

seam carving 图像压缩实验报告

软件 13 杨楠 2021010711

摘要：对接缝裁剪（seam carving）图像压缩算法进行了理论分析，并设计了具体程序，实现这一算法。程序的功能是将一张图片的行、列尺寸各压缩为原来的一半。使用不同尺寸和内容的图片进行测试，得出，该算法在对图片主要内容的保存程度相对较好。对于尺寸较大的图片，程序运行时间较长些。

1 实验环境

使用 C++11 语言，用 OpenCV 4.5.2 编写。在 Windows 系统下编译运行。

2 算法分析

接缝裁剪（seam carving）是图像压缩的一种算法。通过寻找一条对图像视觉效果破坏程度最低的“接缝”（seam），使得图像变窄一条像素。重复此过程，即可达到压缩图像的效果。对于接缝的要求是：选择接缝时，每一行像素中都选取一个像素点，且相邻行所选的像素点位于同一列或者相邻列。

使用动态规划的算法求解接缝如下：从上到下逐行选取像素点，对于一张 $m \times n$ 的图像，假定每个像素 $A[i, j]$ 都已经求得它的破坏度 $d[i, j]$ ，令 $s[i, j]$ 为，选到第 i 行第 j 列的像素点时，当前已经选择的像素点的破坏度的总和和所能达到的最小值。显然有 $s[1, j] = s[d, j], 1 \leq j \leq n$ 。如果再令 $s[i, 0] = s[i, n+1] = \infty, 1 \leq i \leq m$ ，那么 s 的最优子结构满足：

$$s[i, j] = \min\{s[i-1, j-1], s[i-1, j], s[i-1, j+1]\} + d[i, j]$$

那么整张图片的破坏度最小的接缝，其破坏度的值为 $\min\{s[m, j], 1 \leq j \leq n\}$ 。

初始化之后，通过两层循环递推，即可求得接缝破坏度最小值，复杂度为 $\Theta(m+n+mn) = \Theta(mn)$ 。在递推时做好标记，可以获得接缝的具体位置，回溯时的时间复杂度为 $\Theta(n)$ 。所以获取一条接缝的时间复杂度为 $\Theta(mn)$ 。

3 实验设计

本实验编写 C++ 程序，通过调用 OpenCV 相关库，进行 seam carving 操作。OpenCV 使用 Mat 类对图片进行处理。

对于破坏度即能量值的计算，本程序的是通过使用 Sobel 算子，得出该点在 x 轴和 y 轴的梯度，然后取绝对值求和[1]，即：

$$e(I) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right|$$

在具体代码实现上，OpenCV 库中有 Sobel() 函数可直接调用，返回值是所设梯度的 Sobel 图，即 Mat 类型。

对于彩色图，即 RGB 三通道的图，在算出在 x 和 y 梯度上的 Sobel() 值，取绝对值求和之后，由于所得的 Sobel 图是彩色三通道的，所得的某一像素点的能量值是分为 BGR 三个分量的，需要将其转为灰度图。对于灰度图，则直接运算求和即可。

在动态规划的算法中，使用了分情况来处理可能的边界条件。裁剪 seam 时，不是直接删除或者覆盖原图片的接缝像素，而是新建一个新的 Mat 类，通过事先标记，将原图中不是接缝的像素赋值过去。

用 `cut_seam()` 函数中实现了裁剪单条接缝的算法。由于裁剪之后，新图片的破坏度即能量值会发生变化，所以需要重新计算能量值。

将图片的行和列的尺寸都减小一半的具体实现为：获取原图片的行 `row` 和列 `col`。先执行 `col/2` 次 `cut_seam()` 函数，再将图像进行转置，再执行 `row/2` 次 `cut_seam()` 函数，再将图像转置复原。总的时间复杂度为 $\Theta(\max\{m, n\}mn)$ 。

程序运行时，输入图片的路径（不包含引号）（绝对或者相对路径）。如果图片路径不符，那么会输出报错，结束程序。否则，对该图片进行裁剪。裁剪完毕后，会输出所用时间（单位：秒），输出” Done!” 提示，生成并打开裁剪后的图片。关闭图片后，程序结束。

生成的图片名的格式为：时间戳. xxx（末尾扩展名与原图片相同），从而保证每次生成的图片不重名。

4 结果分析

运行程序，分别将长为 640 像素的图片进行压缩，所得图片如下。



ship_640.jpg (640*425)



balloon_640.jpg (640*427)



tree_640.jpg (640*384)





cat_640.jpg (640*427)

上述图片裁剪所用时间，大致都是 5 秒左右。可以看到，该算法对于图片的裁剪压缩比较自然，保留了图片比较重要的内容，并且处理地比较自然。对于 balloon 图片，两气球中间的蓝天被裁剪了些，两个热气球的形状相对完好。而对于 tree 图片，由于树枝树叶、云朵比较复杂，裁剪后相对突出。此外，该算法对于风景中有比较独特的物体的图片裁剪效果比较好，而对于动物以及人像处理就有些不足，比如样例中 cat 图片的处理。

对于内容相同、尺寸不同的图片进行压缩，在时间上有比较明显的差别。比如，裁剪 640*425 的 ship 图片，所用时间为 5 秒，而裁剪 1280*850 的 ship 图片，所用时间为 38 秒，与程序时间复杂度 $\Theta(\max\{m, n\}mn)$ 基本符合。

参考文献

- [1] Shai Avidan and Ariel Shamir. 2007. Seam carving for content-aware image resizing. ACM Trans. Graph. 26, 3 (July 2007), 10-es.
<https://doi.org/10.1145/1276377.1276390>