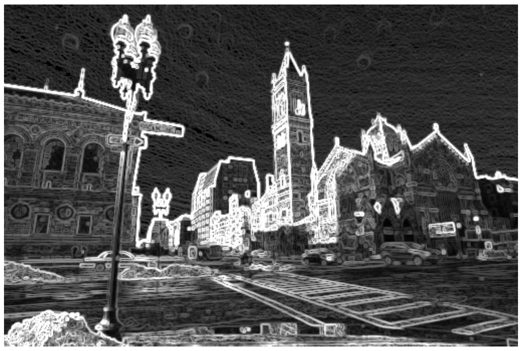
Emily Seitz
5/17/12
6.815 – a13

Image resizing using seam carving

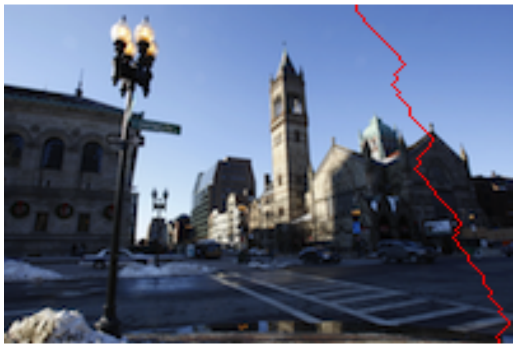
[I first attempted to implement the color2gray function, but after some frustration at failures and speed issues, I switched to seam carving.]

I followed the old pset and the lecture notes from the previous semester to implement my resizing algorithm. First I calculated how many iterations I would need, and then for each iteration I followed the following three steps:

1. *Compute the energy of the image.* I did this by convolving with the Sobel gradient and its inverse. Here is an example of my results on the Copley image.

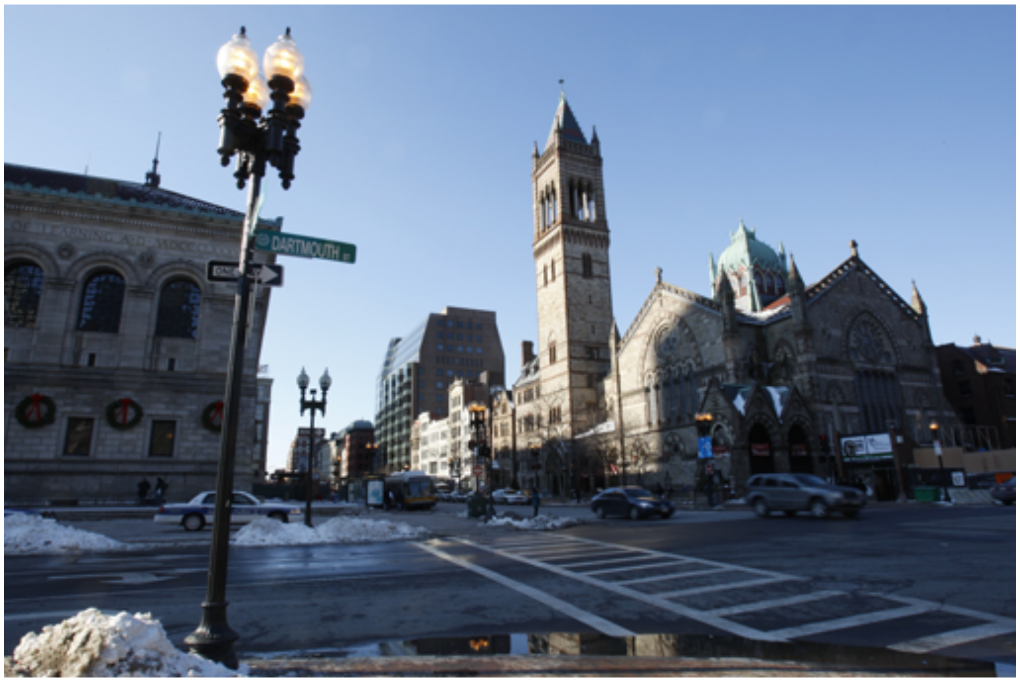


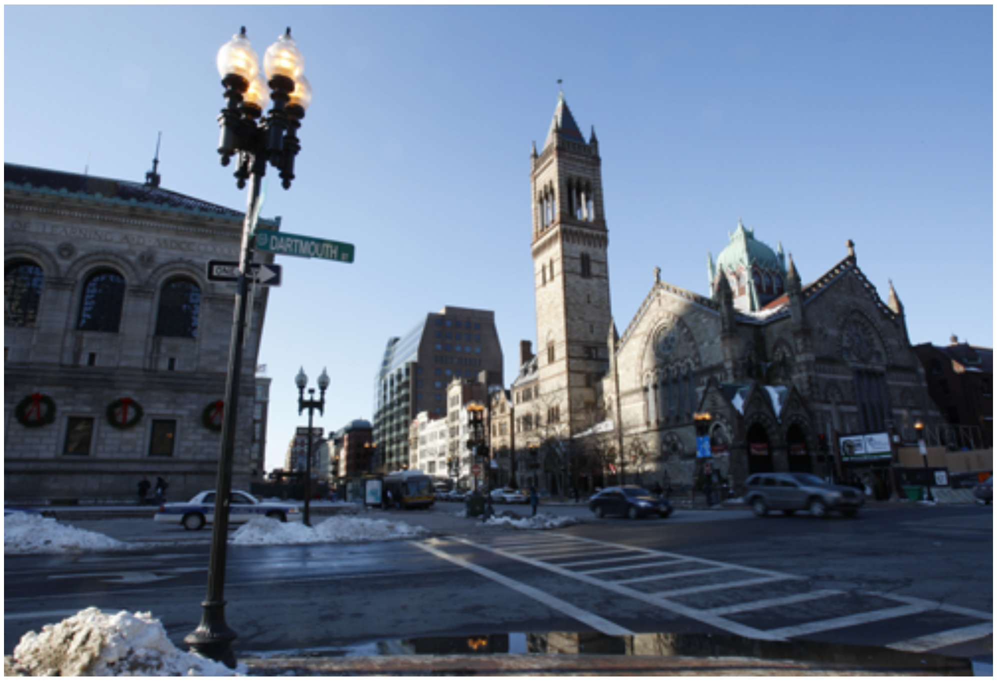
2. *Compute the minimum path through the image.* I did this by computing the minimum cost for each pixel based on its preceding pixels and its current energy value. Then I searched the last column of the image for the lowest cost. Starting there, I backtracked through the image following the least cost path and saving this path as my seam. In testing, I marked the pixels included in the seam, as seen below.



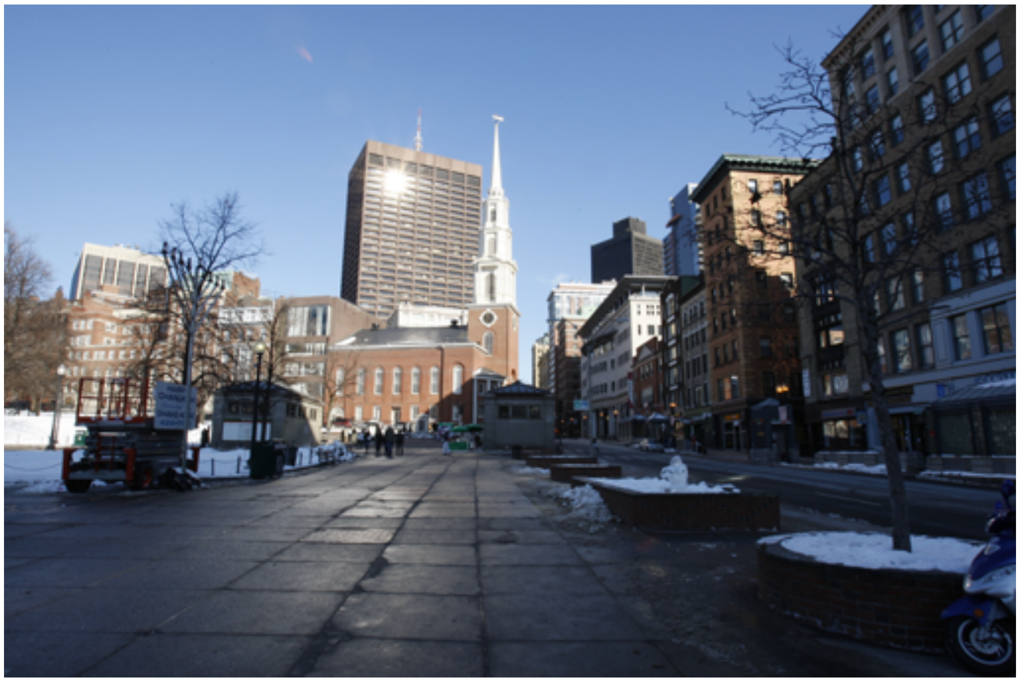
3. *Delete pixels from the seam.* To delete these pixels and shrink the image, I simply iterated through the seam list, made a copy of the row for each pixel, removed the marked one, and put this smaller row in the new image. My results are shown below for the Copley and Park Street images.

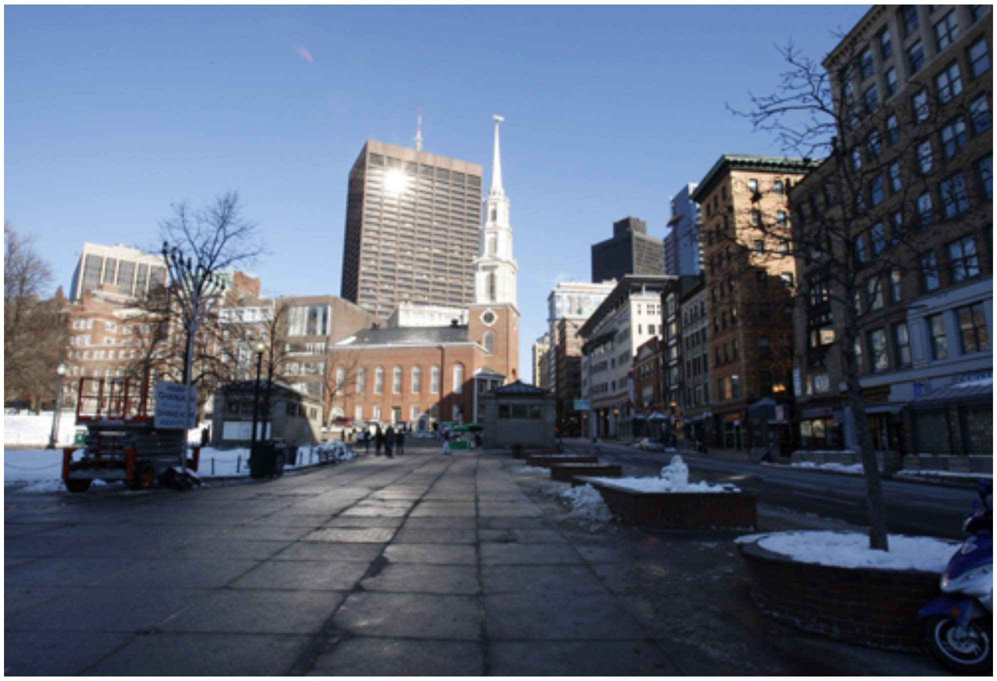
Copley





Park Street





This function was not very difficult to implement after by following the lecture notes from the previous semester. It does take a while to run, though, because it iterates through every pixel of the image once to calculate the cost, and then checks more pixels to find the path (the width of the image * 3). Because of this time issue and because I forgot to run it on my own image, I have included a very small test image that I used from the test images for color2gray.