Assignment 04: Seam Carving

Brock Davis

Part 0: Input Images

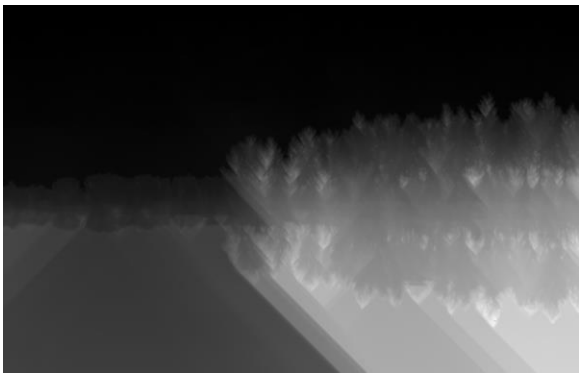
Image1.png (1000x640)

Part 1: Vertical Seaming

pic_1_a.png

This picture was made by taking the grayscale of image1 and taking the derivate of the picture to obtain edges in the y and x direction. Then the hypotenuse of the edges was found to obain corners.

```
derivative_kernel=[[-1,0,1],
                   [-2,0,2],
                   [-1,0,1]]
Ix=cv2.filter2D(img,-
1,derivative_kernel)
Iy=cv2.filter2D(img, -1,
derivative_kernel.T)
path=(Ix**2+Iy**2)**.5
```

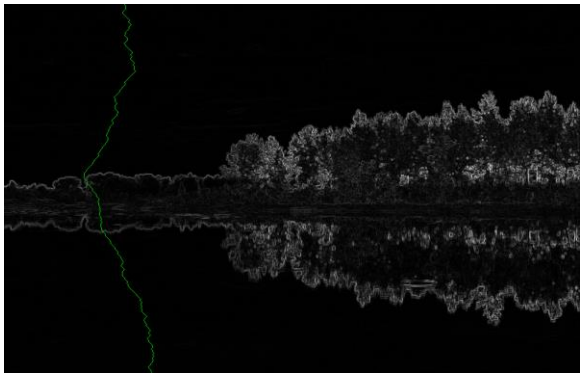


pic_1_b.png

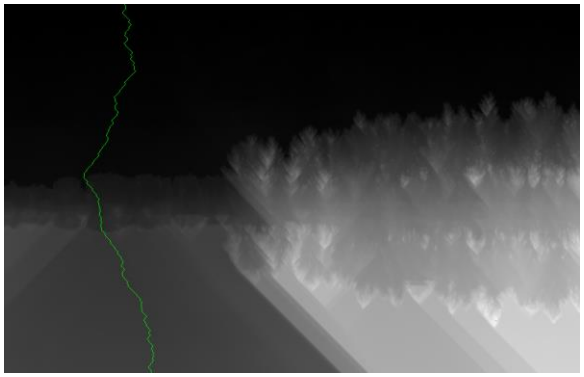
This image was created by using dynamic programming on pic_1_a to determine where to generate seams.

```
kernel=np.float32([[1,1,1]])
h,w=img.shape[:2]
for i in range(1,h):
    row=path[i-1,:]
    row=cv2.erode(row,kernel.T)
    path[i:i+1,:]+=row.T
```

pic_1_c_0.png



pic_1_c_1.png



pic_1_c_2.png

These images were created by showing the seam calculated using pic_1_b in the images image1, pic_1_a, and pic_1_b.

```
if len(img.shape)==2:
   out=np.dstack((img,img,img))
else:
   out=img.copy()
for i in range(img.shape[0]):
    out[i,seam[i]]=np.array(color)
return out
```

pic_1_d.png

This images was created by taking the seam highlighted in pic_1_c_0 out of image 1.

```
for i in range(h):
     row=img[i]
     row=np.concatenate((row[:
     seam[i]],row[seam[i]+1:])
     ,axis=0)
     out[i]=row
return out
```

pic_1_e.png

This image was taken by removing 120 vertical seams from image 1. The removeSeam functions was used 120 times in this process.

```
for i in range(seams):
    seam=getSeam(img)
    img=removeSeam(img,seam)
```

Part 2: Horizontal Seaming

p_2_a.png

This image was made by taking and highlighting a seam on image transformed.

```
trans_img= np.dstack((img[:,:,0].T,
     img[:,:,1].T,img[:,:,2].T))
showSeam(trans_img,
     getSeam(trans_img))
```

pic_2_b.png

This image was image1 after 120 horizontal seams were carved. This combines the concept of transforming the image in pic_2_a with removing multiple seams in pic_1_e.

Part 3: Both

pic_3_b_3.png

pic_3_b_0.png

pic_3_b_1.png

pic_3_b_2.png

These images were created using the retarget function. image1(1000x640px) was changed so fit the dimensions of 640x640(pic_3_b_3.png), 640x480(pic_3_b_2.png), 320x320(pic_3_b_1.png), and 320x240(pic_3_b_0.png).

```
def retarget(img,(w,h),show=False):
  img=img.copy()
  h_img,w_img=img.shape[:2]
  dh=h_img-h
  dw=w_img-w
  if dh<0 or dw<0:
    return img
  for i in range(max(dh,dw)):
    if i<dw:
      seam=getSeam(img)
      img=removeSeam(img,seam)
    if i<dh:
      trans_img=funcs.trans_img(img)
      seam=getSeam(trans_img)
      trans_img=removeSeam(trans_img,seam)
      img=funcs.trans_img(trans_img)
  return img
```