Understanding the in-camera rendering pipeline & the role of AI and deep learning

Michael S. Brown
Professor and Canada Research Chair
York University – Toronto

Senior Research Director
Samsung AI Center – Toronto

The contents of these slides represent my personal opinion and interpretations of the topics and ideas contained within.
Motivation for this tutorial
Scientist’s view of a photograph

Photo by Uwe Hermann
Scientist’s view of a photograph

Photo by Uwe Hermann
Image = radiant energy measurement

Simple models of a camera assume an image is a “quantitative measurement” of scene radiance.
Simple models of a camera assumes an image is a “quantitative measurement” of scene radiance.
Camera = light measuring device

Medical imaging explicitly requires accurate measurements.
Home diagnostic testing requires accurate measurements across different cameras.
Camera = light measuring device?
In-camera photo-finishing is the “secret recipe” of a camera

Photographs taken from three different cameras with the same aperture, shutter speed, white-balance, ISO, and picture style.
In-camera photo-finishing may cause problems for scientific applications!

Which one is correct?
Motivation

• Cameras are the primary tool used to capture digital images.

• Digital images are the primary inputs to CV algorithms.

• CV researchers/engineers should have a basic understanding of how cameras work to inform their algorithms.

• This tutorial aims to provide this basic understanding.
A tutorial in three parts

Part 1: Review of color, color constancy, and color spaces
  • CIE XYZ, chromatic adaption, color temperature, and output color spaces
  • Background on color is necessary to understand Part 2

Part 2: Overview of a typical camera pipeline (ISP)
  • Discuss the processing steps used by most ISPs
  • Note that some steps are their own research topic

Part 3: Deep-learning/AI and the ISP
  • Machine learning for individual ISP components
  • Replacing the whole ISP with DNNs
Part 1: Review of color, color constancy, color temperature, and color spaces
Color and color spaces

• To understand your camera, it is important to review how humans perceive color in a real environment.

• We must also understand how color is encoded by various models and color spaces.

• One of the main roles of the in-camera hardware is to convert the sensor image into a standard output-referred color space suitable for sharing and display.
Color is perceptual

- Color is **not** a primary *physical* property of an object.
- Red, green, blue, pink, orange, purple, yellow, . . .
  - These are words we assign to visual sensations.
  - The assignment of words can vary among cultures.

Which is the "true blue"?
Where do “color sensations” come from?

A very small range of electromagnetic radiation

Generally wavelengths from 380 to 720nm are visible to most individuals.
White light through a prism

Light is separated into “monochromatic” light at different wave lengths.

Isaac Newton
1704 - Opticks
Biology of color sensations

Our eye has three receptors (cone cells). The different cones respond to different ranges of the visible light spectrum.
Cones and rods

- We have additional light-sensitive cells called *rods* that are not responsible for color. Rods are used in low-light vision.
- Cone cells are most concentrated around the fovea of the eye.
Spectral power distribution (SPD)

We rarely see monochromatic light in real world scenes. Instead, objects reflect a wide range of wavelengths. This can be described by a spectral power distribution (SPD) shown above. The SPD plot shows the relative amount of each wavelength reflected over the visible spectrum.
SPD relation to perceived color is not unique

Due to the accumulation effect of the cones, two different SPDs can be perceived as the same color (such SPDs are called “metamers”).

Lettuce SPD

Response (normalized)

SPD of “real lettuce”

Lettuce

Green ink

Result in the same color “sensation”.

SPD of ink in a “picture of lettuce”

Lettuce SPD stimulating
S=0.2, M=0.8,
L=0.8

Green ink SPD stimulating
S=0.2, M=0.8,
L=0.8

Lettuce SPD
stimulating
S=0.2, M=0.8,
L=0.8

Green Ink SPD
Tristimulus color theory

• Before the biology of cone cells was understood, it was empirically known that only three distinct colors (primaries) could be mixed to produce other colors.

• Moses Harris (1766), Thomas Young (1803), Johann Wolfgang von Goethe (1810), Hermann Grassman (1853), James Maxwell (1856) all explored the theory of trichromacy for human vision.

From Harris “The Natural System of Colours"
Tristimulus color theory

Grassman’s Law states that a source color can be matched by a linear combination of three independent “primaries”.

Three lights (shown as lightbulbs) serve as primaries. Each light has intensity, or weights, $R_1$, $G_1$, $B_1$ to match the source light #1 perceived color.

Same three primaries and the weights ($R_2$, $G_2$, $B_2$) of each primary needed to match the source light #2 perceived color.

If we combine source lights 1 & 2 to get a new source light #3, the amount of each primary needed to match the new source light #3 is the sum of the weights that matched lights sources #1 & #2.

This may seem obvious now, but discovering that light obeys the laws of linear algebra was a huge and useful discovery.
Radiometry vs. photometry/colorimetry

• **Radiometry**
  - Quantitative measurements of radiant energy.
  - Often shown as spectral power distributions (SPD).
  - Measures light coming from a source (radiance) or light falling on a surface (irradiance).

• **Photometry/colorimetry**
  - Quantitative measurement of perceived radiant energy based on human’s sensitivity to light.
  - Perceived in terms of “brightness” (photometry) and color (colorimetry).
Quantifying color

- Human cone photoreceptors (L/M/S) were being characterized well into the 2000s.\textsuperscript{1,2}
- The need to quantify color and brightness existed much earlier.
- Since SPDs go through a “black box” (human visual system), the only way to quantify the “black box” is to perform a human study.
- Two key experiments
  - To quantify perceived “brightness” (photometry)
  - To quantify perceived “color” (colorimetry)

\textsuperscript{1}Schnapf et al. “Spectral sensitivity of human cone photoreceptors,” Nature 1987
Experiments for photometry

The “flicker photometry” experiment for photopic sensitivity.

Chromatic source light at a particular wavelength and adjustable radiant power.

Radiant power of chromatic light.

Reference bright light with fixed radiant power.

(Alternating between source and reference @ 17Hz)

Alternate between the source light and reference light 17 times per second (17 hz). A flicker will be noticeable unless the two lights have the same perceived “brightness”.

The viewer adjusts the radiant power of the chromatic light until the flicker disappears (i.e. the lights fuse into a constant color). The amount of radiant power needed for this fusion to happen is recorded.

Repeat this flicker fusion test for each wave length in the source light. This allows method can be used to determine the perceived “brightness” of each wavelength.
Result of the flicker experiments

Perform the flicker experiment for each wavelength.

Amount of radiant power needed for each wavelength to make the reference light.

You need a lot more 400nm light to match the reference than you do the 550nm.

This means you perceive 550nm brighter than 400nm.
The Luminosity Function (written as $\bar{V}(\lambda)$ or $V(\lambda)$) shows the eye’s sensitivity to radiant energy into luminous energy (or perceived radiant energy) based on human experiments (flicker fusion test).

International Commission on Illumination (CIE comes from the French name Commission internationale e l'éclairage) was a body established in 1913 as an authority on light, illumination and color. CIE is still active today -- [http://www.cie.co.at](http://www.cie.co.at)
Radiometric to Photometric

How do we use CIE Y (or $\tilde{y}(\lambda)$)?

SPD1 and SPD2 are clearly different. Which one will be perceived brighter (assuming the same overall radiant power.)

Which SDP is perceived brighter?

SPD1
Y = 0.2989

SPD2
Y = 0.2989

Radiometric
Photometric

CIE Y gives a way to go from radiometric to photometric!
Now can quantify the perceived brightness of different light.
Radiometric vs. photometric units

Radiometric values:
- Radiant Flux (watt)
- Radiant intensity (watt per steradian)
- Irradiance (watt per m²—falling on surface)
- Radiance (watt per m² steradian)

Photometric values (Radiometric values weighted by the Luminosity Function):
- Luminous Flux (lumens)
- Luminous Intensity (candela)
- Illuminance (lux)
- Luminance (candela per m²)

3100 lumens colour brightness
Colorimetry

Based on tristimulus color theory, colorimetry attempts to quantify all visible colors in terms of a standard set of primaries.

\[
\text{Target color} = R1^* + G1^* + B1^*
\]

Three fixed primary lights.
CIE RGB color matching

Human subjects matched test colors by add or subtracting three primaries.

Field of view was 2-degrees (where color cones are most concentrated)

"Standard Observer"
(Willing participant with no eye disease)

Experiments carried out by
W. David Wright (Imperial College) and John Guild (National Physical Laboratory, London) – Late 1920s
For some test colors, no mix of the primaries could give a match! For these cases, the subjects were asked to add primaries to the test color to make the match.

This was treated as a negative value of the primary added to the test color.
CIE RGB results

Plots are of the mixing coefficients of each primary needed to produce the corresponding monochromatic light at that wavelength.

Note that these functions have been scaled such that area of each curve is equal.
Negative values -- the three primaries used did not span the full range of perceptual colors.
CIE 1931 XYZ

• In 1931, the CIE met and approved defining a new canonical basis, termed XYZ that would be derived from Wright-Guild’s CIE RGB data.

• Properties desired in this conversion:
  • Positive values only
  • Pure white light (flat SPD) to lie at $X=1/3$, $Y=1/3$, $Z=1/3$
  • $Y$ would be the luminosity function ($V(\lambda)$)

• Quite a bit of freedom in selecting the XYZ basis
  • In the end, the adopted transform was:

\[
\begin{bmatrix}
X \\
Y \\
Z \\
\end{bmatrix} = \begin{bmatrix}
0.4887180 & 0.3106803 & 0.2006017 \\
0.1762044 & 0.8129847 & 0.0108109 \\
0.0000000 & 0.0102048 & 0.9897952 \\
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B \\
\end{bmatrix}
\]

CIE 1931 RGB

Nice article see: Fairman et al “How the CIE 1931 Color-Matching Functions Were Derived from Wright–Guild Data”, Color Research & Application, 1997
CIE 1931 XYZ

This shows the mixing coefficients $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ for the CIE 1931 2-degree standard observer XYZ basis computed from the CIE RGB data. Coefficients are all now positive. Note that the basis XYZ are not physical SPD like in CIE RGB, but linear combinations defined by the matrix on the previous slide.
How do we use CIE XYZ?

SPD1 and SPD2 are clearly different. Will they be perceived as the same color?

\[
X = \int_{380}^{780} SPD(\lambda) \bar{x}(\lambda) \, d\lambda \\
Y = \int_{380}^{780} SPD(\lambda) \bar{y}(\lambda) \, d\lambda \\
Z = \int_{380}^{780} SPD(\lambda) \bar{z}(\lambda) \, d\lambda
\]
How do we use CIE XYZ?

SPD1 and SPD2 are clearly different. Will they be perceived as the same color?

From their CIE XYZ mappings, we can determine that these two SPDs will be perceived as the same color!

Now we can quantify color!

CIE XYZ gives a way to go from radiometric to colorimetric. Imbedded is also the photometric measurement in the Y value.
It is challenging to visualize the 3D CIE XYZ space. We often don’t plot color in this space.
Luminance-chromaticity space (CIE xyY)

• CIE XYZ describes a color in terms of linear combination of three primaries (XYZ).

• Sometimes it is useful to discuss color in terms of luminance (perceived brightness) and chromaticity (we can think of as the hue-saturation combined).

• CIE xyY space is used for this purpose.
Point “E” represents where $X=Y=Z$ have equal energy ($X=0.33, Y=0.33, Z=0.33$)

In the 1930s, CIE had a bad habit of over using the variables $X,Y$. Note that $x,y$ are chromaticity coordinates, $\bar{x}, \bar{y}$ (with the bar above) are the matching functions, and $X,Y$ are the imaginary SPDs of CIE XYZ.
Usefulness of CIE 1931 XYZ

• CIE XYZ space is a “device independent” space – the XYZ values are not specific to any device.

• Electronic devices (e.g. cameras, flatbed, scanners, printers, displays) can compute mappings of their device specific values to the corresponding CIE XYZ values.

• This provides a canonical space to match between devices (at least in theory).
A caution on CIE xy chromaticity

From Mark D. Fairchild’s book: “Color Appearance Models”

“The use of chromaticity diagrams should be avoided in most circumstances, particularly when the phenomena being investigated are highly dependent on the three-dimensional nature of color. For example, the display and comparison of the color gamuts of imaging devices in chromaticity diagrams is misleading to the point of being almost completely erroneous.”
Fast forward 90+ years

• CIE 1931 XYZ, CIE 1931 xyY (2-degree standard observer) color spaces have stood the test of time.

• Many other studies have followed (most notably - CIE 1965 XYZ 10-degree standard observer), …

• But in the literature (and in this tutorial) you’ll find CIE 1931 XYZ color space remains the preferred standard.
What is perhaps most amazing?

• 90+ years of CIE XYZ, and it is all based on the experiments by Guild and Wright’s “standard observers.”

• How many standard observers were used? 100, 500, 1000?

A standard observer
CIE XYZ is based on 17 (male) standard observers

10 by Wright, 7 by Guild

“The Standard Observers”
Can we talk about cameras now?

Sorry, not yet …
An object’s SPD

In the real world, most objects do not emit an SPD, instead, they reflect an SPD. As a result, an object’s SPD depends on the environmental illumination.

Our earlier example ignored illumination (we could assume it was pure white light). Instead, think of this of how the object reflects different wavelengths.
Color constancy

Our visual system has an amazing ability to compensate for environmental illumination such that objects are perceived as the same color.
Chromatic adaptation example

Example from Andrew Stockman (UCL)
Chromatic adaptation example
Color constancy/chromatic adaptation

• Color constancy (*chromatic adaptation*) is the ability of the human visual system to adapt to scene illumination.

• This ability is not perfect, but it works fairly well.

• *Image sensors do not have this ability!* We will discuss this in part 2 . . this is related to the camera’s white-balance module.
Color constancy (at its simplest)

- The Von Kries transform
- Compensate for L/M/S channel corresponding to the L, M, S response to scene illumination.

\[
\begin{bmatrix}
L'_x \\
M'_x \\
S'_x
\end{bmatrix} = \begin{bmatrix}
1/L_{illu} & 0 & 0 \\
0 & 1/M_{illu} & 0 \\
0 & 0 & 1/S_{illu}
\end{bmatrix} \begin{bmatrix}
L_x \\
M_x \\
S_x
\end{bmatrix}
\]

"Corrected colors"

Long/medium/short cone response with illumination "corrected."

Divide out long/medium/short cone response to the scene's illuminant.

Long/medium/short cone response to scene point \(x\) under some illuminant.
Color constancy for printed media

Printed media (i.e., stuff that reflects light)

Scene illumination

The white paper reflects the light. The paper is almost a perfect reflector. Since we are adapting to the environmental light source, the paper appears white.

The photo also reflects the light, so the colors are perceived correctly.
Color constancy for emissive media

Emissive media (e.g., monitor/tablet, smartphone screen)

The display does not reflect light. Because we are adapting to the environmental lighting, we need the display to match the scene illumination. If we match the illumination, the display will appear “white.”

The displayed image colors will appear differently than intended, since we are adapting to the environment illumination.
Implications of the previous slides

• **Color is intimately connected to scene illumination.**

• Even for emissive displays, we have to consider (or make assumptions) about the illumination in the viewing environment of the display.

• Keep this in mind because it will play a role when we define color spaces used to encode our images.
Understanding color temperature

• In the photography and display communities, an illumination’s “color” is described using a correlated color temperature (CCT).

• White balance on cameras also often uses color temperature to describe illumination.

• Note that this has nothing to do with a light source's “temperature” (i.e., a measurement of heat).

• This is an excellent example of where metamers are used.
  • Recall – a metamer is when two different SPDs appear visually the same color.
SPDs of common illuminations

Figures from Ponce and Forsyth
CIE standard illuminants

SPDs for CIE standard illuminant A, B, C

SPDs for CIE standard illuminant D50, D55, D65

SPDs for CIE standard illuminant E

SPDs for CIE standard illuminants F2, F8, F11

D, E, and F series images from http://www.image-engineering.de
Color temperature

- As mentioned, illuminants are often described by their “color temperature.”
- This mapping is based on theoretical blackbody radiators that produce SPDs for a given temperature expressed in Kelvin (K).
- We map light sources (both real and synthetic) to their closest color temperature.

\[
B_\lambda(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1}
\]

Plank's law
Spectral density of electromagnetic radiation emitted by a blackbody radiator at a given temperature T.
Visible range of a black body radiator SPD

Consider only the visible wavelengths from Plank's equation at a certain temperature.

Black body radiator SPD for different color temperatures

Animation credit: Dariusz Kowalczyk
Plot visible SPDs in CIE xy chromaticity

This curve in the CIE xy plot of the "Planckian Locus" of color temperatures.
Color temperature of an SPD example

(1) Find the light sources SPD mapping to CIE XYZ using the CIE 1931 mapping functions.
(2) Project the CIE xyY value to the Planckian locus line.

Where the projection falls is the Correlated Color Temperature (CCT) of this light source.
So, this example the OLED light source is roughly 4500K.

While we often say "color temperature", we should say "correlated color temperature". The concept is not related to the physical temperature of the light source, but its correlation with the black body radiator's color temperature.
Color temperature

Kelvin Color Temperature Scale

10,000K: 10,000K+: Blue Sky
9,000K
8,000K
7,000K: 7,000K-7,500K: Cool White Seesmart LED
6,000K: 6,000K: Cloudy Sky
5,000K: 5,500K-6,000K: Day White Seesmart LED
4,800K: Direct Sunlight
4,000K: 4,000K-4,500K: Natural White Seesmart LED
4,000K: Clear Metal Halide
3,000K: 3,000K: Halogen
2,800K: 100W Incandescent
2,700K-3,200K: Warm White Seesmart LED
2,200K: High Pressure
1,900K: Candle

Typical description of color temperature used in photography & lighting sources.
Lighting industry uses color temperature

Usage of correlated color temperature in these ads relate to the perceived color of the bulb's light. The heat output of a typical LED bulb is between 60°C-100°C (~333-373K).
White point

• A white point is a color defined in CIE xyY that we want to be considered “white” (or achromatic/neutral).

• This is essentially an illuminant’s SPD in terms of CIE XYZ/CIE xyY
  • Think of it as CIE Yxy value of a white piece of paper under some illumination.

CIE Illuminants
A, B, C, D65, E in terms of CIE xy

<table>
<thead>
<tr>
<th>CIE</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.44757</td>
<td>0.40745</td>
</tr>
<tr>
<td>B</td>
<td>0.34842</td>
<td>0.35161</td>
</tr>
<tr>
<td>C</td>
<td>0.31006</td>
<td>0.31616</td>
</tr>
<tr>
<td>D65</td>
<td>0.31271</td>
<td>0.32902</td>
</tr>
<tr>
<td>E</td>
<td>0.33333</td>
<td>0.33333</td>
</tr>
</tbody>
</table>
Quick summary on color constancy

• Color constancy is our ability to adapt to illumination in the scene.

• Correlated Color Temperature (CCT) — or just color temperature — is a system used to describe scene illumination.

• **Note:** we must factor in the scene illumination when capturing and displaying color images.
Color adaptation is not perfect

“True color constancy, almost never. Inconstancy, nearly 100% of the time.”

Mark Fairchild

“Remember that all [mathematical] models are wrong; the practical question is how wrong do they have to be to not be useful.”

George Box (Statistician pioneer)
Now we are finally done with color?

Almost …
CIE XYZ and RGB

• While CIE XYZ is a canonical color space, images/devices rarely work directly with XYZ.

• RGB primaries dominate the industry, this is because we can produce RGB light sources (LEDs, phosphorus for CRT monitors, filters, etc).

• We are all familiar with the RGB color cube.

• But is the color cube a color space?

By now, you should realize that “red”, “green”, and “blue” have no quantitative meaning as words. We need to know their corresponding SPDs or CIE XYZ values.
**Color model versus color space**

- **A color model** is a mathematical system for describing a color as a tuple of numbers (RGB, HSV, HSL, more…)
- **A color space** is a specific range of colors *within a color model*. The range of color (gamut) can be expressed in CIE XYZ. Color spaces typically also define the viewing environment and, therefore, the “white point” of the space.

<table>
<thead>
<tr>
<th>Color models (not color spaces)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB color model</td>
</tr>
<tr>
<td>HSV color model</td>
</tr>
<tr>
<td>HSL color model</td>
</tr>
</tbody>
</table>
Defining a color space with specific RGB values

The RGB values span a subspace, of CIE-XYZ to define the devices gamut.

We need to define our RGB values.
Problem with just a color model.

RGB values must be specified. If not, this is a **huge** problem for color reproduction from one device to the next.

Which RGB primaries are the right ones?
In 1996, Microsoft and HP defined a set of “standard” RGB primaries.
R = CIE xyY (0.64, 0.33, 0.2126)
G = CIE xyY (0.30, 0.60, 0.7153)
B = CIE xyY (0.15, 0.06, 0.0721)

This was considered an RGB space achievable by most devices at the time.

The white point was set to the D65 illuminant. **This is an important to note.** It means sRGB has built in the assumed viewing condition (6500K daylight).
sRGB's white point

• Color spaces intended for display (called display-referred or output-referred) define a white-point.

• Remember to match the assumed illumination in the viewing environment
  • The “white” of sRGB (i.e., \([1,1,1]\)) is displayed at D65

The positions of the white-point locations are exaggerated here.
Assumed viewing illumination is important

Remember “the dress”?

Image: Scientific America article explaining how viewing environment lighting impacted our perception of the color.
CIE XYZ to sRGB conversion

Matrix conversion:

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} = \begin{bmatrix}
3.2404542 & -1.5371385 & 0.4985314 \\
-0.9692660 & 1.8760108 & 0.0415560 \\
0.0556434 & -0.2040259 & 1.0572252
\end{bmatrix} \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

- D65 is set as the white-point.
- This is the linear sRGB space.
- sRGB also specifies a gamma correction of the values (next slide)
- The CIE refers to this as the Recommendation 709 color space – or Rec.709.
sRGB gamma curve

The actual formula is a bit complicated, but effectively this is gamma ($I' = 255 \times I^{(1/2.2)}$) where $I'$ is the output intensity and $I$ is the linear sRGB ranged 0-1, with a small linear transfer for linearized sRGB values close to 0 (not shown in this plot). This is known as “perceptual encoding” and is intended to allocate more bits based on our nonlinear response to radiant power.
Stevens' power law

- Physical stimulus vs. perceptual sensation
- Stevens' Power Law

\[ S = k I^a \]

Human sensation \( \rightarrow \) Constant \( \rightarrow \) Stimulus intensity

Dr. Stanley Stevens showed that most human sensations follow a power-law relationship between stimuli and sensation.

Stevens' model stated that human perception to brightness followed a power law.

\[ \psi = 10 \Phi^{1/3} \]
Stevens' power law

Interpreting the power law.

A constant (linear) increase in perceived brightness.

The radiant power needs to change exponentially.
The sRGB gamma approximates Steven's \( \frac{1}{3} \) power-law.

The reason we apply gamma is that it remaps the linear color to fit better our visual system's nonlinear response to radiant power.

There is a misconception in many graphics and image processing textbooks that gamma is applied to compensate for displays (CRTs). See a nice writeup about this by Poynton.¹

¹ [https://poynton.ca/PDFs/Rehabilitation_of_gamma.pdf](https://poynton.ca/PDFs/Rehabilitation_of_gamma.pdf)
Before (linear sRGB) & after (sRGB)
Standardization is not new - NTSC/PAL

Both NTSC and sRGB used gamma encodings. Most color spaces use some type of perceptual encoding.
NTSC/sRGB
(know your color space!)

It is important to known which color space your image is in.

Linear-sRGB back to XYZ

\[
\begin{bmatrix}
X \\
Y \\
Z \\
\end{bmatrix} = \begin{bmatrix}
0.4124 & 0.3576 & 0.1805 \\
0.2126 & 0.7152 & 0.0722 \\
0.0193 & 0.1192 & 0.9505 \\
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B \\
\end{bmatrix}
\]

Linear-NTSC back to XYZ

\[
\begin{bmatrix}
X \\
Y \\
Z \\
\end{bmatrix} = \begin{bmatrix}
0.6071 & 0.1736 & 0.1995 \\
0.2990 & 0.5870 & 0.1140 \\
0.0000 & 0.0661 & 1.1115 \\
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B \\
\end{bmatrix}
\]

Many color APIs (e.g., matlab, python) assume the default color space is NTSC. Many research papers use the wrong equations!
An additional fun fact

• Physical stimulus vs. human sensations
• Stevens' Power Law

\[ S = kI^{\alpha} \]

Human sensation vs. Constant

Stimulus intensity

Dr. Stanley Stevens introduced showed that most human sensations follow a power-law relationship between stimuli and sensation.

\( \psi = 0.0015\Phi^{3.5} \)
\( \psi = 10\Phi^{1/3} \)

(curve of pain) (curve for brightness)

Stevens also did experiment on the pain sensation of electrical shock! Turns out our sensitivity is the opposite than with radiometric power to brightness.
CIE XYZ: The mother color space

- ProPhoto
- Adobe RGB
- CIE LAB
- sRGB
- NTSC/PAL
- Display P3
Other common color spaces

This tutorial does not go into the details of the mathematical transformations to other color spaces (we'd need another tutorial for that). You can find the transforms online.

The goal here is to explain the rationale behind each transform so you understand why the other color spaces are introduced.
CIE LAB space

CIE LAB space (also written as CIE \(L^*a^*b^*\)) was introduced as a perceptually uniform color space.

Why?
- CIE XYZ provides a means to map between a physical SPD (radiometric measurement) to a colorimetric measurement (perceptual).
- However, a uniform change in CIE XYZ space does result in an uniform change in perceived color difference (see diagram).
- CIE Lab transforms CIE to a new space where color (and brightness) differences are more uniform.

David MacAdam performed experiments on color perception. This plot is known as the MacAdam ellipses.
CIE 1976 LAB

• Considering the MacAdam experiments and the Steven's power-law, CIE LAB was derived in 1976 by applying various transformations to the CIE XYZ values that result in the following:

• $L^*$ represents a **perceptual brightness** measure between 0-100
  • $L^*$ is a non-linear (gamma) transformation of the Y component of CIE XYZ.
  • $L$ is approximately a cube root of Y (directly from Steven's power law)

• $a^*$ and $b^*$ (often range ±50)
  • Both have similar non-linear transformations applied, and represent approximately:
  • $a^*$ values lying along colors related to red and green
  • $b^*$ values lying along colors related to yellow and blue
  • $a^*=b^*=0$ represents neutral grey colors

**NOTE:** CIE LAB requires the white point to be specified for the transformation. The default white point is D65.
CIE LAB

Chromaticity comparison's between CIE LAB and CIE XYX

Image from Bagdasar et al ICSTCC'17
Color error metric – CIE 2000 Delta E ($\Delta E$)

• Delta E is a color metric based on L*ab space.
• Since L*ab is more uniformly perceptual, distances (e.g., Euclidean distance) in L*ab have more meaning than in CIE XYZ.
• Delta E values have an interpretation as follows.

<table>
<thead>
<tr>
<th>Delta E</th>
<th>Perception</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;= 1.0</td>
<td>Not perceptible by human eyes.</td>
</tr>
<tr>
<td>1 - 2</td>
<td>Perceptible through close observation.</td>
</tr>
<tr>
<td>2 - 10</td>
<td>Perceptible at a glance.</td>
</tr>
<tr>
<td>11 - 49</td>
<td>Colors are more similar than opposite</td>
</tr>
<tr>
<td>100</td>
<td>Colors are exact opposite</td>
</tr>
</tbody>
</table>

In general, a $\Delta E$ of 2 or less is considered to be very good. It means a standard observer could not tell that two colors are different unless they observe them very closely.

Table from https://zschuessler.github.io/DeltaE/learn/
Other color spaces to be aware of

• Adobe RGB
  • Medium gamut color space
  • Used for photo-editing

• Display P3
  • Medium gamut color space
  • Used by Apple devices to accommodate better display technology
  • Similar to Adobe RGB

• ProPhoto (ROMM)
  • Developed by Kodak
  • Intended to encode a wide range of colors and dynamic range

These are known as “output referred” color spaces because they are defined for encoding images for display or output devices. The definition of color spaces also states the space’s preferred dynamic range and viewing environment (although we rarely view in such conditions).
A color space’s gamut is the span of colors that can be represented. The 3D gamuts are plotted in CIE L*ab.
Gamuts expressed in chromaticity are misleading

AdobeRGB plotted in CIE XYZ and then projected to 2D CIE Yxy chromaticity.

*See slide 44 (Mark Fairchild's comments)
ProPhoto encodes over 90% of surface colors (color from reflected light of a surface, i.e., not emitted light). It is recommended to use 16-bit values per channel since the gamut is so large. The white point is D50.
Adobe RGB/Display P3 color space

Medium-gamut AdobeRGB color space. (Apple’s Display-P3 is very similar).

AdobeRGB/Display P3 encodes over ~50% of surface colors. Display P3 is Apple’s encoding color space. It is recommended to use 10-bit per channel. The white point is D65.
sRGB color space*

*Currently, sRGB is the most common color space (designed for 1990s display technology). sRGB encodes ~30% of surface colors. Developed for 8-bit encoding per channel. The white point is D65.
PowerPoint expects images to be in sRGB. When encoded in a wider gamut color space, the image may appear dull. This may seem counter-intuitive because the wider gamut should encode more colors, but this is only possible when the software and display hardware are aware (and capable) of interpreting the color space correctly.
Now, are we really finally done with color?

Yes . . .

But remember that color appearance, measurement, and encoding is its own research field. My slides provide only a basic introduction. The CV community is bad at abusing color terminology or not putting in enough effort to understand color fully.
Congratulations!

Certificate of Completion

This Certificate is presented to

You

on this 03 day of Oct 2023

for completion of

“Tutorial on color for cameras.”
Part 2: Overview of the in-camera rendering pipeline
In-camera rendering

• The image directly captured from the camera’s sensor needs to be processed.

• We can call this process “rendering,” as the goal is to render a digital image suitable for viewing.
An ISP is dedicated hardware that renders the sensor image to produce the final output.

Companies such as Qualcomm, HiSilicon, Intel (and more) sell ISP chips (often as part of a System on a Chip – SoC).

- Companies can customize the ISP.

Many ISPs now have neural processing units (NPUs).
A typical color imaging pipeline

ISO gain and raw-image processing

RGB Demoasicing

Sensor with color filter array (CFA) (CCD/CMOS)

NOTE: This diagram represents the steps applied on a typical consumer camera pipeline. ISPs may apply these steps in a different order or combine them in various ways. A modern camera ISP will undoubtedly be more complex but will almost certainly implement these steps in some manner.
A typical color imaging pipeline

Sensor with color filter array (CFA) (CCD/CMOS)

ISO gain and raw-image processing

RGB Demoasicing

White-Balance & Color Space Transform (CIE XYZ)

Noise Reduction

JPEG/HEIC Compression

Save to file

Image Rescaling (or up-scaling)

Color Manipulation (Photo-finishing)

Mapping to output color space (e.g., sRGB, P3)
Almost all consumer camera sensors are based on complementary metal-oxide-semiconductor (CMOS) technology. We generally describe sensors in terms of number of pixels and size. The larger the sensor, the better the noise performance as more light can fall on each pixel. Smart phones have small sensors!
Camera sensor RGB values

A Near Infrared (NIR) filter is often placed before the sensor. (This is sometimes called a "hot mirror"). This is because red filters often respond to NIR light.

Silicon/Circuitry

Color filters place over the sensor. This forms a Color Filter Array (CFA) also called a “Bayer Pattern” after inventor Bryce Bayer.

Micro-lenses are placed over the diode to help increase light collection on the sensor.

Photodiode

Photons hit the diode and force out electrons. This design is similar to a solar cell!

Bryce Bayer (Kodak)
Camera RGB sensitivity

• The color filter array (CFA) on the camera filters the light into three sensor-specific RGB primaries

Plotted from camera sensitivity database by Dr. Jinwei Gu from Rochester Institute of Technology (RIT). Dr. Gu is now at SenseTime (USA). [http://www.cis.rit.edu/jwgu/research/camspec/](http://www.cis.rit.edu/jwgu/research/camspec/)
Measuring camera sensitivities

• It is not easy to get information on a camera’s spectral sensitivities.
  • This process is called camera or sensor characterization.
  • The sensitivity needs to factor in the entire camera form factor: lens, NIR filter, and CFA.

• You need specialized equipment to measure camera spectral sensitivities.
  • But of course, reviewer 2 will say obtaining sensitivities curves is easy...

Image Engineering GmbH & Co. KG

camSPEC device for measuring camera spectral sensitivity.

https://www.image-engineering.de/products/equipment/measurement-devices/588-camspecs-express
Sensor raw-RGB image

Remember: physical world is measured by radiometric spectral power distributions.

Your camera sensor RGB filter is sensitive to different regions of the incoming SPD.

raw-RGB represents the physical world's SPD "projected" onto the sensor's spectral filters.
Sensors are linear to irradiance

• Camera sensors are decent light measuring devices.
• If you double the amount of light hitting a sensor's pixel, the digital value output of that pixel will double.

Sensor output is linear with respect to irradiance falling over the sensor over a certain amount of time.

\[ I = i \times t \]

Digital value \( I \) is a linear function of irradiate \( i \) and exposure \( t \).
IMPORTANT: raw-RGB sensor images are not in a standard color space.

Color plots show L2 distance between the raw-RGB values with different cameras.
Displaying raw-RGB images

• Inserting a raw-RGB image in your slides, research paper, etc will result in strange colors.

• Why? Our devices (computers, printers, etc) expect the image to be in a standard color space like sRGB.

This is a raw-RGB image. Why does it look bad? Because the raw-RGB values are not sRGB values.
A typical color imaging pipeline

ISO gain and raw-image processing → RGB demoasicaing

Sensor

White-balance & color space transform (CIE XYZ)

Mapping to output color space (e.g., sRGB, P3)

JPEG/HEIC compression

Save to file

Noise reduction

Image rescaling (or up-scaling)

Color manipulation (Photo-finishing)
ISO signal amplification (gain)

- Imaging sensor signal is amplified and digitized.
- Amplification to assist A/D conversion.
  - Need to get the voltage to the range required for the desired digital output.
- This gain is used to accommodate camera ISO settings.
  - Gain to signal applied on the sensor.
  - Note – gaining the signal also gains image noise.

Different ISO settings (note: the exposure will be shorter for higher ISO)
Pixel "intensity"

• We often talk about a pixel's intensity, however, a pixel's numerical value has no unit.

• The digital value of a pixel is based on several factors.
  - Exposure (which is a function of both shutter speed and exposure)
  - Gain (ISO setting on the camera)
  - Camera hardware that digitizes the signal.

• We typically rely on the relative digital values in the image and not the absolute digital values.
Black light subtraction

- Sensor values for pixels with “no light” should be zero.
- However, this is not the case due to sensor noise.
  - The black level often changes as the sensor heats up.
- This can be corrected by capturing a set of pixels that do not see light.
- Place a dark shield around the sensor.
- Subtract the level from the “black” pixels.
Optical black (OB)

Defective pixel mask

- CMOS have pixels that are defective.
- Dead pixel masks are pre-calibrated at the factory
  - Using “dark current” calibration
  - Take an image with no light
  - Record locations reporting values to make “mask.”
- Bad pixels in the mask are interpolated.
Defective pixel mask example

Identifying “dead pixels”

After interpolation

Image courtesy of Lu Zheng and Moshe Ben-Ezra
Flat-field correction

Uniform light falling on the sensor may not appear uniform in the raw-RGB image. This can be caused by the lens, sensor position in the camera housing, etc.

We want to correct this problem such that we get a "flat" (or uniform) output.

Before correction

Apply a correction gain over the sensor values.

After correction

This operation can also be called lens shading correction.
A typical color imaging pipeline

1. Sensor
2. ISO gain and raw-image processing
3. RGB demoasicing
4. White-balance & color space transform (CIE XYZ)
5. Noise reduction
6. JPEG/HEIC compression
7. Save to file
8. Mapping to output color space (e.g., sRGB, P3)
9. Color manipulation (Photo-finishing)
10. Image rescaling (or up-scaling)
CFA/Bayer pattern demosaicing

- Color filter array (CFA) pattern placed over pixel sensors.
- We want an RGB value at each pixel, so we need to perform interpolation.
Simple interpolation

This is a zoomed up version of the Bayer pattern.

At location R5, we have a red pixel value, but no Green or Blue pixel.

We need to estimate the G5 & B5 values at location R5.

\[
G5 = \frac{G2 + G4 + G6 + G8}{4}
\]

\[
B5 = \frac{B1 + B3 + B7 + B9}{4}
\]
Simple “edge aware” interpolation

Captured raw-Bayer image

Neighborhood about red pixel

R  ?  G  ?

Neighboring green values

Weight mask based on red pixel's similarity to neighboring red values.

Missing green pixel value is computed as a weighted-interpolation of the neighboring green values.

Do this procedure also for the blue pixel, B5.
Newer CFA/Bayer patterns

- Newer sensors are starting to use different patterns.
- Quad/Tetra (2x2) and Nona (3x3) are now common on smartphones.
- In low-light situations, the 2x2 or 3x3 layouts are “binned” into a single pixel (a process called binning).

![RGB Demoasicing](Image: Sang-wook Park and Jong-hyun Kim)
Demosaicing in practice

• The prior examples are illustrative algorithms only
• **Camera IPSs use more complex and proprietary algorithms.**
• Demosaicing can be combined with additional processing
  • Highlight clipping
  • Sharpening
  • Noise reduction
• **Demosaicing is an active research area!**
A typical color imaging pipeline

1. ISO gain and raw-image processing
2. RGB demosaicing
3. White-balance & color space transform (CIE XYZ)
4. Noise reduction
5. Mapping to output color space (e.g., sRGB, P3)
6. JPEG/HEIC compression
7. Save to file
8. Color manipulation (Photo-finishing)
9. Image rescaling (or up-scaling)
Noise reduction (NR)

• All sensors inherently have noise
• Most cameras apply additional NR after A/D conversion
• A simple method is described in the next slide
• For high-end cameras, it is likely that cameras apply different strategies depending on the ISO settings, e.g. high ISO will result in more noise, so a more aggressive NR could be used
• Smartphone cameras, because the sensor is small, apply aggressive noise reduction.
• Two main sources of image noise:
  1. The quantum nature of light (photon noise/shot noise); unrelated to the imaging sensor. Follows a Poisson distribution.
  2. Electronic sources associated with the imaging sensor circuitry (dark current and dark noise). Often follows a Normal distribution.

• Gain factor $g$ amplifies noise.
A simple noise reduction approach

- Blur the image based on the ISO setting (higher ISO = more blur)
- Blurring will reduce noise, but also remove detail.
- Add image detail back for regions that have a high signal. We can even boost some parts of the signal to enhance detail (i.e. "sharpening")

**Sketch of the procedure here**

- **Input** $I$
- **Blur input** reduces noise but blurs edges $B(I)$
- Subtract $I-B(I)$ if $(I-B(I)) > T$
- Combine high response values, we may assume are image "content" and not noise. We can add this response back to the image (or even boast it).
- Low response areas we don’t add back, but keep the blurred (noised reduced) result.
- **Output** $NR+Sharpened$
A typical color imaging pipeline

1. **Sensor**
2. **ISO gain and raw-image processing**
3. **RGB demoasacing**
4. **White-balance & color space transform (CIE XYZ)**
5. **Save to file**
6. **JPEG/HEIC compression**
7. **Mapping to output color space (e.g., sRGB, P3)**
8. **Color manipulation (Photo-finishing)**
9. **Image rescaling (or up-scaling)**
Color mapping/colorimetric stage

• This step in the IPS converts the sensor raw-RGB values to a device independent color space

Camera sensors have their own spectral response.

We need to map it into a standard response (CIE XYZ).

We will use CIE XYZ in this tutorial, but most cameras use a related space called ProPhoto.
Two step procedure

(1) apply a white-balance correction to the raw-RGB values
(2) map the white-balanced raw-RGB values to CIE XYZ

**White balance**

3x3 diagonal matrix

**Color space transform (CST)**

3x3 full matrix (or polynomial function)
How does white balance (WB) work?

Sensor's response to illumination ($\ell$):

$$\begin{bmatrix}
\ell_r \\
\ell_g \\
\ell_b
\end{bmatrix} = \begin{bmatrix} 0.2 \\
0.8 \\
0.8 \end{bmatrix}$$

(“White-balanced” raw-RGB image)

White-balance diagonal matrix:

$$\begin{bmatrix}
r_{wb} \\
g_{wb} \\
b_{wb}
\end{bmatrix} = \begin{bmatrix} 1/\ell_r & 0 & 0 \\
0 & 1/\ell_g & 0 \\
0 & 0 & 1/\ell_b
\end{bmatrix} \begin{bmatrix} r \\
g \\
b \end{bmatrix}$$
White balance
(computational color constancy)

• The challenging part for white-balance is determining the proper white-balance setting!
• Users can manually set the white balance
  • Camera specific white-balance matrices for common illuminations
  • These can be manually selected by the user
• Otherwise auto white balance (AWB) is performed
  • In computer vision, we often refer to AWB as "illumination estimation"
  • Since the hard part is trying to determine what the illumination in the scene is.
Cameras can pre-calibrate their sensor's response for common illuminations. Typical mapping of WB icons to related color temperatures.

<table>
<thead>
<tr>
<th>WB SETTINGS</th>
<th>COLOR TEMPERATURE</th>
<th>LIGHT SOURCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>☁️ House</td>
<td>10000 - 15000 K</td>
<td>Clear Blue Sky</td>
</tr>
<tr>
<td>☀️ Sun</td>
<td>6500 - 8000 K</td>
<td>Cloudy Sky / Shade</td>
</tr>
<tr>
<td>☀️ Noon</td>
<td>6000 - 7000 K</td>
<td>Noon Sunlight</td>
</tr>
<tr>
<td>☀️ Sun</td>
<td>5500 - 6500 K</td>
<td>Average Daylight</td>
</tr>
<tr>
<td>☀️ Sunset</td>
<td>5000 - 5500 K</td>
<td>Electronic Flash</td>
</tr>
<tr>
<td>☀️ Sunset</td>
<td>4000 - 5000 K</td>
<td>Fluorescent Light</td>
</tr>
<tr>
<td>☀️ Sunset</td>
<td>3000 - 4000 K</td>
<td>Early AM / Late PM</td>
</tr>
<tr>
<td>☀️ Sunset</td>
<td>2500 - 3000 K</td>
<td>Domestic Lightning</td>
</tr>
<tr>
<td>☀️ Sunset</td>
<td>1000 - 2000 K</td>
<td>Candle Flame</td>
</tr>
</tbody>
</table>

Image from ExposureGuide.com
Examples of manual WB matrices

Sunny

\[
\begin{bmatrix}
2.0273 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 1.3906 \\
\end{bmatrix}
\]

Nikon D7000

Incandescent

\[
\begin{bmatrix}
1.3047 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 2.2148 \\
\end{bmatrix}
\]

Shade

\[
\begin{bmatrix}
2.4922 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 1.1367 \\
\end{bmatrix}
\]

Daylight

\[
\begin{bmatrix}
2.0938 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 1.5020 \\
\end{bmatrix}
\]

Canon 1D

Tungsten

\[
\begin{bmatrix}
1.4511 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 2.3487 \\
\end{bmatrix}
\]

Shade

\[
\begin{bmatrix}
2.4628 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 1.2275 \\
\end{bmatrix}
\]

Daylight

\[
\begin{bmatrix}
2.6836 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 1.5586 \\
\end{bmatrix}
\]

Sony A57K

Tungsten

\[
\begin{bmatrix}
1.6523 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 2.7422 \\
\end{bmatrix}
\]

Shade

\[
\begin{bmatrix}
3.1953 & 0 & 0 \\
0 & 1.0000 & 0 \\
0 & 0 & 1.2891 \\
\end{bmatrix}
\]

Pre-calibrated white-balance matrices for different brands of cameras.
Auto white balance (AWB)

• If manual white balance is not used, then an AWB algorithm is performed.

• AWB must determine the sensor's raw-RGB response to the scene illumination from an arbitrary image.

• AWB is not easy and this remains an open research problem.
AWB is not easy

Given an arbitrary raw-RGB image, determine what is the camera's response to the illumination.

The idea is that something that is white* is a natural reflector of the scene's illuminations SPD.

So, if we can identify what is "white" in the raw-RGB image, we are observing the sensor's RGB response to the illumination.

* It doesn’t have to be "white", but grey – sometimes we call these scene points "achromatic" or "neutral" regions.
AWB: "Gray world" algorithm

• This method assumes that the average reflectance of a scene is achromatic (i.e. gray)
  • Gray is just the white point not at its brightest, so it serves as an estimate of the illuminant
  • This means that image average should have equal energy, i.e. R=G=B
• Based on this assumption, the algorithm adjusts the input average to be gray as follows:

First, estimate the average response:

\[
\begin{align*}
R_{avg} &= \frac{1}{N_r} \sum R_{sensor}(r) \\
G_{avg} &= \frac{1}{N_g} \sum G_{sensor}(g) \\
B_{avg} &= \frac{1}{N_b} \sum B_{sensor}(b)
\end{align*}
\]

- \( r = \text{red pixels values}, \ g = \text{green pixels values}, \ b = \text{blue pixels values} \)
- \( N_r = \# \text{ of red pixels}, \ N_g = \# \text{ of green pixels}, \ N_b = \# \text{ blue pixels} \)

Note: \# of pixel per channel may be different if white balance is applied to the RAW image before demosaicing. Some pipelines may also transform into another colorspace, e.g. LMS, to perform the white-balance procedure.
AWB: "Gray world" algorithm

- Based on the image average R/G/B value, white balance can be expressed as a matrix as:

\[
\begin{bmatrix}
R' \\
G' \\
B'
\end{bmatrix} = \begin{bmatrix}
G_{avg}/R_{avg} & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & G_{avg}/B_{avg}
\end{bmatrix}\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

Matrix scales each channel by its average and then normalizes to the green channel average.
AWB: "White patch" algorithm

• This method assumes that "highlights" (bright spots) represent specular reflections of the illuminant
  • This means that maximum R, G, B values are a good estimate of the white point
• Based on this assumption, the algorithm works as follows:

\[
R_{\text{max}} = \max(R_{\text{sensor}}(r)) \quad G_{\text{max}} = \max(G_{\text{sensor}}(g)) \quad B_{\text{max}} = \max(B_{\text{sensor}}(b))
\]

r = red pixels values, g = green pixels values, b = blue pixels values
AWB: "White patch" algorithm

• Based on RGB max, white balance can be expressed as a matrix as:

\[
\begin{bmatrix}
R' \\
G' \\
B'
\end{bmatrix} = \begin{bmatrix}
G_{\text{max}}/R_{\text{max}} & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & G_{\text{max}}/B_{\text{max}}
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

White-balanced sensor raw-RGB

Matrix scales each channel by its maximum value and then normalizes to the green channel's maximum.

Sensor raw-RGB
AWB example

Input

Gray World

White Patch

White-Balance & Color Space Transform (CIE XYZ)
Better AWB methods

• Gray world and white patch are very basic algorithms.
  • These both tend to fail when the image is dominated by large regions of a single color (e.g. a sky image).

• There are many AWB methods in the literature.
• Camera ISPs often still use simple algorithms with lots of "tuning" . . .
Color space transform – part 2

• Process used on cameras involves interpolation from factory presets.

• The need for interpolation is related to white-balance only approximating true color constancy.

Color space transform is applied after the white balance. In fact, the matrix we use to perform the CST is based on the white-balance CCT.

raw-RGB values → white-balance raw-RGB → WB-raw-RGB mapped to CIE XYZ
Factory pre-calibration

Illumination 1 (CCT 2500 K)

Illumination 2 (CCT 6500 K)

CST matrices ($T_{I_1}$ and $T_{I_2}$) are calibrated for two different illuminations (I1 and I2). Depending on the temperature of the white-balance, we use the corresponding CST.
Lightboxes are used to perform this calibration under different illuminations. Lightboxes are able to reproduce standard illuminants (e.g., D65, incandescent, fluorescent, etc).

X-rite lightbox

GTI lightbox

Telelumen can replace the light source in a lightbox to allow tunable SPDs.
Given a new illumination \((l_a)\) and its estimated correlated color temperature (CCT), we construct a CST matrix by blending the two factory pre-calibrated matrices.

\[
T_{l_a} = gT_{l_1} + (1 - g)T_{l_2}
\]

where \(g = \frac{CCT_{l_a}^{-1} - CCT_{l_2}^{-1}}{CCT_{l_1}^{-1} - CCT_{l_2}^{-1}}\)
Weighting functions

\[ g = \frac{\text{CCT}_{l_1}^{-1} - \text{CCT}_{l_2}^{-1}}{\text{CCT}_{l_1}^{-1} - \text{CCT}_{l_2}^{-1}} \]

\[ T_{l_a} = gT_{l_1} + (1 - g)T_{l_2} \]
A typical color imaging pipeline

- Sensor
- ISO gain and raw-image processing
- RGB demosaicing
- White-balance & color space transform (CIE XYZ)
- Noise reduction
- Save to file
- JPEG/HEIC compression
- Mapping to output color space (e.g., sRGB, P3)
- Color manipulation (Photo-finishing)
- Image rescaling (or up-scaling)
Color manipulation

• This is the stage where a camera applies its "secret sauce" to make the images look good.

• This procedure is called by many names:
  • Color manipulation
  • Photo-finishing
  • Color rendering or selective color rendering
  • Yuv processing engine

• DSLR will often allow the user to select various photo-finishing styles.

• Smartphones often compute this per-image.

• Photo-finishing may also tied to geographical regions!
DSLR "picture" styles

- **Standard**: Glowing prints with crisp finishes. It is the basic color of EOS DIGITAL.
- **Portrait**: For transparent, healthy skin for women and children.
- **Landscape**: Crisp and impressive reproduction of blue skies and green trees in deep, vivid color.
- **Neutral**: Subjects are recorded in rich detail, giving the greatest latitude for image processing.
- **Faithful**: Accurate recording of the subject's color, close to the actual image seen with the naked eye.
- **Monochrome**: Filter work and sepia tone with the freedom of digital monochrome.

From Canon's user manual.
Picture styles

Example of four different picture styles from Nikon
This image is the *same* raw-RGB image processed in four different ways.
Nonlinear color manipulation

Color manipulation can be implemented using a 3D look up table (LUT) and a 1D LUT tone-curve. The 3D LUT table acts like a 3D function: \( f(R, G, B) \rightarrow R', G', B' \)

The 1D LUT table is applied per channel: \( g(R) \rightarrow R', g(G) \rightarrow G', g(B) \rightarrow B' \)

The 3D and 1D LUT can change based on picture style.
Picture styles

Each style has its own 3D LUT and 1D LUT.
Global tone map example (1D LUT)

Input

Enhancing contrast (called an S-curve)

Darkening the image

Brightening the image

Enhancing contrast (called an S-curve)
3D LUT color manipulation visualization

Visualization as a displacement map of a slice of the 3D LUT mapping, warping an input and output value
Local tone mapping (LTM)

NOTE: On many cameras, esp smartphones, a local tone map is applied as part of the photo-finishing. This helps bring out highlights in the image.
Selective color manipulation

• "Select" colors can be manipulated, especially skin tone.
• Sometimes called preferred color correction (PCC)
Color and Imaging Conference (CIC) papers

Examples of papers addressing preferred skin color.

Investigation of Effect of Skin Tone to Facial Attractiveness, Yan Lu¹, Jie Yang¹, Kaida Xiao¹, Michael Pointer¹, Changjun Li², and Sophie Wuerger³; ¹University of Leeds (UK), ²University of Science and Technology Liaoning (China) and ³University of Liverpool (UK)

Preferred Skin Colours Observed by Three Ethnic Groups under different Ambient Lighting Conditions, Mingkai Cao, Ming Ronnier Luo, Rui Peng, Yuechen Zhu, and Xiaoxuan Liu, Zhejiang University, and Guoxiang Liu, Huawei Technologies Co, Ltd. (China)

Preferred Skin Reproduction Centres for Different Skin Groups, Rui Peng, Ming Ronnier Luo, Mingkai Cao, Yuechen Zhu, and Xiaoxuan Liu, State Key Laboratory of Modern Optical Instrumentation, and Guoxiang Liu, Hisilicon (China)

Are We Alike? Skin Color Perception in Portrait Image and AR-based Humanoid Emoji, Yuchun Yan and Hyeon-Jeong Suk, KAIST (South Korea)
Typical color imaging pipeline

ISO gain and raw-image processing

RGB demosaicing

White-balance & color space transform (CIE XYZ)

Noise reduction

JPEG/HEIC compression

Save to file

Mapping to output color space (e.g., sRGB, P3)

Image rescaling (or up-scaling)

Color manipulation (Photo-finishing)
Re-scaling image and sRGB conversion

Often, the entire image is processed by the ISP and then rescaled for the view-finder or to fit the requested output resolution.
Rescale can also be “digital zoom"
Typical color imaging pipeline

Image rescaling (or up-scaling) → Sensor → ISO gain and raw-image processing → RGB demosaicing

→ Color manipulation (Photo-finishing) → White-balance & color space transform (CIE XYZ) → Noise reduction

→ Mapping to output color space (e.g., sRGB, P3) → JPEG/HEIC compression → Save to file
Final sRGB conversion (or other color space)

- Map from *photo-finished* CIE XYZ image to sRGB
- Apply the sRGB (2.2)$^{-1}$ gamma encoding

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} = \begin{bmatrix}
3.2404542 & -1.5371385 & 0.4985314 \\
-0.9692660 & 1.8760108 & 0.0415560 \\
0.0556434 & -0.2040259 & 1.0572252
\end{bmatrix} \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

sRGB is known as an "output-referred" or "display-referred" color space. It is intended for use with display devices.
JPEG compression scheme

Take original image and Break it into 8x8 blocks

8x8 DCT (Forward DCT) on each block

Quantize DCT coefficients via Quantization “Table”

C'(u,v) = round(C(u,v)/T(u,v))

"Zig-zag" Order Coefficients

Differential coding DC component

RLE

AC Vector

JPEG applies almost every compression trick known.
1) Transform coding, 2) psychovisual (loss), 3) Run-length-encoding (RLE), 4) Difference coding, and Huffman.

f(x,y) - 128
(normalize between −128 to 127)
JPEG quality

• The amount of quantization applied on the DCT coefficients amounts to a “quality” factor
  • More quantization = better compression (smaller file size)
  • More quantization = lower quality

• Cameras generally allow a range that you can select
Note: sRGB/JPEG is slowly being replaced

• sRGB was developed for monitors in the 1990s – it is an old standard.
• High Efficient Image Encoding (HEIC)
  • Better compression than JPEG
• Apple iPhone has started to use HEIC to replace JPEG
• HEIC supports multiple color spaces. Apple uses Display P3 – a variation on a Digital Cinema Initiative P3 space.
• The P3 gamut is 25% wider than sRGB
• There is also a gamma encoding similar to sRGB.
Typical color imaging pipeline

1. Sensor
2. ISO gain and raw-image processing
3. RGB demosaicing
4. White-balance & color space transform (CIE XYZ)
5. Noise reduction
6. Mapping to output color space (e.g., sRGB, P3)
7. Color manipulation (Photo-finishing)
8. Image rescaling (or up-scaling)
9. JPEG/HEIC compression
10. Save to file
Circular buffer for ZSL and multi-frame processing

- Many ISPs store the last few RAW images in memory (i.e., a circular buffer).
- These images can be used for many purposes (image stabilization, temporal noise reduction, etc.).
- Another purpose is to ensure “Zero Shutter Lag” (ZSL).
  - There is often a delay between when a person presses the “capture” and the camera capturing the image. This is called “shutter lag”.
  - So, a previous image in the buffer can be saved instead.

You click “save”, but due to shutter lag, an image in the ZSL buffer is saved.

Buffer of RAW images

Current sensor image

\[ I_t, I_{t-1}, I_{t-2}, I_{t-3} \]

Photo saved is actually this image!
ISP hardware will often divide these operations into two components – (1) Bayer-Processing and (2) Image Processing.
Pipeline comments

• Again, it is important to stress that the exact steps mentioned in these notes only serve as a guide to what takes place in a camera.

• Smartphone camera pipelines are more complex.

• Note: for the different camera makes/models, the operations could be performed in a different order and in different ways (e.g., combining sharpening with demosaicing).
What about machine vision cameras?

• Some industrial/machine vision cameras provide minimal ISP processing
• For example, some will only perform white-balance and apply a gamma to the raw-RGB values.
• This means the output is in a camera-specific color space.

Typical machine vision pipeline

ISO gain and raw-image processing → RGB Demoasicing → White-Balance (per-channel gain) → per-channel gamma
The algorithms on an ISP are often predefined. Camera engineers can “tune” the algorithm parameters to produce the output they want. The "tuning" of the ISP is a labor-intensive procedure.
Congratulations!

Certificate of Completion
This Certificate is presented to

You

on this 3 day of Oct 2023

for completion of

“In-camera rendering pipeline”
Part 3:
AI targeting ISP components
A bit more complex ISP

Lens

Sensor

Defective pixel correction
Defective pixel correction/Normalization
Black level correction

Lens shading correction

3As
Auto-exposure
Auto-focus
Auto-white-balance

White-balance

Demosaicing

Noise reduction

Bayer processing routines

shutter duration/ISO (gain)/focus parameters

View-finder or compression/file (JPEG/HEIC)

Color mapping to display-referred color space (sRGB, P3)

Image resizing/super-resolution

Add grain/noise

Local and global tone-mapping

General and selective color manipulation

Color space transform to CIE XYZ/ProPhoto

Photo-finishing routines
Use deep learning for hard problems

The highlighted components are camera pipeline steps that are challenging and areas AI can make notable gains:

- AWB (illumination estimation)
- Demosaicing
- Noise reduction
- Super-resolution
Use deep learning for hard problems

The highlighted components are camera pipeline steps that are challenging and areas AI can make notable gains:

- AWB (illumination estimation)
- Demosaicing
- Noise reduction
- Super-resolution
Digital zoom

A distinguishing feature in the smartphone camera market is zoom quality.
Machine learning (ML) for super-resolution

• SR has been addressed by machine learning methods for a long time.

• Required "training data"
  • Quality of results are directly correlated to training data suitability.

• Before deep learning, used "non-learnable" machine learning.
  • Hand-crafted features
  • Conditional random fields
  • K-Nearest Neighbor
  • Support vector machines
Early example – Freeman 2002

To address the lack of resolution independence in current models, we developed a fast and simple one-pass, training-based super-resolution algorithm for creating plausible high-frequency details in zoomed images.

Example-based Super-Resolution

Super-resolution or image interpolation—these are usually considered to be the highest resolution we can attain, which leads to unnatural images. Techniques such as nearest neighbor interpolation, bilinear interpolation, bicubic interpolation, and Gaussian blur can produce images that look very smooth and lack details. These techniques are often used by computers to render images more efficiently.

Super-resolution algorithms use a combination of image enhancement techniques, such as edge detection, color correction, noise reduction, and image sharpening. These algorithms work by analyzing the low-resolution image and comparing it to a high-resolution reference image. They then use this information to create a new, higher-resolution image that looks more natural.

For example, a super-resolution algorithm might use a search dictionary to find similar low-resolution patches in the input image and replace them with high-resolution patches from the training images. This process is repeated until the entire image has been upscaled.

Training images were small photo collection.

Search dictionary for similar low-res patches, replace with high-res patch.
Enter deep learning

Approach ML problems with learnable processing graphs inspired by biological neurons.
Super-resolution was target of early CNNs

Dong, Loy, He, Tang [ECCV'14]

Let network learn "feature."
Let network learn how to reconstruct.
Early CNN approaches were not always the "best."

Highly hand-crafted methods still worked well.

But, CNN learned everything.
(The care now was in optimizing the CNN).
Super-resolution with very deep networks

Kim, Lee, Lee CVPR’16

Accurate Image Super-Resolution Using Very Deep Convolutional Networks

Jiwon Kim, Jung Kwon Lee and Kyoung Mu Lee
Department of ECE, ASRI, Seoul National University, Korea
{kim, derucci, kyoungm}@snu.ac.kr

Abstract

We present a highly accurate single-image super-resolution (SR) method. Our method uses a very deep convolutional network inspired by VGG-net used for ImageNet classification [19]. We find increasing our network depth shows a significant improvement in accuracy. Our final model uses 20 weight layers. By cascading small filters many times in a deep network structure, contextual information over large image regions is exploited in an efficient way. With very deep networks, however, convergence speed becomes a critical issue during training. We propose a simple yet effective training procedure. We learn residuals only and use extremely high learning rates (10^4 times higher than SRCNN [15]) enabled by adjustable gradient clipping. Our proposed method performs better than existing methods in accuracy and visual improvements in our results are easily noticeable.

1. Introduction

We address the problem of generating a high-resolution image from a single low-resolution input image. In the recent years, a number of techniques have been proposed to address the problem of super-resolution (SR). These methods can be divided into two main categories: cycle-based methods and network-based methods. Cycle-based methods use a series of transformation networks to warp and scale the input image. Network-based methods use deep convolutional neural networks (CNNs) to learn the mapping between low-resolution (LR) and high-resolution (HR) images.

Figure 1: Our VDSR improves PSNR for scale factor ×2 on dataset Set5 in comparison to the state-of-the-art methods (SR-CNN uses the public slower implementation using CPU). VDSR outperforms SRCNN by a large margin (0.87 dB).

- Pairs of convolution layers + nonlinear activations
- Prediction is added to upsampled low-res input
- Special care for gradient clipping
Took "SR" to the next level visually

Ground Truth (PSNR, SSIM) [22] 22.92, 0.7379
A+ (22.90, 0.7332)
RFL [18] (23.00, 0.7439)
SelfEx [11] (23.15, 0.7487)
SRCNN [5] (23.50, 0.7777)
VDSR (Ours)
Adding adversarial loss to SR

In corporate additional loss that considers how realistic the solution is with respect to image distribution.

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi

Twitter

{cledig, ltheis, fhuszar, jocaballero, aacostadas, aaitken, atejani, jtotz, zehanw, wsh}@twitter.com

Abstract

Despite the breakthroughs in accuracy and speed of single image super-resolution using faster and deeper convolutional neural networks, one central problem remains largely unsolved: how do we recover the finer texture details when we super-resolve at large upscaling factors? The behavior of optimization-based super-resolution methods is principally driven by the choice of the objective function. Recent work has largely focused on minimizing the mean squared reconstruction error. The resulting estimates have high peak signal-to-noise ratios, but they are often lacking high-frequency details and are perceptually unsatisfying in the sense that they fail to match the fidelity expected at

1. Introduction

The highly challenging task of estimating a high-resolution (HR) image from its low-resolution (LR) counterpart is referred to as super-resolution (SR). SR received substantial attention from within the computer vision research community and has a wide range of applications [22, 70, 42].

4× SRGAN (proposed)  original

"pixel-wise average of possible solutions"

Natural Image Manifold

MSE-based Solution

GAN-based Solution
SRGAN structure
GAN loss adds another visual level!
AIM/NTIRE SR challenges

New Trends in Image Restoration and Enhancement (NTIRE)

Advances in Image Manipulation (AIM)
• These workshops run regular challenges on super resolution

- Current winning solutions are transformer-based.
- Interestingly, solutions do not use GANs! GANs often help with visual appearance. GANs do not necessarily beat benchmarks. Benchmarks are based on RMSE/SIMM losses.

See CVPR’18 paper – Perception-Distortion Tradeoff
Use deep learning for hard problems

The highlighted components are camera pipeline steps that are challenging and areas AI can make notable gains:

- **AWB (illumination estimation)**
- Demosaicing
- Noise reduction
- Super-resolution
Recall why illumination estimation is hard

What is the sensor's response to illumination?

Given an arbitrary input image, predict the scene illumination.

Getting this *incorrect* has significant impact on image quality/color reproduction.
Many ML approaches before deep learning

Cheng et al CVPR'15

Derive some features (usually histogram statistics)

Apply ML method to predict illumination of scene.

Effective Learning-Based Illuminant Estimation Using Simple Features

Dongliang Cheng¹ Brian Price² Scott Cohen² Michael S. Brown¹
¹National University of Singapore 
{dcheng, brown}@comp.nus.edu.sg
²Adobe Research 
{price, scohen}@adobe.com

Abstract

Illumination estimation is the process of determining the chromaticity of the illumination in an imaged scene in order to remove undesirable color casts through white-balancing. While computational color constancy is a well-studied topic in computer vision, it remains challenging due to the ill-posed nature of the problem. One class of techniques relies on low-level statistical information in the image color distribution and works under various assumptions (e.g. Grey-World, White-Patch, etc.). These methods have an advantage that they are simple and fast, but often do not perform well. More recent state-of-the-art methods employ learning-based techniques that produce better results, but often rely on complex features and have long evaluation and training times. In this paper, we present a learning-based method based on four simple color features and show how to use this with an ensemble of regression trees to estimate the illumination. We demonstrate that our approach is not only faster than existing learning-based methods in terms of both evaluation and training time, but also gives the best results reported to date on modern color constancy data sets.

1. Introduction and Related Work

An RGB image captured by a camera is a combination of illumination and scene color. When the illumination is not sufficiently white (e.g. daylight), this can cause a notable color cast in the image. One of the key pre-processing steps applied to most images is to remove color casts caused by illumination to achieve an approximation to the true scene color.
Improving AWB with CNN

Hu et al. CVPR'17

Predicts local estimates over the image and their confidence. Pools confident weighted estimates for final result.
What did it learn?

The method appears to learn to identify pixels that are most likely "neutral/achromatic" scene patches.
Exploiting two cameras for AWB

Abdelhamed et al CVPR’21

Leveraging the Availability of Two Cameras for Illuminant Estimation

Abdelrahman Abdelhamed  Abhijith Punnappurath  Michael S. Brown
Samsung AI Center – Toronto
{a.abdelhamed, abhijith.p, michael.h}@samsung.com

Abstract
Most modern smartphones are now equipped with two rear-facing cameras – a main camera for standard imaging and an additional camera to provide wide-angle or telephoto zoom capabilities. In this paper, we leverage the availability of these two cameras for the task of illumination estimation using a small neural network to perform the illumination prediction. Specifically, if the two cameras’ sensors have different spectral sensitivities, the two images provide different spectral measurements of the physical scene. A linear 3x3 color transform that maps between these two observations – and that is unique to a given scene illuminant – can be used to train a lightweight neural network comprising no more than $1460$ parameters to predict the scene illumination. We demonstrate that this two-camera approach with a lightweight network provides results on par or better than much more complicated illuminant estimation methods operating on a single image. We validate our method’s effectiveness through extensive experiments on radiometric data, a quasi-real two-camera dataset we generated from an existing single camera dataset, as well as a new real image dataset that we captured using a smartphone with two rear-facing cameras.

Two views of the same scene with different sensors – is essentially a 6-channel camera.

1. Introduction
An overwhelming percentage of consumer photographs
Exploiting two cameras for AWB

State-of-the-art results with very lightweight network
Use deep learning for hard problems

The highlighted components are camera pipeline steps that are challenging and areas AI can make notable gains:

- AWB (illumination estimation)
- Demosaicing
- Noise reduction
- Super-resolution
Demosaicing role is to interpolate 2/3 (66%) of your sensor image!

Captured raw-Bayer image
DNN for demosaicing (and denoising)

Gharbi et al SIGGRAPH Asia, 2016

- Early work found that you got denoising for free.
- Network similar to SR-residual.
- Training data/experimental data 100% synthetic.
Deep CNN for demosaicing

Examined #1 – shallow network with deep channels (SRNet)

Examined #2 - deep layers with residual (SR-ResNet)

Syu et al - arXiv 2018
Paper showed that Deep SR-ResNet is a good DNN for demosiacing too.
More recent approach

Considers the following:

(1) Demosaicing is hardest in high-frequency areas

Liu et al CVPR’21 [Huawei]

(2) Green channel most reliable (since we have a lot of green)
Demosiacing with "self guidance"

Guidance map

Green channel upsampling trained with its own loss
Produce nice visual results

Self-guidance NET
Use deep learning for hard problems

The highlighted components are camera pipeline steps that are challenging and areas AI can make notable gains:

- AWB (illumination estimation)
- Demosaicing
- **Noise reduction**
- Super-resolution
Non-deep-learning noise reduction

- One of the best-performing methods was based on non-local means (2007).
- Block-matching with 3D filtering [BM3D]
- It is slow, but works well.

---

Dabov et al TIP'07

Abstract: We propose a novel image denoising strategy based on an enhanced sparse representation in transform domain. The enhancement of the capability is achieved by grouping similar 2D image fragments (e.g., blocks) into 2D data arrays which we call "grids". Collaborative filtering is a special procedure developed to deal with these 2D grids. We realize it using the three successive steps: 3D transformation of a group, shrinkage of the transform spectrum, and inverse 2D transformation. The result is a 2D volume that consists of the jointly shrunk grouped image blocks. By attenuating the noise, the collaborative filtering reveals even the finest details hidden in grouped blocks and at the same time preserves the essential unique features of each individual block. The shrunk blocks are then restored to their original positions. Because these blocks are overlapping, for each pixel we obtain many different estimates which need to be combined. Aggregation is a particular averaging procedure which is exploited to take advantage of this redundancy. A novel nonlocal collaborative Wiener filtering is also developed. The experimental results demonstrate that this computationally scalable algorithm achieves state of the art denoising performance in terms of both peak signal-to-noise ratio and subjective visual quality.

1. Introduction

Plenty of denoising methods exist, originating from various disciplines such as probability theory, statistics, partial differential equations, linear and nonlinear filtering, spectral and multiresolution analysis. All these methods rely on some implicit or explicit assumptions about the noise-

For small reference patch R, find similar patches.
Average the patches.
DNN for denoising (DnDNN)

- Straight-forward network based on deep residual learning (Kim SR-ResNet).

- Introduced batch normalization to the network.

- Predicts the residual noise layer.

Zhang et al. TIP’17

Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang

Abstract—Discriminative model learning for image denoising has been recently attracting considerable attentions due to its favorable denoising performance. In this paper, we take one step forward by investigating the construction of end-to-end denoising convolutional neural networks (DnCNNs) to enhance the progress in very deep architectures, learning algorithm, and regularization method for image denoising. Specifically, residual learning and batch normalization are utilized to speed up the training process as well as to boost the denoising performance. The main new insight in our work is that we train a single end-to-end DnCNN model to tackle several general image denoising tasks, such as Gaussian denoising, single image super-resolution and JPEG image deblurring. The experimental results demonstrate that our DnCNN model can not only exhibit high effectiveness in several general image denoising tasks, but also can be efficiently implemented by utilizing GPU computing.

Index Terms—Image Denoising, Convolutional Neural Networks, Residual Learning, Batch Normalization

I. INTRODUCTION

Image denoising is a classical yet still active topic in low level vision since it is an indispensable step in many practical applications. The goal of image denoising is to recover a clean image from a noisy observation $y$ which follows an image degradation model $y = x + v$. One common assumption is that $v$ is additive white Gaussian noise (AWGN) with standard random field (RF) models $\mathcal{M}(n, \mathcal{D}_n)$. In particular, the NSR models are popular in state-of-the-art methods such as BMD [1], LSN [2], NSR [3] and WNSN [4]. Despite their high denoising quality, most of the image prior-based methods typically suffer from two major drawbacks. First, these methods generally involve a complex optimization problem in the testing stage, making the denoising process time-consuming [5, 6, 7, 8, 9]. Thus, most of the prior-based methods can barely achieve high performance without sacrificing computational efficiency. Second, the models in general are non-convex and involve several manually chosen parameters, providing some way to lose denoising performance.

To overcome the limitations of prior-based approaches, several discriminative learning methods have been recently developed to learn prior image models in the context of supervised inference procedures. The resulting models are able to get rid of the time-consuming optimization procedures in test phase. Schmidt and Roth [10] proposed a cascade of shrinkage fields (CSF) method that unifies the random field-based model and the variational half-quadratic optimization algorithm into a single learning framework. Chen et al. [11, 12] proposed a trainable nonlinear reaction diffusion (TNRD) model which learns a modified set of experts $\mathcal{E}$ image prior by enforcing a fixed number of gradient descent inference steps. Some of the other related work can be found in [11, 12]. Although CSF and TNRD have shown promising results toward bridging the gap between computational efficiency and denoising quality, their performance are inherently restricted to the specific forms of prior. To overcome these drawbacks, we train an end-to-end SCR which contains the following layers:...
DnCNN result

<table>
<thead>
<tr>
<th>Methods</th>
<th>BM3D</th>
<th>WNNM</th>
<th>EPLL</th>
<th>MLP</th>
<th>CSF</th>
<th>TNRD</th>
<th>DnCNN-S</th>
<th>DnCNN-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma = 15$</td>
<td>31.07</td>
<td>31.37</td>
<td>31.21</td>
<td>-</td>
<td>31.24</td>
<td>31.42</td>
<td>31.73</td>
<td>31.61</td>
</tr>
<tr>
<td>$\sigma = 25$</td>
<td>28.57</td>
<td>28.83</td>
<td>28.68</td>
<td>28.96</td>
<td>28.74</td>
<td>28.92</td>
<td>29.23</td>
<td>29.16</td>
</tr>
<tr>
<td>$\sigma = 50$</td>
<td>25.62</td>
<td>25.87</td>
<td>25.67</td>
<td>26.03</td>
<td>-</td>
<td>25.97</td>
<td>26.23</td>
<td>26.23</td>
</tr>
</tbody>
</table>

- Method trained on synthetic noise data.
- Beats BM3D and is much faster.
- BM3D does not require training data!
Need for real denoising dataset

Abdelhamed et al. CVPR 2018

**SIDD:** Smartphone Image Denoising Dataset

- 30,000 images
- 5 cameras
- 160 scene instances
- 15 ISO settings
- Direct current lighting
- Three illuminations

**Interesting finding**

- When trained on synthetic only, BM3D beat DnCNN
- When trained on real data, DnCNN wins
- Implies noise models in literature are not accurate
Denoising contest at CVPR'20

- Winning solutions (Baidu) relied on neural architecture search
- Samsung method used a U-net + multi-scale residual nets
ISPs with multi-frame (burst) imaging
Why multi-frame?

• Two primary applications where is currently multi-frame is used
  • Low-light imaging (or "night mode" or "extended ISO")
  • High-dynamic-range (HDR) imaging

• Many ISP now support multi-frame image.

• Possible to do super-res with multi-frame.
  • For time sake, I won't discuss those methods
ISP with multi-frame

Sensor

Defective pixel correction

Black level correction/Normalization

Lens shading correction

Lens shading correction

Black level correction/Normalization

Auto-exposure

Auto-focus

Auto-white-balance

White-balance

Demosaicing

Align

Multi-frame/Burst unit

Multi-frame/Burst unit

Multi-frame/Burst unit

Demosaicing

Align

Fuse

View-finder or compression/file (JPEG/HEIC)

Color mapping to display-referred color space (sRGB, P3)

Image resizing/super-resolution

Add grain/noise

Local and global tone-mapping

General and selective color manipulation

Color space transform to CIE XYZ/ProPhoto

Photo-finishing routines

Super-resolution

HDR

Low-light

Bayer processing routines

shutter duration/ISO (gain)/focus parameters

3As
ISP with multi-frame

Sensor

Defective pixel correction

Black level correction/Normalization

Lens shading correction

3As
  Auto-exposure
  Auto-focus
  Auto-white-balance

White-balance

Demosaicing

Denoising

Lens shading correction

Auto-exposure

Auto-focus

Auto-white-balance

Multi-frame/Burst unit

Color mapping to display-referred color space (sRGB, P3)

View-finder or compression/file (JPEG/HEIC)

Super-resolution

Image resizing/
super-resolution

Add grain/noise

Local and global tone-mapping

General and selective color manipulation

Color space transform to CIE XYZ/ProPhoto

HDR

Multi-frame/Burst unit

Low-light

Demosaicing

Align

Fuse

Multi-frame/Burst unit

Photo-finishing routines

shutter duration/ISO (gain)/focus parameters

3As

Auto-exposure

Auto-focus

Auto-white-balance

Demosaicing

Add grain/noise

Local and global tone-mapping

General and selective color manipulation

Color space transform to CIE XYZ/ProPhoto
Low-light imaging is essentially a noise-reduction problem.
Multi-frame for low-light

Early work on low-light image is from Samsung SAIT

Moon et al – ICCE 2013

A Fast Low-Light Multi-Image Fusion with Online Image Restoration
Young-Su Moon, Shi-Hwa Lee, Yong-Min Tai, and Jungak Cho
Samsung Advanced Institute of Technology, Samsung Electronics, Korea

Abstract—This paper presents a new low-light multi-frame fusion algorithm to get a bright and clear shot even under dark conditions. To this end, using multiple short-exposure images and one proper-exposure blurry image as an input, a new hierarchical block-wise temporal noise filtering is done. Finally, an online image restoration of the denoising result is conducted along with the blurry image input. Test results on real low-light scene show its effectiveness like fast processing speed and satisfactory visual quality.

I. INTRODUCTION

Digital camera photos taken under a low-light condition reveal significant image artifacts such as motion blur by long-exposure shooting or strong noise corruption by High-ISO setting. Furthermore, as camera sensor’s resolution increases, such artifacts are getting worse due to lack of incoming lights on each sensor cell.

To solve it, many research works have been studied. In a multi/short exposure shooting mode need to be geometrically aligned. To achieve this effectively, global image motion between a reference short-exposure input image and other short-exposure input images is estimated with a fast and effective method using a translation model [4]. For convenience, the first short-exposure input image is selected as the reference. Since actual image motion between the input frames is complicated, subsequent block-based local motion estimation is required.

Proposed alignment
(1) Works on a Laplacian pyramid.
(2) global motion alignment
(3) local motion correction
(4) Temporal fusion
  (to avoid ghosting)
Burst for low-light and HDR

No deep learning – this paper describes a fast/robust alignment estimation based on bilateral fitter. Robust merging is guided by a reference frame. All is done in the Bayer/RAW frame.

High-dynamic-range is in terms of bit depth. This paper claims the denoising/fusion can upsample from 10bit to 12bit.
Multi-frame for low light

Learns a multi-scale burst encoder/decoder framework.
Input is RAW, output is sRGB.

Each encoder is at a different resolution scale.
Google pixel phones multi-frame

Wronski et al SIGGRAPH 2019

Not necessarily for low-light, but does target RAW.
This paper uses multiple frames and very small camera motion (from hand tremors) to perform demosaicing and super-resolution. By exploiting motion, they can fill in missing Bayer data too.
ISP with multi-frame

- Sensor
  - Defective pixel correction
  - Black level correction/Normalization
  - Lens shading correction
  - 3As
    - Auto-exposure
    - Auto-focus
    - Auto-white-balance
  - White-balance
  - Demosaicing
  - Align
  - Fuse

- Lens
  - shutter duration/ISO (gain)/focus parameters

- Bayer processing routines
  - White-balance
  - Demosaicing

- Multi-frame/Burst unit
  - 3As
  - Auto-exposure
  - Auto-focus
  - Auto-white-balance

- Super-resolution
  - Image resizing/super-resolution
  - Add grain/noise
  - Local and global tone-mapping
  - General and selective color manipulation
  - Color space transform to CIE XYZ/ProPhoto

- Photo-finishing routines
  - View-finder or compression/file (JPEG/HEIC)
  - Color mapping to display-referred color space (sRGB, P3)
An f-stop adjusts the amount of light that falls on the sensor generally by a factor of 2. So, a +1 f-stop increases the amount of light by two times. An -1 f-stop reduces the amount of light by \( \frac{1}{2} \). We assume the ISO is not adjusted.

This is often called an Exposure Value adjustment. Change of EV is a change in stop.
Exposure fusion


Exposure Fusion

Tom Mertens\textsuperscript{1} \quad Jan Kautz\textsuperscript{1} \quad Frank Van Reeth\textsuperscript{1}
\textsuperscript{1}Hasselt University - EDM
transnationale Universiteit Limburg
Belgium
\textsuperscript{2}University College London
UK

Abstract

We propose a technique for fusing a bracketed exposure sequence into a high quality image, without converting to
HDR first. Skipping the physically-based HDR assembly step simplifies the acquisition pipeline. This avoids camera
response curve calibration and is computationally efficient. It also allows for including flash images in the sequence.
Our technique blends multiple exposures, guided by simple quality measures like saturation and contrast. This is done
in a multiresolution fashion to account for the brightness variation in the sequence. The resulting image quality is
comparable to existing tone mapping operators.

1. Introduction

Digital cameras have a limited dynamic range, which is
lower than one encounters in the real world. In high
dynamic range scenes, a picture will often turn out to be under-
or overexposed. A bracketed exposure sequence \cite{5, 17, 26}
allows for acquiring the full dynamic range, and can be
mated into a single high dynamic range image. Upon dis-
play, the intensities need to be remapped to match the typ-
ically low dynamic range of the display device. Through a

Simple method that fused multiple exposed (and rendered)
images to a single ‘fused’ output.

Works on Laplacian pyramid.

Proposed heuristics for determining weights for fusion.
- Namely: saturation, contrast, "exposedness" at each level
Exposure fusion gave spectacular results compared to existing methods in 2007.

Simple algorithm makes it suitable for real-time deployment on device.
Paper examined three strategies.

1. Multi-frame and CNN to predict final HDR.
2. Multi-frame and CNN to predict blending weights, then HDR.
3. Multi-frame and CNN to predict blending weights and align misaligned regions

Found that (#2) is the best; (3) works for small motions.
Summary

• Deep learning is good at addressing hard ISP components
  • Demosiacing, denoising, AWB, super-resolution (digital zoom)
  • These are components that are ill-posed problems (many-to-one solutions)

• Hand-crafted solutions still work well

• GANs shows promise for visual results (not necessarily benchmarks)

• Many current SOTA solutions are based on neural architecture search (NAS)
Part 2: 
AI-based ISPs
Replacing the conventional ISP

- Conventional Image Signal Processor (ISP)
- Replace with a single DNN
- Replace with modular DNNs
Single DNN replacement
Modeling camera rendering with a DNN

Nam and Kim CVPR'17

- The paper is motivated by "reversing" the ISP from sRGB to RAW

- Addresses the scene-dependent nature of ISPs for radiometric calibration

- However, the framework can be used for "forward rendering" (RAW to sRGB)

- This work is often overlooked due to the focus on radiometric calibration
This paper works on image patches and is trained per camera.

To encode local context information, a learnable histogram feature is used and polled at different scales.

The local histogram feature provides spatial context for converting from RAW to sRGB (or sRGB to RAW).
Modeling camera with DNN

Results of rendering RAW to sRGB.
ISP replacement to mimic better camera

CVPRW’19 (NTIRE workshop)

Replacing Mobile Camera ISP with a Single Deep Learning Model

Andrey Ignatov
andrey@vision.ee.ethz.ch

Luc Van Gool
vangool@vision.ee.ethz.ch

Radu Timofte
timofte@vision.ee.ethz.ch

ETH Zurich, Switzerland

Abstract

As the popularity of mobile photography is growing constantly, lots of efforts are being invested now into building complex hand-crafted camera ISP solutions. In this work, we demonstrate that even the most sophisticated ISP pipelines can be replaced with a single end-to-end deep learning model trained without any prior knowledge about the sensor and optics used in a particular device. For this, we present PyNET, a novel pyramidal CNN architecture designed for fine-grained image restoration that implicitly learns to perform all ISP steps such as image denoising, demosaicing, white balancing, color and contrast correction, deblocking, etc. The model is trained to convert RAW Bayer data obtained directly from mobile camera sensor into photos captured with a professional high-end DSLR camera, making the solution independent of any particular mobile ISP implementation. To validate the proposed approach on the real data, we collected a large-scale dataset consisting of 10 thousand full-resolution RAW-RGB image pairs captured in the wild with the Huawei P20 camera phone (12.3 MP Sony Exmor IMX380 sensor) and Canon 5D Mark IV DSLR. The experiments demonstrate that the proposed solution improves the quality of mobile ISP artifacts under both flat and learned conditions.
ISP replacement to mimic better camera

Images are globally aligned, and then patch wise aligned.

Additional perceptual loss (VGG) is included in training at different U-net scales.

Training images
RAW from smartphone
sRGB from DSLR
Images are misaligned!
ISP replacement to mimic better camera
This paper is essentially a learned ISP. However, it learns to process noisy RAW to clean sRGB.
"Learning to see in the dark"

Key to this paper is the careful alignment of data.

Results show for very low-light cases so significant performance.
CRISPnet (color reproduction ISP)

Souza and Heidrich, Arxv 2021

CRISPnet: Color Rendition ISP Net

Matheus Souza and Wolfgang Heidrich
King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
{matheus.meioirodesouza,wolfgang.heidrich}@kaust.edu.sa

Abstract. Image signal processors (ISPs) are historically grown legacy software systems for reconstructing color images from noisy raw sensor measurements. They are usually composed of many heuristic blocks for denoising, demosaicking, and color restoration. Color reproduction in this context is of particular importance, since the raw colors are often severely distorted, and each smartphone manufacturer has developed their own characteristic heuristics for improving the color rendition, for example of skin tones and other visually important colors.

In recent years there has been strong interest in replacing the historically grown ISP systems with deep learned pipelines. Much progress has been made in approximating legacy ISPs with such learned models. However, so far the focus of these efforts has been on reproducing the structural features of the images, with less attention paid to color rendition. Here we present CRISPnet, the first learned ISP model to specifically target color rendition accuracy relative to a complex, legacy smartphone ISP. We achieve this by utilizing both image metadata (like a legacy ISP would), as well as by learning simple global semantics based on image classification — similar to what a legacy ISP does to determine the same type. We also contribute a new ISP image dataset consisting of both high dynamic range monitor data, as well as real-world data, both captured with an actual cell phone ISP pipeline under a variety of lighting conditions, exposure times, and gain settings.

Keywords: image signal processor; image restoration; color rendition.
Motivation is that one day we will have RAW images from a smartphone, but no hardware to render it, so a DNN will be used instead. The paper refers to this as legacy ISPs.

Some interesting ideas:
(1) WB (from RAW) is injected into network layers.
(2) Global semantics is also incorporated into architecture.

Training data is Apple iPhone images. This DNN essentially learns to "render" RAW like Apple.

Results against other DNN ISPs.
A two stage DNN-based ISP
Two stage ISP

Proposes a "restore-net" and "enhance-net".
CameraNet considers real ISP stages

**Bayer processing routines**

1. Defective pixel correction
2. Black level correction/Normalization
3. Lens shading correction
4. 3As (Auto-exposure, Auto-focus, Auto-white-balance)
5. White-balance
6. Demosaicing
7. Noise reduction

**Photo-finishing routines**

1. Image resizing/super-resolution
2. Add grain/noise
3. Local and global tone-mapping
4. General and selective color manipulation
5. Color space transform to CIE XYZ/ProPhoto

- View-finder or compression/file (JPEG/HEIC)
- Color mapping to display-referred color space (sRGB, P3)

**Flowchart Diagram**
CameraNet – Training data preparation

Details are not very clear, but Photoshop is claimed to be used to process RAW to denoised RAW. Lightroom is used to generate enhanced images. Assumed trained per sensor type.
CameraNet

RestoreNet/EnhanceNet share similar structure.

EnhanceNet uses a dilated convolution.

Fig. 4. The structure of U-Net-like Restore-Net and Enhance-Net modules in the proposed CameraNet system.
Results

(a) Raw image  
(b) Result by one-stage setting  
(c) Result by two-stage setting  
(d) Groundtruth
Three stage DNN-based ISP
Winner of the night photography challenge (2022)

**NTIRE 2022 Challenge on Night Photography Rendering**

Egor Erslov, Alex Sarchik, Denis Shepelev, Nikola Banic, Michael S. Brown, Radu Timofte, Karlo Kolčević, Michael Freeman, Vasily Tolsin, Dmitriy Bochov, Illya Semenkov, Marko Subašić, Sven Lendrulić, Arseniy Terekhin, Shuai Liu, Chaoyu Peng, Hao Wang, Ran Zhu, Yongjiang Li, Lei Lei, Zhihao Li, Si Yi, Ling-Hao Han, Ruiqi Wu, Xin Jin, Chunle Guo, Furkan Kını, Sami Mentes, Barış Özcan, Furkan Kınar, Simone Bianco, Raimondo Schettini, Wei Li, Yipeng Ma, Tao Wang, Rui Kang Xu, Fenglong Song, Wei-Ting Chen, Hao-Hsiang Yang, Zhi-Kai Huang, Huay-Chang, Sy-Yen Kao, Zixin Liang, Shangchen Zhou, Ruicheng Feng, Chongyi Li, Xiangyu Chen, Binbin Song, Shile Zhang, Lin Liu, Zhendong Wang, Dohoon Ryu, Hyukyoung Bae, Taesung Kwon, Chaitra Desai, Nikhil Akalwadi, Anmog Joshi, Chinmaye Mandi, Sampada Malagi, Akash Uppin, Sai Sudheer Reddy, Ramesh Ashok Tabb, Ujwala Patil, Uma Madhavagudi

**Abstract**

This paper reviews the NTIRE 2022 challenge on night photography rendering. The challenge solicited solutions that processed RAW camera images captured in night scenes to produce a photofinished output image encoded in the standard RGB (sRGB) space. Given the subjective nature of this task, the proposed solutions were evaluated based on the mean opinion of viewers asked to judge the visual appearance of the results. Michael Freeman, a world-renowned photographer, further ranked the solutions with the highest mean opinion scores. A total of 13 teams competed in the final phase of the challenge. The proposed methods provided by the participating teams represent state-of-the-art performance in nighttime photography. Results from the various teams can be found here: https://nightimaging.org/

**NIGHT PHOTOGRAPHY RENDERING CHALLENGE**

Welcome to the "Night Photography" challenge part of the NTIRE workshop at CVPR 2022.

**NEWS AND UPDATES**

- Teams were asked to process night RAW images to sRGB
- Toloka was used to evaluate results.
- Professional photographer Michael Freeman also evaluated.
- Winning team was from Xiaomi (net slide)
Liu et al. NTIRE’22/CVPRW’22 (Xiaomi)

Deep-FlexISP: A Three-Stage Framework for Night Photography Rendering

Shuai Liu  Chaoyu Feng  Xiaotao Wang  Hao Wang  Ran Zhu  Yongqiang Li  Lei Lei
Xiaomi Inc., China
{liu.shuai21, fengchaoyu, wangxiaotao, wanghao05, zhuruan, liyongqiang, leileil1}@xiaomi.com

Abstract

Night photography rendering is challenging due to images’ high noise level, low vivid color, and low dynamic range. In this work, we propose a three-stage cascade framework named Deep-FlexISP, which decomposes the ISP into three weakly correlated sub-tasks: raw image denoising, white balance, and Color-to-RGB mapping, for the following considerations. First, task decomposition can enhance the learning ability of the framework and make it easier to converge. Second, weak correlation sub-tasks do not influence each other too much, so the framework has a high degree of freedom. Finally, noise, color, and brightness are essential for night photography. Our framework can flexibly adjust different styles according to personal preferences with the real learning ability and the degree of freedom. Compared with the other Deep-ISP methods, our proposed Deep-FlexISP shows state-of-the-art performance and achieves first place in people’s choice and photographer’s choice in NTIRE 2022 Night Photography Render Challenge.

1. Introduction

Night photography is a challenging task due to several reasons. First, the low light condition will cause high-level noise in the raw image. Second, it is hard to estimate the
FlexISP

Custom denoiser. Training data unclear (possibly in-house Xiaomi denoiser used to generate ground truth).
Network was conditioned in noise level. Allowing adjustment.

F4C was used for white-balance. Two networks were used, each predicting biased results towards warm/cold ground truth. User can "slide" between results.

Images were manually adjusted (lightroom?) at different levels. Users could "slide" between results.
Baseline was a simple software ISP given to participants.

PyNet is single DNN method.

HERN (Enhancement network)

FlexISP.
Misc: RAW to bits
RAW to bit

RAWtoBit: A Fully End-to-end Camera ISP Network

Woosok Jeong and Seung-Won Jung*
Department of Electrical Engineering, Korea University, Seoul, Korea
{561woed, svjung83}@korea.ac.kr

Abstract. Image compression is an essential and last processing unit in the camera image signal processing (ISP) pipeline. While many studies have been made to replace the conventional ISP pipeline with a single end-to-end optimized deep learning model, image compression is barely considered as a part of the model. In this paper, we investigate the designing of a fully end-to-end optimized camera ISP incorporating image compression. To this end, we propose RAWtoBit network (RBN) that can effectively perform both tasks simultaneously. RBN is further improved with a novel knowledge distillation scheme by introducing two teacher networks specialized in each task. Extensive experiments demonstrate that our proposed method significantly outperforms alternative approaches in terms of rate-distortion trade-off.

Combined RAW to sRGB with encoder.

Encoder/Decoder (sRGB) [compression]
Paper shows that a knowledge distillation strategy is best to learn the RAW to sRGB with bit encoder.
DNN-based ISP considerations and challenges
Training data

- It is important to remember that RAW images are sensor-specific
  - This means we often need to train ISPs (and ISP modules) per sensor
  - Modern smartphones can have 3-4 different sensors
  - Capturing training data can be overwhelming for camera engineers

- Care is required when capturing training data
  - Many of the low-light/HDR papers, the real contribution is the carefully captured training data
  - Again, this needs to be captured "per" sensors

- Single stage ISPs have limited "tune-ability"
  - Conventional IPS are designed to be tunable
  - DNNs are often tuned by changing the training data
Consideration for DNN-based ISP

• A conventional ISP is still required to produce training data
• Can we beat conventional ISPs?

Current ISP workflow

Team of Image Quality Engineers tune ISP parameters to produce desired images.

Neural-ISP workflow

Team of Image Quality Engineers process thousand of RAW images with a "software ISP" to produce training data?
Tutorial summary

• Background on color and color spaces
  • This topic mixes many disciplines
  • Color constancy and terminology for illumination (e.g. color temperature)

• Overview of basic steps on camera pipeline

• Discussion of more modern multi-frame methods

• Discussion of some recent AI-based methods
I hope you have learned more about color and the in-camera rendering pipeline.

I encourage you to state your assumptions about your image's color space in your research papers:

For example, replace this: "Our input is an RGB image …"

to: "Our input is an RGB image encoded in standard RGB…”

Such a small clarification in your paper will greatly help other researchers.
Acknowledgements
And of course...

“The Standard Observers”