

浙江大学

硕士研究生读书报告



题目 基于内容感知的图像缩放算法

作者姓名 张楚润

作者学号 21951432

学科专业 软件工程

所在学院 软件学院

提交日期 2019 年 12 月

Seam Carving for Content-Aware Image Resizing

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

By

Churun Zhang

Zhejiang University, P.R. China

2019

摘要

本文提出了一种十分新颖的图像缩放算法——基于内容感知的图像缩放算法。等比例缩放无法满足不同图像尺寸的要求，图像裁剪会丢失图像边缘的一些关键信息，任意比例缩放又会使图像变形失真。本文提出的方法可以保留图片中的重要信息，近乎完美地解决了这一痛点。同一幅图片，在任意尺寸下看起来都可以很舒服真实。

关键词：图像缩放，自适应，人机交互

Abstract

In this paper, a novel image scaling algorithm based on content perception is proposed. Equal scale can not meet the requirements of different image sizes, image clipping will lose some key information of image edges, and arbitrary scale will make the image distorted. The method proposed in this paper can retain the important information in the picture, and almost perfectly solve this pain point. The same picture can look comfortable and real in any size.

Keywords: Image zooming, Adaptive presentation, human-computer interaction

1 Seam Carving 算法简介

改变图像大小的传统方式是按比例缩放或者裁剪。缩放时未按照原比例会使得图像失真，裁剪则可能丢失位于边缘的重要信息。Seam Carving 算法针对于传统图像缩放算法存在的痛点应运而生，是一种基于内容感知的图像缩放算法。对于蓝天、湖水、草坪等图像背景来说，相邻区域像素点的颜色很相近，可以合理删减或增加背景的像素点来达到缩放图像的目的。Seam Carving 能够以水平和垂直两种方式简单地压缩和拉伸图片。

2 Seam Carving 算法简介

第 34 届 SIGGRAPH 2007 数字图形学年会上，以色列的两位教授 Shai Avidan 和 Ariel Shamir，展示了 Seam Carving 这种新的缩放裁剪图像方法。Seam Carving 算法能计算出图像上的关键部分和不重要区域，从而使得随意改变一个图像的高宽比但不会让内容变得扭曲成为可能。Seam Carving 的缝补算法使得图片缩放后可以仍然维持整体的完整性。Seam Carving 可以将原本窄镜头的夕阳照片修改成广角镜头的夕照，而中心主题的阳光不会因图片拉宽而变形。

3 Seam Carving 算法流程

1、计算图像能量图

能量图一般是图像像素的梯度模值。为了简化计算，可先转换成灰度图像，然后直接用 x 、 y 方向上的差分取绝对值后再相加。

2、寻找最小能量线

最小能量线指的是需要被移除的那一行。首先需要以图像第一行或最后一行为开始行进行迭代。找出最后一行需要被移除的像素点后，设其坐标为 $P(x,y)$ ，然后往上一行寻找，寻找的点为 P 点的在 $y-1$ 行中的三个相邻像素点中的能量最小值像素，也就是寻找的坐标为 $(x-1,y-1)$ 、 $(x,y-1)$ 、 $(x+1,y-1)$ 。

3、移除得到的最小能量线，让图片的宽度缩小一个像素

移除最小能量线，同时所有位于最小能量线右边的像素点左移一个单位，从而实现图像缩小宽度缩小一个单位。

移除的时候为了让图像看起来自然，需要在移除缝线的地方进行平均，假设

移除坐标为 $P(x,y)$, 那么移除后 $P(x-1,y)$ 的像素值为 $P(x-1,y)$ 与 $P(x,y)$ 的像素值的平均。 $P(x+1,y)$ 的像素值为 $P(x-1,y)$ 与 $P(x,y)$ 的像素值的平均, 然后才能把 $P(x+1,y)$ 移动到 $P(x,y)$ 的位置。对于图像的放大算法原理一样, 先找到最小能量线, 设能量线上点的坐标为 $P(x,y)$, 则在 $P(x,y)$ 、 $P(x+1,y)$ 中心位置插入新的像素, 像素值为 $P(x,y)$ 与 $P(x+1,y)$ 的平均。

4 算法的 Matlab 实现

`main.m` 函数是程序运行的主函数; `energyIm.m` 函数处理得到输入图像的能量图; `seamImage.m` 函数计算各个像素点的累积能量值; `backTrack.m` 函数记录能量线中各点在图像中的位置; `removeSeam.m` 删除一条能量线; `SeamCarving.m` 函数删除掉输入图像的指定列数。(代码详见附件)

5 实验效果及存在的问题

首先使用的是借鉴代码给出的泰姬陵图片来查看运行效果, 删除 20 列像素点的效果还比较令人满意。但是当删除列数增加时, 缩小效果不符合预期。如图 1 所示, 程序运行结果出现顾左不顾右的现象 (右边柱子之间的蓝天本应也被删除几列像素点)。

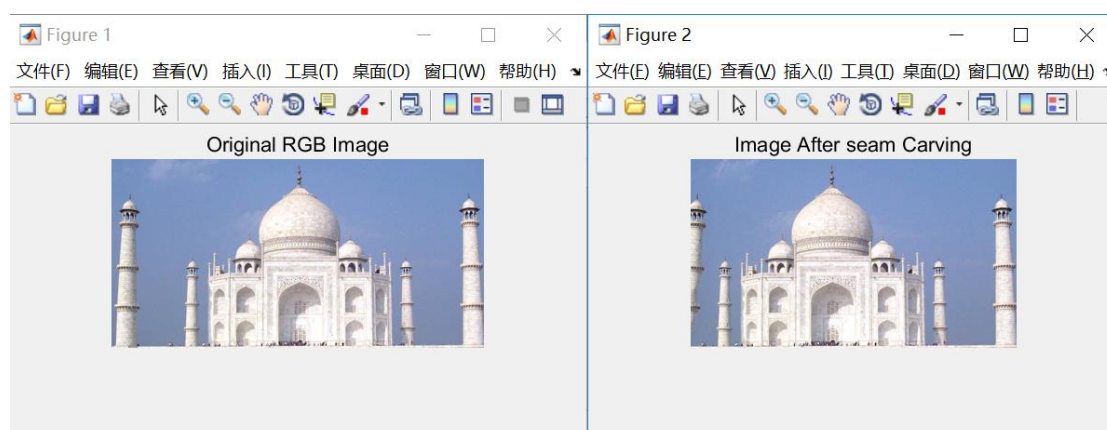


图 1. 对称图像的运行效果

查看代码之后, 未找到逻辑错误的地方。如图 2 所示, 换一张左右边缘都很简单无颜色变化的图像来看, 图像的缩小按照预期将左右边缘删去了规定的列数。

进一步地, 在图 2 的原图左侧涂鸦, 期望程序能够裁剪掉图片右侧边缘。程

序运行结果如图 3 所示，符合预期。程序没有出现顾左不顾右的现象，不仅裁掉了左侧边缘空白的几列，还裁掉了右侧较宽的空白边缘。

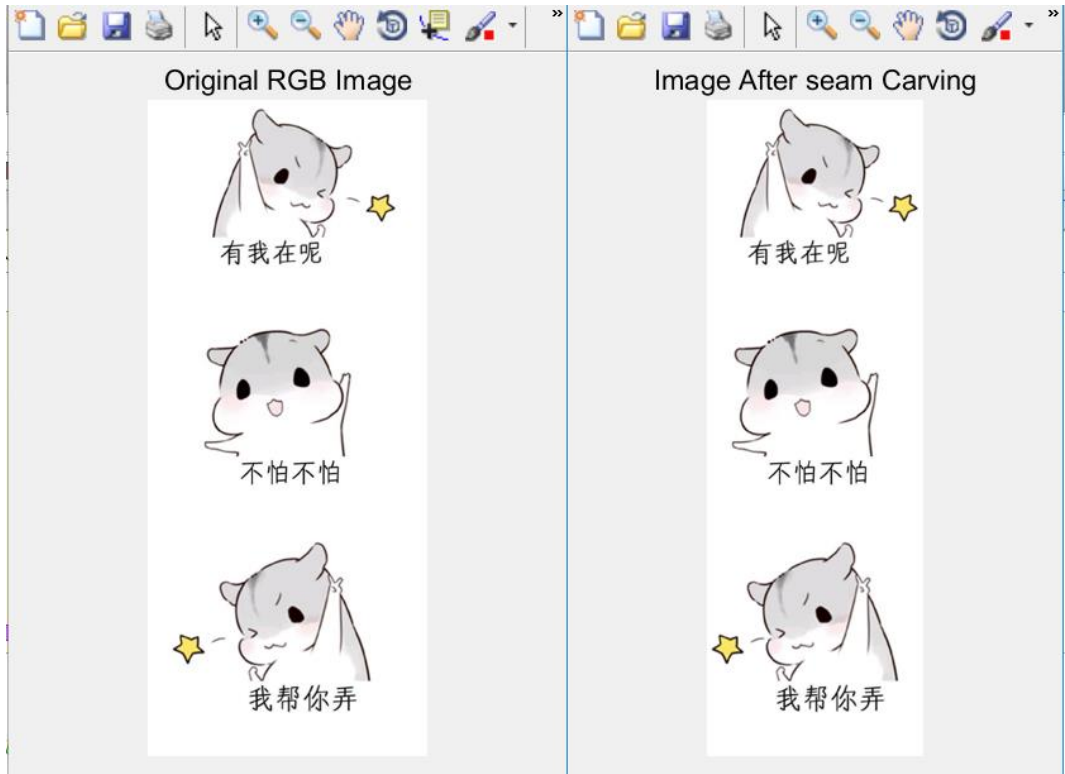


图 2. 边缘简单图像的运行效果

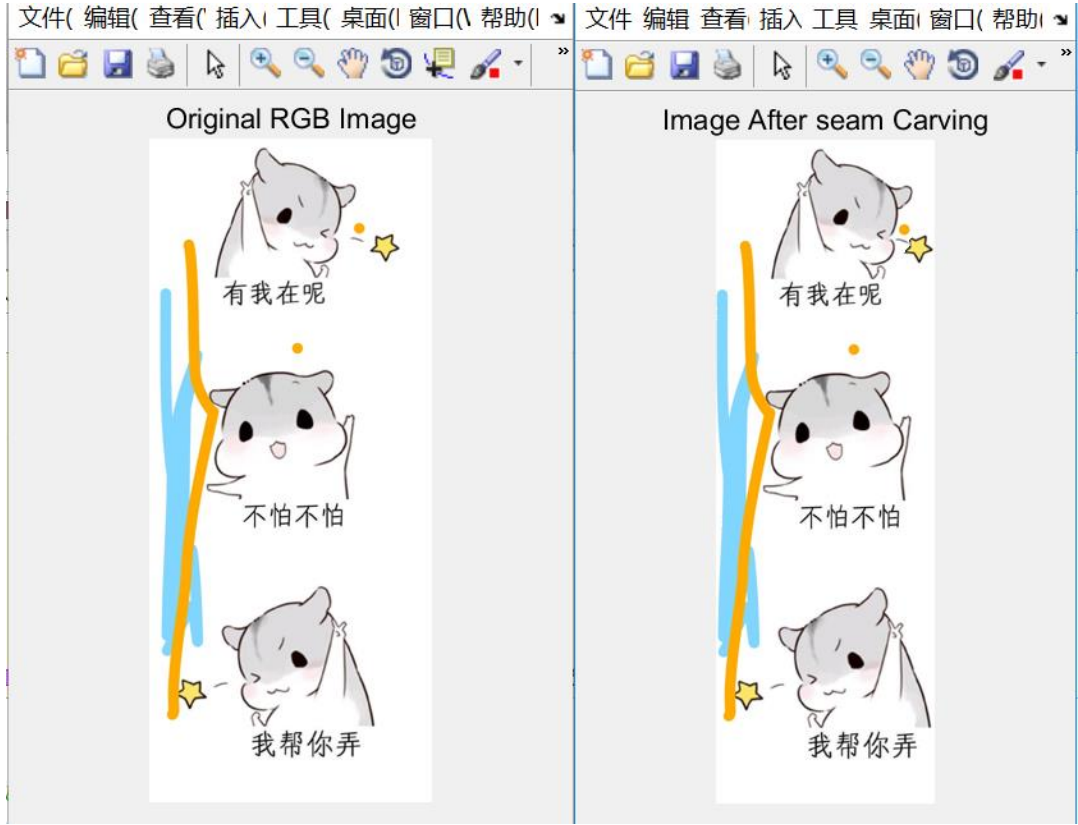


图 3. 检测算法是否只顾及左边

最后，对包含人像的图片查看运行效果。如图 4 所示，可以看出程序对图像左侧、中间、右侧的像素点都进行了删减。除了右侧人脸头发左上角被删减后处理得不自然外，效果还比较令人满意。

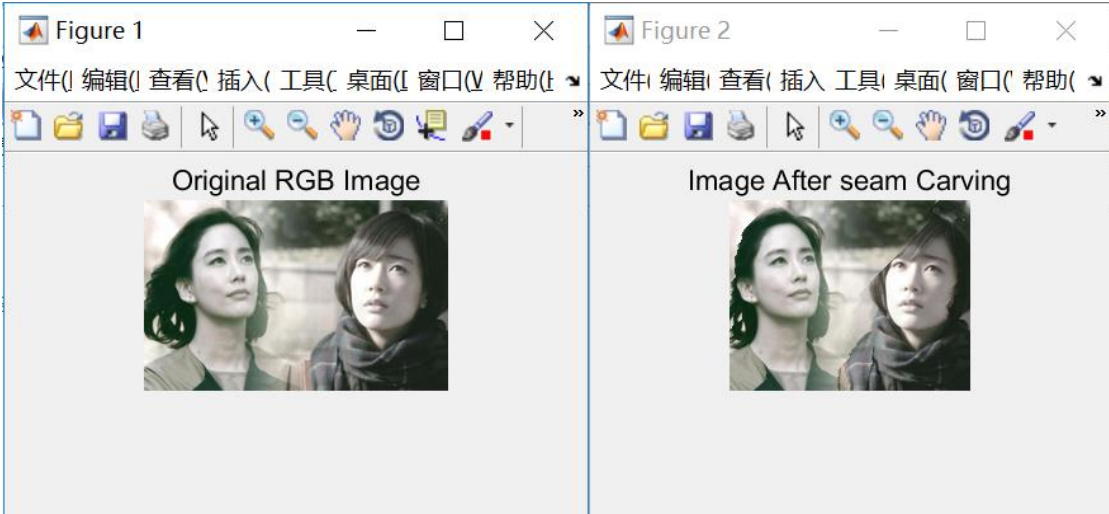


图 4. 包含人脸图像的运行效果

4 小结

现在实现的程序鲁棒性不够好，对一些图片的缩小效果并不能令人满意。另外，程序的功能还不够完善，还可以添加横向删减像素、横纵向增加像素、增删后均值化处理等功能。在编写 MATLAB 程序的时候，求图像微分的方法没有理解透彻，应该是用到了 Sobel 算子或者 Prewitt 算子之类的，用差分代替微分求图像偏导。