Computational Photography
Homework 3 – Image Retargeting through Seam Carving

Amiya R Panda
UIN - 727006179

Method

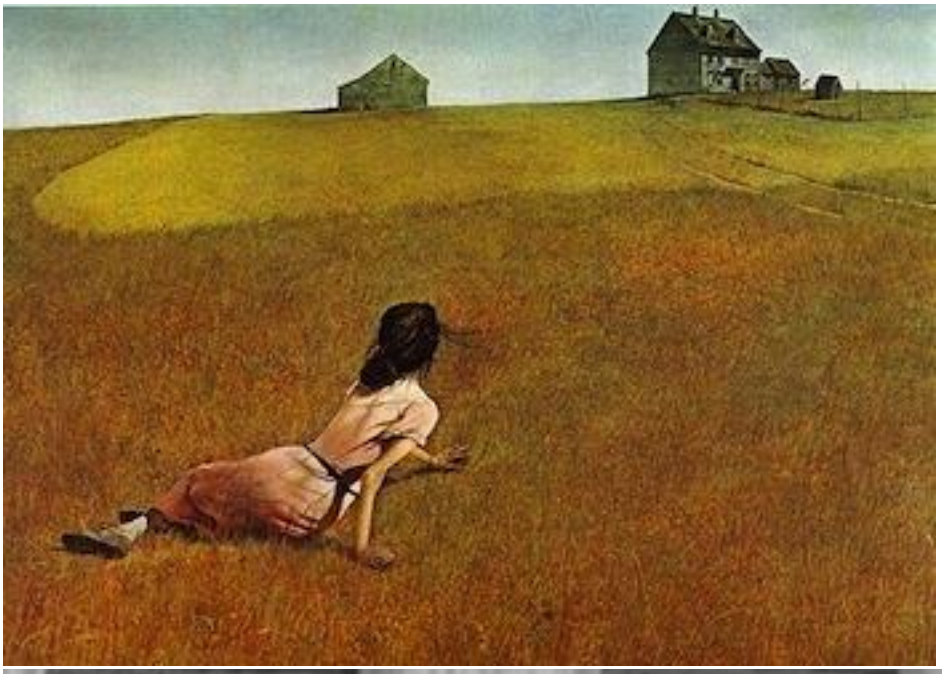
1. Initially, the images are loaded into MATLAB and along with the mask (if given). If mask is not given, that is, there is no requirement to preserve a certain area in the image then We construct a mask of size equal to that of the input image (excluding color dimension).
2. The main target of the assignment to reduce the size of the input image vertically and horizontally keeping the most important information intact. As human eyes are more sensitive to gradients, we have considered it as the energy function. The code is written in order to eliminate the seams (connected set of pixels) with lowest energy, hence lowest information.
3. First, the cost function or energy is defined and is calculated by sum the absolute value of the gradient in both x direction and y direction for the gray image which is achieved by converting the original input image by Using “rgb2gray ()” function. We have also experimented with the absolute of the sum of gradient in x and y direction but found the output slightly bad for “Image_03”.
4. Then, accumulated cost matrix using forward energy is constructed. This step is implemented with dynamic programming. The value of each pixel is equal to its corresponding value in the energy map added to the minimum new neighbor energy introduced by removing one of its three top neighbors (top-left, top-center, and top-right).
5. Then, backtracking is done from the bottom to the top edge of the accumulated cost matrix to find the minimum seam. All the pixels in each row after the pixel to be removed are shifted over one column to the left if it has index greater than the minimum seam.
6. The above-mentioned steps are repeated till the desired width or height is achieved.
7. For the final image, in order to preserve the faces of image, before generating energy map, the region protected by mask are weighted with a very high positive value. This guarantees that the minimum seam will NOT be routed through the masked region so that we can prevent pixels at this region from removing or distorting because of seam insertion.

Observation

Although, dynamic programming is used for the implementation, it takes considerably large amount of time to complete the task. Further improvements are required to optimize in terms of time. Further, we can experiment with different energy functions for more accuracy. This technique will not work properly with images with rich contents (spread across), or text, or faces. Ad in order to tackle these problems, further improvements are required.

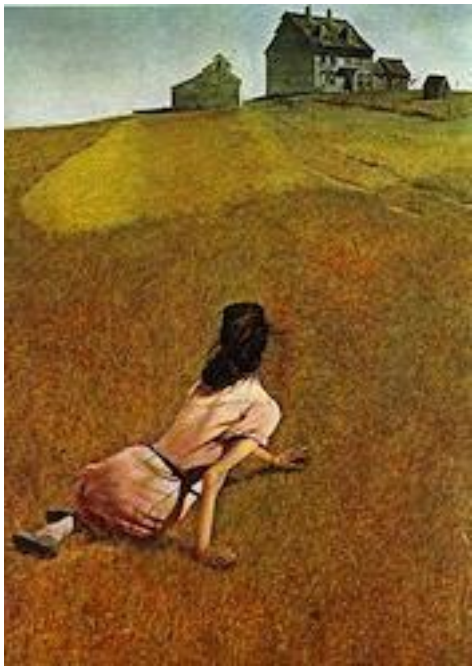
Below mentioned are the time required for completion of the task for provided images.

ORIGINAL IMAGES:




CIMARRON

VERTICAL CUT:



HORIZONTAL CUT:

For vertical cut:

Image_01 ...Elapsed time is 1318.2 seconds.

image_02 ...Elapsed time is 2333.24 seconds.

image_03 ...Elapsed time is 361.779 seconds.

image_04 ...Elapsed time is 1995.43 seconds.

For vertical cut:

Image_01 ...Elapsed time is 1331.42 seconds.

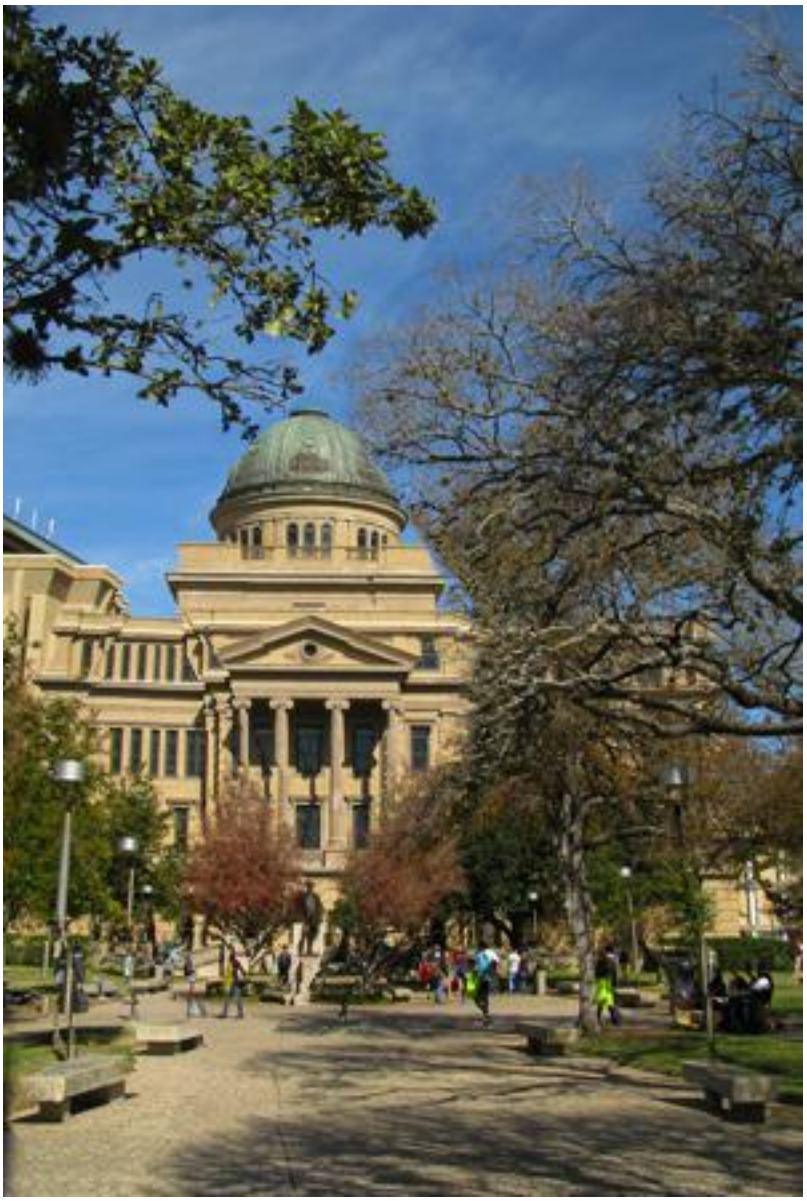
image_02 ...Elapsed time is 1939.29 seconds.

image_03 ...Elapsed time is 461.459 seconds.

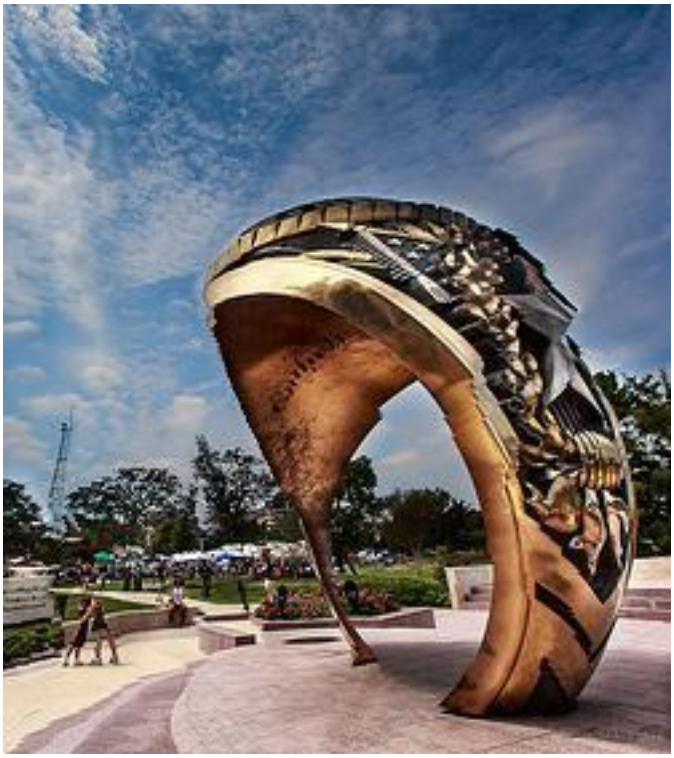
image_04 ...Elapsed time is 1949.38 seconds.

EXAMPLE 1:

Example 2:



Example 3:



We can see the artifacts for the vertical cut as the richness of the image is more and spread throughout the image. For proper working of the image, we need to use the masking technique as used for image_04. This is also evident for a few other examples given.