

Seam Carving 图像压缩实验报告

软件 21 杨朗 2021012224

2023 年 4 月 13 日

1 实验简述

本次实验是对 seam carving 算法的实现，seam carving 是一种基于内容的图像压缩算法，其核心思想是通过动态规划找到图像中能够被删除的最小能量的一条路径，从而达到图像压缩的目的。实验实现的程序基本目标是打开一张图片，然后使用 seam carving 算法将 $m \times n$ 大小的图片压缩成 $\frac{m}{2} \times \frac{n}{2}$ 大小。同时，本次实验编写了相应的 GUI 界面，使得用户可以通过 GUI 界面进行操作，还实现了一些其他功能，比如可以选择不同的能量函数，自定义连续切除接缝数量，调整压缩方向等。最后，实验还比较了不同能量函数对压缩效果的影响。

1.1 实验环境

- 操作系统：macOS Monterey 12.3
- QT 版本：5.15.8
- 编译器：clang++ 13.0.0
- 编译选项：-std=c++11 -O

2 算法细节

2.1 相关概念

1. 能量函数 $E(x, y)$: $E(x, y)$ 表示图像中坐标为 (x, y) 的像素点的能量， $E(x, y)$ 越大，表示该像素点越不容易被删除。
2. 能量矩阵 E : E 是一个 $m \times n$ 的矩阵，其中 $E[i, j]$ 表示图像中坐标为 $[i, j]$ 的像素点的能量。
3. 最小能量路径 P_{min} : P_{min} 是一个 $m \times n$ 的向量，其中邻接包含算法所求的最小能量接缝的像素。

2.2 算法流程

Algorithm 1 seam carving 算法流程

- 1: 输入: 图片 I , 切除一条接缝
 - 2: 输出: 切除一条接缝后的图片 I'
 - 3: 计算每个像素的能量得到能量矩阵 E
 - 4: 动态规划计算最小能量路径 P_{min}
 - 5: 删除 I 中最小能量路径 P_{min} 中的像素点
 - 6: 输出: 切除一条接缝后的图片 I'
-

2.3 算法实现

2.4 动态规划求解接缝

计算能量矩阵后, 我们可以通过动态规划求解最优的接缝, 即总能量最小的接缝。

定义一个累积能量矩阵 M , $M[x, y]$ 表示: 从第一行逐行选择到第 y 行, 且第 y 行选择的点为 (x, y) 时, 所累积的能量值的最小值。显然, M 具有最优子结构:

$$M[x, y] = \min(M[x-1, y-1], M[x, y-1], M[x+1, y-1]) + E[x, y]$$

定义一个最小能量路径矩阵 P , $P[x, y]$ 表示: 从第一行逐行选择到第 y 行, 且第 y 行选择的点为 (x, y) 时, 所累积的能量值的最小值的最小能量路径上 (x, y) 的前一个点的横坐标。于是最小能量路径矩阵 P 可以通过递推求解:

$$P[x, y] = \begin{cases} x-1, & \text{if } M[x-1, y-1] < M[x, y-1], M[x+1, y-1] \\ x, & \text{if } M[x, y-1] < M[x-1, y-1], M[x+1, y-1] \\ x+1, & \text{if } M[x+1, y-1] < M[x-1, y-1], M[x, y-1] \end{cases}$$

可以通过一次动态规划求出 M, P , 且该动态规划算法时间复杂度为 $\Theta(mn)$ 。

若图像大小为 $m \times n$, 则所有可能接缝中, 能量最小的接缝的最后一个点的横坐标为

$$x_{min} = \operatorname{argmin}_{x \in [1, n]} M[x, m]$$

于是可以通过 x_{min} 和 P 逆向递推求解最小能量路径 P_{min} 。此过程时间复杂度为 $\Theta(mn)$ 。

2.5 能量函数

2.5.1 梯度类能量函数:

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

通过卷积算法计算图像在 x, y 两个方向的偏导, 然后将绝对值求和近似得到梯度作为能量。[1] 本次实验实现了采用了多种不同的一阶微分算子作为卷积核, 包括 Sobel 算子、Prewitt 算子、Scharr 算子、Robert 算子等。

同时, 本次实验还额外使用了二阶微分算子, 即 Laplacian 算子。该算子得到的能量函数同样基于梯度, 定义略有不同:

$$E(x, y) = \left| \frac{\partial^2 I}{\partial x^2} \right|$$

本次实验编写的卷积算法时间复杂度为 $\Theta(mn)$, 其中 m, n 分别表示图像的宽度和高度。

2.5.2 Forward 能量函数:

$$E(x, y) = \min\{C_L(x, y), C_U(x, y), C_R(x, y)\};$$

其中, C_L, C_U, C_R 分别表示: 接缝连接 (x, y) 和 $(x - 1, y - 1)$ 带来的代价、接缝连接 (x, y) 和 $(x, y - 1)$ 带来的代价、接缝连接 (x, y) 和 $(x + 1, y - 1)$ 带来的代价:¹

$$C_L(x, y) = |I(x + 1, y) - I(x - 1, y)| + |I(x, y - 1) - I(x - 1, y)|$$

$$C_U(x, y) = |I(x + 1, y) - I(x - 1, y)|$$

$$C_R(x, y) = |I(x + 1, y) - I(x - 1, y)| + |I(x, y - 1) - I(x + 1, y)|$$

以 $C_L(x, y)$ 为例解释代价: 接缝连接 (x, y) 和 $(x - 1, y - 1)$, 那么去除接缝后。原来 $(x - 1, y)$ 这个像素点会被移动到 $(x, y - 1)$ 下方和 $(x + 1, y)$ 的左方, 贡献出新的能量代价, 即 $|I(x + 1, y) - I(x - 1, y)|$ 和 $|I(x, y - 1) - I(x - 1, y)|$ 。

不同于梯度类能量函数, Forward 能量函数是一种基于像素点之间的相似性来计算能量的能量函数。这种能量函数的设计思想是考虑移除接缝引起周围的像素点的变化, 而不是简单地考虑接缝能量最小化。按照此种能量函数的结果寻找接缝, 可以使得接缝被移除后拼接在一起而产生的新的差异尽可能的小, 从而使得图像的拼接效果更好, 物体的畸变得到改善。[2]

显然, Forward 能量函数的计算仍为时间复杂度为 $\Theta(mn)$ 。

3 实验结果与分析

3.1 基本效果

程序 GUI 及运行效果如图1所示。可以看到, 尽管图像移除了不少像素, 但人物和山这两类主体物体的形状基本保持不变。如图中按钮所示, 本次实验的程序还实现了更换能量函数或算子, 显示能量图, 自定义裁剪数量、方向等功能。

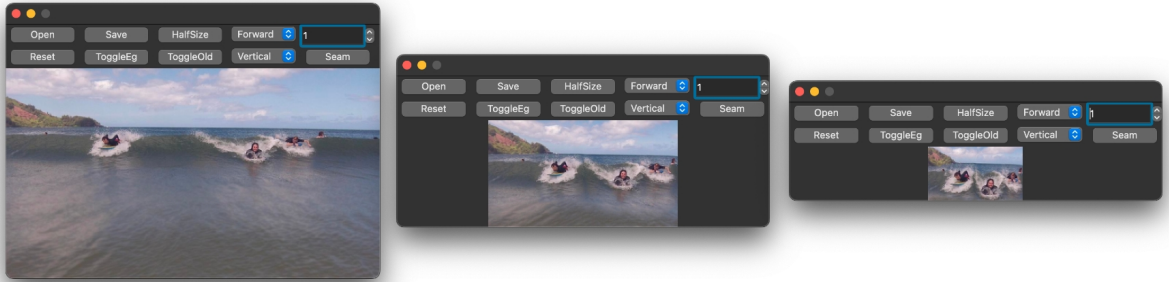


图 1: 左侧程序界面中为原图, 中间为进行一次长宽折半缩放的效果, 右侧为进行两次长宽折半缩放后的图。

¹ 此处的坐标表示是基于从上到下进行 dp 寻找接缝的前提。

3.2 畸变对比

根据前文分析，Forward 能量函数的设计思想是考虑移除接缝引起周围的像素点的变化，而不是简单地考虑接缝能量最小化。于是实验对比了 Forward 能量函数的畸变效应与梯度类能量函数的畸变效应，如图4所示。

在4中，第二列的企鹅的脸部和翅膀发生偏移，而第三列的企鹅的脸部和翅膀则没有发生偏移；第二列的花茎发生了明显的折叠，而第三列的花茎则没有发生折叠；第二列的黑色裤子失真，而第三列的黑色裤子则未受影响；第二列的板凳的立柱明显变细，图形边缘的暗角变得不自然，而第三列的板凳形状完好，暗角自然；第二列的花叶的透视关系变化，而第三列的花的透视关系基本不变。

经过实验验证，在几乎所有情况下使用 Forward 能量函数处理的结果都要比梯度类能量函数的结果更自然。² 因此 Forward 的设计思想是有效的，能够使得图像的拼接效果更好。

不过，在某些情况下，有时候无论能量函数是什么，都会出现一些不可避免的畸变。比如图3中，左侧为原图，右侧为基于 Forward 能量压缩的效果。注意到，即使是用抑制畸变效果最好的 Forward 能量函数，也无法完全避免马主体能量过低被过度裁剪的问题。

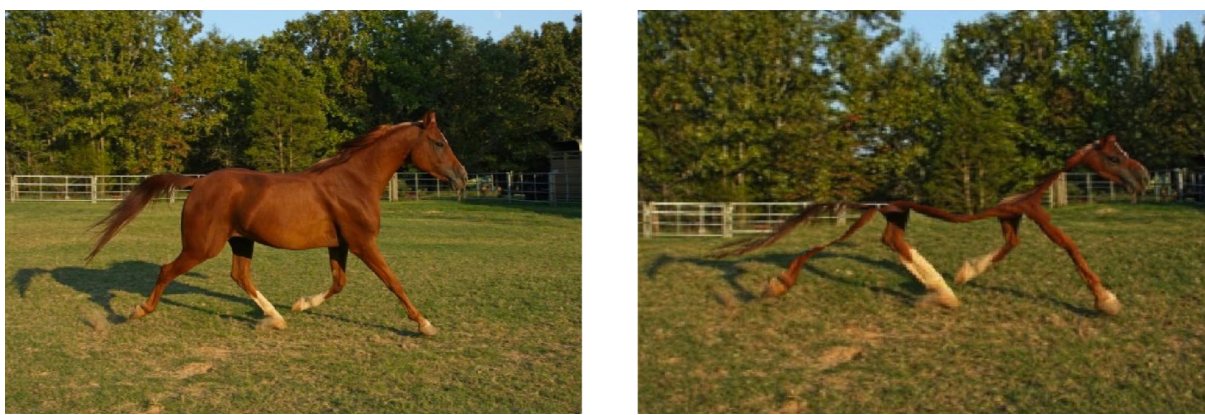


图 3: 左侧为原图，右侧为基于 Forward 能量压缩的效果。

3.3 运行时间

3.3.1 能量图分析

观察图2中的能量图，可以发现梯度类能量函数的能量图中，能量值总是集中在板凳条的轮廓，而板凳两侧空处能量小，其次小的是两个立柱。

而 Forward 能量函数的能量图中，这些边缘的能量值并不总是较大，比如在板凳中间构成的重复的横条上，Forward 能量函数认为这里引起的能量变化较小。

于是在梯度类能量函数中，优先去除两边和立柱，发生了明显的畸变。而在 Forward 能量函数中，也会时常考虑去除板凳中间的横条，板凳的形状保持完好。这便是图4中第四行的板凳图压缩产生明显好坏差别的原因。

²图4中仅用 Sobel 算子的结果代表所有梯度类能量函数的结果，因为实验中所有的基于梯度的能量函数的处理结果的畸变程度都相差不大。

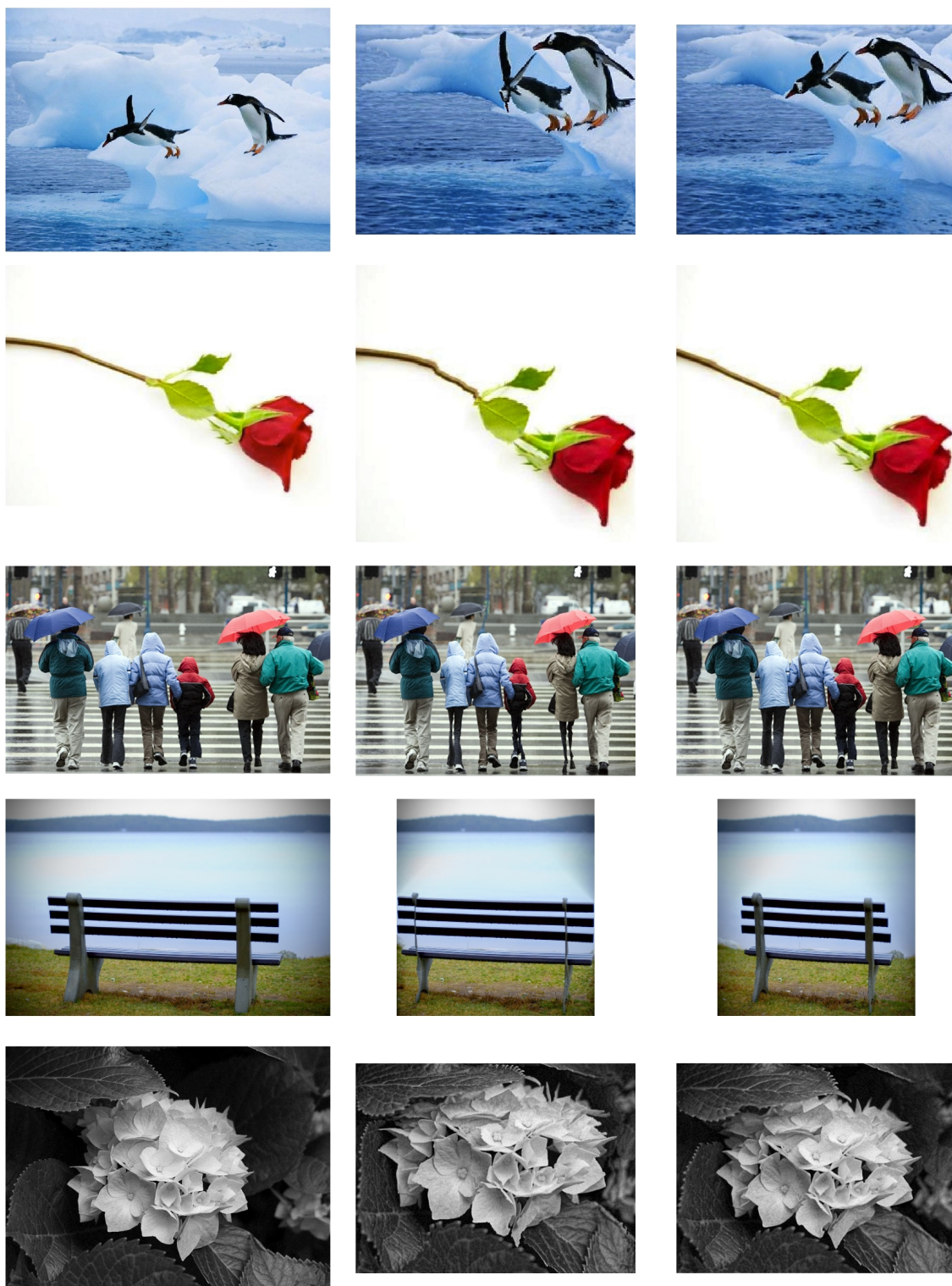


图 2: 基于不同能量函数的 seam carving 算法产生的畸变对比。其中第一列为原图, 第二列为基于 (Sobel 算子) 梯度能量压缩的效果, 第三列为基于 Forward 能量压缩的效果。

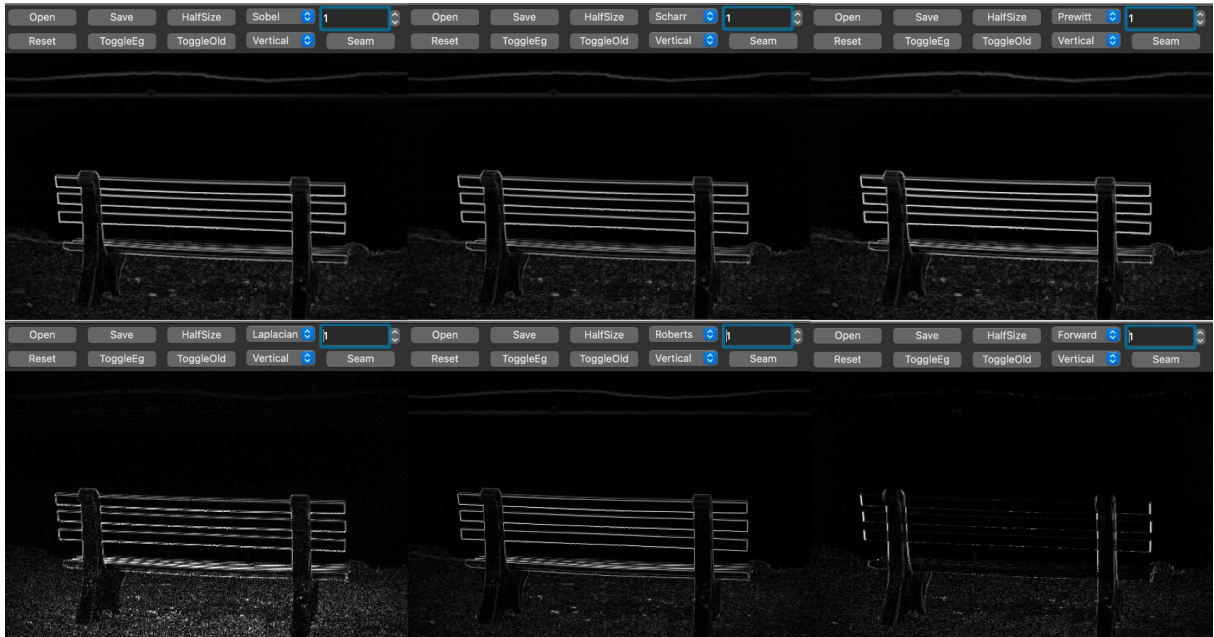


图 4: 六种不同能量函数（包括算子不同）下的能量图对比。灰度值大的区域代表能量值大，灰度值小的区域代表能量值小。

3.4 运行时间

图5中为本次实验的程序在进行长宽折半压缩时的运行时间测量输出。每一行输出的是每移除一条缝的三个过程的运行时间，分别是能量计算、缝的求解、缝的移除。左侧为刚开始运行时的输出，右侧为运行将要结束时的输出。可以发现程序运行的主要耗时体现在能量计算上，且在压缩快要结束时，能量计算和缝得求解的耗时大约变为刚开始的 $\frac{1}{4}$ 这与时间复杂度分析中的预期一致：时间复杂度为 $\Theta(mn)$ 。

calcEnergy: 25562us	calculateSeam: 5671us	removeSeam: 5695us	calcEnergy: 7107us	calculateSeam: 1604us	removeSeam: 2373us
calcEnergy: 25662us	calculateSeam: 5198us	removeSeam: 5632us	calcEnergy: 6948us	calculateSeam: 1612us	removeSeam: 2397us
calcEnergy: 24943us	calculateSeam: 5421us	removeSeam: 5563us	calcEnergy: 7065us	calculateSeam: 1663us	removeSeam: 2085us
calcEnergy: 23948us	calculateSeam: 4893us	removeSeam: 5450us	calcEnergy: 7031us	calculateSeam: 1661us	removeSeam: 2362us
calcEnergy: 25412us	calculateSeam: 5150us	removeSeam: 5587us	calcEnergy: 6992us	calculateSeam: 1631us	removeSeam: 2373us
calcEnergy: 25034us	calculateSeam: 4894us	removeSeam: 5574us	calcEnergy: 6943us	calculateSeam: 1674us	removeSeam: 2375us
calcEnergy: 23191us	calculateSeam: 4872us	removeSeam: 5269us	calcEnergy: 6966us	calculateSeam: 1690us	removeSeam: 2173us
calcEnergy: 21156us	calculateSeam: 4886us	removeSeam: 4801us	calcEnergy: 6866us	calculateSeam: 1684us	removeSeam: 2349us
calcEnergy: 24263us	calculateSeam: 4941us	removeSeam: 5466us	calcEnergy: 6869us	calculateSeam: 1570us	removeSeam: 2305us
calcEnergy: 25096us	calculateSeam: 4905us	removeSeam: 5625us	calcEnergy: 6864us	calculateSeam: 1571us	removeSeam: 2185us

图 5: ““

4 结论

本次实验较好地复现了 seam carving 算法，该算法能够有效地在对图像进行压缩的同时保持图像主要内容的完整性。同时实验还实现了基于 Forward 能量函数的优化方法。发现该方法能够大幅抑制 seam carving 算法产生的物体畸变，压缩结果的自然性明显强于基于梯度能量函数的 seam carving 算法。不足的是，在某些情况下 seam carving 总会出现一些难以避免的畸变。

参考文献

- [1] AVIDAN, S. AND SHAMIR, A. (2007). *Seam carving for content-aware image resizing*. In ACM SIGGRAPH 2007 papers (SIGGRAPH '07). Association for Computing Machinery, New York, NY, USA, 10–es. <https://doi.org/10.1145/1275808.1276390>.
- [2] AVIDAN, S. AND SHAMIR, A. (2008). *Improved Seam Carving for Video Retargeting*. ACM Trans. Graph. 27, 3 (August 2008), 1–9. <https://doi.org/10.1145/1360612.1360615>