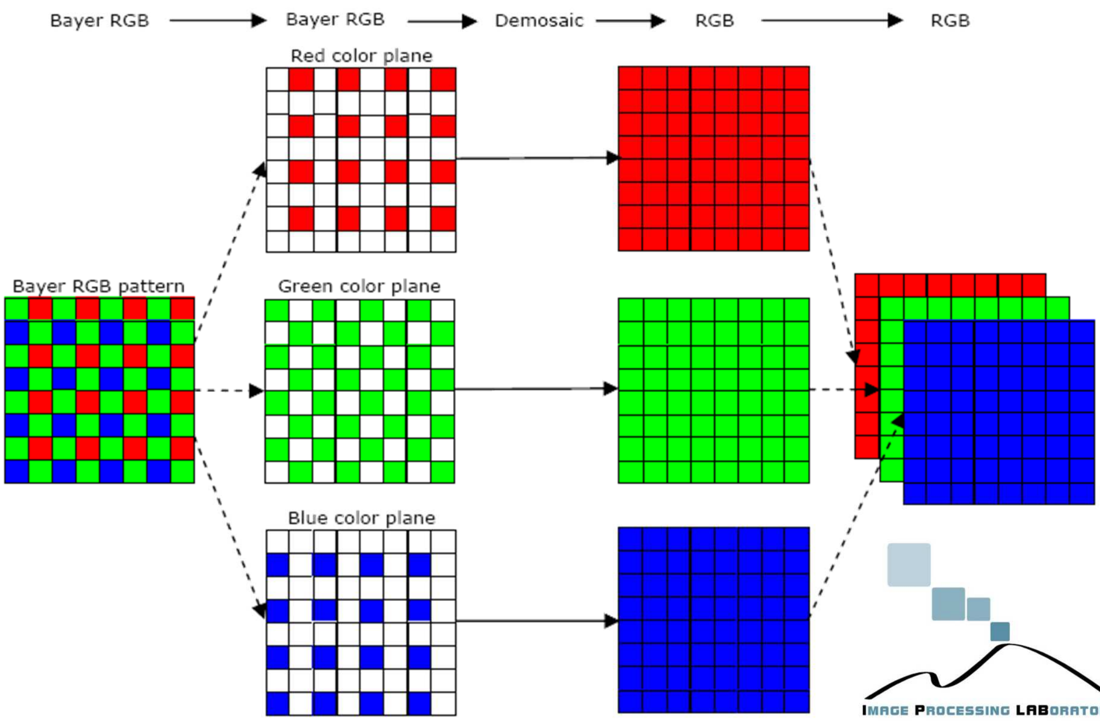
Demosaicing

Multimedia

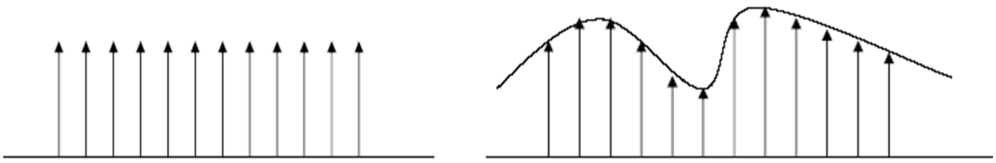
Demosaicing



Ideal Interpolation

Shannon Sampling Theorem:

When a “train of impulse” *comb(x)* is multiplied by *f(x)*, it gives us a “sampled version” of *f(x)*



comb(x)f(x), in frequency domain, becomes convolution.
Convolving with an impulse and shifting *comb(s)* * *F(s)* is replicating the spectrum *F(s)* at the different impulse locations.

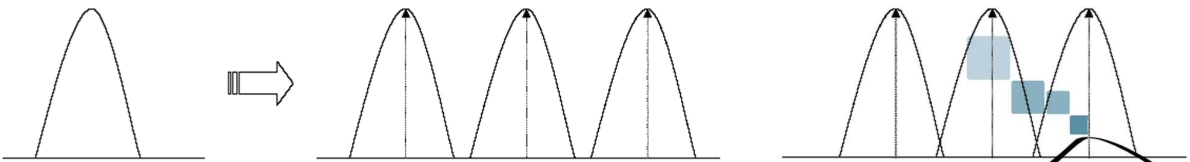


IMAGE PROCESSING LABORATORY

Spatial aliasing in the form of a Moiré pattern

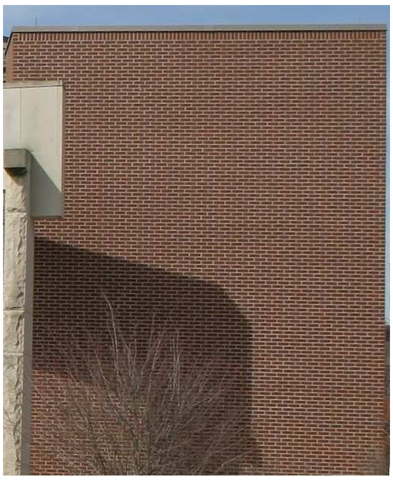
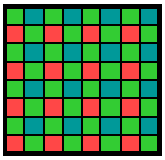


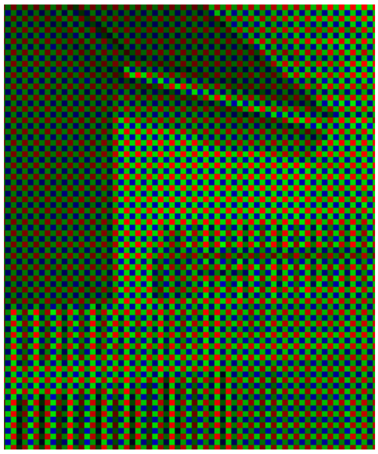
IMAGE PROCESSING LABORATORY

CFA



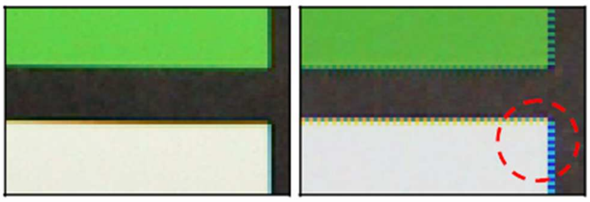
Bayer CFA

CFA image

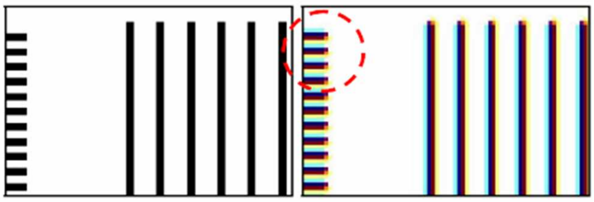


Color artifacts

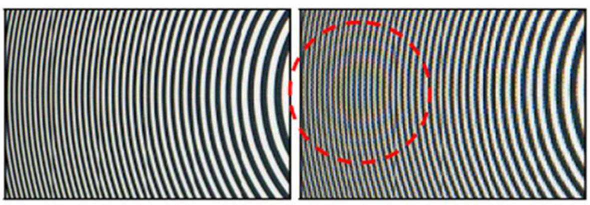
- Common artifacts
  - Most visual artifacts appear at edges and areas of high frequency



Zipper effect



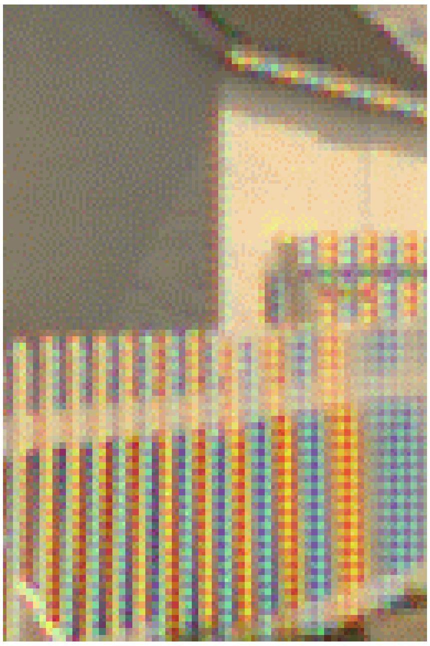
False color



Aliasing

Blurring

Zipper Effect



Bilinear interpolation

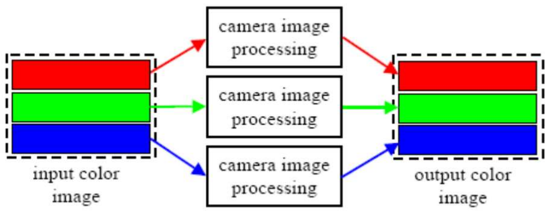
Weighted interpolation

IMAGE PROCESSING LABORATORY

Processing

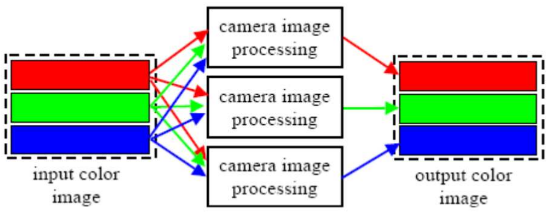
Component-wise processing

- each color plane processed separately
- omission of the spectral information results in color shifts and artifacts



Spectral model based processing

- essential spectral information utilized during processing
- computationally very efficient - most widely used in camera image processing



Vector processing

- image pixels are processed as vectors
- computationally expensive

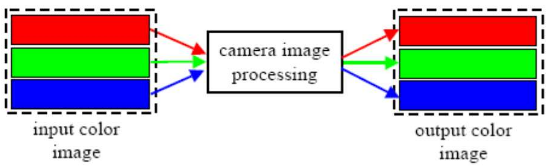
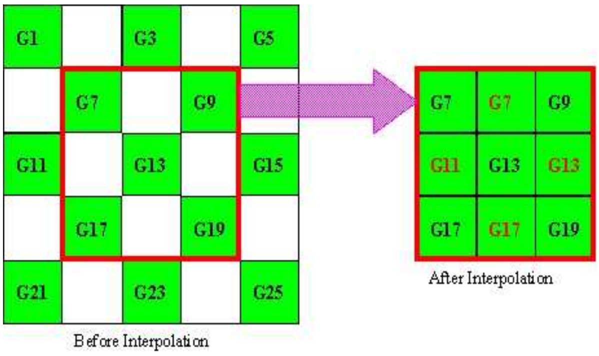


IMAGE PROCESSING LABORATORY

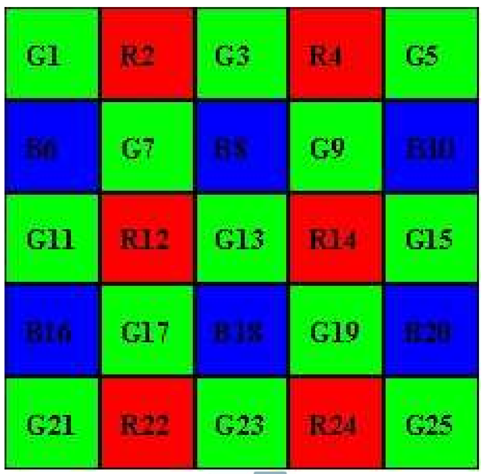
Color Interpolation - Nearest Neighbor Replication

- Each interpolated output pixel is assigned the value of the nearest pixel in the input image
- The nearest neighbor can be any one of the upper, lower, left and right pixels
- For example, for a 3x3 block in green plane, we assume the left neighboring pixel value is used to fill the missing ones



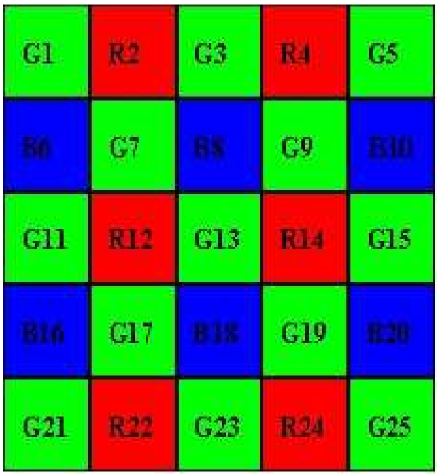
Color Interpolation - Bilinear Interpolation

- Interpolation of green pixels
  - The average of the upper, lower, left and right pixel values is assigned as the G value of the interpolated pixel
  - G8 = (G3+G7+G9+G13) / 4
- Interpolation of a red/blue pixel at a green position
  - The average of two adjacent pixel values in corresponding color is assigned to the interpolated pixel.
  - B7 = (B6+B8) / 2
  - R7 = (R2+R12) / 2



Color Interpolation - Bilinear Interpolation

- Interpolation of a red/blue pixel at a blue/red position
  - The average of four adjacent diagonal pixel values is assigned to the interpolated pixel
  - R8 = (R2+R4+R12+R14) / 4
  - B12 = (B6+B8+B16+B18) / 4



Bilinear

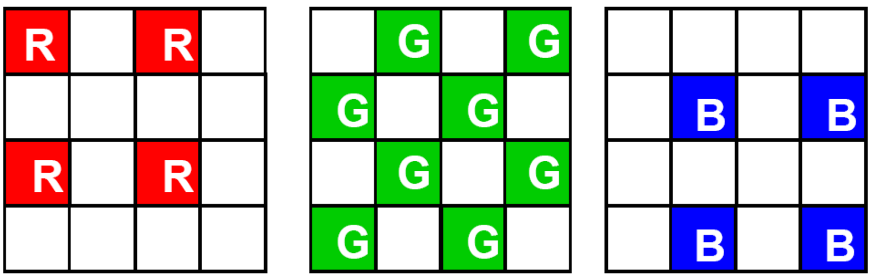
- Simple realization with ***3 by 3*** filter kernels

$$R_F(n_1,n_2) = \mathbf{F}_R \otimes M_R(n_1,n_2)$$

$$G_F(n_1,n_2) = \mathbf{F}_G \otimes M_G(n_1,n_2)$$

$$B_F(n_1,n_2) = \mathbf{F}_B \otimes M_B(n_1,n_2)$$

$$\mathbf{F}_R = \mathbf{F}_B = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \Big/ 4 \quad , \quad \mathbf{F}_G = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \Big/ 4$$

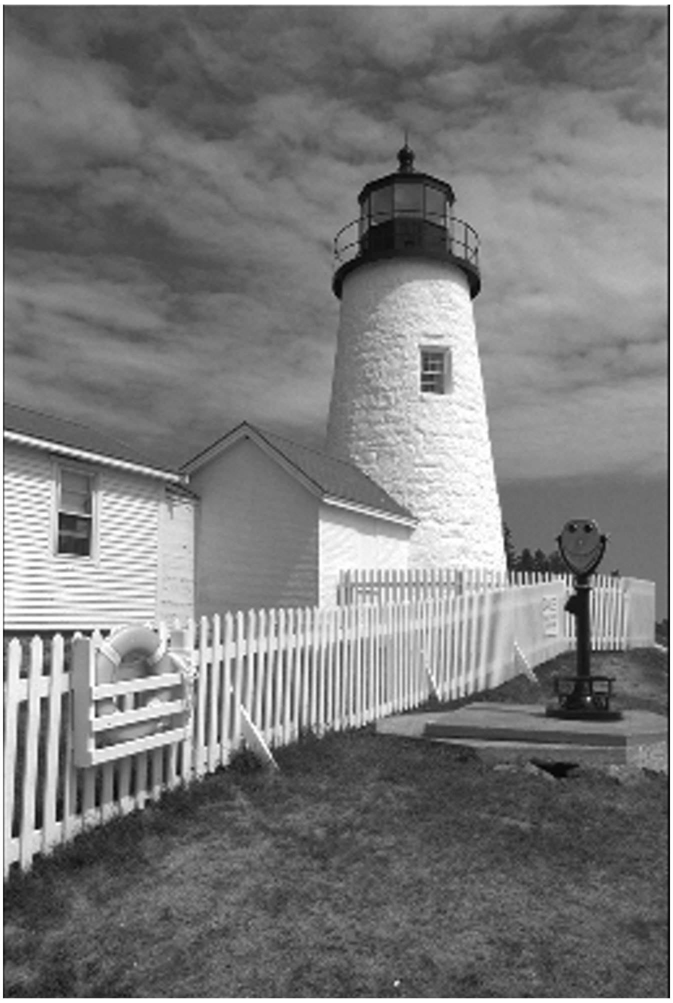


$M_R(n_1,n_2)$ $M_G(n_1,n_2)$ $M_B(n_1,n_2)$

NG LABORATORY

Lighthouse
original

Lighthouse
Interpolated color image



Lighthouse
red interpolated
with bicubic
interpolator

IMAGE PROCESSING LABORATORY

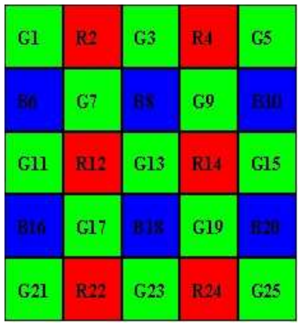
Lighthouse
prefiltered red interpolated
with bilinear interpolator

IMAGE PROCESSING LABORATORY

EDGE-DIRECTED INTERPOLATION

Interpolation of green pixels :

First, define two gradients, one in horizontal direction, the other in vertical direction, for each blue/red position. For instance, consider B8 : define two gradients as



$$\Delta H = |G7 - G9| \text{ and } \Delta V = |G3 - G13|$$

Define some threshold value T

The algorithm then can be described as:

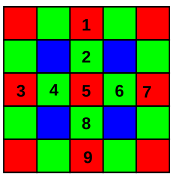
```
If ΔH < T AND ΔV > T,
        G8 = (G7 + G9) / 2;
Else if ΔH > T AND ΔV < T,
        G8 = (G3 + G13) / 2;
Else
        G8 = (G3 + G7 + G9 + G13) / 4
End
```

The choice of T depends on the images and can have different optimum values from different neighborhoods. A particular choice of T is $T = (\Delta H + \Delta V)/2$

```
If ΔH < ΔV ,
        G8 = (G7 + G9) / 2;
Else if ΔH > ΔV,
        G8 = (G3 + G13) / 2;
Else
        G8 = (G3 + G7 + G9 + G13) / 4
End
```

Demosaicking Approaches

The edge-directed interpolation idea can be modified by using larger regions (around the pixel in question) with more complex predictors and by exploiting the texture similarity in different color channels.



Edge-directed interpolation

1. Calculate horizontal gradient ΔH = | (R3 + R7)/2 – R5 |
2. Calculate vertical gradient ΔV = | (R1 + R9)/2 – R5 |
3. If ΔH > ΔV,
   G5 = (G2 + G8)/2
   Else if ΔH < ΔV,
   G5 = (G4 + G6)/2
   Else
   G5 = (G2 + G8 + G4 + G6)/4

Edge-Directed Interpolation

- **Two-step interpolation**
- **Step 1**: Edge-directed interpolation for the *G* channel
  - Interpolation direction is chosen to avoid interpolating across edges.
  - Interpolation is performed along edges.
  - To determine a preferred interpolation direction, second-order derivatives of *B* or *R* values are used.
- **Step 2**: Bi-linear interpolation of color differences ***R-G*** or ***B-G***
  - To utilize inter-channel correlations according to the color-difference rule

Edge-Directed Interpolation

Step 1: Edge-directed interpolation for the *G* channel

For interpolating a missing *G* value at the *B* pixel, *B44,*

(1) Compute magnitudes of horizontal and vertical second-order spatial derivatives of measured *B* values.

$$\alpha = |(B_{42} + B_{46})/2 - B_{44}|$$
$$\beta = |(B_{24} + B_{64})/2 - B_{44}|$$

(2) Classify the direction and existence of an edge around the *B* pixel, *B44,* into three cases.

(3) Select a proper directional averaging operation for the interpolation.

$$G_{44} = \begin{cases} \dfrac{G_{43} + G_{45}}{2} & \text{if } \alpha < \beta \\ \dfrac{G_{34} + G_{54}}{2} & \text{if } \alpha > \beta \\ \dfrac{G_{43} + G_{45} + G_{34} + G_{54}}{4} & \text{if } \alpha = \beta \end{cases}$$

R_{11}	G_{12}	R_{13}	G_{14}	R_{15}	G_{16}
G_{21}	B_{22}	G_{23}	B_{24}	G_{25}	B_{16}
R_{31}	G_{32}	R_{33}	G_{34}	R_{35}	G_{36}
G_{41}	B_{42}	G_{43}	B_{44}	G_{45}	B_{46}
R_{51}	G_{52}	R_{53}	G_{54}	R_{55}	G_{56}
G_{61}	B_{62}	G_{63}	B_{64}	G_{65}	B_{66}

Edge-Directed Interpolation

Step 2: ***R* and *B* channel interpolation**

(1) Missing *R* values are given by the bi-linear interpolation of color differences *R*-*G*.

Missing *R* values, *R34*, *R43*, *R44*, are given by

$$R_{34} = \frac{R_{33} + R_{35}}{2} - \frac{G_{35} - 2 \cdot G_{34} + G_{33}}{2}$$

$$R_{43} = \frac{R_{33} + R_{53}}{2} - \frac{G_{53} - 2 \cdot G_{43} + G_{33}}{2}$$

$$R_{44} = \frac{R_{33} + R_{35} + R_{53} + R_{55}}{4} - \frac{G_{33} + G_{35} + G_{53} + G_{55} - 4 \cdot G_{44}}{4}$$

R_{11}	G_{12}	R_{13}	G_{14}	R_{15}	G_{16}
G_{21}	B_{22}	G_{23}	B_{24}	G_{25}	B_{16}
R_{31}	G_{32}	R_{33}	G_{34}	R_{35}	G_{36}
G_{41}	B_{42}	G_{43}	B_{44}	G_{45}	B_{46}
R_{51}	G_{52}	R_{53}	G_{54}	R_{55}	G_{56}
G_{61}	B_{62}	G_{63}	B_{64}	G_{65}	B_{66}

(2) Missing *B* values are given by the bi-linear interpolation of color differences *B*-*G*.

Missing *B* values, *B34*, *B43*, *B33*, are interpolated in the similar manner.

IMAGE PROCESSING LABORATORY

CONSTANT-HUE-BASED INTERPOLATION

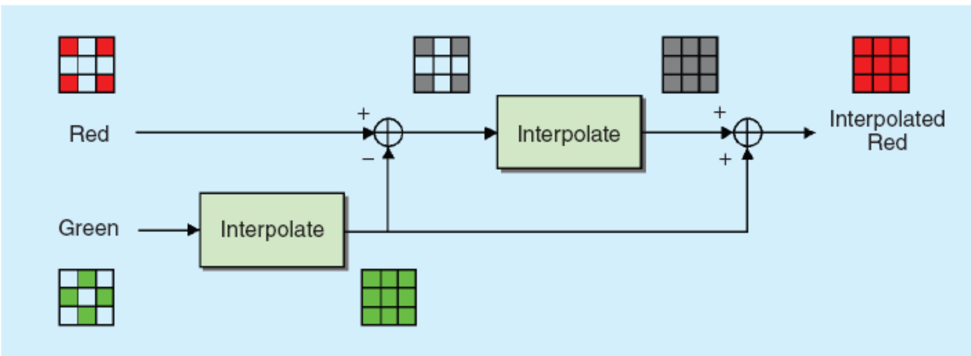


IMAGE PROCESSING LABORATORY

Adaptive interpolation

Using Laplacian For Enhancement: Use the second-order gradients of red/blue channels to enhance green channel.

Step 1: Edge-directed interpolation for the *G* channel

For interpolating a missing *G* value at the *R* pixel, *R5*,

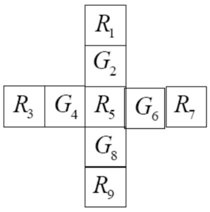
(1) Compute magnitude sums of first-order derivatives and second-order derivatives.

$$\alpha = |R_3 - 2\cdot R_5 + R_7| + |G_6 - G_4|$$
$$\beta = |R_1 - 2\cdot R_5 + R_9| + |G_8 - G_2|$$

(2) Classify the direction and existence of an edge around a red center pixel R_5 into three cases.

(3) Select a proper directional interpolation scheme.

$$G_5 = \frac{G_4 + G_6}{2} - \frac{R_3 - 2\cdot R_5 + R_7}{2}, \quad if\ \alpha < \beta$$
$$G_5 = \frac{G_2 + G_8}{2} - \frac{R_1 - 2\cdot R_5 + R_9}{2}, \quad if\ \alpha > \beta$$
$$G_5 = \frac{G_2 + G_4 + G_6 + G_8}{4} - \frac{R_1 + R_3 - 4\cdot R_5 + R_7 + R_9}{4}, \quad if\ \alpha = \beta$$



Adaptive interpolation

Step 2: Edge-directed interpolation for the *R* and *B* channels

In the case of the R-channel interpolation

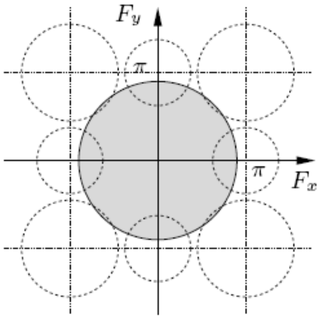
$$R_2 = \frac{R_1 + R_3}{2} - \frac{G_1 - 2G_2 + G_3}{2}$$
$$R_4 = \frac{R_1 + R_7}{2} - \frac{G_1 - 2G_4 + G_7}{2}$$

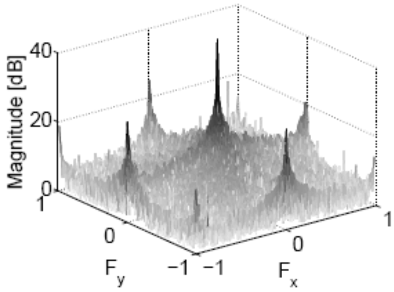
R_1	G_2	R_3
G_4	B_5	G_6
R_7	G_8	R_9

$$R_5 = \frac{R_3 + R_7}{2} - \frac{G_3 - 2\cdot G_5 + G_7}{2}, \quad if\ \alpha < \beta$$
$$R_5 = \frac{R_1 + R_9}{2} - \frac{G_1 - 2\cdot G_5 + G_9}{2}, \quad if\ \alpha > \beta$$
$$R_5 = \frac{R_1 + R_3 + R_7 + R_9}{4} - \frac{G_1 + G_3 - 4\cdot G_5 + G_7 + G_9}{4}, \quad if\ \alpha = \beta$$
$$\alpha = abs(G_3 - 2\cdot G_5 + G_7) + abs(R_7 - R_3)$$
$$\beta = abs(G_1 - 2\cdot G_5 + G_9) + abs(R_9 - R_1)$$

Frequency domain approaches (1)

$$
\begin{aligned}
R_s(\omega_1,\omega_2) &= \frac{1}{4}[R(\omega_1,\omega_2) - R(\omega_1-\pi,\omega_2-\pi) + R(\omega_1-\pi,\omega_2) - R(\omega_1,\omega_2-\pi)]\\
G_s(\omega_1,\omega_2) &= \frac{1}{2}[G(\omega_1,\omega_2) + G(\omega_1-\pi,\omega_2-\pi)]\\
B_s(\omega_1,\omega_2) &= \frac{1}{4}[B(\omega_1,\omega_2) - B(\omega_1-\pi,\omega_2-\pi) + B(\omega_1-\pi,\omega_2) - B(\omega_1,\omega_2-\pi)]
\end{aligned}
$$



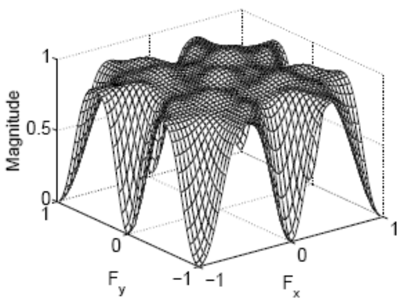


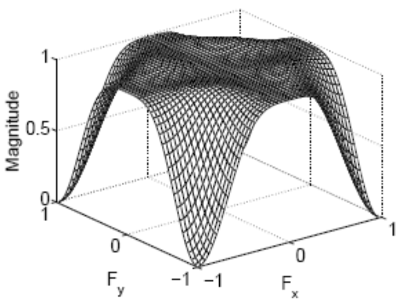
$$I_s(\omega_1,\omega_2) = L(\omega_1,\omega_2) + C_1(\omega_1-\pi,\omega_2-\pi) + C_2(\omega_1-\pi,\omega_2) - C_2(\omega_1,\omega_2-\pi)$$

Frequency domain approaches (2)

$$I_s(\omega_1,\omega_2) = L(\omega_1,\omega_2) + C_1(\omega_1-\pi,\omega_2-\pi) + C_2(\omega_1-\pi,\omega_2) - C_2(\omega_1,\omega_2-\pi)$$

$$
\begin{aligned}
L(\omega_1,\omega_2) &= \frac{R(\omega_1,\omega_2) + 2G(\omega_1,\omega_2) + B(\omega_1,\omega_2)}{4}\\
C_1(\omega_1,\omega_2) &= \frac{-R(\omega_1,\omega_2) + 2G(\omega_1,\omega_2) - B(\omega_1,\omega_2)}{4}\\
C_2(\omega_1,\omega_2) &= \frac{-R(\omega_1,\omega_2) + B(\omega_1,\omega_2)}{4}.
\end{aligned}
$$

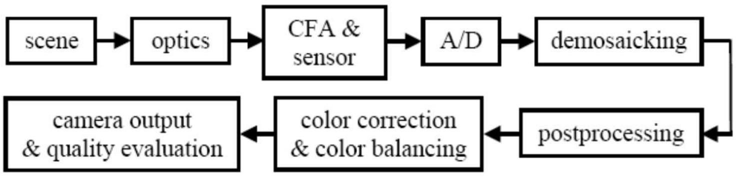




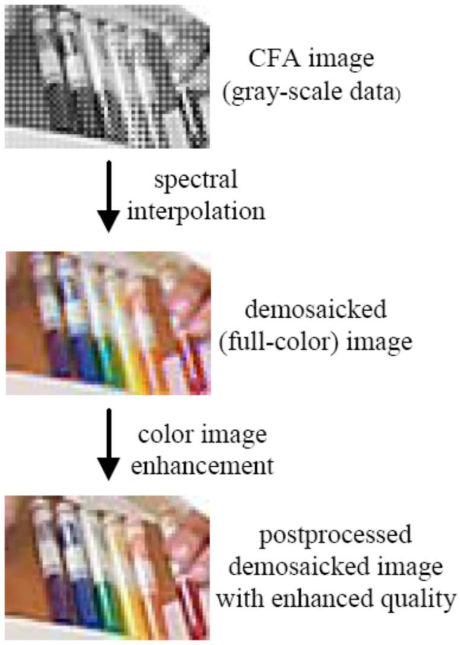
Post processing

Full-color image enhancement

- postprocessing the demosaicked image is an optional step
- implemented mainly in software and activated by the end-user

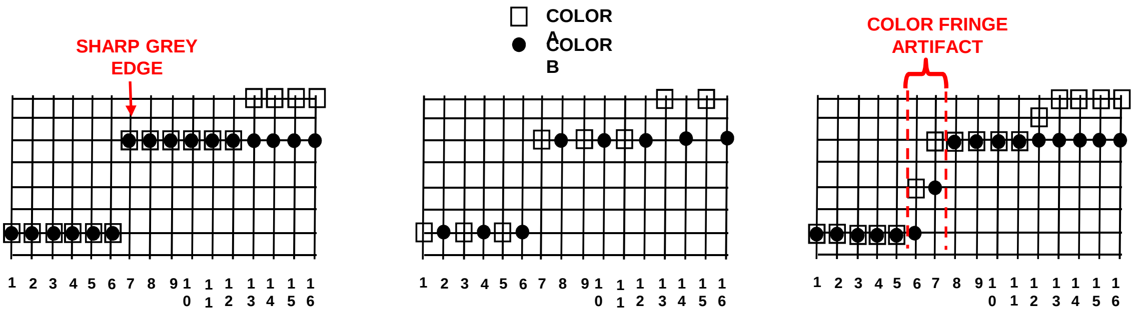


- localizes and eliminates false colors created during demosaicking
- improves both the color appearance and the sharpness of the demosaicked image
- unlike demosaicking, postprocessing can be applied iteratively until certain quality criteria are met



Aliasing

- Color interpolation can provide images with objectionable aliasing artifacts, such as “color fringes” near sharp edges.



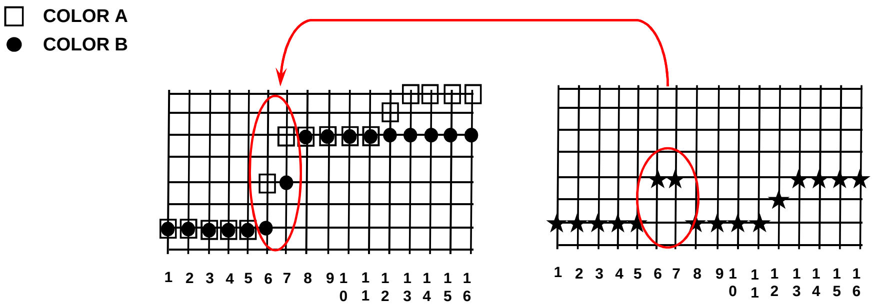
Distribution of light intensity incident to an image sensing array

Illumination incident to the image sensing array in which alternate pixels are overlapped by different colored filters

Distribution of linearly interpolated color light intensity values for the pixels of the image sensing array

Inter-channel differences

- Because of inter-channel correlation, the difference between two colors in a neighborhood is nearly constant;
- The difference between two colors rapidly increases and decreases in the area of sharp grey edges, where color interpolation has introduced false colors;



Information signal corresponding to colors A and B

Difference between colors A and B for each pixel

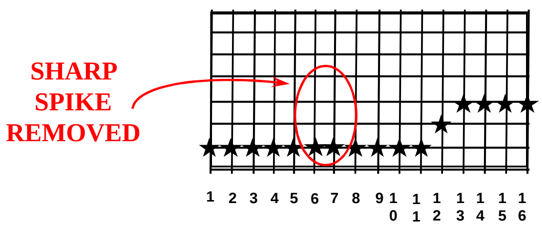
Median filter

- Median filter, over a given support (e.g. 3x3 mask), operates to remove sharp spikes and valleys, leaving sharp monotonically increasing or decreasing edges intact;

$$v_{RG} = median\ \{R_{ij} - G_{ij} \mid (i, j) \in \Re\}$$

$$v_{BG} = median\ \{B_{ij} - G_{ij} \mid (i, j) \in \Re\}$$

where $\Re$ is the support of the median filter



Color A minus color B, median filtered with a support of 5 pixels

References

- http://www.site.uottawa.ca/~edubois/courses/CEG4311/slides/InterpolationRGBcomponents.ppt
- http://www.dmi.unict.it/~battiato/EI_MOBILE0708/EI_MOBILE0708.htm
- Gunturk BK, Glotzbach J, Altunbasak Y, Schafer RW, Mersereau RM. Demosaicking: Color Filter Array Interpolation. IEEE Signal Proc Magazine January 2005; 22(1): 44-54.

IMAGE PROCESSING LABORATORY