

# Seam Carving 实验报告

2015011308 计 53 唐适之

## 效果

本实验中实现了 Seam Carving 算法，包括长、宽两个方向上的放大、缩小，以及将用户指定的区域移除。

输入见 input 文件夹，输出见 output 文件夹。编号为 1~6 的是给定图片，编号为 a~d 的是自选图片。对于每个输入图片，在 output 文件夹中包含各图的 Seam Carving 结果和 seam 位置两张图片。若为缩小，seam 图中标出的是被删除的接缝；若为放大，seam 图中标出的是被复制的接缝。各图片进行的操作如下：

1. 双向缩小
2. 双向放大
3. 宽度缩小
4. 宽度缩小
5. 移除图片中的相机并保持大小不变
6. 双向缩小
  - a. 宽度缩小
  - b. 高度放大
  - c. 宽度缩小且高度放大
  - d. 移除图片中的人并保持大小不变

## 实现

本实验中使用的是 Sobel 算子，采用了  $7 \times 7$  大小的区域估计偏导数。实验表明，若采用过小的区域，会导致图片中的直线变弯，采用  $7 \times 7$  的大小能在一定程度上缓解此问题。

main.cpp 中的 dp 函数用于解算最小能量的纵向 seam。函数内分为三部：1. DP 计算累积能量；2. 找出能量最小的路径；3. 从图片中移除该路径，并更新辅助数组，例如每个像素对应于原图中的位置。

附加功能如下：

1. 放大。main.cpp 中的 carveHorInc 函数用于执行放大。若要放大 k 列，则先执行删除 k 列的操作，但实际上并不删除而是复制这 k 列。若 k 较大，为避免图片失真，每次最多只放大总宽度的  $1/3$ ，反复执行此操作，直到放大到指定大小。
2. 双向缩放。将图片转置即可将宽度缩放的函数用于高度缩放。Seam Carving 的原论文指出：对于每一步应该进行宽度缩放还是高度缩放，理论上存在一种最优方案，但找出这种方案需要  $O(n^4)$  的时间复杂度，其中 n 是图片边长，效率过低不可接受。故本实验中采用的是先进行宽度缩放、再进行高度缩放的策略。见 main.cpp:167~170。
3. 对象移除。本程序将会弹出一个 GUI 窗口让用户标记移除区域，先将此区域标记为优先并执行删除操作，然后再缩放到用户指定大小。GUI 见 ui.cpp，优先删除见 dp 函数中对 delMask 变量的处理，以及 delMasked 函数。