Photographic Technology

PhotoTechEDU series

Lecture 06: Feb. 28, 2007

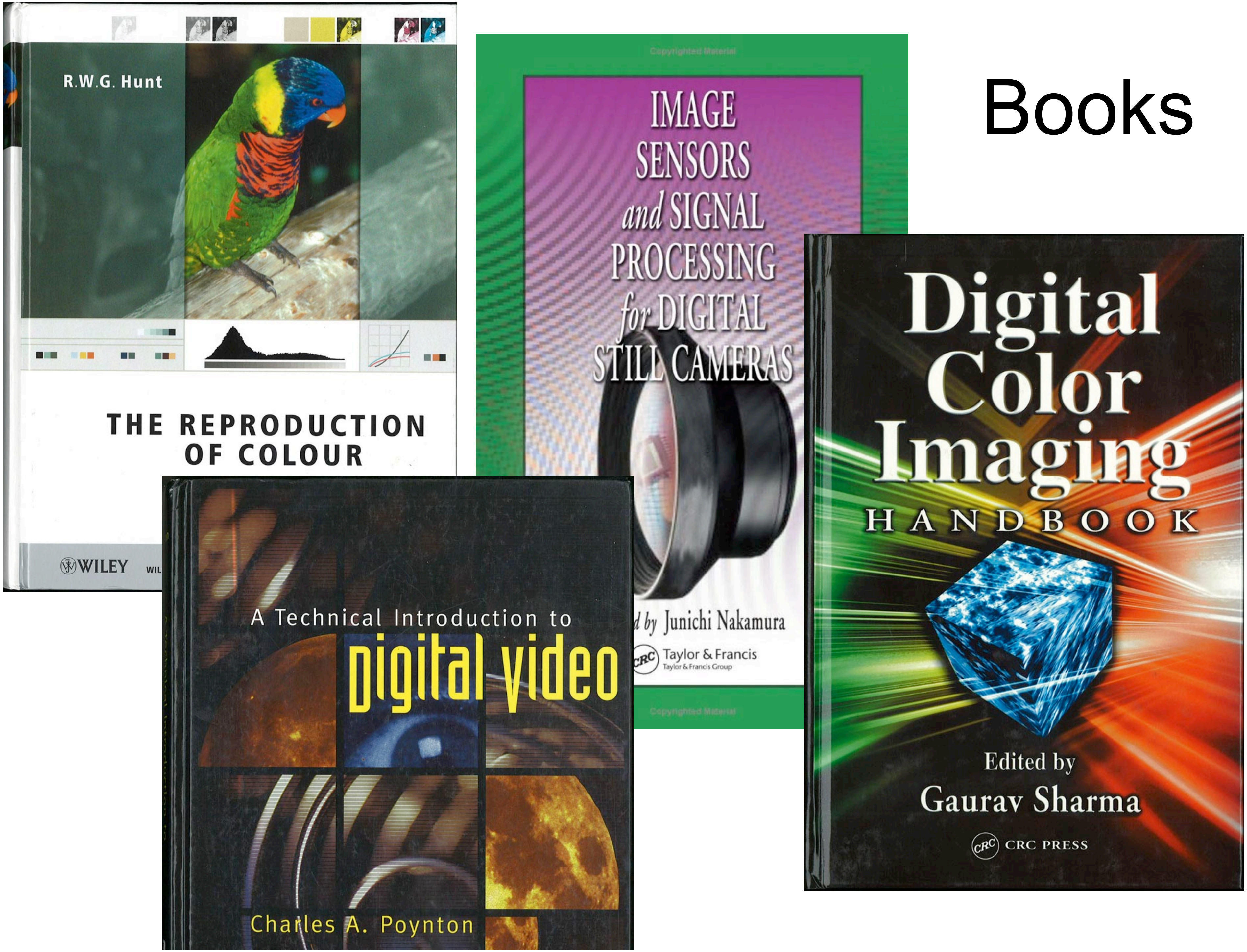
Digital camera image-processing pipelines

Richard F. Lyon

Google Research

dicklyon@google.com

Books


R.W.G. Hunt
THE REPRODUCTION OF COLOUR
WILEY
IMAGE SENSORS and SIGNAL PROCESSING for DIGITAL STILL CAMERAS
d by Junichi Nakamura
CRC Taylor & Francis
Taylor & Francis Group
Digital Color Imaging HANDBOOK
Edited by
Gaurav Sharma
CRC PRESS
A Technical Introduction to
digital video
Charles A. Poynton

chapter twelve

Color image processing for digital cameras

Ken Parulski
Kevin Spaulding
Eastman Kodak Company

Contents

0-8493-0900-X/03/$0.00+$1.50

Ken Parulsky and Kevin Spaulding chapter in Sharma book — a great source of details

Simplified pipelines
(Parulski & Spaulding)

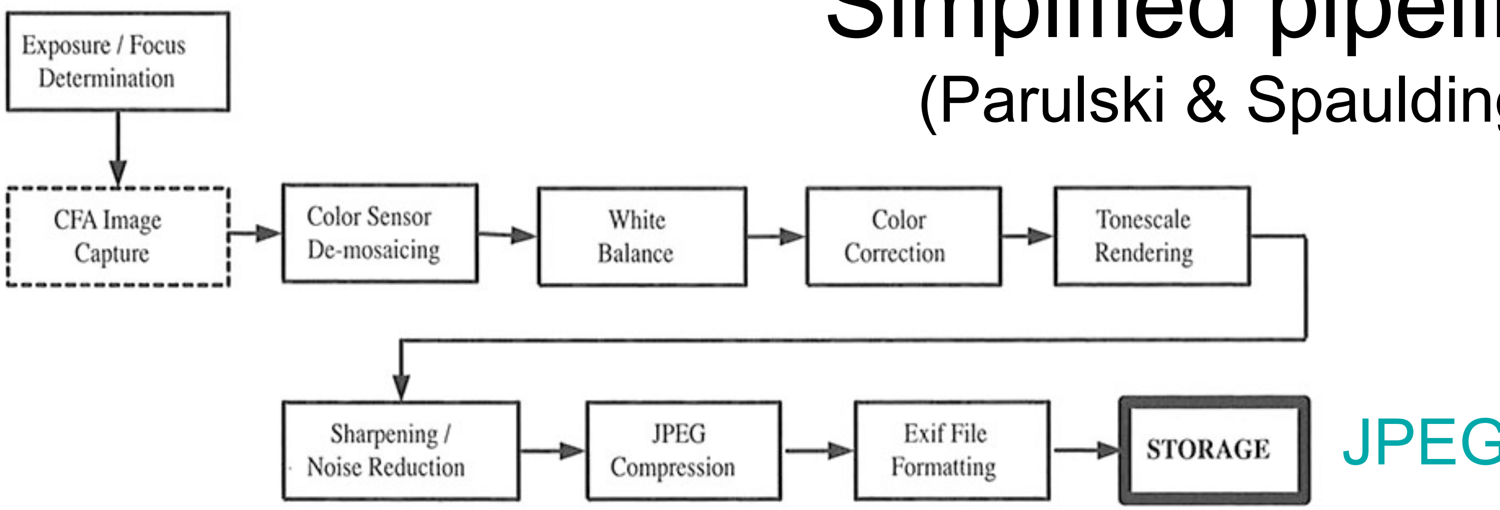


Figure 12.3 Example final still image processing flow in a consumer camera.

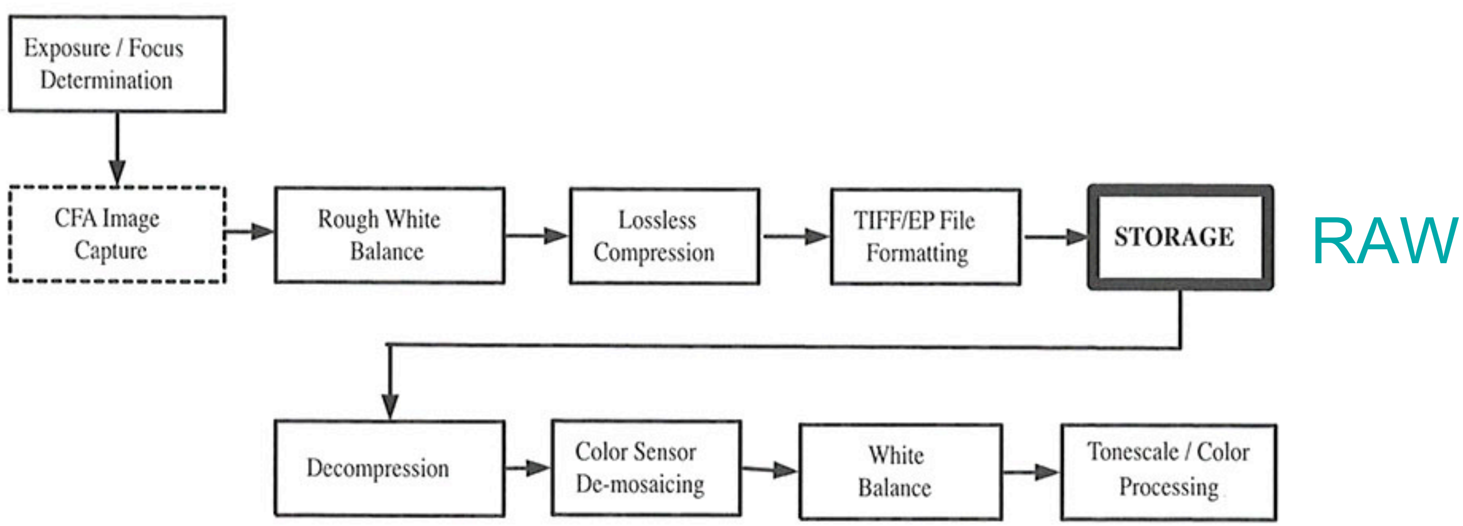


Figure 12.4 Example professional camera workflow.

On-chip Amplifier

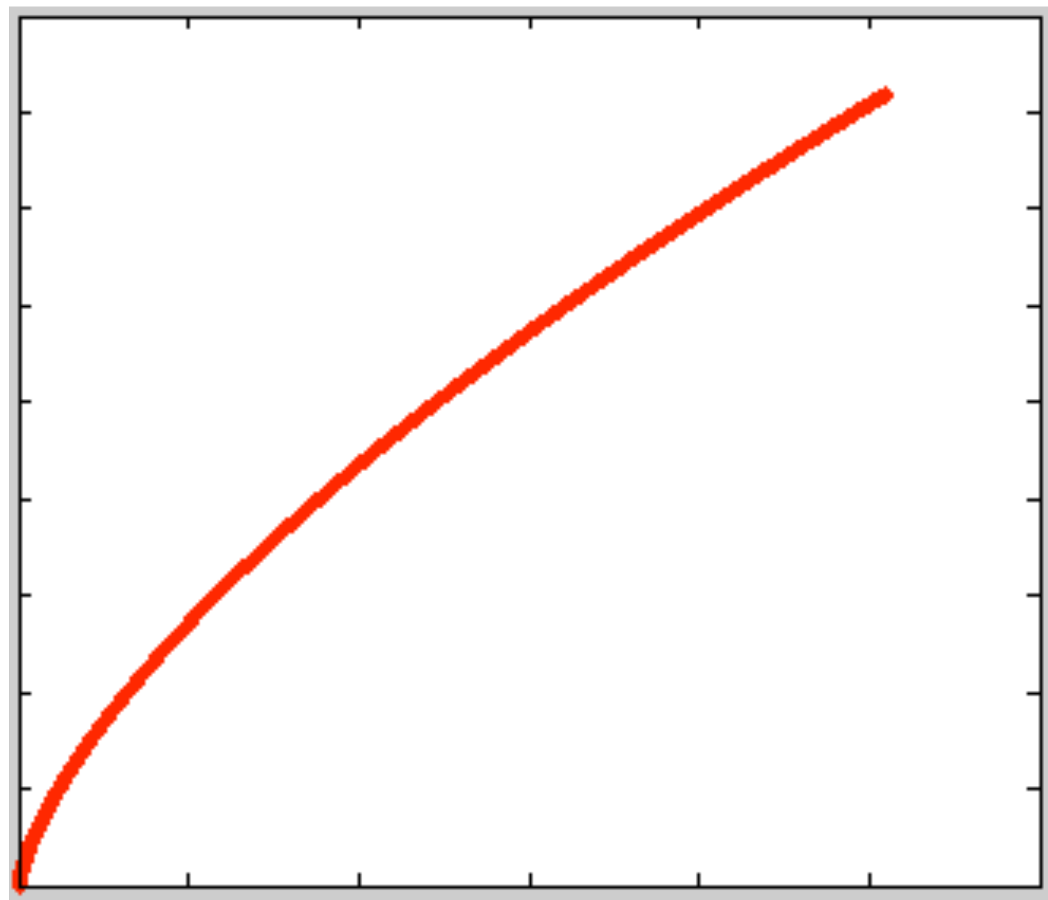
- Optional correlated double sampling (CDS) (in column amps for CMOS)
- Possibly multiple channels in parallel (2 sides, 3 colors, or more)
- Optional programmable-gain amplifier (PGA), possibly different gain per channel
- Optional external amplifier, too

Analog-to-Digital Conversion (ADC)

- Optional black clamp based on light-shielded edge pixels
- 10 to 14 bits, usually linear to 10 bits or better
- Optionally follow by linearization lookup table (LUT), or gamma LUT, or lossless or nearly-lossless encoding

Gamma curve at sensor raw data?

- Example gamma curve for reducing hi-bit data to 8-bit data, when noise floor is 30 e^- and full well is 58000 (max shot noise 240 e^-).
- Slope, or quantization step size, changes by a ratio near 8:1, to stay a constant fraction of noise.



Example/reference: **Programmable DSP Platform for Digital Still Cameras,**

Wissam Rabadi, Raj Talluri, Klaus Illgner, Jie Liang, Youngjun Yoo

http:focus.ti.com/lit/an/spra651/spra651.pdf

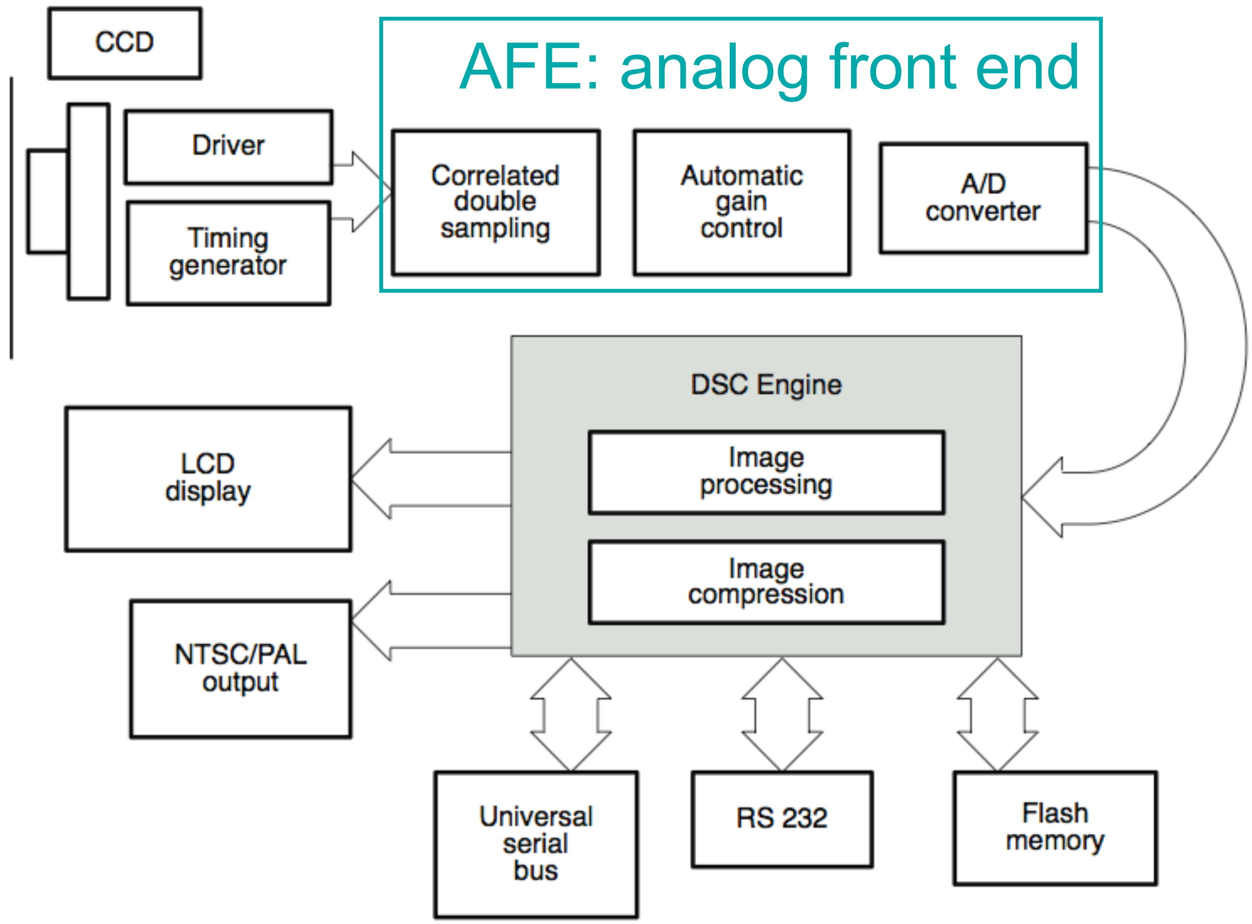


Figure 3. Digital Still Camera Block Diagram

Dark-level Adjustment

- Optional “black clamp” in ADC
- Dark-frame subtraction (usually for CMOS only, to remove fixed-pattern offsets of pixel transistors)
- Dark-level drift adjustment based on light-shielded edge pixels
- Flare (glare) estimation and subtraction (usually about 1% of average intensity)

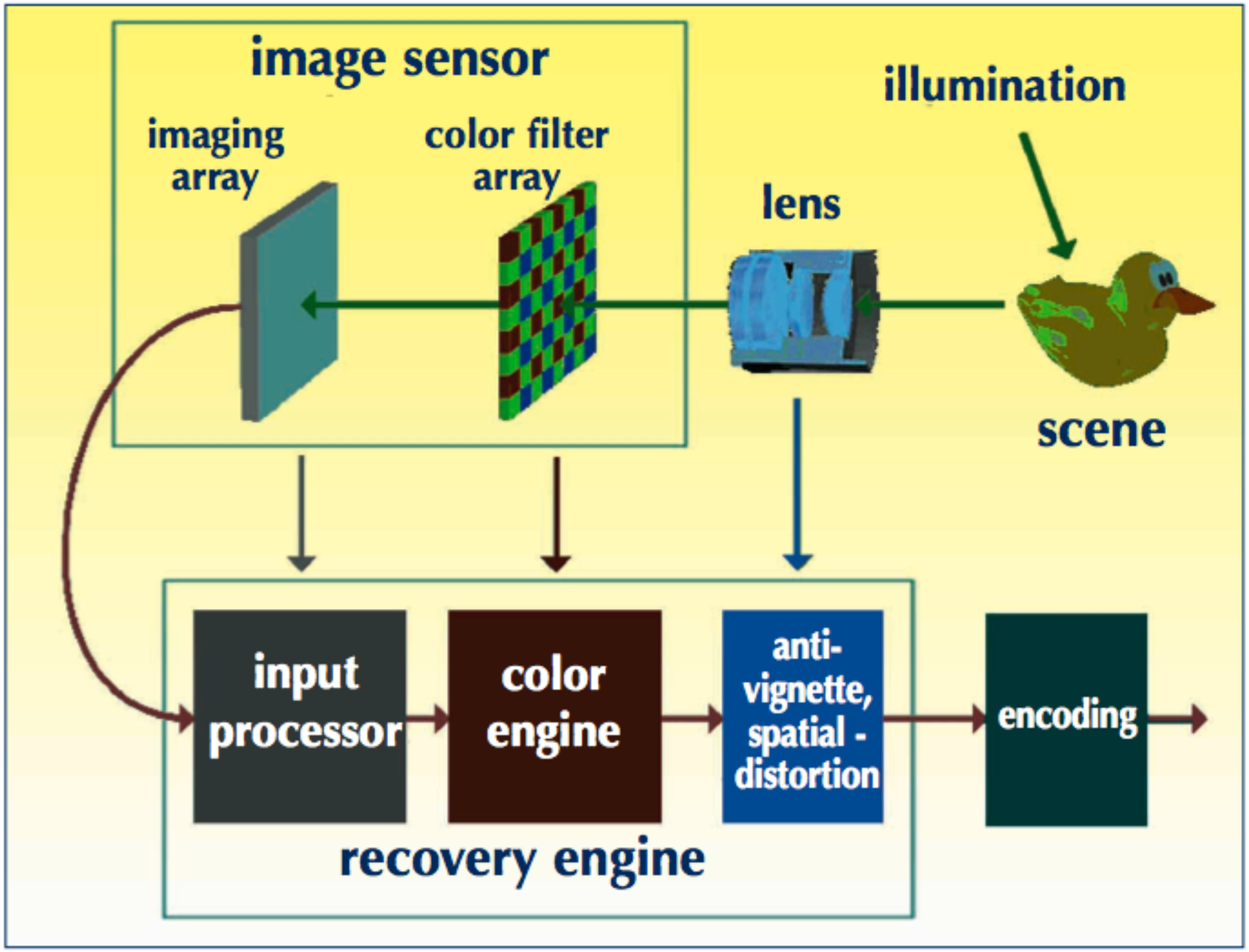
Spatial Gains (vignetting correction)

- Spatial gains per channel
  - 2D polynomial, or sparse table
- Account for lens, microlens, filter crosstalk effects
- => now numbers are proportional to scene light intensity, with black at zero

Example/reference: **An Introduction to the Digital Still Camera Technology,**

Massimo Mancuso Sebastiano Battiato

http://www.dmi.unict.it/~battiato/download/DSC1.pdf



White-point Estimation

- Either AWB estimate from image data (gray-world assumption ++), or
- Manual WB setting and calibration data
- Optionally apply channel-balancing gains now (but still keep track…)
- Optionally clip highlights to white

De-mosaic (color interpolation)

- Many complicated algorithm choices
- Sacrifice some chroma resolution for more luma resolution
- Optionally include gains for ISO speed or AE
- Optionally include highlight neutralization
- Respect edges

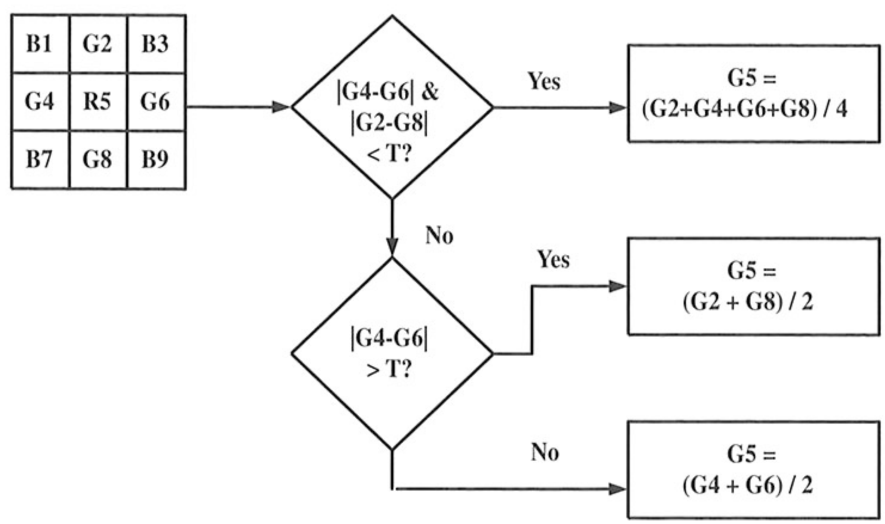


Figure 12.8 Example of an edge-sensing de-mosaicing algorithm for calculating the missing green values.

Compute and Apply Color Matrix

- 3x3 matrix transform of pixel RGB values
- Map white point of scene to white point of output space
- Preserve relative colors and preferred color rendering
- Use an intermediate space (e.g. LMS or RIMM RGB) for white-point adaptation
- Optionally also adjust color saturation

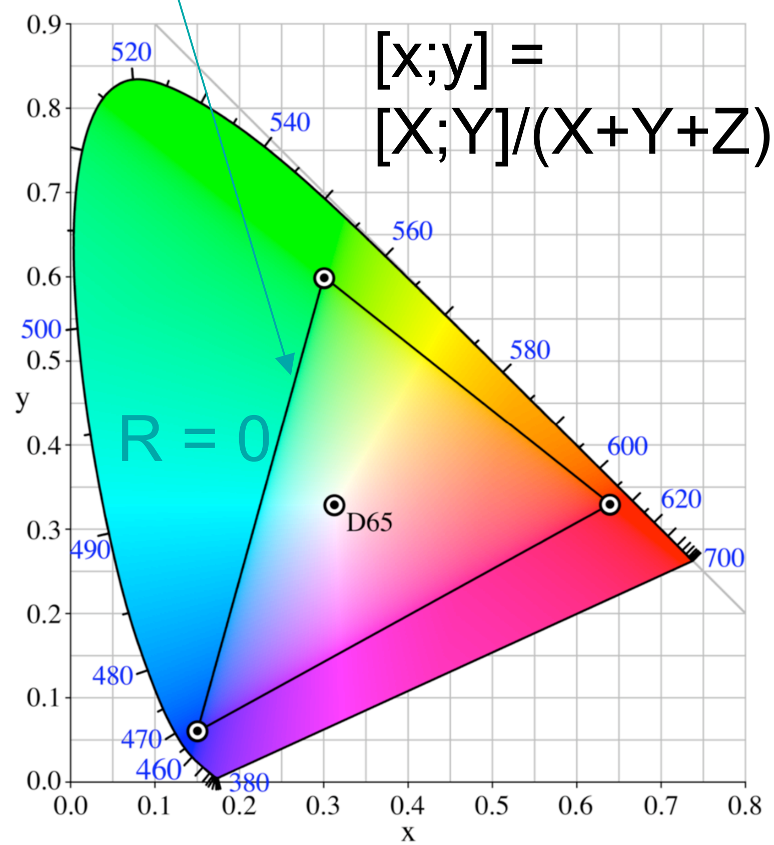
Matrix example: XYZ to sRGB

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

RGB neutral [1; 1; 1] comes from matrixing the color of a D65 illuminant, at XYZ of [0.9504; 0.9999; 1.0891] (a somewhat bluish white).

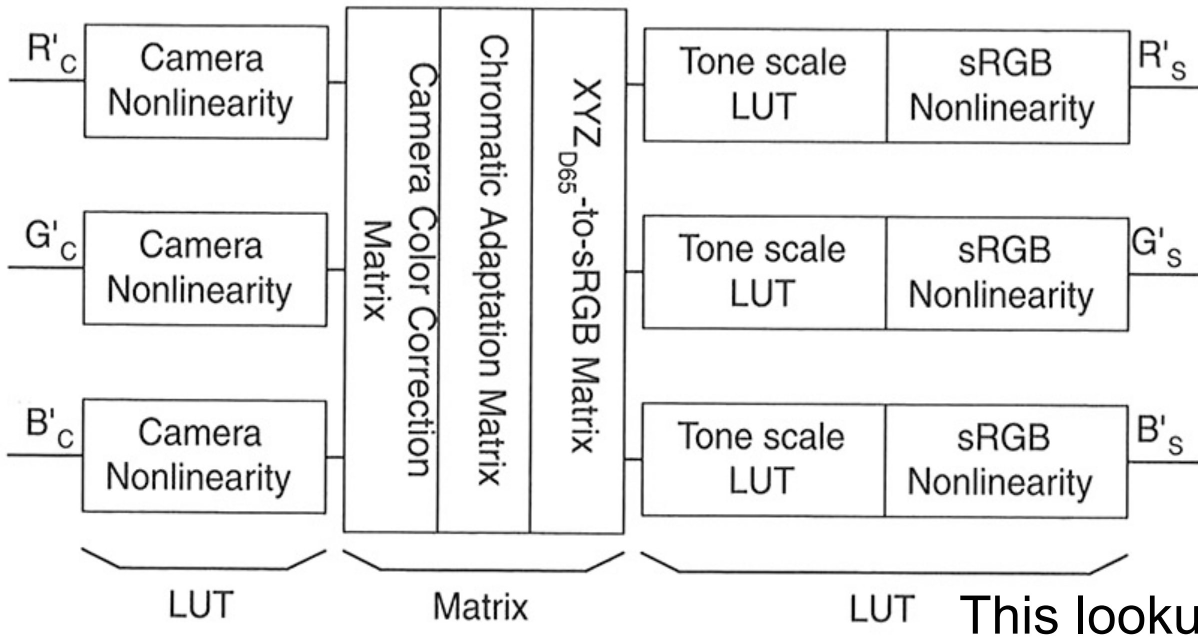
Color triangle is where all of R, G, and B are non-negative.

For example, this triangle edge:
R = 0 = 3.2410X–1.5374Y–0.4986Z



LUT-Matrix-LUT rendering

(Parulski & Spaulding)



This lookup table does both tone rendering and gamma compression

Figure 12.15 LUT-matrix-LUT sequence.

RIMM RGB as before. However, the same tone reproduction can be achieved by populating the LUT appropriately.) Because the groups of LUTs and matrices can be cascaded together, it can be seen that this imaging chain reduces to a simple LUT–matrix–LUT sequence. This is about as simple as the tone scale/color processing can get, and it is representative of the processing used in many consumer digital cameras.

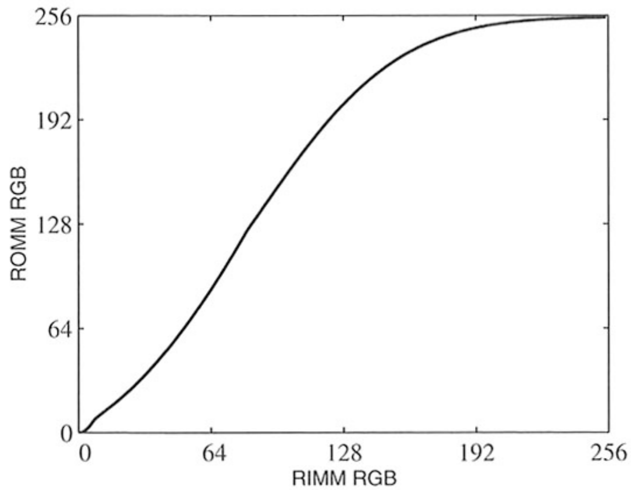


Figure 12.13 Typical ROMM RGB tone scale LUT.

Tone-rendering Nonlinearity "s-curve"

- Optionally compute auto-adjust parameters
  - Exposure (gain; compensate for over/under exposure)
  - Contrast (gamma; boost midtone contrast)
  - Highlight (shoulder compression)
  - Shadow (black level offset and soft landing)
- Compute and apply tone LUT
  - Map to 8-bit or 16-bit output
  - Optionally include colorspace gamma curve

Intermediate Color Spaces:
input-referred (RIMM) and output-referred (ROMM) versions

$$\begin{bmatrix} R_{RIMM} \\ G_{RIMM} \\ B_{RIMM} \end{bmatrix} = \begin{bmatrix} 1.3460 & -0.2556 & -0.0511 \\ -0.5446 & 1.5082 & 0.0205 \\ 0.0000 & 0.0000 & 1.2123 \end{bmatrix} \begin{bmatrix} X_{D50} \\ Y_{D50} \\ Z_{50} \end{bmatrix} \qquad (12.1)$$

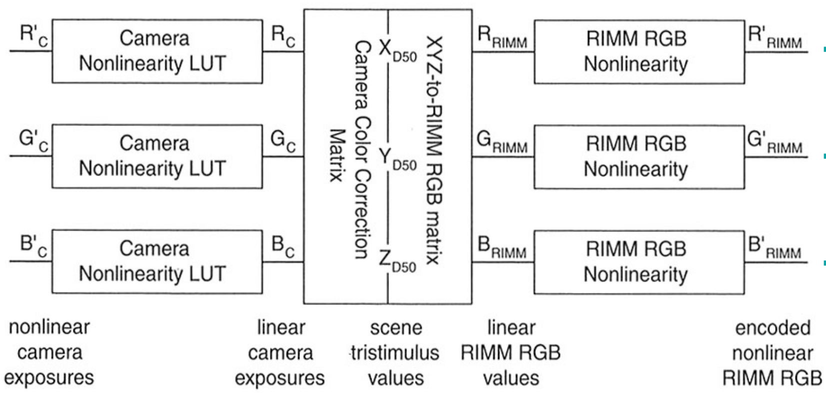


Figure 12.12 Model that uses a LUT–matrix–LUT processing chain to comput RIMM RGB scene color values.

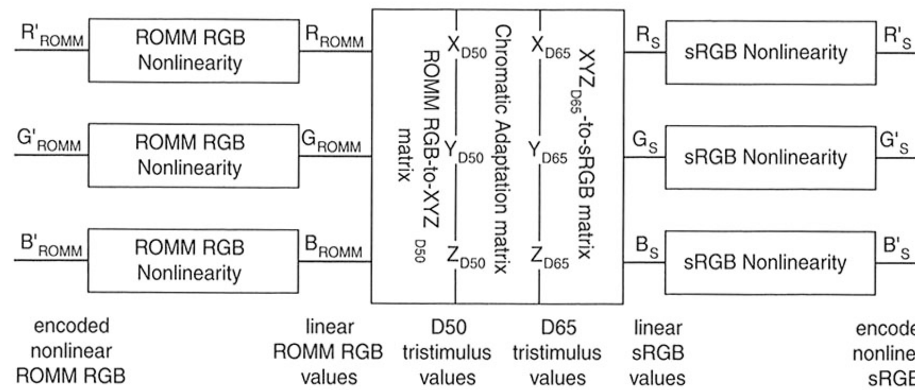


Figure 12.14 Example output model for converting an image in ROMM RGB color encoding to sRGB.

Finish Up

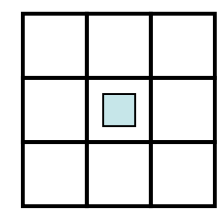
- Matrix to final RGB space if not already there
  - In a raw converter, you may for example render to sRGB for on-screen viewing, and AdobeRGB for file output (or perhaps use system's ICC color profiles to map to the chosen space)
  - Soft limit again, since matrix can create value > 1 (too-bright color) or < 0 (outside color triangle)
- Apply gamma nonlinearity for chosen output space spec
  - sRGB has linear and power-law segments
- Save to file, perhaps with compression
  - JPEG uses conversion to Y-Cb-Cr space, another 3x3 matrix transform and offset of color pixel values

Noise Reduction

- Nonlinear filters of various sorts
  - Rank-conditioned rank-selection (RCRS) filters: median, despeckling, etc.
  - Limited-change smoothing filters to suppress small fluctuations
  - Edge-preserving smoothers
  - Luma-dependent chroma gain reduction

RCRS filters on 3x3 neighborhoods

- Select value of some rank to use in center,
  - conditioned on the rank of the center value in the neighborhood
- Median:
  - 1 2 3 4 5 6 7 8 9 rank of center pixel
  - 5 5 5 5 5 5 5 5 5 selected rank
  - always select rank-5 (median) element

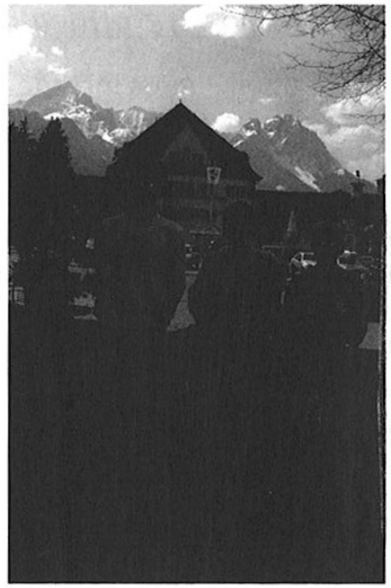


- Despeckle:
  - 1 2 3 4 5 6 7 8 9 rank of center pixel
  - 2 2 3 4 5 6 7 8 8 selected rank
  - replace with nearest value iff an extreme outlier
- Others in between these extremes

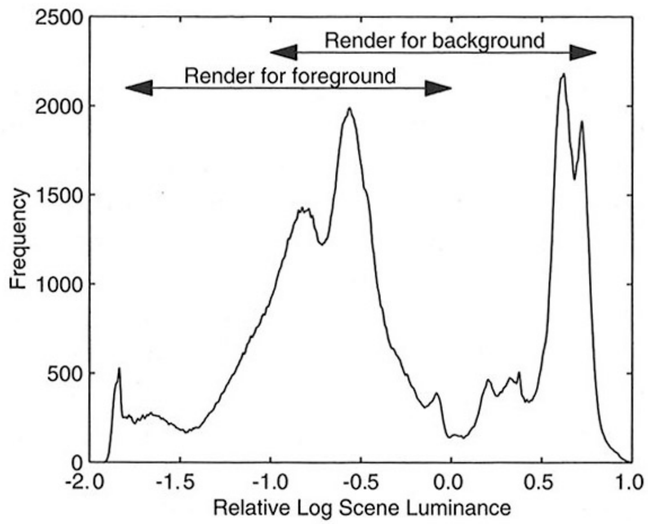
More Optional Extras

- Lens distortion correction
  - or warp to non-rectiliear, e.g. cylindrical
- High-dynamic-range re-mapping
  - e. g., based on Land’s Retinex model
- Stitching and blending for panoramas
  - Using detailed camera geometric calibration and/or image feature matching

(a)



(b)



(c)

Figure 12.9 ***(See color insert)*** Images generated from a high-dynamic-range scene.

The HDR problem: what to do with all that dynamic range

Big Topics Barely Mentioned

(each could be an hour lecture or more)

- Auto white balance and auto exposure
- Illuminant adaptation matrix computation
- Auto tone rendering, color saturation, etc.
- Noise reduction algorithms
- Handling ICC color profiles
- Calibration of all the processing parameters
- Implementation and performance