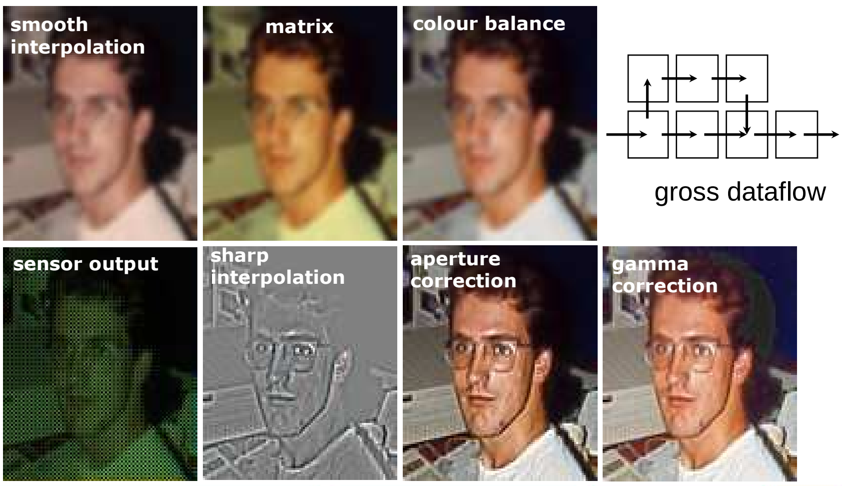
Color Interpolation

Mirko Guarnera

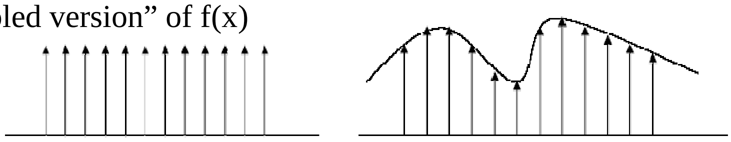
STMicroelectronics

Colour Processing Flow

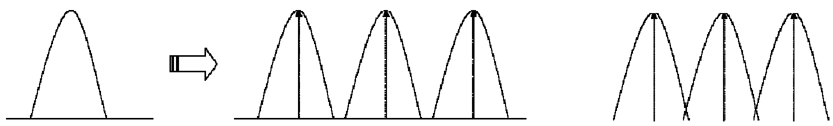


Ideal Interpolation

Shannon Sampling Theorem:
When a "train of impulse" comb(x) is multiplied by f(x), it gives us a "sampled version" of f(x)

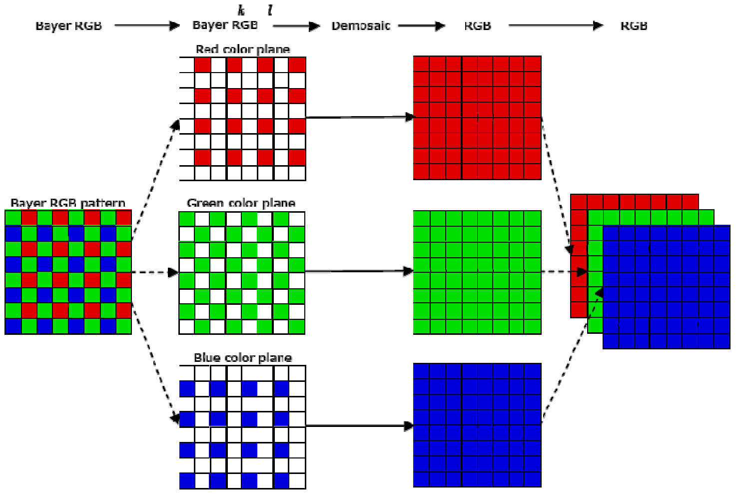


comb(x)f(x), in frequency domain, becomes convolution
Convolving with an impulse is shifting comb(s) * F(s) is replicating the spectrum F(s) at the different impulse locations

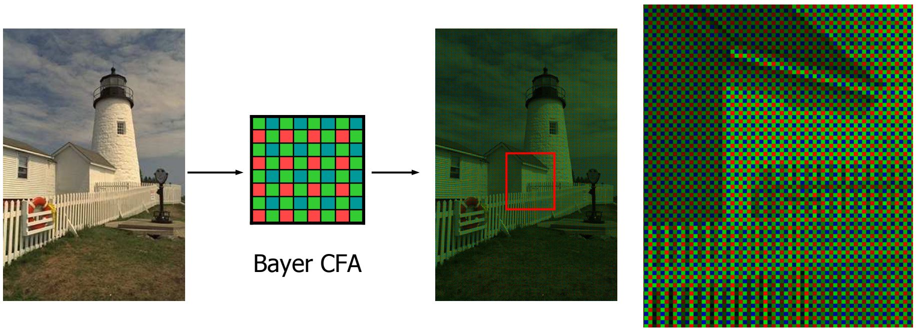


Color Interpolation

$$c(x,y)=\sum_k \sum_l c(x_k,y_l)h(|x-x_k|)h(|y-y_l|)$$



CFA



Bayer CFA

CFA image

Single-Chip Digital Camera

- Images suffer from color artifacts when the samples are not estimated correctly.

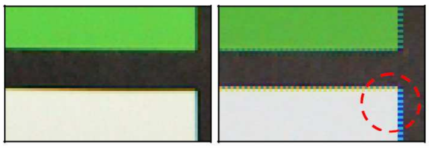
Original image

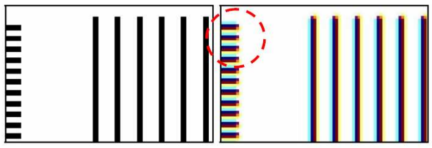
Bilinearly interpolated from CFA-filtered samples

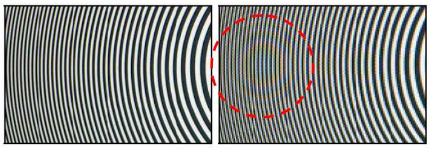
Color artifacts

❖ Common artifacts

➢ Most visual artifacts appear at edges and areas of high frequency

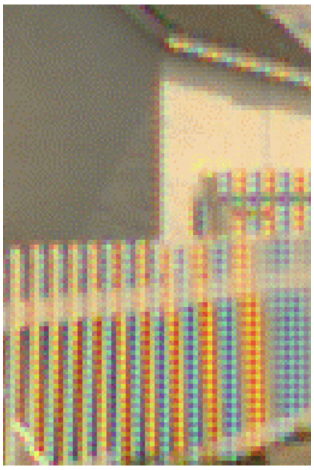

Zipper effect


False color


Aliasing


Blurring

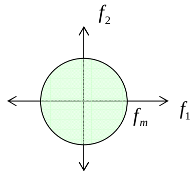
Zipper Effect


Bilinear interpolation

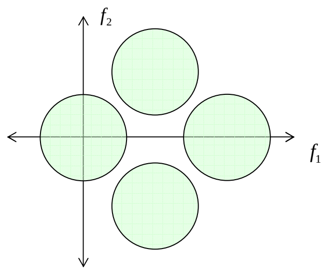

Weighted interpolation

Aliasing

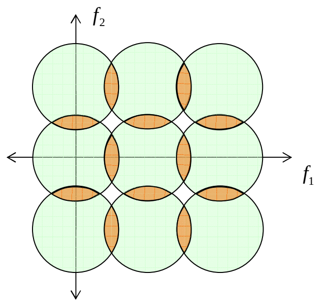
Frequency spectrum of an image:



After CFA sampling: 



Green channel

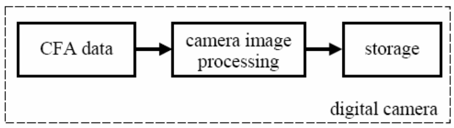


Red/Blue channel

Implementation

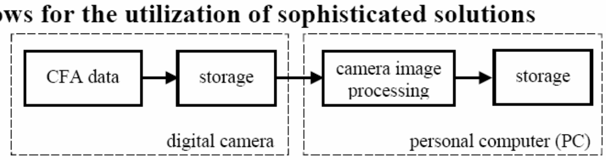
Conventional digital camera

- real-time constraints (computational simplicity requiremen



Using a companion personal computer (PC)

- PC interfaces with the digital camera which stores the ima in the raw CFA format
- allows for the utilization of sophisticated solutions



Models

Ability to follow the spectral image characteristics

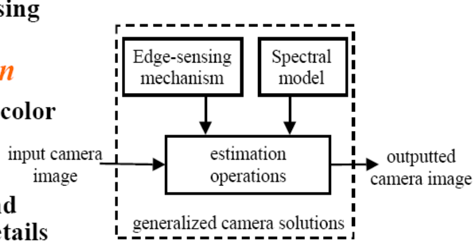
- component-wise (marginal) processing (component → component)
- spectral model-based processing (vector → component)
- vector processing (vector → vector)

Ability to follow the structural image content

- non-adaptive processing
- (data) adaptive processing

Most practical camera solution

- spectral model used to eliminate color shifts and artifacts
- edge-sensing mechanism used to eliminate edge-blurring and to produce sharply-looking fine details

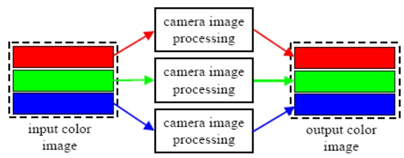


Color Interpolation | Mirko Guarnera

Processing

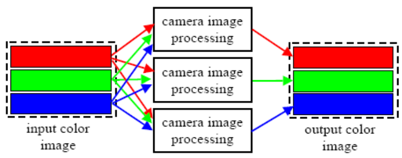
Component-wise processing

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



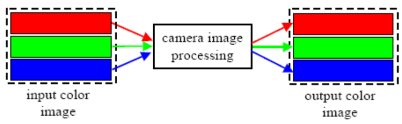
Spectral model based processing

- 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



Color Interpolation | Mirko Guarnera

SM = interchannel correlation

- **Properties of color images**
  - Three color channels have different *DC* components, but ***have very similar high frequency components***.
  - Three color channels ***have very similar characteristics such as a texture and edge location***.
- Most existing demosaicking methods utilize not only intra-channel correlation but also ***inter-channel correlation among three primary color channels***, to interpolate decimated color channels.
- **How to utilize inter-channel correlation**
  - Color-ratio rule
  - Color-difference rule

Color correlation

- **Color-Ratio Rule (1) :**
  - Based on a simplified model of color-image formation by viewing a Lambertian non-flat surface patch:

$$I_a(x) = \rho_a(x) \times \left(\vec{N}(x), \vec{l}\right) \quad , \quad a = R, G, B$$

$I_a(x)$: Pixel intensity of channel a at the location of x

$\vec{N}(x)$: Surface normal, $\vec{l}$: Light source

$\rho_a(x)$: Albedo of channel a of the surface material

  - If the albedo is constant within a given object of an image, the color-ratio rule will hold true within the object region:

$$K_{a,b}(x) = \frac{I_a(x)}{I_b(x)} = \frac{\rho_a}{\rho_b} = \text{constant}$$

Color Correlation

- **Color-Ratio Rule (2) :**
  - Combination with a linear signal model: *signals are linearly varying*
    - Linear interpolation model is not valid in the domain of the color ratio. ⇒ Lack of convexity

$$K_{a,b}(x+\tau) \neq \tau \cdot K_{a,b}(x+1) + (1-\tau) \cdot K_{a,b}(x)$$

    - Linear interpolation model holds true in the logarithmic domain of the color ratio.

$$\log\left(K_{a,b}(x+\tau)\right) = \tau \cdot \log\left(K_{a,b}(x+1)\right) + (1-\tau) \cdot \log\left(K_{a,b}(x)\right)$$

⇒ Convex rule, but high computation cost

  - Constant hue-based method, etc

Color Correlation

- **Color-Difference Rule (1) :**
  - Color-difference signals are approximately flat within a given object in an image.
    - Color-difference signals:

$$D_{a,b}(x) = I_a(x) - I_b(x) \quad , \quad a,b = R,G,B$$

  - Combination with a linear signal model: *signals are linearly varying*
    - Linear interpolation model holds true in the domain of the color difference.

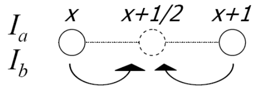
$$D_{a,b}(x+\tau) = \tau \cdot D_{a,b}(x+1) + (1-\tau) \cdot D_{a,b}(x)$$

⇒ Low computational cost

  - Gradient-based method, etc

Color correlation

- **Color Difference Rule (2) :**
  - Example:



Linear interpolation scheme of a color difference

$$D_{a,b}\left(x+\frac{1}{2}\right)=\frac{1}{2}\cdot D_{a,b}\left(x+1\right)+\frac{1}{2}\cdot D_{a,b}\left(x\right)$$

⇩

Interpolation of color channel I_a using the other color channel I_b

$$I_a\left(x+\frac{1}{2}\right)=\left\{\frac{I_a\left(x+1\right)+I_a\left(x\right)}{2}\right\}-\left\{\frac{I_b\left(x+1\right)-2\cdot I_b\left(x+1/2\right)+I_b\left(x\right)}{2}\right\}$$

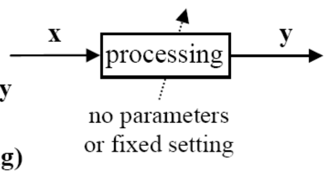
Averaging of I_a

2nd derivative calculated from I_b : Image sharpening (Peaking)

Classification: Non adaptive vs adaptive

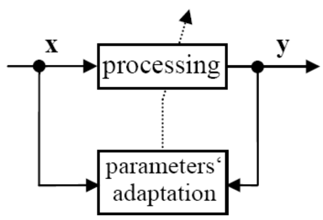
Non-adaptive processing

- **no data-adaptive control**
- **often reduces to linear processing - easy to implement**
- **inefficient performance (image blurring)**



Adaptive processing

- **edge-sensing weights used to follow the structural content**
- **nonlinear processing**
- **essential in producing high-quality, sharply looking images**

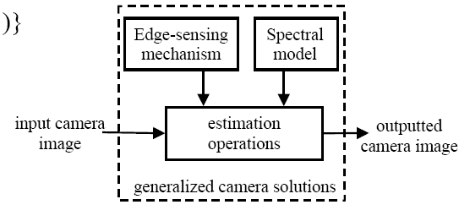


Data adaptive

Construction • using spatial, structural, and spectral characteristics

$$\mathbf{x}_{(p,q)} = \sum_{(i,j)\in\zeta} \{w'_{(i,j)} \Psi(\mathbf{x}_{(i,j)}, \mathbf{x}_{(p,q)})\}$$

$$w'_{(i,j)} = w_{(i,j)} / \sum_{(i,j)\in\zeta} w_{(i,j)}$$



Spatial characteristics

- local neighborhood area ζ

Structural characteristics

- edge-sensing mechanism λ

$\lambda(z) \rightarrow \{w_{(i,j)}, (i,j) \in \zeta\}$

z denotes the CFA image

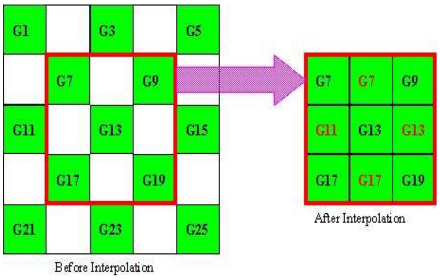
Spectral characteristics

- spectral model Ψ

Non adaptive schemes

Color Interpolation - Nearest Neighbor Replication

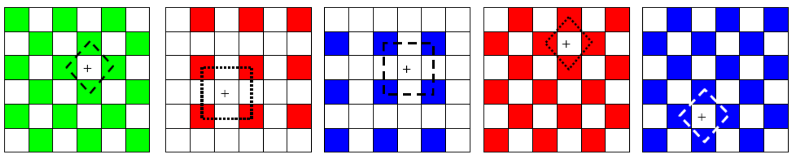
- Each interpolated output pixel is assigned the value of the nearest pixel in the input image
- The nearest neighbor can be any one of the upper, lower, left and right pixels
- For example, for a 3x3 block in green plane, we assume the left neighboring pixel value is used to fill the missing ones



Local Neighbour area

Features

- approximation using a shape mask ζ
- shape and size of ζ vary depending on the CFA used and processing task (demosaicking, resizing, etc.)
- shape masks widely used in the demosaicking process:

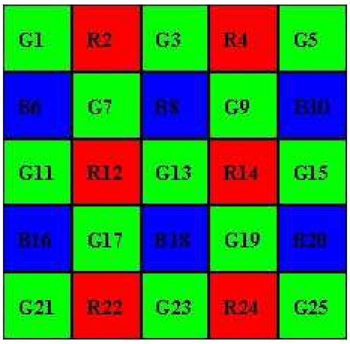


$$(a,d,e) \quad \zeta = \{(p-1,q),(p,q-1),(p,q+1),(p+1,q)\}$$

$$(b,c) \quad \zeta = \{(p-1,q-1),(p-1,q+1),(p+1,q-1),(p+1,q+1)\}$$

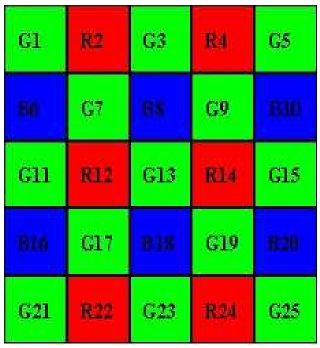
Color Interpolation - Bilinear Interpolation

- 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

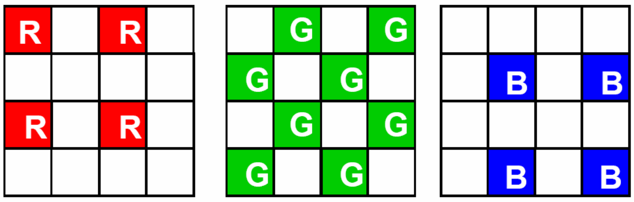
- Interpolation of a red/blue pixel at a blue/red position
  - The average of four adjacent diagonal pixel values is assigned to the interpolated pixel
  - R8 = (R2+R4+R12+R14) / 4
  - B12 = (B6+B8+B16+B18) / 4



Bilinear

- Simple realization with *3 by 3* filter kernels

$$R_F(n_1,n_2) = \mathbf{F}_R \otimes M_R(n_1,n_2)$$
$$G_F(n_1,n_2) = \mathbf{F}_G \otimes M_G(n_1,n_2)$$
$$B_F(n_1,n_2) = \mathbf{F}_B \otimes M_B(n_1,n_2)$$
$$\mathbf{F}_R = \mathbf{F}_B = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \Big/ 4 \quad , \quad \mathbf{F}_G = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \Big/ 4$$



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

Adaptive schemes

Edge sensing mechanism (ESM)

Essential to produce sharply looking images

- structural constraints impposed on the camera solution relate to the form of the ESM operator λ used to generate the edge-sensing weights $\lambda(z) \rightarrow \{w_{(i,j)}, (i,j) \in \zeta\}$
- both structural and spatial characteristics are considered in the ESM construction

Concept

- ESM operator λ uses some form of inverse gradient of the samples in the CFA image

$$w_{(i,j)} = \frac{1}{1 + f(\Delta_{(i,j)})}$$

- large image gradients usually indicate that the corresponding vectors are located across edges (penalized through small weights)

ESM

Conventional designs:

- operate on large (5x5,7x7) neighborhood
- specialization on a particular CFA (e.g. Bayer CFA):

for shape mask $\zeta = \{(p-1,q),(p,q-1),(p,q+1),(p+1,q)\}$

$$\begin{aligned}
w_{(p\text{-}1,q)} &= 1/(1+|z_{(p\text{-}2,q)} - z_{(p,q)}| + |z_{(p\text{-}1,q)} - z_{(p+1,q)}|) \\
w_{(p,q-1)} &= 1/(1+|z_{(p,q-2)} - z_{(p,q)}| + |z_{(p,q-1)} - z_{(p,q+1)}|) \\
w_{(p,q+1)} &= 1/(1+|z_{(p,q+2)} - z_{(p,q)}| + |z_{(p,q+1)} - z_{(p,q-1)}|) \\
w_{(p+1,q)} &= 1/(1+|z_{(p+2,q)} - z_{(p,q)}| + |z_{(p+1,q)} - z_{(p-1,q)}|)
\end{aligned}$$

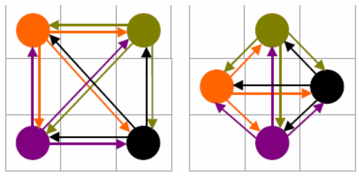
for shape mask $\zeta = \{(p-1,q-1),(p-1,q+1),(p+1,q-1),(p+1,q+1)\}$

$$\begin{aligned}
w_{(p\text{-}1,q-1)} &= 1/(1+|z_{(p\text{-}2,q-2)} - z_{(p,q)}| + |z_{(p\text{-}1,q-1)} - z_{(p+1,q+1)}|) \\
w_{(p\text{-}1,q+1)} &= 1/(1+|z_{(p\text{-}2,q+2)} - z_{(p,q)}| + |z_{(p\text{-}1,q+1)} - z_{(p+1,q-1)}|) \\
w_{(p+1,q-1)} &= 1/(1+|z_{(p+2,q-2)} - z_{(p,q)}| + |z_{(p+1,q-1)} - z_{(p-1,q+1)}|) \\
w_{(p+1,q+1)} &= 1/(1+|z_{(p+2,q+2)} - z_{(p,q)}| + |z_{(p+1,q+1)} - z_{(p-1,q-1)}|)
\end{aligned}$$

ESM

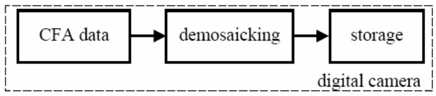
Cost-effective, universal design

- **operates within the shape mask ζ**
- **aggregation concept defined here over the four-neighborhoods only**
- **desing suitable for any existing CFA**

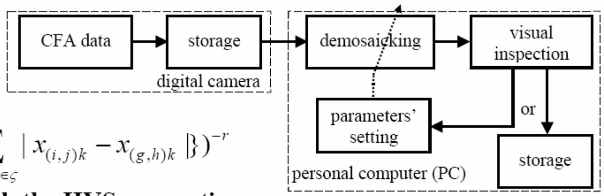


Fully automated solution

$$w_{(i,j)} = 1/(1 + \sum_{(g,h)\in\zeta} |x_{(i,j)k} - x_{(g,h)k}|)$$



End-user control based solution



$$w_{(i,j)} = \beta(1 + \exp\{\sum_{(g,h)\in\zeta} |x_{(i,j)k} - x_{(g,h)k}|\})^{-r}$$

- **may better match the HVS properties**

Edge Sensing

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

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

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

Define some threshold value T

The algorithm then can be described as:

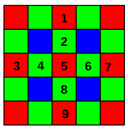
```
If ΔH < T AND ΔV > T,
        G8 = (G7 + G9) / 2;
Else if ΔH > T AND ΔV < T,
        G8 = (G3 + G13) / 2;
Else
        G8 = (G3 + G7 + G9 + G13) / 4
End
```

The choice of T depends on the images and can have defferent 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

- Edge-Directed Interpolation: Based on the assumption that color channels have similar texture, various edge detectors can be used.



Edge-directed interpolation

1. Calculate horizontal gradient $\Delta H = | (R3 + R7)/2 - R5 |$
2. Calculate vertical gradient $\Delta V = | (R1 + R9)/2 - R5 |$
3. If $\Delta H > \Delta V$,
   $G5 = (G2 + G8)/2$
   Else if $\Delta H < \Delta V$,
   $G5 = (G4 + G6)/2$
   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.

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

→ example

(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.

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

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

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

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

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

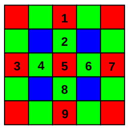
R_{11}	G_{12}	R_{13}	G_{14}	R_{15}	G_{16}
G_{21}	B_{22}	G_{23}	B_{24}	G_{25}	B_{16}
R_{31}	G_{32}	R_{33}	G_{34}	R_{35}	G_{36}
G_{41}	B_{42}	G_{43}	B_{44}	G_{45}	B_{46}
R_{51}	G_{52}	R_{53}	G_{54}	R_{55}	G_{56}
G_{61}	B_{62}	G_{63}	B_{64}	G_{65}	B_{66}

(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.

Demosaicking Approaches

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



1. Calculate horizontal gradient ΔH = |G4 – G6| + |R5 – R3 + R5 – R7|
2. Calculate vertical gradient ΔV = |G2 – G8| + |R5 – R1 + R5 – R9|
3. If ΔH > ΔV,
   G5 = (G2 + G8)/2 + (R5 – R1 + R5 – R9)/4
   Else if ΔH < ΔV,
   G5 = (G4 + G6)/2 + (R5 – R3 + R5 – R7)/4
   Else
   G5 = (G2 + G8 + G4 + G6)/4 + (R5 – R1 + R5 – R9 + R5 – R3 + R5 – R7)/8

Adaptive interpolation

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

<u>Step 1:</u> **Edge-directed interpolation for the *G* channel**

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

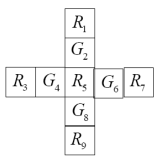
(1) Compute magnitude sums of first-order derivatives and second-order derivatives.

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

(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.

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



Adaptive interpolation

<u>Step 2:</u> 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

$$R_5 = \frac{R_3 + R_7}{2} - \frac{G_3 - 2 \cdot G_5 + G_7}{2} \quad , \; if \; \alpha < \beta$$

$$R_5 = \frac{R_1 + R_9}{2} - \frac{G_1 - 2 \cdot G_5 + G_9}{2} \quad , \; if \; \alpha > \beta$$

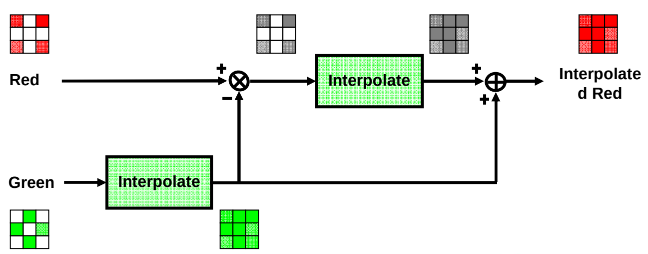
$$R_5 = \frac{R_1 + R_3 + R_7 + R_9}{4} - \frac{G_1 + G_3 - 4 \cdot G_5 + G_7 + G_9}{4} \quad , \; if \; \alpha = \beta$$

$$\alpha = abs(G_3 - 2 \cdot G_5 + G_7) + abs(R_7 - R_3)$$

$$\beta = abs(G_1 - 2 \cdot G_5 + G_9) + abs(R_9 - R_1)$$

Demosaicking Approaches

- Constant-Hue-Based Interpolation: Hue does not change abruptly within a small neighborhood.
  - Interpolate green channel first.
  - Interpolate hue (defined as either color differences or color ratios).
  - Estimate the missing (red/blue) from the interpolated hue.



Constant Hue

To utilize inter-channel correlation according to the color-ratio rule

Step 1 : The *G* channel is interpolated by the bi-linear method.

Step 2 : The *R* and the *B* channels are interpolated by averaging their neighboring color ratios (***R*/*G***), (***B*/*G***), and then multiplying the average color ratios by its *G* value.

$$G_{22} = \frac{G_{12}+G_{21}+G_{23}+G_{32}}{4}$$

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

$$B_{33} = G_{33}\frac{\frac{B_{22}}{G_{22}}+\frac{B_{24}}{G_{24}}+\frac{B_{42}}{G_{42}}+\frac{B_{44}}{G_{44}}}{4}$$

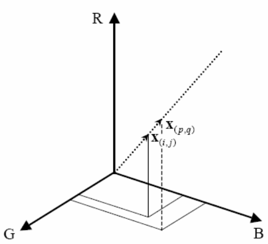
R_{11}	G_{12}	R_{13}	G_{14}	R_{15}
G_{21}	B_{22}	G_{23}	B_{24}	G_{25}
R_{31}	G_{32}	R_{33}	G_{34}	R_{35}
G_{41}	B_{42}	G_{43}	B_{44}	G_{45}
R_{51}	G_{52}	R_{53}	G_{54}	R_{55}

we define blue "hue value" as : B/G. red "hue value" can be analogously defined.
Considering the interpolation of blue pixel values : there are three different cases of blue pixel value interpolations.

Vector SM

Geometric interpretation

- from three-component vector expression $y = y_1 = y_2 = \frac{-b}{2a}$



for G component

$$x_{(p,q)2} = \frac{x_{(p,q)1}x_{(i,j)1}x_{(i,j)2} + x_{(p,q)3}x_{(i,j)2}x_{(i,j)3}}{x_{(i,j)1}^2 + x_{(i,j)3}^2}$$

for R component

$$x_{(p,q)1} = \frac{x_{(p,q)2}x_{(i,j)1}x_{(i,j)2} + x_{(p,q)3}x_{(i,j)1}x_{(i,j)3}}{x_{(i,j)2}^2 + x_{(i,j)3}^2}$$

for B component

$$x_{(p,q)3} = \frac{x_{(p,q)1}x_{(i,j)1}x_{(i,j)3} + x_{(p,q)2}x_{(i,j)2}x_{(i,j)3}}{x_{(i,j)1}^2 + x_{(i,j)2}^2}$$

Directional filtering

- The idea behind…

Directional Filtering

- n different 5x5 filter kernels are used to follow gradient orientation

$$\mathrm{filter}(x, y, \alpha) = K \exp\left(-\frac{\tilde{x}^2}{2\sigma_x^2} - \frac{\tilde{y}^2}{2\sigma_y^2}\right)$$

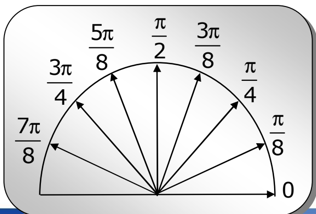
where

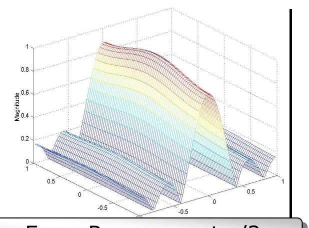
$$\tilde{x} = x\cos(\alpha) - y\sin(\alpha) \qquad \sigma_x = 8$$

$$\tilde{y} = x\sin(\alpha) + y\cos(\alpha) \qquad \sigma_y = 0.38$$

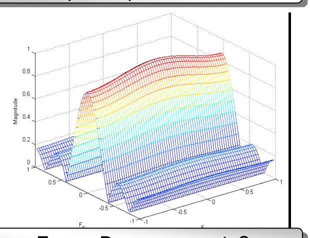
and K is the Normalization Constant

Involved α Directions



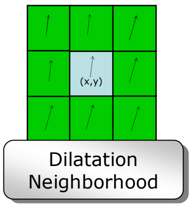


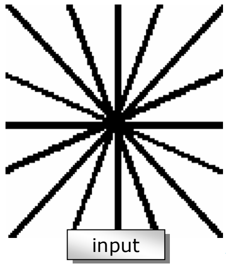
Freq. Response at π/2

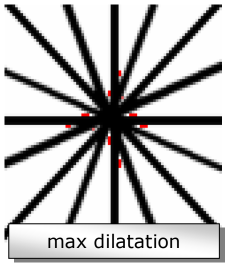


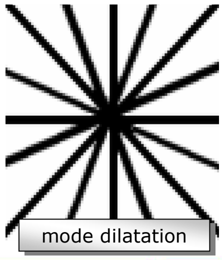
Freq. Response at 0

DF









- To find a more precise orientation the ***statistical weighted mode***, on 3x3 neighborhood of the point (x,y), is achieved.

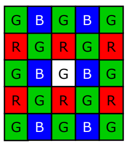
$$\alpha = \underset{\substack{i=-1..1\\ j=-1..1}}{mode}(\alpha'_{(x+i,y+j)})$$

- The max wrong values (underlined in red) have been removed.

DF: peaking

The Directional Filtering on the know value has been used to obtain the color correlations. For instance, on a known central Green value :

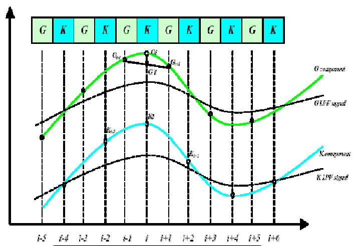
$$\Delta_{Peak} = G - G_{LPF\_DF}$$



Therefore, for the unknown *H* values (R/B):

$$H = H_{LPF\_DF} + \Delta_{Peak}$$

The enhancement step takes into account the colors correlation, adding a peaking term to the color channels to be estimated in a central pixel.



Correlated Color Components Retrieval

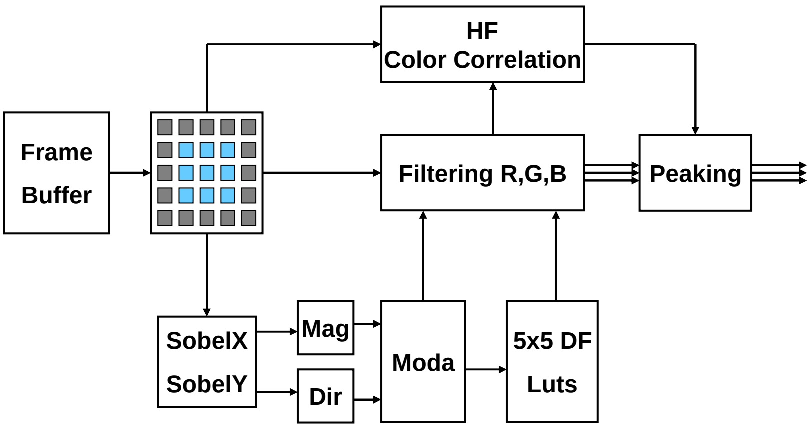
Peak – no Peak

The differences between the first release and the second release are drastically visible.

The main enhancements are introduced by the new full gradient and the weighted mode dilatation jointly to the unfiltered Bayer pattern values.

A further enhancement has been introduced by considering the cross-correlation among the Bayer channels.

Block Scheme

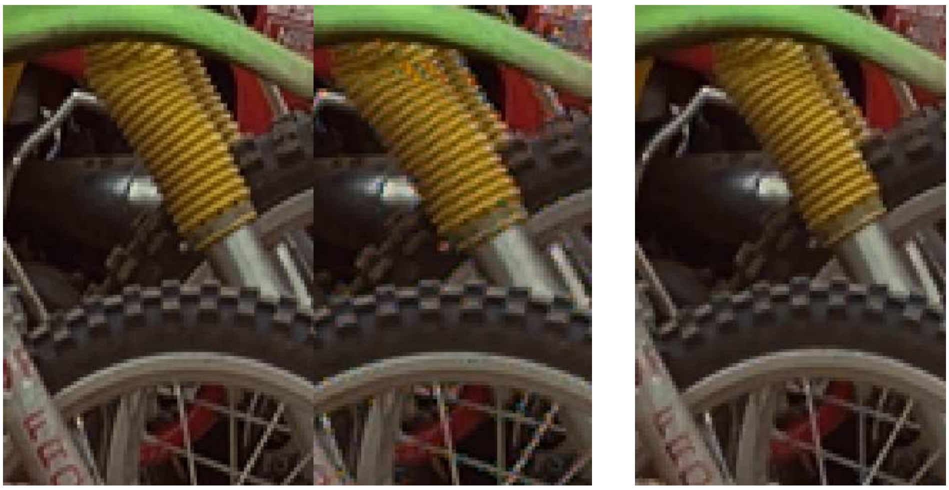


Results

Original Image | DF Color Interpolation | IGP

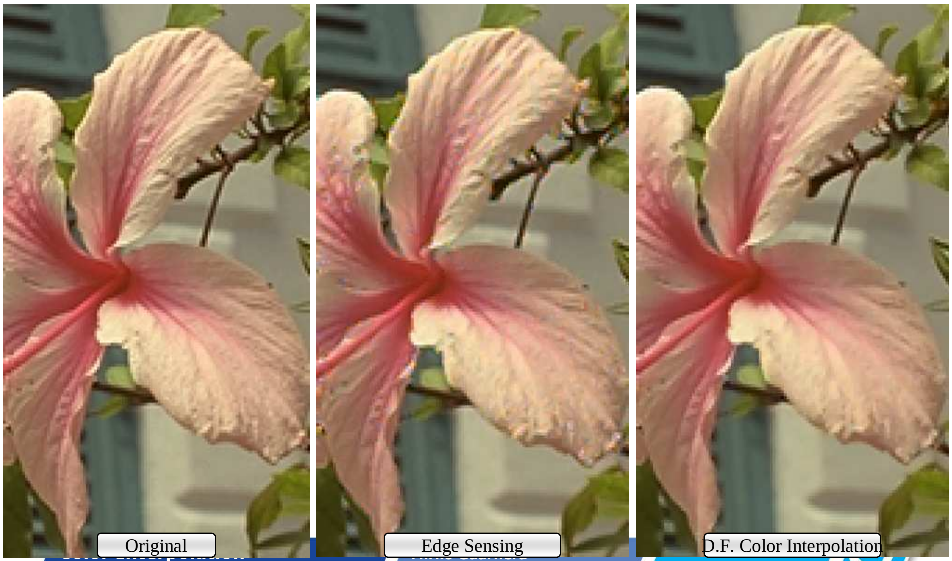
Comparison of Directional Color Interpolation and IGP (Recon) Color Interpolation (involving Color Interpolation + Antialiasing + Peaking)

Results (1/6)



Original | Edge Sensing | D.F. Color Interpolation

Results (2/6)



Results (3/6)

Results (4/6)

Original | Edge Sensing | D.F. Color Interpolation

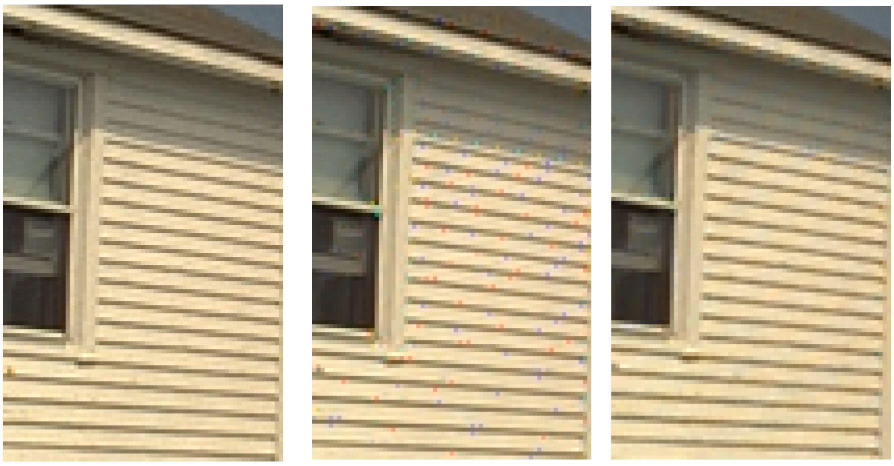
Color

Results (5/6)

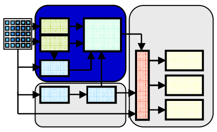
Original | Edge Sensing | D.F. Color Interpolation

Color Interpolation | Mirko Guarnera

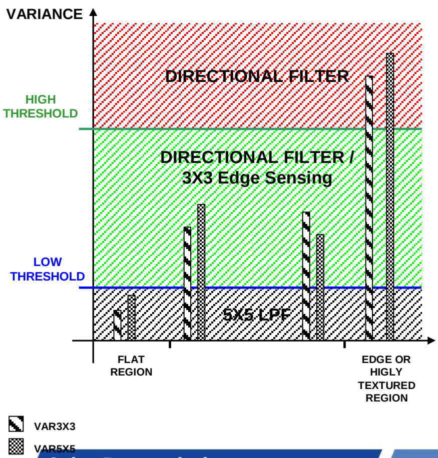
Results (6/6)



Original | Edge Sensing | D.F. Color Interpolation



Neighbor Analysis: variances

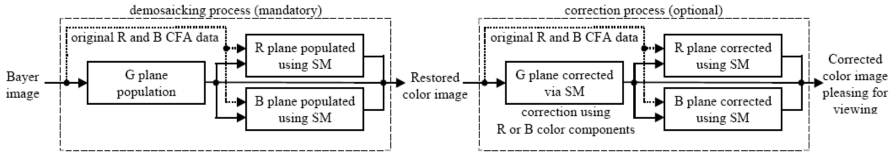


Color-processing sideshows

- Aliasing
  - Resolution of sensor
    < spatial frequency of scene
- False color
  - Aliasing occurs only in one or two of the color planes
- Moire patterns
  - Overlapped patterns with near frequency

Post processing

- it is often connected with the demosaicked image postprocessing viewed as a correction step

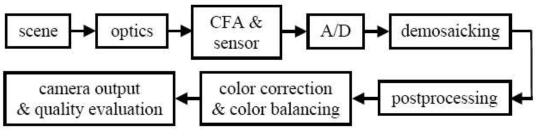


- demosaicking and demosaicked image postprocessing are two processing steps are fundamentally different, although may employ similar, if not identical, signal processing concepts
- postprocessing of demosaicked images is a novel application of great importance to both the end-users and the camera manufacturers

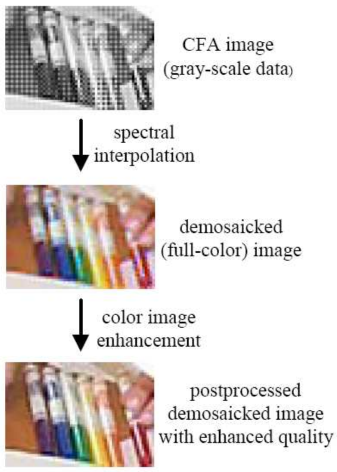
Post processing

Full-color image enhancement

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



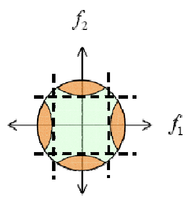
- localizes and eliminates false colors created during demosaicking
- improves both the color appearance and the sharpness of the demosaicked image
- unlike demosaicking, postprocessing can be applied iteratively until certain quality criteria are met



Aliasing

Aliasing Cancelling

- Alias Cancelling: Based on the assumption that red, green, and blue channels have similar frequency components, the high-frequency components of red and blue channels are replaced by the high-frequency components of green channel.



Red/Blue channel

Alias Cancellation Results

Standard Interpolation

Alias Cancellation

Alias Cancellation Results

Standard Interpolation

Alias Cancellation

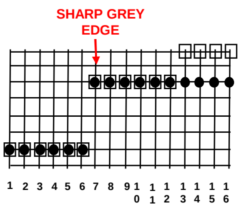
Alias Cancellation Results

Standard Interpolation

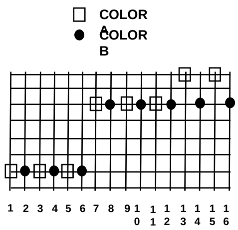
Alias Cancellation

Aliasing

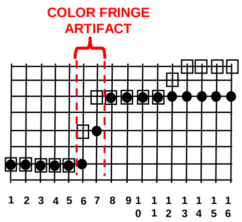
- Color interpolation can provide images with objectionable aliasing artifacts, such as "color fringes" near sharp edges.



Distribution of light intensity incident to an image sensing array



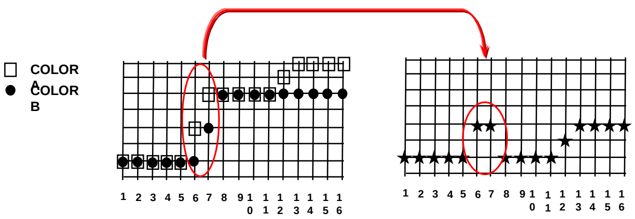
Illumination incident to the image sensing array in which alternate pixels are overlapped by different colored filters



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;



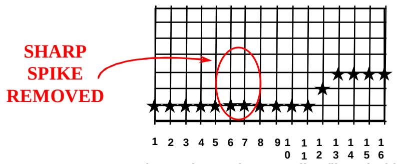
Information signal corresponding to colors A and B

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;

$$v_{RG} = median\{R_{ij} - G_{ij} \mid (i,j) \in \Re\}$$
$$v_{BG} = median\{B_{ij} - G_{ij} \mid (i,j) \in \Re\}$$
$$\text{where } \Re \text{ is the support of the median filter}$$



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

Color reconstruction with anti-aliasing

- The median values are appropriately subtracted or added to the color received by each pixel to obtain the other two:

  - ✓ Pixels which receive only Red light:

$$\hat{R}_{CENTER} = R_{Bayer}$$
$$\hat{G}_{CENTER} = \hat{R}_{CENTER} - v_{RG}$$
$$\hat{B}_{CENTER} = \hat{R}_{CENTER} - v_{RG} + v_{BG}$$

  - ✓ Pixels which receive only Green light:

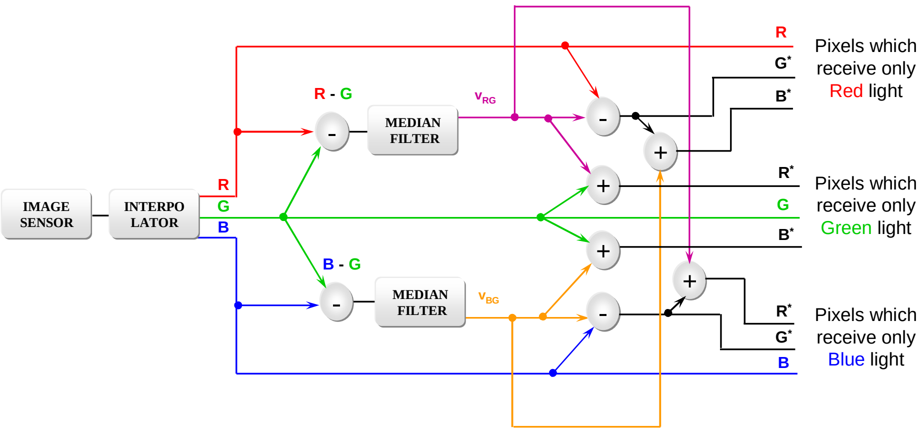
$$\hat{G}_{CENTER} = G_{Bayer}$$
$$\hat{R}_{CENTER} = \hat{G}_{CENTER} + v_{RG}$$
$$\hat{B}_{CENTER} = \hat{G}_{CENTER} + v_{BG}$$

  - ✓ Pixels which receive only Blue light:

$$\hat{B}_{CENTER} = B_{Bayer}$$
$$\hat{G}_{CENTER} = \hat{B}_{CENTER} - v_{BG}$$
$$\hat{R}_{CENTER} = \hat{B}_{CENTER} - v_{BG} + v_{RG}$$

Block scheme

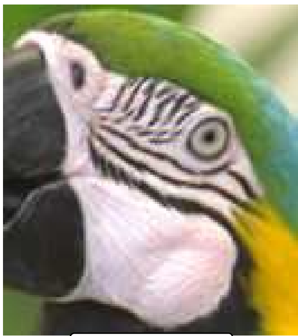
- The rules presented in the previous slide are schematically represented by the block diagram below:



Block diagram representing our anti-aliasing algorithm

Initial color interpolated image and post-processed image

Original



DF Color Interpolated

Post-processed with our method

Variance of inter-channel differences

- The variance of inter-channel differences can be used mainly for two purposes:

1. To discriminate whether the antialiasing should be performed or not;
2. To achieve the color correction by using more the flatter color difference domain than the other one.

Local Statistics Computation

- The local statistics are computed by the weighted sample mean and the variance from a running square window h, whose central value is A_{ij}:

Expectation value

$$E[A_{ij}] = \frac{\sum_{k,l \in h} e(k,l) \cdot A_{kl}}{\sum_{k,l \in h} e(k,l)}$$

Variance

$$\sigma_A^2(i,j) = \frac{\sum_{k,l \in h} e(k,l) \cdot (A_{kl} - E[A_{ij}])^2}{\sum_{k,l \in h} e(k,l)}$$

Weight function

$$e(k,l) = 1 - (A_{ij} - A_{kl})^2$$

Flat color differences region

- In the color difference domain, a flat color difference neighborhood is characterized by:

1. Expectation value close to the central value

$$\left| E[G_{ij} - R_{ij}] - (G_{ij} - R_{ij}) \right| < MeanThreshold$$
$$\left| E[G_{ij} - B_{ij}] - (G_{ij} - B_{ij}) \right| < MeanThreshold$$

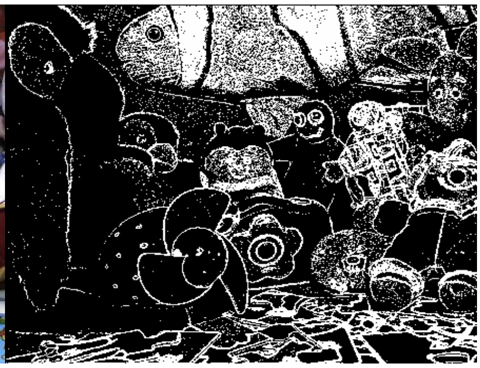
2. Low Variance value

$$\sigma^2_{(G-R)}(i, j) < VarianceThreshold$$
$$\sigma^2_{(G-B)}(i, j) < VarianceThreshold$$

Homogeneous vs Inhomogeneous regions

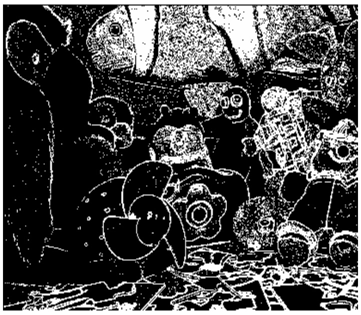
- According to the Expectation value and to the Variance, a map of homogeneous (black) vs. inhomogeneous (white) regions can be achieved

Processed image



Map of homogeneous vs. inhomogeneous regions

Variance Antialiasing: adaptive application

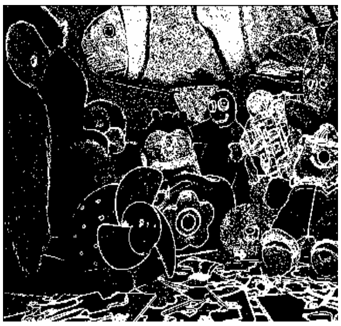


- Homogeneous regions (black) can be left unchanged or can be low pass filtered;
- Inhomogeneous regions (white) are processed by the color correction algorithm.

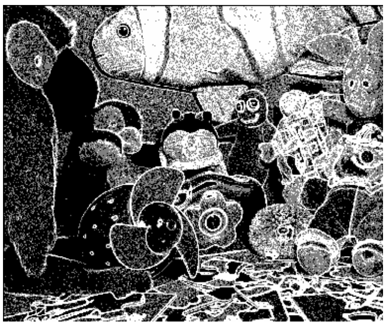
Adaptive application with two thresholds for variance

- Two different thresholds for the variance allow us to identify three regions (black, grey, white) and three different behaviors:

1. Low pass filter; 2. Nothing; 3. Color correction algorithm.



Map of variance with one threshold



Map of variance with two thresholds

Variance Antialiasing:
Color Correction algorithm (1/3)

First Step: Two updated values for the **green channel** are calculated using each color difference domain:

$$G_{ij}^{R} = R_{ij} + \nu_{GR}$$
$$G_{ij}^{B} = B_{ij} + \nu_{GB}$$

Where

$$\nu_{GR} = median\{G_{kl} - R_{kl} | (k,l) \in h\}$$
$$\nu_{GB} = median\{G_{kl} - B_{kl} | (k,l) \in h\}$$

h is the support of the 3x3 local windows

Variance Antialiasing:
Color Correction algorithm (2/3)

Second Step: The updated $\hat{G}_{ij}$ value is determined by the weighted sum of the two updated values, computed in the first step, and the initially interpolated value G_{ij}.

$$\hat{G}_{ij} = (1-\beta) \cdot G_{ij} + \beta \cdot \{(1-a) \cdot G_{ij}^{R} + a \cdot G_{ij}^{B}\}$$

The weight a is expressed as:

$$a = \frac{\sigma_{(G-R)}^{2}}{\sigma_{(G-R)}^{2} + \sigma_{(G-B)}^{2}}, 0 < a < 1$$

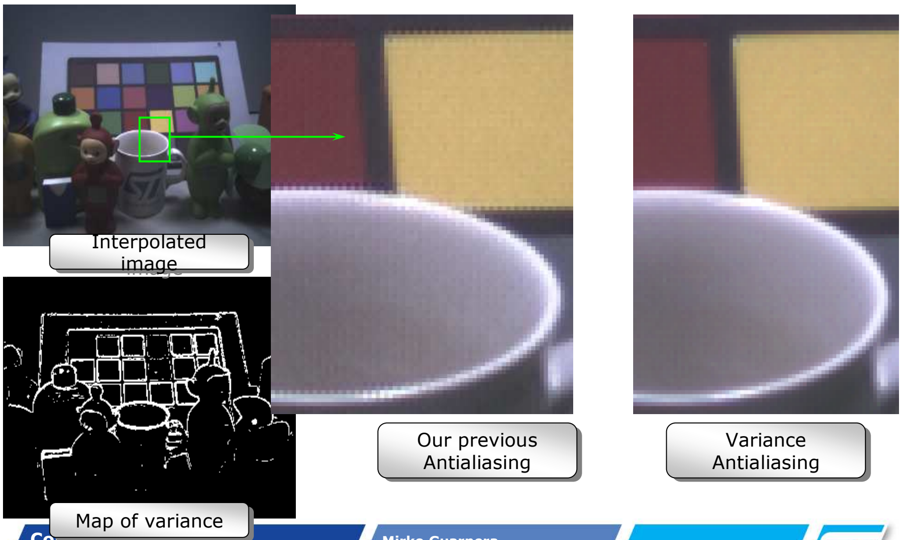
Variance Antialiasing: Color Correction algorithm (3/3)

Third Step: Red and Blue channels are updated according to the updated Green channel and the medians of inter-channel differences.

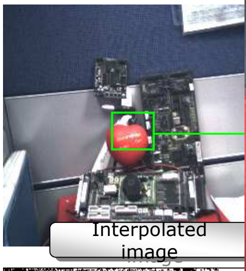
$$\hat{R}_{ij} = \hat{G}_{ij} - \nu_{GR}$$
$$\hat{B}_{ij} = \hat{G}_{ij} - \nu_{GB}$$

The updated $\hat{G}$, $\hat{R}$ and $\hat{B}$ will be involved in filtering the following pixels.

Results (1/2)



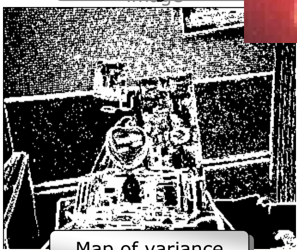
Results (2/2)



Interpolated image

Our previous Antialiasing

Variance Antialiasing

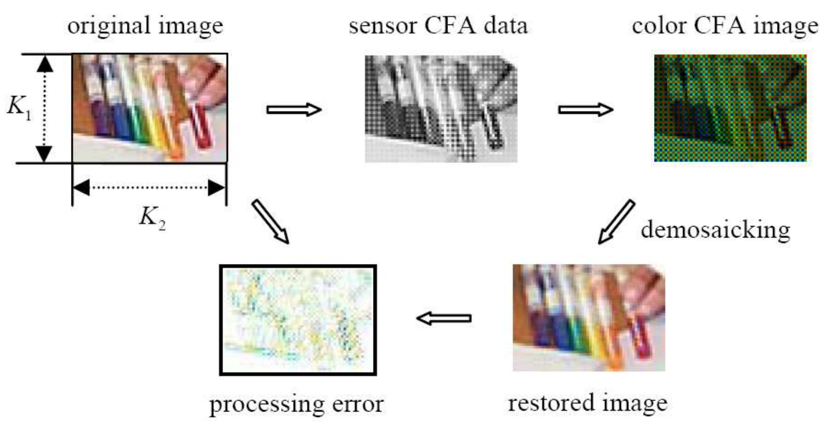


Map of variance

Bibliography

- Adams, James E. "Interactions between color plane interpolation and other image processing functions in electronic photography" Proceedings of SPIE Vol. 2416 P.144-151
- Sakamoto,Tadashi. et.al., "Software pixel interpolation for digital still cameras suitable for a 32-bit MCU" IEEE Transactions on Consumer Electronics. Vol.44, No.4, P.1342-1352, November 1998
- Zen,Hidemori. et.al., "A New digital signal processor for progressive scan CCD" IEEE Transactions on Consumer Electronics. Vol.44, No.2, P.289-295, May 1998
- Parulski,Kenneth A. "Color Filters and Processing Alternatives for one-chip cameras" IEEE Transactions on Electron Devices. Vol.ED-32, NO.8, August 1985
- Chan,Wen-Hsin. et.al., "A Mega-Pixel resolution PC Digital Still Camera" Proceedings of SPIE Vol.2654. P.164-171
- Tsai, Y.Tim. et.al., "Optimized Image Processing Algorithms for a Single Sensor Camera" 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Vol.2, P.1010-1013
- Adams,James E. et.al., "Design of practical color filter array interpolation algorithms for digital cameras" Proceedings of SPIE Vol.3028 P.117-125
- Chang, Ed. et.al., "Color Filter Array Recovery Using a Threshold-based Variable Number of Gradients" to be published in Proceedings of SPIE, January, 1999
- Keys, Robert.G. et.al., "Cubic Convolution Interpolation for Digital Image Processing" IEEE Transactions on Acoustic, Speech and Signal Processing, Vol ASSP-29, P1153-1160, 1981
- Simons, K.W. "Digital Image Reconstructon and Resampling for Geometric Manipulation" Proceedings of IEEE Symp. on Machine Processing of Remotely Sensed Data P.3A-1--3A-11,1975
- Mitchell, Don P. et.al., "Reconstruction Filters in Computer Graphics" Computer Graphics, (SIGGRAPH'88 Proceedings), Vol.22, No.4, P.221-228, August 1988
- Schreiber, William F. et.al., "Transformation between Continuous and Discrete Representations of Image: A Perceptual Approach" IEEE Transactions on Pattern Analysis and Machine Intelligence Vol.PAMI-7 No.2, P.178-186 March 1985

Quality evaluation



Color Interpolation Mirko Guarnera

Metrics

Mean Absolute Error (MAE)

$$MAE = \frac{1}{3NM}\sum_{k=1}^{3}\sum_{i=1}^{N}\sum_{j=1}^{M}\left| o_{i,j}^{k} - x_{i,j}^{k} \right|$$

Mean Square Error (MSE)

$$MSE = \frac{1}{3NM}\sum_{k=1}^{3}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(o_{i,j}^{k} - x_{i,j}^{k} \right)^2$$

Normalised Color Difference (NCD)

- **color chromaticity preservation**
- **expressed in CIE LUV color space**

$$NCD = \frac{\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sqrt{\left(L_{i,j}^{o} - L_{i,j}^{x}\right)^2 + \left(u_{i,j}^{o} - u_{i,j}^{x}\right)^2 + \left(v_{i,j}^{o} - v_{i,j}^{x}\right)^2}}{\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sqrt{\left(L_{i,j}^{o}\right)^2 + \left(u_{i,j}^{o}\right)^2 + \left(v_{i,j}^{o}\right)^2}}$$

Color Interpolation Mirko Guarnera

Metrics

$$PSRN = 20 \times \log_{10}\left(\frac{255}{\sqrt{\frac{1}{N\times M}\sum_{i=1}^{N\times M}(A_i - B_i)^2}}\right)$$

- Higher values (expressed in decibel) of the PSNR generally imply better quality.

- ΔE_{ab} wich measures the eucledian distance between the origina image and the interpolated one in the CIELAB. greater than 2.3 indicates that the differences are visible, mean differences greater than 10 mean that differences between the two images are so hight that the comparison is not worthwhile.

$$\Delta E^*_{ab} = \frac{1}{N \times M}\sum_{i=1}^{N\times M}\left\| A_i^{Lab} - B_i^{Lab} \right\|$$