Image Processing
for
Digital Cameras

Ramakrishna Kakarala

Agilent Technologies

Outline

1. The digital still camera (DSC) market
2. Anatomy of DSC - a signal processing view
3. Image pipeline in a DSC
    - cleaning up the acquisition
    - color interpolation
    - color correction
    - display compensation
    - compression
4. Frontier areas - research topics
5. Conclusions

Digital still cameras

www.dpreview.com

Typical features ($300-$1,000)

- 3-5 Million pixels (megapixels), i.e, 2588x1954
- Flash memory storage (CF, SD, MS)
- Video clip capture (MPEG, Quicktime)
- LCD preview

Large and growing market

- 14M units (2003)
- digital > film (unit vol) in 2008

www.infotrends-rgi.com/Press/itPress/2003/6.25.03

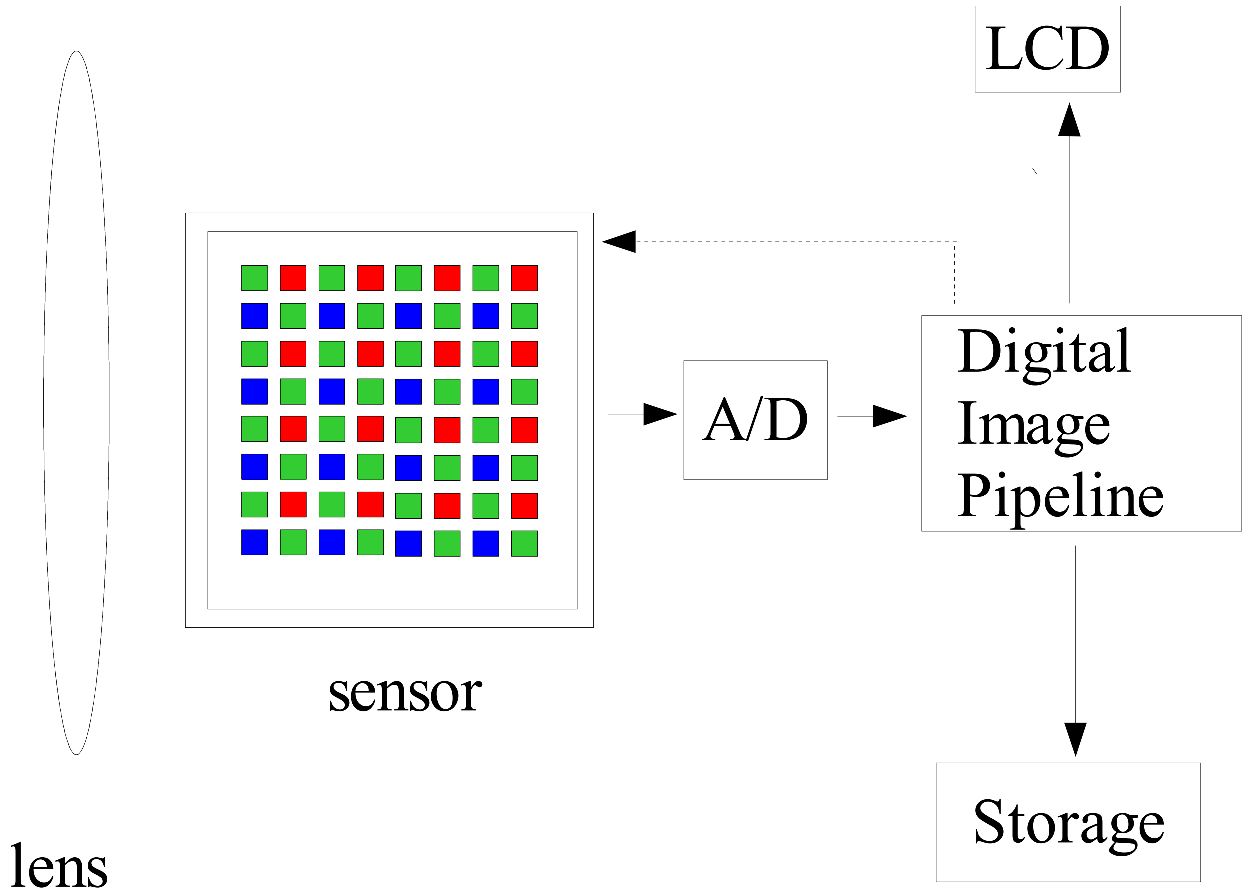
Embedded DSCs

Cellphone cameras appeal to
- o phone makers
- o wireless operators
- o most new designs

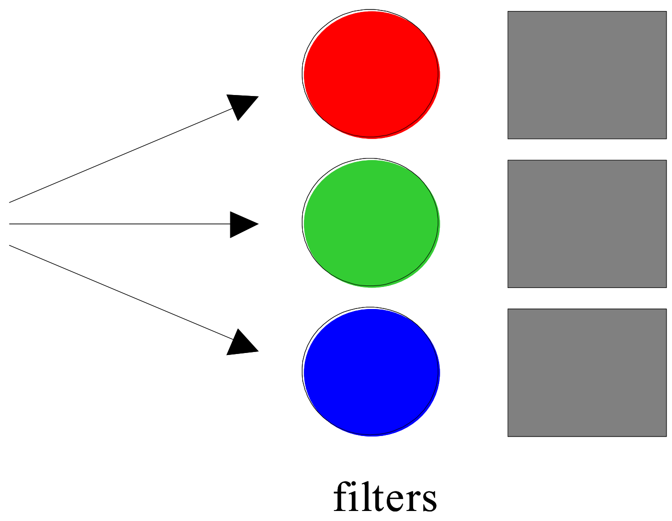
What are their uses?
- o journalism
- o “always-with-you” camera
- o security

Webcams: a lesson

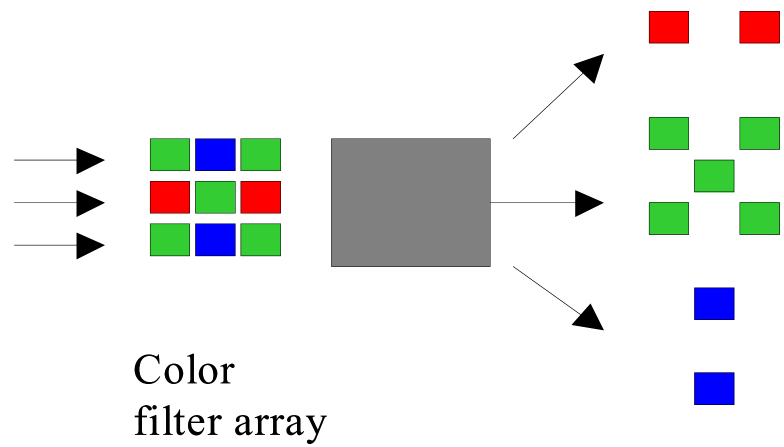
"Anatomy" of digital still camera



Sensor layout



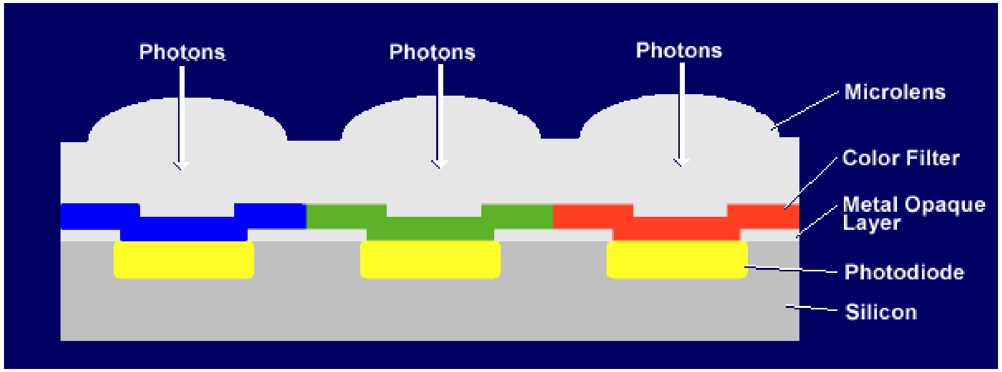
Three sensors:
Expensive, bulky



One sensor:
Requires interpolation

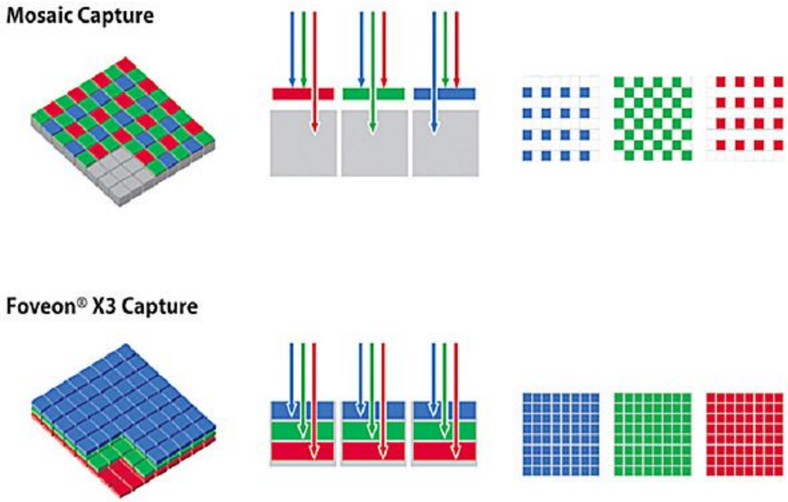
Sensor layout variations

microlenses

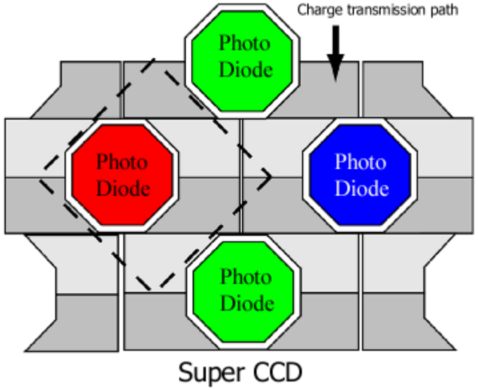


micron.com/products/imaging/technology/pixel.html

3 color pixels

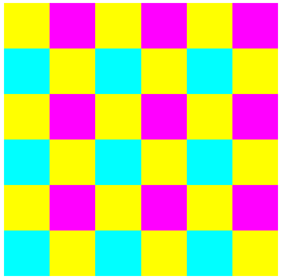


www.x3f.info



Hexagonal mosaic

home.fujifilm.com



CMY

www.dpreview.com

Digital camera image pipeline

Processing stages after A/D

a) Photo-response non-uniformity (PRNU)

b) White balance

c) Color interpolation

d) Enhancements: sharpen, noise-clean

e) Color correction

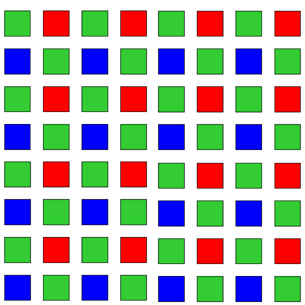
f) Tone mapping

Response correction

- Illumination variation : cos^4
  - off-axis θ illumination reduced by cos(θ)^4
  - removable by calibration, spatial gain
- Bad pixel correction
  - "Stuck-on", "stuck-off" pixels
  - "hot" pixels visible in low-light/long exposure

Bad pixel removal

- Simple: thresholding, median filtering
- Sequential analysis (Tan & Acharya,ICASSP '99)



Algorithm

a) Calculate in neighboorhood

$$S = min|X - Xn|$$

b) if S > T consistently ==> **bad pixel**

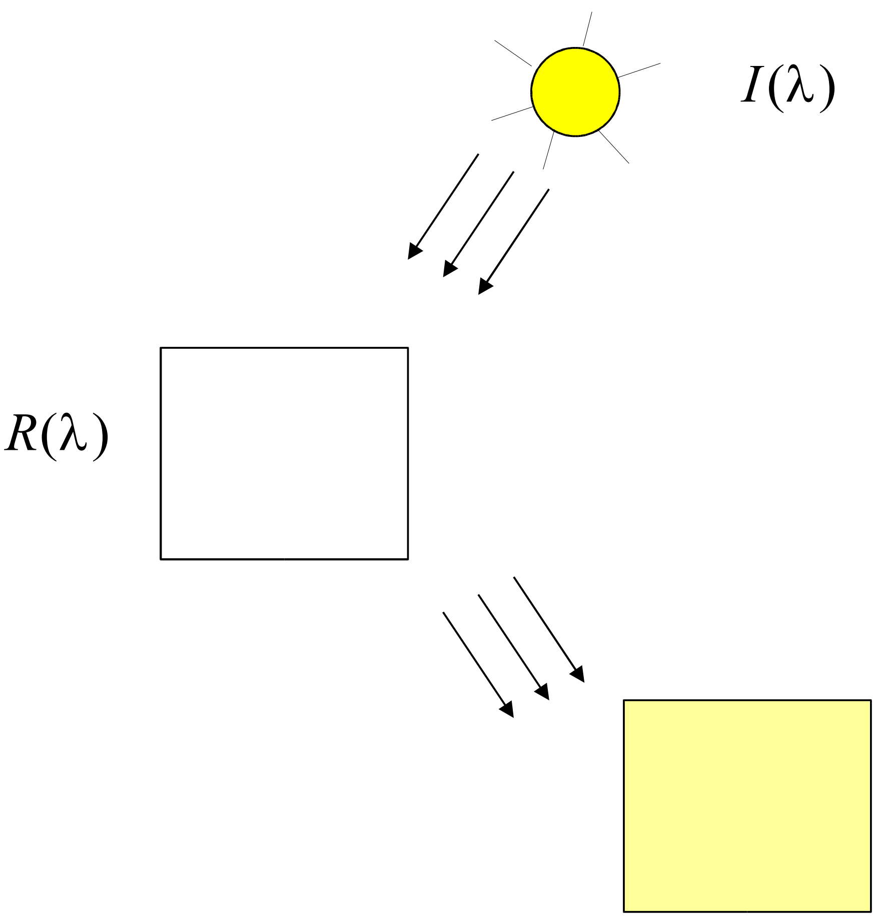
r.kakarala@ieee.org

Bad pixel correction

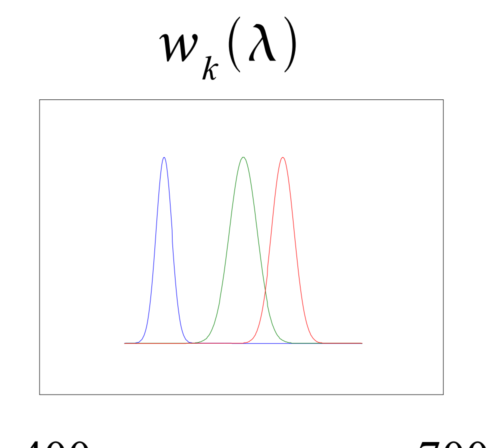
www.mediachance.com/dce/

r.kakarala@ieee.org

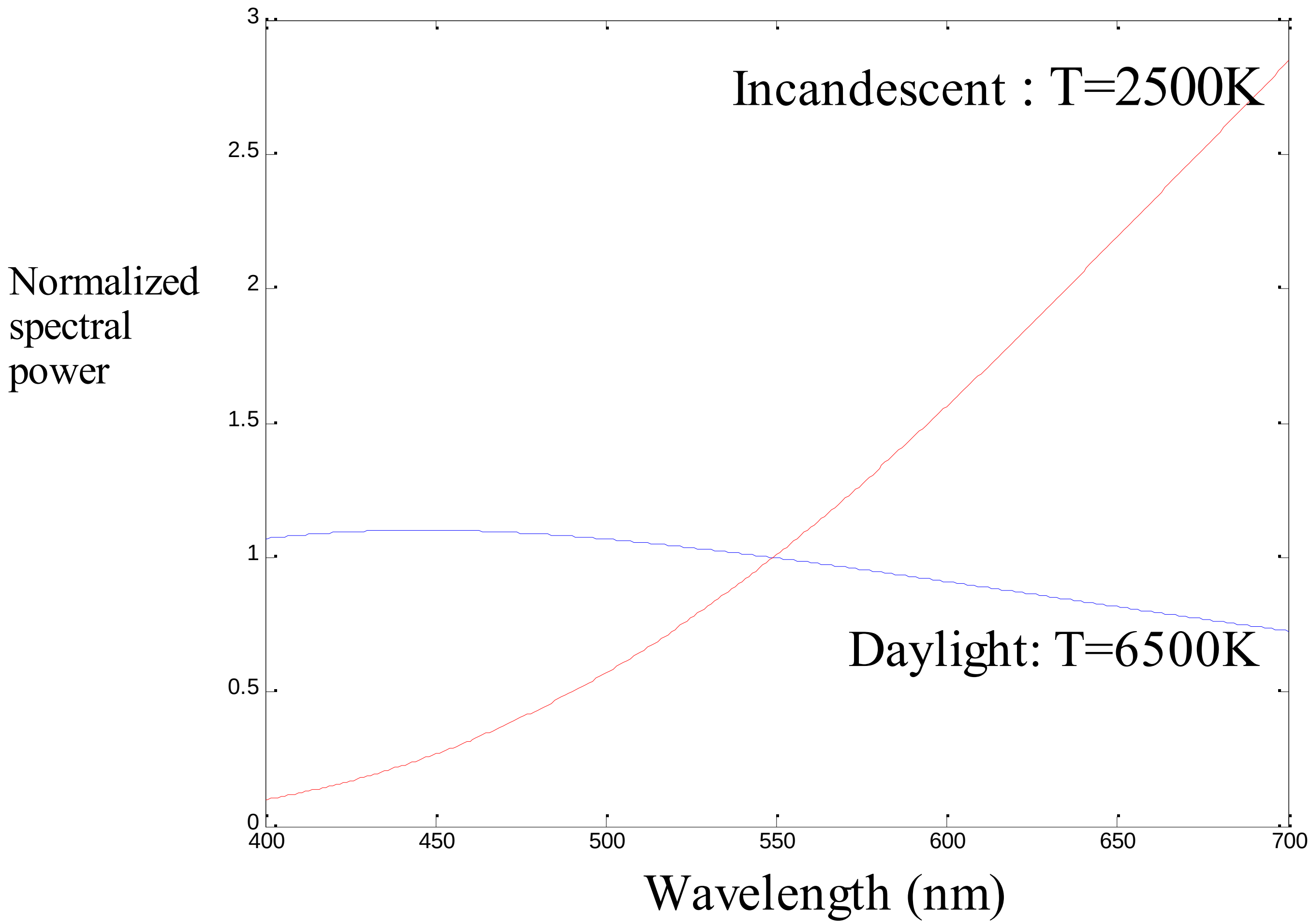
Color imaging



$$s_k = \int I(\lambda) R(\lambda) w_k(\lambda) d\lambda,$$
$$k = 1,2,3$$



Color temperature



$$I_B(\lambda) = \frac{c_1}{\lambda^5} * \left[\exp\left(\frac{c_2}{\lambda * T}\right) - 1 \right]^{-1}$$

r.kakarala@ieee.org

White balance

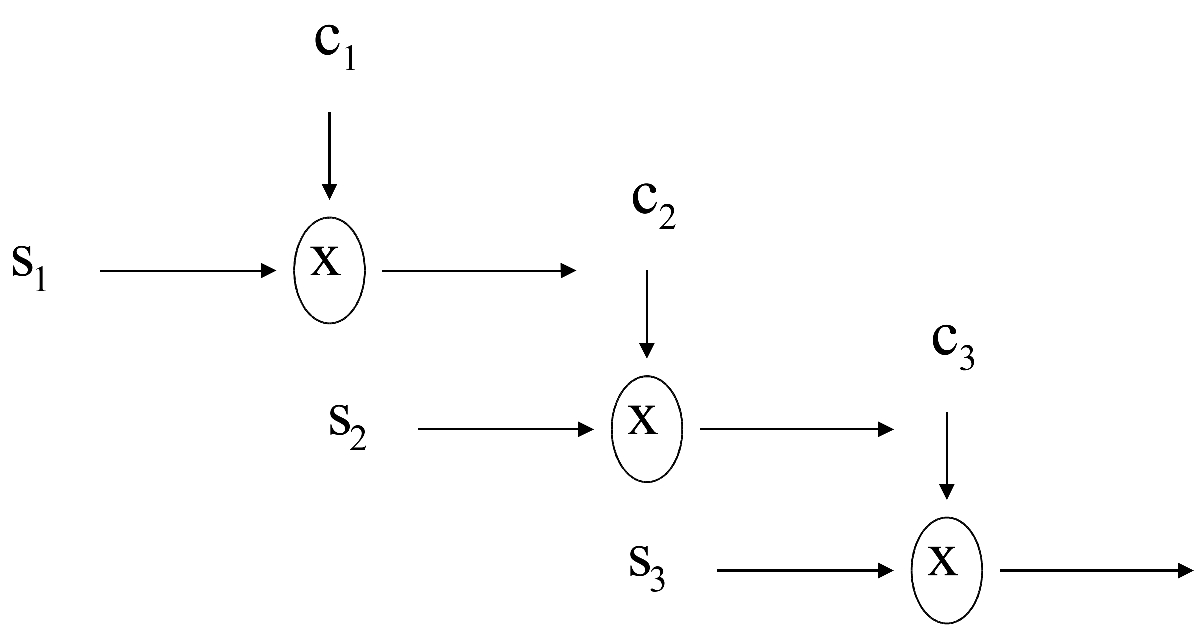
Camera set for incandescent

flourescent

Nikon Coolpix 990: Source Prof. C K Shene

White balancing

Choose gains c1, c2, c3 to compensate for illuminant



Auto white balance

Many algorithms:

- Gray world: mean R, G, B values should be equal
- Correlation method (Tominaga-Wandell)

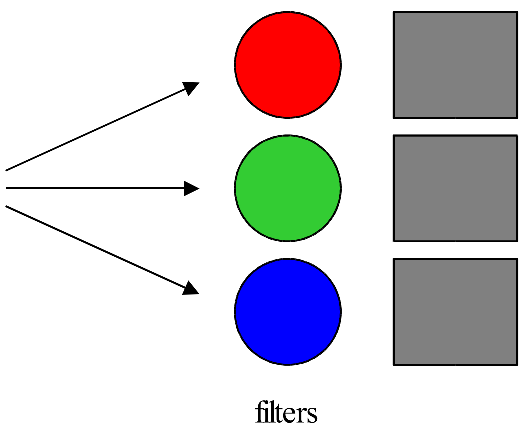
  Convex hull of R,B values matched to gamut of illuminant color temperature.

Real-world problems

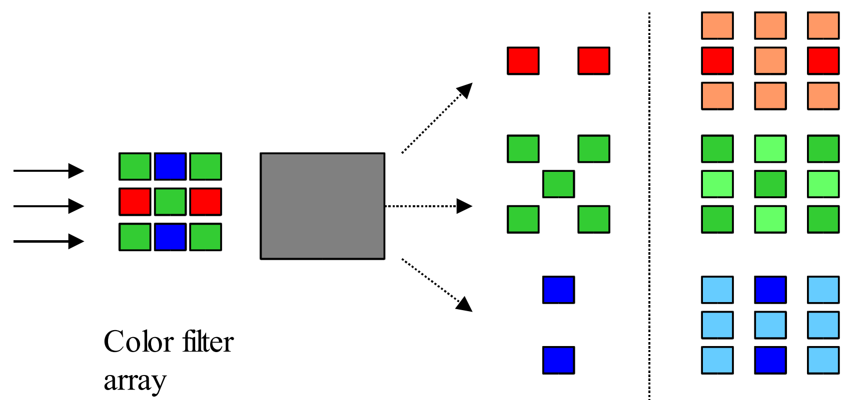
- Large uniform patches of color
- Multiple illuminants
- Non standard bulbs

Demosaicking

- Color from monochromatic sensors: two methods



Three sensors:
Expensive, bulky



One sensor:
Requires interpolation

"Demosaiced" color image

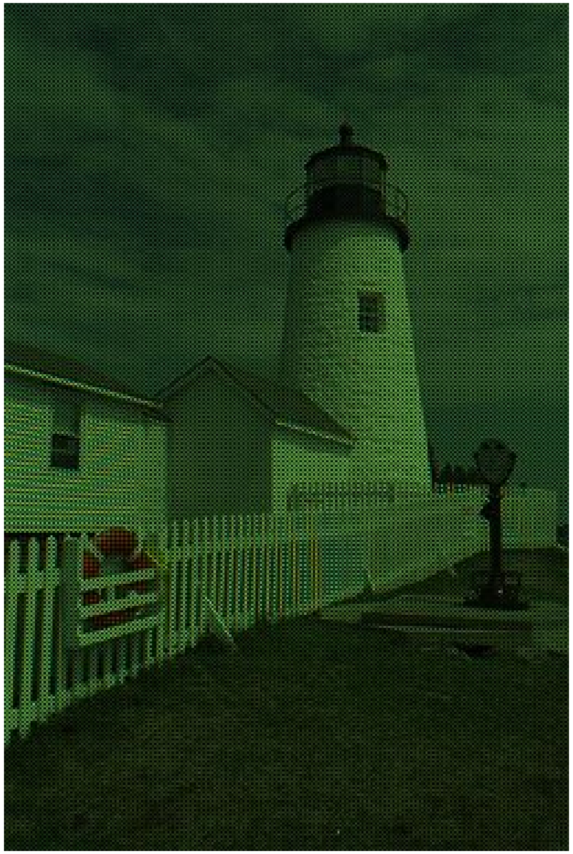
Research topics:
(a) interpolation algorithms
(b)color filter array patterns

r.kakarala@ieee.org

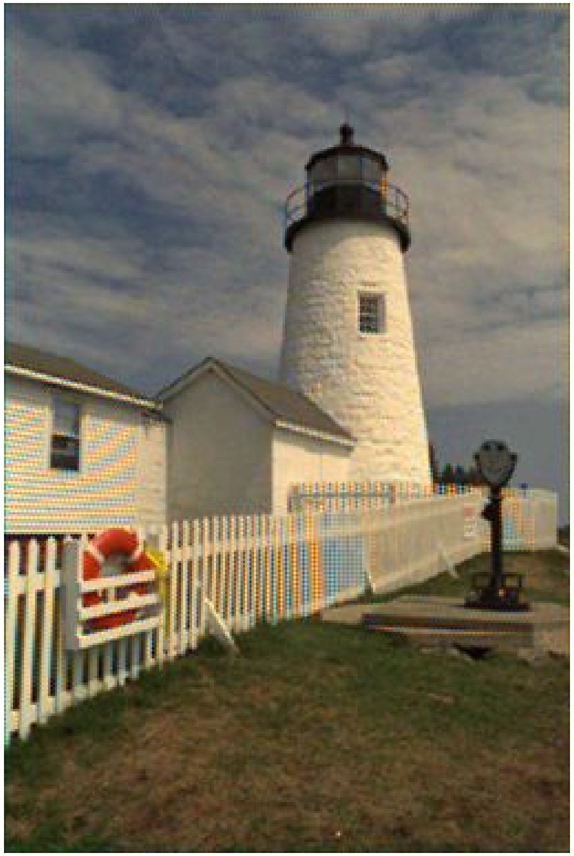
Demosaicking

Original

What the sensor sees: mosaiced image



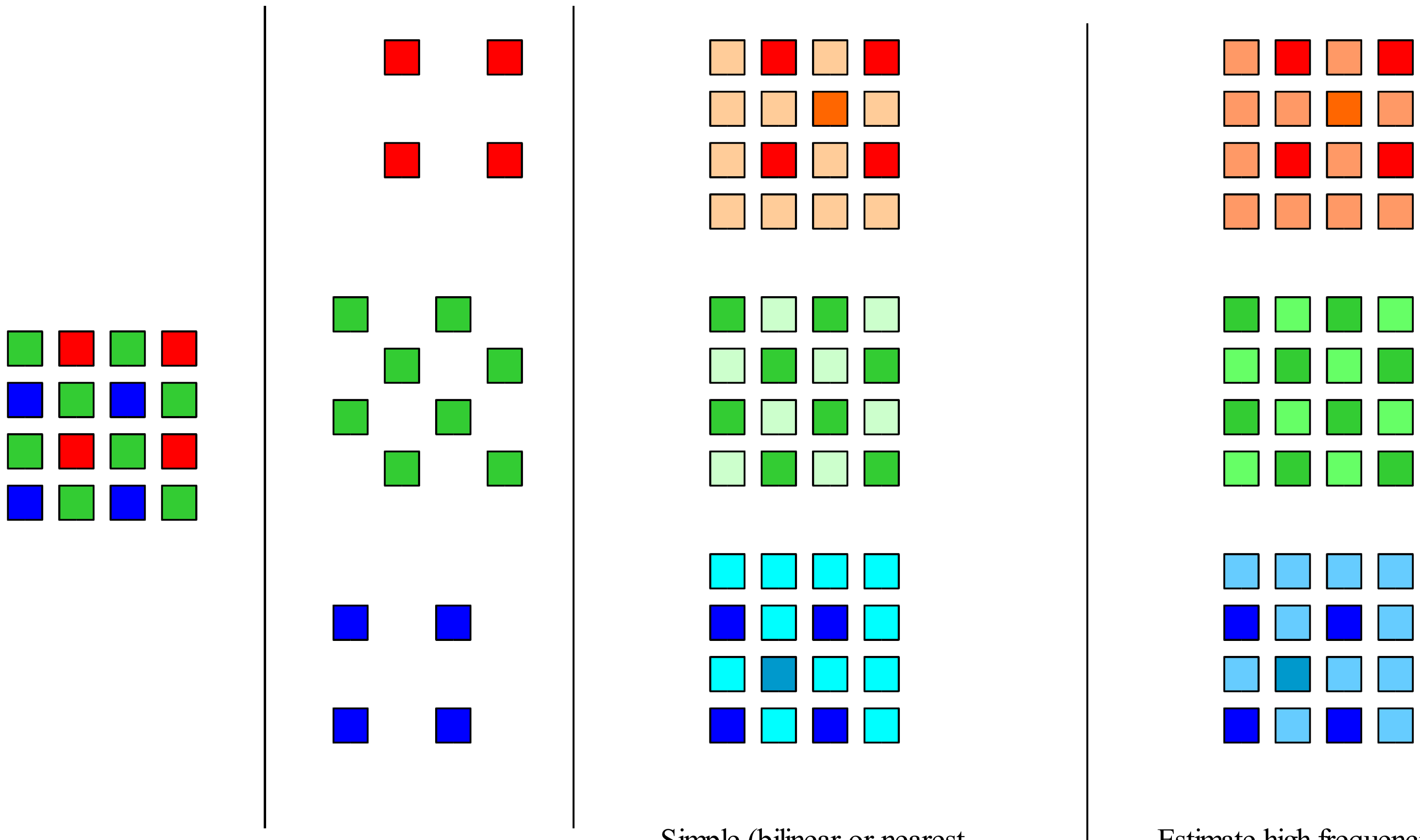
Demosaiced image: note color aliasing



Bilinear interpolation

r.kakarala@ieee.org

Demosaicking algorithms



Method:
G=B-Bav+Gav

Simple (bilinear or nearest neighbour) interpolation

Estimate high frequencies and incorporate original

PVM (Kakarala-Baharav)

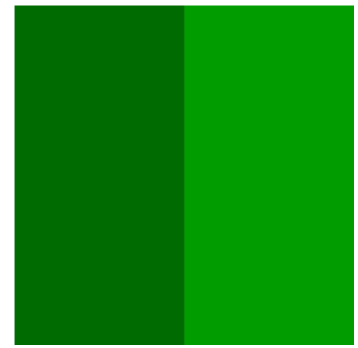
Hamilton-Adams (kodak)

r.kakarala@ieee.org

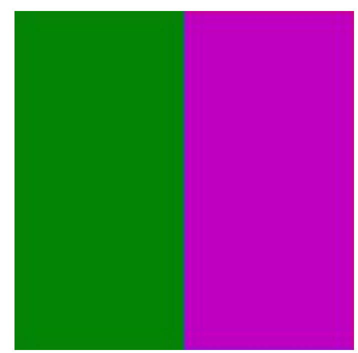
Idea: detect edges in color

- Edges: locations of sharp changes--they occur in luminance & chrominance

- Always interpolate along edges, rather than across them



Luminance edge



Chrominance edge (isoluminant)

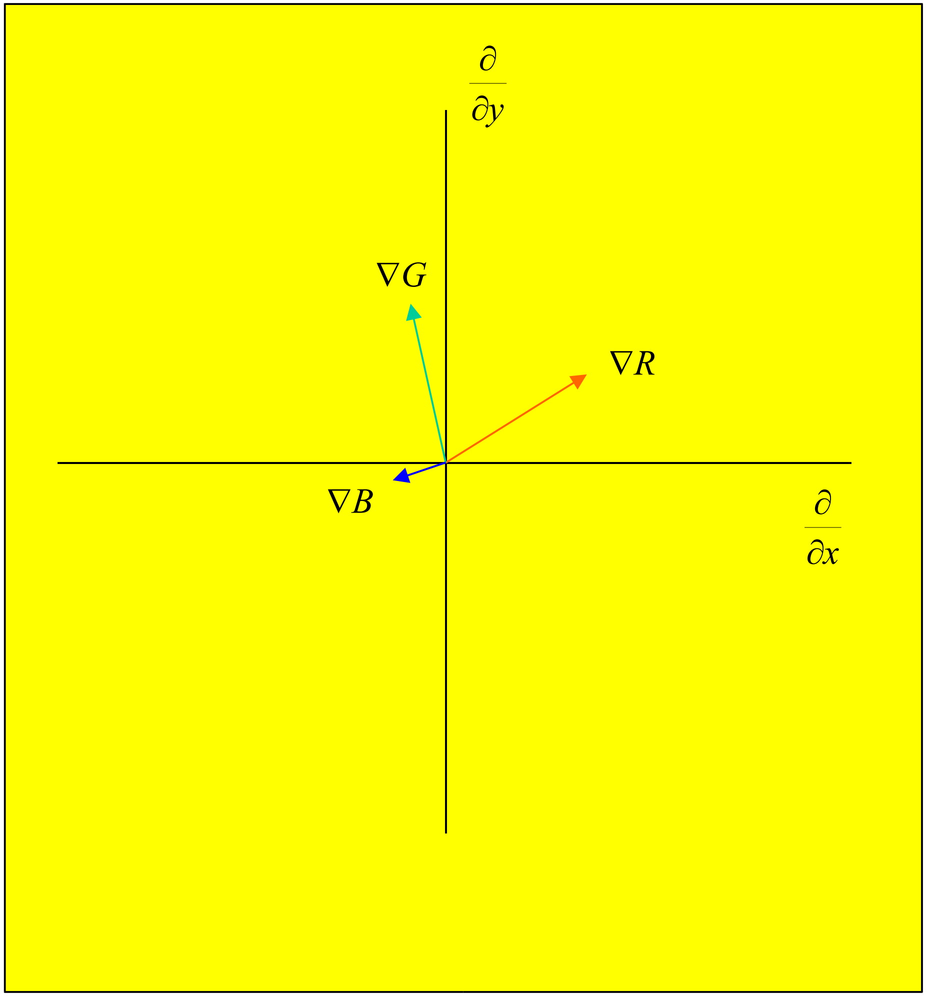
Detecting color edges

- In monochrome images, use gradient

$$\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \qquad \|\nabla I\| > T \Rightarrow edge$$

- In color images, 3 gradients

$$\nabla R \quad \nabla G \quad \nabla B$$



r.kakarala@ieee.org

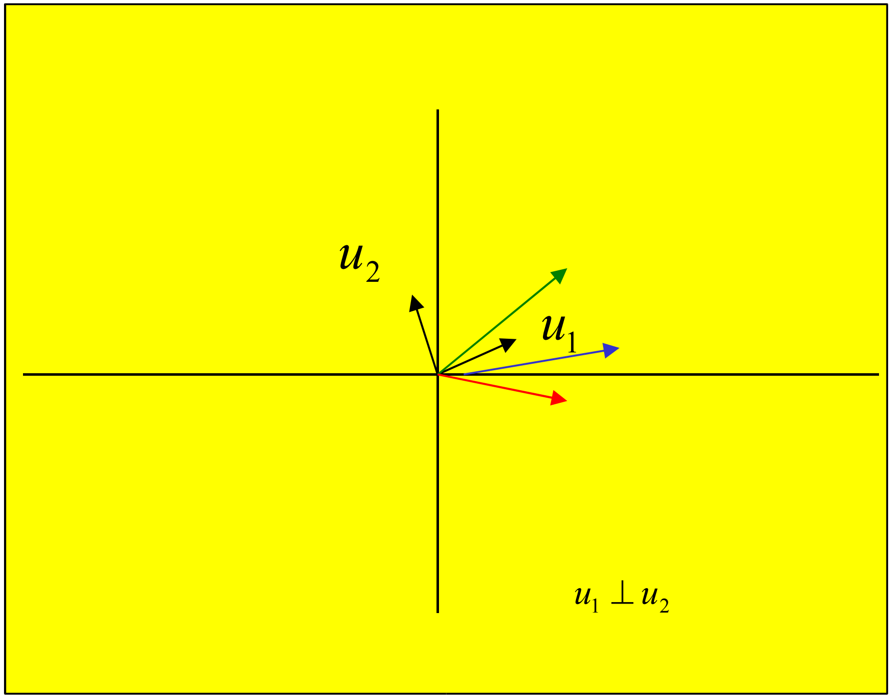
Interpreting color gradients

- Form Jacobian

$$J = \begin{bmatrix} \partial R/\partial x & \partial G/\partial x & \partial B/\partial x \\ \partial R/\partial y & \partial G/\partial y & \partial B/\partial y \end{bmatrix} = [\nabla R \quad \nabla G \quad \nabla B]$$

- SVD of Jacobian

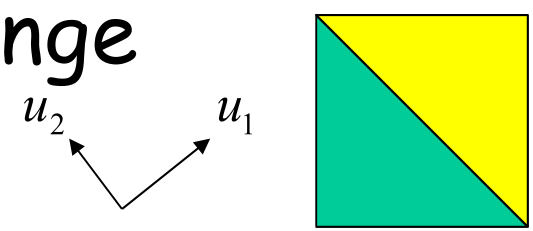
$$J = USV^t = \sigma_1 u_1 v_1^t + \sigma_2 u_2 v_2^t \approx \sigma_1 u_1 v_1^t$$



- U_1 = principal vector; U_2 = residual vector

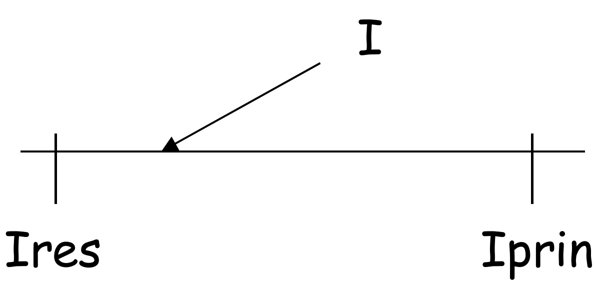
Principal & residual vectors

- Principal vector -> steepest change; residual vector -> least change



- Interpolate along both vectors + add

$$I = \frac{\sigma_1}{\sigma_1 + \sigma_2} I_{res} + \frac{\sigma_2}{\sigma_1 + \sigma_2} I_{prin}$$



- Interpolation along residual, principal vectors not usually possible
  - -- replace with horizontal, vertical interpolation

- Scheme requires SVD
  - -- develop simple approximations

Simple interpolation

- Add horizontal & vertical interpolants

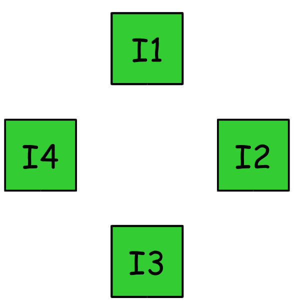
$$I = \lambda I_{res} + (1-\lambda) I_{prin}$$

$$\Updownarrow$$

$$I = \alpha I_{horiz} + (1-\alpha) I_{vert}$$

$$I_{horiz} = \frac{(I_2 + I_4)}{2}$$

$$I_{vert} = \frac{(I_1 + I_3)}{2}$$



Choose α based on principal vector

Simple interpolation (cont'd)

- Interpolation rule

$$I = \alpha I_{horiz} + (1-\alpha) I_{vert}$$

- Quantize so that $\alpha \in \left\{ 0, 1, \frac{1}{2} \right\}$

Choosing quantized weight

Form Jacobian

$$J = \begin{bmatrix} \partial R/\partial x & \partial G/\partial x & \partial B/\partial x \\ \partial R/\partial y & \partial G/\partial y & \partial B/\partial y \end{bmatrix} = \begin{bmatrix} \nabla R & \nabla G & \nabla B \end{bmatrix}$$

Count # of times $|\partial y| > |\partial x|$

Assignment: if (0) or (1) $\alpha = 0$ (vertical)

if (2) or (3) $\alpha = 1$ (horiz)

"Majority Rule"

Additional case

- Add components of Jacobian

$$\|J\| = \|\nabla R\| + \|\nabla G\| + \|\nabla B\|$$

- If $\|J\| < T \Rightarrow \alpha = \frac{1}{2}$ (average horiz. and vertical)

  else use previous rule to set $\alpha \in \{0,1\}$

Why majority rule?

- Good predictor of principal vector

$$J = USV^t = \sigma_1 u_1 v_1^t + \sigma_2 u_2 v_2^t \approx \sigma_1 u_1 v_1^t$$

majority |dy|>|dx| --> |u1(2)|>|u1(1)|?

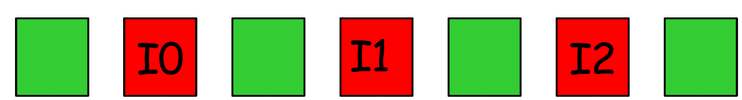
accurate 91 % of time (1000 trials)

Why majority rule? (cont'd)

- Allows mix of 1st and 2nd derivatives

$$\tilde{J} = \begin{bmatrix} \partial^n R / \partial x^n & \partial G / \partial x & \partial^m B / \partial x^m \\ \partial^n R / \partial y^n & \partial G / \partial y & \partial^m B / \partial y^m \end{bmatrix}; n, m = 1,2$$

- majority |dy^n|>|dx^n|
- both 1st and 2nd derivs can be estimated in some locations, e.g.,



dx = I2-I0; dx^2 = 2I1-I2-I0

r.kakarala@ieee.org

PVM



Hamilton-Adams (kodak)



r.kakarala@ieee.org

Noise cleaning

Many algorithms:

$$\begin{array}{ccc} X_1 & X_2 & X_3 \\ X_4 & X_c & X_6 \\ X_7 & X_8 & X_9 \end{array}$$

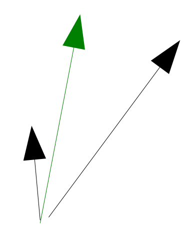
1) Sigma filtering (J. S. Lee, 1982)

$$X = \text{mean}\{Xi: |Xi-Xc| < \sigma\}$$

2) Global and local σ estimates (Kodak patents)

3) Vector median of a set: smallest total distance

$$VM = argmin_w \left\{ X_w : \sum_{k=1}^{9} \left\| X_k - X_w \right\| \right\}$$



After market noise cleaners

r.kakarala@ieee.org

www.michaelalmond.com/Articles/noise

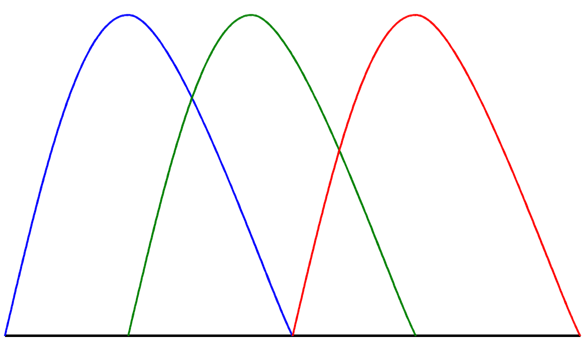
r.kakarala@ieee.org

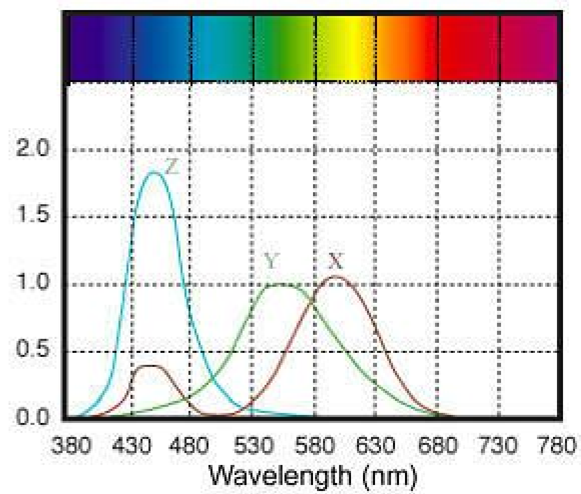
www.michaelalmond.com/Articles/noise

r.kakarala@ieee.org

Color correction

Sensor response → CIE XYZ (device independent) → sRGB (display)





$$\begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

http://www.creativepro.com/story/feature/13036-2.html

r.kakarala@ieee.org

Color correction: Sensor to XYZ

"Maximum ignorance" method

Solve: $W = ST \rightarrow T = pinv(S)W$

W = n x 3 XYZ responses
S = n x 3 sensor RGB responses
T = 3x3 transformation matrix

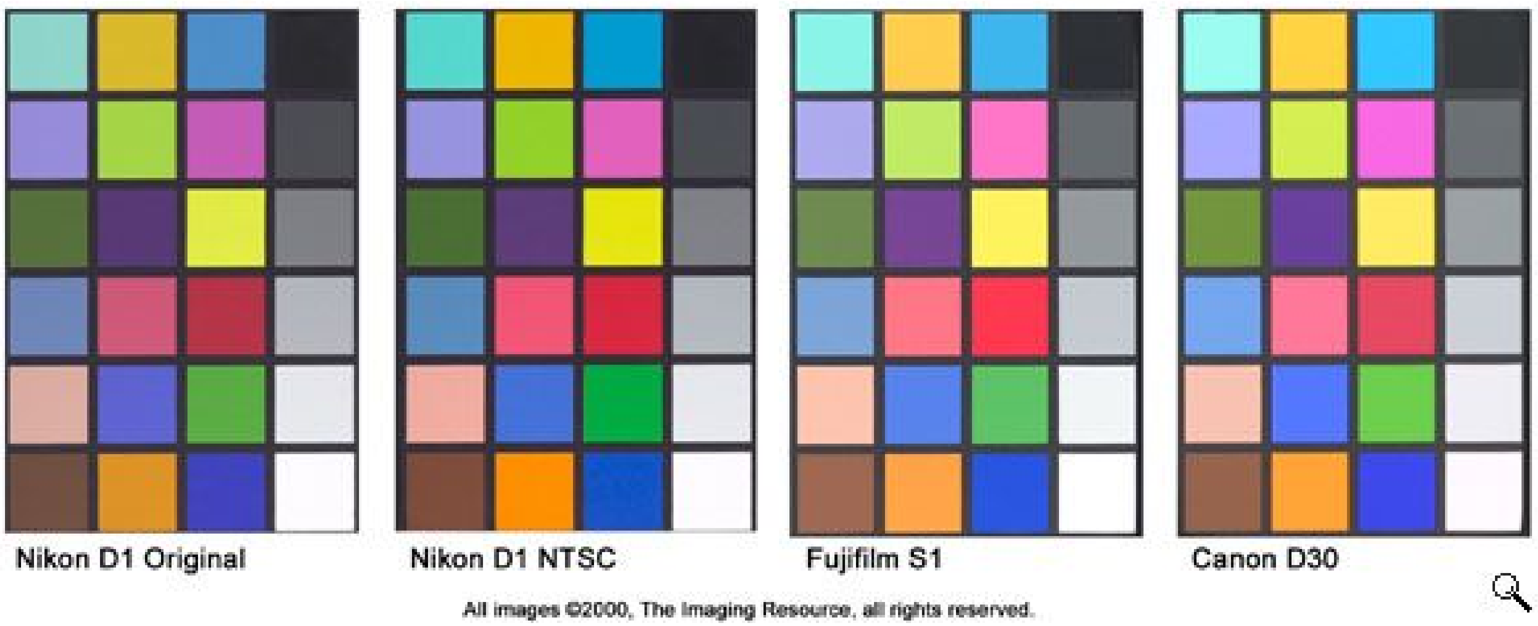
Surface based (Vrhel-Trussell)

Bayesian (Brainard-Freeman)

White-point preserving (Finlayson-Drew)

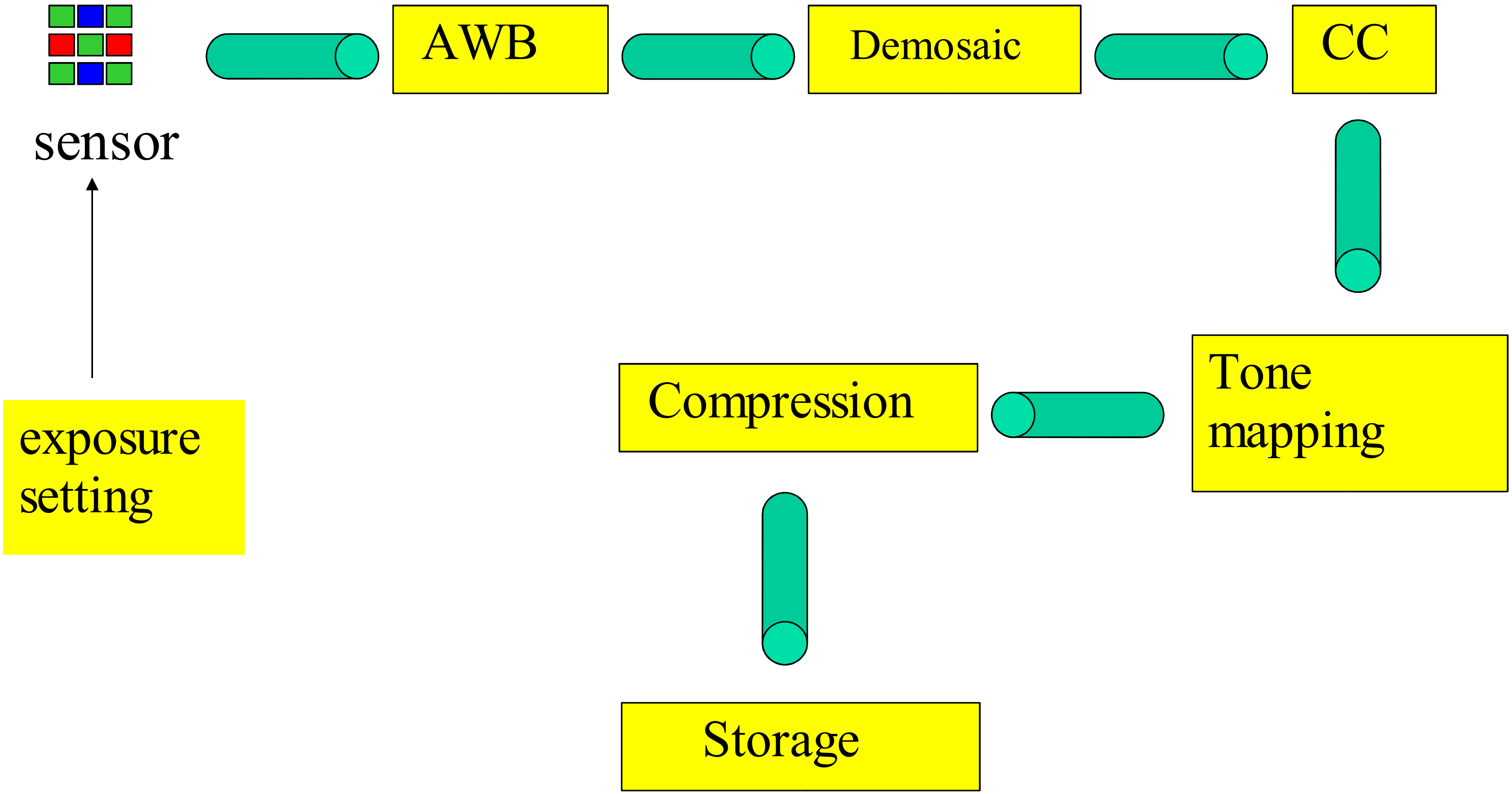
r.kakarala@ieee.org

A comparison of 3 consumer digital cameras



http://www.imaging-resource.com/PRODS/D30/3TPICS1.HTM

A Digital Camera Image Pipeline



Tone mapping

Gamma correction

Display response

$$y = x^{\gamma}$$

Compensation

$$x = s^{1/\gamma}$$

Typical value $\gamma = 2.2$

Tone mapping: best use of display dynamic range

Ex: Nikon D1 has 12 bits/pixel ---> Display is 8 bits.

Tone mapping methods

Simple: Histogram equalization

Others: Adaptive logarithmic (Drago, et al 2003)

display $L_d \leftarrow L_w$ world

$$L_d = \frac{0.01 * L_{dmax}}{\log_{10}(L_{wmax}+1)} \frac{\ln(L_w+1)}{\ln\left(2+8\left(\frac{L_w}{L_{wmax}}\right)^{\frac{\ln(b)}{\ln(0.5)}}\right)}$$

r.kakarala@ieee.org

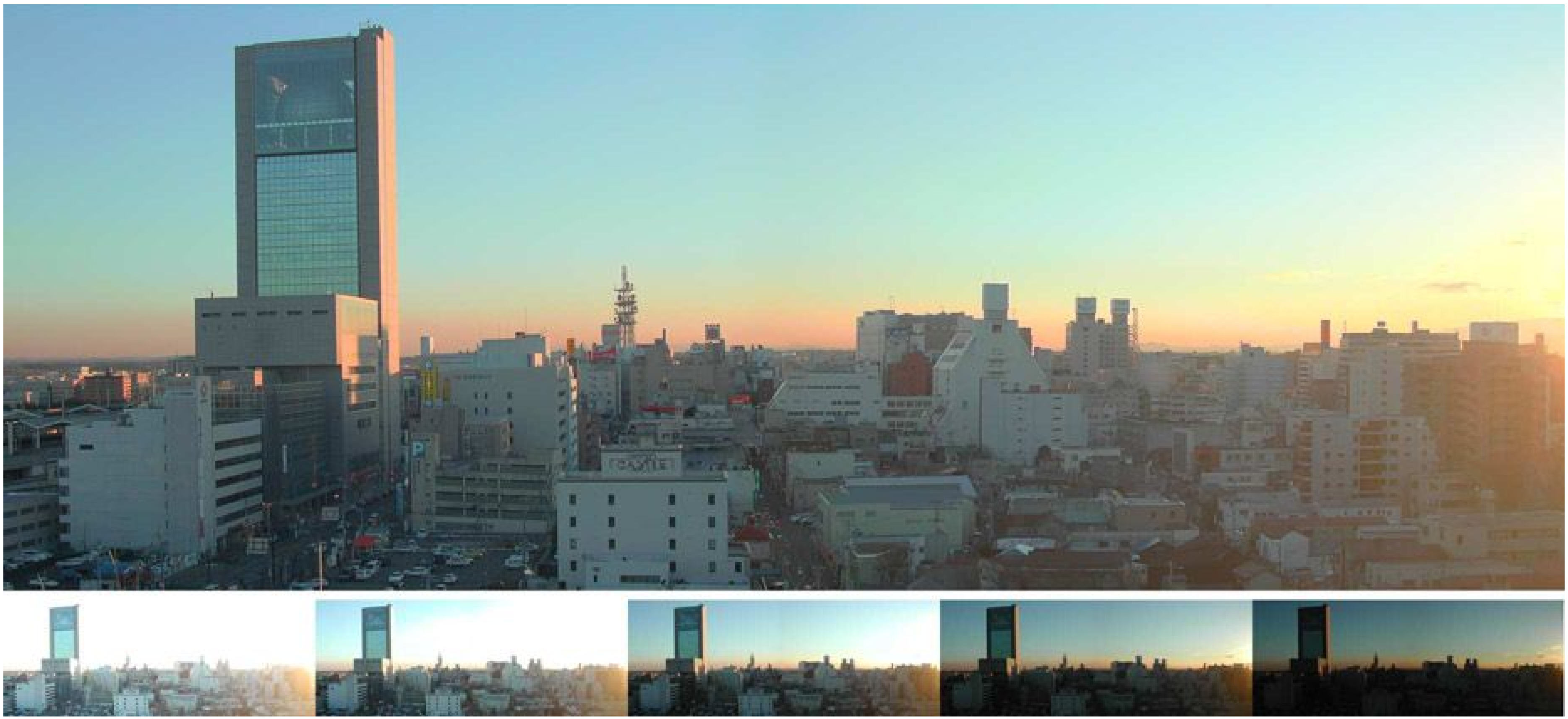
before tone map

r.kakarala@ieee.org

after tone map

r.kakarala@ieee.org

Fusion of multiple exposure images (Drago et al)



"Frontier" Areas

More stages in pipeline

- Image fusion (focus and exposure)
- Optical distortion removal (narrower cameras)

Host vs embedded implementation

A standard pipeline?