COMPGX03 - Crack Detection

Qi Wang
ucabqw3@ucl.ac.uk

January 2018

1 Abstract

Cracks are an important indicator reflecting the safety status of infrastructures. A crack is the separation of an object or material into two or more pieces under the action of stress. The fracture of a solid usually occurs due to the development of certain displacement discontinuity surfaces within the solid. If a displacement develops perpendicular to the surface of displacement, it is called a normal tensile crack or simply a crack; if a displacement develops tangentially to the surface of displacement, it is called a shear crack, slip band, or dislocation. The touch probe kit can be used as a crack detector, it can scan the surface of the object and collect the geometrical data from the surface of the object. The collected data can then be used to construct digital three-dimensional models. There are a variety of technologies for digitally acquiring the shape of a 3D object, this paper will investigate different contact approaches to probe the subject through physical touch and conclude the best approach to create GIS (geographic information system) maps.

2 Introduction

This project will propose a contacted approach to detect the crack on the surface of the object.

The basic tool of the a touch-probe kit with the capacity to force sensing and actuation functionality. The chip of the controller is Raspberry pi 3, there is a force sensing resistor connected to the chip through ADC. The force sensing resistor will exhibit a decrease in the resistance with the increase in force applied to the surface of the sensor. Then the change of the resistance will affect the output voltage, ADC takes the voltage as the input and then convert it into the digital number from 0 to 1000. The aim of this force sensing is to show that how much force is applied to the end-effector actuator. The actuator consists of x-y-z-axis servos and the force sensing resistor is attached to the z-axis. whenever z-axis accesses to the surface of the plane, the force sensor can detect the pressure and return the force of the pressure. The aim of this project is to investigate software-based ways of achieving this surface scanning and thus detect the crack. Different approaches and varying control parameters of system will produce different experiment results. After illustrating each result, we will evaluate the results and choose the best set of parameters and approach. The details of each part of the kit will be given in the next chapters.

3 Hypothesis

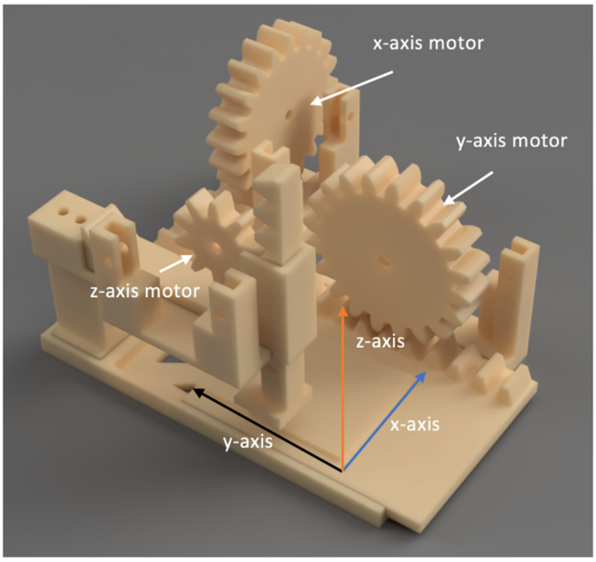

Figure 1: touch-probe kit

As a contacted approach to detect the crack, the accuracy and the display of the experimental result is very essential. In this experiment, we use Force sensor resistor as our tool to acquire the experiment data, the force data, which can be viewed as a discrete data sample. The geometrical information of a surface contained the crack is a 3D data set. The horizontal position information can be represented as the mesh grid. Its vertical information, the height of the surface in each x-y position can be represented as the force data. To be more specific, the FSR(force sensor resistor) is attached on the z-axis of the kit. When the servo drives the z-axis down to a fixed height, it will engage with the surface of the object, then FSR will sense the force of pressure between itself and the surface. The force of the pressure can stand for the height of the surface, the crack can be found where there is a sudden drop of the height right after a sharp raise of the height change.

Different approaches to set up the experiment:

a. Poking the surface of the object:

   The principle of poking the surface is like a sewing machine, the needle of the sewing machine is the FSR. The surface of the object is divided into a $14 * 16$ (low resolution) or $34 * 39$ (high resolution) mesh grid. Then the 'needle' will poke each point of intersection of the mesh grid. Every time the FSR will return the force of the pressure on the object surface once the 'needle' poke the surface. The 'needle' has to rise and move to the next point in order to poke the point. So the basic logic of poking reflected in the software implementation contains two nested for loops, the exterior for loop choose the next x-axis value and the interior for loop choose the next y-axis value, once the end effector arrives that point, z-axis will drop down to poke the surface. That is the working principle of poking.

b. Drafting the surface of the object:

The principle of drafting FSR is different from Poking. Drafting, just as its name implies that the FSR will be drafted all the way from initial position to the end point without any rising or dropping. The surface of the object is also divided into a $14*16$ (low resolution) or $34*39$ (high resolution) mesh grid.The resolution of this approach raises noticeably comparing to that of poking. However, the result of this approach is unoptimistic due to the existing inevitable fraction between FSR and the surface of the object.

4 Methods

1. a. Equipment and components used:

   The working platform of the crack detection is displayed in the figure 1. The End effector is installed on the end of the z-axis. There are three servos that drive the gears and move the x-y-z axis. The x and y axis move horizontally, only z axis moves vertically.

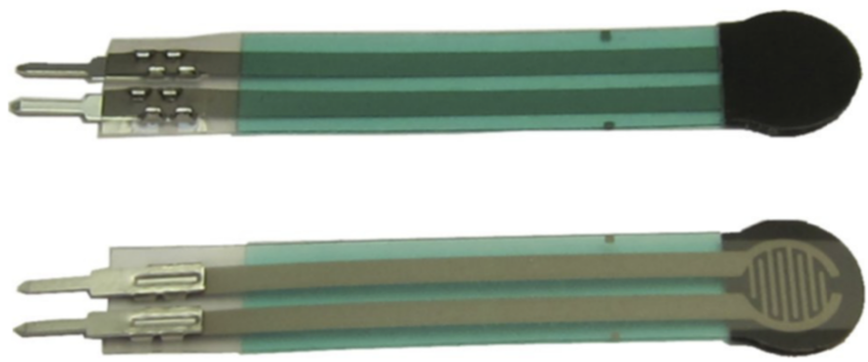

Figure 2: FSR 400

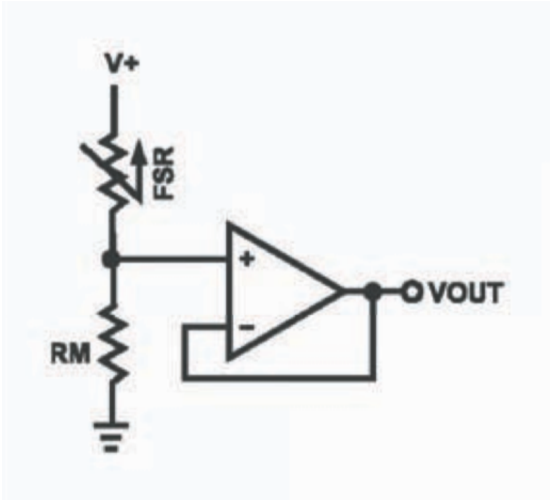

Figure 3: FSR 400 schematic

FSR 400 are two-wire device with a resistance that depends on the applied force. From the Figure 3, we can derive the output voltage when the force is applied on the FSR.

$$V_{out} = \frac{R_M * V+}{R_M + R_{FSR}}$$

As $V+$ is connected on the $V_{DD}(3.3V)$ of the Raspberry Pi, so $MAX(V_{OUT}) = 3.3V$. Figure 4 is captured on the FSR document and this illustrates how the applied force changes the output voltage. One issue needs to be noticed that the $V+$ applied on FSR is 5V, but in practise $V+$ applied on our FSR is only 3.3V, in another word, we need to remap the output voltage to force convention based on the reality.

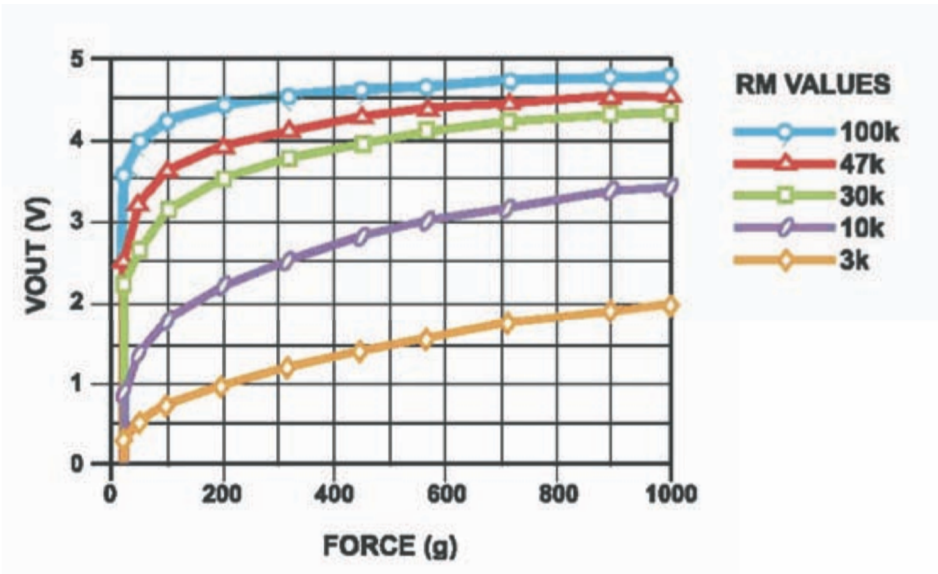

Figure 4: Force to voltage convention

So we sample the green curve in the figure 4 and multiply $\frac{3.3}{5}$ to the y-axis value and keep x-axis value unchanged. The reason why the green curve is being chose is the resistance of the resistor in the component of our kit is $30k\Omega$.

b. Mapping function:

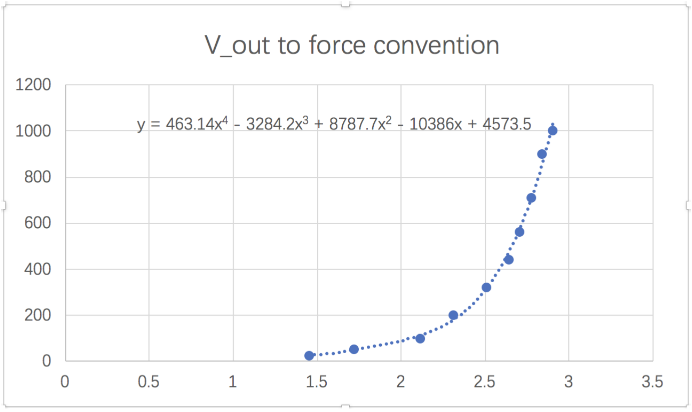

Figure 5: Voltage to force convention

So in my mappingFxn function, the polynomial is to convert the V_{out} to the readable force information. The highest order of the polynomial is 4, it is the most suitable order that can fit the data, the littler order will be under-fitting, while the higher order will be over-fitting. But the data from the Figure 4 is insufficient, so the curve that we simulated is uncompleted, therefore, we use the linear function to fit the curve from x=0 to x=1.4. The input opf the mapping functoin is the data of the Message returned the FSR and go through by the ADC, it is the force information. However, converting the force to V_{OUT} is the first step, then we fit the derived V_{out} into the polynomial in order to get the force data. Additionally, the coefficients of the polynomial are shrunk by 100 times as $10N = 1KG$.

```
function y=mappingFxn(x)
  x = x * 3.3 / 1000;
    if x >= 1.45
        y = 4.6314*x^4 - 32.842*x^3 + 87.877*x^2 - 103.86*x + 45.735;
    else
        y = 0.172 * x;
end
```

Figure 6: mappingFxn

c. Actuation Functionality:

CONTINUED

TowerPro MG90 - Micro Servo

Specifications

Modulation:	Analog
Torque:	**4.8V:** 30.60 oz-in (2.20 kg-cm) **6.0V:** 34.70 oz-in (2.50 kg-cm)
Speed:	**4.8V:** 0.11 sec/60° **6.0V:** 0.10 sec/60°
Weight:	0.49 oz (14.0 g)
Dimensions:	**Length:** 0.91 in (23.1 mm) **Width:** 0.48 in (12.2 mm) **Height:** 1.14 in (29.0 mm)
Motor Type:	(add)
Gear Type:	Metal
Rotation/Support:	Dual Bearings
Rotational Range:	180°
Pulse Cycle:	20 ms
Pulse Width:	400-2400 µs
Connector Type:	(add)

Brand:	Tower pro
Product Number:	(add)
Typical Price:	(add)
Compare:	add+

Figure 7: servo specification

The frequency of the PWM of MG90 is 50Hz. As it is shown in the specification of this micro servo, the pulse cycle is 20ms, so one period of the whole duty is 20ms, it can be implemented 50 times within 1 sec. Therefore, we set 50Hz as the frequency in python file.

The working principal of the servo is that the length of the pulse width represents the amount of rotation. So larger PWM represents larger amount of rotation.The gear has 20 teeth representing 360 degrees. So each teeth represents 18 degrees. So the degree of rotation requires in order to move one cm can be calculated by:

$$a = \frac{360}{4.5 * \pi}$$

The radius of the X-Y gear 2.25 cm, so the circumference of the gear is then 4.5π.

The conclusion of testing gear for several times is that the gear will rotate o degree when the duty cycle is 2.5, the gear will rotate 180 degree when duty cycle is set to be 12.5. So the mapping between duty cycle and gear rotation is:

$$\frac{12.5 - 2.5}{180 - 0}$$

which means that the gear rotate 1 degree when the duty cycle is $b = \frac{1}{18}$.

So how many duty cycle required to move one cm is:

$$ratio = a * b = 1.4147$$

The mapping function between duty cycle and X-Y coordinator is :

$$pw_x = -1.4147 * x + 12.5;$$
$$pw_y = 1.4147 * y + 5;$$

The work space of kit is $0 < x < 3, 0 < y < 4$. The kit will collide when the input x and y are beyond the work space.

2. Algorithm:

a. The following algorithm is for poking in high resolution

Table 1: algorithm for poking in high resolution

```
Algorithm: poke in high resolution
---------------------------------
x ← 0, y ← 0, step ← 0.1
repeat
    repeat
        move x and y, drop z
        pause 0.4 sec
        read and store FSR data
        pause 0.3 sec
        keep x and y, raise z
        y ← y + step
    until y ← 4
    x ← x + step
until x ← 3
end
```

b. The following algorithm is for drafting in high resolution

Table 2: algorithm for drafting in high resolution

```
Algorithm: draft in high resolution
x ← 0, y ← 0, step ← 0.1
repeat
    repeat
        move x and y, drop z
        pause 0.4 sec
        read and store FSR data
        pause 0.3 sec
        y ← y + step
    until y ← 4
    x ← x + step
until x ← 3
end
```

As can be seen that the only difference between poking and draft is that drafting algorithm has no need to raise the z-axis while for the poking algorithm it is the opposited case. Additionally, the surface of the object is divided into $34 * 39$ (high resolution) mesh grid, in the low resolution mode, the surface is then divided into a $14 * 16$ mesh grid.

The visualization is another important issue. The above figures show that the surface of the object is very rough, however, it does not distinguish the difference between camber concave and the crack. The solution to this problem would be set the threshold to decide whether it is a crack or not. For instance, if the force data is above 1.2N , then this position is a part of normal surface, however, if the data is below 1.2N, this position or this point is probably a part of crack.

In order to visualize the result in a better way, noise in the image would better be eliminated or improved. However, a true image should be in RGB form, there should be three color channel in the z-axis as the dataset. Therefore, the image we get from the output of our kit is not truly a image, since the data contained in the z axis is the force of pressure. Nevertheless, we can still use that force information and treat it like a image. As can be seen from the figure above, there are so much noise that needed to be handled, the surface of the object should be smooth, however, due to the limitation of the equipment, the output of our kit is a surface that are very rough, image processing techniques become the main issue to be solved.

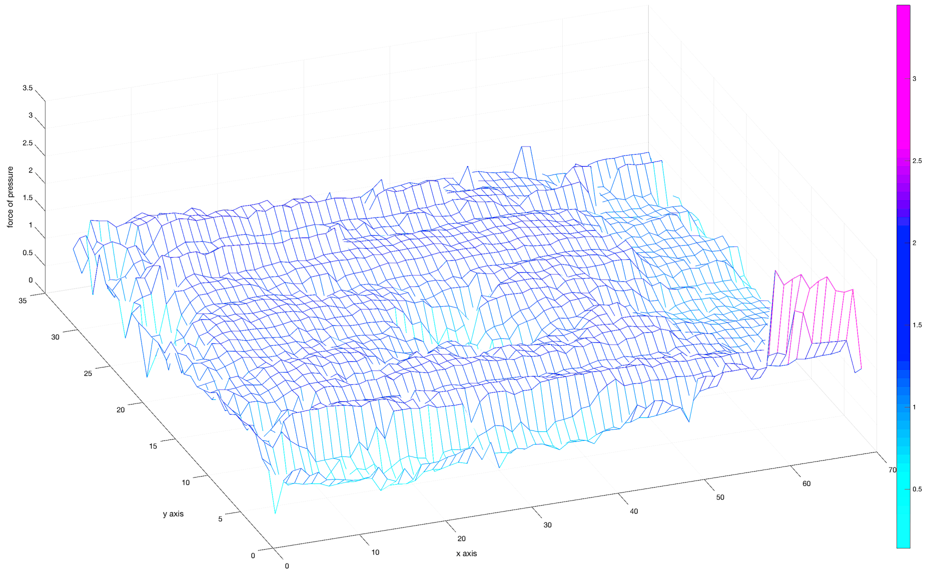

Figure 8: original image

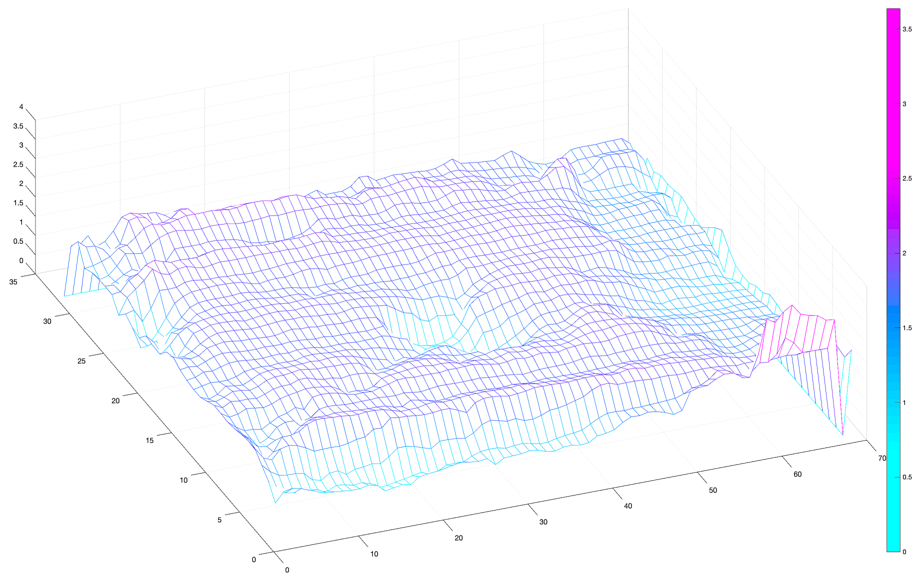

Figure 9: smooth image

The original images have so many useless details, especial when in the high resolution. as can be seen from the original image, figure 8. The original image has many needle-like peaks and valleys, which is caused by isolated points with high or low force data, this is disturbing. Therefore i design an averaging filter which will be effective to smooth the original image. The image can be split into many equal-sized windows, the size and the shape of the window vary in terms of the resolution of the images. The resolution of my output image is 32*69. Due to the limitation of the equipment, this resolution is relatively

CONTINUED

high. One issue needs to be noticed is that there is a trade off between high resolution and the experimental duration. So the shape of my filter is a square and the window size is 4 points to be handled once a time. So after the original image goes through the averaging filter, it conducts the smooth image, figure 9. The algorithm of the averaging filter is demonstrated as follows:

$$g(x) = \frac{1}{N_D} \sum_{X_n \in D} g(x_n)[1]$$

where N_D is the pixel number in region D and X_n are all the pixels in D. D can be square or circular, and its window size can be adjusted according to the resolution of the image. A too large window size of D will eliminate the details of cracks and a too small window size of D is not able to effectively smooth the original image. Therefore, the window size of the average filter should be initialized properly.

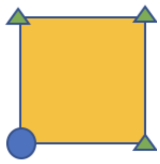

Figure 10: window of the filter

In the averaging filter algorithm, the weight of each point should be taken into consideration. Figure 10 shows the window of the averaging filter, the blue circle indicates the point which will go through the filter, while the other three green triangles in the concerns indicates the nearby additive points which will have influence on the current point. Different points will be assigned to different weights. According to the Cartesian distance, the weight on the current point should be the biggest, the weights on two nearby points should be intermediate, while the point in the direction of diagonal line should be assigned to the least weight.

5 Results

1. Comparison between drafting and poking:
   Figure 11 display the 3D map of object(poking in high resolution)

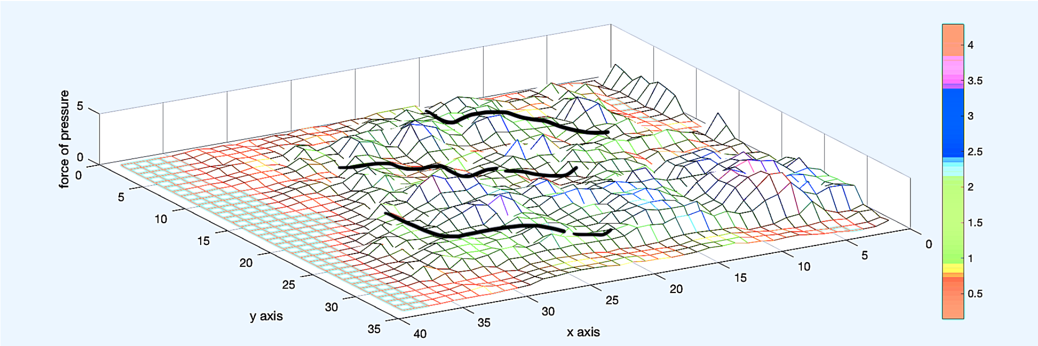

Figure 11: poking in high resolution

As can be seen that, the surface of the object has been displayed in the mesh grid, although there are some protuberance on the top right corner which is already existed on the object and is inevitable. to be more specific, the green mesh indicates that the height of the surface is relatively low due to the low force of pressure between FSR and the surface of the object and vice versa. So this approach, poking in high resolution, is relatively acceptable. There are three crack curves in figure 8. As the surface of the object is not smooth, so it is hard to distinguish between the crack and camber concave. However, camber concave can be seen as the crack in some ways.

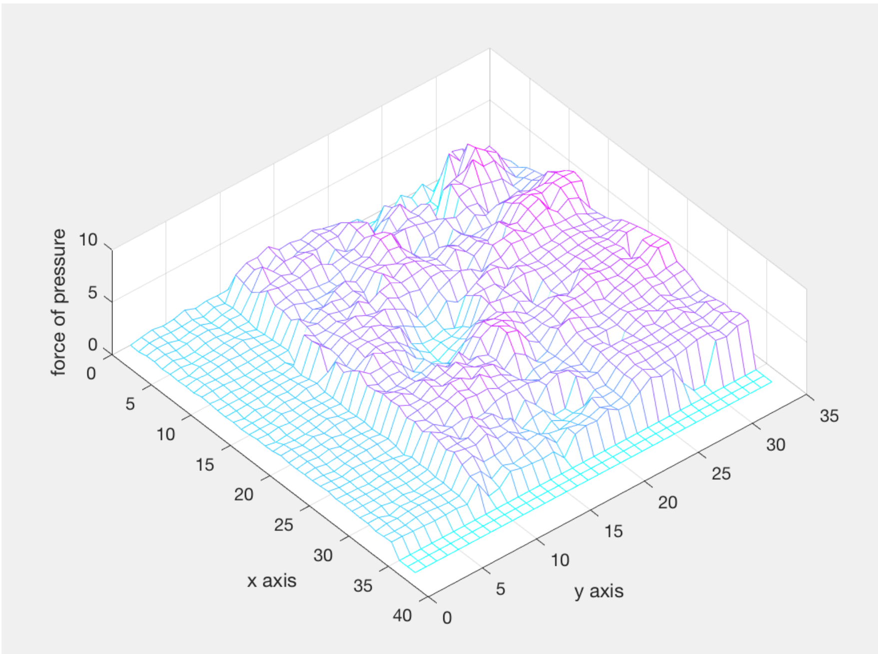

Figure 12: draft in high resolution

The result of drafting in high resolution is unacceptable. We can barely see that there are some protuberances and camber concave in figure 12, but no specific shape or crack can be recognized. As a result of inevitable friction between the object and the FSR is so large that the measurement is inaccurate, especially when the material of attachment which is attached on the FSR is rubber. Moreover, the value which returned by FSR in that point is sometimes not matched in that point due to the friction. The shape of the protuberance will have some

CONTINUED

influence on the result of the measurement. If it is a sharp protuberance, for instance, the angle between the side of the protuberance and surface is 90 degrees or less than 90 degrees, the friction will be so large and there will be a delay during the measurement. Because the movement of FSR in that point will be slow and this is the main reason why we can not recognize any shape in the figure 12.

In conclusion, poking the FSR is better than drafting in terms of crack detection.

2. Comparison between high resolution and low resolution

Due to the non-ideal resulut of drafting approach, so this comparison will use the result of poking.

Figure 13: rubber

The resolution can be different by two approaches, this first one is to change the number of the mesh grid. The second one is to polish the shape of the sensor, there is a rubber attached on the sensor, the shape of the rubber is like a hemisphere. This shape can guarantee uniform strength distributed on the FSR, however, this is not the case when high resolution is the priority. The resolution can be upgraded when the shape of the rubber is like a needle. Figure 14 below illustrate how the performance of the detection is improved when the shape of the rubber is change into a needle.

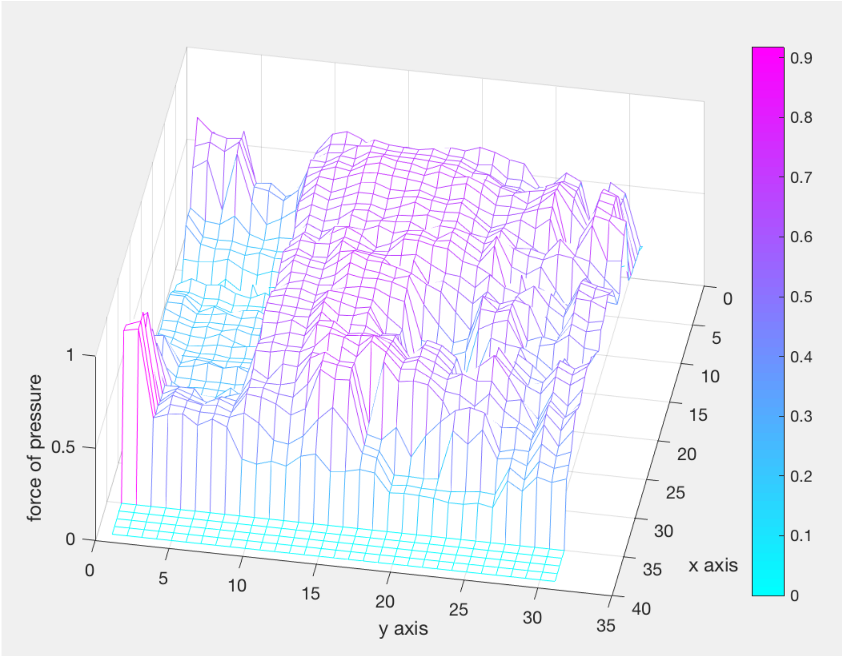

Figure 14: drafting when the shape of the rubber is changed

As we can see very clearly, the crack on the surface of the object has been detected, although the approach of detection is drafting.

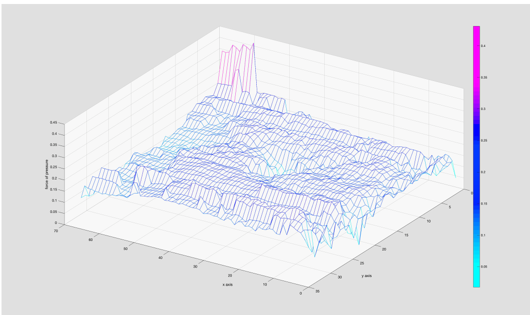

Figure 15: poking in low resolution

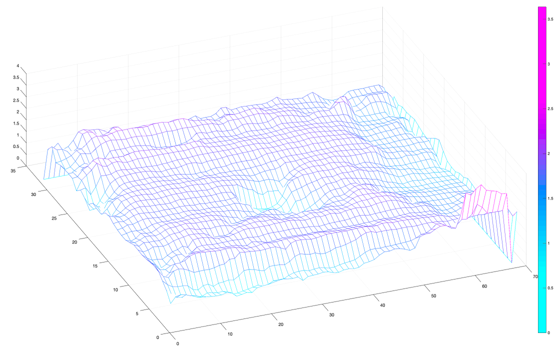

Figure 16: poking in high resolution

As can be seen that figure 15 and 16 are the result of poking. Both of these two result are non-ideal, but we can still recognize the crack on the object. There is a trade of between higher resolution and the sensitivity to noise. Although with some noise in the figure 11, it is more recognizable comparing to poking in low resolution.

CONTINUED

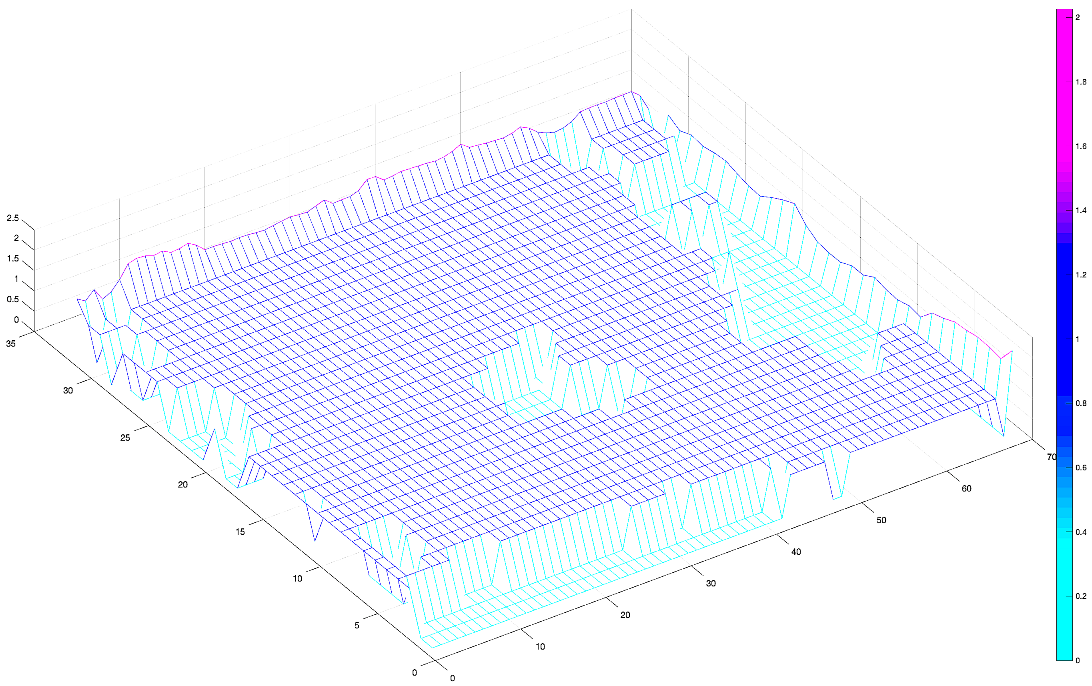

Figure 17: modified image

Figure 17 is the modified image, as can be seen, all the crack on the object surface has been showed very clearly after each value of the point has been classified by a threshold classifier. The height of the crack is set to 0, while the other part of the surface has its original value. This is the final result of our experiment.

6 Discussion

Sometimes high resolution does not equal to better results, high resolution is susceptible to the impact of the noise. The noise can be arisen by the friction between the rubber and the surface. When drafting approach is applied, the friction will cause a serious problem, the delay. The value of force which should belong to the last point would have impact on the current measurement. To be more specifically, the large friction will slow down the movement of servo when the surface of rubber is attached on the surface of the object firmly.

In addition, The location of the crack can not only be found but also be localized. As the surface of the object has been divided by mesh grid, So it is very easy to localize the location of the crack and the size and the shape of the crack, which is very essential to evaluate the degree of the damage and thus take the corresponding remedy measures.

7 Conclusion

This paper presents a crack detection and classification approach for surface the solid material.A detailed description of the image processing techniques and the optimal parameter settings are given in the experimental section. The proposed approach is easy to implement and effective. The proposed image processing technique for crack detection and classification may be suitable for other state monitoring applications, but this approach is only valid for contacted crack detection. Future work may include the following: developing a crack mosaic algorithm to connect the discontinuous crack parts and then delete the preserved crack like objects.

References

[1] Wenyu Zhang, Zhenjiang Zhang, Dapeng Qi, and Yun Liu. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors*, 14(10):19307–19328, 2014.

END OF COURSEWORK