Image 1: seals.jpg  source: assignment

Original Image

System Generated Image
Input Dimension 375 x 500

Output Dimension 300 x 400

Sequence V 20 H 20 V 30 H 30 V 10 H 25 V 40

Explanation

Bridge and Seals are the important regions in the image. The gradient change near the sea would be very less hence it forms the low energy region thus is removed more.
Image 2: trees.jpg    source: assignment

Original Image

System Generated Image
The bark of trees form repetitive patterns and hence forms the low energy regions. That’s why we see some thinning there, because they are the ones to be removed first.
Input Dimension 375 x 500
Output Dimension 325 x 450
Sequence V 25 H 50 V 25
Explanation
Except the ground all other regions are pretty distinct hence important that’s why we see the path getting reduced.
SUOY DUTT JAIN – PSET1 – CS LOGIN: suyog

Image 4: algo.jpg  source: http://www.di.unipi.it/fun07/DisegnoFunRuotato.jpg

Original Image

System Generated Image
Matlab Resized Image

Input Dimension 405 x 543

Output Dimension 300 x 450

Sequence V 93 H 105

Explanation

The characters form the dominant content in the image hence we see the background getting reduced and the letters coming closer.
Image 5: building.jpg source: http://eigenclass.org/hiki/
Original Image

System Generated Image
Matlab Resized Image

Input Dimension 450 x 600
Output Dimension 350 x 480
Sequence V 120 H 100

Explanation

They sky and water forms the unimportant region hence after seam removals we observe that the prominent content i.e. buildings are retained as it is and appear closer.
Original Image
In this region almost all regions are equally important, that's why we observe a squeezing effect in this image.

Original Image

System Generated Image
Matlab Resized Image

Input Dimension 480 x 640

Output Dimension 250 x 500

Sequence V 140 H 230

Explanation

The floor forms a repeating pattern and the objects are the important part of the image that’s why after seam removal we see the objects coming little closer.
Image 8: Kupka.jpg
Original Image

System Generated Image
Input Dimension 437 x 520
Output Dimension 350 x 400
Sequence V 120 H 87

Explanation

The brownish pattern as well as the water contribute to the low energy region in the image and thus are removed during resizing.

Input Dimension 500 x 332

Output Dimension 500 x 250

Sequence V 82

Explanation

The entire top region of the duck’s body is consist of the same texture and hence we see very little gradient change in horizontal direction therefore we see some thinning effect on the duck after removal of vertical seams.

Original Image

System Generated Image

Matlab Resized Image
Explanation

The rock is most probably, acting as more important object in this image and thus we observe this artifact.
Matlab Resized Image

Input Dimension 458 x 458
Output Dimension 350 x 350
Sequence V 108 H 108

Explanation

The skin color forms the major part of this image and thus seen as an unimportant region. That’s why we get a distorted face after resizing.

source: http://www.photoshopessentials.com/images/photo-effects/photo-fill/main-cropped.jpg
EXTRA CREDIT:

1. Using Entropy to calculate energy function instead of gradient:

Calculating energy function using entropy will use the distribution of color and brightness instead of gradients and I used ‘entropyfilt’ function in Matlab to do so. It seems pretty intuitive because the regions are which are uniform in the image will have very less randomness and thus low entropy hence it will contribute in the optimal seam removal by having the lowest energy. The idea works out well in some cases but there are cases in which gradient based energy function works better. It entirely depends on the type of image. But I observed that using entropy makes the removal process really slow.

To see the results of my implementation, in vertical_seam.m comment line 3 and uncomment line 4 and then call reduce_width or reduce_height as desired.


Example 1: Reducing from 384 x 512 to 300 x 400

Original Image:
We can see that using entropy as energy function produces better results in the above example. Because there are patches in this image which are very uniform and hence they are getting removed first, thus it’s avoiding thinning of the tree which can be seen if we use gradient as the energy function.
Example 2: Reducing from 378 x 512 to 280 x 400

Reduced using entropy:
Reduced using gradient:

We can see that using entropy as energy function does not produce good results in the above example. There are artifacts in both the images but it’s much lesser if we use the gradient to calculate energy.
2. Seam Insertion:

I have implemented seam insertion as told in the paper. The three additional functions are `compute_seam_matrix`, `increase_height`, `increase_width`. They can be used for seam insertions. For seam insertion, I calculate the optimal seams to be removed and store their order. Then insert all the seams orderly by averaging the values of neighboring pixels for every inserted seam.

Some results for seam insertion:

1. Original Image:
Increase Height and Width

Decrease Height and Increase Width
2. Original Image
Increase Height and Increase Width

Decrease Height and Increase Width
3. Developing a GUI for doing all operations:

I developed a GUI to perform basic operations. It provides functionalities to load an image and then specify the new dimensions you need before clicking resize. It supports both decreasing and increasing the size of the existing images. Even decreasing height and increasing width or vice versa works. There is a button to view optimal vertical and horizontal seam. There is an option to save the modified image as well. It saves the new image in the folder of existing image by adding a prefix ‘result_’ before the file name.

To use the GUI please open the file SeamGUI.m in Matlab and press F5.

Some snapshots:
After Loading Image it tells the dimension of new image:

After new dimensions and pressing resize:
On Pressing Show Seam Button: