数字图像处理实验报告

姓名：卜梦煜　　学号：1120192419　　班级：07111905

1. 实验名称

图像缩放

2. 实验目的

- 掌握数字图像处理的基本原理与方法。
- 阅读、理解缩放技术相关论文。
- 编程实现考虑像素的差异化处理的图像缩放操作。

3. 实验内容

设计一个算法，输入一张原始图片，输出一张改变长宽比的图像（长宽比可以自行指定），算法要求必须考虑像素的差异化处理。

4. 实验环境

PyCharm 2021.2.3、python 3.9、opencv-python 4.5.5.62、numpy 1.20.3

5. 算法描述

5.1 概述

Seam Carving 算法是基于图像内容的缩放算法。该算法定义并利用图像的能量图来确定图像中的重要区域，使用动态规划算法寻找图像中能量值最低的缝对应的路径，通过迭代插入或删除能量值最低的缝来实现图像的像素差异化的缩放。

本文主要改进了能量图的生成方法与图像放大中缝隙选取方法。

能量图的生成方法方面，定义能量图为梯度图和视觉显著度图的加权和，从而实现对区域边缘信息的有效提取，以及对视觉主体区域内容的有效保留。

图像放大中缝隙选取方面，选取不重叠的k条像素带代替最小的k条像素带，从而在非重要信息区域，像素带的增加能得到较均匀的分布，一定程度上解决了非重要信息区域变形扭曲的问题。

5.2 Seam Carving 图像缩放算法

Seam Carving 算法利用图像梯度生成能量图，确定图像的重要区域。能量函数如下:

$$e(\boldsymbol{I}) = \left|\frac{\partial}{\partial_x}\boldsymbol{I}\right| + \left|\frac{\partial}{\partial_y}\boldsymbol{I}\right|$$

实际编程中，常使用 Sobel 算子做边缘检测，获得图像的能量图。

设原始图像I的大小为$n \times m$，其中n为行数，m为列数，则垂直像素带S表示为:

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

对于一条包含n个像素点的垂直像素带或水平像素带S，定代价函数为:

$$E(S) = \sum_{i=1}^{n} e(S(i))$$

最优路径为代价最小的像素带:

$$E^* = \min_S E(S) = \min_S \sum_{i=1}^{n} e(S(i))$$

最优路径可通过动态规划算法得到。对于某点像素，定义积累的能量值为该点所在最优路径中，从边缘到该点的路径上的点的能量和，记为$M(i,j)$，则状态转移方程为:

$$M(i,j) = e(i,j) + \min \begin{cases} M(i-1, j-1) \\ M(i-1, j) \\ M(i-1, j+1) \end{cases}$$

从最小值位置进行回溯，即可找到整条最优路径。

找到最优路径后，对于图像缩小操作，删除该条路径，对于图像放大操作，在该条像素带的左侧或右侧使用均值插值插入一条像素带。

由能量图定义可知，点的能量值反映了该点所在区域的视觉重要程度，能量值越大，该点所在区域越重要。因此，选择能量最小的像素带删除，可以尽量避免图像中视觉重要区域发生变形扭曲。

5.3 Seam Carving 算法的改进

5.3.1 能量函数的改进

本文改进能量函数的定义。对于 Seam Carving 算法中的梯度能量图，只能检测出图像的边缘化结构的信息，这是有局限性的，对于非重要区域存在较多边缘信息，如草地、水波等内容，会导致该区域的重要度超过视觉主题区域，从而导致容易造成图像是真变形。针对这一不足,本文提出了一种新的能量函数生成方式,即结合梯度图与视觉显著度图的能量图。梯度图与 Seam Carving 算法一致，采用一阶导数生成。视觉显著度图采用 FT（Frequency-tuned）显著性检测算法生成。

FT 算法从频域角度分析图像，通过提取图像的低频部分的信息，能够简洁高效地计算

出图像的显著度。像素显著度计算公式如下 L:

$$S(P) = ||I_\mu - I_{\omega_{hc}}(p)||$$

其中，I_μ为图像在 Lab 颜色空间的平均特征，$I_{\omega_{hc}}(p)$为像素p在高斯平滑后的 Lab 颜色空间的特征。

FT 算法步骤如下:

（1）对图像进行 5*5 的高斯平滑。

（2）转换颜色空间为 Lab 颜色空间。

（3）计算政府图片的 L、a、b 的平均值。

（4）计算每个像素 L、a、b 值与图像的 L、a、b 均值的欧氏距离，得到显著度图。

（5）归一化操作。将图像中每个像素的显著度值除以显著度图最大的显著度值，得到归一化的显著度图。

5.3.2 图像放大操作的改进

本文改进了图像放大操作中缝隙的选取方式。Seam Carving 方法选取前 k 条最小的像素带作为最优路径，实验发现，这些路径容易出现“汇聚”现象，即这些像素带呈现出树状结构，这会导致树根部分的像素放大时重复选取一部分像素，图像部分存在明显的拉伸变形的问题。

针对这一问题，本文的改进方法为，选取 k 条不重叠的像素带作为增加像素带的位置。在 Seam Carving 方法中，图像放大仅计算一次能量图，之后选取 k 条最小的像素带为最优路径插入像素。本文将这一过程改为“算-选-删”的循环过程：计算能量图，选取 1 条最优路径，记录该路径并在图中删除该路径。通过这种方式，保证选取的 k 条路径时不重叠的，这也就使得插入的像素带是均匀分布的，避免了变形现象。

6. 结果分析

本文对 Seam Carving 算法在能量图生成与放大时缝隙选取方法两个方面做了改进。以下分两方面分析实验测试的效果：算法效果、与传统 Seam Carving 方法的比较。

6.1 算法效果

本文的图像缩放效果如下。其中，图（a）为原始图像，图（b）为水平缩小 20%，垂直缩小 10%的图像，图（c）为水平放大 10%，竖直放大 15%的图像。可以看到，无论是缩小还是放大，图像均未产生明显变形，效果较好。

（a）原始图像

（b）缩小图像

（c）放大图像

6.2 与传统 Seam Carving 方法的比较

本文采用传统 Seam Carving 算法与改进能量图的 Seam Carving 算法进行对比分析。图（a）为原始图像，图（b）传统 SC 算法能量图，图（c）本文改进的 SC 算法的能量图，图（d）传统 SC 算法结果，图（e）本文改进的 SC 算法结果。

如图所示，本文改进的 SC 算法相比传统 SC 算法，更多地关注图像中视觉主体的内容，从而使得视觉主体不易产生形变。图（d）为传统 SC 算法，左侧小狗有明显的拉伸变形，图（e）使用本文改进的 SC 算法，视觉主体部分保留较好，无变形问题。

（a）原始图像

（b）传统 SC 算法能量图

（c）本文改进的 SC 算法的能量图

（d）传统 SC 算法结果

（e）本文改进的 SC 算法结果

7. 实验优点

（1）仔细阅读、理解传统 Seam Carving 算法的论文，并自己实现了论文代码。

（2）改进了能量图的生成方式。

（3）改进了图像放大操作中像素带选取方法。

8. 总结

这次实验让我收获很多。首先，我更深入理解了数字图像处理的基本原理与方法，尤其是对像素差异化操作的图像缩放的方法有了一定程度的掌握。其次，我熟悉了 opencv-python 库图像操作的一些函数，能够利用 opencv-python 库实现图像的缩放操作。最后，我阅读了 Seam Carving 算法的相关论文，论文阅读能力与理解能力有了一些提升。

在实验的过程中，我也遇到了一些问题。如在改进图像放大操作中的缝隙选取方法时，最初的方法是，在一次 dp 的基础上，限制像素可重复选取的次数上限来避免汇聚现象的产生，写完代码运行时才发现，这种汇聚现象在很多时候是汇聚到一个点，比如对$n \times m$的图像，$n > m$时，最优的水平像素带一定会在第 n 列前汇聚到一点。这就导致限制像素可重复选取的次数上限会导致无法选出 k 条像素带，即这个优化思路时错误的。遇到的其他问题还有对库函数不够了解，编码不够熟练等。但通过尝试、探索、查阅资料、阅读文献等方法，也顺利克服了这些问题。