高级图像处理与分析课程实验



高级图像处理与分析课程实验报告

学号 JG15225001

姓名 祁振东

日期 2016年3月24日

|  |  |
| --- | --- |
| **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** | [见附录。](#daima) |
| （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） |
| **实验中的问题** | 变换时是选择Mat类型的读取方法还是用指针的方法读取。  对于每个WithStep都要\*i才能得到每个像素的坐标。 |
| （包括在实验 |
| 中遇到的问 |
| 题，以及解 |
| 决问题的方 |
| 法） |
| **实验结果** | VS2013 express版本 opencv2  直接添加工程即可 |
| （实验完成后 |
| 的源码和打 |
| 包文件的说 |
| 明） |

int main()

{

logTran();//对数变换

gammaTran();//伽马变换

BS();//补色变换

HD();//灰度变换与二值化

waitKey(0);

return 0;

}

void logTran()

{

Mat src = cvLoadImage("..\\..\\pic\\h.jpg");

Mat dst = cvCreateImage(src.size(), IPL\_DEPTH\_8U, 1);

double c = 50.0;

int M = 0;

int N = 0;

if (src.empty()){

std::cout << "Src pic is empty\n" << std::endl;

return;

}

M = src.rows;

N = src.cols;

int j = 0;

double gray = 0.0;

for (int i = 0; i < M; i++){

for (j = 0; j < N; j++){

gray = (double)src.at<Vec3b>(i, j)[0];

gray = c\*log((double)(1 + gray));

(uchar)dst.at<uchar>(i, j) = (uchar)saturate\_cast<uchar>(gray);

}

}

imshow("对数变换",dst);

}

void gammaTran()

{

Mat src = cvLoadImage("..\\..\\pic\\h.jpg");

Mat dst = cvCreateImage(src.size(),IPL\_DEPTH\_8U, 1);

double gamma = 0.8;

double comp = 0.8;

int M = 0;

int N = 0;

if (src.empty()){

std::cout << "Src pic is empty" << std::endl;

return;

}

M = src.rows;

N = src.cols;

int j = 0;

uchar gray = 0;

for (int i = 0; i < M; i++){

for (j = 0; j < N; j++){

gray = (float)src.at<Vec3b>(i, j)[0];

gray = pow((gray + comp) / 255.0, gamma) \* 255;

(uchar)dst.at<uchar>(i, j) = saturate\_cast<uchar>(gray);

}

}

imshow("伽马变换，gamma = 0.8;comp = 0.8;", dst);

}

void BS()

{

Mat src = cvLoadImage("..\\..\\pic\\h.jpg");

Mat dst = cvCreateImage(src.size(), IPL\_DEPTH\_8U, 1);

if (src.empty()){

std::cout << "Src pic is empty\n" << std::endl;

return;

}

int M = src.rows;

int N = src.cols;

int j = 0;

uchar gray = 0.0;

for (int i = 0; i < M; i++){

for (j = 0; j < N; j++){

(uchar)src.at<Vec3b>(i, j)[0] = 225 - (uchar)src.at<Vec3b>(i, j)[0];

(uchar)src.at<Vec3b>(i, j)[1] = 225 - (uchar)src.at<Vec3b>(i, j)[1];

(uchar)src.at<Vec3b>(i, j)[2] = 255 - (uchar)src.at<Vec3b>(i, j)[2];

}

}

imshow("补色变换", src);

}

void HD()

{

Mat src = cvLoadImage("..\\..\\pic\\h.jpg");

Mat dst = cvCreateImage(src.size(), IPL\_DEPTH\_8U, 1);

if (src.empty()){

std::cout << "Src pic is empty\n" << std::endl;

return ;

}

int M = src.rows;

int N = src.cols;

int j = 0;

uchar gray = 0.0;

for (int i = 0; i < M; i++){

for (j = 0; j < N; j++){

gray = (uchar)src.at<Vec3b>(i, j)[0];

(uchar)src.at<Vec3b>(i, j)[1] = gray;

(uchar)src.at<Vec3b>(i, j)[2] = gray;

dst.at<uchar>(i, j) = gray>128 ? 255 : 0;

}

}

imshow("二值化图像", dst);

imshow("灰度图", src);

}