

高级图像处理与分析课程实验报告1

#

学号	姓名	日期
SA18225428	许强	2019.04.01

实验名称	灰度变换
实验内容	1、利用 OpenCV 读取图像。具体内容：用打开 OpenCV 打开图像，并在窗口中显示。 2、灰度图像二值化处理 具体内容：设置并调整阈值对图像进行二值化处理。 3、灰度图像的对数变换 具体内容：设置并调整 r 值对图像进行对数变换。 4、灰度图像的伽马变换 具体内容：设置并调整γ 值对图像进行伽马变换。 5、彩色图像的补色变换 具体内容：对彩色图像进行补色变换。
实验完成情况（包括完成的实验内容及每个实验的完成程度。注意要贴出每个实验的核心代码）	5个模块全部完成
实验中的问题（包括在实验中遇到的问题，以及解决问题的方法）	此次5个基本功能完成，补色变换需要将RGB转为CMY或者HSI色域，补色后再转为RGB，开始配置OpenCV环境比较麻烦，需要熟悉一下。
实验结果（实验完成后的源码和打包文件的说明）	代码注释中含有部分说明

```
1  •//
2  // Created by XQ on 2019-03-28.
3  //
4
5  #include<iostream>
6  #include<string>
7
```

```

8   #include <opencv2/imgcodecs.hpp>
9   #include <opencv2/highgui.hpp>
10  #include <opencv2/imgproc/imgproc.hpp>
11  #include <opencv2/opencv.hpp>
12  #include <opencv2/core/types_c.h>
13  #include <opencv2/core/core_c.h>
14  #include <opencv2/highgui/highgui.hpp>
15
16  using namespace std;
17  using namespace cv;
18
19
20  bool loadImage(string &, int);
21  bool binaryImage(string &, int flag = 1, int r = 80);
22  bool logTransformImage(string &, int flag = 1, double r = 5);
23  bool gammaImage(string &, int flag = 1, double r = 1.2);
24  bool complementColorImage(string &, int flag = 1);
25  int main(){
26      string str = "/Volumes/数据/图片/2k/lostwall.jpg";
27      cout << loadImage(str,1) << endl;
28      cout << binaryImage(str,0) << endl;
29      cout << logTransformImage(str,0) << endl;
30      cout << gammaImage(str,0) << endl;
31      cout << complementColorImage(str) << endl;
32  }
33
34
35  /*利用 OpenCV 读取图像。
36  *
37  * 具体内容
38  * 用打开 OpenCV 打开图像，并在窗口中显示
39  *
40  * */
41  bool loadImage(string &src, int flag = 1){
42      Mat img = imread(src, flag);
43      imshow("simpleOpenImage",img);
44      waitKey(0);
45      destroyAllWindows();
46      return true;
47  }
48
49  /*灰度图像二值化处理
50  *
51  * 具体内容
52  * 设置并调整阈值对图像进行二值化处理。
53  *
54  * */

```

```

55  bool binaryImage(string &src, int flag, int r){
56      Mat image, res;
57      image = imread(src, flag);
58      imshow("inputImage", image);
59
60      //用于二值化的图像
61      res = image.clone();
62      int rows = res.rows;
63      int cols = res.cols;
64
65      //二值化
66      for (int i = 0; i < rows; i++){
67          for (int j = 0; j < cols; j++){
68              auto gray = res.at<uchar>(i, j);
69              if (gray > r)    gray = 255;
70              else    gray = 0;
71              res.at<uchar>(i, j) = saturate_cast<uchar>(gray);
72          }
73      }
74      imshow("binaryImage", res);
75      waitKey(0);
76      destroyAllWindows();
77      return true;
78
79  }
80
81  /*灰度图像的对数变换
82  *
83  * 具体内容:
84  *   设置并调整 r 值对图像进行对数变换。
85  *
86  * */
87  bool logTransformImage(string &src, int flag, double r){
88      Mat image, res;
89      image = imread(src, flag);
90      imshow("inputImage", image);
91
92      //用于对数变换
93      res = image.clone();
94      int rows = res.rows;
95      int cols = res.cols;
96
97      //对数变换
98      for (int i = 0; i < rows; i++){
99          for (int j = 0; j < cols; j++){
100              auto gray = (double)res.at<uchar>(i, j);
101              gray = r * log((double)(1 + gray));

```

```

102         res.at<uchar>(i, j) = saturate_cast<uchar>(gray);
103     }
104 }
105     normalize(res, res, 0, 255, NORM_MINMAX);
106     convertScaleAbs(res, res);
107     imshow("logTransformImage", res);
108     waitKey(0);
109     destroyAllWindows();
110     return true;
111 }
112
113 /*灰度图像的伽马变换
114 *
115 * 具体内容:
116 * 设置并调整γ值对图像进行伽马变换。
117 *
118 * */
119 bool gammaImage(string &src, int flag, double r){
120     Mat image, res;
121     image = imread(src, flag);
122     imshow("inputImage", image);
123
124     //用于伽马变换的图像
125     res = image.clone();
126     int rows = res.rows;
127     int cols = res.cols;
128
129     //伽马变换
130     for (int i = 0; i < rows; i++){
131         for (int j = 0; j < cols; j++){
132             auto gray = (double)res.at<uchar>(i, j);
133             int a = 1;
134             gray = a * pow(gray, r);
135             res.at<uchar>(i, j) = saturate_cast<uchar>(gray);
136         }
137     }
138     normalize(res, res, 0, 255, NORM_MINMAX);
139     convertScaleAbs(res, res);
140     imshow("gammaImage", res);
141     waitKey(0);
142     destroyAllWindows();
143     return true;
144 }
145
146 /*彩色图像的补色变换
147 *
148 * 具体内容:

```

```

149     * 对彩色图像进行补色变换。
150     *
151     * */
152 bool complementColorImage(string &src, int flag){
153     Mat image, res;
154     image = imread(src, flag);
155     imshow("inputImage", image);
156
157     //用于补色变换的图像
158     res = image.clone();
159
160     //补色变换
161     Vec3b pixel, temp;
162
163     for(int i = 0; i < image.rows; i++){
164         for(int j = 0; j < image.cols; j++){
165             pixel = image.at<Vec3b>(i, j);           //RGB->CMY
166
167             /*
168             pixel里的三个通道是BGR，其补色是CMY色域的，变换关系如下：
169             C=255-R;
170             M=255-G;
171             Y=255-B;
172             */
173
174             temp[0] = 255 - pixel[2];           //C=255-R;
175             temp[1] = 255 - pixel[1];           //M=255-G;
176             temp[2] = 255 - pixel[0];           //Y=255-B;
177
178             res.at <Vec3b>(i, j) = temp;
179
180         }
181     }
182     imshow("complementColorImage", res);
183     waitKey(0);
184     destroyAllWindows();
185     return true;
186
187 }

```