Seam Carving

November 11, 2016

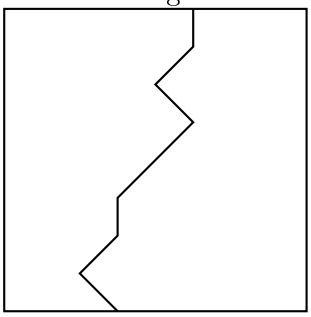
You can resize an image by scaling or cropping it, but what if the pieces of the image that you want are not all in one rectangular area, and you don't want to make those parts of the image smaller by scaling?[1]

In that case, you might instead choose to eliminate one pixel from each row (to make the image one pixel narrower) or one pixel from each column (to make the image shorter) while somehow optimizing for the "best" pixels to remove. In this problem, we focus on removing one pixel from each row.

We'll assume an image is an n column by m row array of pixels $A[1 \ldots n][1 \ldots m]$, where each pixel is an "energy" rather than a color. Energies are non-negative numbers representing the importance of the pixel.

A legal seam must include one pixel from every row. Each pair of seam pixels in neighbouring rows must be either in the same column or one column apart (i.e., on a diagonal). The cost of a seam is the total energy of all the pixels in the seam. The best seam is the one with lowest cost.

So, a seam of pixels to remove often looks a little like a "lightning bolt" moving down, down-and-left, and down-and-right from the top to the bottom of the image, such as this:



1. Circle two non-overlapping seams in this diagram that have different costs. Indicate their costs and which seam is better:

```
1 8 7 5 6 2 4
9 5 1 2 8 8 7
6 6 2 1 9 5 4
```

2. Give a recurrence for the cost of the best partial seam that has pixels only from row 0 up to i and ends at the pixel in row i and column j. Your recurrence should be in terms of the seams ending at pixels in the row above, row $i-1$. Assume $i > 0$.

```
C(i, j) =
```

3. Give the cost for a partial seam that only has a pixel in the very first row, $i = 0$. (This is our base case.)

```
C(0, j) =
```

[1] This method was developed by Avidan and Shamir.

1 Dynamic Programming

1. Give a pseudocode algorithm that finds the cost of the best seam in an energy array using dynamic programming.
2. Give a pseudocode algorithm that takes the dynamic programming table and produces the column numbers of the pixels in the best seam. (So, if the best seam has a pixel at column 3 in the first row and column 4 in the second, then your solution should give the list $[3, 4]$.)

2 Bonus: Implement Awesomely

Look up seam carving and design an implementation. Maybe extend it to apply to video!