实验报告 4

姓名：蒋卓洋
学号：59119125

1、实现

(1)位置：main.cpp
(2)实现：

① recursive_bezier()

```
cv::Point2f recursive_bezier(const std::vector<cv::Point2f> &control_points, float t)
{
    // TODO: Implement de Casteljau's algorithm
    ////////////////////Solution/////////////////////
    ////Name:JiangZhuoyang
    ////StudentID:58119125
    ////FinishDate:21/10/28

    //It is obviously a Recurrence function

    //1.First construct the terminal condition:
    if(control_points.size() == 2){
        return (1-t)*control_points[0]+t*control_points[1];
    }
    else{
    //2.What we need to do is get 't' point_set of the the input point_set.  It is equal to link the 't' point
        //(1).construct a points set inorder to renew the input:
        std::vector<cv::Point2f> temp_points;
        //(2).use loop to re new the point:
        for(int i = 0;i < control_points.size(); i++){
            temp_points.push_back( (1-t)*control_points[i]+t*control_points[i+1] );
        }
    //3.do recursive operation:
        return recursive_bezier(temp_points,t);
    }

    ////////////////////////////////////////////////
}
```

② bezier()

```
void bezier(const std::vector<cv::Point2f> &control_points, cv::Mat &window)
{
    // TODO: Iterate through all t = 0 to t = 1 with small steps, and call de Casteljau's
    // recursive Bezier algorithm.
    ////////////////////Solution//////////////////////
    ////Name:JiangZhuoyang
    ////StudentID:58119125
    ////FinishDate:21/10/28

    //1.use loop to draw
    for(double t = 0.0; t <= 1.0; t += 0.001){
        //(1)get the point on bazier curve of 't' now:
        auto point = recursive_bezier(control_points,t);
        //(2)draw the point:
        window.at<cv::Vec3b>(point.y,point.x)[1] = 255;
    }

    ////////////////////////////////////////////////
}
```

2、结果

- 实验结果如下：

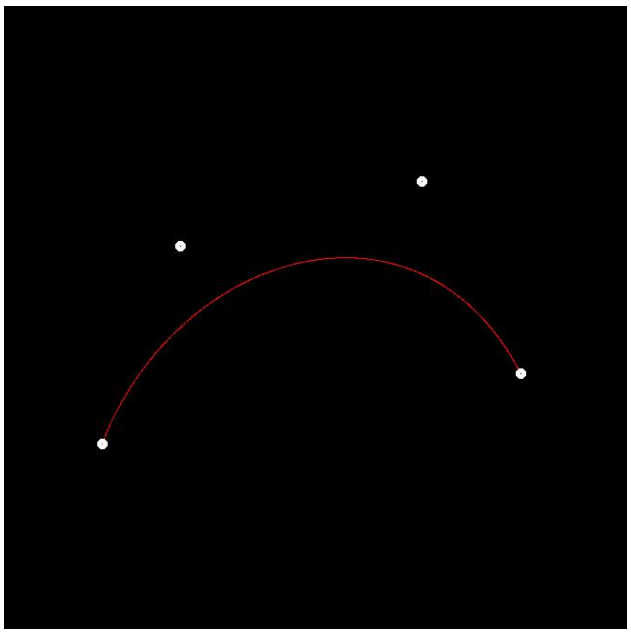

图 1. 要求曲线 1

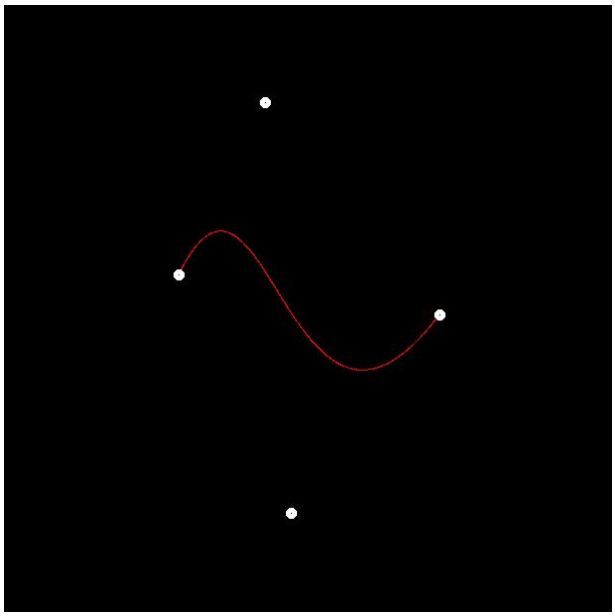

图 2.要求曲线二