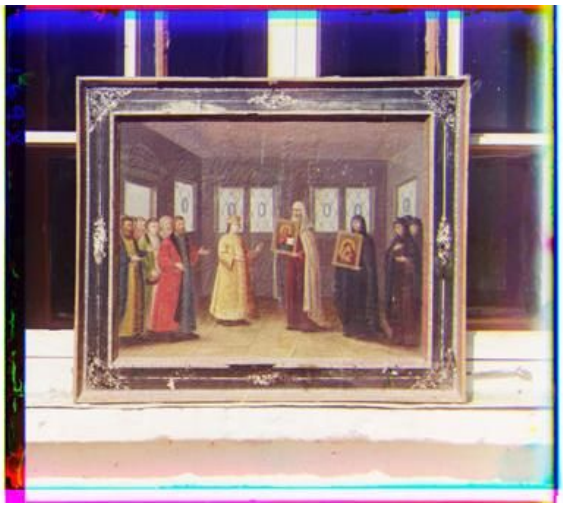
Lucas Junginger
EECS 442 W16
Homework Assignment #1



Lucas Junginger
EECS 442 W16
Homework Assignment #1

Part 1:

To align the color channels I took an exhaustive approach which produces all possible 2D shifts between the fixed channel and either of the other two. Then I converted these shifts into an array and selected the shift with the highest dot product to estimate the most similar filter positions.

The dot product is the major limitation of this algorithm because it is comparing the three color channels to each other and assuming that it must be better when they have similar intensities. This works well for aligning images with little variation in color and hard edges because this will maximize the dot product. In images lacking hard edges, such as the sitting man or the train station, the algorithm seemed to perform poorly.

I did not implement an algorithm to fix the edges but I can imagine that it would look something like using the known shifts to circularly reposition the part of each channel that goes out of bounds.

Part 2:

#	image	baseline	nn	linear	adagrad
1	balloon.jpeg	0.179239	0.045978	0.026208	0.018966
2	cat.jpg	0.099966	0.034885	0.030397	0.023248
3	ip.jpg	0.231587	0.094795	0.055499	0.042694
4	puppy.jpg	0.094093	0.018500	0.012963	0.014503
5	squirrel.jpg	0.121964	0.032543	0.025011	0.035574
6	candy.jpeg	0.206359	0.067228	0.054731	0.041370
7	house.png	0.117667	0.035348	0.031996	0.029170
8	light.png	0.097868	0.031183	0.025562	0.025853
9	sails.png	0.074946	0.023224	0.018102	0.020303
10	tree.jpeg	0.167812	0.033854	0.017796	0.027312
	average	0.139150	0.041754	0.029826	0.027899

I had a hard time with this assignment because it's my first time using MATLAB and I see that my algorithms, specifically the adaptive gradient, could be further optimized because it was not much better than my linear interpolation algorithm. Similarly, I did not have time to tackle correcting edges as I felt like going around the borders and copying pixels from inside would be quite laborious and could simply not change my results that much.

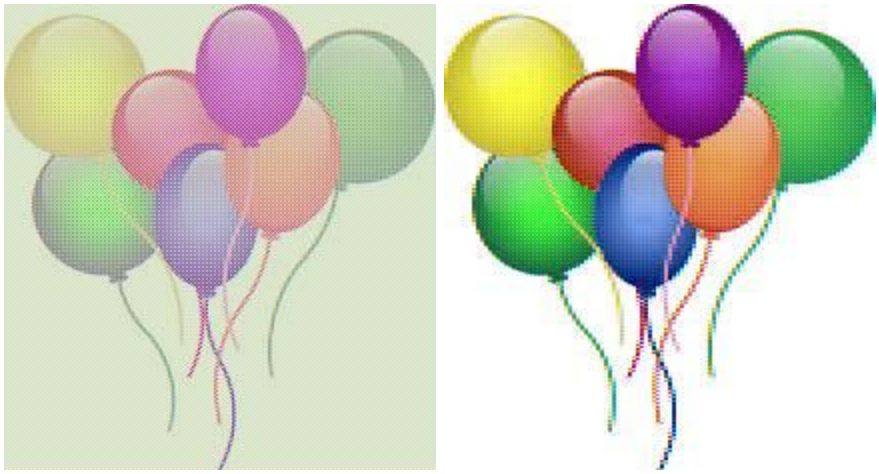
Lucas Junginger
EECS 442 W16
Homework Assignment #1

For some images like the balloon and the candy the adaptive algorithm (candy on left) performed a lot better seemingly because of the large amount of color edges and uniform regions of color. This is something the NN (right) and baseline algorithms struggle with because they are not using any queue to detect edges. The linear algorithm in this case does a good job at blending edges but it ends up distorting the colors at the edges by incorrectly averaging outside of the color edge.

There's a lot that I feel could still be done on the adaptive gradient, but again I had little time and MATLAB efficiency to implement these methods. I believe an interesting way to expand the detection of edges would be to devise an algorithm that not only looks at the difference between one pixel and the next but tries to produce a connected area that follows the edge. The color region select tool in Photoshop comes to mind.

The adaptive gradient algorithm was not as good at demosaicing images that were monotone in color and had few edges. The squirrel picture, for example, had very few defined edges and big changes in color and so I can imagine the algorithm was making most of its choices based on small differences in intensity. This seems to distort colors around light edges as can be seen all throughout this image. From left to right (baseline, NN, linear, adaptive)

Lucas Junginger
EECS 442 W16
Homework Assignment #1



Lucas Junginger
EECS 442 W16
Homework Assignment #1