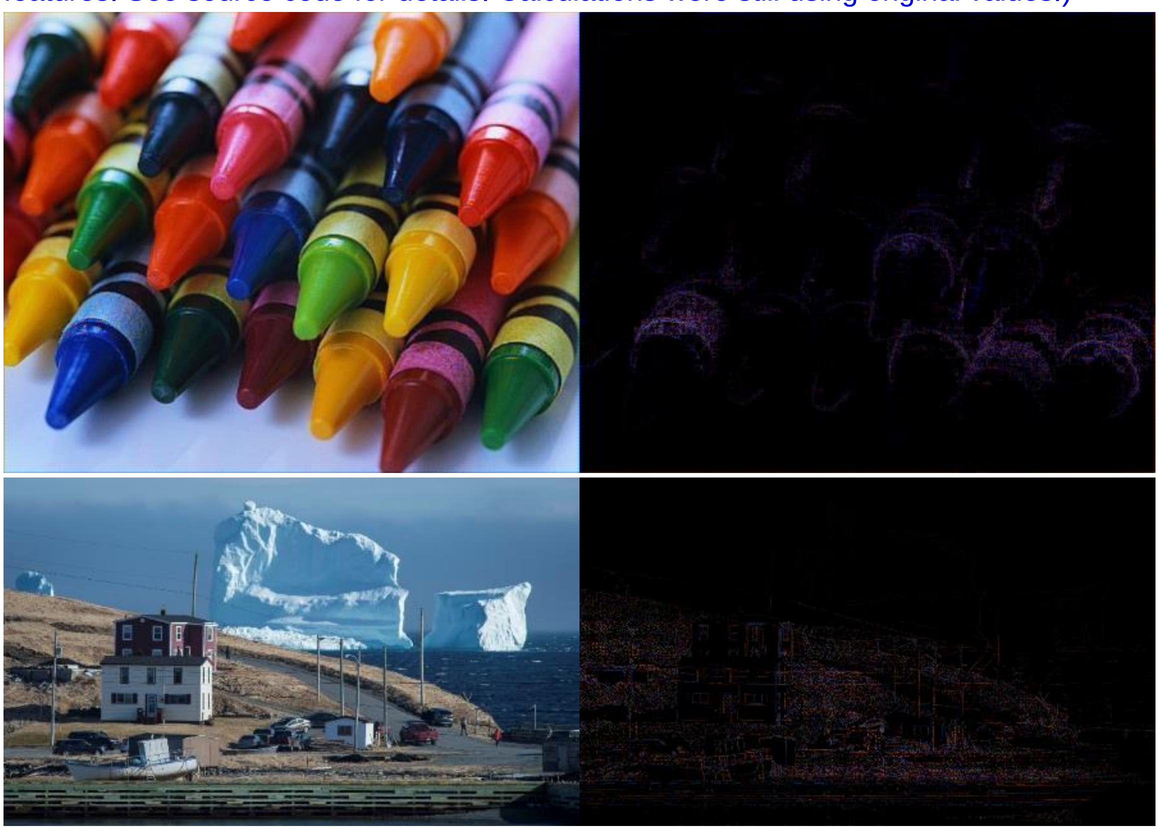
Name : Yen-Ting Liu (ytliu2)

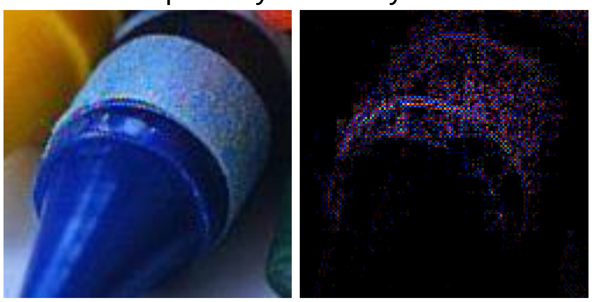
Part-1 : Linear Interpolation

1) Insert your linear interpolated test image(hope.jpg) here:

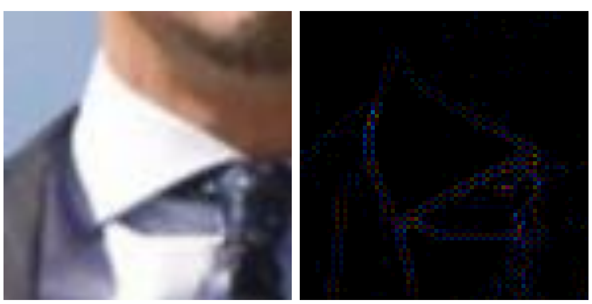
2) Display the map/plot of all the 3 training images here:
(All error maps in this assignment were histogram equalized by contrast limited adaptive histogram equalization (CLAHE), with clipping limit capped at 0.5 to enhance error features. See source code for details. Calculations were still using original values.)



3) Post close-up of any artifacts you came across.


X0=(41,281); (W,H)=(124,124)


X0=(781,733); (W,H)=(126,126)


X0=(540,209); (W,H)=(73,73)

All these close-up views can see color shifts, as if different color channels are slightly offset by a pixel or two. When doing simple linear interpolation, we cannot possibly know if the inferred pixel is actually a sharp color edge (transition). This leads to color bleeding to unwanted location, manifests as color shifts we observed.

4) Average_per_pixel error and Max_pixel_error for each of 3 training images :

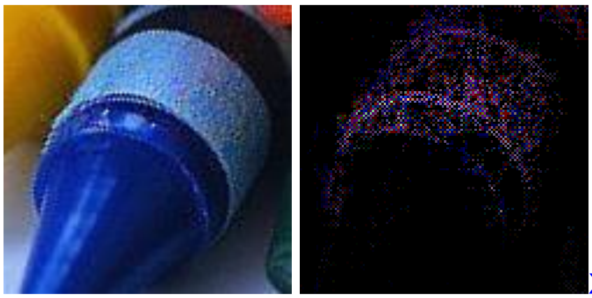
Image	**Average_per_pixel_error**	**Max_pixel_error**
Crayons	70.92	37636
Iceberg	38.32	23409
Tony	19.40	26569

Part-2 : Freeman Method

5) Insert your Freeman Method test image(hope.jpg) here:

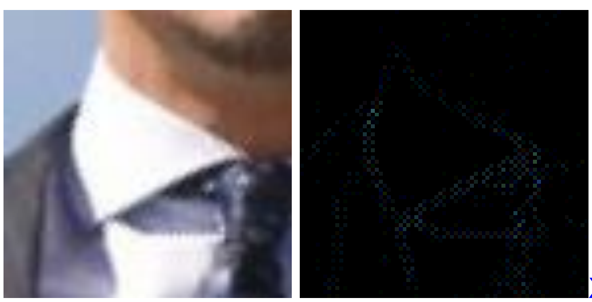
6) Display the map/plot of all the 3 training images here:
(All Freeman method reconstructions below use median filter with kernel size (3, 3).)

7) Post close-up of any artifacts you came across.



X0=(41,281); (W,H)=(124,124)

X0=(781,733); (W,H)=(126,126)



X0=(540,209); (W,H)=(73,73)

Color shifts from linear interpolation are resolved in Freeman method, however, these interpolations still do not factor in edges, therefore, when a sharp edge is interpolated, it is interleaved with actual-inferred pixels, forming zipper-like artefacts. This is especially profound around thin sharp lines like those cables in ` Iceberg`.

Since color shifts are corrected but still have positional error, error map (Freeman) would show most errors as white instead of different interleaved color (interpolation). This is clearly visible in `Tony` close up error map.

8) Average_per_pixel error and Max_pixel_error for each of 3 training images :

Image	Average_per_pixel_error	Max_pixel_error
Crayons	53.05	28900
Iceberg	25.20	23409
Tony	14.30	26569

Part-3 : Images of your choice

1) Post 2 images of your choice here and the corresponding error maps of your outputs with the Freeman method.
   (Following colored images are reconstructed results.)

2) Any image that breaks the method and why do you think so?
   Based on the observation in `Tony`, demosaicing tends to fail around sharp edges. My assumption is that sharp edges are lost when doing mosaicing, doing linear interpolation is essentially a low pass filter on the image, therefore, color restoration around these parts will fail.

   Freeman method tries to avoid restoration error by using green channel (which has twice the number of pixels), therefore, smooth regions should have superior performance.

However, long, sharp features are not fixable this, this is visible in `Iceberg` where the cable still has zippering artifact.

Therefore, I choose these two images from the SIPI database

- baboon (4.2.03) has lots of sharp color transitions.
- airplane (4.2.05) has uniform cloud background and lots of ridges at its bottom.

Results agree with the observation, where uniform background yields near perfect reconstruction, where rough edges (sharp color transitions) failed despite additional efforts to fix them.

Part-4 : Bonus

Post any extra credit details/images/references used here.

After scrolling through different demosaicing algorithms, I decided to implement Malvar-He-Cutler demosaicing, like Freeman, they both improve the output by correcting the bilinear result, allowing me to reuse code snippets.

The linear interpolation method disregard relationships across RGB channels, which results in color misalignment and zipper artifacts. Freeman method fixes this through median filter, but this does not factor in the underlying textures. Therefore, we would assume color displacements are eliminated but sharp edges would still show zipper artifacts, which can be seen in the close-up image below.

If we want to find *where* the color should be, we can use gradient of these colors as an estimator. Malvar-He-Cutler uses Laplacian of different colors, and use these to augment interpolated pixels, i.e. use Laplacian of red and blue to estimate interpolated green pixels on red and green position.

For green channel, it can be corrected by

$$\begin{cases} \hat{G}_R = \hat{G}^{bl} + \alpha \nabla_R \\ \hat{G}_B = \hat{G}^{bl} + \alpha \nabla_B \end{cases}$$

where $\hat{G}_R$ is green pixel interpolated on red pixel, $\hat{G}^{bl}$ is the bilinear interpolated green pixel, ∇_R is the Laplacian of red channel.

MHC determines a set of weights, α, β, and γ for the Laplacian corrections for red, green, and blue Laplacians. These are obtained by minimized the MSE over the Kodak image suite. Here, I am going to directly use their result

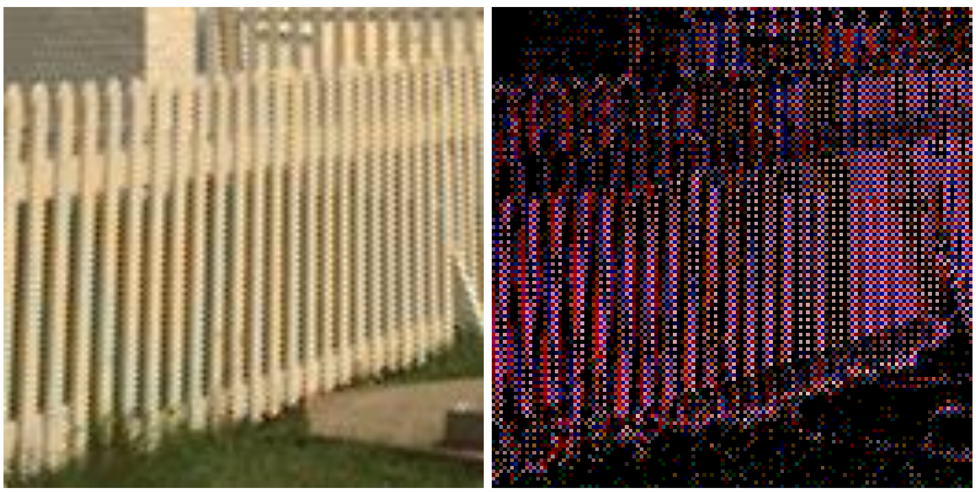
$$\alpha = \frac{1}{2}, \beta = \frac{5}{8}, \gamma = \frac{3}{4}$$

The close-up result clearly indicates these Laplacian corrections term can effectively restore these zipper artifacts. However, locations with much denser fences still exhibit artifacts like other demosaicing methods since the information loss is simply too significant.

The following are close up view at X0=(204, 443), (W,H)=(130,130) of different demosaicing methods implemented in this assignment. Bonus error map shows a hue in the background due to the fact that Laplacian correction spans across the entire image, however, average pixel error is lower.

linear interpolation

Freeman



MHC

Image	Average_per_pixel_error	Max_pixel_error
Linear	107.04	18225
Freeman	67.51	17689
MHC	49.81	7744

Kodak image suite, image 19, demosaiced.