

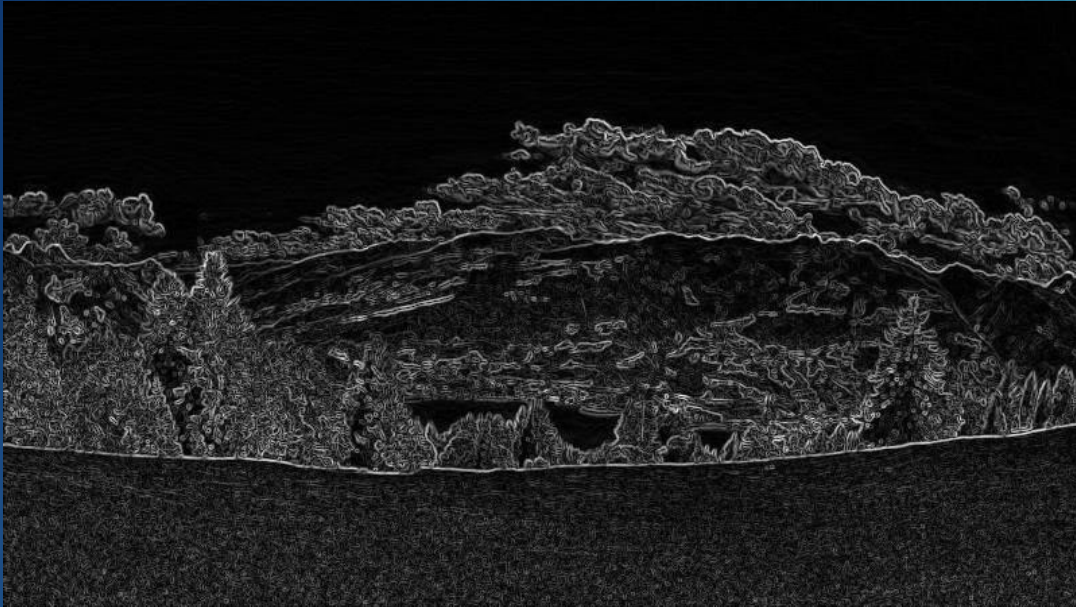
Seam Carving

NOAH SHERRY

Input Image



Vertical Seaming

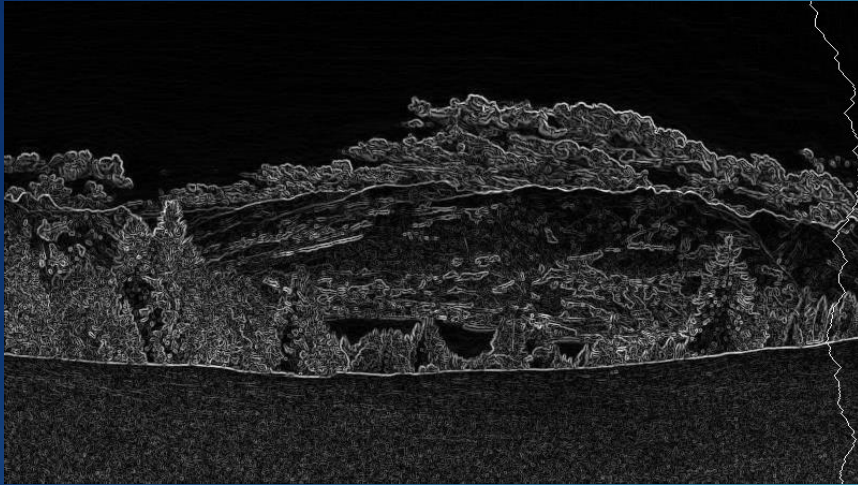


```
edge_kernel=np.float32([[1,2,1],[0,0,0],  
                        [-1,-2,-1]])  
Iy=cv2.filter2D(img*1.0,-1,edge_kernel)  
Ix=cv2.filter2D(img*1.0,-  
1,edge_kernel.T)    I=(Ix*Ix+Iy*Iy)**.5
```



```
x=np.vstack((left,row,right))  
x=x.min(axis=0)  
x+=I[i]  
O[i]=x
```

Vertical Seaming (cont.)



```
h,w = O.shape[:2]

x = np.argmin(O[h-1])
path=[[x,h-1]]
while path[0][1]>0:
    x,y=path[0]
    y-=1
    mid=O[y,x]
    left=mid
    if x>0:
        left=O[y,x-1]
    right=mid
    if x<w-1:
        right=O[y,x+1]
    if left<mid and left<right:
        path.insert(0,[x-1,y])
    elif right<mid and right<left:
        path.insert(0,[x+1,y])
    else:
        path.insert(0,[x,y])
return np.array(path)
```


Vertical Seaming (cont.)



<- 1 seam

```
i=0
rows=[]
for row in img:
    x,y=seam[i]
    newrow = np.vstack((row[:x],row[x+1:]))
    rows.append(newrow)
    i+=1
img=np.stack(rows)
return img
```



<- 50 seams

Horizontal Seaming



```
h,w = O.shape[:2]
y = np.argmin(O[:,w-1])
path=[[w-1,y]]
while path[0][0]>0:
    x,y=path[0]
    x-=1
    mid=O[y,x]
    down=mid
    if y<h-1:
        down=O[y+1,x]
    up=mid
    if y>0:
        up=O[y-1,x]

    if down<mid and down<up:
        path.insert(0,[x,y+1])
    elif up<mid and up<down:
        path.insert(0,[x,y-1])
    else:
        path.insert(0,[x,y])
return np.array(path)
```

```
i=0
cols=[]
for i in range (0,len(img[0])):
    col = img[:,i]
    x,y=seam[i]
    newcol = np.vstack((col[:y],col[y+1:]))
    cols.append(newcol)
cols = np.array(cols)
rows = [cols[:,0]]
for i in range (1,len(img)-1):
    rows.append(cols[:,i])
img = np.array(rows)
return img
```


Retargeting

```
def retarget(img, imgbw, finalw,finalh):  
    h,w = img.shape[:2]  
  
    while w>finalw:  
        img = removeFirstVerSeam(img)  
        print(str(w)+" "+str(finalw))  
        h,w = img.shape[:2]  
    while h>finalh:  
        img = removeFirstHorSeam(img)  
        print(str(h)+" "+str(finalh))  
        h,w = img.shape[:2]
```



Retargeting (Cont.)

