ECE5/7382 – 计算机视觉基础

家庭作业-3

分配时间：9月18日

到期日： 9月29日

最高可能等级：55分（ECE5382），65分（ECE7382）

一般准则

* 请将您的回复作为 ZIP 文件上传到 Canvas，并遵循以下文件名约定
* ZIP 文件的名称 FIRSTNAME\_LASTNAME\_Homework3.zip
* 将名字替换为您的名字
* 将姓氏替换为您的姓氏

**不遵守文件名约定将导致扣分。**

* **在“可交付成果和问题”部分中**[**提供对每个问题的详细回答**](#_Deliverables_&_Questions)  **，包括支持数学论证**。

提交您的解决方案

请确保上传到画布的 ZIP 文件包含以下组件：

* 完成家庭作业的 Python 代码**-3.ipynb**
* 回答本文档中包含的所有问题。请将此文档转换为 PDF 并将其包含在 ZIP 存档中。

**对于 7382 名学生：** 将您的示例图像包含在 ZIP 存档中。

内容感知图像大小调整

作为此任务的一部分，您将开发 Python 例程，以实现 Shai Avidan 和 Ariel Shamir 的 SIGGRAPH 2007 论文“用于内容感知图像大小调整*的接缝雕刻”中描述的内容感知图像大小调整技术的简化版本*。

本文可在此处获得： <http://www.faculty.idc.ac.il/arik/SCWeb/imret/index.html>。请阅读本文，重点是第 3、4.1 和 4.3 节。

对于此作业，我们将注意力限制在缩小图像或缩小图像上。目标是实现该算法并分析其在多个图像上的性能。

### 接缝雕刻示例：有关示例，请参阅讲座幻灯片/视频。

### 基本理念

如讲座中所述，接缝雕刻试图通过去除以低梯度能量（相对平滑的区域）为特征的像素来调整图像大小，同时保留以高梯度能量为特征的更具信息性的像素。这项工作的核心是使用两种方法之一计算每像素梯度能量。

* 该术语 表示与要调整大小的图像关联的图像函数。
* 该术语 表示图像中的空间位置。
* 这些项分别表示 和 方向上的导数滤波的结果。OpenCV 函数 **Sobel（...）** 可用于计算这些项

#### 接缝

接缝雕刻算法通过识别以最小累积梯度能量为特征的单像素宽水平和垂直轨迹（称为接缝）来进行。

* 垂直接缝是从上到下的 8 个连接的像素路径，每行包含一个且仅包含一个像素。
* 最佳垂直煤层的特征是垂直方向上累积的梯度能量最小（跨行累积梯度能量）。
* 水平接缝是从左到右的 8 个连接的像素路径，每列包含一个且仅包含一个像素。
* 最佳水平煤层的特征是水平方向上累积的梯度能量最小（跨柱的累积梯度能量）。

#### 使用动态编程找到最佳垂直接缝

以下受动态规划启发的方法用于分两个阶段找到最佳垂直接缝。

#### 第一阶段：

在第一阶段，我们试图找到累积梯度能量图，该图指定穿过图像中每个像素的垂直接缝的能量/成本。下面公开的递归关系用于填充矩阵的条目。

下图是样本能量函数递归的各个步骤 。

包含日历的图片

自动生成说明

#### 第 2 阶段：

在第二阶段，通过回溯识别构成最佳垂直接缝的像素。M 最后一行中的最小值 表示最佳垂直接缝的结束。从这个像素位置开始，我们通过选择前一个像素中的 3 个最近邻像素中的最小值来识别最佳垂直接缝中的下一个像素。该过程迭代，直到我们到达图像的第一行，如下所示：

图形用户界面、应用程序、表格

自动生成说明

### 实施注意事项

* 强烈建议将接缝雕刻算法的输入图像转换为灰度。这可以使用 **im2double（...）**  来完成随附起始代码。
* 要计算梯度能量，您需要在每个像素处计算X和Y方向上的图像梯度。请使用 OpenCV 函数 **Sobel（...）** 函数来完成此任务。

|  |
| --- |
| 帮助： [图像渐变 — OpenCV-Python 教程测试版文档 （opencv24-python-tutorials.readthedocs.io）](https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_gradients/py_gradients.html) |

* 在计算矩阵之前，建议用Inf填充梯度能量图像（使用  **OpenCV中的copyMakeBorder**函数）。填充避免了在图像边界实现方程（5）时的复杂性。 有关更多详细信息，请参阅讲座幻灯片/视频。

|  |
| --- |
| 帮助[： OpenCV：为您的图像添加边框](https://docs.opencv.org/3.4/dc/da3/tutorial_copyMakeBorder.html)，  [Python OpenCV |cv2.copyMakeBorder（） 方法 - GeeksforGeeks](https://www.geeksforgeeks.org/python-opencv-cv2-copymakeborder-method/) |

### 要实现的 Python/OpenCV 代码

起始代码 **HWK3\_SeamCarving.ipynb** 旨在帮助您开始使用作业的第二部分。

* 该代码使用以下**接口调用**shrink\_width的 Python 函数

**[outp\_im] = 收缩宽度（inp\_im，像素数）**

该函数接受灰度输入图像**inp\_im**，以及 指定要雕刻的垂直接缝数的参数 numPixels。您将使用此函数将输入图像的大小减小规定的像素数。

* 函数 **shrinkWidth** 通过重复调用函数来迭代缩小图像的宽度

**[optVertSeamPath，EnergyFn，mtxM] = findVerticalSeam（im\_old）**

这应该接受输入图像**im\_old**。该函数应输出与图像im\_old相关的垂直接缝、能量函数 （**EnergyFn**） 和累积梯度能量 （**mtxM**）。

注意：该列的索引为零，符合 Python 的约定。

起始代码使用此信息将图像的宽度缩小 1 个像素。 “帮助程序函数”单元格中的第 29-32 行实现此功能。

* 作为确定最佳垂直接缝的一部分，您必须另外实现功能或代码片段来完成以下任务
  + 计算每个像素位置的能量函数。 使用方程（3）中定义的能量函数的定义。
  + 计算矩阵以实现接缝雕刻中的前向递归步骤。

### 可交付成果和问题

* 完整的源代码 **findOptVertSeam（）** （20分）

1. 问题-1 in **家庭作业-3.ipynb** （15 分）

* “*两次调用 findVerticalSeam（）****有什么区别****”*

请在此处提供您的回复

* “对 *np.transpose（）* ***的两个调用试图完成什么****？”*

请在此处提供您的回复

* “*为什么能量函数 E 在两次调用* ***findVerticalSeam（） 时保持不变****”*

请在此处提供您的回复

插入能量函数E的屏幕截图

插入矩阵M的屏幕截图用于垂直接缝

插入矩阵M的屏幕截图用于垂直接缝

* *“你认为第24、30行发生了什么？为什么变量在两个调用之间颠倒了？*

请在此处提供您的回复

* *“评论为什么你认为接缝看起来像原始图像的内容一样。*

请在此处提供您的回复

1. 问题-2 （10 分）

在湖上调用 **shrinkWidth** 函数.jpg numPixel = 100 （换句话说，将宽度缩小 100 像素）。

插入原始图像的屏幕截图

插入智能调整大小的屏幕截图

1. 问题-3 in **家庭作业-3.ipynb** （10分，各2分）

“*结果对你来说是否合理。如果不评论你认为似乎正在发生的事情。*

请在此处提供您的答案

插入能量函数E的屏幕截图

插入矩阵M的屏幕截图用于垂直接缝

插入矩阵M的屏幕截图用于垂直接缝

插入数字像素= 50的智能调整大小的屏幕截图

问题4：仅限ECE-7382学生 （10 分）

尝试在您选择的图像上进行接缝雕刻。

插入原始图像的屏幕截图

插入矩阵M的屏幕截图用于垂直接缝

插入矩阵M的屏幕截图用于垂直接缝

插入能量函数E的屏幕截图

插入数字像素= 50的智能调整大小的屏幕截图

确认

接缝雕刻任务改编自多个来源：德克萨斯大学奥斯汀分校的 CS 376、华盛顿大学的 CSE 455 和斯坦福大学的 CS 131。