

## Results

In order to test the performance of the two algorithms, we ran a set of images through the Matlab scripts for our implementations of the recent demosaicking methods as well as a simple bilinear interpolation method and compared the results. The Bayer color filter array was first applied to the images before running them through our demosaicking algorithms. The images we used were chosen as they contained areas of high spatial frequency, which served to stress the algorithms. As well as visually comparing the resulting images, we calculated mean-squared error (MSE) and peak signal to noise ratio (PSNR) for each image as an objective measure of performance. These values are presented in Tables 1 and 2 at the end of this section. Table 3 shows the average computation time for the cubic spline and color difference demosaicking methods.

### *Picket Fence from Lighthouse Image*

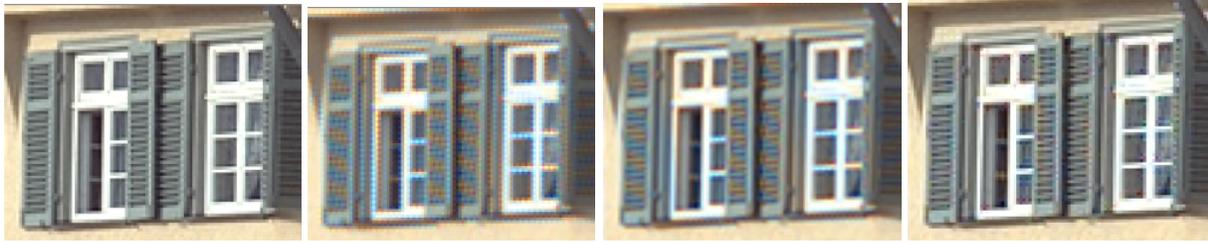
The first set of images, shown in Figure 16, depict the picket fence region of the lighthouse image. The high number of vertical edges in the image caused problems for the bilinear and cubic spline demosaicked images, which suffer from a great deal of color artifacts. While the cubic spline method results in a generally smooth image with color artifacts, the bilinear method causes grainy block artifacts as well as the color issues due to its lack of adaptive edge checking. The color difference method yields a relatively clean result with a few discolored pixels visible.



**Figure 16.** From left to right: original, bilinear, cubic spline, and color difference images for the picket fence part of the lighthouse image.

### *Windows from Buildings Image*

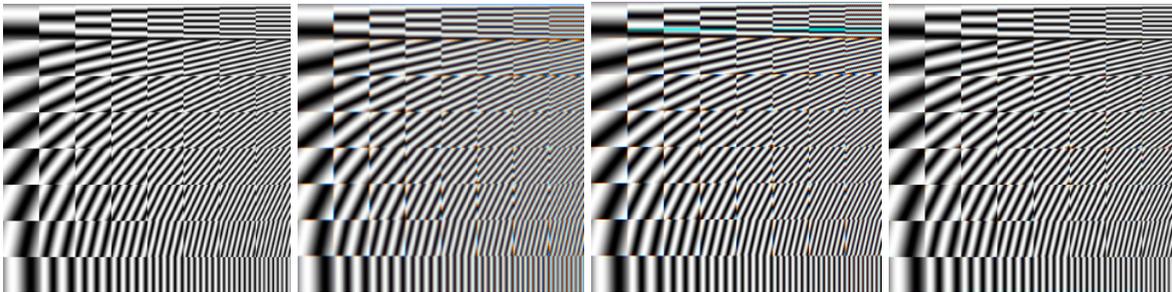
Figure 17 presents demosaicked images of the windows from the buildings image. The closely spaced horizontal lines on the window shutters caused a great deal of graininess and color artifacting when using the bilinear method, while the cubic spline method yielded a cleaner image with discoloration along the window shutters. The color difference method performed quite well, causing few visible artifacts.



**Figure 17.** From left to right: original, bilinear, cubic spline, and color difference images for the windows in the buildings image.

### ***Spatial Frequency Image***

The spatial frequency images in Figure 18 consist of a series of black and white alternating lines that curve around, testing each method’s efficiency in dealing with high spatial frequencies in vertical, horizontal, and diagonal directions. As with the other images, the bilinear method performed the most poorly, with discoloration clearly visible throughout the higher frequency regions. The cubic spline method performed better, with less discoloration overall, but still noticeable. The color difference method eliminated almost all discoloration within the individual cells of the image, but still suffered from some artifacts between cell edges.



**Figure 18.** From left to right: original, bilinear, cubic spline, and color difference images for the spatial frequency image.

### ***Mean-Squared Error, Peak Signal to Noise Ratio, and Computation Time***

Along with the qualitative results described above, we measured the quantitative efficiency with which the demosaicking methods reconstructed the original images. Tables 1 and 2 tabulate the MSE and PSNR for a series of images including the ones described above.

Mean-Square Error		Linear Interpolation		Bilinear Interpolation	Cubic Spline Interpolation			Color Difference Space Interpolation		
		Horizontal	Vertical		No Median Filtering of Orientation Map	Median Filter Width=5	Median Filter Width=7	Adaptive	Horizontal	Vertical
		MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE
Sailboats	R	-	-	64.40		24.64*		4.75	17.05	14.75
	G	38.62	24.66	20.04	5.98	4.67	4.63	3.37	10.68	9.29
	B	-	-	47.71		24.60*		5.53	17.95	15.93
Buildings	R	-	-	383.20		208.42*		24.26	143.34	87.49
	G	234.35	165.94	121.49	38.31	28.12	29.52	20.67	76.57	52.68
	B	-	-	347.87		191.59*		30.29	152.60	92.64
Lighthouse1	R	-	-	129.06		66.75*		8.98	46.80	36.16
	G	79.00	62.02	43.35	12.60	9.63	9.58	7.27	28.61	23.93
	B	-	-	125.31		66.54*		10.19	46.53	40.96
Lighthouse2	R	-	-	124.96		75.26		12.29	22.25	51.80
	G	43.00	97.11	47.54	18.31	14.76	14.76	10.96	16.54	33.13
	B	-	-	114.55		69.47*		16.77	27.27	58.07
Freq Orientation	R	-	-	1381.50		679.85*		32.93	122.09	142.77
	G	616.65	645.50	411.90	45.17	61.07	61.07	36.65	83.83	90.18
	B	-	-	1149.20		451.67*		29.56	126.73	138.95

\* Cubic-Spline Interpolation on red and blue channels does not use adaptive direction selection.

Table 1. Mean-squared error data for linear, bilinear, cubic spline, and color difference interpolation methods.

Peak Signal to Noise Ratio		Linear Interpolation		Bilinear Interpolation	Cubic Spline Interpolation			Color Difference Space Interpolation		
		Horizontal	Vertical		No Median Filtering of Orientation Map	Median Filter Width=5	Median Filter Width=7	Adaptive	Horizontal	Vertical
		PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
Sailboats	R	-	-	30.04		34.21*		41.36	35.81	36.44
	G	32.26	34.21	35.11	40.36	41.44	41.48	42.86	37.85	38.45
	B	-	-	31.34		34.22*		40.71	35.59	36.11
Buildings	R	-	-	22.30		24.94*		34.28	26.57	28.71
	G	24.43	25.93	27.29	32.30	33.64	33.43	34.98	29.29	30.91
	B	-	-	22.72		25.31*		33.32	26.30	28.46
Lighthouse1	R	-	-	27.02		29.89*		38.60	31.43	32.55
	G	29.15	30.21	31.76	37.12	38.30	38.31	39.52	33.57	34.34
	B	-	-	27.15		29.90*		38.05	31.45	32.01
Lighthouse2	R	-	-	27.16		29.37*		37.24	34.66	30.99
	G	31.80	28.26	31.36	35.50	36.44	36.44	37.73	35.95	32.93
	B	-	-	27.54		29.71*		35.89	33.77	30.49
Freq Orientation	R	-	-	16.73		19.81*		32.96	27.26	26.58
	G	20.23	20.03	21.98	31.58	30.27	30.27	32.49	28.90	28.58
	B	-	-	17.53		21.58*		33.42	27.10	26.70

\* Cubic-Spline Interpolation on red and blue channels does not use adaptive direction selection.

Table 2. Peak signal to noise ration data for linear, bilinear, cubic spline, and color difference interpolation methods.

As the quantitative results indicate, the color difference algorithm outperforms the bilinear and cubic spline approaches in both MSE and PSNR. For the color difference algorithm, we tweaked the Matlab script to turn on and off the adaptive edge-checking. Without edge detection, the color difference algorithm performs much more poorly, although it still manages to beat the linear and bilinear algorithms. The cubic spline interpolation was also run with some adjustments, namely changing the median filter width. Although in most cases the changes in results were minimal, completely doing away with the filter for the spatial frequency orientation image improved the results a noticeable amount, due to the fact that important edges were being removed during the filtering stage. On average, the color difference method outperformed the

cubic spline method by 1.94 dB, the bilinear approximation method by 11.05 dB, and the linear approximation scheme by 9.86 dB.

Another metric we used to compare the cubic spline and color difference demosaicking methods was computation time. Since most demosaicking algorithms are implemented in digital camera hardware, low computational intensity is favorable as it allows for images to be captured and processed more quickly. Table 3 presents the computation time of our algorithms.

MATLAB Computation Time (seconds)		
Cubic Spline Interpolation		Color Difference
	no median filtering	
<b>54.577</b>	<b>13.73</b>	<b>1.119</b>

**Table 3.** Matlab computation time for cubic spline and color difference algorithms.

The color difference algorithm runs much more quickly than the cubic spline interpolation. It is also computationally less intense, as it requires only addition, subtraction, and division by powers of two—all easily implemented in hardware. As the results show, however, the median filtering stage of the cubic spline method consumes most of the computation time and eliminating it decreases the run time by over 70%. This is important to note since in some images, such as the frequency orientation test image, removing the median filtering step actually increases the image quality.