**SEAMCARVER PROJECT REPORT**

**INDEX**:

1. Problem Statement

2. Related Concepts

3. Code

4. Test Cases

5. Complexity

**Problem Statement:**

Seam-carving is a content-aware image resizing technique where the image is reduced in size by one pixel of height (or width) at a time. A vertical seam in an image is a path of pixels connected from the top to the bottom with one pixel in each row; a horizontal seam is a path of pixels connected from the left to the right with one pixel in each column.

Energy Calculation: Calculates the energy of each pixel. The energy of pixel(x,y) is, under-root of delta(x) square of and delta(y) square. Delta(x) square is equal to sum of squares of Red(x,y), Green(x,y) and Blue(x,y).

**Related Concepts:**

Some of the concepts and data structures that we have used in building the SeamCarver are:

* Picture
* ResizeDemo
* StopWatch

**Code**

Classes:

1. SeamCarver

**SeamCarver:**

* SeamCarver(Picture picture) – A constructor that initializes a new object of Picture class.
* Energy(int x, int y) – calculates the energy of each pixel of the image after assigning the energy of the edges as 1000. We define the Color using picture class and calculate the delta values using the formula.
* findVerticalSeam() – if the axis is vertical, then we define an array distTo that defines the distance between them with width size. We also create a 2D edgeTo array. We use relax method to find the vertical seam that calculates the least energy cost and adds it to the present pixel cost. We thus get the minimum energy pixel in each row which results in the seam.
* findHorizontalSeam() - The same process is applied for calculating the horizontal seam.
* relax(int prev, int x, int y, double[] lastDistTo, Boolean axis) – This checks the energies of each pixel and the adjacent 3 pixels for the least energy and adds it to the present cost. It also updates the previous edges to the edgeTo array.
* removeVerticalSeam(int[] seam) – It creates a new picture object with size decreased by 1 each time this method executes. We use color of the existing pixel that needs to be removed, to set the color of the new image. We iterate through height and width and if the pixel is to the right of the seam, it is shifted left to remove the resp. pixel. We set the new picture as the existing picture.
* removeHorizontalSeam(int[] seam) – It creates a new picture object with size decreased by 1 each time this method executes. We use color of the existing pixel that needs to be removed, to set the color of the new image. We iterate through width and height and if the pixel is to the bottom of the seam, it is shifted up to remove the resp. pixel. We set the new picture as the existing picture.