

图像分割

姓名：胡天扬

学号：3190105708

专业：自动化（控制）

课程：数字图像处理与机器视觉

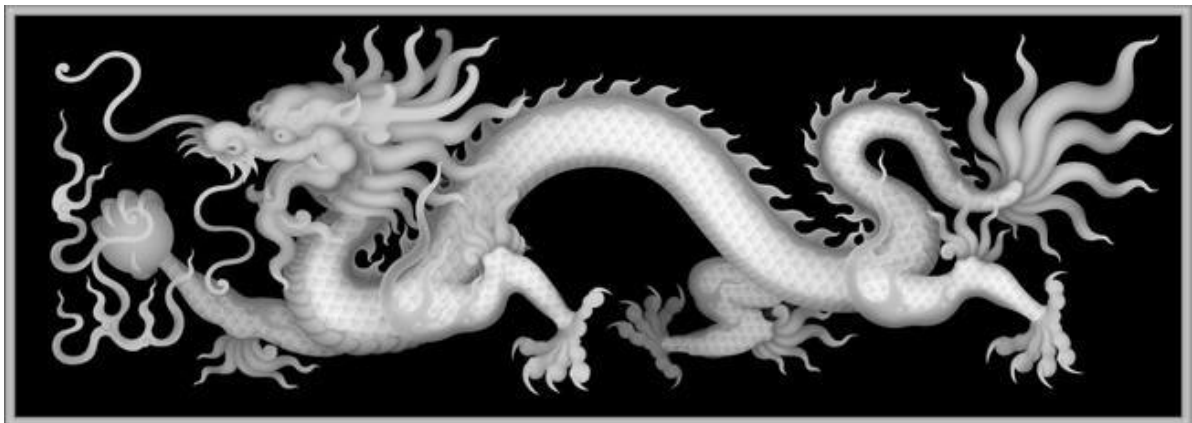
指导教师：姜伟

一、题目要求

自选一张内容简单的灰度图像，用一种方法实现图像前景、背景分割，并提取前景区域边缘。给出灰度图像、分割后二值化图像、边缘提取结果图像，以及边缘的链码表示。

二、原图

原图是一张单通道的灰度图，像素为 232 x 650。



三、大津法二值化

调用内置函数 `cv::threshold`，配合可选参数可以实现不同的二值化方法，这里用了大津法 (OTSU)。

```
1  double cv::threshold(  
2      cv::InputArray src,  
3      cv::OutputArray dst,  
4      double thresh,  
5      double maxValue,  
6      int thresholdType  
7  );
```

main.cpp

```

1 cv::Mat bin_image;
2 cv::threshold(input_image, bin_image, 0, 255, cv::THRESH_BINARY |
  cv::THRESH_OTSU);
3 showImage(bin_image, "Binary image", image_size, 0,
  ".../result/bin_image.png");

```



四、Canny边缘检测

与上次滤波一样，用函数指针来加入滑杆，分别调节Canny算法的两个阈值。

main.cpp

```

1 showImage(input_image, "Edge image", image_size, 0, "",
  Segment::cannyTrackbar);

```

Segment.cpp

```

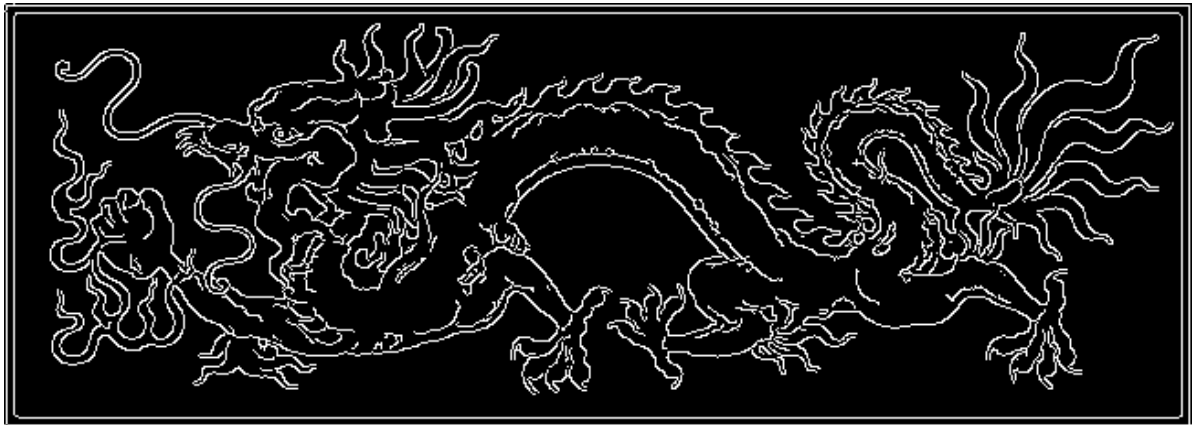
1 void Segment::cannyTrackbar(cv::Mat &src, const std::string &win_name)
2 {
3     canny_data.src = src;
4     canny_data.win_name = win_name;
5     cv::createTrackbar("threshold 1", "Edge image", nullptr, 255,
6         cannyLowThrCallback, (void*)& canny_data);
7     cv::createTrackbar("threshold 2", "Edge image", nullptr, 255,
8         cannyHighThrCallback, (void*)& canny_data);
9 }
10
11 void Segment::cannyLowThrCallback(int low_threshold, void *data)
12 {
13     cv::Mat dst;
14     cv::Canny(canny_data.src, dst, low_threshold,
15         canny_data.high_threshold);
16     canny_data.low_threshold = low_threshold;
17     cv::imshow(canny_data.win_name, dst);
18 }
19
20 void Segment::cannyHighThrCallback(int high_threshold, void *data)
21 {
22     cv::Mat dst;
23     cv::Canny(canny_data.src, dst, canny_data.low_threshold,
24         high_threshold);
25     canny_data.high_threshold = high_threshold;
26 }

```

```

24 cv::imshow(canny_data.win_name, dst);
25 }

```



五、边界链码

5.1 获取二值化图像的边界链码

这里使用内置函数 `cv::findContours` 获取链码。

```

1 void cv::findContours(
2     cv::InputOutputArray image,
3     cv::OutputArrayOfArrays contours, // type ---
4     std::vector<std::vector<cv::Point>>
5     cv::OutputArray hierarchy, // type --- std::vector<cv::Vec4i>
6     int mode,
7     int method,
8     cv::Point offset = cv::Point()
9 );

```

其中 `mode` 表示轮廓提取的方式，可选参数如下，这里选用 `cv::RETR_TREE`。

参数	含义
<code>cv::RETR_EXTERNAL</code>	只提取最外面的轮廓
<code>cv::RETR_LIST</code>	提取所有轮廓并将其放入列表
<code>cv::RETR_CCOMP</code>	提取所有轮廓并将组织成一个两层结构，顶层为外部轮廓，第二层为 hole 的轮廓
<code>cv::RETR_TREE</code>	提取所有轮廓并组织成轮廓嵌套的完整层级结构

`method` 表示轮廓展现的方法，可选参数如下，这里选用 `cv::CHAIN_APPROX_SIMPLE`。

参数	含义
<code>cv::CHAIN_APPROX_NONE</code>	将轮廓中的所有点的编码转换成点
<code>cv::CHAIN_APPROX_SIMPLE</code>	压缩水平、垂直和对角直线段，仅保留它们的端点
<code>cv::CHAIN_APPROX_TC89_L1</code>	应用 <i>Teh – Chin</i> 链近似算法中的一种风格

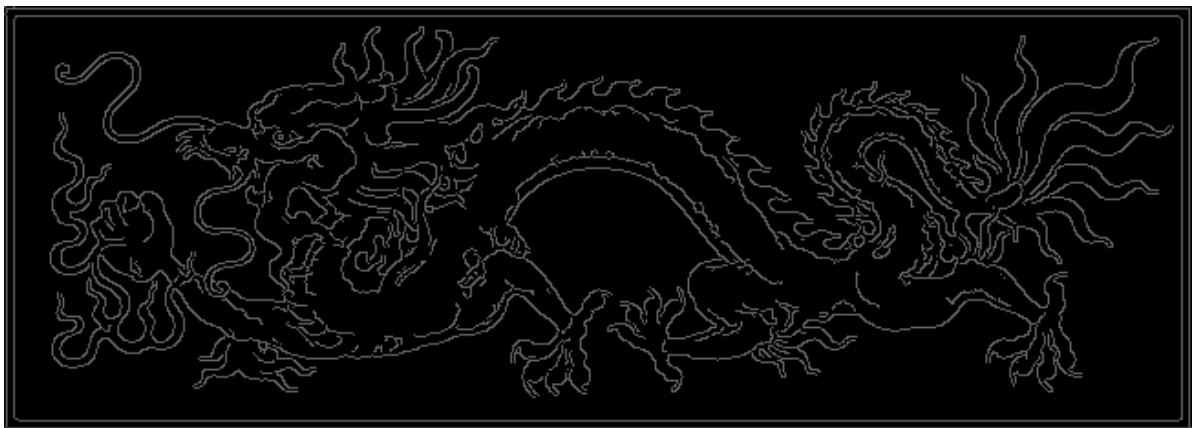
5.2 绘制获取的轮廓图像

这里使用内置函数 `cv::drawContours`

```
1 void cv::drawContours(  
2     cv::InputOutputArray image,           // 用于绘制的输入图像  
3     cv::InputArrayOfArrays contours,      // 点的vectors的vector  
4     int contourIdx,                      // 需要绘制的轮廓的指数 (-1 表示 "all")  
5     const cv::Scalar& color,             // 轮廓的颜色  
6     int thickness = 1,                     
7     int lineType = 8,                     
8     cv::InputArray hierarchy = noArray(),  
9     int maxLevel = INT_MAX,               
10    cv::Point offset = cv::Point()  
11 )
```

main.cpp

```
1 // contour  
2 std::vector<std::vector<cv::Point>> contours;  
3 std::vector<cv::Vec4i> hierarchy;  
4 cv::findContours(edge_image, contours, hierarchy, cv::RETR_TREE,  
5     cv::CHAIN_APPROX_SIMPLE);  
6 cv::Mat contour_image = cv::Mat::zeros(input_image.rows, input_image.cols,  
7     CV_8UC3);  
8 cv::drawContours(contour_image, contours, -1, cv::Scalar(100, 100, 100));  
9 showImage(contour_image, "Contour image", image_size, 0,  
10     "../result/contour_image.png");
```



5.3 写入文件

链码的变量类型为 `std::vector<std::vector<cv::Point>>`，使用 STL 容器的特性进行遍历，并结合 csv 以逗号分割的特点，将坐标点写入 `contour_data.csv` 中，每一行为一个特定的轮廓，每一格为特定轮廓上的一个点。

```

1 // write to file
2 std::ofstream outfile;
3 outfile.open("../result/contour_data.csv", std::ios::out);
4 for (const auto & contour : contours)
5 {
6     for (auto point : contour)
7         outfile << "\"" << point << "\", ";
8     outfile << std::endl;
9 }

```

WPS Office

contour_data.csv

Menu Home Insert Page Layout Formulas Data Review View Special Features

Unsyncronized Share

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
[0, 0]	[1, 1]	[1, 3]	[0, 4]	[1, 5]	[1, 229]	[0, 230]	[0, 231]	[1, 231]	[2, 230]	[649, 23]	[648, 22]	[648, 1]	[6, 1]	[5, 0]	[4, 1]	[1, 1]					
[0, 230]	[1, 229]	[2, 230]	[1, 231]																		
[1, 2]	[2, 1]	[647, 1]	[648, 2]	[648, 22]	[647, 23]	[2, 230]	[1, 229]	[1, 5]	[0, 4]	[1, 3]											
[6, 5]	[6, 6]	[5, 7]	[5, 224]	[6, 225]	[7, 225]	[8, 226]	[642, 22]	[643, 22]	[644, 22]	[644, 7]	[642, 5]										
[7, 5]	[642, 5]	[644, 7]	[644, 22]	[642, 22]	[8, 226]	[7, 225]	[6, 225]	[5, 224]	[5, 7]												
[294, 20]	[291, 20]	[289, 20]	[288, 20]	[287, 20]	[286, 20]	[287, 20]	[287, 20]	[289, 20]	[289, 21]	[290, 21]	[291, 21]	[292, 21]	[288, 20]	[288, 20]	[286, 20]	[287, 20]	[288, 20]	[289, 20]	[291, 20]	[292, 20]	[293, 20]
[140, 19]	[141, 20]	[146, 20]	[141, 200]																		
[310, 19]	[310, 19]	[309, 20]	[309, 20]	[308, 20]	[307, 20]	[307, 20]	[307, 20]	[304, 20]	[300, 20]	[300, 20]	[303, 21]	[302, 21]	[301, 21]	[298, 20]	[298, 20]	[296, 20]	[296, 20]	[297, 20]	[297, 20]	[297, 20]	[296, 20]
[308, 198]																					
[154, 19]	[153, 19]	[151, 19]	[150, 19]	[148, 19]	[149, 20]	[149, 20]	[150, 20]	[152, 20]	[152, 20]	[153, 20]	[153, 20]	[155, 20]	[160, 20]	[156, 20]	[153, 20]	[153, 20]	[152, 20]	[152, 20]	[151, 20]	[150, 20]	[149, 20]
[136, 18]	[139, 19]	[139, 19]	[141, 19]	[142, 19]	[147, 19]	[147, 19]	[146, 19]	[142, 19]	[141, 19]	[140, 193]											
[310, 18]	[309, 18]	[309, 19]	[310, 19]	[310, 19]	[311, 19]	[311, 19]	[311, 19]	[310, 19]	[310, 19]	[309, 19]	[309, 189]										
[413, 18]	[413, 19]	[414, 19]	[415, 19]	[413, 18]	[413, 18]	[414, 18]	[418, 19]	[419, 19]	[415, 187]												
[307, 18]	[307, 19]	[308, 19]	[307, 19]	[307, 19]	[308, 19]	[308, 19]	[308, 19]	[307, 192]													
[305, 18]	[305, 18]	[303, 19]	[303, 19]	[302, 19]	[302, 19]	[299, 19]	[298, 19]	[299, 19]	[301, 19]	[302, 19]	[302, 19]	[303, 19]	[303, 19]	[304, 19]	[304, 19]	[305, 189]					
[204, 18]	[203, 18]	[203, 18]	[202, 18]	[203, 18]	[203, 18]	[204, 18]	[205, 18]	[206, 18]	[206, 18]	[205, 186]											
[193, 18]	[192, 18]	[192, 18]	[191, 18]	[190, 18]	[191, 18]	[193, 186]															
[151, 184]																					
[123, 18]	[122, 18]	[122, 18]	[121, 18]	[121, 19]	[117, 19]	[109, 19]	[108, 19]	[108, 20]	[105, 20]	[104, 20]	[103, 20]	[104, 20]	[106, 20]	[107, 20]	[108, 20]	[110, 20]	[110, 20]	[112, 20]	[113, 20]	[114, 20]	[115, 20]
[361, 18]	[361, 18]	[360, 18]	[360, 18]	[362, 18]	[361, 18]	[360, 18]	[360, 18]	[361, 184]													
[443, 18]	[442, 18]	[439, 18]	[438, 18]	[437, 18]	[435, 18]	[434, 18]	[433, 18]	[426, 18]	[425, 18]	[424, 18]	[423, 18]	[422, 18]	[421, 18]	[422, 18]	[425, 18]	[426, 18]	[428, 18]	[426, 18]	[425, 18]	[422, 18]	[421, 18]
[147, 18]	[147, 18]	[148, 18]	[149, 183]																		
[384, 18]	[383, 18]	[381, 18]	[380, 18]	[378, 18]	[381, 18]	[382, 18]	[384, 18]	[385, 180]													
[208, 18]	[209, 18]	[208, 18]	[209, 18]	[209, 18]	[210, 18]	[212, 18]	[214, 18]	[214, 18]	[215, 18]	[212, 18]	[211, 18]	[211, 18]	[213, 18]	[214, 18]	[215, 18]	[215, 18]	[214, 18]	[214, 18]	[215, 18]	[216, 18]	[217, 18]
[182, 18]	[183, 18]	[184, 18]	[184, 18]	[181, 18]	[180, 18]	[179, 18]	[176, 18]	[175, 18]	[173, 18]	[171, 18]	[170, 18]	[171, 18]	[172, 18]	[173, 18]	[175, 18]	[176, 18]	[179, 18]	[180, 18]	[181, 18]	[182, 18]	[183, 18]
[435, 17]	[434, 17]	[433, 17]	[430, 18]	[424, 18]	[422, 17]	[423, 17]	[422, 17]	[423, 17]	[423, 18]	[430, 18]	[431, 17]	[432, 17]	[434, 17]	[435, 17]	[436, 17]	[438, 17]	[436, 17]	[435, 177]			
[393, 17]	[392, 17]	[392, 17]	[391, 17]	[388, 17]	[387, 17]	[388, 17]	[389, 17]	[391, 17]	[393, 176]												
[334, 17]	[333, 17]	[331, 17]	[330, 17]	[330, 17]	[332, 17]	[333, 17]	[334, 17]	[336, 17]	[336, 18]	[338, 18]	[340, 18]	[344, 18]	[347, 18]	[348, 18]	[349, 18]	[350, 18]	[351, 18]	[354, 19]	[354, 19]	[356, 19]	[356, 19]

contour_data

100%

六、总结

本次作业要求实现前景背景的分割和轮廓提取等功能，分别用了大津法和Canny算子，因为它们实在太有名了。不过这张图的背景是全黑图像，本身区分就很明显，所以效果很好，如果把Canny的阈值调小则会产生更多轮廓边界。另外由于最近考试周太忙了所以这次就直接调用了内置函数，来不及及自己实现。（主要是边界链码提取写起来太麻烦了）