

结合多种方法的 Seam Carving 实现图像缩放

李春良

(北京理工大学, 计算机学院, 学号 1120193226, 班级 07111908)

摘要: 本文使用 Seam Carving 方法实现图像缩放(改变图片长宽比、图像重定向), 成功复现了改进最小累积能量函数的 Improved Seam Carving 方法, 并尝试结合了梯度法、Canny 边缘检测、和显著性区域检测的残差谱(Spectral Residual)算法来改进能量函数, 探究了这些结合方法对处理效果的影响。

关键词: Seam Carving; 图像缩放; 梯度法; 边缘检测; 显著性区域检测

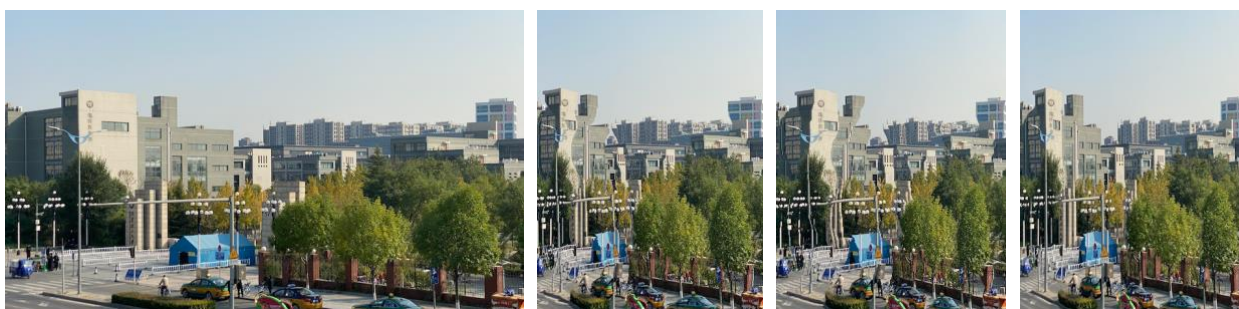


Figure 1 左 1 原图, 左 2 Seam Carving、Removal Mode, 能量函数 e_1 , 左 3 Improved Seam Carving、Removal Mode, 能量函数 e_4 (残差谱算法), 左 4 Improved Seam Carving、Removal Mode, e_3 (Canny 边缘检测)。

1 实验背景

缩放是一种常见的图像处理操作, 图像的智能缩放方法可以允许图片在不同尺寸的显示设备上呈现较好的视觉效果。本实验的目的是改变图片的长宽比, 并且尽可能考虑图像的差异化处理。

传统的图像缩放技术主要包括拉伸(Scaling)、裁剪(Cropping)和边缘填充黑色(Letter Boxing)^[1], 这些方法没有考虑到图像内容的差异, 因此不能保证图像中的感兴趣区域被保留和没有明显的视觉失真。

差异化缩放方法通常考虑图像内容, 可以保证感兴趣区域被保留。Seam Carving 方法^[2-3]的重要性图(能量图)仅仅是 L_1 范数下的图像梯度值, 这一方法通过删除或复制相邻接的一条穿过图像重要性最低区域的像素带(裁剪线)来实现图像缩放, 裁剪线通过动态规划的方式计算。Shift-maps 方法^[4]计算图的割, 一次移除一整个物体而不是每次删除单条裁剪线。这一算法的能量图同时考虑像素间颜色和梯度的差异性。Nonhomogeneous warping^[5]配合人脸检测器使用 L_2 范数计算重要性图, 通过优化一个单方向的变形(warping)函数, 对图像的列进行非同质性的挤压, 从而减小图像宽度。它的效果使得图片区域最大变形量与重要性成正比。Scale-and-stretch^[6]方法把图像划分为若干个四边形网格, 通过计算最优缩放因子, 对视觉上认为重要的区域产生尽可能小的形变, 同时允许不重要的区域有一定的扭曲变形。此外, 还有 Energy-based deformation^[7], Multi-operator^[8]等方法。这些方法中, 除了 Shift-maps 方式实现差异化缩放需要指定 Shift-map 的方向、显著性图和边界条件, 在人为选择下通常有较好的效果之外, 其它方式都无需人工干预。

本文选择较为容易实现的 Seam Carving 方法以实现图像缩放, 同时试图对该方法的能量函数进行改进以充分挖掘其潜力, 主要实验效果如图 1 所示。本文主要实现了移除、插入、混合三种方式的图像缩放, 实现了基本 Seam Carving 方法、改进了 Seam 查找的 Seam Carving 方法, 进行了结合梯度法、Canny 边缘检测、残差谱的显著性区域检测算法的对能量函数的改进。

2 Seam Carving 的基本原理

Seam Carving 算法的基本思想^[2]是去除图像中的不重要区域，区域的重要程度由能量函数定义，通过不断插入或者移除能量最低的裁剪线 Seam 来实现图像尺寸的缩小或增大。

2.1 基本算子

Seam Carving 的目标是以保留感兴趣区域的方式来移除像素，问题就在于如何选择要移除的像素。在直觉上当然是直接移除不会被注意到的像素。因此 Seam Carving 原方法定义的能量函数为

$$e(\mathbf{I}) = \left| \frac{\partial}{\partial x} \mathbf{I} \right| + \left| \frac{\partial}{\partial y} \mathbf{I} \right| \quad (1)$$

如果直接移除能量最低的像素，则会使得图像无法保持矩形；而如果保证图像每一行移除的像素数量相等，尽管这可以保证图像仍然为矩形，但是为图像带来了 zigzag 效应，破坏了图像内容。如果使用裁剪的方式保留能量最高的区域，虽然这使得图像的主要区域被保留，但是丢失了周围信息。直接移除能量最低的完整一列显然是更好的策略，但这一方法仍然会带来图像上的不自然。Seam Carving 方法定义了 Seam：

令 \mathbf{I} 为一个 $n \times m$ 的图像，一条垂直的裁剪线 Seam 被定义为：

$$\mathbf{s}^x = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, s.t. \forall i, |x(i) - x(i-1)| \leq 1 \quad (2)$$

其中， x 是一个映射 $x: [1, \dots, n] \rightarrow [1, \dots, m]$ ，即一个行坐标对应一个列坐标，一条裁剪线是一个行坐标连续（每一行只有一个像素），列坐标由映射决定的坐标点构成的集合，这是一个在图像上从上到下的 8-连通的通路。类似的，可以定义水平裁剪线：

$$\mathbf{s}^y = \{s_j^y\}_{j=1}^m = \{(y(j), j)\}_{j=1}^m, s.t. \forall j, |y(j) - y(j-1)| \leq 1 \quad (3)$$

对于一条垂直裁剪线来说，坐标点对应的像素点集合为 $\mathbf{I}_s = \{\mathbf{I}(s_i)\}_{i=1}^n = \{\mathbf{I}(x(i), i)\}_{i=1}^n$ 。与直接移除能量最低的完整一列的局部效果相同，图像中被移除一列右侧（下侧）的所有像素都向左（向上）移动来补全了被移除区域。因此除了被移除区域以外，剩余区域都是完整的。如果将 $|x(i) - x(i-1)| \leq 1$ 改为 $|x(i) - x(i-1)| \leq k$ ，那么在 $k=0$ 时获得一列像素，在 $1 \leq k \leq m$ 时候获得分段连通或不连通的像素路径。

对于一个被定义的能量函数 e ，移除一条裁剪线的代价是这条裁剪线上所有的像素的能量总和。 $E(\mathbf{s}) = E(\mathbf{I}_s) = \sum_{i=1}^n e(\mathbf{I}(s_i))$ 。即，最优的裁剪线 \mathbf{s}^* 可以使得裁剪线的代价最小。

$$\mathbf{s}^* = \min_{\mathbf{s}} E(\mathbf{s}) = \min_{\mathbf{s}} \sum_{i=1}^n e(\mathbf{I}(s_i)) \quad (4)$$

可以使用动态规划方法来查找这条最有裁剪线。将递推式(5)从图像的第一行一直计算到最后一行。

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1)) \quad (5)$$

矩阵 M 中最后一行中最小值所在的列对应的裁剪线就是我们要找的最优裁剪线，从后向前回溯即可找到最有裁剪线。对于水平裁剪线的情形是类似的。

2.2 使用 Seam Carving 实现改变图像长宽比的基本方式

在实际实现 Seam Carving 时，只需要实现一个列方向的算法，行方向算法可以通过转置来实现。实现改变图像缩放有三种方式，移除一些裁剪线，插入一些裁剪线，以及移除和插入裁剪线相结合。本文将其定

义为函数 `changeAspectRatio(colRatio, rowRatio, Mode)`. `Mode` 是枚举量 `REMOVAL`, `INSERTION`, `MIXED`. 4.1 部分展示了在使用基本的 `Seam` 查找方式和能量函数 1 的情形下三种方式的差别。

2.2.1 移除 Remove

移除裁剪线是最经典的缩放方式，直接按照 `Seam Carving` 寻找裁剪线的方式，按照计算能量图、计算最小累积能量图、寻找最优裁剪线，移除最优裁剪线的步骤多次循环，每次循环移除一条裁剪线的方式，即可缩小图像的长和宽。

2.2.2 插入 Insert

相比于移除裁剪线，插入裁剪线的优势在于图片中的信息会被完整保留。假设要插入 k 条裁剪线，做法是先移除 k 条裁剪线，完整记录这 k 条裁剪线的像素位置信息。然后将这 k 条裁剪线插入图像的对应位置。在 k 较大时，将其分成多次进行可以获得更好的效果，在本实验中， k 被拆分为不大于图像列宽/5 的多次插入，经过验证这种方式可以保证图像主体不会产生显著的形变。

2.2.3 混合缩放模式 Mixed

本实验将混合模式定义为先进行裁剪线的插入再进行移除，即先保证图片信息完整，再减少图像信息。如果将这一方式定义为保持图像面积不变，那么在输入和输出比例不同的前提下，一定有一侧需要插入裁剪线而另一侧恰好需要移除裁剪线。出于成本和性能考虑，本文没有考虑对整体来说最优的裁剪线插入和移除次序，而直接将其分成了插入和移除两个步骤。

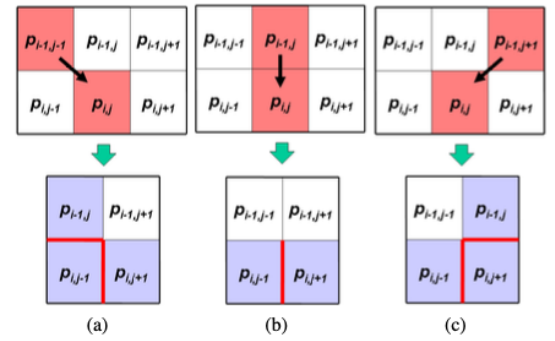


Figure 3.1: 对像素 $p_{i,j}$ 使用前向能量和计算三种可能的裁剪线。在移除裁剪线后会创建灰色的相邻像素和红色的边，在 a、b、c 三种情形下成本都以新创建边的前向差异定义。

3 算法改进

对 `Seam Carving` 的改进主要在于对能量函数的改进和 `Seam` 查找方式的改进。

3.1 改进最小累积能量图计算的 `Seam` 查找方式

`Seam Carving` 原作者提出了这一方式^[3]。原始算法选择最优裁剪线是能量最低的裁剪线，这忽略了一个问题，在裁剪线移除后，原本不相邻的像素变得邻接，使得新的能量被引入了图像。本文直接引用这一研究的结论，即使用以下方程进行最小累积能量图计算。

三个成本函数，如图 3.1 所示：

$$C_L(i, j) = |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j - 1)| \quad (a)$$

$$C_U(i, j) = |I(i, j + 1) - I(i, j - 1)| \quad (b)$$

$$C_R(i, j) = |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j + 1)| \quad (c)$$

裁剪线的成本函数：

$$M(i, j) = P(i, j) + \min\{M(i - 1, j - 1) + C_L(i, j), M(i - 1, j) + C_U(i, j), M(i - 1, j + 1) + C_R(i, j)\} \quad (6)$$

其中， $P(i, j)$ 是额外的能量计算方式（例如结合人脸检测、人为给定权重）的能量函数。

这是 `Seam Carving` 的一个重要改进，它显著提高了 `Seam Carving` 对重要内容形态保留的能力，本实验

实现了这一改进。本实验定义的 ImprovedSeamCarvingResize 类通过继承 SeamCarvingResize 类后重写最小累积能量图计算相关函数，实现了这一改进。

3.2 改进的能量函数

能量函数 1 和 2 是仅基于梯度的能量计算方式，能量函数 3 和 4 结合了其他方式对能量进行计算。

3.2.1 能量函数 1 – 考虑色彩的梯度法（水平垂直向前差分法）

原始的 Seam Carving 算法的能量函数考虑了像素的梯度，但通常直接以灰度图像来计算梯度，这里能量函数 1 以 1 范数考虑 R、G、B 三个通道的梯度，能量函数如(7)式所示。

$$e_R = \left| \frac{\partial}{\partial x} I_R \right| + \left| \frac{\partial}{\partial y} I_R \right|, \quad e_G = \left| \frac{\partial}{\partial x} I_G \right| + \left| \frac{\partial}{\partial y} I_G \right|, \quad e_B = \left| \frac{\partial}{\partial x} I_B \right| + \left| \frac{\partial}{\partial y} I_B \right|$$

$$e_1 = e_R + e_G + e_B \quad (7)$$

其中， $\frac{\partial}{\partial t} f = f(t+1) - f(t)$ 。

3.2.2 能量函数 2 – 考虑色彩的梯度法（混合水平垂直向前差分法和罗伯特差分法）

为能量函数 1 加入罗伯特差分法，可以在理论上提升对于斜 45 度方向边缘的检测能力。在度量梯度时使用 1-范数还是 ∞ -范数的问题上，经过实验，两种范数在效果上没有显著差异，故继续沿用 1-范数。

$$e'_R = \left| \frac{\partial}{\partial x} I_R \right| + \left| \frac{\partial}{\partial y} I_R \right| + |I_R(x+1, y+1) - I_R(x, y)| + |I_R(x+1, y) - I_R(x, y+1)|$$

e'_G 和 e'_B 的形式类似。能量函数：

$$e_2 = e'_R + e'_G + e'_B \quad (8)$$

3.2.3 能量函数 3 – 结合 Canny 边缘检测

Canny 边缘检测[9]是一种先平滑后求导数的方法，首先利用二维高斯滤波器对图像进行平滑。使用大小为 5×5 的高斯核， $\sigma = 1.4$ ：

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}$$

利用一阶导数算子（Sobel 算子）来检测图像边缘，得到每个像素点的梯度值和方向。

$$H_h = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad H_v = \begin{bmatrix} -1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad G = \sqrt{H_h^2 + H_v^2}, \quad \theta = \arctan\left(\frac{H_v}{H_h}\right)$$

然后进行非极大值抑制，考虑梯度幅度图中的小邻域，比较当前像素与同梯度方向相邻像素的梯度值，如果该像素的梯度值不大于相邻像素的梯度值，则将其值置为 0，否则将其保留。最后进行双阈值边缘连接，利用两个阈值 lowThreshold 和 highThreshold 进行两次边缘提取，以高阈值结果为基础，参照低阈值结果来连接图像边缘。

实验使用 Canny 边缘检测的结果应用到 Seam Carving 中时，以能量函数 1 为基础，设canny(i,j)为

Canny 边缘检测的结果

$$e_3(i, j) = \left(1 + \frac{canny(i, j)}{\max_{i=1}^n \max_{j=1}^m canny(i, j)} \right) e_1(i, j) \quad (9)$$

即利用 Canny 边缘检测的结果对能量函数 1 中的边缘进行增强。

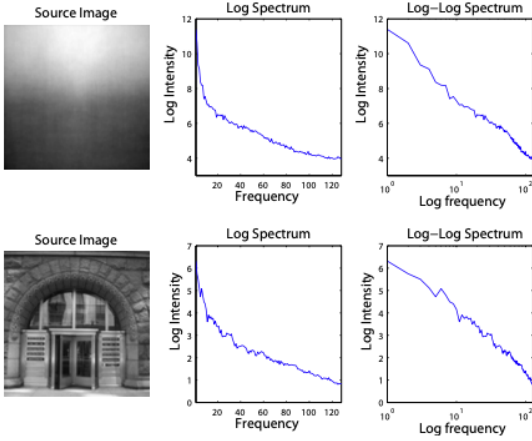


Figure 3.2.4: log 谱和 log-log 谱示例

的显著性信息。

实验使用残差谱算法的结果应用到 Seam Carving 中时，以能量函数 1 为基础，设 $SR(i, j)$ 为残差谱算法进行显著性区域检测的结果

$$e_4(i, j) = SR(i, j) * e(i, j) \quad (10)$$

即利用残差谱算法进行显著性区域检测的结果对能量函数 1 中的边缘进行增强。

4 实验结果与分析

实验环境：OpenCV 4.5.4-2, CMake 3.21.1, g++ version: Apple clang version 13.0.0 (clang-1300.0.29.3)

本文进行的实验目的有两个，首先实现 Seam-Carving 改变图片长宽比，其次尝试实现更好的 Seam-Carving。基于此，设计了三个实验，首先验证本文实现的算法的确可以实现突破缩放、其次比较不同的最小累积能量函数计算方法的差异，最后比较四种改进的能量函数在效果上的差异。



Figure 4.1-1 最左边为原图，左数第二张为只移除 Seam 的结果，第三张为只插入 Seam 的效果，第四张是保持图像面积不变的混合模式的效果。

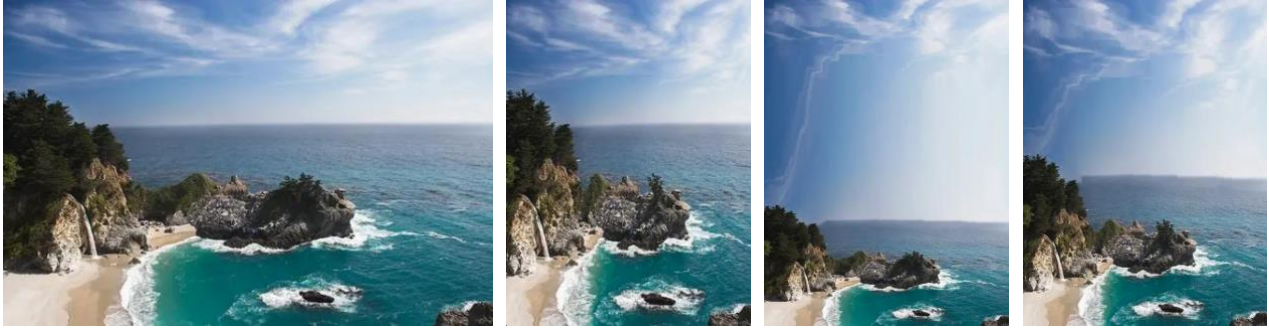


Figure 4.1-2 最左边为原图，左数第二张为只移除 Seam 的结果，第三张为只插入 Seam 的效果，第四张是保持图像面积不变的混合模式的效果。

4.1 验证多种基于 Seam Carving 的重定向方式改变图像长宽比

在使用 e_1 作为能量函数，使用非改进的 Seam 查找方式的情况下，对两张图片分别进行 Remove (RMEOVAL)、Insert (INSERTION)、Mixed (MIXED) 三种方式的实验。

图 4.1-1 将比例为 4:3 的图像缩放至 3:4，MIXED 模式最接近原图，REMOVAL 模式最自然。

图 4.1-2 显示了将 16:9 的图像缩放至 3:4 的效果，只移除裁剪线的效果和实验要求的差异化缩放完全一致。

显然，三种改变图像长宽比的策略都是有效的。在大部分情况下 REMOVAL 模式都有最自然的效果，MIXED 模式带来最贴近原图的效果，为了便于比较，实验 4.2 和 4.3 都以 REMOVAL 模式进行。

4.2 使用不同最小累积能量函数查找 Seam 的差异

在同为 REMOVAL 模式，能量函数都是 e_1 的情形下，对原始最小能量累积函数计算方式（原始函数 5）和改进的最小能量累积函数计算方式（改进函数 6）的处理效果进行对比。

可以看到，在图 4.2-1 和图 4.2-2 中，改进函数

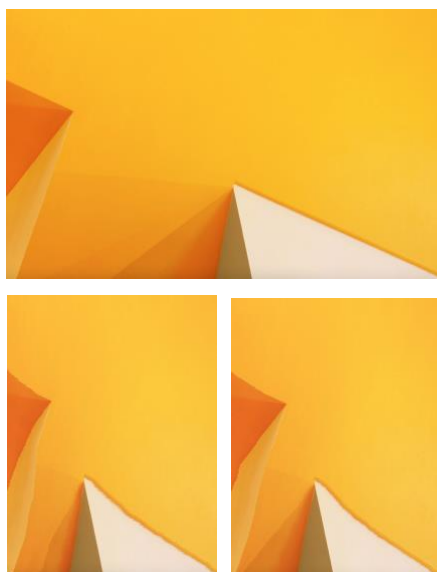


Figure 4.2-1 第一排为原图；第二排左 1 为原始效果，左 2 改进的能量累积函数的效果。

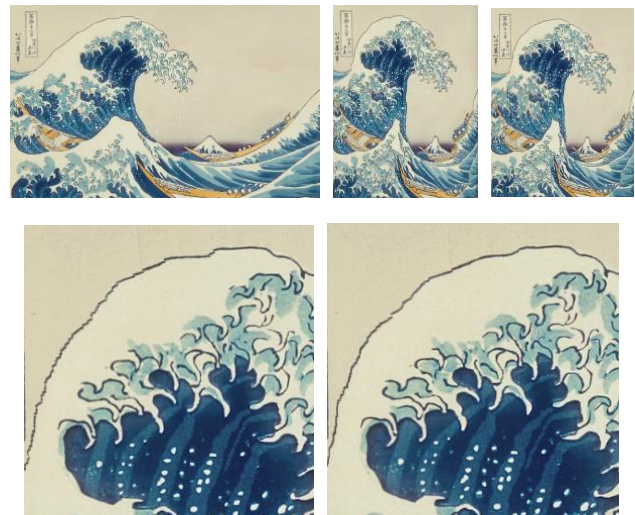


Figure 4.2-3 第一排，左数 1 为原图，；2 为原始结果，3 为改进的结果。第二排，左 1 为原始函数结果放大，左 2 位改进函数结果放大。改进方法显示出了更平滑的边缘处理。

都显示出了更好的形状保留能力，图 4.2-1 左侧的三角形比例更接近原图，而图 4.2-2 的树木年轮中心形状变形，改进函数显著小于原始函数。在图 4.2-3 中，改进函数则显示出了更加平滑的形状边缘。总体来说，改进函数的效果显著优于原始函数。



Figure 4.2-2 左数第一张为原图，第二张为原始能量累积函数的结果，第三张改进的能量累积函数的效果。

4.3 使用不同能量函数的差异

使用改进最小能量累积函数计算方式的 Improved Seam Carving，在 REMOVAL 方式下，使用不同的能量函数进行实验，比较不同能量函数的效果，所有图片都被处理成 3:4 的比例。这一部分的实验结果如图 4.3 所示。请注意， e_3 和 e_4 在本实验中仅为 Improved Seam Carving 设计。



Figure 4.3 每一排最左边都为原图，其余图片使用能量函数：左 1 为 e_1 ，左 2 为 e_2 ，左 3 为 e_3 （with Canny），左 4 为 e_4 (with Saliency)

对于第一排图片来说，第一排左 3 和左 4 相比于左 1 和左 2 都更好地保留了茶壶的形状，而左 3 相比左 4 还更完好的保留了背景线条直线的流畅性。

对于第二排图片，显著图加强的 e_4 获得了更好的效果，城堡完整并且左右主体与图片边缘的距离都更贴近原图，而 Canny 加强的 e_3 反而获得了最糟糕的效果。

对于第三排图片， e_3 没有完整保留 logo 的形状，但除了 logo 以外的线条是最流畅的； e_4 在背景中产生了更加生硬的过渡，其线条流畅性甚至不如 e_1 和 e_2 。

第四排图片中，Canny 加强的 e_3 算法显示出来了出色的形状保留能力，四个算法中只有这一算法仍然保持了楼主体的矩形形状和水面倒影的正确性。

总的来说， e_3 相比 e_1 和 e_2 展示出了更强的形状检测和保留能力，而 e_4 的形状保留能力优于 e_1 和 e_2 ，略逊于 e_3 。而在梯度法中引入罗伯特差分的 e_2 与 e_1 没有显著差异。

5 结束语

本文使用 Seam Carving 方法实现图像缩放，实现了裁剪线的移除、增加、以及混合方式改变图片比例。成功复现了改进最小累积能量函数的 Improved Seam Carving 方法，并且改进了能量函数。

实验发现，三种图像缩放方式中 MIXED 和 REMOVAL 通常可以取得更加自然的效果，而 MIXED 方式有待进一步优化，这或许会让缩放效果更加自然。对于改进最小能量累积图计算的 Improved Seam Carving，其对感兴趣区域的保留效果显著强于原始 Seam Carving，因此对于 Seam 的查找方式，可以认为仍然有改进空间。

对于能量函数而言，使用 Canny 增强的能量函数具有很强的保留物体规则边缘的能力，而残差谱增强的能量函数在某些情形下具有优于其余三种能量函数的形状和边缘保留能力。然而在梯度法中增加罗伯特差分并不比仅使用水平垂直差分更优越。基于显著性检测的方法没有显著增强最终效果，推测原因是残差谱算法增强的能量函数达不到实验预期，显著性检测算法在 Seam Carving 中的潜力还远未被挖掘完。

参考文献

- [1] Ke, Jun, Majumder, et al. Seam carving based aesthetics enhancement for photos[J]. Signal Processing. Image Communication: A Publication of the the European Association for Signal Processing, 2015, 39(Pt.C):509-516.
- [2] Avidan S, Shamir A. Seam carving for content-aware image resizing[J]. ACM Transactions on Graphics, 2007, 26(3):10.1-10.9.
- [3] Rubinstein M, Shamir A, Avidan S. Improved seam carving for video retargeting[J]. ACM Trans. Graph, 2008, 27(3):23-31.
- [4] Pritch Y, Kav-Venaki E, Peleg S. Shift-Map Image Editing[C]// Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009.
- [5] Wolf L. Non-homogeneous content-driven video-retargeting[J]. proc iccv, 2008.
- [6] Wang Y S, Tai C L, Sorkine O, et al. Optimized Scale-and-Stretch for Image Resizing[J]. ACM Transactions on Graphics, 2008, 27(5):118.
- [7] Karni Z, Freedman D, Gotsman C. Energy-Based Image Deformation[J]. Computer Graphics Forum, 2009, 28(5):1257-1268.

- [8] Rubinstein M , Shamir A , Avidan S . Multi-operator media retargeting[J]. ACM Transactions on Graphics, 2009, 28(3):23-33.
- [9] Canny J . A Computational Approach to Edge Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, PAMI-8(6):679-698.
- [10] Itti L . A model of saliency-based visual attention for rapid scene analysis[J]. IEEE Trans, 1998, 20.
- [11] Hou X , Zhang L . Saliency Detection: A Spectral Residual Approach[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2007.
- [12]Radhakrishna Achanta[†], Sheila Hemami[‡], Francisco Estrada[†], et al. Frequency-tuned salient region detection[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2009.
- [13]Cheng, Mitra N J , Huang X , et al. Salient Object Detection and Segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 37(3):1.
- [14] Ahmadi M , Karimi N , Samavi S . Context-aware saliency detection for image retargeting using convolutional neural networks[J]. Multimedia Tools and Applications, 2021, 80(10).
- [15] Yang C , Zhang L , Lu H . Graph-Regularized Saliency Detection With Convex-Hull-Based Center Prior[J]. IEEE Signal Processing Letters, 2013, 20(7):637-640.