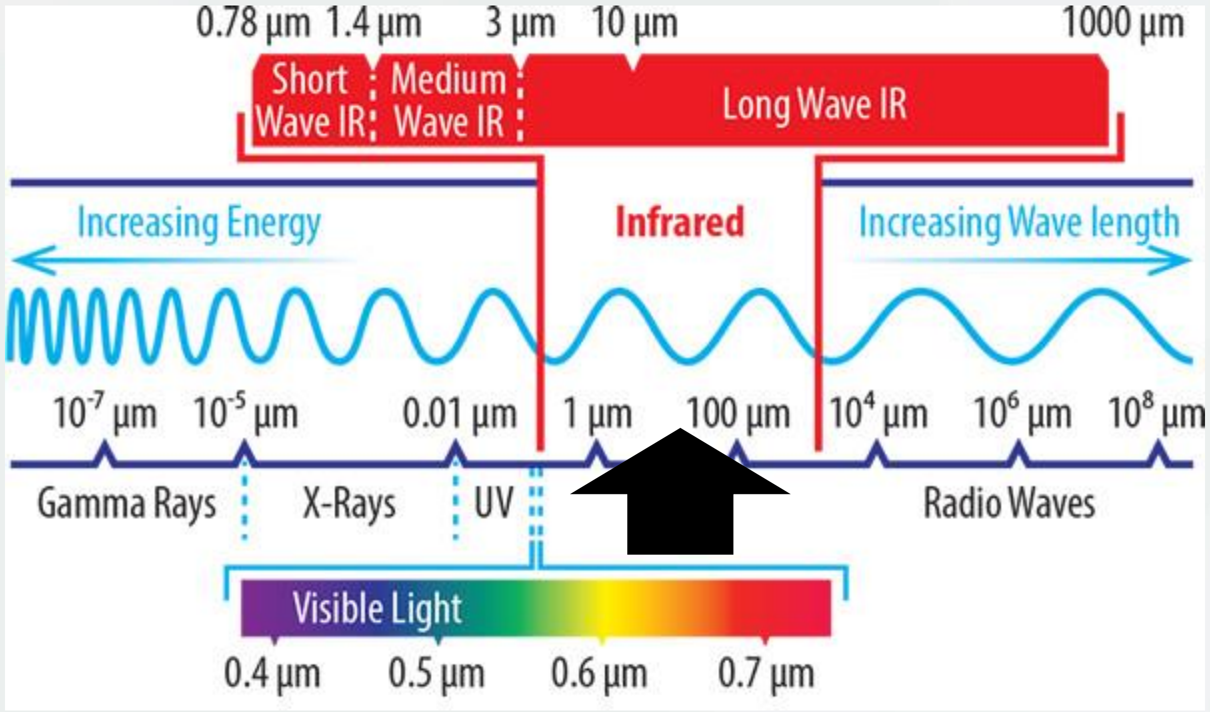
Outline

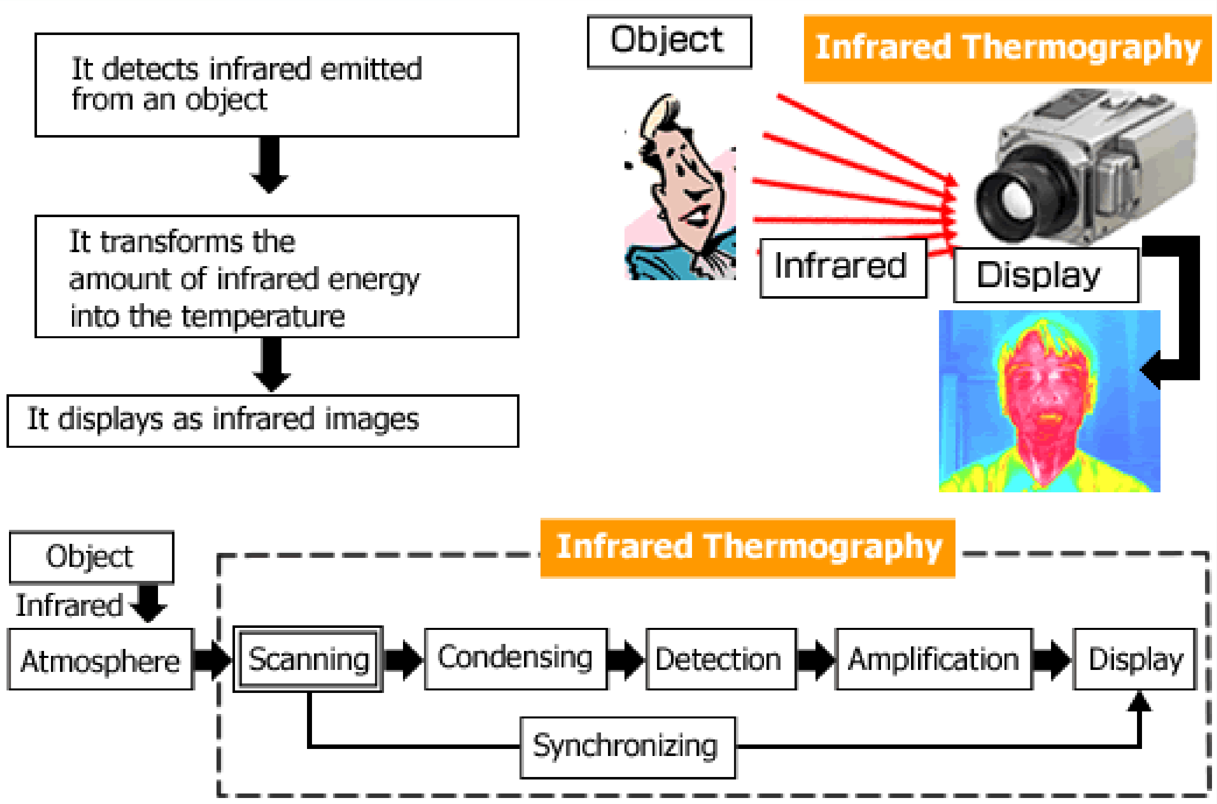
- **Introduction**
  - ✓ **Infrared Thermography**
  - ✓ **AOI system**
- **Materials and Methods**
- **Results**
  - ✓ Digital image process of Infrared Thermography
  - ✓ Find the holes from the image and make marks
- **Summary**

Temperature measurement system

Infrared Thermography

- Infrared thermography is the technique of converting infrared energy (radiant heat) into an image that a person can see and understand.
- The infrared energy emitted from an object is directly proportional to its temperature. Therefore temperatures are accurately measured by the infrared camera.

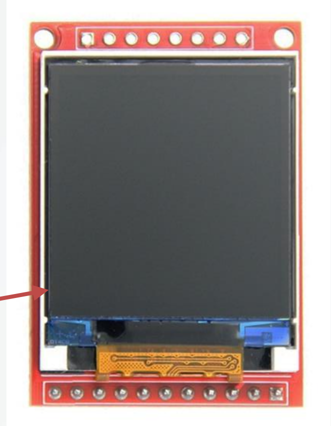




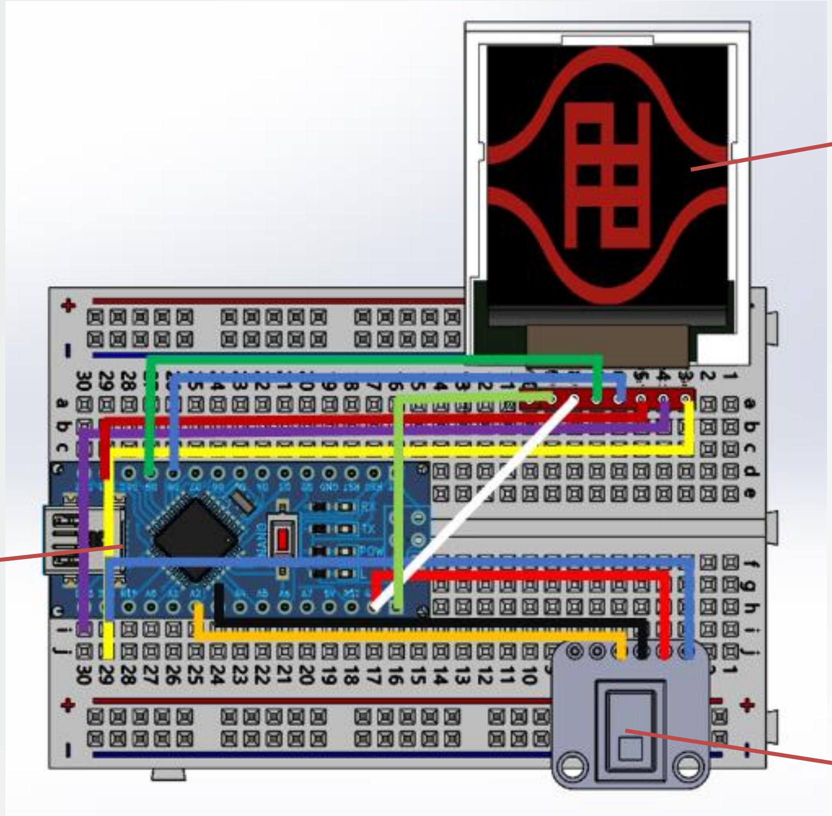
Materials and Methods

Hardware components

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality as the Arduino Duemilanove but in a different package.

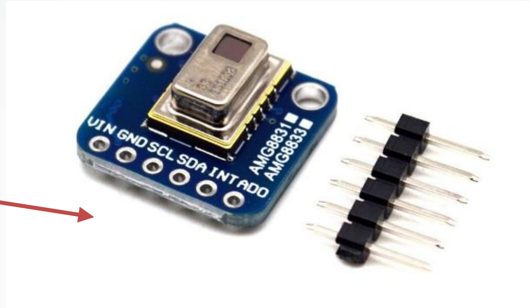


TFT LCD



Test Circuit

Arduino Nano

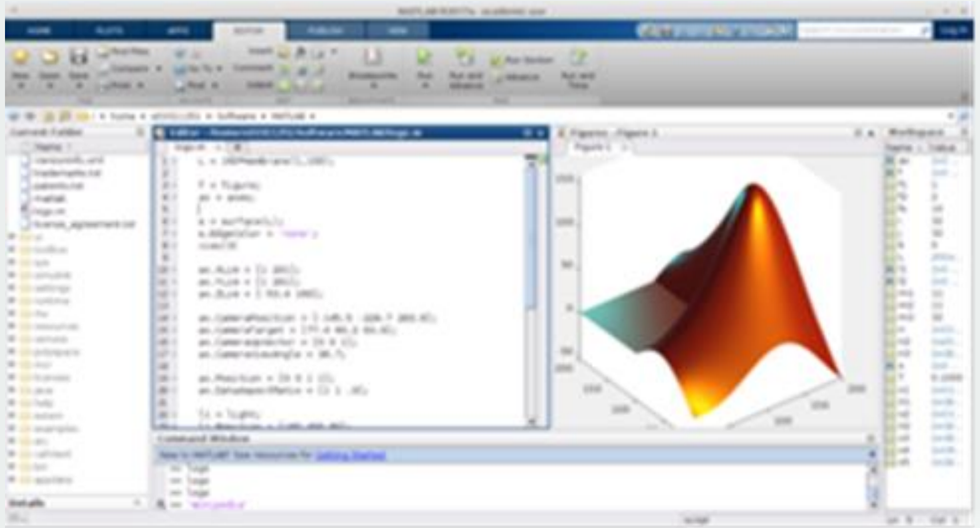


AMG8833

Materials and Methods

MATLAB

- MATLAB is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.



Code:

```
>> x=[27.00 27.25 29.00 26.00 28.25 28.25 25.25 24.75;27.00
29.50 30.00 27.00 29.50 28.00 24.75 24.50;28.00 29.75 30.25
30.25 30.50 27.50 24.50 25.50;30.25 30.25 30.50 30.25 30.50
27.50 24.75 27.75;30.25 30.50 30.50 30.50 30.25 29.00 26.50
30.25;30.25 30.50 30.25 30.50 30.75 30.25 30.00 29.75;30.00
29.75 30.25 30.25 30.75 30.75 30.25 27.50;29.75 30.50 30.25
30.50 30.25 30.25 29.25 26.75]
>> V=imresize(x, 50,'bicubic');
>> imshow(V)
>> colormap(jet)
>> caxis([20 40])
>> colorbar
```

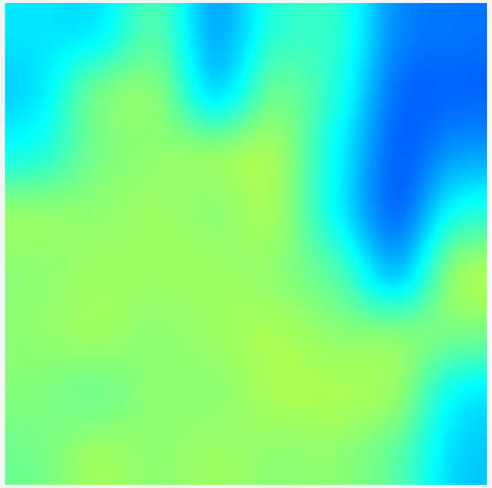
Results

Digital image process of Infrared Thermography

8X8 matrix data

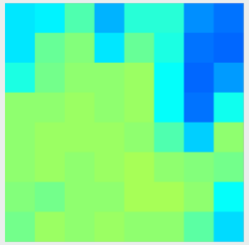
27	27.25	29	26	28.25	28.25	25.25	24.75
27	29.5	30	27	29.5	28	24.75	24.5
28	29.75	30.25	30.25	30.5	27.5	24.5	25.5
30.25	30.25	30.5	30.25	30.5	27.5	24.75	27.75
30.25	30.5	30.5	30.5	30.25	29	26.5	30.25
30.25	30.5	30.25	30.5	30.75	30.25	30	29.75
30	29.75	30.25	30.25	30.75	30.75	30.25	27.5
29.75	30.5	30.25	30.5	30.25	30.25	29.25	26.75

MATLAB
bicubic resize



400X400 pixel image

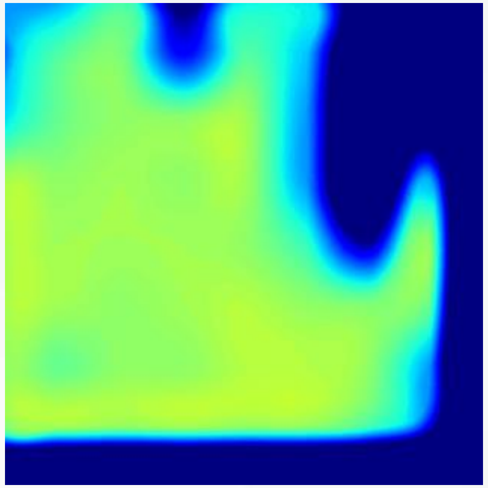
MATLAB



8X8 pixel image

[[165 172 193 134 186 186 113 97]
[165 198 203 165 198 184 97 90]
[184 200 205 205 207 179 90 120]
[205 205 207 205 207 179 97 181]
[205 207 207 207 205 193 151 205]
[205 207 205 207 210 205 203 200]
[203 200 205 205 210 210 205 179]
[200 207 205 207 205 205 195 158]]

Python
bicubic resize

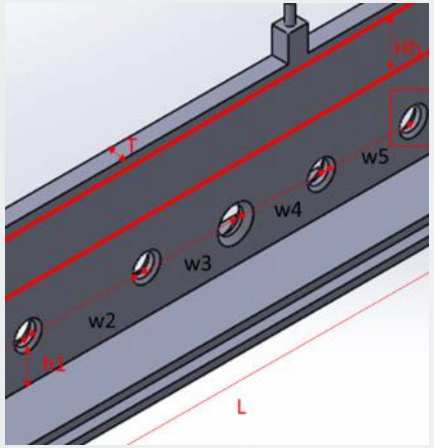


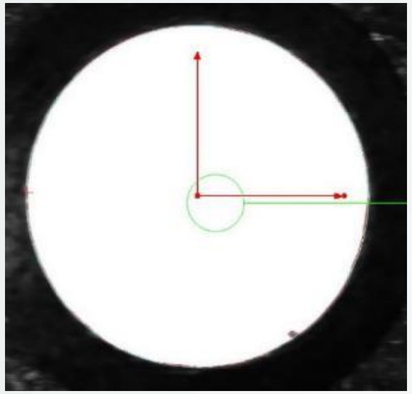
400X400 pixel image

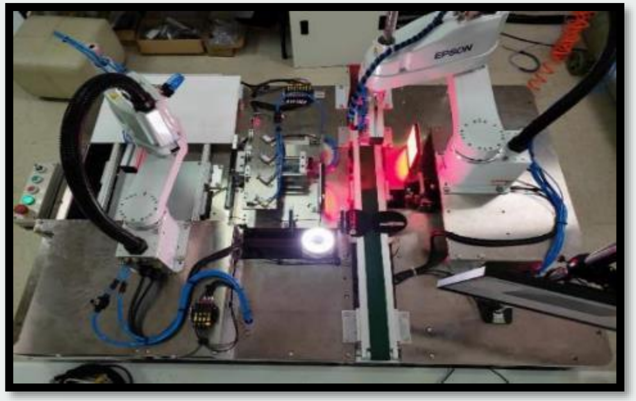
AOI system-defect detection and measurement

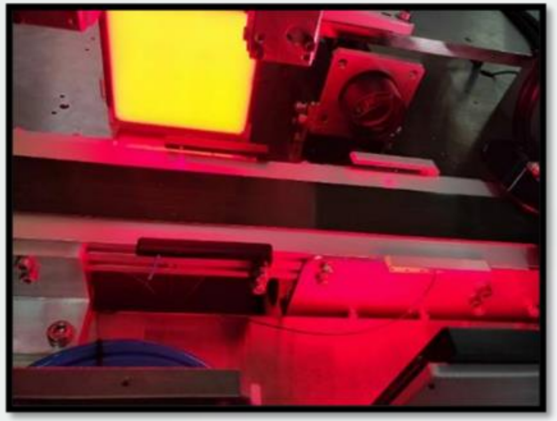
Automated Optical Inspection

明志科技大學
MING CHI UNIVERSITY OF TECHNOLOGY







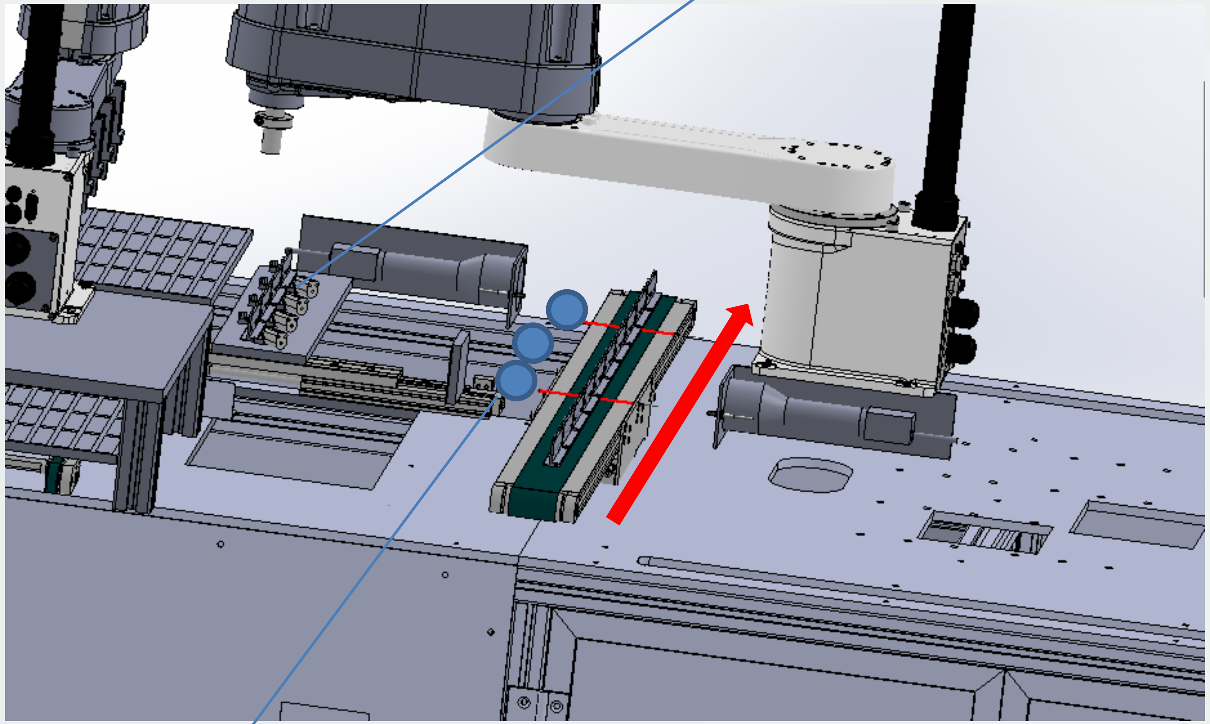


1. Detection resolution 50 μm.
2. Detection efficiency 12pcs/min (60 holes).
3. Create AOI defect database.
4. Programming and testing of robotic arm, integration of machine software and hardware.
5. Customized equipment size.

Unit definition-conveyor + CCD on both sides

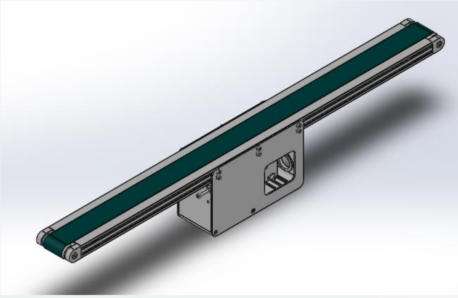
明志科技大學
MING CHI UNIVERSITY OF TECHNOLOGY

Cylinder for gripper



Trigger CCD to take pictures & confirm workpiece.
(Optical fiber, fiber for confirm the workpiece)

Conveyor



Robot

CCD

Light

Moving process:
The conveyor moves the workpiece and takes pictures by CCD.

Case Video

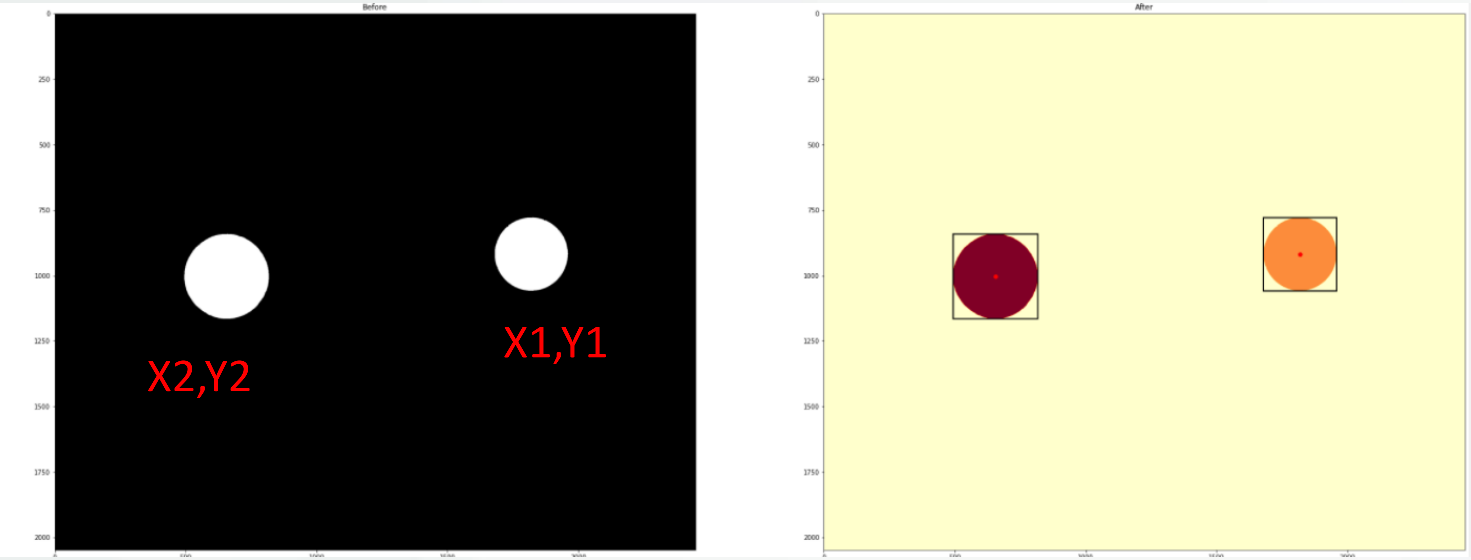
Results

Find the holes from the image and make marks

Modified labeling example

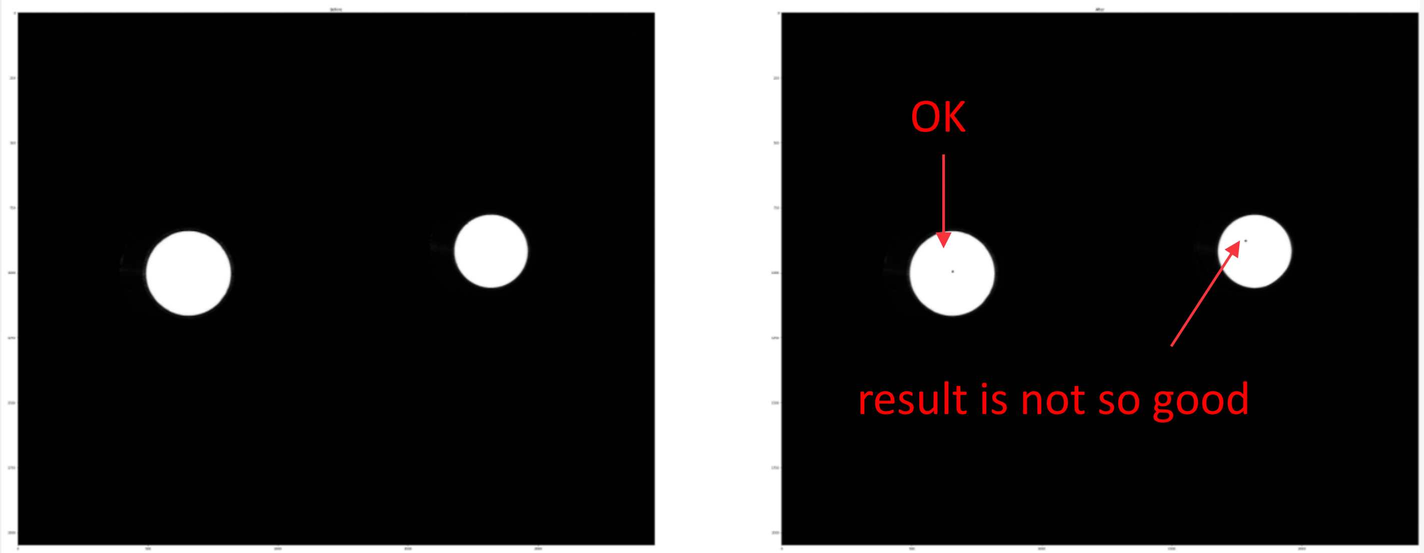
Hole coordinates

- X1,Y1=
  1818.38 917.78
- X2,Y2=
  655.21 1002.81



Modified Hough Circle example

```
circles = cv2.HoughCircles(img,
          cv2.HOUGH_GRADIENT,1,500,param1=8,
          param2=10,minRadius=180,maxRadius=330)
```



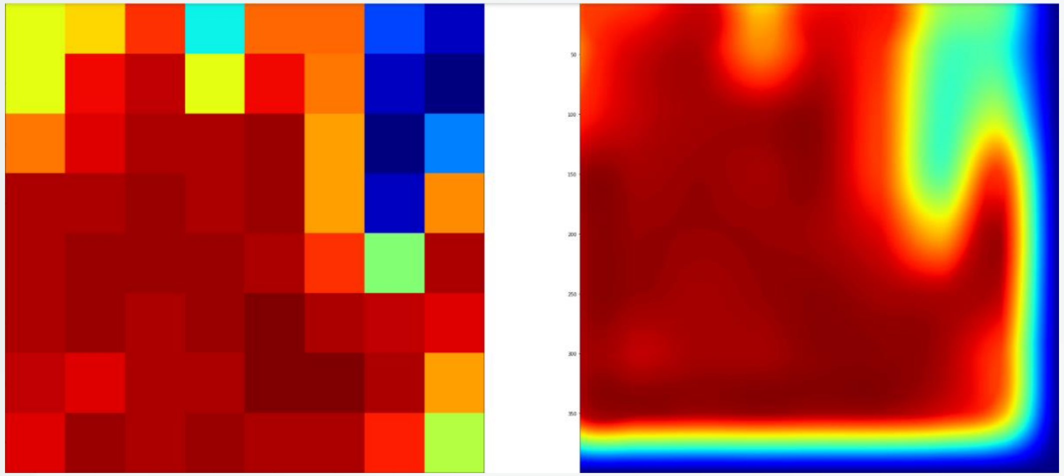
Summary

- MATLAB can process input data from 8x8 matrix to a figure.
- Bicubic algorithms let images more smoothly by fill in the vacancy of the matrix data.
- Modified labeling example to find the holes from the image and make marks.
- Modified Hough Circle example to find the holes from the image but the result is not so good so far.

THANK YOU！

Python Code


```python
# Opening the image and converting it to grayscale.
a = Image.open('/content/drive/MyDrive/DIP colab/temperature measurement_8_8.png').convert('L')
a = numpy.asarray(a)
a1 = Image.open('/content/drive/MyDrive/DIP colab/OUTPUT/temperature measurement_8_8output.png').convert('L')
a1 = numpy.asarray(a1)
print(a)
print(a1)


fig = plt.figure(figsize=(40, 40))
ax = fig.add_subplot(1, 2, 1)
imgplot = plt.imshow(a, cmap='jet')
ax.set_title('Before')
ax = fig.add_subplot(1, 2, 2)
imgplot = plt.imshow(a1,  cmap='jet')
ax.set_title('After')

fig, ax = plt.subplots(ncols=1,nrows=1,
     figsize=(6, 6))
# plots the label image on the
# previous plot using colormap
ax.imshow(a1, cmap='jet',vmin=150, vmax=255)
plt.savefig('/content/drive/MyDrive/DIP colab/OUTPUT/temperature measurement_8_8output2.png')
```



Python Code


```python
# Function: bicubic
# Author: YuYao
# Time: 10/09/2019

import numpy as np
import matplotlib.pyplot as plt
import cv2


def base_function(x, a=-0.5):
    # describe the base function sin(x)/x
    Wx = 0
    if np.abs(x)<=1:
        Wx = (a+2)*(np.abs(x)**3) - (a+3)*x**2 + 1
    elif 1<=np.abs(x)<=2:
        Wx = a*(np.abs(x)**3) - 5*a*(np.abs(x)**2) + 8*a*np.abs(x) - 4*a
    return Wx


def padding(img):
    h, w, c = img.shape
    print(img.shape)
    pad_image = np.zeros((h+4, w+4, c))
    pad_image[2:h+2, 2:w+2] = img
    return pad_image


def draw_function():
    a = -0.5
    x = np.linspace(-3.0, 3.0, 100)
    y = np.zeros(len(x))
    for i in range(len(x)):
        y[i] = base_function(x[i], a)
    plt.figure("base_function")
    plt.plot(x, y)
    plt.show()


def bicubic(img, sacle, a=-0.5):
    print("Doing bicubic")
    h, w, color = img.shape
    img = padding(img)
    nh = h*sacle
    nw = h*sacle
    new_img = np.zeros((nh, nw, color))

    for c in range(color):
        for i in range(nw):
            for j in range(nh):

                px = i/sacle + 2
                py = j/sacle + 2
                px_int = int(px)
                py_int = int(py)
                u = px - px_int
                v = py - py_int

                A = np.matrix([[base_function(u+1, a)], [base_function(u, a)], [base_function(u-1, a)], [base_function(u-2, a)]])
                C = np.matrix([base_function(v+1, a), base_function(v, a), base_function(v-1, a), base_function(v-2, a)])
                B = np.matrix([[img[py_int-1, px_int-1][c], img[py_int-1, px_int][c], img[py_int-1, px_int+1][c], img[py_int-1, px_int+2][c]],
                               [img[py_int, px_int-1][c], img[py_int, px_int][c], img[py_int, px_int+1][c], img[py_int, px_int+2][c]],
                               [img[py_int+1, px_int-1][c], img[py_int+1, px_int][c], img[py_int+1, px_int+1][c], img[py_int+1, px_int+2][c]],
                               [img[py_int+2, px_int-1][c], img[py_int+2, px_int][c], img[py_int+2, px_int+1][c], img[py_int+2, px_int+2][c]]])
                new_img[j, i][c] = np.dot(np.dot(C, B), A)
    return new_img

if __name__ == '__main__':

    sacle = 50
    path = "/content/drive/MyDrive/DIP colab/temperature measurement_8_8.png"
    img = cv2.imread(path)
    new_img = bicubic(img, sacle)
    cv2.imwrite( "/content/drive/MyDrive/DIP colab/OUTPUT/temperature measurement_8_8output.png", new_img)
    print("Finish")
```

Python Code
Modified labeling example

明志科技大學 MING CHI UNIVERSITY OF TECHNOLOGY

```
# Opening the image and converting it to grayscale.
a = Image.open('/content/drive/MyDrive/DIP colab/RCIM
D_SC_TEST1.jpg').convert('L')
# a is converted to an ndarray.
a = numpy.asarray(a)
# Threshold value is determined by
# using Otsu's method.
thresh = threshold_otsu(a)
print('thresh=',thresh)
# The pixels with intensity greater than
# "theshold" are kept.
b = a > 220

c1 = label(b)
# c is saved as label_output.png

# On the labelled image c, regionprops is performed
d1 = regionprops(c1)

fig = plt.figure(figsize=(40, 40))
ax = fig.add_subplot(1, 2, 1)
imgplot = plt.imshow(b, cmap='gray')
ax.set_title('Before')
ax = fig.add_subplot(1, 2, 2)
imgplot = plt.imshow(c1,  cmap='YlOrRd')
ax.set_title('After')
```

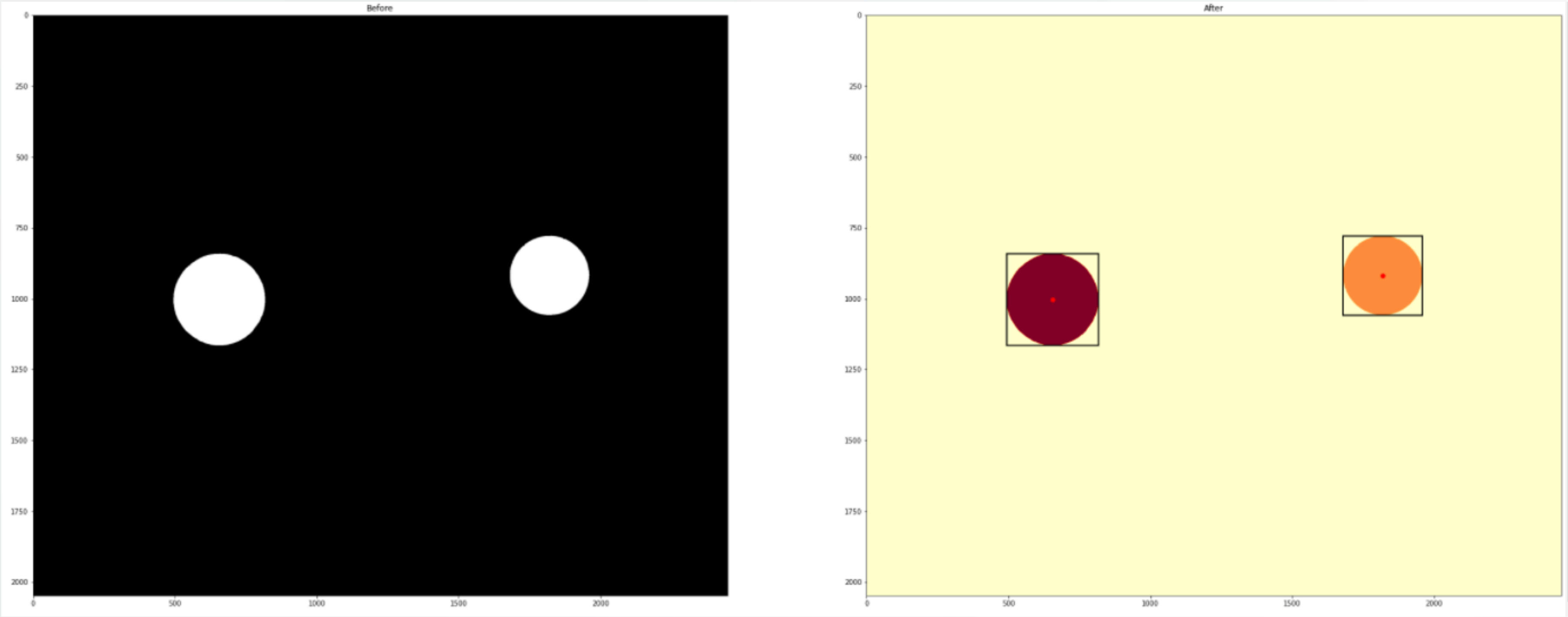
```
for i in d1:
    # Printing the x and y values of the
    # centroid where centroid[1] is the x value
    # and centroid[0] is the y value.
    #print(i.centroid[1], i.centroid[0])
    # Plot a red circle at the centroid, ro stands
    # for red.
    plt.plot(i.centroid[1],i.centroid[0],'ro')
    # In the bounding box, (lr,lc) are the
    # co-ordinates of the lower left corner and
    # (ur,uc) are the co-ordinates
    # of the top right corner.
    lr, lc, ur, uc = i.bbox
    # The width and the height of the bounding box
    # is computed.
    rec_width = uc - lc
    rec_height = ur - lr

    # Rectangular boxes with
  # origin at (lr,lc) are drawn.
    rect = mpatches.Rectangle((lc, lr),rec_width,
           rec_height,fill=False,edgecolor='black',
           linewidth=2)
    # This adds the rectangular boxes to the plot.
    ax.add_patch(rect)
print('DW=',rec_width)
print('DH=',rec_width)
#plt.savefig('/content/drive/MyDrive/DIP colab/OUTPUT/8888#.png')
plt.show()
```

Python Code

Modified Hough Circle example

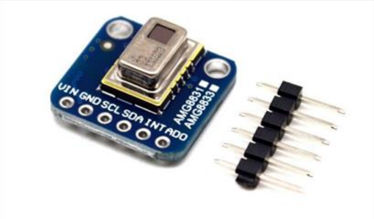
```
import numpy as np
import scipy.ndimage
from PIL import Image
import cv2
import matplotlib.pyplot as plt

# opening the image and converting it to g
rayscale
a = Image.open('/content/drive/MyDrive/DIP
 colab/RCIMD_SC_TEST1.jpg')
a = a.convert('L')
# Median filter is performed on the
# image to remove noise.
img = scipy.ndimage.filters.median_filter(
a,size=10)
# Circles are determined using
# Hough circles transform.
circles = cv2.HoughCircles(img,
          cv2.HOUGH_GRADIENT,1,500,param1=
8,
          param2=10,minRadius=180,maxRadiu
s=330)
```

```
# circles image is rounded to unsigned intege
r 16.
circles = np.uint16(np.around(circles))
# For each detected circle.
for i in circles[0,:]:
  # An outer circle is drawn for visualizatio
n.
    cv2.circle(img,(i[0],i[1]),i[2],(0,255,0)
,2)
  # its center is marked
    cv2.circle(img,(i[0],i[1]),2,(0,0,255),3)
# Saving the image as houghcircles_output.png
fig = plt.figure(figsize=(80, 80))
ax = fig.add_subplot(1, 2, 1)
imgplot = plt.imshow(a, cmap='gray')
ax.set_title('Before')
ax = fig.add_subplot(1, 2, 2)
imgplot = plt.imshow(img, cmap='gray')
ax.set_title('After')
#plt.savefig('/content/drive/MyDrive/DIP cola
b/OUTPUT/RCIMD_SC_TEST1_1_Hough Circle.png')
plt.show()
```


Before
After

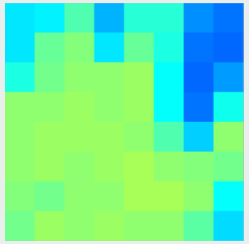
Results

Digital image process of Infrared Thermography



Data

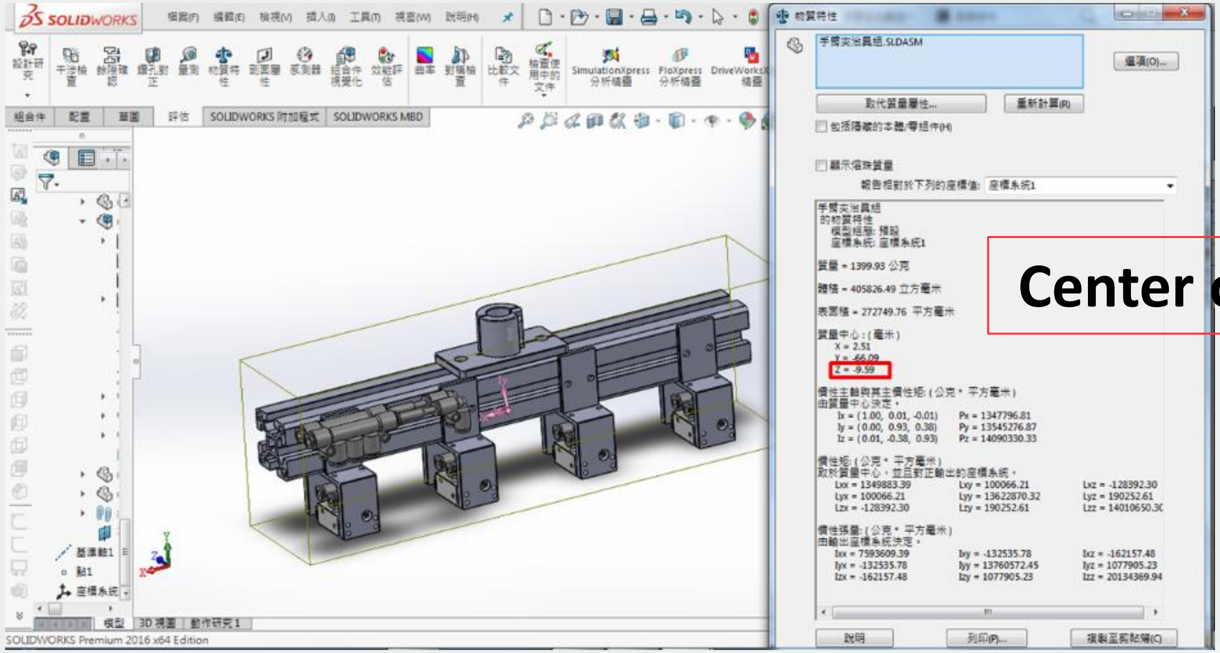
27	27.25	29	26	28.25	28.25	25.25	24.75
27	29.5	30	27	29.5	28	24.75	24.5
28	29.75	30.25	30.25	30.5	27.5	24.5	25.5
30.25	30.25	30.5	30.25	30.5	27.5	24.75	27.75
30.25	30.5	30.5	30.5	30.25	29	26.5	30.25
30.25	30.5	30.25	30.5	30.75	30.25	30	29.75
30	29.75	30.25	30.25	30.75	30.75	30.25	27.5
29.75	30.5	30.25	30.5	30.25	30.25	29.25	26.75



image

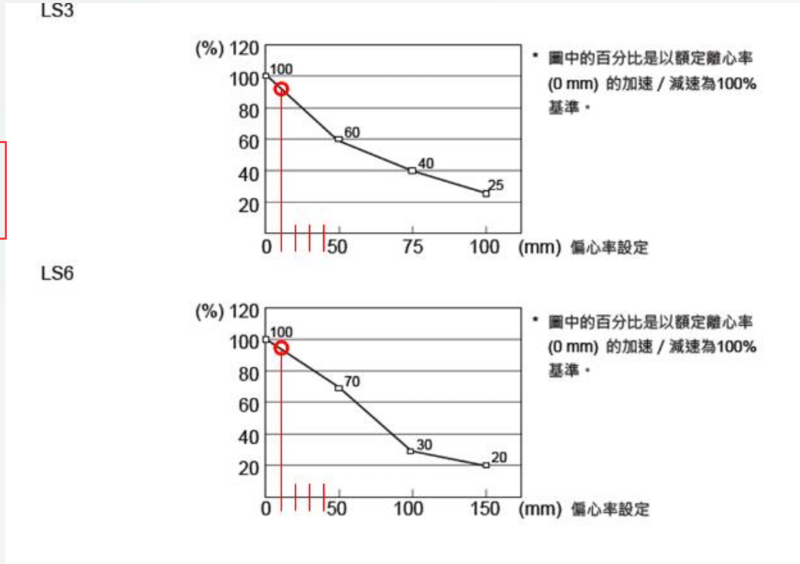
```
[[165 172 193 134 186 186 113 97]
 [165 198 203 165 198 184 97 90]
 [184 200 205 205 207 179 90 120]
 [205 205 207 205 207 179 97 181]
 [205 207 207 207 205 193 151 205]
 [205 207 205 207 210 205 203 200]
 [203 200 205 205 210 210 205 179]
 [200 207 205 207 205 205 195 158]]
```

End controller and robot inertia calculation



Center of mass

Rated eccentricity setting

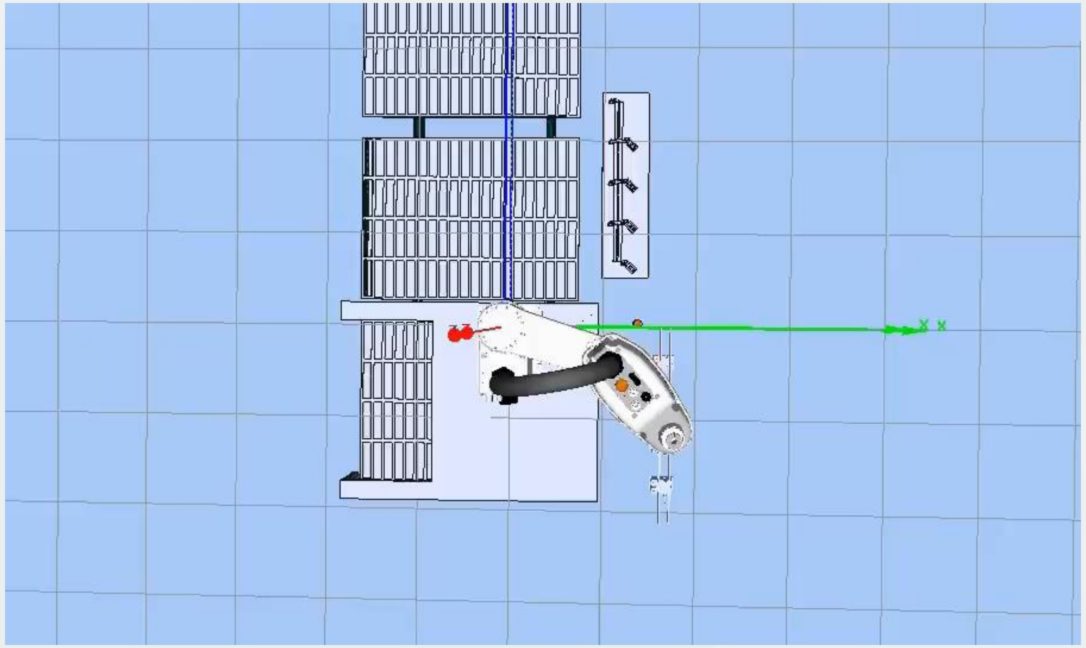


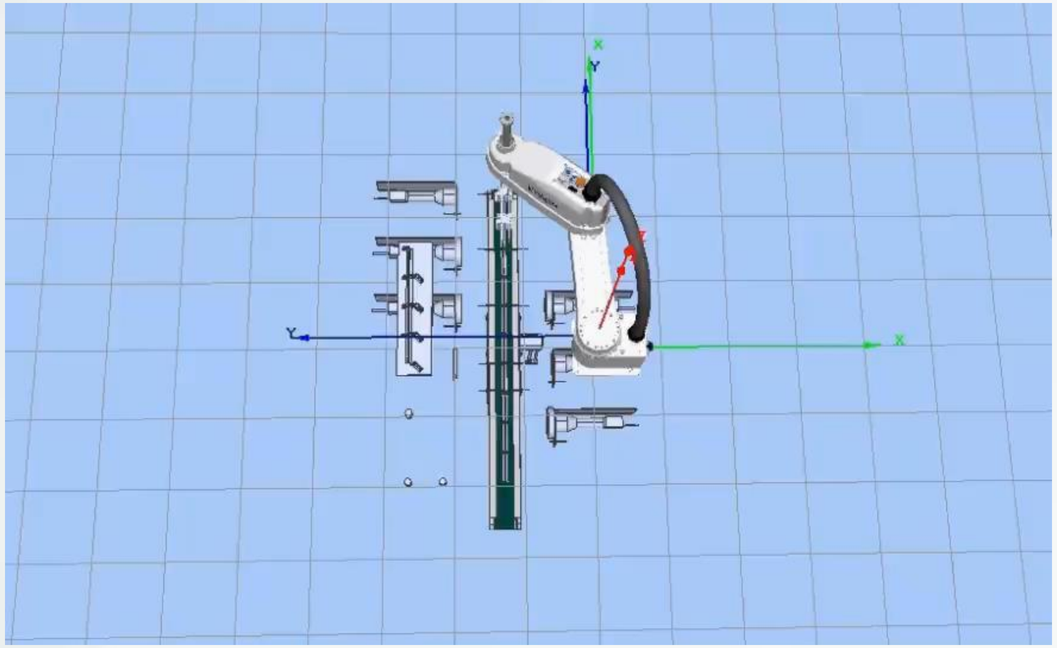
EPSON Robot LS3 Setup				
項次	項目	參數	單位	備註
1	重量	1.4	kg(K)	
2	長度	172	mm	
3	慣性	0.01	kg-cm^2	
4	離心率	9.6	mm	

EPSON Robot LS6 Setup				
項次	項目	參數	單位	備註
1	重量	1.6	kg(K)	
2	長度	172	mm	
3	慣性	0.01	kg-cm^2	
4	離心率	9.6	mm	

Inertia calculation

Robot simulation





- ➢ **Confirm moving range**
- ➢ **Avoid singularity points**