CS 378 Computer Vision

Problem Set 1

CHAO-YEH CHEN

CS account: chaoyeh

II. Programming problem

5. Real results!

A. Groceries

Original picture: 375x500
Reduce width: 375x380
Increase Width: 375x600

The path in the center has less energy than the cans in the two sides. When we reduce the width by seam carving algorithm, the width of the path reduce but the size of the cans is preserved.

Similarly, when we increase the width of the image, the width of path is increased while the size of cans remain the same.

Fig. 9

1	2	3
4		
5		

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function

B. Trees

Original picture: 375x500
Reduce width: 375x380
Increase Width: 375x600

At first, I expected the path in the image will be narrowed when we reduce the width. However, the result shows that the width of trees in the right hand side is reduced dramatically.

The reason is that the surface of path and the grass is not smooth compared to the surface of the trunk of the trees in the right.

When increasing the width, the seam carving result shows wider trees in the right but the width of path in the center is similar to the original image.

Fig. 10

1	2	3
4		
5		

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function

C. Seals

Original picture: 375x500
Reduce height: 225x500
Increase Width: 375x700

Fig. 11

1
2
3
4
5

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function

There are several parts in the image contain large energy: seals, barrier and handrails.

When we reduce the height, the height of barrier and seals remain the same as the original image. The sky and space between seals is compressed.

Similarly, when we increase the width of the image, seam carving algorithm preserves the width of some seals' body.

D. Me

Original picture: 480x640
Reduce width: 480x470
Increase Width: 480x840

Fig. 12

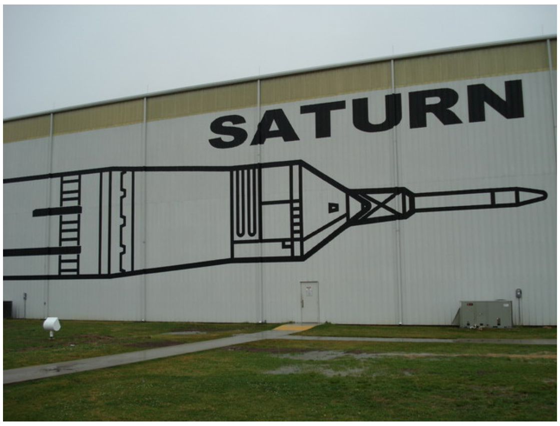
1	
2	3
4	
5	

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function

The image contains only one portrait. The background is very simple and smooth.

The results of reducing and increasing width show that the seam carving algorithm can preserve the size of the portrait effectively.

E. Saturn

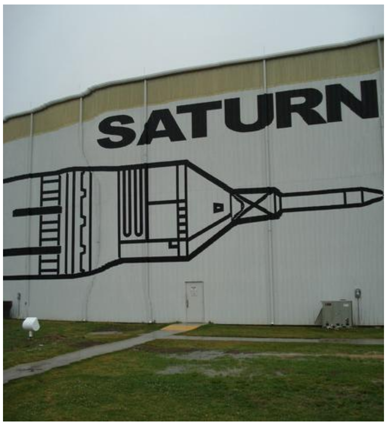


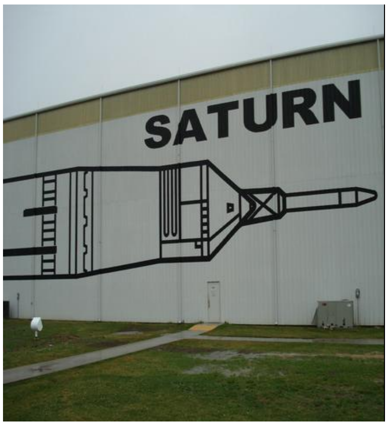
Original picture: 480x640
Reduce width: 480x440
Increase Width: 480x890

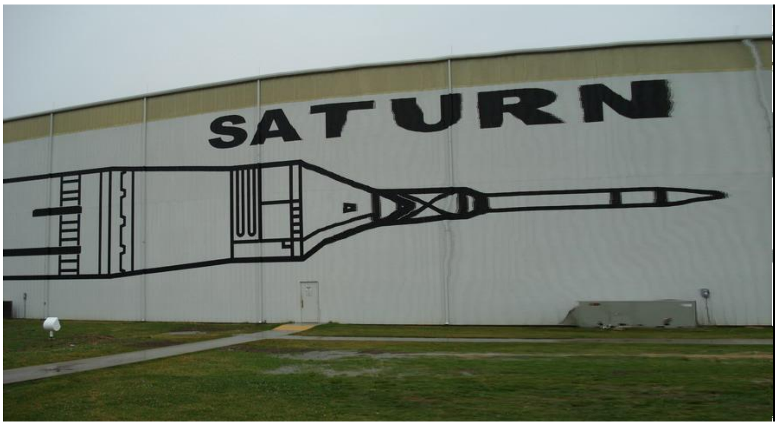
Fig. 13

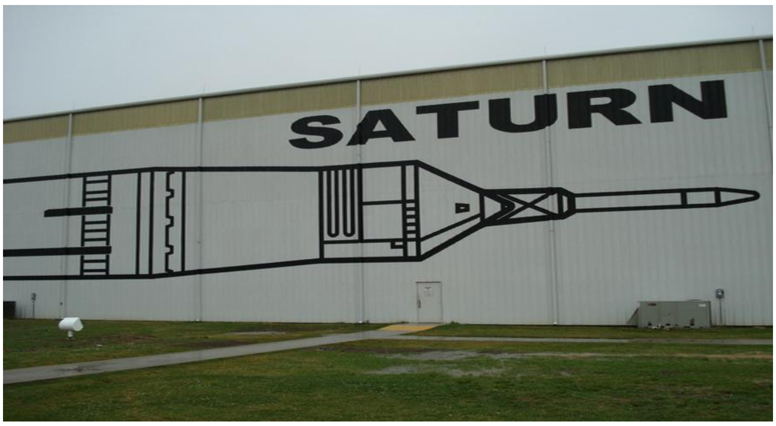
1	
2	3
4	
5	

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function







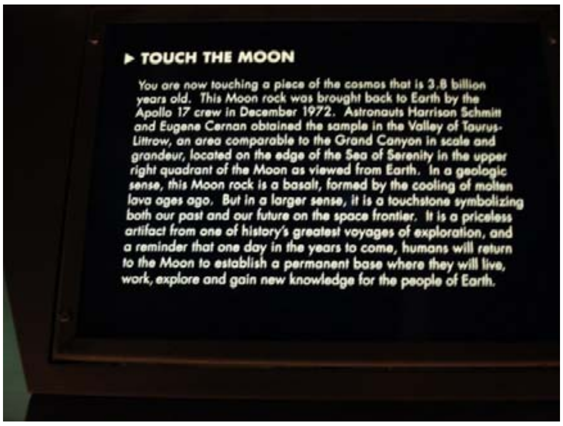


First let's look at the reduce width result. The rocket body is compressed in different ways. For the rocket parts that contain more complicated pattern, the width can be preserved better than some simple parts. Therefore, the relative width of parts of rocket is changed.

For the case that increases the width, we find that the front part of the rocket is stretched much more than other parts. This is because the front part contains fewer edges than other parts.

The parts being compressed and stretched are different. It is because that for the reducing width process, we find the path with global minimum cumulative energy for many times while for increasing width process we find first n minimal paths at one time.

F. Words

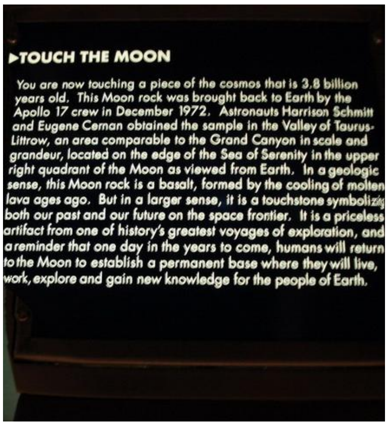


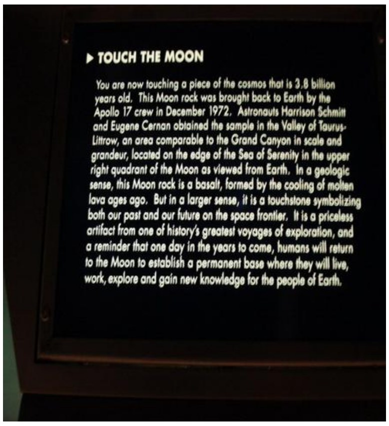
Original picture: 480x640
Reduce width: 480x520
Increase Width: 480x840

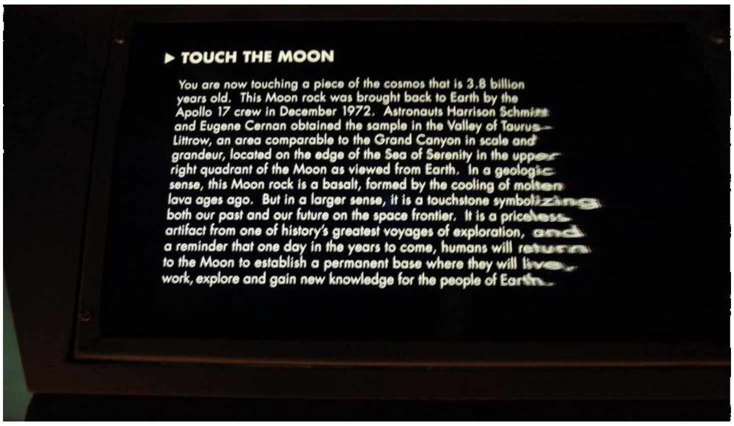
Fig. 14

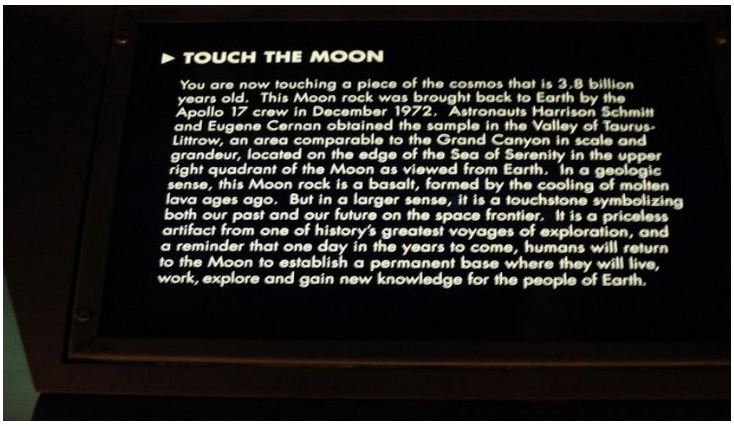
1	
2	3
4	
5	

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function









This is a good example for seam carving algorithm.

The words in the images can be preserved when we reduce the width.

However, when we increase the width for a large amount at a time, some of the words will be slightly stretched.

G. Lake

Original picture: 480x640
Reduce width: 480x440
Increase Width: 480x940

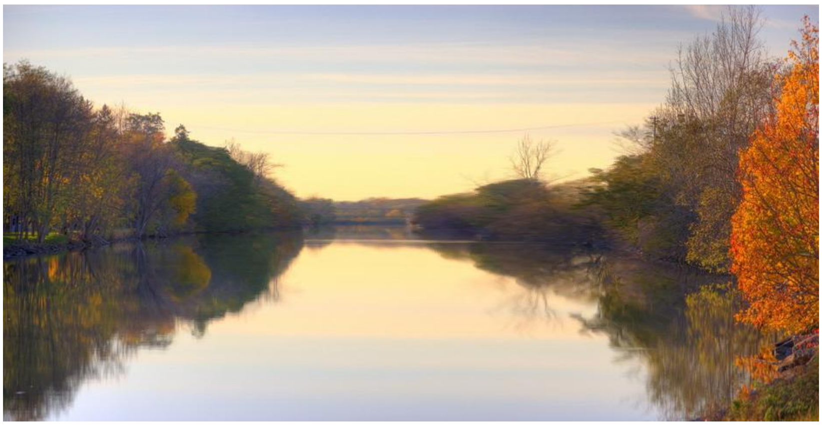
Fig. 15

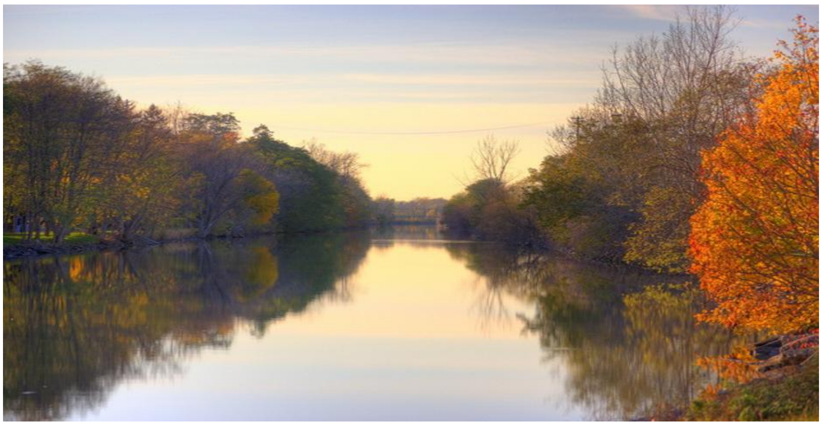
1	
2	3
4	
5	

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function

The river and the sky in the center of the image have lower energy than other parts.

Therefore, when we reduce or increase the width of the image, the river and sky in the center will be compressed or stretched.





"lake" is provided by Tzeng-Yu

H. Me

Original picture: 480x640
Reduce width: 480x440
Increase Width: 480x840

Fig. 16

1	
2	3
4	
5	

1: Original image
2,4: Seam carving result image
3,5: Image resize by matlab function

Similar to D, the background is simple compared to the portrait in the front.

We can preserve the size of portrait in reducing or increasing width.

III. Extra Credit

1. Mark an object to be removed

If we want to remove an object when reducing the width of the image, we can let the seam that we will cut out go through the object that we want to remove.

A simple method to do this is making the energy of the object become small. Then the chance that seams going through it will increase.

My program lets the user choose four points to form the boundary of the object that want to remove. The energy of points inside the boundary will become zero. In addition, the width of the boundary will be calculated and the image will reduce its width in the same amount.

1: original image
2:I mage resize by matlab
3: seam carving result
4: seam carving with object removal

Original seam carving algorithm will preserve the tree in the center while the object removal version can make the tree almost disappear.

Fig. 17

1	2
3	4

Original picture: 480x640
Reduce width: 480x258

“Trees” is provided by Jolca

2. **Avoid warping regions containing people's faces**

When we reduce or increase the width of this image by original seam carving algorithm, the portrait can be preserved well. However, if we want to reduce the height of the image, the minimal horizontal cumulative energy seams will cross human face since the boundary of clothes generates larger energy output.

To preserve the parts of human skin, we first change the RGB image to the HSV color space. After several observations, I found my skin color is located between 0.03 and 0.05 in the hue channel. Thus the parts of human skin can be detected. After that, we can increase the energy of the human skin parts to prevent the seams cross them.

Besides, I found that some edges in the clothes have similar hue value as human body. We don't want to increase their energy. Thus, by the fact that human skin are smooth since they are in the same color, I only increase the energy of the pixel which is located in the range of hue of skin color and has low energy originally. The strategy avoids treating the edge of other object as human skin parts.

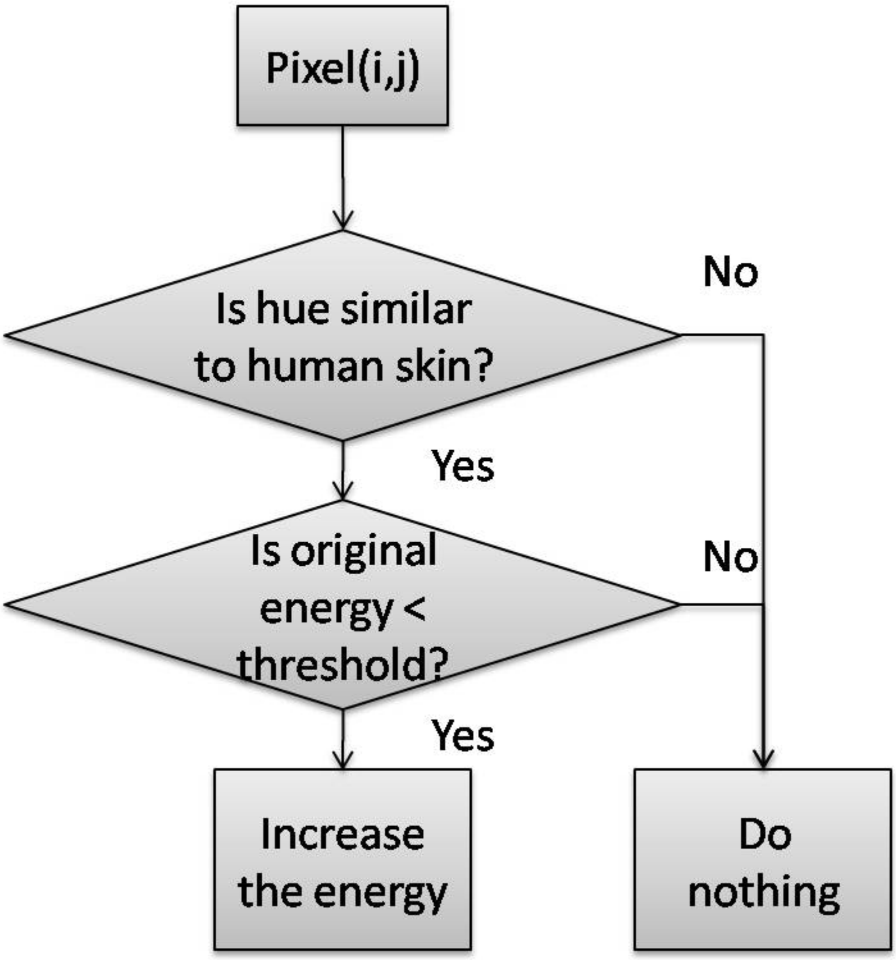


Fig. 18 Detecting human skin and avoiding edges

The result images:

Original picture: 480x640
Reduce height: 380x640

Fig. 19

1	2
3	4

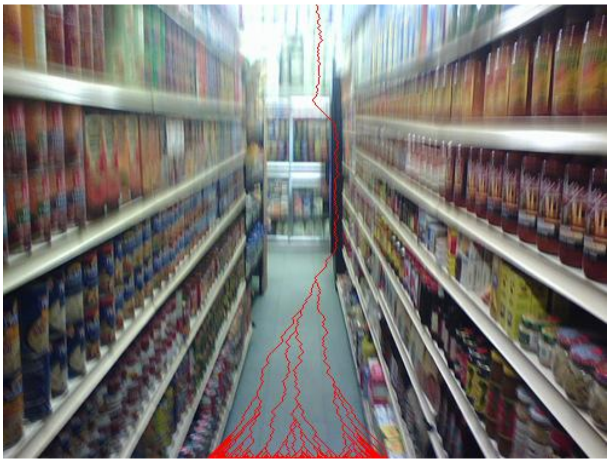
1: original image
2: Image resize by matlab
3: seam carving Avoid warping regions containing people's faces
4: seam carving result

Original seam carving algorithm will cut some part of the human faces while the new method can preserve the size of human face.

3. **Improved increasing width strategy.**

When we want to increase the width of the image by N, we cannot repeat increasing the width by one for N times. The reason is that the minimal cumulative energy seam will be all the same. Thus we will just repeat the same seam for N times and make the image unnatural.

Instead of that, we must choose the first N smallest cumulative energy seams to enlarge the image. However, the problem is that the first N smallest seams are coming from very similar paths. As in the following image, if we increase the width by doing this, we just repeat the same seam for N times.



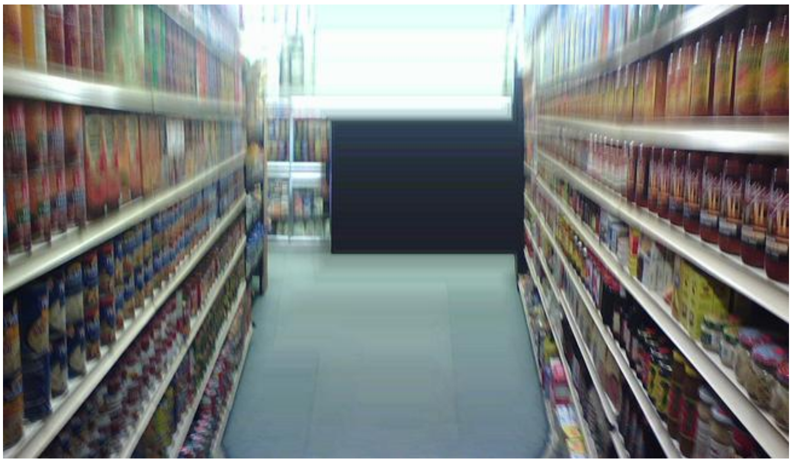


Fig. 20

1
2

1: the first 150 cumulative energy seam

2: increase width image by interpolating the seam with neighbor pixels

The resulting image is artificial

To solve this problem, an adjustment is necessary. When the first N minimum seams are tracing back for the cumulative path, the system forbids going through a pixel twice. This means the first minimum cumulative energy seam will be the same as before. But for the second one, it will begin in the bottom pixel and look for the smallest cumulative pixel except the pixels in the first seam.

This strategy can prevent all of the seams converge to the same path.

The first few seams will be the same as the seams in original algorithm. However, the following seams will meet the "barrier" from the first few seams and they can only choose other ways. As a result, the following seams begin to go through the pixel that doesn't attain low energy. We will enlarge some part that we want to preserve.

Fig. 21

1
2

1: the first 150 cumulative energy seam
2: increase width image by interpolating the seam with neighbor pixels

The resulting image is corrupt in some parts we want to preserve

Finally, a tradeoff is made to solve the problem.

Each time the pixel passed by the seam will slightly increase its cumulative energy. The amount of increasing is proportional to its original cumulative energy, position, and the order of seams. The amount of increasing is proportional to the order of seams. Besides, the amount of increasing is large in the lower position since most of the pixel has large cumulative energy here.

This action makes the seams less likely to being passed by the next seams but the first few seams will not form a barrier for the later seams

The resulting enlarged image is better-looking than previous ones.

The cumulated energy formula:

M(i,j)_after being passed = M(i,j)_origin + 0.1 * M(i,j)_origin * (i)^(1/3) * (order of this seam)^(1/3)

Fig. 22
1
2

1: the first 150 cumulative energy seam

2: increase width image by interpolating the seam with neighbor pixels