seam-carving

Seam carving algorithm using forward energy and original energy

Requirements

- opencv-python
- numpy
- imageio

Usage

```
python seam_carving.py -input <IMAGE_PATH>  -size <REMOVAL_SEAM_NUMBERS>
                        [-output <OUTPUT_IMAGE_PATH>] [-energyfn ("forward" | "origin")] [-direction ("horizontal" | "vertical")]
```

- `-input` : input image path.
- `-size` : number of seams is going to be removed
- `-output` : (Optional) output image path.
- `-energyfn` :(Opional) choose which function to compute the energy
- `-direction` : (Optional) choose vertical or horizontal.

Example

The input image is on the left and the result of the algorithm is on the right.

Vertical Seam Removal

Horizontal Seam Removal

Complexity

def get_min_seam_mask(image): impelements the DP for computing the minimum-cost seam in the image Time Complexity : It iterates on all the image pixels and compute the path up to that pixel each time with O(1). So the overall time complexity is O(width * height). Space Complexity : Is uses an array the size of the image so the overall space complexity is O(width * height)

Comparison between Energy Functions

Forward enegrgy gives better results than origin function. Forward energe is faster because it is a dp algorithm

The result of resizing of the Forward energy(left picture).The result of the resizing of the origin energy(Right picture)

More information about Forward energy function on https://avikdas.com/2019/07/29/improved-seam-carving-with-forward-energy.html.

Refrences

Some parts of the code(forward energe function) are used from other implementations: * https://github.com/andrewdcampbell/seam-carving