机器视觉单层NCC模板匹配

SZ170410221-朱方程

Code

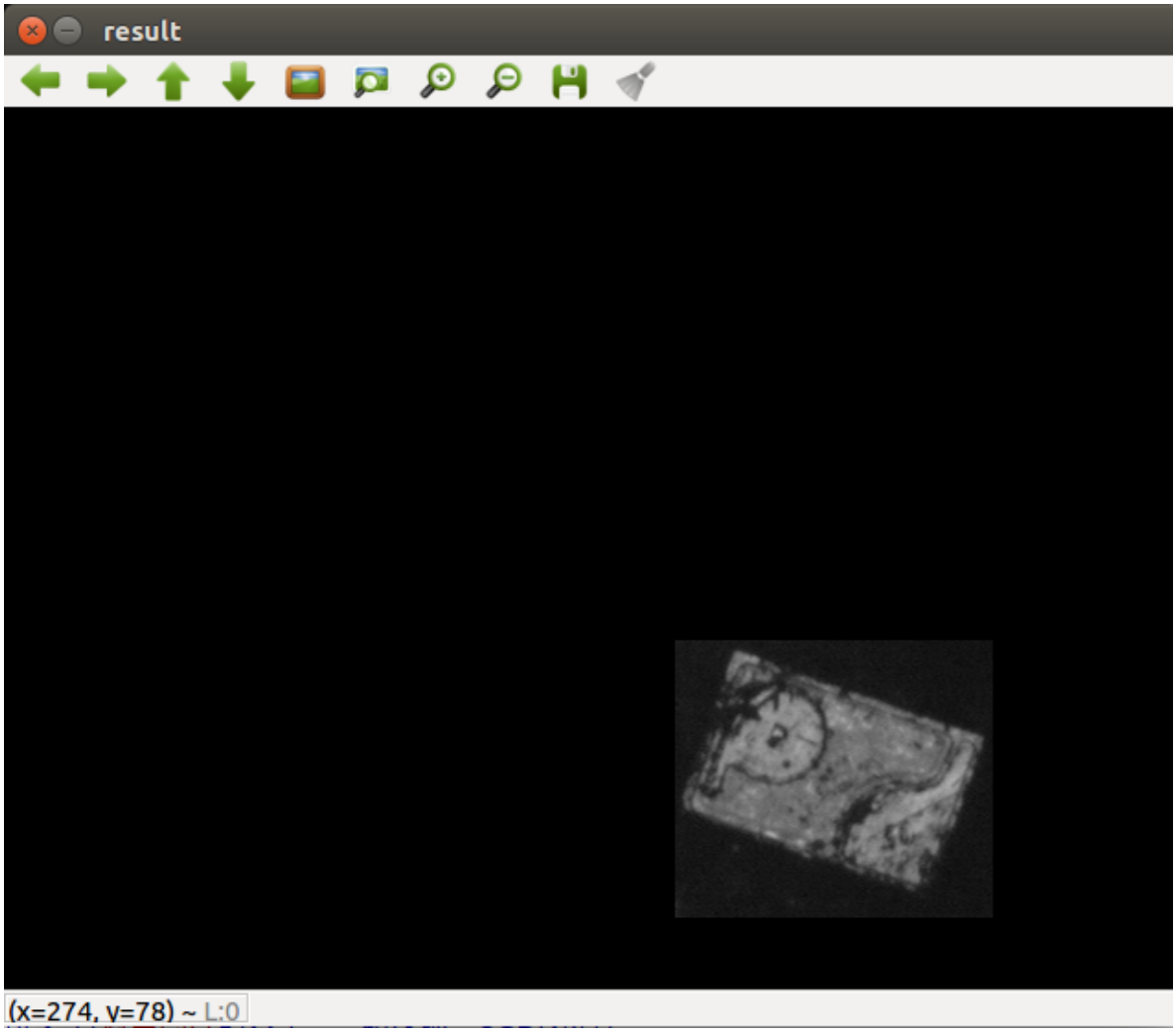
```
//运行平台Ubuntu16.04
#include <stdlib.h>
#include <cv.h>
#include <math.h>
#include <highgui.h>
#include <opencv2/opencv.hpp>
#include <opencv2/core/core.hpp>
#include "ros/ros.h"
#include "std_msgs/String.h"
#include "std_msgs/Bool.h"
#include "std_msgs/Float32.h"
#include <geometry_msgs/Twist.h>
#include "sensor_msgs/Image.h"
#define LINEAR_X 0
using namespace cv;
using namespace std;

//机器视觉作业: 单层NCC模板匹配
int main(int argc, char **argv)
{

    VideoCapture capture;
        capture.open(0);//打开 zed 相机

    ROS_WARN("*****START");
    ros::init(argc,argv,"trafficLaneTrack");
        ros::NodeHandle n;

    if (!capture.isOpened())
    {
        printf("摄像头没有正常打开\n");
        return 0;
    }
    waitKey(100);
    Mat frame;
    while (ros::ok())
    {
        clock_t begin,finish;
        begin=clock();
        Mat Temp = imread("/home/fangcheng/Library/pattern.bmp",0);
        Mat Img = imread("/home/fangcheng/Library/IMAGEB34.bmp",0);
        imshow("Temp",Temp);
        imshow("Img",Img);
        double mt=0.0000;
        double st_square=0.0000;
        double sum_mt=0.0000;
```

```
        double sum_st=0.0000;
        int n=Temp.rows*Temp.cols;
        //模板平均灰度
        for(int i=0;i<Temp.rows;i++){
            for(int j=0;j<Temp.cols;j++){
                sum_mt+=Temp.at<uchar>(i,j);
            }
        }
        mt=(double)sum_mt/n;
        //模板方差
        for(int i=0;i<Temp.rows;i++){
            for(int j=0;j<Temp.cols;j++){
                sum_st+=(Temp.at<uchar>(i,j)-mt)*(Temp.at<uchar>(i,j)-mt);
            }
        }
        st_square=(double)sum_st/n;

        float NCC[Img.rows][Img.cols]={0.0};
        double mf = 0.0;
        double sf_square = 0.0;
        double sum_mf = 0.000000;
        double sum_sf = 0.000000;
        double sum = 0.000000;
        float max_NCC=0.00;
        int X=0;//左上角点横坐标
        int Y=0;//左上角点纵坐标
        for(int i=0;i<Img.rows-Temp.rows+1;i++){
            for(int j=0;j<Img.cols-Temp.cols+1;j++){
                sum_mf = 0.000000;
                for (int k = 0; k <Temp.rows; k++){
                    for (int l = 0; l < Temp.cols; l++){
                        if((i+k) <Img.rows && (j+l)<Img.cols){
                            sum_mf+=Img.at<uchar>(i+k,j+l);
                        }
                    }
                }
                mf=sum_mf/n;

                sum_sf = 0.000000;
                for (int k = 0; k <Temp.rows; k++){
                    for (int l = 0; l < Temp.cols; l++){
                        if((i+k)<Img.rows && (j+l)<Img.cols)
                        sum_sf+=(Img.at<uchar>(i+k,j+l)-mf)*(Img.at<uchar>
(i+k,j+l)-mf);
                    }
                }
                sf_square=sum_sf/n;

                sum = 0.000000;
                for (int k = 0; k <Temp.rows; k++){
                    for (int l = 0; l < Temp.cols; l++){
                        if((i+k)<Img.rows && (j+l)<Img.cols)
                        sum+=((Temp.at<uchar>(k,l)-mt)/sqrt(st_square))*
((Img.at<uchar>(i+k,j+l)-mf)/sqrt(sf_square));
                    }
                }
                NCC[i][j]=(float)sum/n;
                if (abs(NCC[i][j])>max_NCC)//记录NCC最大值及相应的位置
```

```
                {
                    max_NCC=NCC[i][j];
                    X=i;
                    Y=j;
                }
            }
        }
        printf("最大NCC绝对值=%f\n\n",max_NCC);
        Mat result = Mat::zeros(Img.rows, Img.cols,CV_8U);
        for (int k = 0; k <Temp.rows; k++){
            for (int l = 0; l < Temp.cols; l++){
                if((X+k) <Img.rows && (Y+l)<Img.cols){
                    result.at<uchar>(X+k,Y+l)=Img.at<uchar>(X+k,Y+l);
                }
            }
        }
        imshow("result",result);
        int coordinate[]={0,0};
        coordinate[0]=X+Temp.rows/2;
        coordinate[1]=Y+Temp.cols/2;
        /*
        //坐标计算
        int start_position[]={0,0};
        int end_position[]={0,0};
        for (int i = 0;i < result.rows;i++)
        {
            for (int j = 0;j < result.cols;j++)
            {
                if (result.at<uchar>(i , j )>0){
                    if (start_position[0]==0 && start_position[1]==0){
                        start_position[0]=i;
                        start_position[1]=j;
                    }
                    end_position[0]=(i>end_position[0])?i:end_position[0];
                    end_position[1]=(j>end_position[1])?j:end_position[1];
                }
            }
        }
        int coordinate[]={0,0};
        coordinate[0]=(start_position[0]+end_position[0])/2;
        coordinate[1]=(start_position[1]+end_position[1])/2;
        */
        printf("中心坐标= (%d,%d)\n\n", coordinate[0], coordinate[1]);
        finish=clock();
        printf("耗时%fs\n\n\n\n\n\n\n",(double)(finish-
begin)/CLOCKS_PER_SEC);
        ros::spinOnce();
        waitKey(10000);
    }
    return 0;
}
```

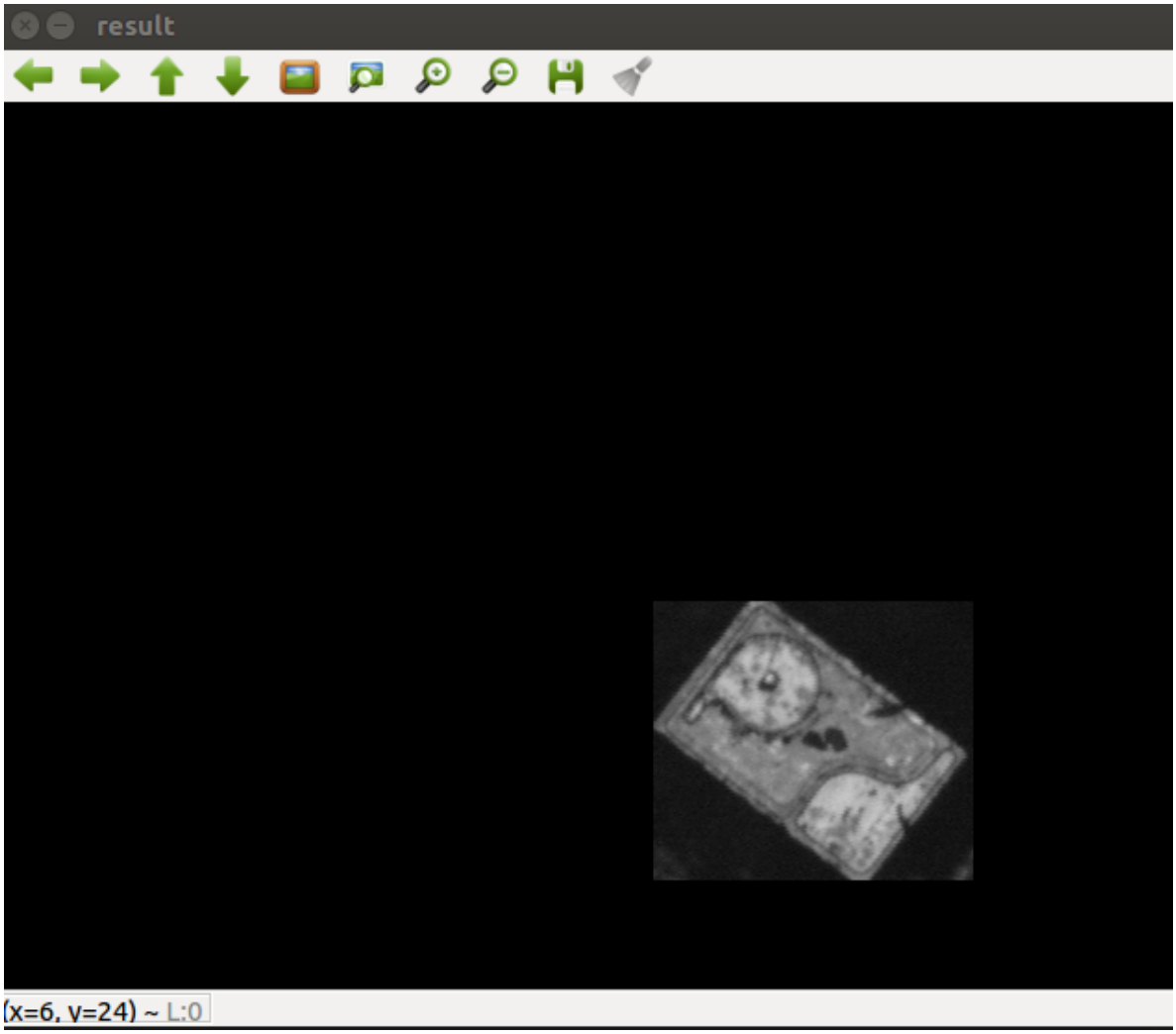
匹配结果

IMAGEB15



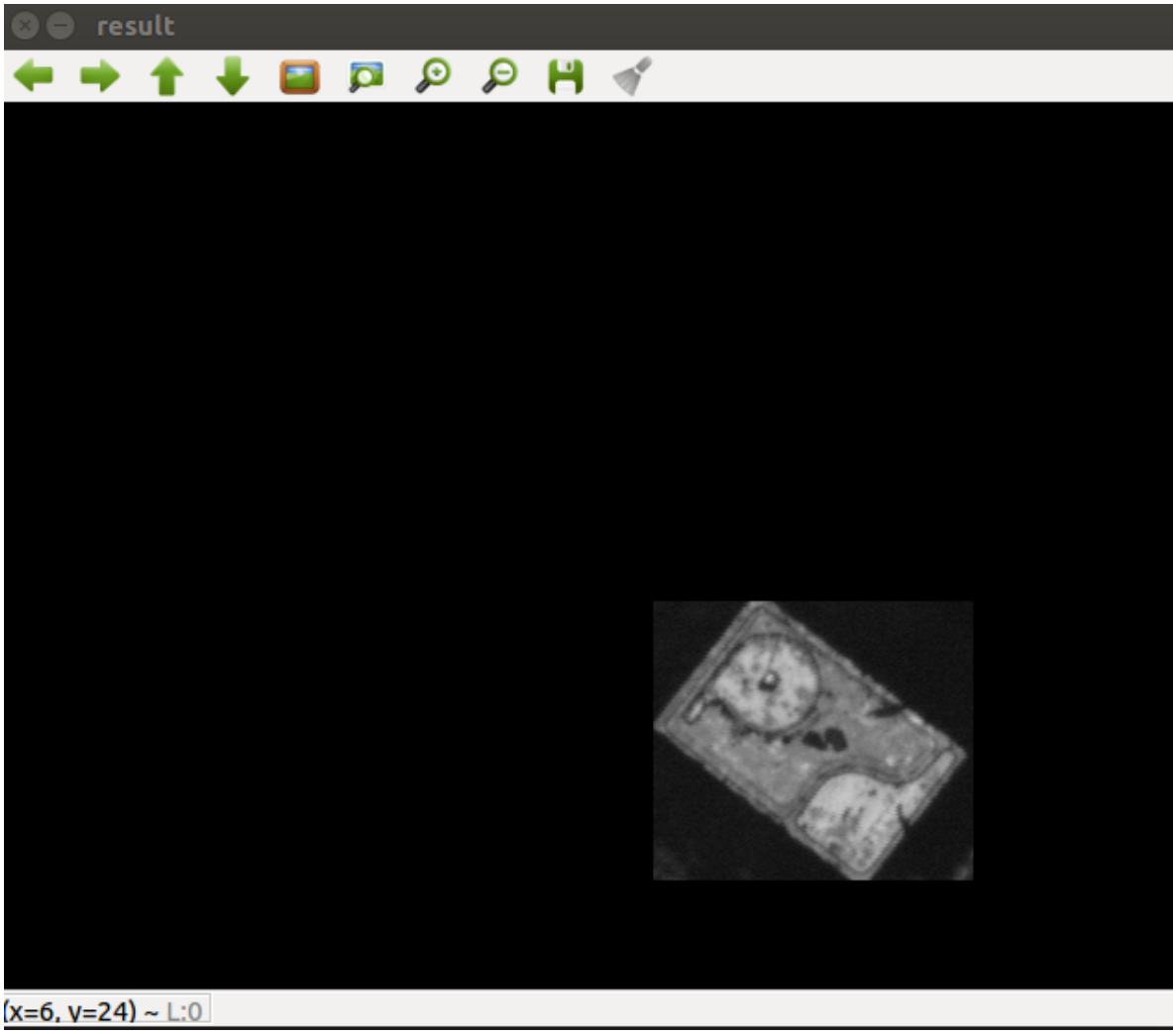

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588239291.204072477]: *****START
最大NCC绝对值=0.745581
中心坐标=（365,451）
耗时103.500431s
```

IMAGEB16



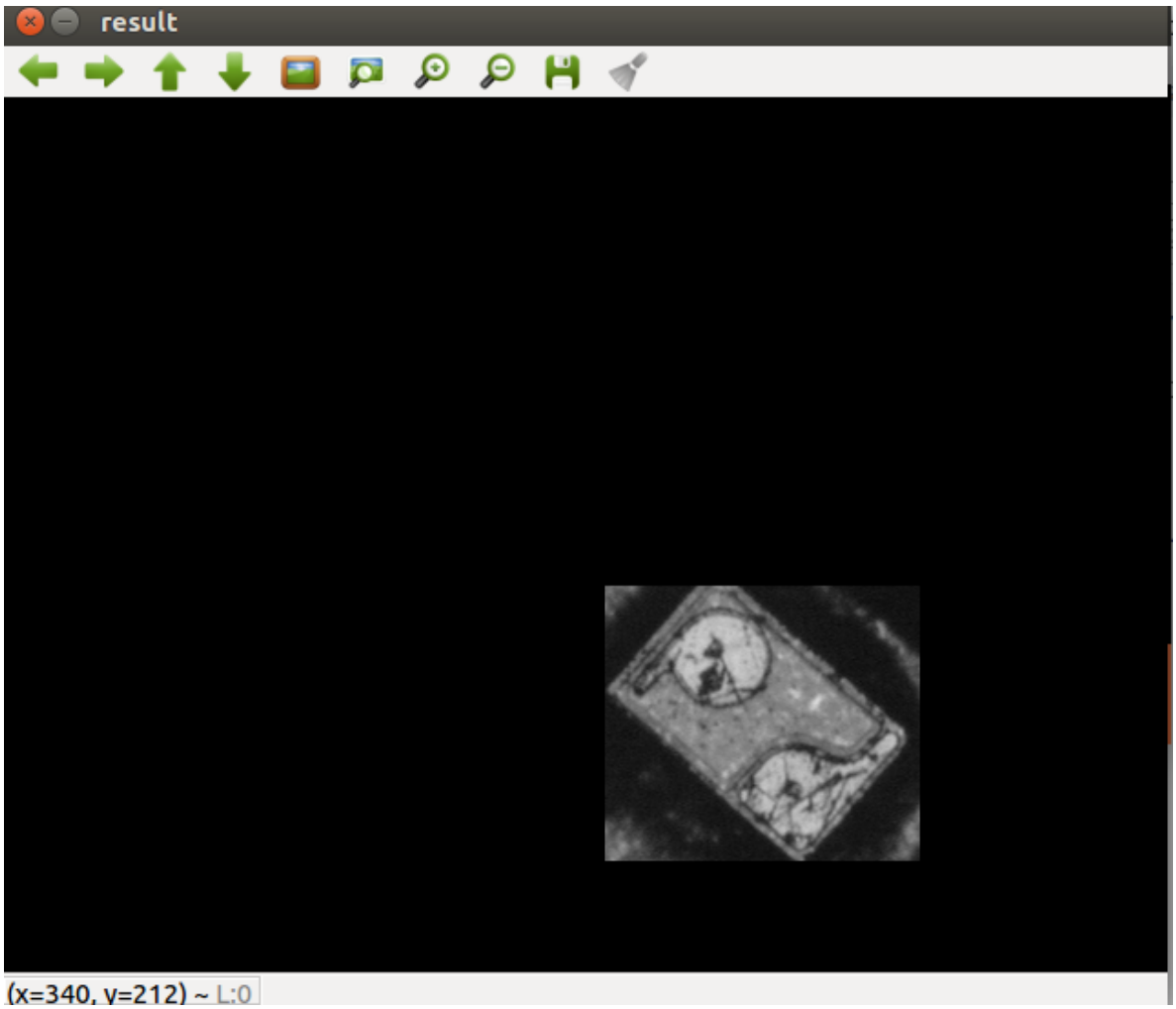

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588239514.231894363]: *****START
最大NCC绝对值=0.713959

中心坐标=（345,441）

耗时110.526402s
```

IMAGEB17



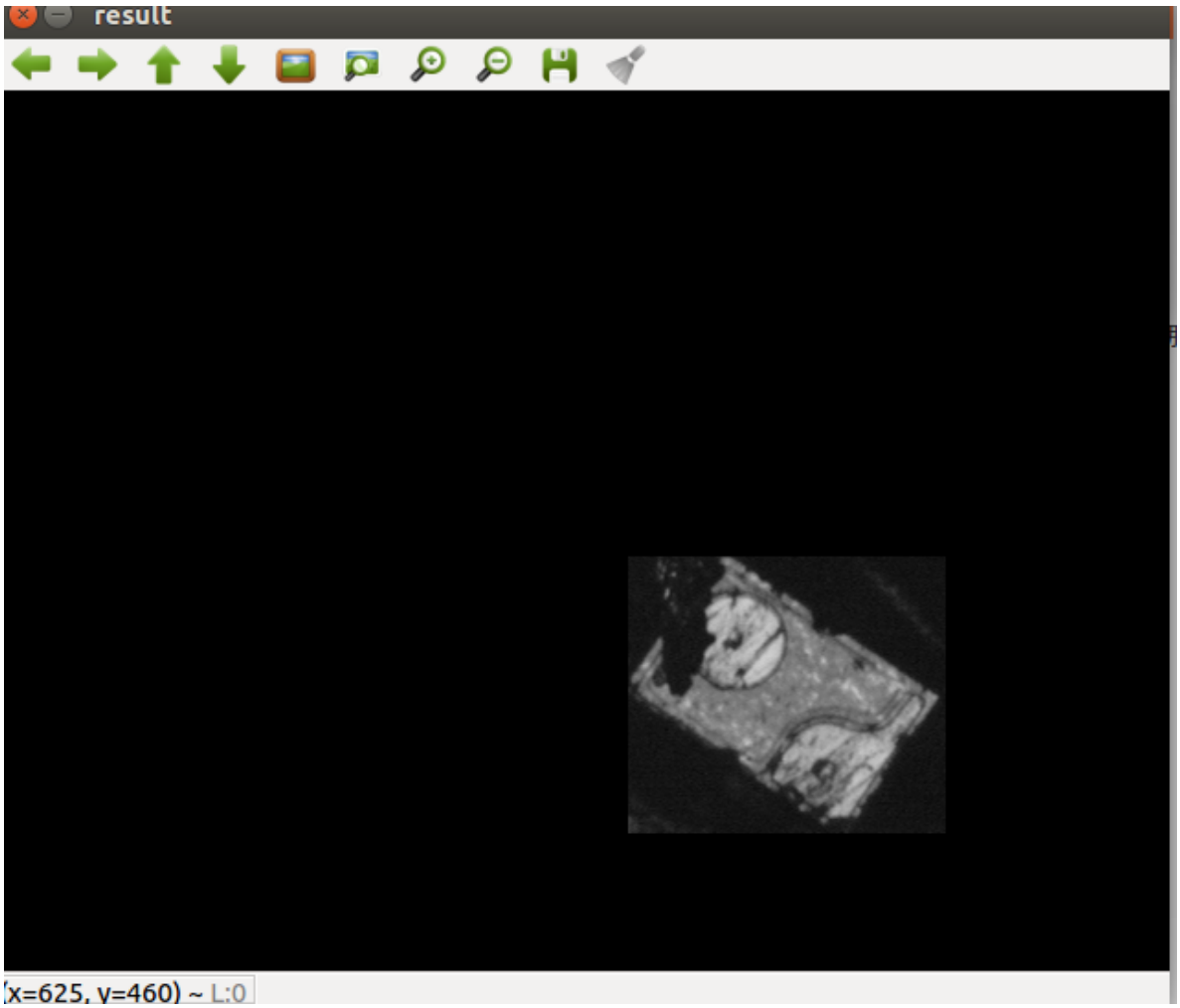

```
最大NCC绝对值=0.775676

中心坐标=（310,432）

耗时108.723505s
```

IMAGEB18



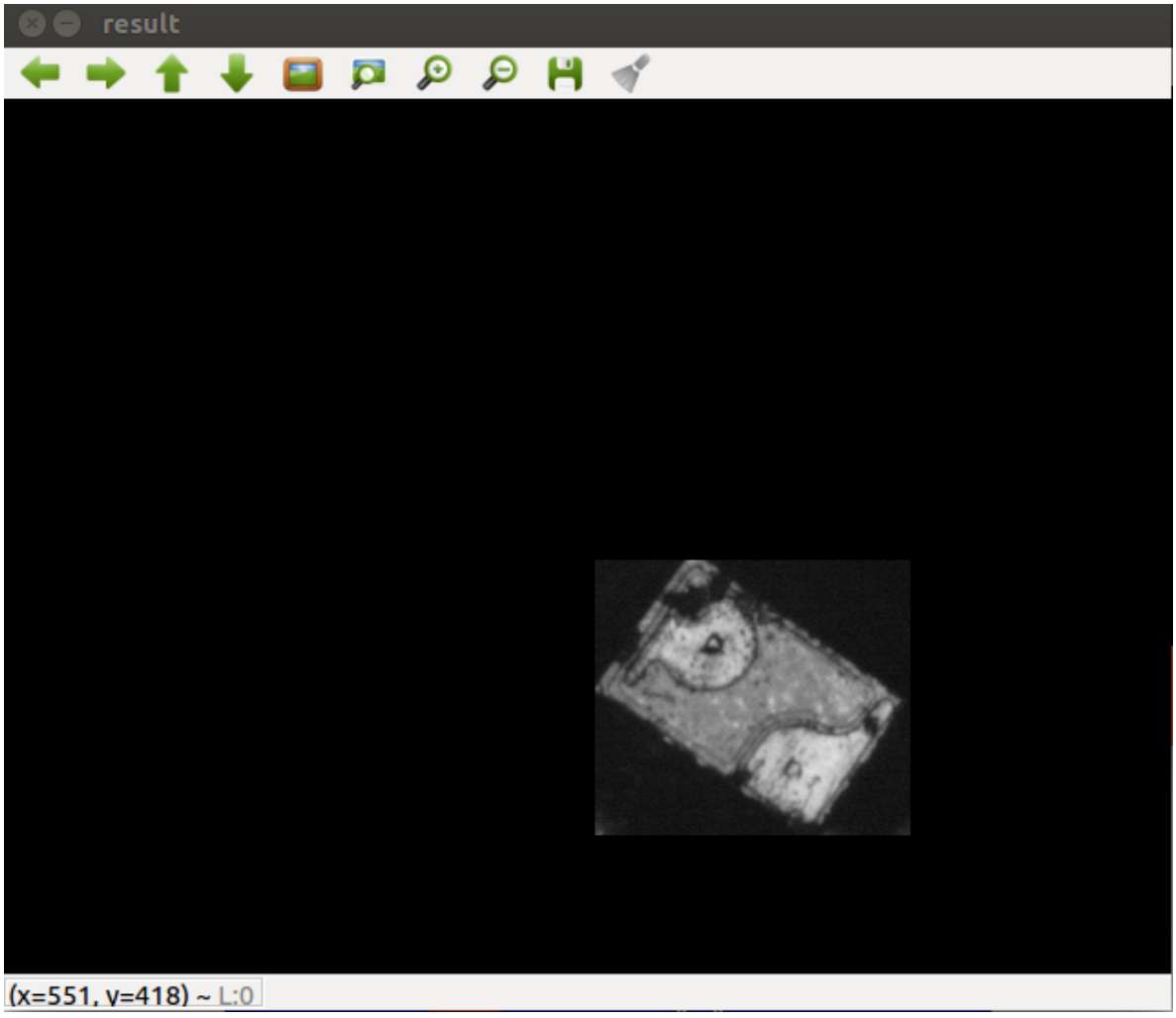

```
[[Afangcheng@fc:~/catkin_ws$ rosrun machine_vision
[ WARN] [1588240169.972016903]: *****START
最大NCC绝对值=0.637071

中心坐标=（343,417）

耗时106.363099s
```

IMAGEB19



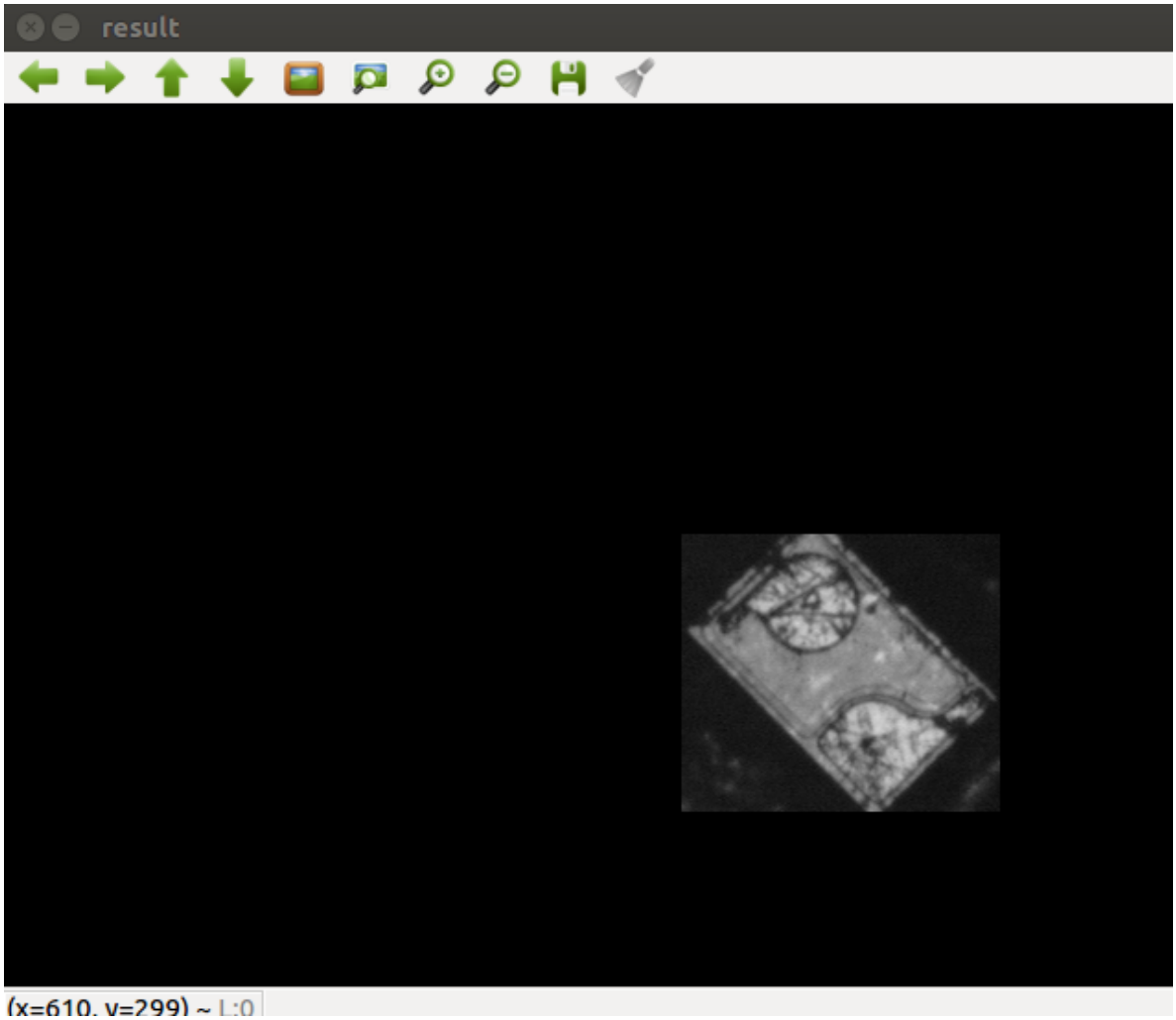

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588240340.179832931]: *****START
最大NCC绝对值=0.743881

中心坐标=（329,431）

耗时106.138852s
```

IMAGEB20



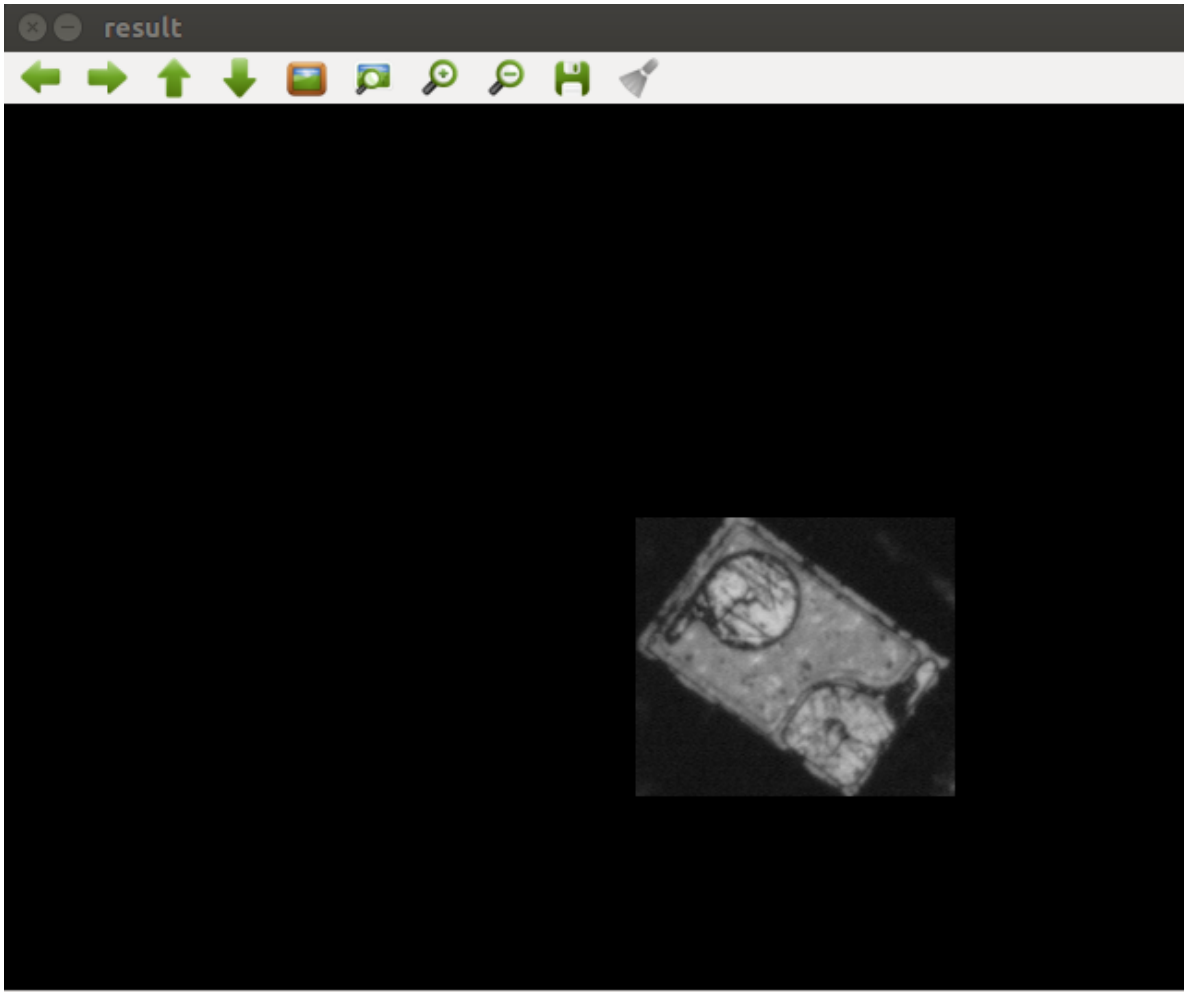

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588240485.792022357]: *****START
最大NCC绝对值=0.766676

中心坐标=（328,410）

耗时105.969435s
```

IMAGEB21



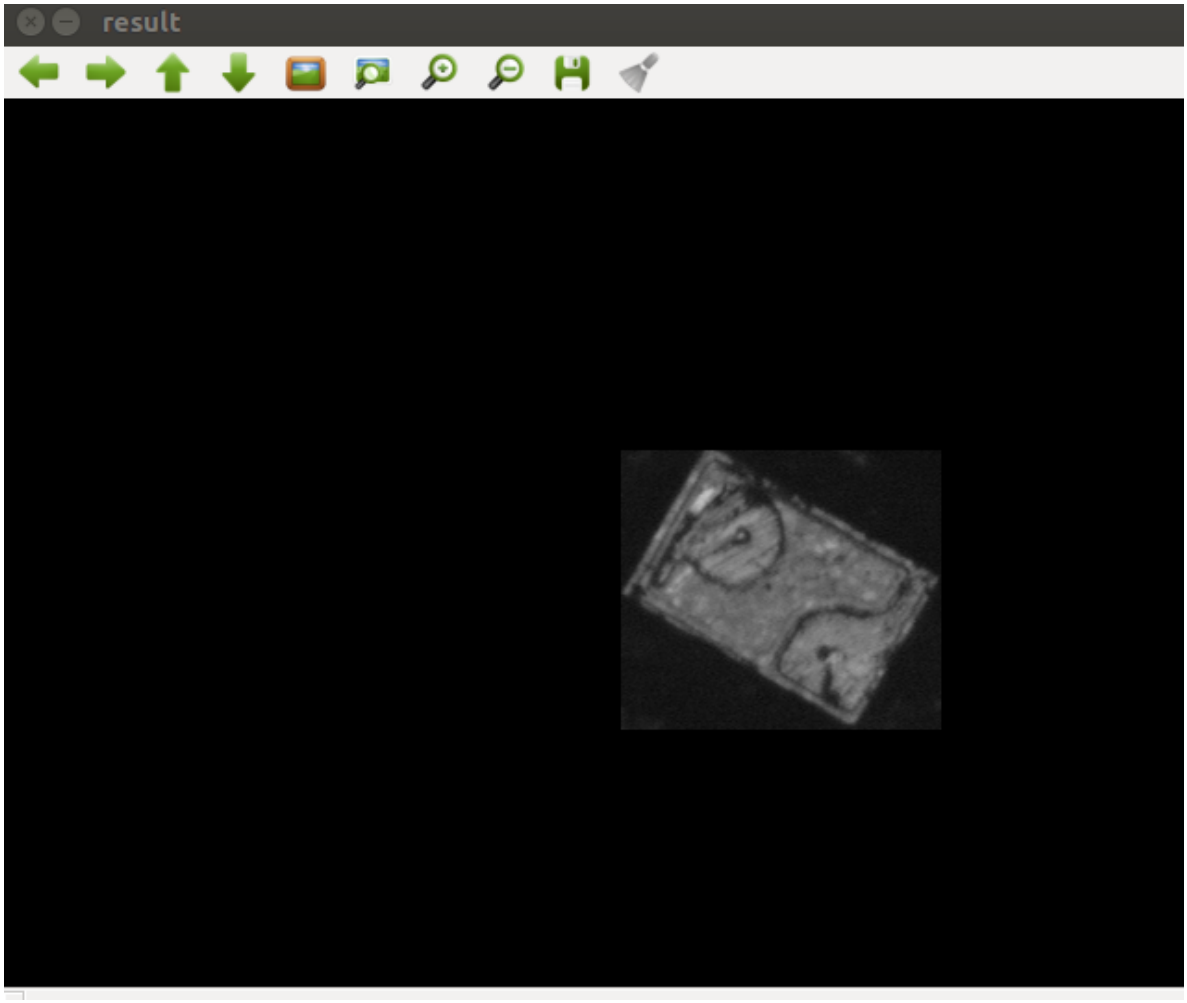

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588240633.256642373]: *****START
最大NCC绝对值=0.680513

中心坐标=（309,455）

耗时106.321650s
```

IMAGEB22



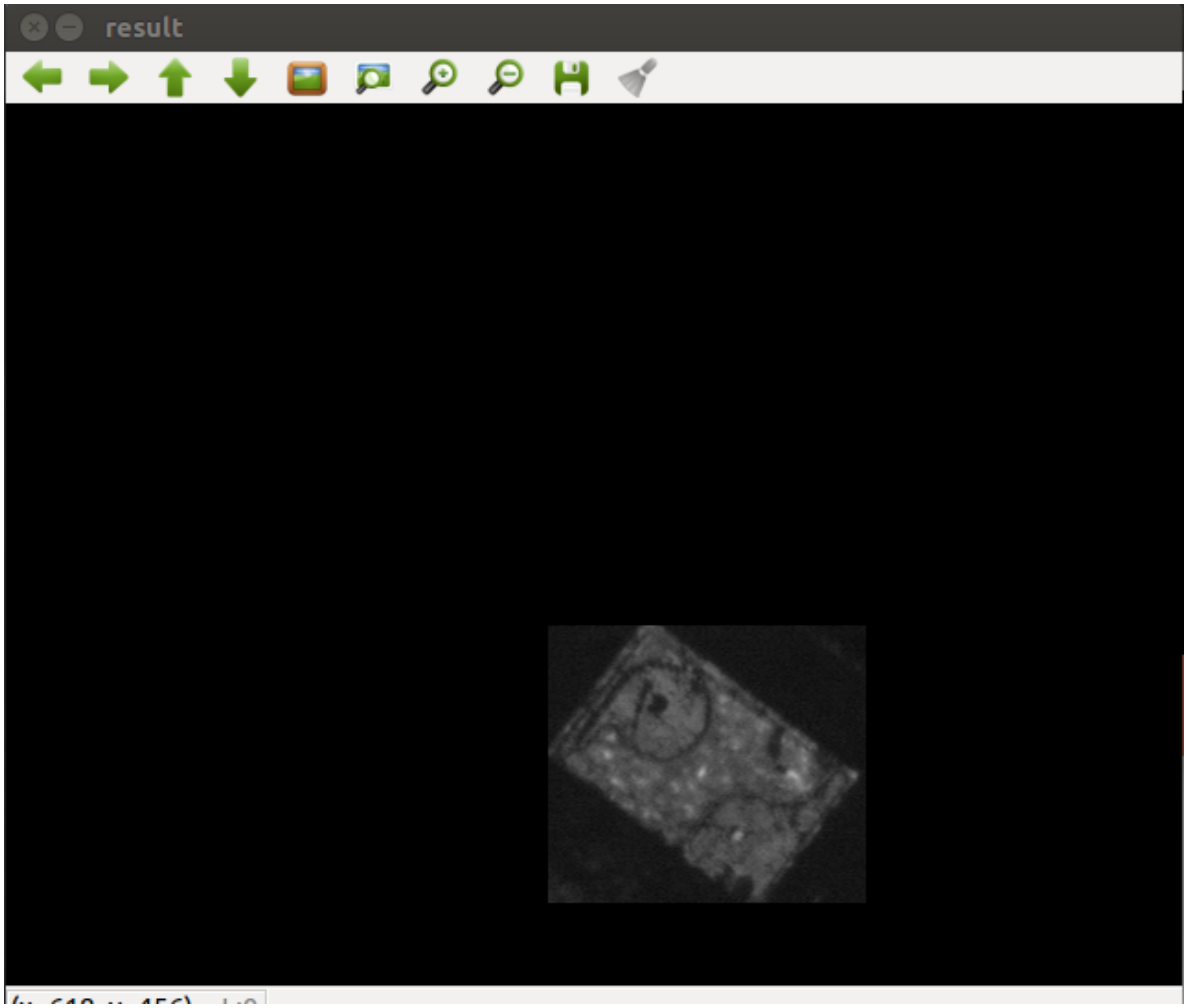

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588240782.648106002]: *****START
最大NCC绝对值=0.766037

中心坐标=（299,428）

耗时104.439491s
```

IMAGEB23



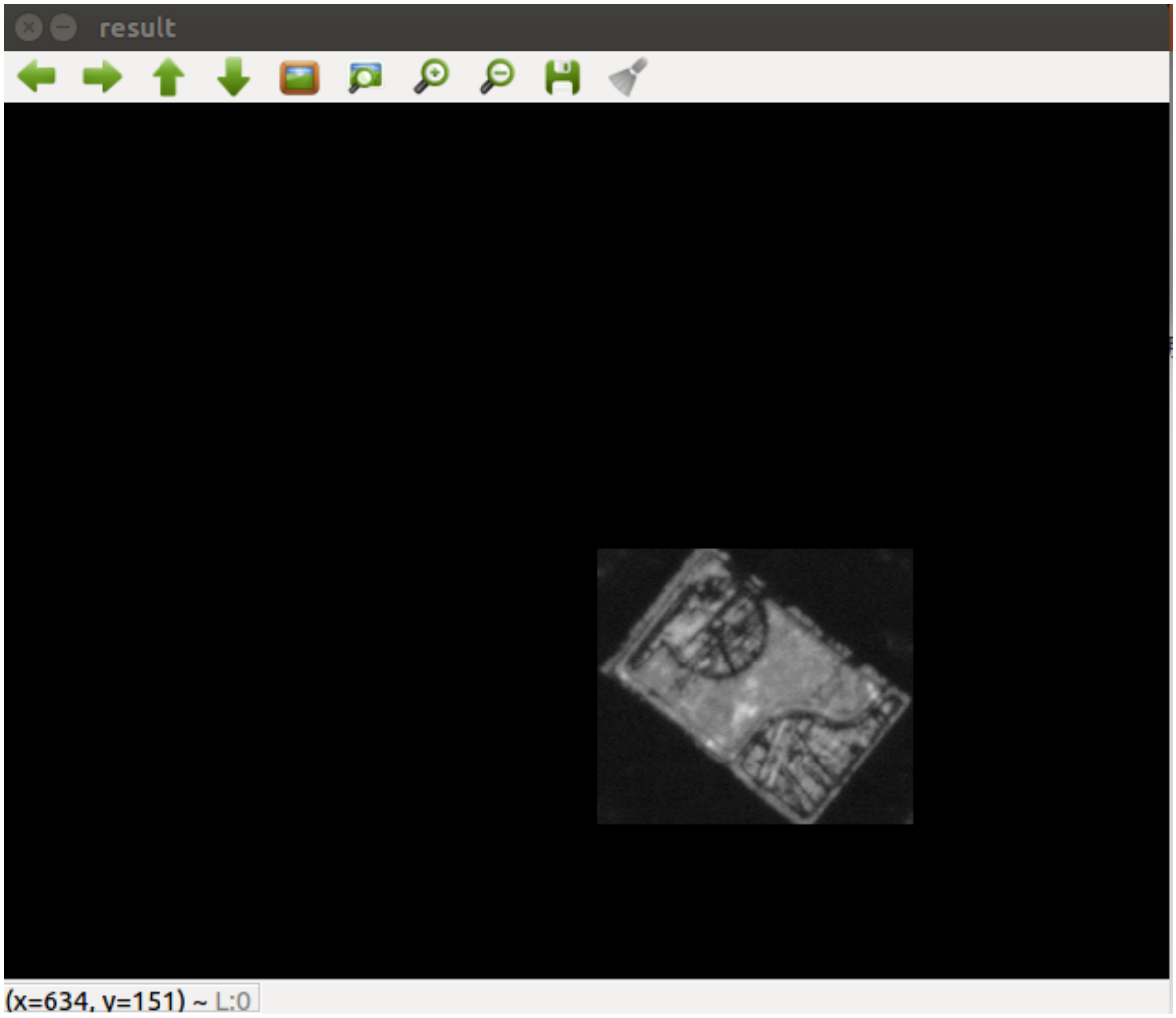

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588240929.488100437]: *****START
最大NCC绝对值=0.831356

中心坐标=（265,420）

耗时105.356836s
```

IMAGEB24



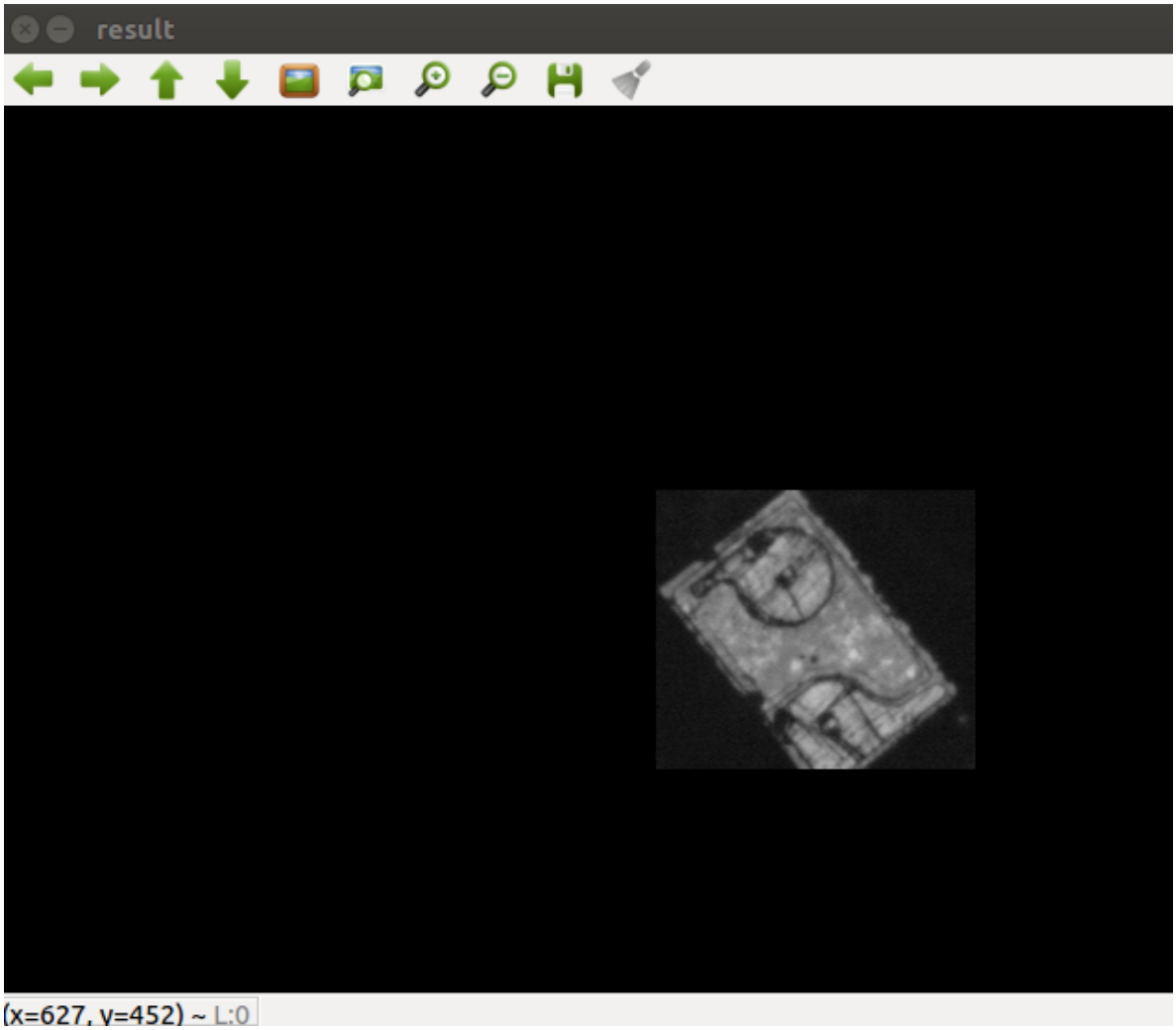

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588241082.968211569]: *****START
最大NCC绝对值=0.769872

中心坐标=（359,381）

耗时104.187946s
```

IMAGEB25



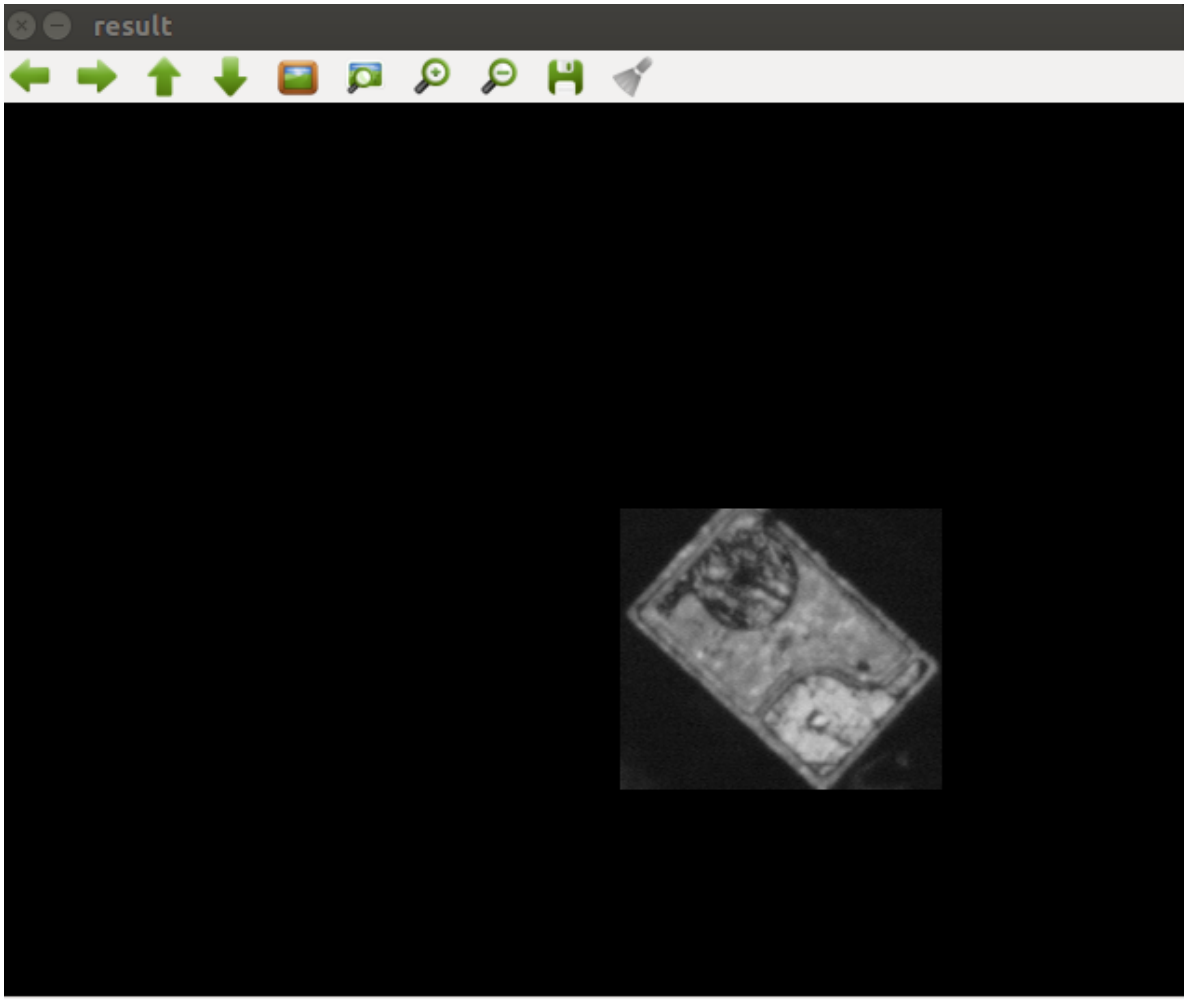

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588241253.916113843]: *****START
最大NCC绝对值=0.759582

中心坐标=（319,413）

耗时106.075575s
```

IMAGEB26



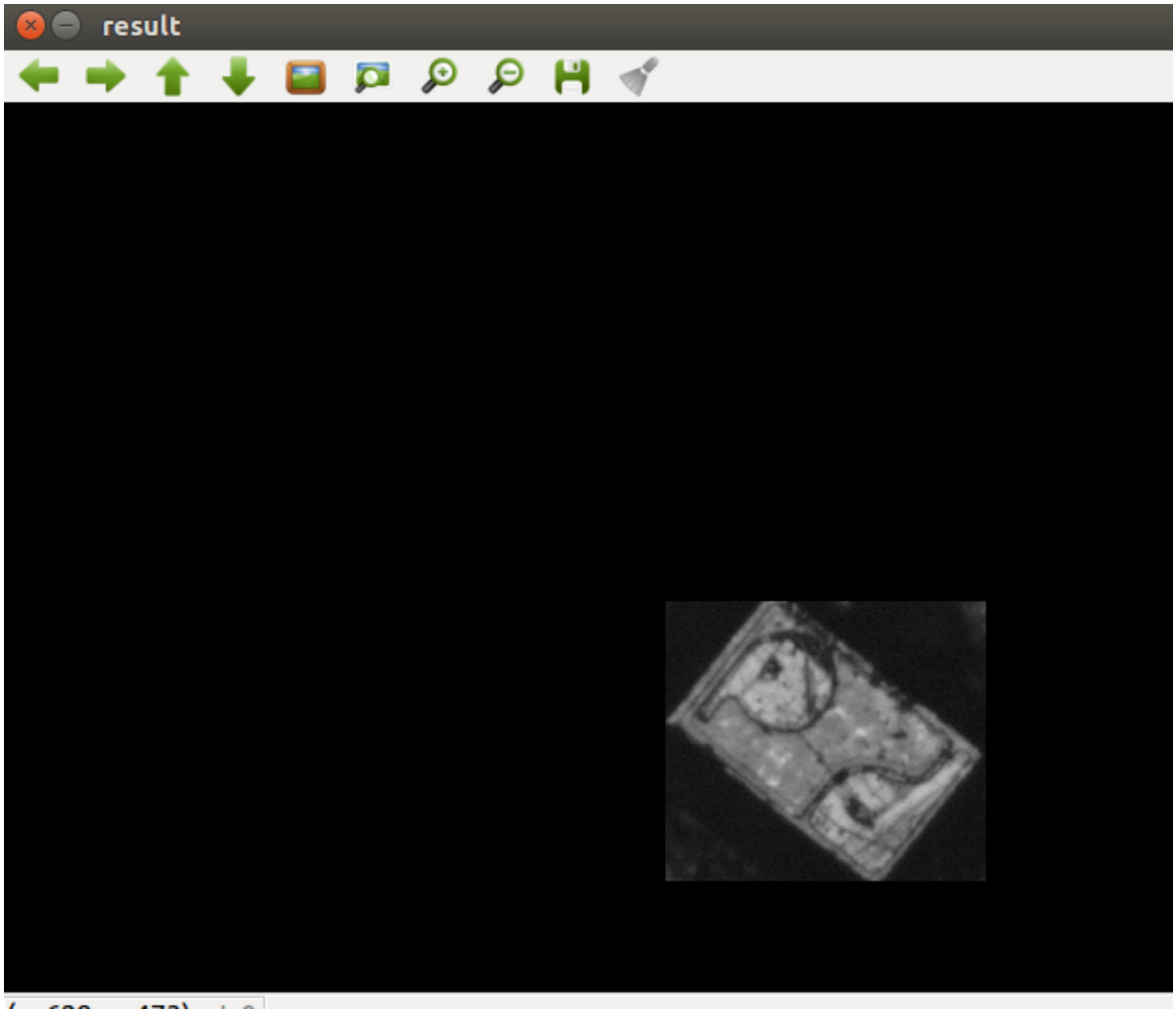

```
^[[Afangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588241397.396694660]: *****START
最大NCC绝对值=0.638022

中心坐标=（283,443）

耗时106.531226s
```

IMAGEB27



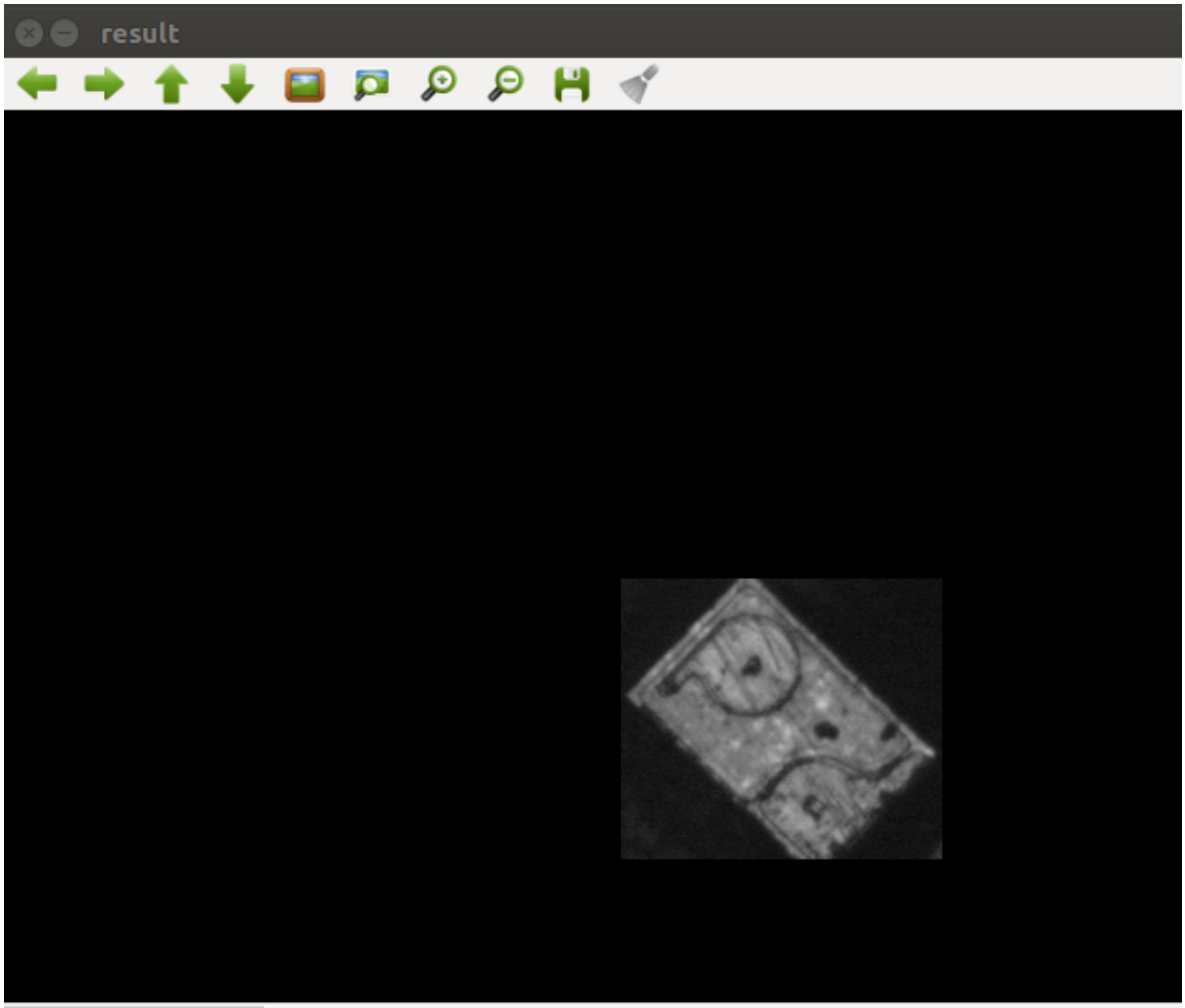

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588241547.884091085]: *****START
最大NCC绝对值=0.695881

中心坐标=（293,423）

耗时106.420560s
```

IMAGEB28



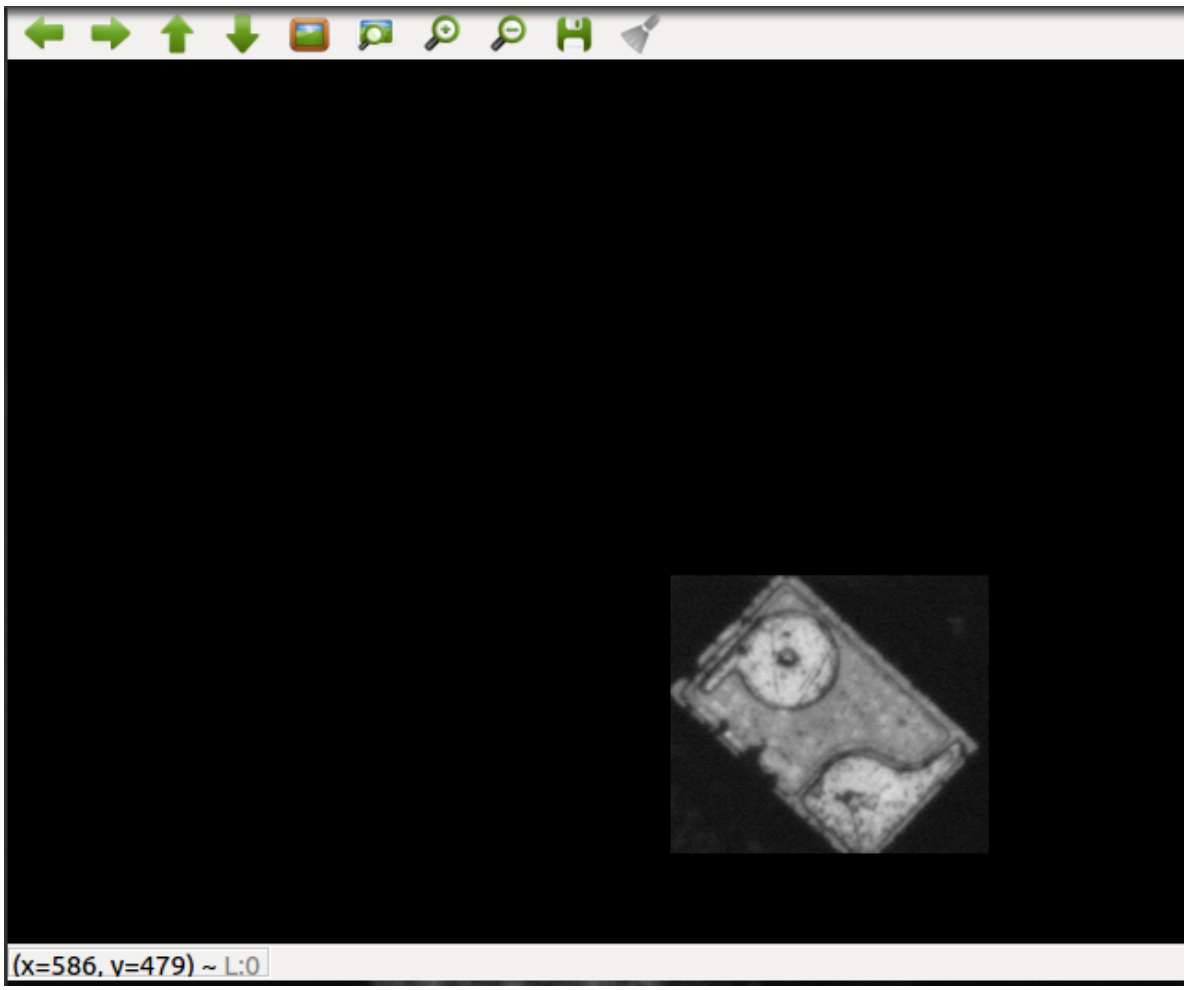

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588241689.158520588]: *****START
最大NCC绝对值=0.723782

中心坐标=（344,444）

耗时105.862029s
```

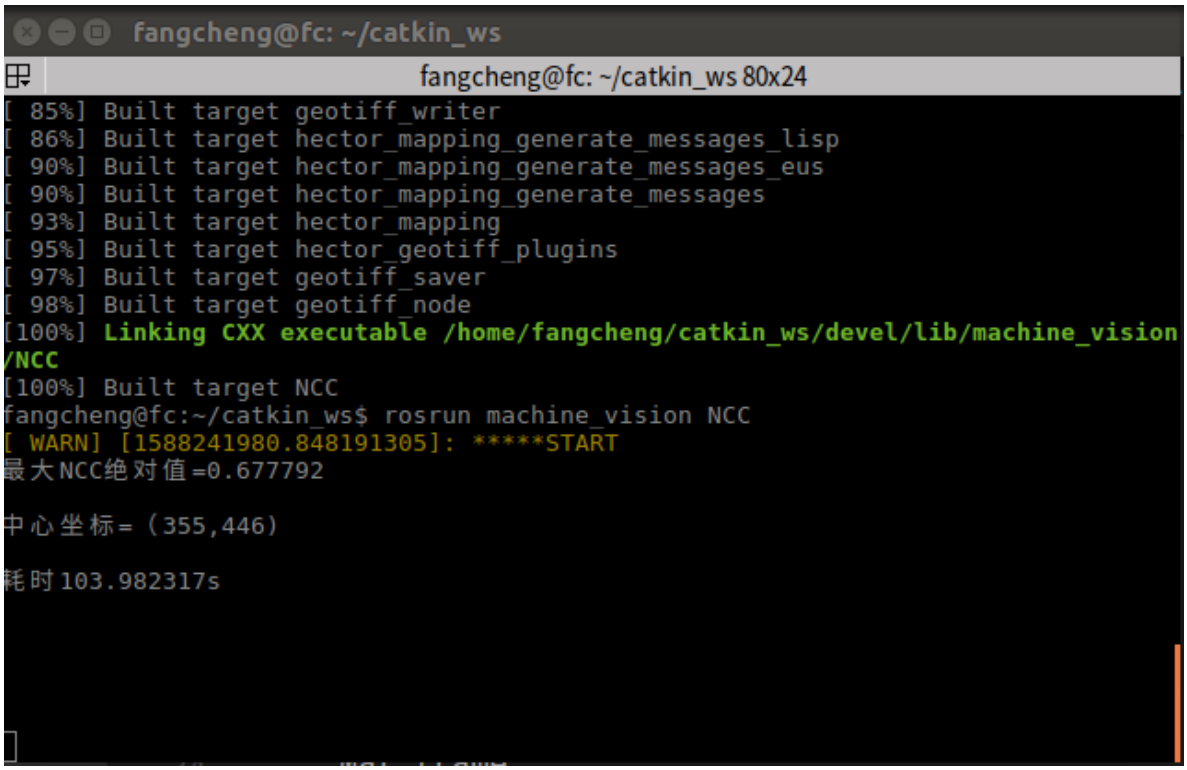
IMAGEB29



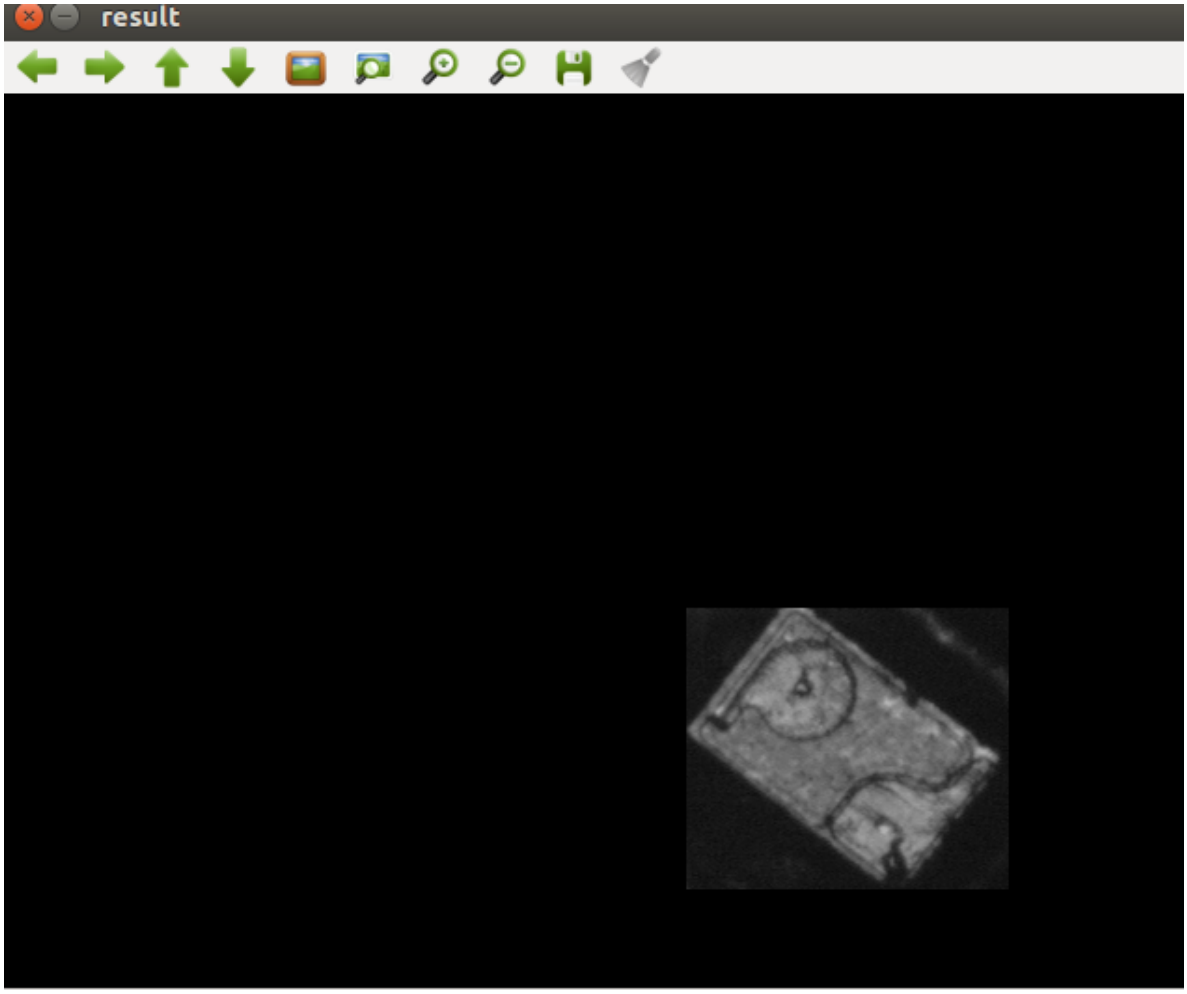

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588241834.896023374]: *****START
最大NCC绝对值=0.686344

中心坐标=（327,420）

耗时107.267858s
```

IMAGEB30


(x=586, y=479) ~ L:0

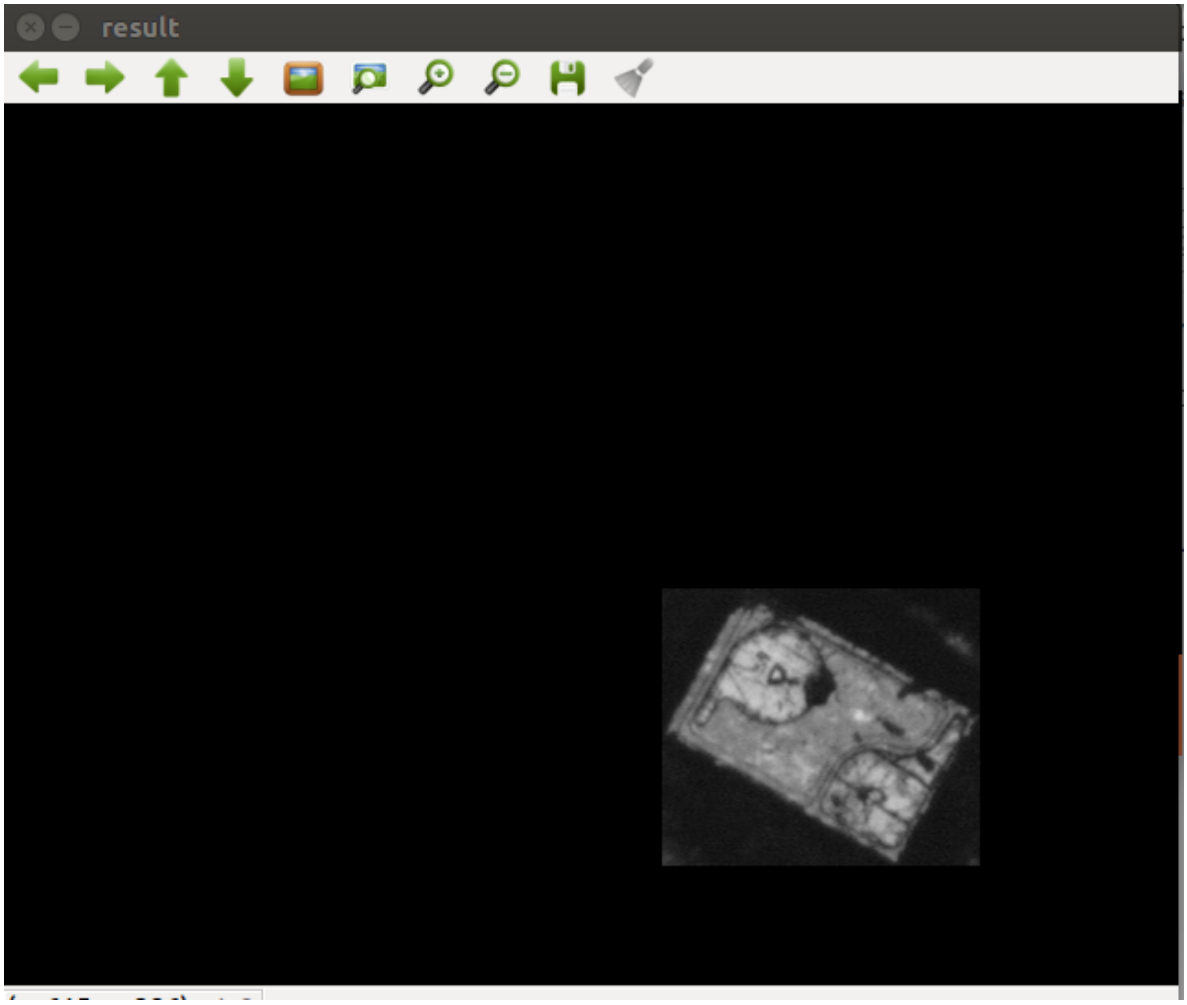

fangcheng@fc: ~/catkin_ws
fangcheng@fc: ~/catkin_ws 80x24
[85%] Built target geotiff_writer
[86%] Built target hector_mapping_generate_messages_lisp
[90%] Built target hector_mapping_generate_messages_eus
[90%] Built target hector_mapping_generate_messages
[93%] Built target hector_mapping
[95%] Built target hector_geotiff_plugins
[97%] Built target geotiff_saver
[98%] Built target geotiff_node
[100%] Linking CXX executable /home/fangcheng/catkin_ws/devel/lib/machine_vision/NCC
[100%] Built target NCC
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[WARN] [1588241980.848191305]: *****START
最大NCC绝对值=0.677792
中心坐标=（355,446）
耗时103.982317s

IMAGEB31



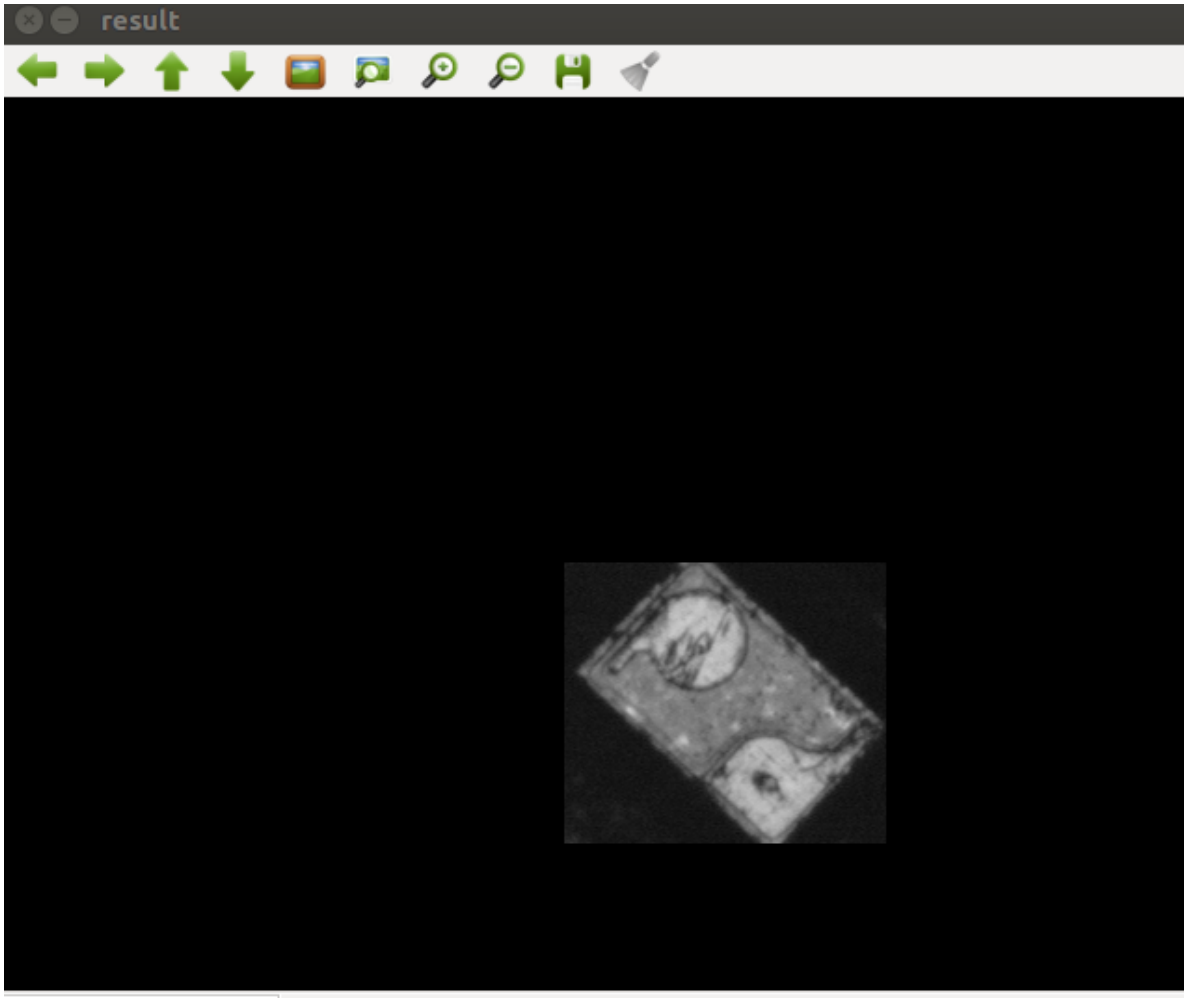

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588242132.311441863]: *****START
最大NCC绝对值=0.746040

中心坐标=（351,454）

耗时106.496033s
```

IMAGEB32



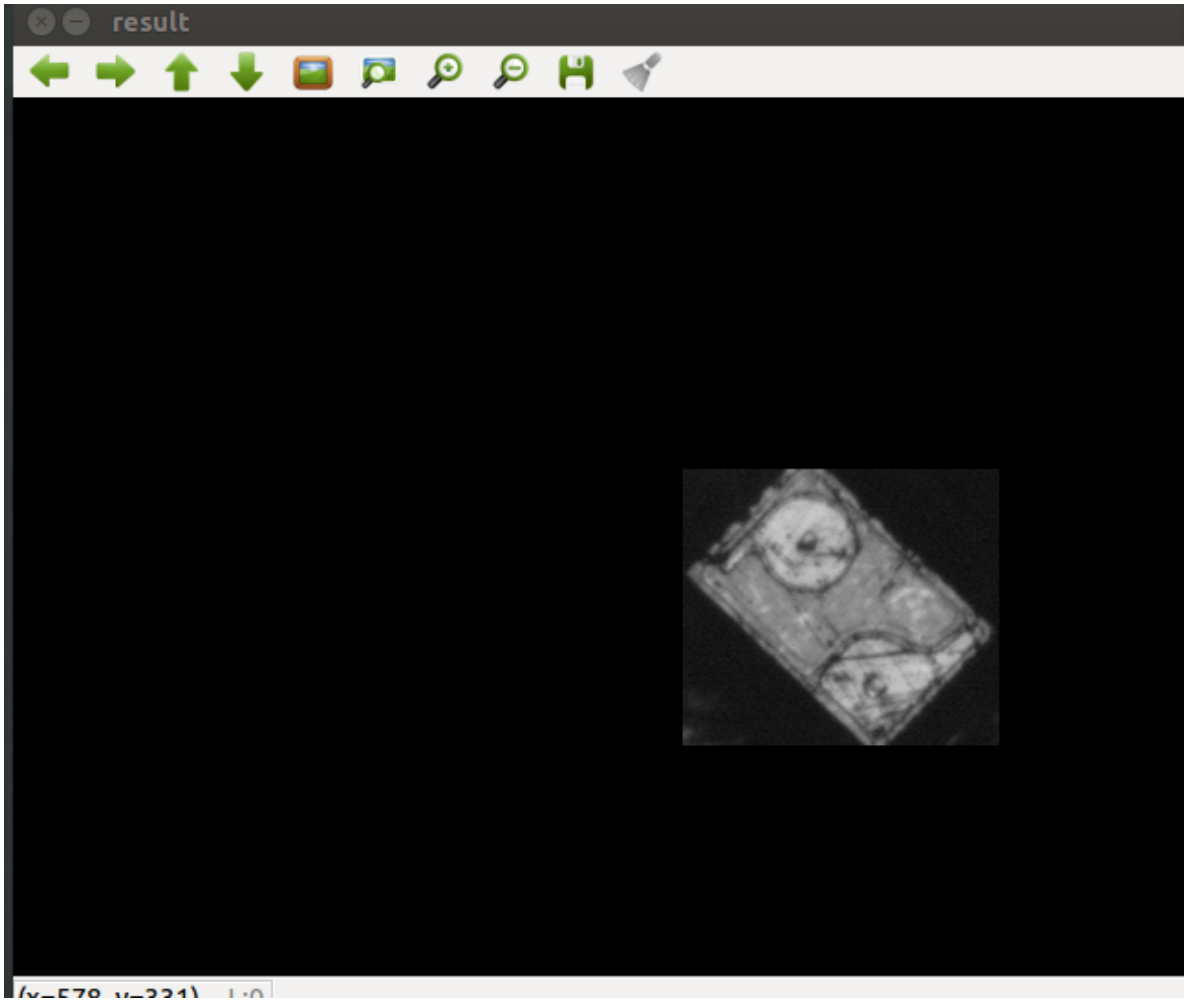

```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588242279.999962342]: *****START
最大NCC绝对值=0.809127

中心坐标=（339,445）

耗时103.060579s
```

IMAGEB33




```
fangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588242417.360254683]: *****START
最大NCC绝对值=0.670733

中心坐标=（325,389）

耗时104.210838s
```

IMAGEB34




```
^[[Afangcheng@fc:~/catkin_ws$ rosrun machine_vision NCC
[ WARN] [1588242557.036073441]: *****START
最大NCC绝对值=0.682105

中心坐标=（278,452）

耗时102.916704s
```