习题二

Code:VS2013+opencv330x64d

|  |
| --- |
| #include <opencv2/opencv.hpp>  #include <iostream>  using namespace std;  using namespace cv;  //方法一：利用opencv函数addWeighted（）进行融合  Mat imgFusion\_1(Mat srcImage, Mat logoImage)  {  vector<Mat> srcchannels;  vector<Mat> logochannels;  Mat srcTemp = srcImage.clone();  Mat logoTemp = logoImage.clone();  split(srcTemp, srcchannels);//分离通道  split(logoTemp, logochannels);    int imgcols = (int)((srcImage.cols - 1) / 2) - (int)((logoImage.cols - 1) / 2);  int imgrows = (int)((srcImage.rows - 1) / 2) - (int)((logoImage.rows - 1) / 2);    for (int i = 0; i <= 2; i++)//opencv公式处理  addWeighted(srcchannels[i](Rect(imgcols, imgrows, logoImage.cols, logoImage.rows)), 0.5,  logochannels[i], 0.5, 0., srcchannels[i](Rect(imgcols, imgrows, logoImage.cols, logoImage.rows)));    merge(srcchannels, srcTemp);//合并通道  merge(logochannels, logoTemp);    imwrite("Fusion\_1.jpg", srcTemp);  return srcTemp;  }  //方法二：逐像素处理融合  Mat imgFusion\_2(Mat srcImage, Mat logoImage)  {  vector<Mat> srcchannels;  vector<Mat> logochannels;  Mat srcTemp = srcImage.clone();  Mat logoTemp = logoImage.clone();  split(srcTemp, srcchannels);//分离通道  split(logoTemp, logochannels);  for (int i = 0; i <= 2; i++)  {  //逐像素处理时不是uchar型  srcchannels[i].convertTo(srcchannels[i], CV\_32F);  logochannels[i].convertTo(logochannels[i], CV\_32F);  }  int imgcols = (int)((srcImage.cols - 1) / 2) - (int)((logoImage.cols - 1) / 2);  int imgrows = (int)((srcImage.rows - 1) / 2) - (int)((logoImage.rows - 1) / 2);  //逐像素处理  int imgcolsLim = logoImage.cols, imgrowsLim = logoImage.rows;  for (int i = 0; i < imgrowsLim; i++)  for (int j = 0; j < imgcolsLim; j++)  {  srcchannels[0].at<float>(i + imgrows, j + imgcols) = (float)(0.5\*(srcchannels[0].at<float>(i + imgrows, j + imgcols)) + 0.5\*(logochannels[0].at<float>(i, j)));  srcchannels[1].at<float>(i + imgrows, j + imgcols) = (float)(0.5\*(srcchannels[1].at<float>(i + imgrows, j + imgcols)) + 0.5\*(logochannels[1].at<float>(i, j)));  srcchannels[2].at<float>(i + imgrows, j + imgcols) = (float)(0.5\*(srcchannels[2].at<float>(i + imgrows, j + imgcols)) + 0.5\*(logochannels[2].at<float>(i, j)));  }  for (int i = 0; i <= 2; i++)  {  srcchannels[i].convertTo(srcchannels[i], CV\_8U);  logochannels[i].convertTo(logochannels[i], CV\_8U);  }    merge(srcchannels, srcTemp);  merge(logochannels, logoTemp);  imwrite("Fusion\_2.jpg", srcTemp);  return srcTemp;  }  int main(int argc, char \*argv[])  {  Mat Fusion\_1, Fusion\_2;  Mat logoImage = imread("swust.jpg");  Mat srcImage = imread("campus.jpg");  if (!logoImage.data) { printf("logo.jpg input error！ \n"); return false; }  if (!srcImage.data) { printf("campus.jpg input error！\n"); return false; }    Fusion\_1 = imgFusion\_1(srcImage, logoImage);  //Fusion\_2 = imgFusion\_2(srcImage, logoImage);  namedWindow("swust");  imshow("swust", logoImage);  namedWindow("campus");  imshow("campus", srcImage);  //namedWindow("campus\_swust");  imshow("campus\_swust\_1", Fusion\_1);  //imshow("campus\_swust\_2", Fusion\_2);  waitKey();  return true;  } |

分别用OpenCV自带的函数以及逐像素两种方法进行实现，效果分别如图一、二。



图1.方法一效果图



图2.方法二效果图