Seam Carver

**Problem Statement** : The intention of the project is to reduce the size of the image , either vertically or horizontally . The removal of seam ( either horizontal or vertical) should be one pixel height or width . Resizing using this technique would preserve the basic aspects of a picture like ; aspect ratio , set of objects etc .

Original Image Resized Image

 

**Concepts related to the project :**

Vertical Seam – One pixel per row is removed from top to bottom . Their indices put together form a vertical seam .

Horizontal Seam – One pixel per row is removed from left to right . Their indices put together form a horizontal seam .

Topological sort – In order to find the seam, we need to find the shortest path between 2 points . For that purpose ,we use topological sort .

**Procedure followed in the program / functions used :**

* The picture object is created which will later be assigned to the input that we send . We then create an array which will accommodate the energies at each pixel of the picture .
* The energy has been calculated in a private method energyar() , where an array called energyarr (width x height ) is assigned 1000 at its borders . The rest of the elements in the array are filled by calling another private method called energy . Here colour objects are got by sending x and y parameter as the input and then their rbg values are retrieved .
* Another array called eneryarrt (height X width ) is created by transposing the energyarr for horizontal seam calculation .
* findVerticalSeam () – Here we will call the energyar() to initialize both the arrays .Then cumulative energies are calculated using topological sort (from top to bottom) on energyarrt matrix by sending to a private function called cumulativevert() . After the cumulative matrix is found , the returned array is sent as an input to another private function called seamcalc() where the seam is calculated.
* findHorizontalSeam () - Here we will call the energyar() to initialize both the arrays .Then cumulative energies are calculated using topological sort (from top to bottom) on energyarr matrix by sending to a private function called cumulativehorizontal () . After the cumulative matrix is found, the returned array is sent as an input to another private function called seamcalc() where the seam is calculated.
* removehorizontalSeam (int[] seam )- Seam array is sent as an input to this function . Here 2 loops are run and when the pixel to be removed is encountered , then the pixel below it will take its position . The rest of the pixels will remain the same .
* removeVerticalSeam(int [] seam ) - removehorizontalSeam (int[] seam )- Seam array is sent as an input to this function . Here 2 loops are run and when the pixel to be removed is encountered , then the pixel right it will take its position . The rest of the pixels will remain the same.

Test cases :

Initially without writing all the methods I wrote only energy method and submitted and got 61/100. Later started to write the other methods where the score became less and less to 51/100. Later I did some changes in the find seam method and resubmitted it and got to know that we need to handle the edge case of 8x1, 1x1 and 1x8 pictures. So handled them and got 53/100. Now I have handled some exceptions and got 65/100. Now I have some code errors where my height and widths that I need to consider were wrong. So I have changes them and got 75/100. Then again I did not call a method in find vertical and horizontal seams and finally got 81/100.

Complexities:

* picture() - O(H+W)
* width() -O(W)
* Height() - O(H)
* energyar() - O(H\*W)
* energy(int x, int y) -O (1)
* seamcalc(double [][] array) - O(N)
* findVerticalSeam()- O(N)
* mattranspose(double [][] array) - O(H\*W)
* findHorizontalSeam()- O(N)
* removeHorizontalSeam() -O(H\*W)
* removeVertcicalSeam() -O(H\*W)