

基于内容的尺寸变换与物品移除

Image Scaling and Object Removal via Carving Seams

廖烙锋 (15307130197)

刘佰川 (16307130214)

林阳昊 (16307130122)

1 问题描述

在尺寸变换问题中，我们需要在尽可能不影响原图片或原视频的重要内容的情况下，对图片或视频中的像素进行削除或扩增。问题的核心在于如何从图片中找到一条能量最小的宽度为1像素的线，或是在视频中找到一个能量最小的厚度为1像素的流形，并对线或流形上的像素进行削除或扩增。

2 算法描述

2.1 最小割问题

定义1 (割) 一个 s - t 割 $C = (S, T)$ 是一种 V 的划分使得 $s \in S, t \in T$ 。 C 的割集是集合 $\{(u, v) \in E : u \in S, v \in T\}$ 。

定义2 (割的容量) 一个 s - t 割的容量是

$$c(S, T) = \sum_{(u, v) \in (S \times T) \cap E} c_{uv} = \sum_{(i, j) \in E} c_{ij} d_{ij},$$

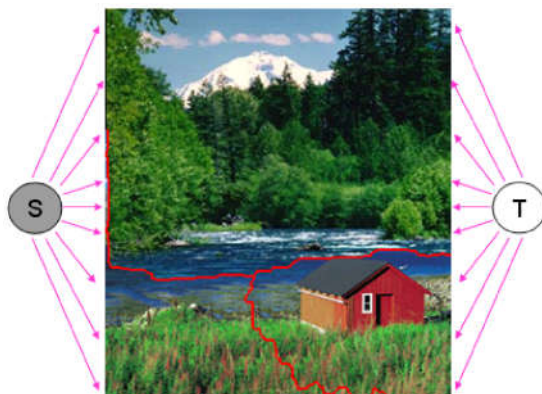
其中 $d_{ij} = 1$ 如果 $i \in S$ 并且 $j \in T$, 0 反之。

定义3 (最小 s - t 割问题) 计算 $c(S, T)$ 的最小值，即找到 S 和 T 使 s - t 割的容量达到它的最小值。

2.2 从图片缩减到最小割

我们先将从图片中找到一条能量最小的宽度为1像素的线这一问题转化为一个最小 s - t 割问题，进而将这一算法应用到视频方面。以下算法说明假设需要找的线是垂直方向的，如果需要找水平方向的线，只需将图片进行旋转。

我们需要用图片中的所有像素构建一个网格状的有向图（见图一），图中的每个顶点都代表一个像，与其相邻像素相连（相连方式在之后解释）。此外图中还需建立虚拟节点源点 s 、目标点 t 。源点 s 与图像最左一列的像素以无限容量的边连接，目标点 t 与图像最右一列的像素以无限容量的边连接。这样的一个有向图的最小 s - t 割的左侧一列像素即是我们需要的能量最小的线。



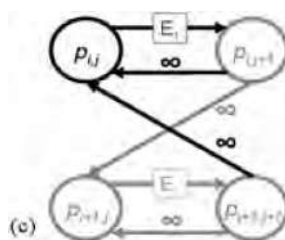
图一：最小割问题与图片收缩的联系。

2.3 像素顶点间的边

一条线要应用于图像尺寸变换，必须满足以下两个条件：单调性(Monotonicity)、连接性(Connectivity)。单调性保证这条线在图片的每一行只有一个像素，连接性保证这条线中相邻两个像素的水平距离不超过1。

为了让求解出来的最小割对应的线满足这两个条件，我们在任意两个左右相邻像素点 $p_{i,j}$ 和 $p_{i,j+1}$ 间都加上一条**从右向左**的边 $(p_{i,j+1}, p_{i,j})$ ；在任意两个对角相邻像素点 $p_{i,j}$ 和 $p_{i+1,j+1}$ 或 $p_{i,j}$ 和 $p_{i-1,j+1}$ 间都加上一条**从右向左**的边 $(p_{i+1,j+1}, p_{i,j})$ 或 $(p_{i-1,j+1}, p_{i,j})$ 。这些边的容量都是无限。

为了使求出的最小割对应的线能代表图像中能量最小的线，我们在任意两个左右相邻像素点 $p_{i,j}$ 和 $p_{i,j+1}$ 间都加上一条**从左向右**的边 $(p_{i,j}, p_{i,j+1})$ ，这条边的容量为像素点 $p_{i,j}$ 的能量。这使得在这个图中的任意一条满足单调性和连接性要求的割的容量 $c(S, T)$ 的值都相当于这条割左侧一列像素点的能量总和。



图二：从图片到带权图构建的示例。

2.4 将有向图的结构拓展到视频

假设我们需要在视频的 $X \times Y \times T$ 三维空间中寻找一条垂直（ Y 方向）的流形，那么只需要将图片情况下我们在 $X \times Y$ 平面上应用的容量无限的两种边应用到 $X \times T$ 平面上，就可以保证流形在时间方向上的单调性和连接性了。

2.5 从单向收缩到任意伸缩

删除多列像素时，我们可以按照下面的算法步骤进行。纵向的调整只需要将图像旋转90度即可。详情参见算法一与算法二。

算法一：收缩图像算法

1. 求解一个最小割问题，移除一列像素。
 2. 在移除像素后的图片上重新构建一个带权图，再次求解。
 3. 重复以上步骤知道尺寸缩小到指定大小。
-

算法二：拉伸图像算法

1. 求解出最小割问题的前K条最小路径。
 2. 在上述Seams的位置重复像素点。
-

2.6 从Seam Carving到物品消除

延续算法的思路，我们需要移除图片上的像素同时保证最小的信息损失。因此，我们在构建图的时候，可以将指向被消除的像素的边置上负权值。这样在求解最小割问题时，选出来的最小割更有可能经过需要移除的像素点。

3 使用指南

本次项目完全开源，可以从以下地址获取源代码。

<https://github.com/Chronos-LYH/VideoSeamCarver>

若需运行我们的UI界面，直接在我们所上交的文件夹下运行UI.py文件即可，UI.py依赖的文件有ig2nx.py,targetArea.py,video.py（算法文件）golf.gif,1544672096862777.png（初始化界面的图片和视频）

我们的UI界面根据我们完成的任务分成了三块，image seam对应图像裁剪，可以完成对于图像的基于内容的任意尺寸裁剪、object remove对应物品移除，可以让用户对想要消除的物品进行选点利用算法进行物品移除、video seam对应于视频裁剪，可以完成对于视频的基于内容的任意尺寸裁剪。目前的尝试中pyqt只支持对gif的显示，不支持对mov、MP4等内容的显示，故原视频和结果视频的文件类型皆为gif。

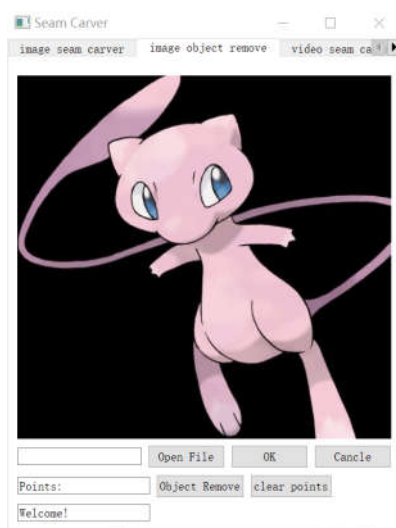
3.1 图片裁剪使用步骤

因为此任务是基于内容的裁剪，为了显示我们算法的有效性，并未对图片等比缩放，美观性存在缺陷，且分辨率低的图片和视频显示更好的效果。

- Openfile打开本机系统目录，选择想要处理的图片。若选择正确的图片格式文件（jpg,png,jpeg,bmp），图片的文件名将显示在第一行的文本框，点击OK确认选择图片，图片将会显示到界面，点击Cancel取消此次选择。
- 选择图片文件后，右下角的文本框将给出图片大小的提示。
- 在width和height文本框中设置想要的图片的大小，点击seam运行算法执行任意尺寸的裁剪。同时右下角的文本框将给出程序正在运行或是运行结束的提示。
- 最后的结果显示在界面上。参见图三。

3.2 图片物品消除使用步骤

- Openfile打开本机系统目录，选择想要处理的图片。若选择正确的图片格式文件，图片的文件名将显示在第一行的文本框，点击OK确认选择图片，图片将会显示到界面，点击Cancel取消此次选择。
- 选择图片文件后，最下面一行的文本框将给出图片大小的提示。
- 直接在图片上点击想消除物品的左上角和右下角，点击的坐标将显示在第二行的文本框中，注意选择的坐标将对物品移除的效果有很大的影响，若想移除选择的点点击clear points将会消除点击过的所有的点，若确认选择，点击object remove进行任务。若选择的点大于了2个，在点击object remove后，将在最下一行输出错误提示信息，并且消除所有的点。
- 同时最下一行的文本框将给出程序正在运行或是运行结束的提示。
- 最后的结果显示在界面上。参见图四。

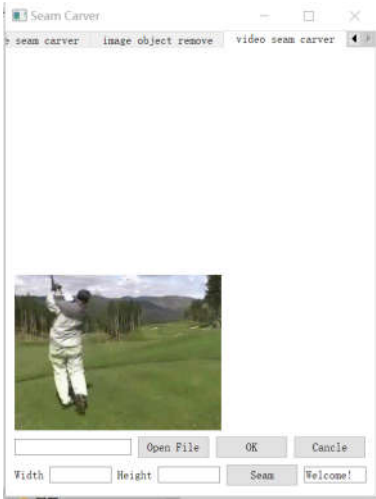


图四：物品消除的操作界面。

3.3 视频裁剪使用步骤

- Openfile打开本机系统目录，选择想要处理的视频。选择正确的图片格式文件（目前只支持gif），视频的文件名将显示在第一行的文本框，点击OK确认选择图片，图片将会显示到界

- 面，点击Cancel取消此次选择。
- 在width和height文本框中设置想要的视频的大小，点击seam运行算法执行任意尺寸的裁剪。同时右下角的文本框将给出程序正在运行或是运行结束的提示。
 - 最后的结果显示在界面上。参见图五。



图五：视频剪裁的操作界面。

3.4 使用示例

我们的每个结果展示请在result文件夹下查看。参见表格一。

表格一：result文件夹下示例图片及其对应内容

Dolphin_augment_movie, dolphin_agument_seams	基于内容的图像扩充
Image_seam	基于内容的图像缩减
Remove_target_movie	物品移除
Waterski_low_resolution, seam_waterski, result_waterski	对应基于内容的视频缩减

用户也可以在UI下进行这三类任务，结果不仅在UI上显示，并且会自动保存至out文件夹目录下，裁剪任务的结果文件名为result，物品移除任务的结果为Remove_target_movie。

Have fun!

4 代码结构

注1：前端开发环境为PyQt = 5.9.6，后端开发环境为iGraph = 0.7.1, numpy = 1.16.3, imageio = 2.5.0, PIL = 5.4.1。

注2：本次UI打包成exe可执行文件，需要在同目录下放ffmpeg.exe才能运行

4.1 前端代码结构

```
class TabDemo(QTabWidget):
    """继承窗口类"""

    def __init__(self, parent=None):
        """初始化"""

    def tab1UI(self):
        """图片裁剪窗口"""

    def tab2UI(self):
        """物品移除UI窗口"""

    def tab3UI(self):
        """视频裁剪UI窗口"""

    def object_remove(self):
        """物品移除前后端交互"""

    def seamer(self):
        """图片裁剪前后端交互"""
        ...

    def seamer_video(self):
        """视频裁剪前后端交互"""

    def showDialog(self):
        """选择图片文件"""

    def showDialog1(self):
        """选择视频文件 目前pyqt只支持显示gif"""

    def showVideo(self):
        """显示gif"""

    def showImg(self):
        """显示图片"""

    def contextMenuEvent(self, event):
        """右键的退出设置"""

    def addPoints(self):
        """tab2鼠标选点"""
        ...

    def clearPoint(self):
        """清除点"""
        ...

class myLabel(QLabel):
    """可以响应鼠标点击的QLabel类"""

    def toText(lis):
        """输出选择点的信息"""
```

4.2 后端代码结构

```
video.py
├── images
├── OUT_FOLDER
├── L1Norm(pixel1, pixel2)
├── VideoWrapper
│   ├── __init__(self, img)
│   ├── getValue(self, node)
│   └── setValue(self, node, value)
├── VideoSeamCarver
│   ├── __init__(self, img, mode='vertical')
│   ├── _initializeGraph(self)
│   ├── _addEdges(self, t, i, j)
│   ├── _removeEdges(self, t, i, j)
│   ├── _getUpwardNode(self, t, i, j)
│   ├── _getBackwardNode(self, t, i, j)
│   ├── _getDiagonalNodes(self, t, i, j)
│   ├── _getAffectedPoss_beforeInsertion(self, t, i, j)
│   ├── _getAffectedPoss_afterInsertion(self, t, i, j)
│   ├── _flatten(self, seam)
│   ├── _energy(self, t, i, j)
│   ├── Solve(self)
│   ├── SolveK(self, k)
│   ├── RemoveSeam(self, seam)
│   ├── AugmentSeam(self, seam)
│   ├── ShowGraph(self)
│   ├── GenerateVideoWithSeam(self, seam)
│   ├── GenerateVideoWithSeams(self, seams)
│   ├── GenerateVideo(self)
│   ├── ShowImg(self, seam, numCols)
│   ├── shrink_hor(self, rm_count, save_hist=False, want_trans=False)
│   ├── augment_hor(self, aug_count, save_hist=True, want_trans=False)
│   ├── trans(self, video)
│   └── scale_hor(self, pix_count, save_hist=False)
```


5 成员贡献

我们的贡献和创新：整个项目未参考任何源码，皆为我们自己独立完成。实现算法并将算法应用到物品移除等现实任务中。

- 刘佰川：完全前端设计、前后端交互、撰写报告、PRE展示准备，小组讨论。
- 廖烙锋：完全两大应用算法实现，算法模式设计，报告撰写与排版，PRE展示准备小组讨论。
- 林阳昊：基础框架设计、核心算法实现与性能优化、报告撰写、PRE展示准备，小组讨论。