Computer Graphics – Spring 2013

**Exercise 1 – Seam Carving**

Submission: Thursday, April 4th 2013

Seam carving is a novel algorithm for resizing images while maintaining as much information as possible from the source image. A "seam" in this context is an 8-connected path of pixels from the top of the image to the bottom or from the left to the right.

Seam carving uses a dynamic programming method to compute a directed energy map over the image. Using this map it finds a seam with the least energy. Removing this seam produces a smaller image and applying this process repeatedly allows reducing the size of the image freely and changing its aspect ratio.

In order to increase the size of the image, seam carving finds *k* seams with the least energy and duplicates them in ascending order.

**Part 1**

Please read the Seam Carving paper here:

<http://www.faculty.idc.ac.il/arik/papers/imret.pdf>

In essence, your seam carving implementation should follow these general steps:

1. Compute the energy function over the image. This should produce a numeric value for every pixel.  
   The energy function you are going to use is the most simple one:  
   For every pixel calculate the Red, Green and Blue value differences from the pixel to its 8 neighbors and sum up their values. This is essentially the gradient of the pixel.
2. Decide on a seam direction – vertical or horizontal. To simplify, you can transpose the image when a horizontal seam is chosen (the rest of the steps assume a vertical seam).
3. Compute the dynamic programming map from the second row to the bottom.
4. Find the lowest energy seam by selecting the path with the least energy:
   * Start from the top most pixel with the lowest value
   * Repeat by proceeding downwards to its lowest energy neighbor
   * Take notice to choose only one pixel per column/row
5. Remove the seam, adjusting each row to its new size which is one pixel short of the old size.

Note that these are the steps for reducing the size of the image by one row or column.

For increasing the image size, refer to the article for a slightly different procedure.

**Part 2**

Please read Section 5 (up to 5.2, not including) in the paper Improved Seam Carving for Video Retargeting:

<http://www.faculty.idc.ac.il/arik/SCWeb/vidret/vidretLowRes.pdf>

Now incorporate forward energy into your solution from part 1.

Find images that emphasize the contribution given by forward energy and run both parts on them to compare.

**Implementation**

1. Your solution should be implemented in C++ or Java.
2. For C++: Use the Image Processing library CImg: <http://cimg.sourceforge.net/index.shtml>

All you need to do in order to use it is to include the header file CImg.h, which contains all the implementation of the library.

For Java: Use the package ImageIO.

1. Your program should receive the following command line arguments:

<input image filename> <output # columns> <output # rows> <energy type> <output image filename>

Where:

<input image filename> = Full path to the input image

<output # columns> = Number of columns of the resized output image

<output # rows> = Number of rows of the resized output image

<energy type> = A boolean argument where '0' = regular energy and '1' = forward energy

<output image filename> = Full path to the output image (where your program will write the output image to)

1. C++ submission: please put all your code into one file titled seam.cpp and make sure that it compiles (and runs) in Linux using:

g++ -o seam seam.cpp -O2 -L/usr/X11R6/lib -lm -lpthread -lX11

Submit only seam.cpp inside a .zip file titled <id1>\_<id2>.zip

Java submission: submit an executable .jar file titled <id1>\_<id2>.jar

**General instructions**

1. Submission is in pairs
2. Submissions and any questions are to be directed to Noa at noafish@post.tau.ac.il