

Content-Aware Image Resizing Report

郑凯文

2021 年 4 月 18 日

1 Aspect Ratio Change

使用课上的能量函数

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

将图像转化为灰度图后，对每个像素点计算能量，这可以简洁地用下列代码实现

```
dx, dy = np.gradient(cv2.cvtColor(image, cv2.COLOR_BGR2GRAY).astype(np.float64))
energy = np.abs(dx) + np.abs(dy)
```

找能量最低的seam时可以通过动态规划完成

$$\mathbf{M}(i, j) = E(i, j) + \min(\mathbf{M}(i-1, j-1), \mathbf{M}(i-1, j), \mathbf{M}(i-1, j+1))$$

若使图像缩短 k 个像素，则依次去除 k 个能量最低的seam；若使图像拉长 k 个像素，则将能量最低的 k 个seam复制一份（找能量最低的 k 个seam时，可以依次去除 k 个能量最低的seam，并将它们记录下来）。

我选取了达利的油画《记忆的永恒》作为原图，分别缩短、拉长100个像素，效果如图1。

2 Object Removal

去除图像中的物体时，我仍使用了上节的方法，不断地去除seam使得目标区域完全消失，再进行拉长以恢复原大。为了使得目标区域优先被去除，将其中的像素点的能量值置为一个很小的数，这样一来使得每个seam可以去除尽量多的目标区域，二来目标区域外的seam段仍遵循着最小能量原则。我去除的目标是桌上的红色钟表，为此使用Windows10画图工具手动将其抠出，程序将非黑色背景的部分作为目标区域。效果如图2。



(a) 原图



(b) 缩短100个像素

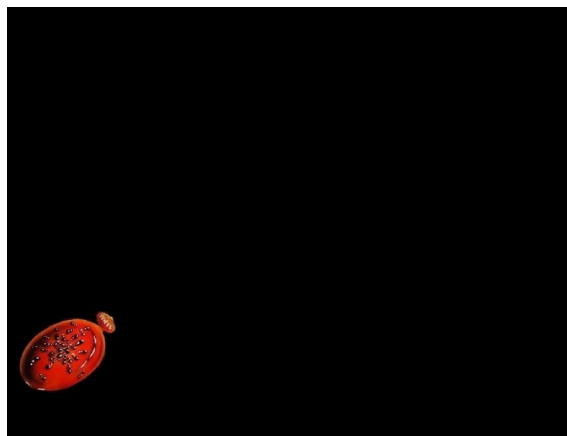


(c) 拉长100个像素

图 1: Image Resizing



(a) 原图



(b) 目标区域



(c) 去除目标区域



(d) 恢复原图大小

图 2: Object Removal