Drew Hurdle
September 6th 2014
Algorithms and Data Structures II
Professor Cummings

Seam Carving

Section I : Description of the algorithm

The Seam Carving algorithm is, at its core, a shortest-path algorithm. It uses an assumed acyclic graph representation of pixels which guarantees that the pixel's natural ordering can be used for a topological sort.

My algorithm uses a helper class called TopoPixel to store data about individual pixels. This data includes the cost from the source, the parent pixel (as an int, explained later), the cost (energy) of this pixel, and the cost of the parent.

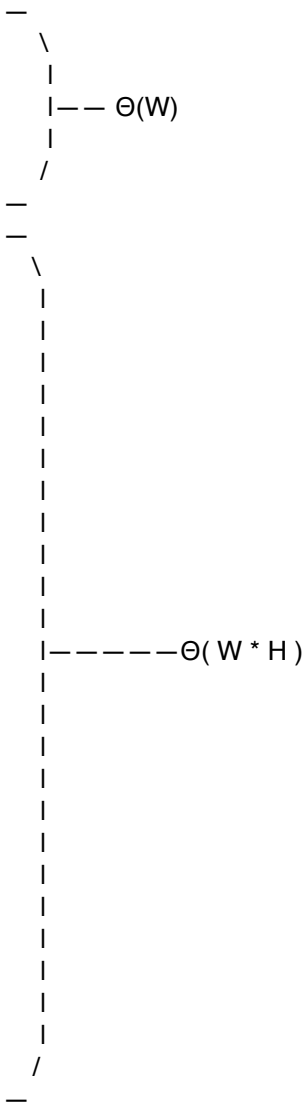
How it works:

- Create a matrix of TopoPixels with the same dimensions as the image (width X height)
- For each pixel in the first row/column of pixels in the image,
  - Calculate the energy for the pixel
  - Add it to the matrix in the location corresponding to the pixel's position in the image.
- For every pixel in the image,
  - For the three (max) child pixels of the pixel (the three adjacent or diagonal pixels which are in a further row/column from the starting row/column)
    - If they don't exist in the matrix of TopoPixels
      - Create it, calculate its energy, and add it to the matrix with the parent cost of the current pixel's costFromSource and a parent value of the current pixel's X or Y (depending on direction of the seam).
    - Otherwise
      - If the child pixel's parentCost is greater than the costFromSource of the current pixel
        - Replace the parent value of the child pixel with the current pixel's X or Y and update the parentCost and costFromSource according to this change.
- Find the cheapest costFromSource value in the last row/column of the matrix.
- Follow the parent link back through the matrix, storing the parent integers in an array (backwards).
- Return the resulting array.

Section II : Θ Analysis of the algorithm

- Create a matrix of TopoPixels with the same dimensions as the image (width X height)
- For each pixel in the first row/column of pixels in the image,
  - Calculate the energy for the pixel
  - Add it to the matrix in the location corresponding to the pixel's position in the image.
- For every pixel in the image,
  - For the three (max) child pixels of the pixel (the three adjacent or diagonal pixels which are in a further row/column from the starting row/column)
    - If they don't exist in the matrix of TopoPixels
      - Create it, calculate its energy, and add it to the matrix with the parent cost of the current pixel's costFromSource and a parent value of the current pixel's X or Y (depending on direction of the seam).
    - Otherwise
      - If the child pixel's parentCost is greater than the costFromSource of the current pixel
        - Replace the parent value of the child pixel with the current pixel's X or Y and update the parentCost and costFromSource according to this change.
- Find the cheapest costFromSource value in the last row/column of the matrix.
- Follow the parent link back through the matrix, storing the parent integers in an array (backwards).
- Return the resulting array.

For this analysis we'll be assuming a vertical seam for simplifying the explanations



	Total : W + (W * H) + H + W = 2W + H + (WH)
	Linear : Θ(W * H)

Explanation of analysis

All of the operations the algorithm performs are linear with respect to W or H or (W * H).

The first is the addition of each pixel in the first row, which is W for obvious reasons.

The second is the main double-for loop going through every pixel in the image and calculating its children's parents and energies. This is (W * H) because all the operations performed on each pixel is constant.

The third is the finding of the cheapest pixel in the last row. This is H.

The fourth and last is the following of the cheapest pixel's parents up through the matrix. This is W.

Because every part of the total is linear, we classify it simply as the largest of the linear components (W * H).

Help Received

None. As with most labs, I discussed the concepts of the lab with Jon before, during, and after finishing it, but nothing detailing how anything worked (at least until we were both done with it).