

图形学实验 PA5: 接缝裁剪

指导教师: 胡事民 助教: 方晓楠

2019 年 12 月 1 日

1 实验综述

本实验要求实现课上所讲的接缝裁剪 (Seam Carving) 算法, 对图像的尺寸进行修改。具体来说, 对于一张给定的输入图像, 程序应当支持删除其中的若干行或者若干列。你只需要在给定的代码框架中补充接缝裁剪算法的关键部分。我们将在 4 个测例上评测你的代码。

2 细节说明

接缝裁剪算法由 Shai Avidan 和 Ariel Shamir 在 SIGGRAPH 2007 的论文 Seam Carving for Content-Aware Image Resizing 中提出。该论文希望解决的问题是根据图像的内容特征动态地调整图像长宽比, 使得图像能呈现在不同长宽比的显示设备上。

以下我们以减小图像的宽度为例。形式化的定义如下: 给定一幅 $n \times m$ 的图像 I , 我们希望减少它的列数, 得到一幅 $n \times m'$ 的图像 I' , 其中 $m' < m$ 。

调整图像长宽比有两种简单的方式: 等比例缩放 (scaling) 会使图像中物体的比例失调, 而直接裁剪 (cropping) 出图像中一块 $n \times m'$ 的子矩形则会丢失图像中的信息。

接缝裁剪算法采用迭代式的方法: 每一轮迭代中在每一行各删除一个像素, 使得图像的宽度减 1, 迭代 $m - m'$ 次就使得图像达到了目标宽度。

如何确定每一轮删除哪些像素? 主要有两个标准。其一是被删除的像素视觉上的重要性不高, 其二是被删除的像素应该在空间上是连续的。视觉上的重要性采用下面的能量定义:

$$e_1 = \left| \frac{\partial I}{\partial x} \right| + \left| \frac{\partial I}{\partial y} \right|$$

每一个像素的能量为该点梯度的模长, x, y 分别表示图像的列坐标和行坐标。我们采用 L_1 范数来计算模长。导数采用后向差分进行估计, 即

$$\frac{\partial I}{\partial x}(y, x) = I(y, x) - I(y, x - 1)$$

$$\frac{\partial I}{\partial y}(y, x) = I(y, x) - I(y - 1, x)$$

请注意一般在图像处理的程序中第一维是行坐标 y , 第二维是列坐标 x 。为了保证被删除像素的连续性, 我们要求相邻两行被删除的像素列坐标相差不超过 1。被删除像素的集合定义为:

$$\mathbf{s}^x = \{s_i^x\}_{i=1}^n = \{(i, x(i))\}_{i=1}^n, 1 \leq x(i) \leq m, |x(i) - x(i - 1)| \leq 1$$

这样定义出的集合 \mathbf{s}^x 称为一个接缝 (seam)。

在每一轮迭代的过程中, 我们希望找到一条能量最小的接缝并将其删除, 即求解

$$\mathbf{s}^* = \min_{\mathbf{s}} E(\mathbf{s}) = \min_{\mathbf{s}} \sum_{i=1}^n e(I(s_i))$$

这个问题可以通过动态规划 (递推) 的方法求解, 定义 $M(i, j)$ 为从第 1 行出发, 以 (i, j) 为终

点的路径所能取得的最小能量总和，则有状态转移方程：

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

在递推过程中同时记录下每次转移所选择的前序位置，就可以倒推出最优接缝上的所有点：

若 $M(i, j) = e(i, j) + M(i-1, j-1)$ ，记 $pred(i, j) = -1$

若 $M(i, j) = e(i, j) + M(i-1, j)$ ，记 $pred(i, j) = 0$

若 $M(i, j) = e(i, j) + M(i-1, j+1)$ ，记 $pred(i, j) = 1$

如果要减小图片的行数，只需把上述算法定义中的行和列的定义颠倒一下，同样可以用动态规划的方法求解出最优接缝。

3 框架代码说明

3.1 环境配置与编译

我们推荐使用带有 CMake 套件的 Ubuntu 系统进行编程，Windows 10 下可以使用 Ubuntu Subsystem，如果你遇到了编译问题，请先尝试自行解决。我们的框架代码没有任何外部依赖，请在包含有 run_all.sh 的文件夹下打开终端，并执行：

```
bash ./run_all.sh
```

这段脚本会自动设置编译，并在 4 个测例上运行你的程序。你的程序最终会被编译到 bin/文件夹中，而输出的图片位置在 output/文件夹中。

3.2 实现步骤

本次实验需要用到 vecmath 库及 Image 类。注意本次使用的 Image.hpp 和 Image.cpp 文件与之前有所不同，请使用本次提供的版本。

本次作业所有需要新填写的地方都被标记了 TODO (PA5)，请全文查找该字样，每实现完成一个功能，你就可以去掉一个 TODO。你需要实现的主要的函数是 Solver.hpp 中的 dist 函数和 Solver::cutOneColumn, Solver::cutOneRow 函数，分别对应的功能是计算两个向量之间的 L1 距离，删除图像中的一列以及删除图像中的一行。你需要了解各个变量的含义，以及他们怎样和具体的数学公式对应。请注意 Solver 类中行列坐标的顺序与 Image 类中的顺序是相反的，但在所有代码中行坐标均以变量 y 表示，列坐标均以变量 x 表示。

4 测试用例

为了测试代码是否正确无误，我们构建了以下 4 个测试用例，你也可以使用其他的 bmp 格式图片进行自我测试。但我们在检查作业的时候将主要检查这 4 个样例的输出结果。



5 提交说明

请将你的代码放入 `code` 文件夹（无须包含可执行文件和输出结果），和一个 `REPORT.pdf` 文档一起打包成 `zip` 文件提交至网络学堂。

在 `README.pdf` 文档中主要回答以下问题：

- 代码框架中更新图像梯度的步骤可以做怎样的优化？
- 如果要让用户能在交互界面上动态地更改图片的宽度并实时展示结果，应该对程序作哪些改动？
- 你在完成作业的时候和哪些同学进行了怎样的讨论？是否借鉴了网上/别的同学的代码？
- (可选) 你对本次编程作业有什么建议？文档或代码中有哪些需要我们改进的地方？

本次作业的 `Deadline` 以网络学堂为准。迟交的同学将得到一定的惩罚：晚交 3 天内分数将降低为 80%，3 天以上 1 周以内分数降为 50%，迟交一周以上的同学分数为 0。在检查你的作业时，助教的自动化脚本会在 `code` 目录下运行 `run_all.sh` 文件，不能成功编译者可能会被酌情扣分。