Siggraph Course 2017
Path Tracing in Production
Part 1

Manuka: Weta Digital's Spectral Renderer

Johannes Hanika, Weta Digital

Motivation

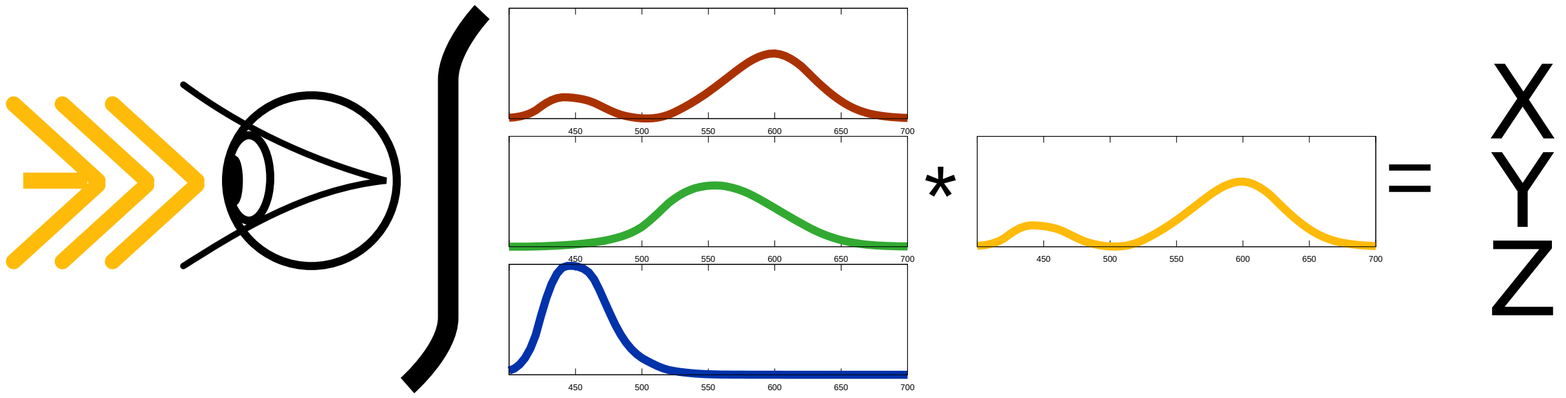
- Weta Digital is a VFX house
- we care about matching plate a lot
- natural and supernatural phenomena
  - both usually means physically-based light transport
- we use our in-house renderer:

Outline of this talk

- colour reproduction
- spectral rendering/sampling
- colour management/textures
- advantages of spectral sampling

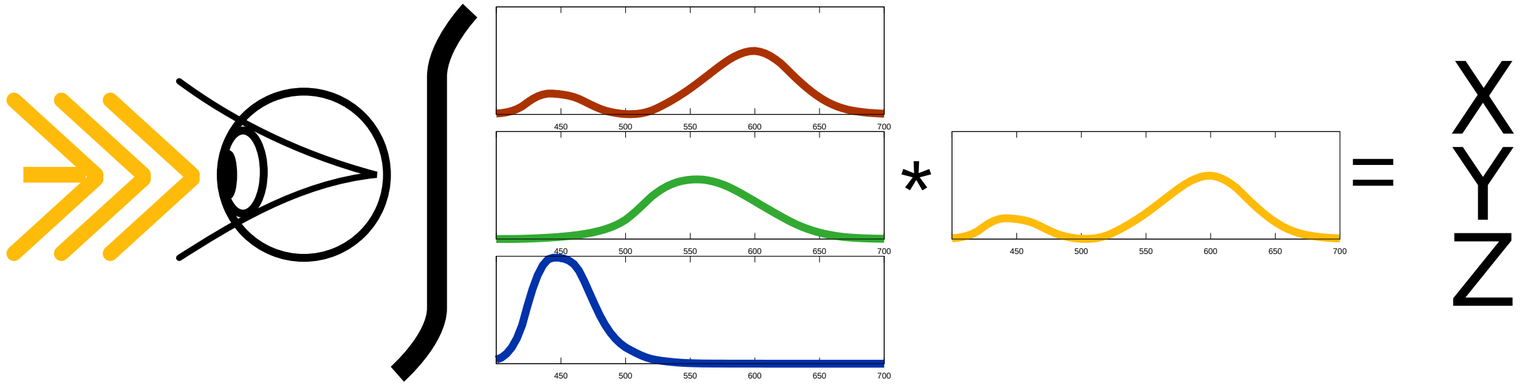
What does the eye see?

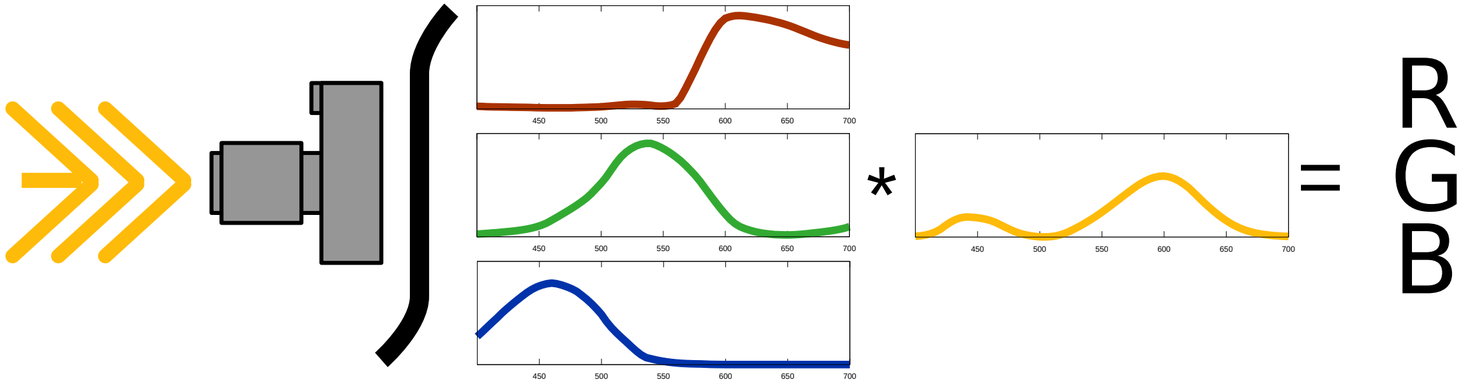
- standardised process, we are going to work under the assumption this is true
- incoming light is projected to the 1931 2-degree XYZ colour matching functions



What does a camera see?

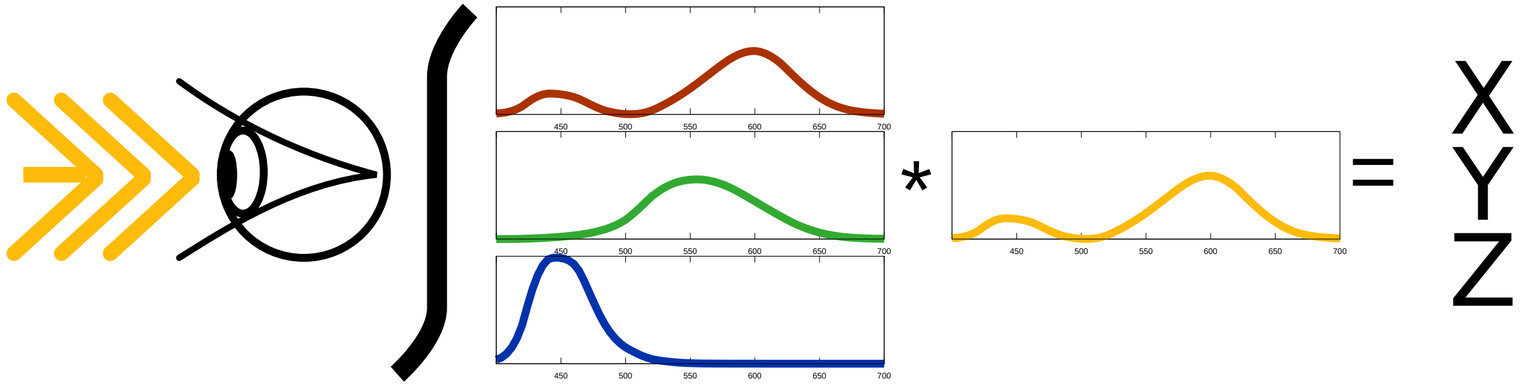
- analogous to the eye: project to colour filter array spectra
- CFA varies a lot with the device

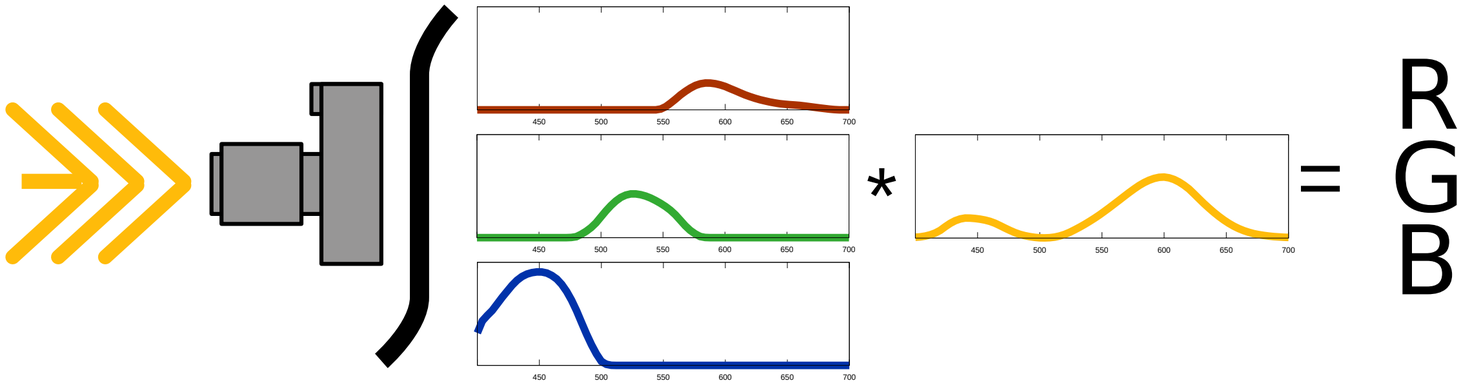




What does a camera see?

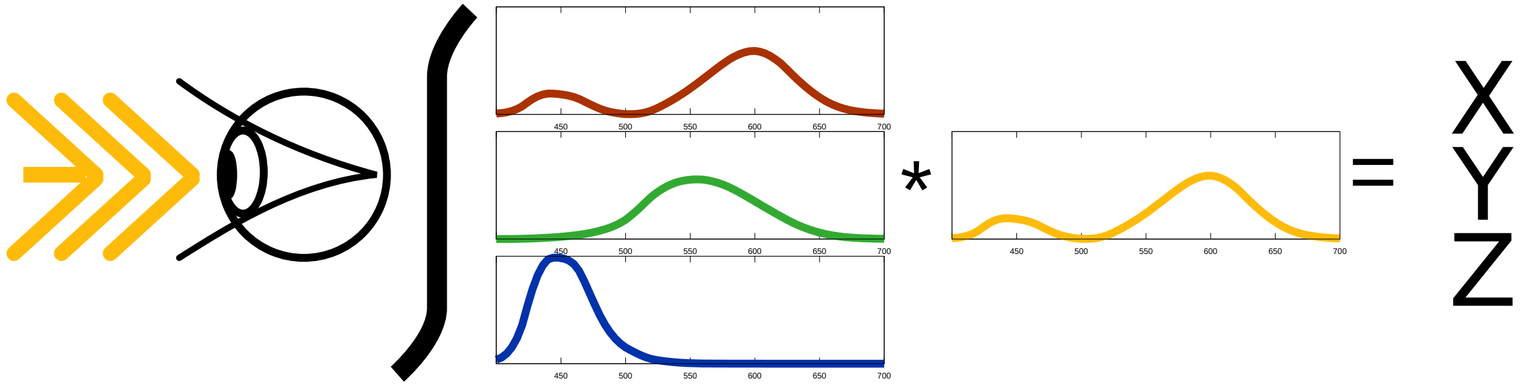
- analogous to the eye: project to colour filter array spectra
- CFA varies a lot with the device

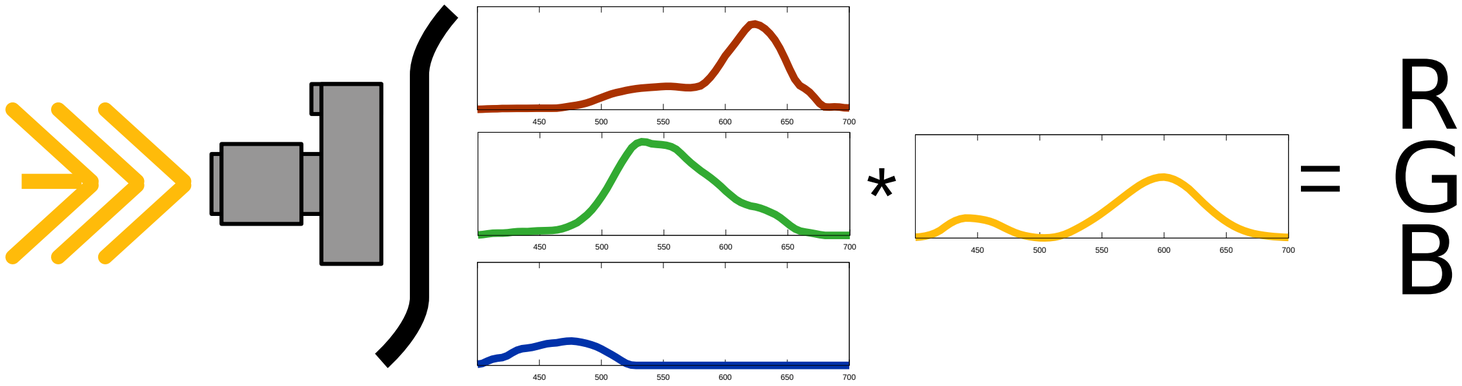




What does a camera see?

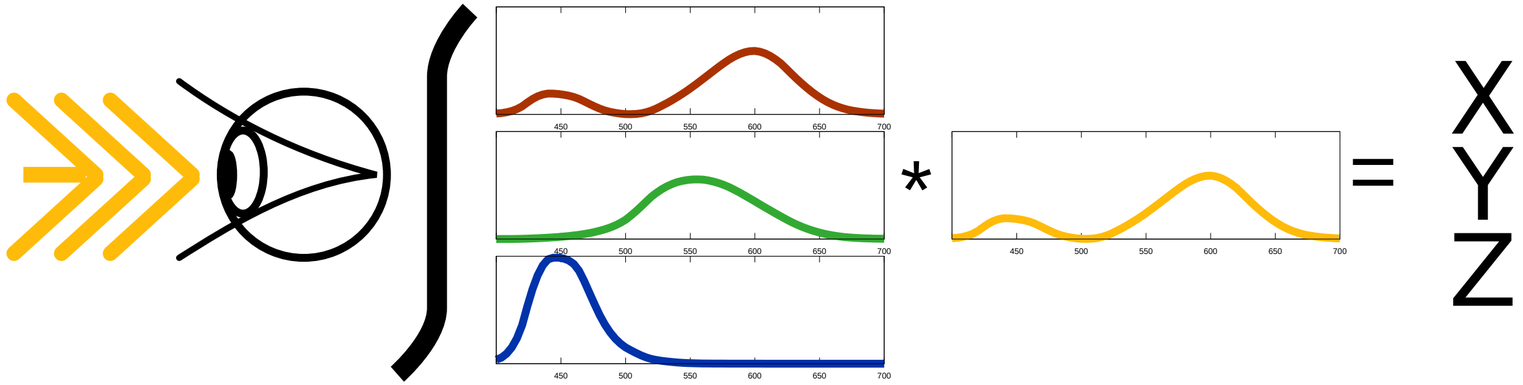
- analogous to the eye: project to colour filter array spectra
- CFA varies a lot with the device

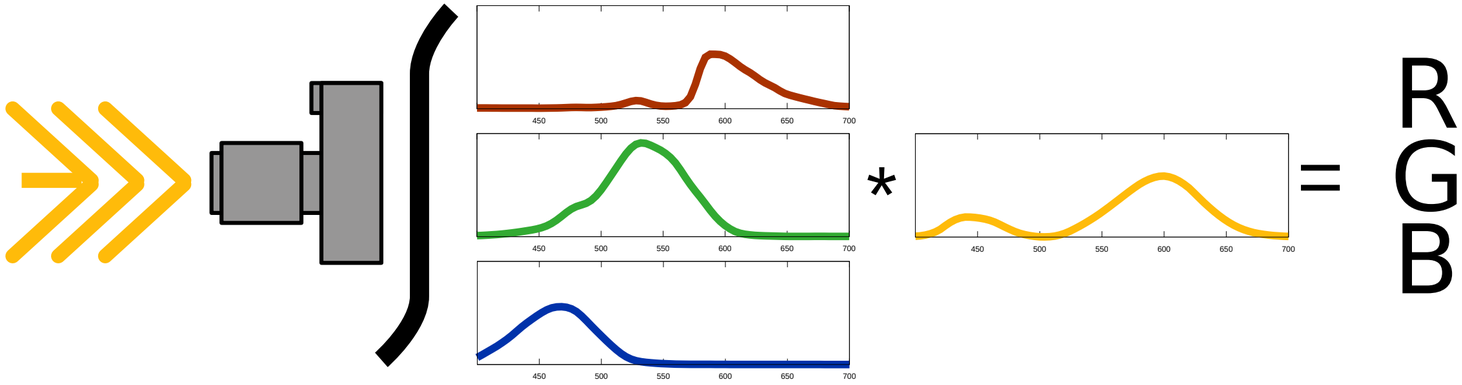




What does a camera see?

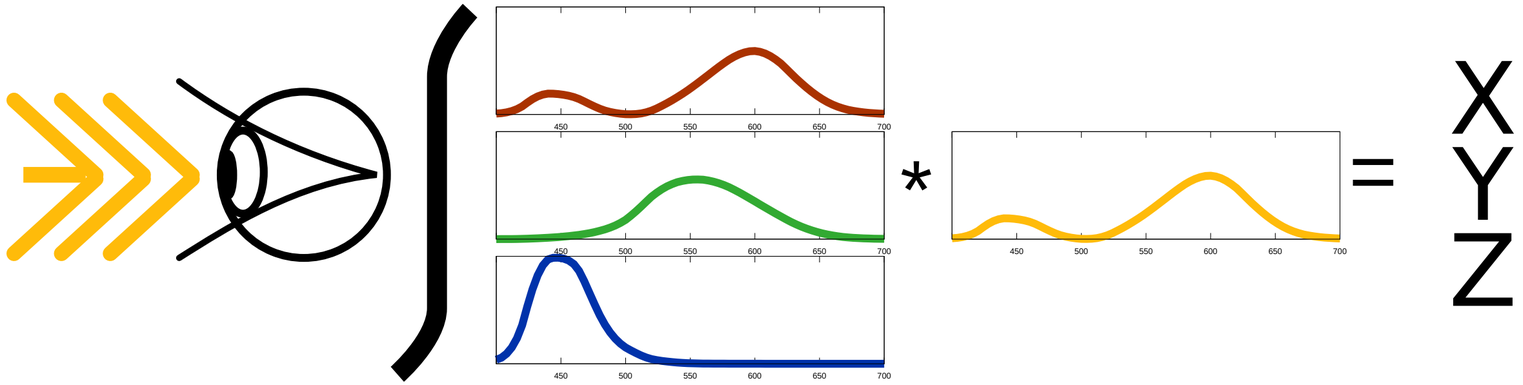
- analogous to the eye: project to colour filter array spectra
- CFA varies a lot with the device

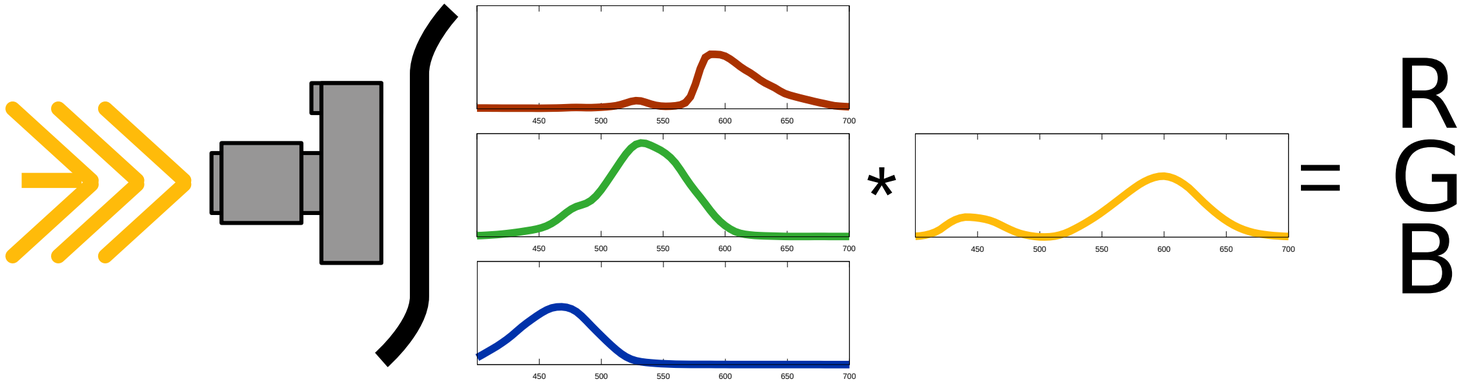




What does a camera see?

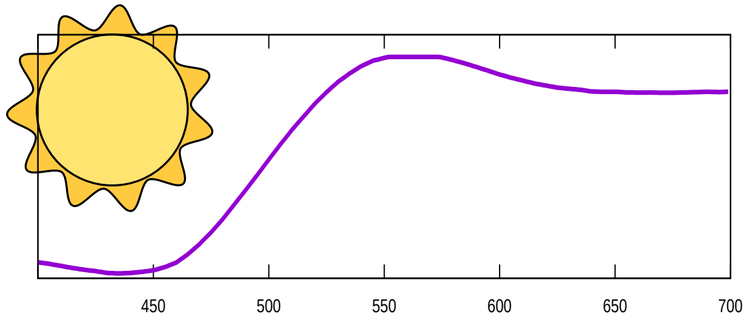
- analogous to the eye: project to colour filter array spectra
- camera RGB to XYZ can in general not be described by a 3×3 matrix!

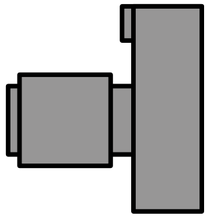




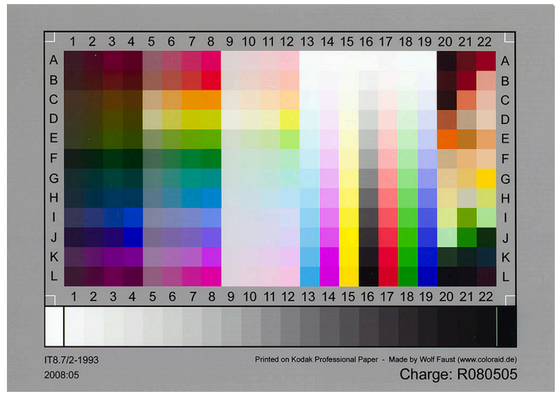
What does a camera see?

- So how do we determine XYZ coordinates from cam RGB then?
  - can be done in various ways (fitting/LUT ...)
  - here, matrix comes from linear least squares fit to calibration chart
  - we use the IT8 target



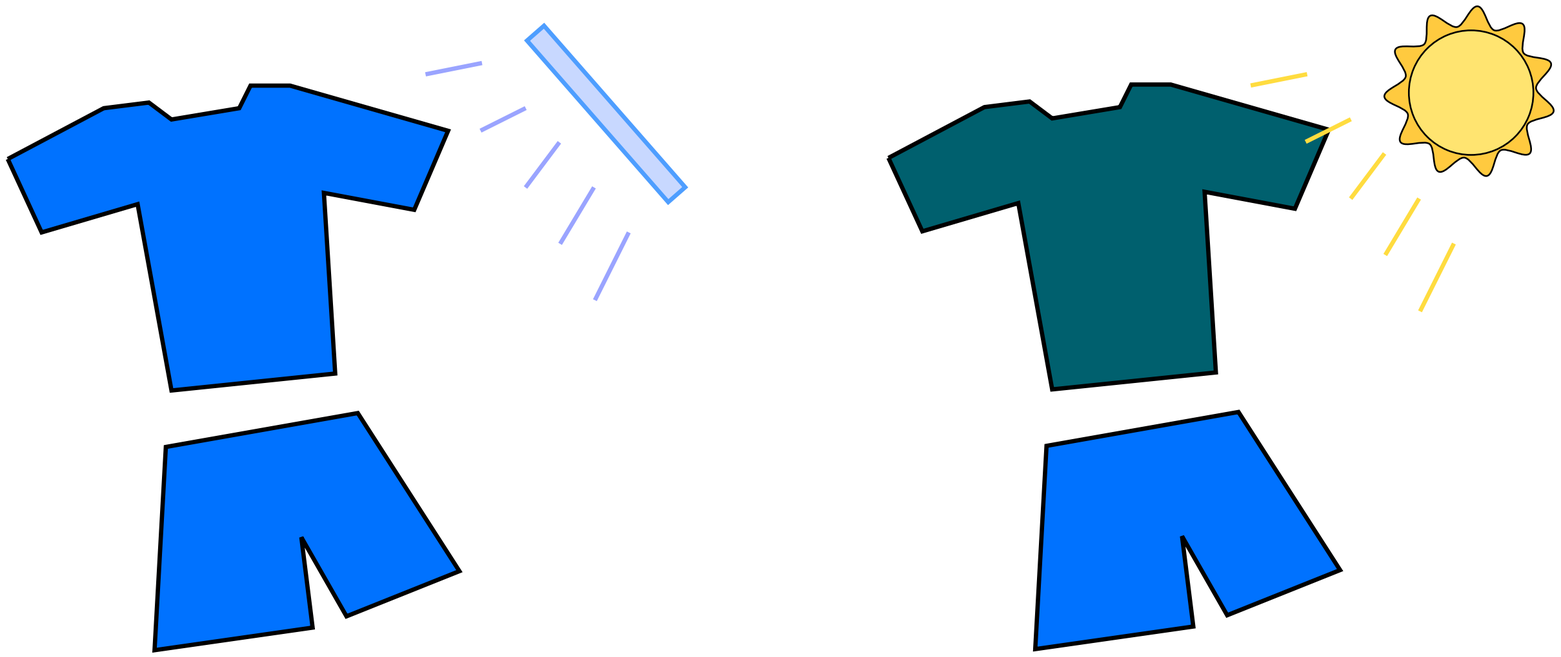


$$\begin{matrix} R \\ G \\ B \end{matrix} * \quad = \begin{matrix} X \\ Y \\ Z \end{matrix}$$



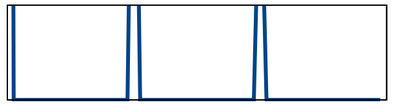
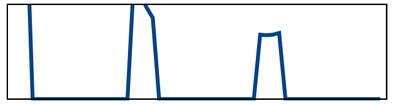
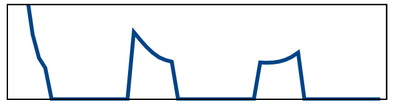
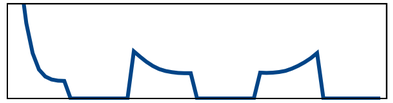
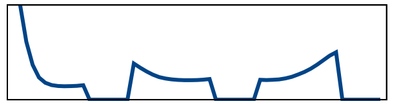
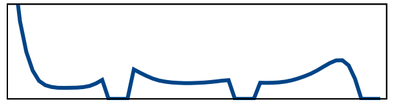
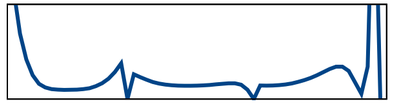
Metamerism

- classic example of illumination metamerism:
  - go into a shop, buy a shirt to match your pants (incandescent lighting)
  - outside in the sun, colours don't match any more!
- observer metamerism: different for different camera responses



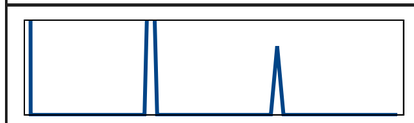
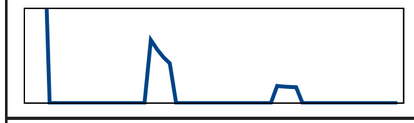
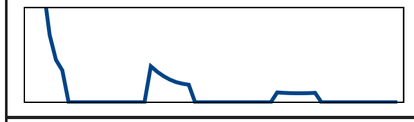
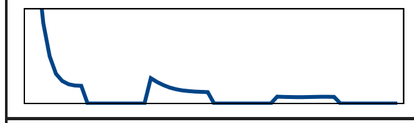
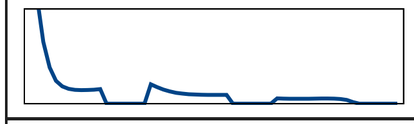
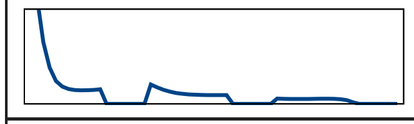
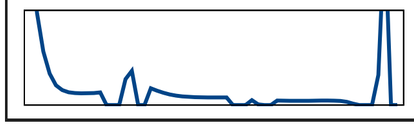
Metameric spectra for the XYZ CMF observer

photographed by different cameras

spectrum	XYZ	c00	c01	c02	c03	c04	c05	c06	c07	c08	c09	c10	c11
													
													
													
													
													
													
													

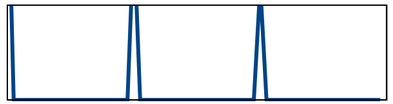
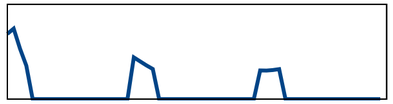
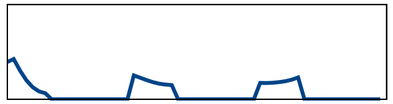
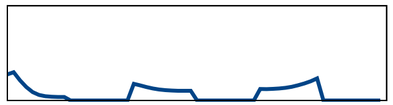
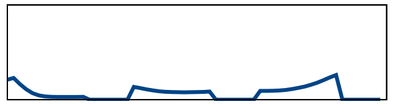
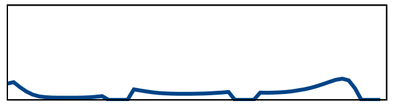
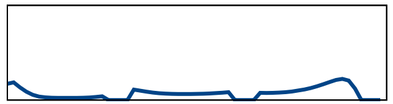
Metameric spectra for the XYZ CMF observer

photographed by different cameras

spectrum	XYZ	c00	c01	c02	c03	c04	c05	c06	c07	c08	c09	c10	c11
													
													
													
													
													
													
													

Metameric spectra for the XYZ CMF observer

photographed by different cameras

spectrum	XYZ	c00	c01	c02	c03	c04	c05	c06	c07	c08	c09	c10	c11
													
													
													
													
													
													
													

Colour reproduction using a renderer

- integrate over the pixel

$$I_p(\lambda) = \int_{\mathcal{P}} h_p(\mathbf{X}) f(\mathbf{X}) d\mathbf{X}$$

- and project into camera RGB by multiplying the colour matching fuction $\bar{r}(\lambda)$

$$I_r = \int_{\Lambda} \bar{r}(\lambda) I_p(\lambda) d\lambda$$

- for a simple single-bounce path:

$$I_r = \int_{\Lambda} \bar{r}(\lambda) \int_{\mathcal{P}} h_p(\mathbf{X}) \cdot \underbrace{L_e(\lambda) \cdot G \cdot f_r(\lambda) \cdot G \cdot W(\lambda)}_{\text{measurement contribution}} \cdot d\mathbf{X} d\lambda$$

Indirect lighting in a renderer

- for a simple single-bounce path

$$I_r = \int_{\mathcal{P}} h_p(\mathbf{X}) \int_{\Lambda} \bar{r}(\lambda) \cdot L_e(\lambda) \cdot G \cdot f_r(\lambda) \cdot G \cdot W(\lambda) \cdot d\lambda d\mathbf{X}$$

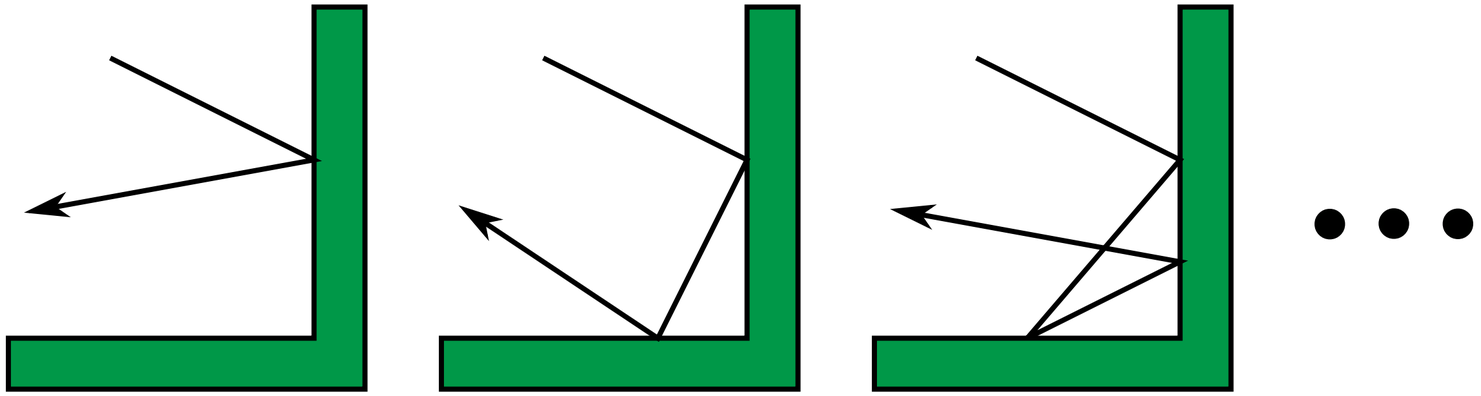
- we can swap the integrals
  - first convert to RGB and then accumulate in an RGB frame buffer
- but RGB rendering:

$$\underbrace{\int_{\Lambda} \bar{r}(\lambda) \cdot L_e(\lambda) \cdot f_r(\lambda) d\lambda}_{\text{reflected light in RGB}} \neq \underbrace{\int_{\Lambda} \bar{r}(\lambda) \cdot L_e(\lambda) d\lambda}_{\text{light in RGB}} \cdot \underbrace{\int_{\Lambda} \bar{r}(\lambda) \cdot f_r(\lambda) d\lambda}_{\text{Bsdf in RGB}}$$

- **Ouch!**

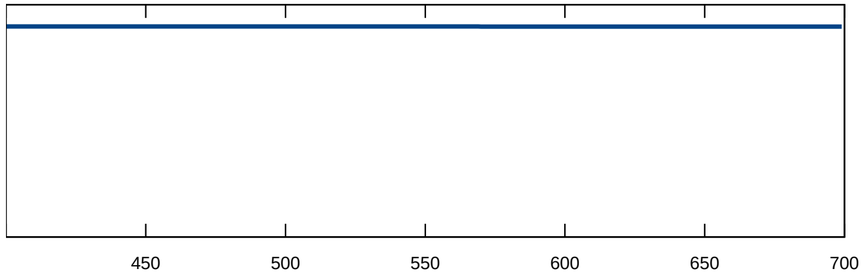
Computing indirect light in RGB/tristimulus

- completely wrong, but how bad is it?
- simple experiment:
  - simulate diffuse bounces
  - multiply albedo by itself
  - observe colour change with number of bounces
  - depends on working space (Adobe RGB, rec709, ACES, ..)
- spectral as reference

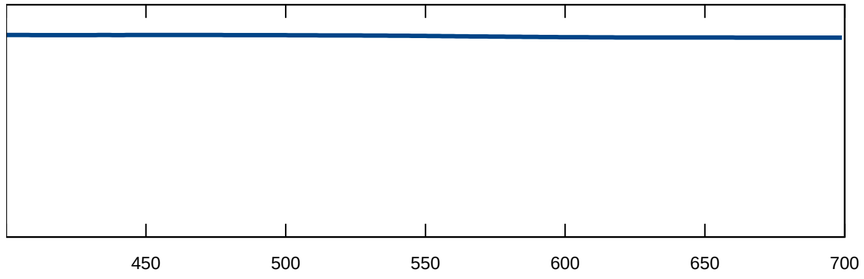


Indirect lighting, reflectance in rec709 (1.00, 0.95, 0.90)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
Adobe RGB															
rec709															
spectral															

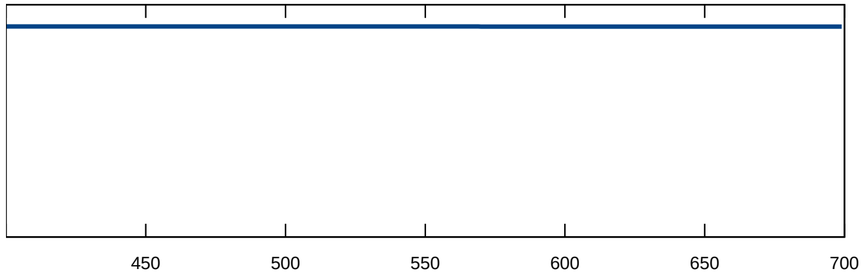


illuminant E white (whitepoint does not match rec709)

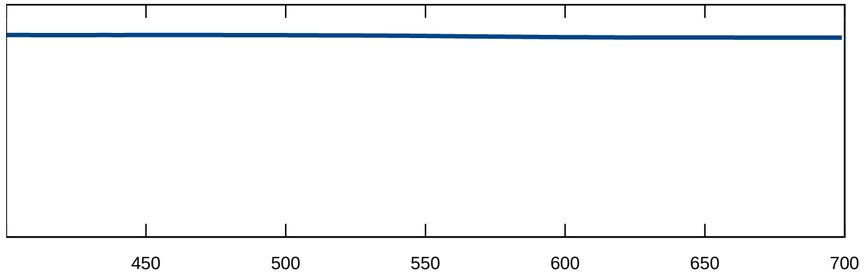


Indirect lighting, reflectance in rec709 (1.00, 0.95, 0.90)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACEScg															
Adobe RGB															
rec709															
spectral															

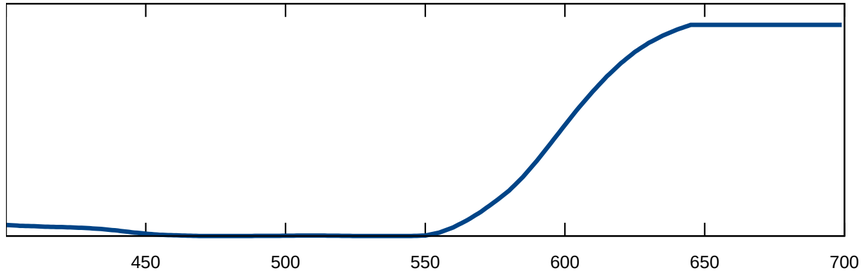


illuminant E white (whitepoint does not match rec709)

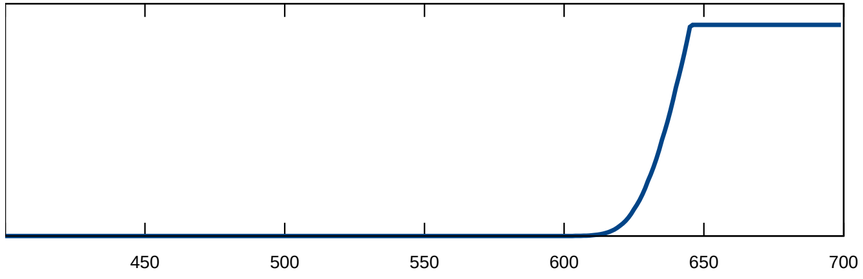


Indirect lighting, reflectance in rec709 (0.97, 0.01, 0.00)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACEScg															
Adobe RGB															
rec709															
spectral															

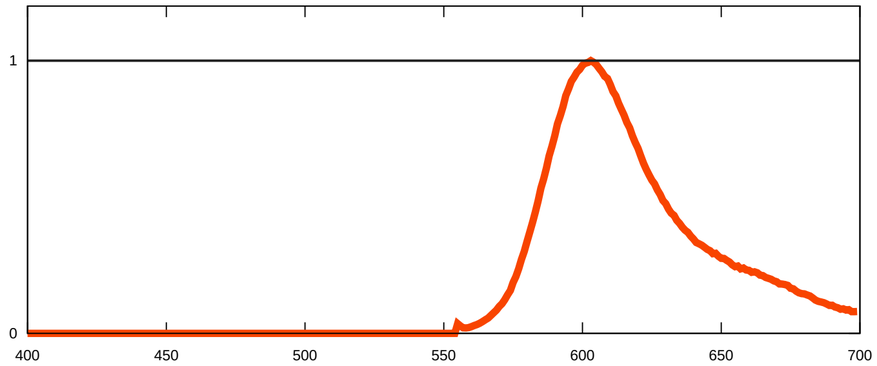


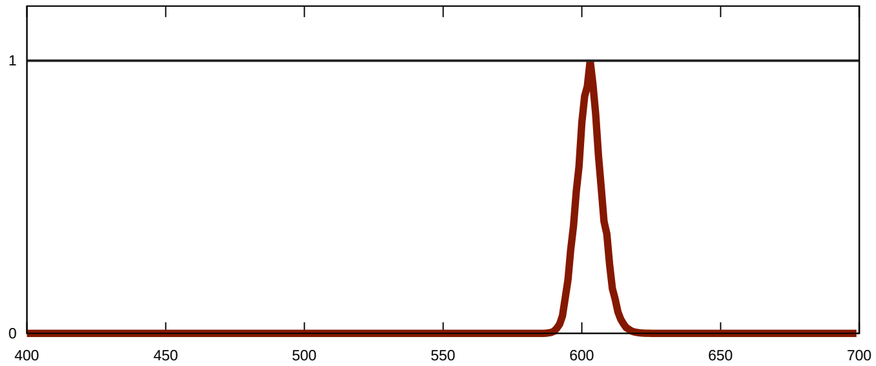
most saturated red (gamut mapped to valid reflectance)



Valid reflectances

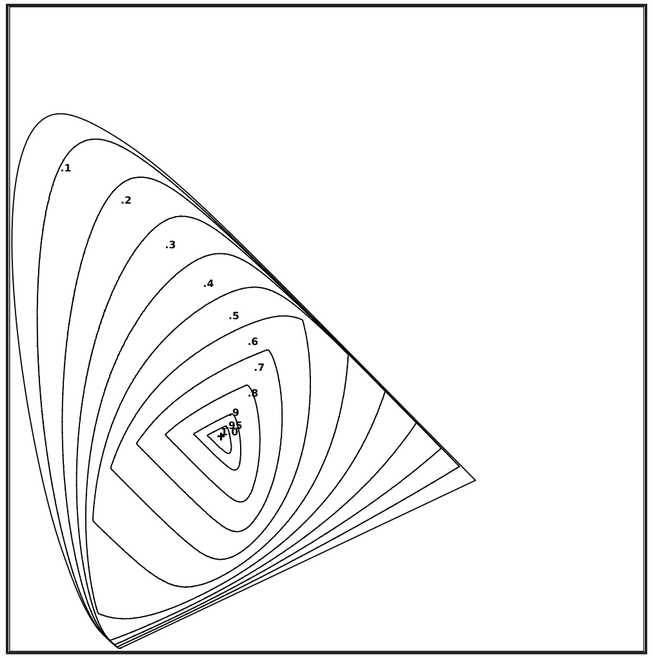
- limit on reflectances is energy conservation
- may not exceed 1.0 for any wavelength
- trade off between brightness and saturation

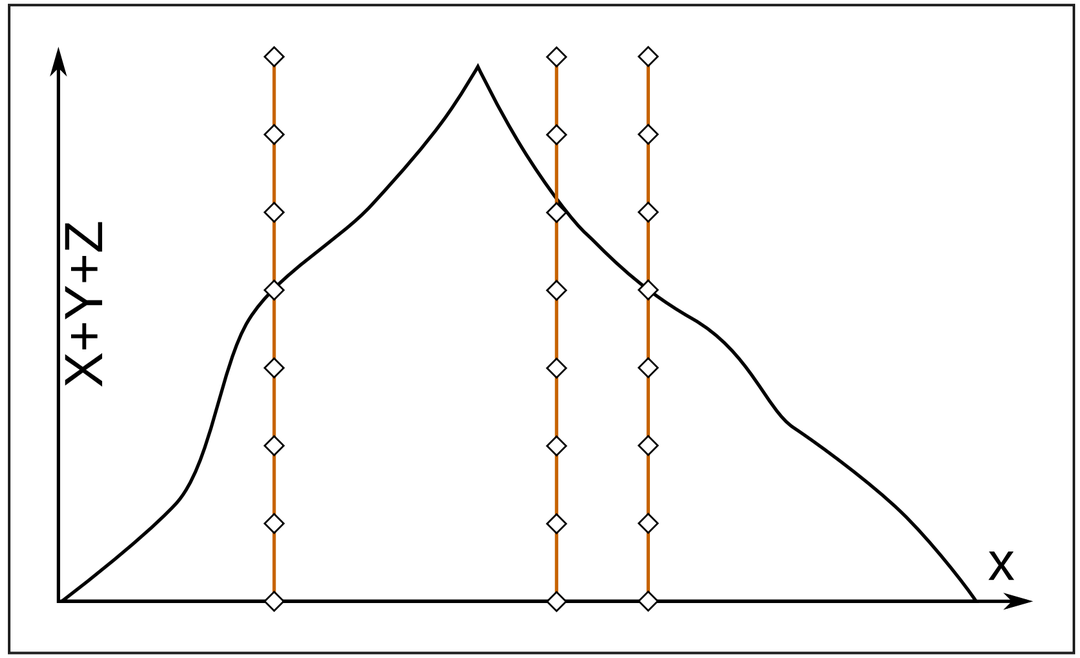




Gamut mapping

- need some means to find the closest valid reflectance

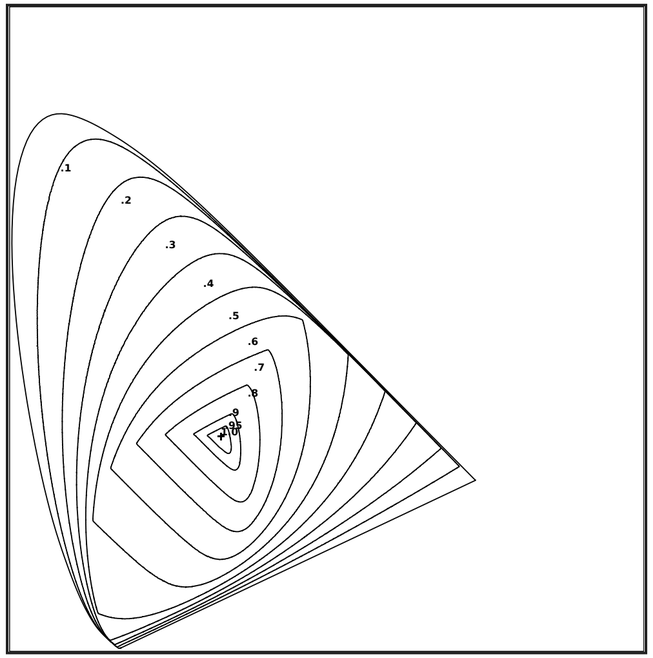


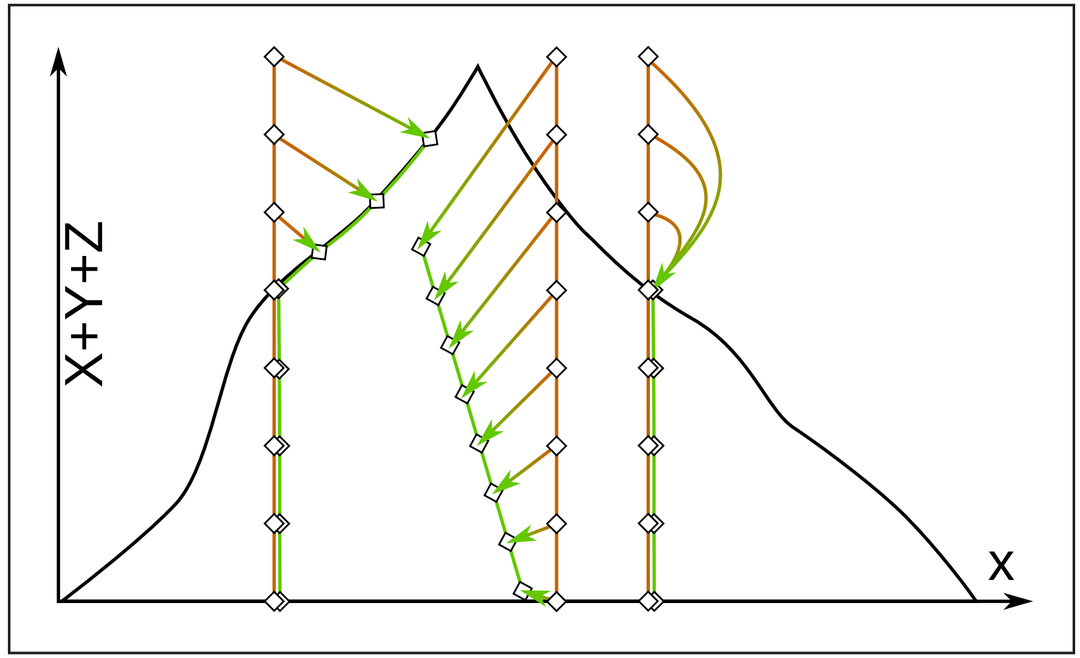


- for instance
  - smallest perceptual error ΔE
  - linear transform
  - simple clamping

Gamut mapping

- need some means to find the closest valid reflectance

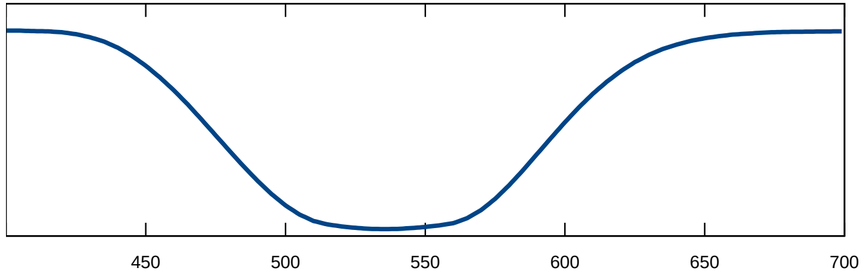




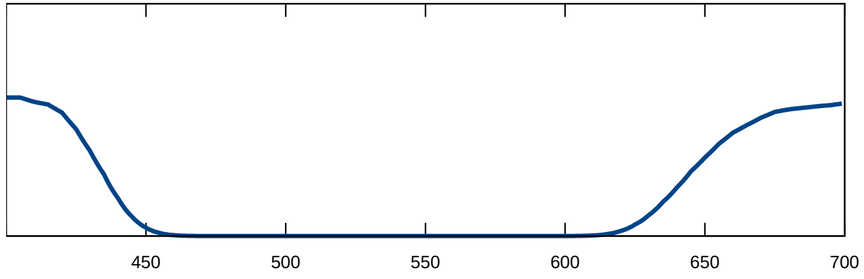
- for instance
  - smallest perceptual error ΔE
  - linear transform
  - simple clamping

Indirect lighting, reflectance in rec709 (0.96, 0.00, 0.72)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACEScg															
Adobe RGB															
rec709															
spectral															



magenta
(not saturated)

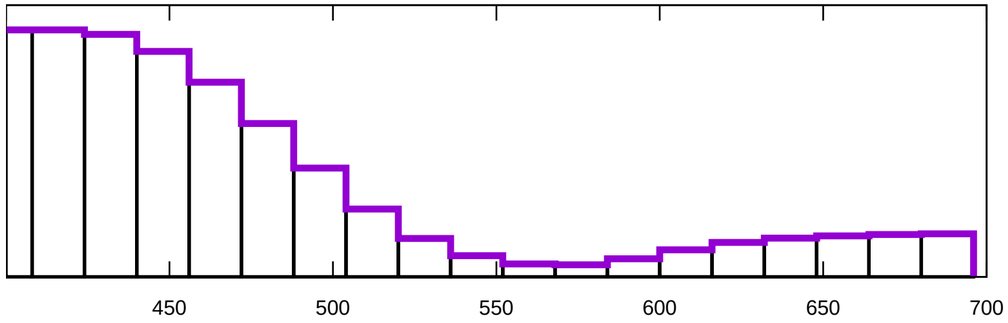


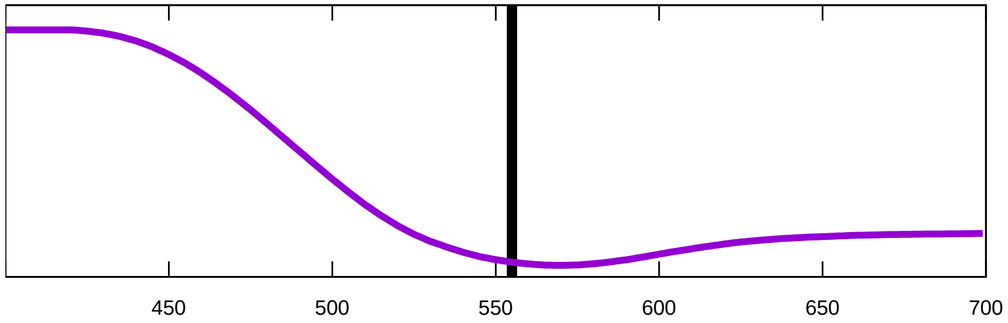
RGB transport?

- why would we do it? it's wrong!
- let's go spectral! does it cost us anything?

Spectral rendering

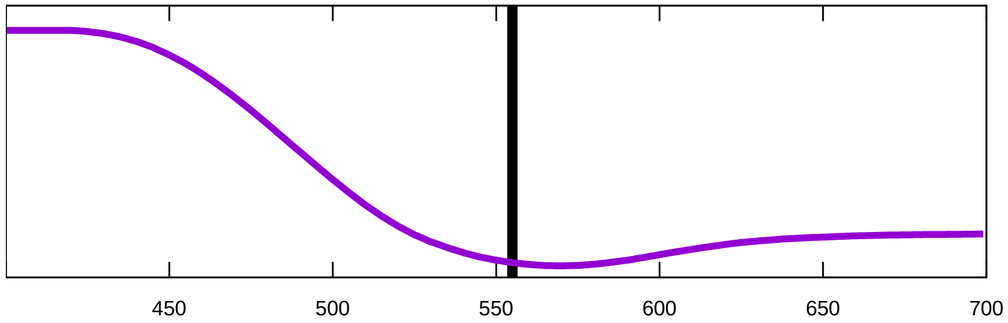
- many different approaches in literature
- for simple methods, basically two choices:
  - evaluate path for fixed set of wavelengths (60 bins at 5nm spacing)
  - Monte Carlo (evaluate for one single wavelength)

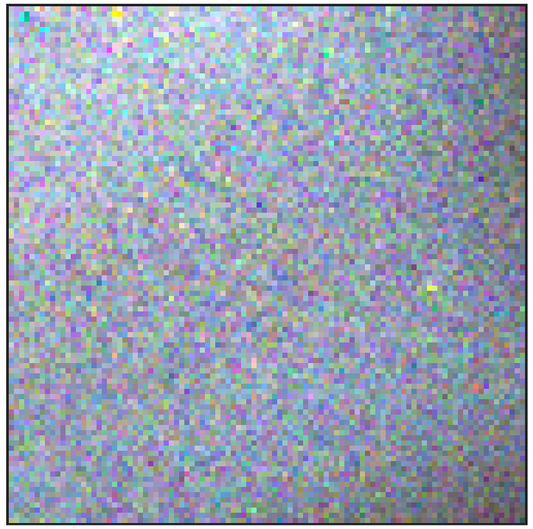




Spectral rendering: Monte Carlo

- compute everything for one single random wavelength

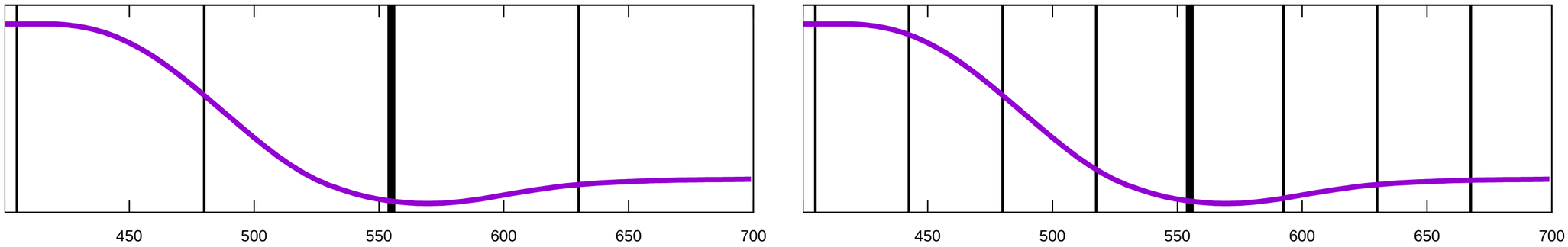




- Monte Carlo means colour noise

Spectral rendering: Monte Carlo

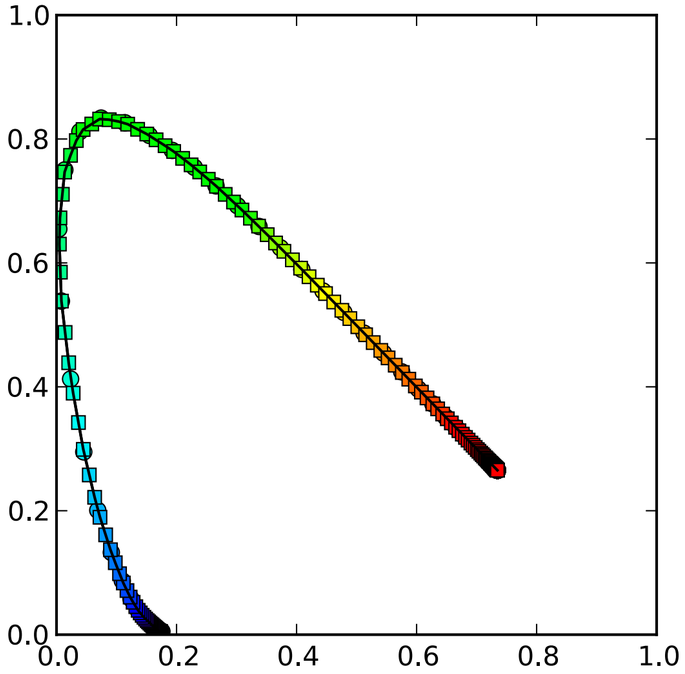
- *hero wavelength* sampling:



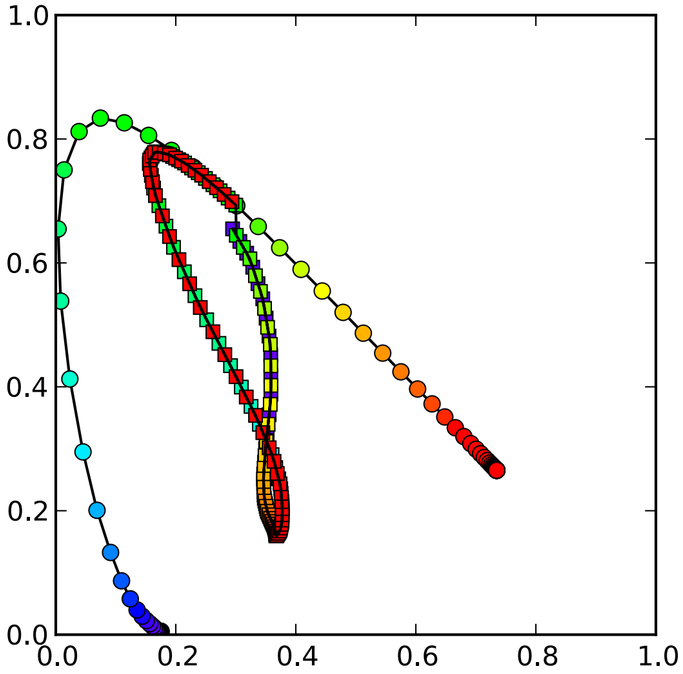
- hero determines importance sampling
- trace a stratified set of wavelengths along with the path
- combine using MIS

Spectral rendering: Hero wavelength sampling

- compute illuminant E white with random hero wavelength
- varying the hero wavelength:
  - what is the 1d locus of all possible colour answers?



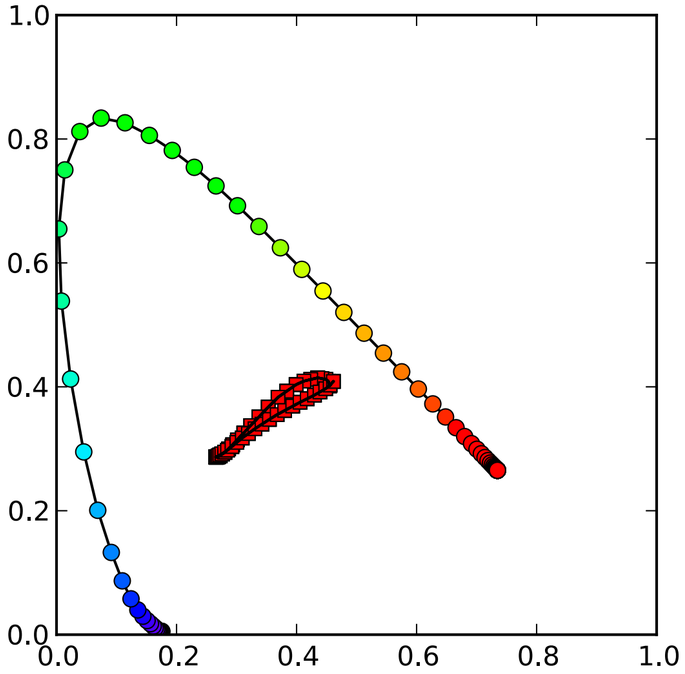
1 wavelength



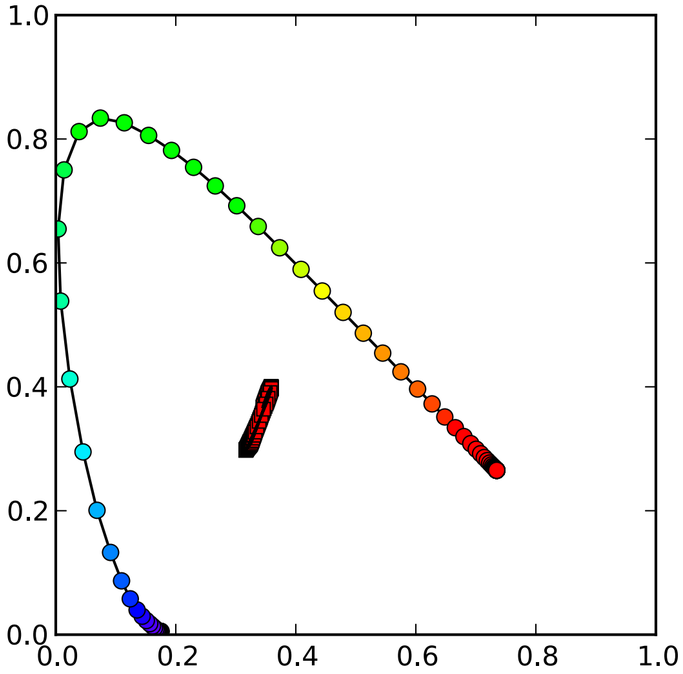
2 wavelengths

Spectral rendering: Hero wavelength sampling

- compute illuminant E white with random hero wavelength
- varying the hero wavelength:
  - what is the 1d locus of all possible colour answers?



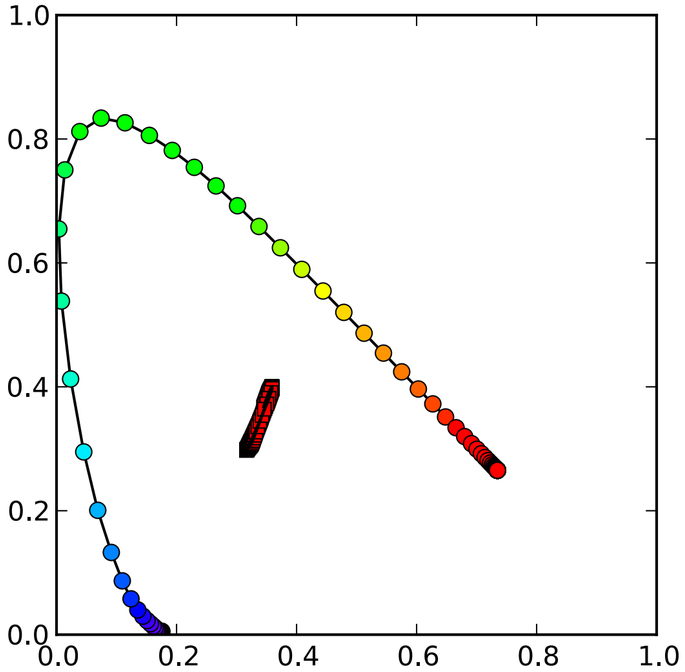
3 wavelengths



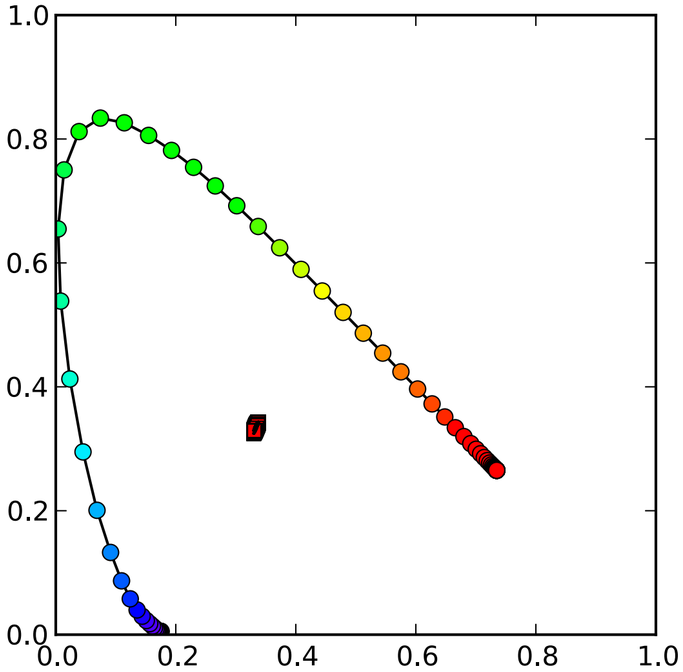
4 wavelengths (SSE)

Spectral rendering: Hero wavelength sampling

- compute illuminant E white with random hero wavelength
- varying the hero wavelength:
  - what is the 1d locus of all possible colour answers?



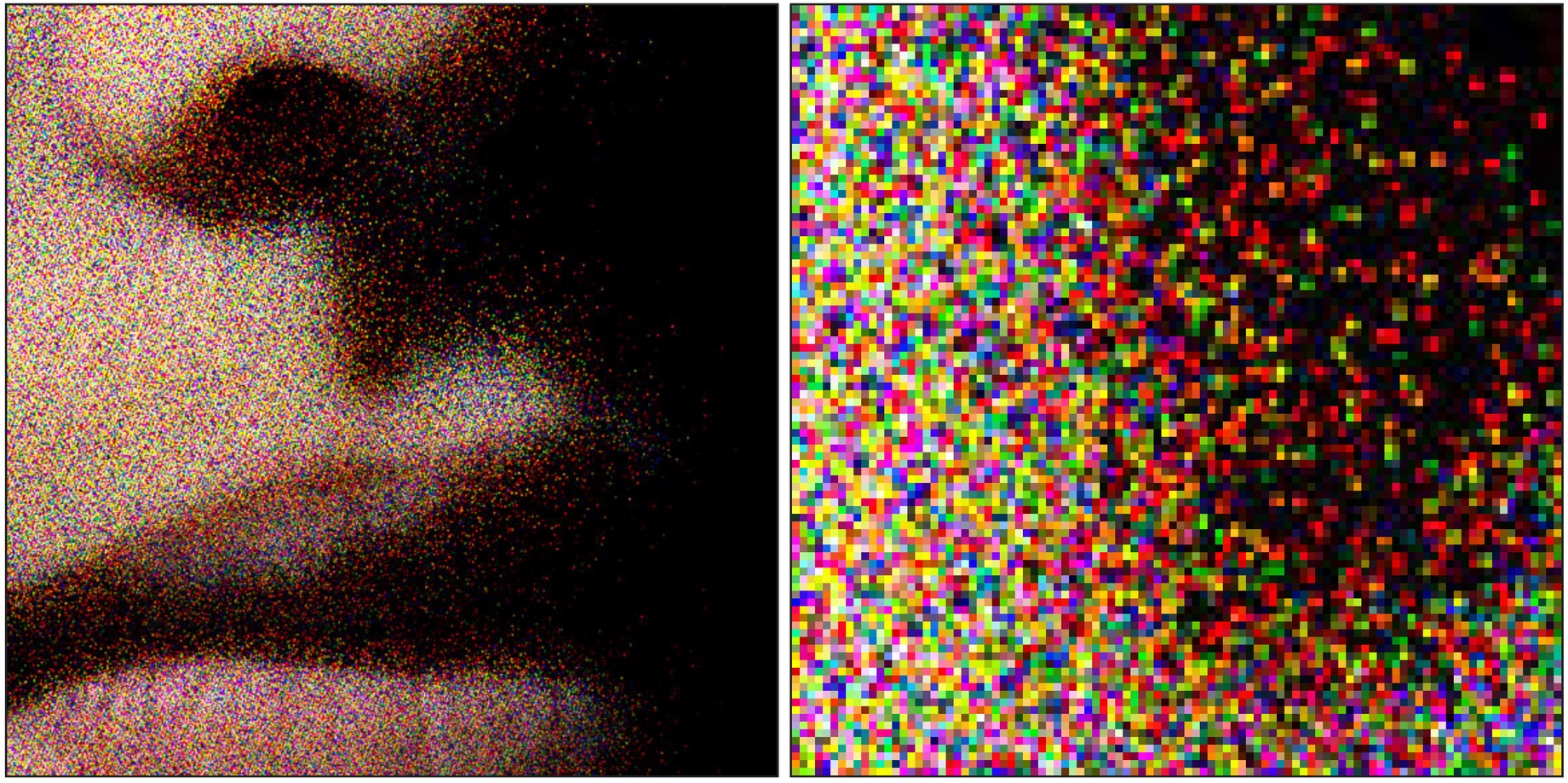
4 wavelengths (SSE)



8 wavelengths (AVX)

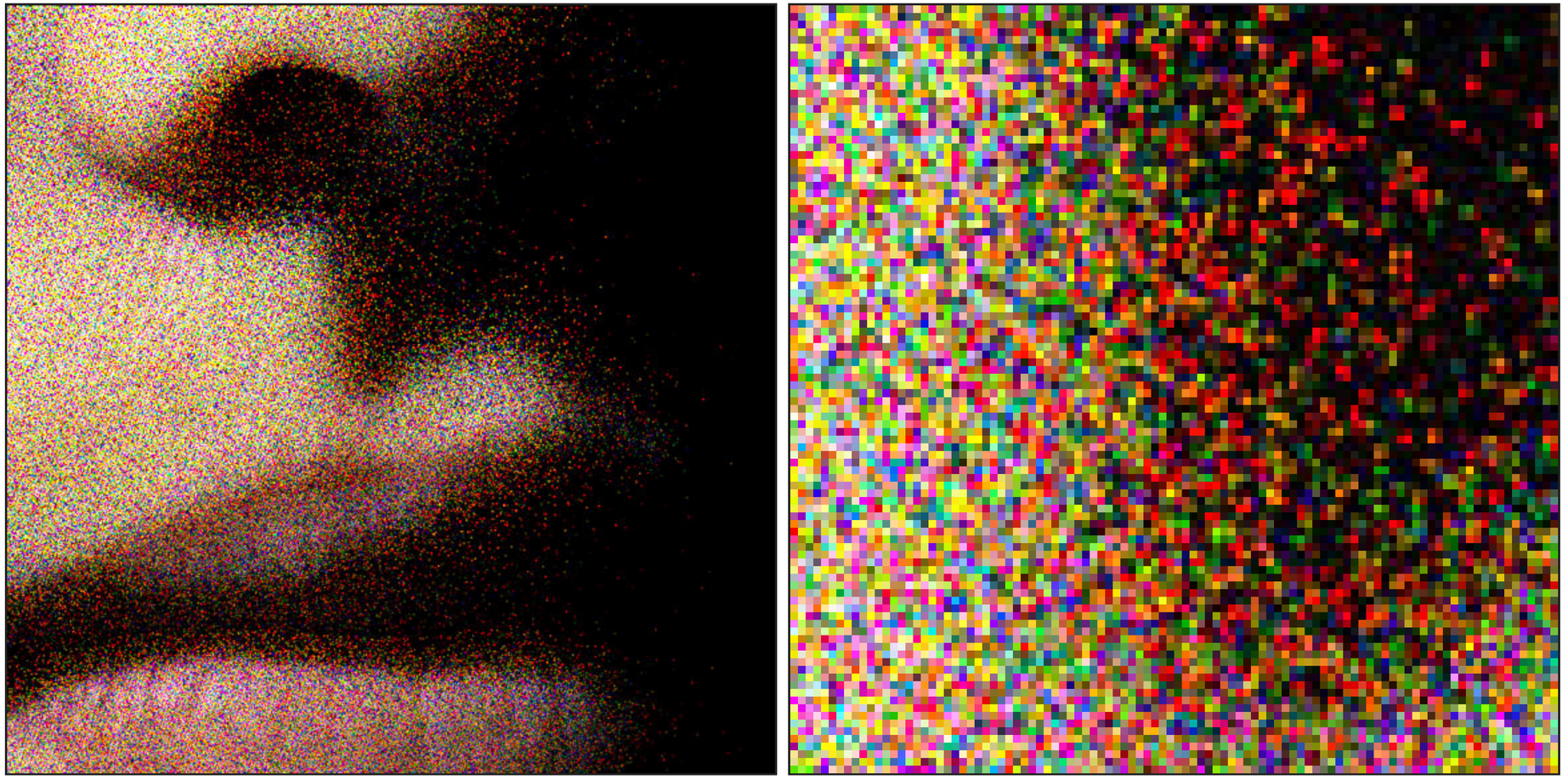
Hero wavelength image comparison 4spp

- skin material with single wavelength



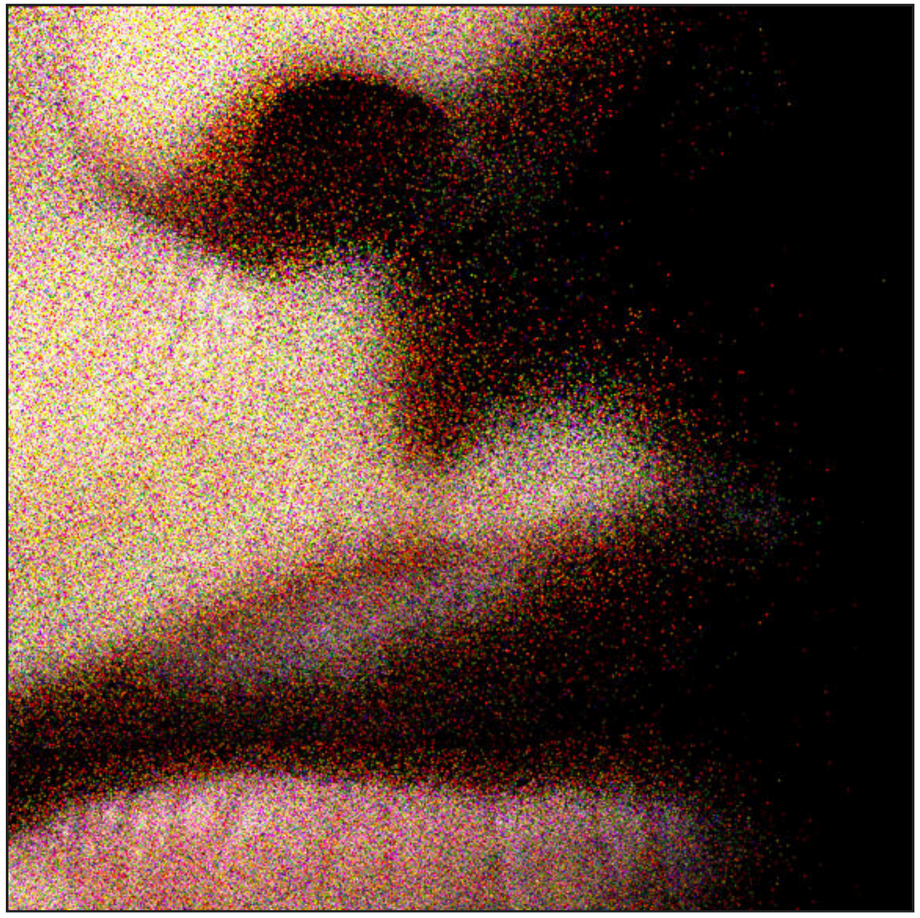
Hero wavelength image comparison 4spp

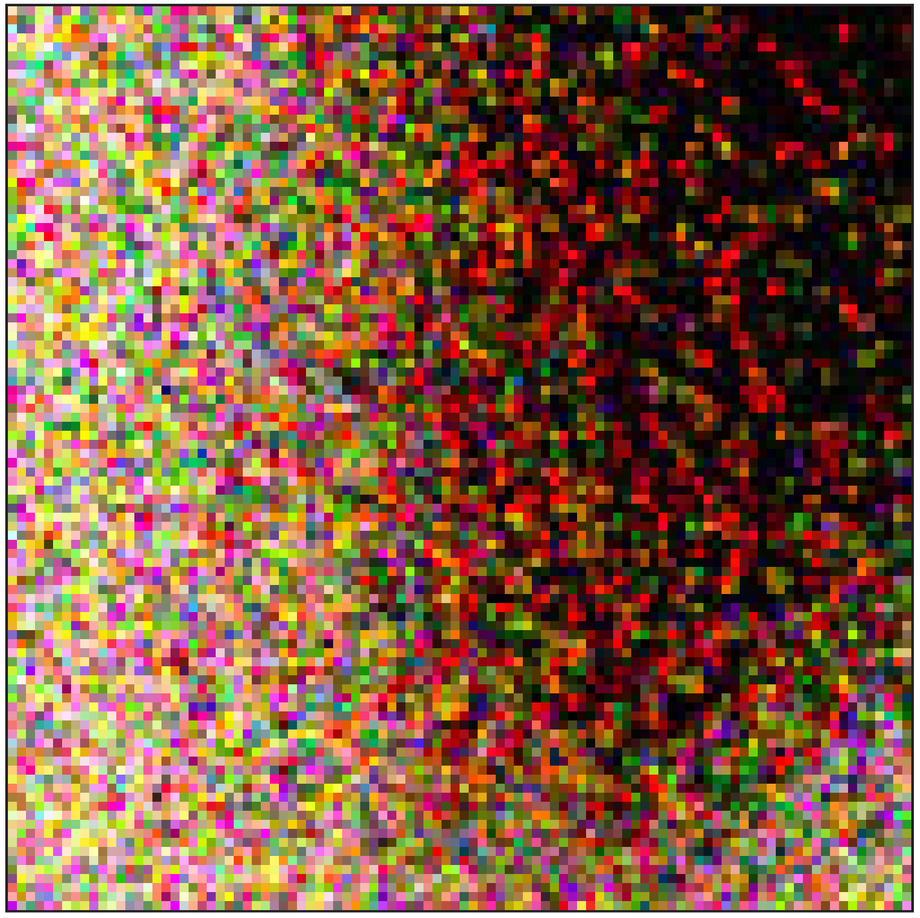
- skin material with 2 wavelengths



Hero wavelength image comparison 4spp

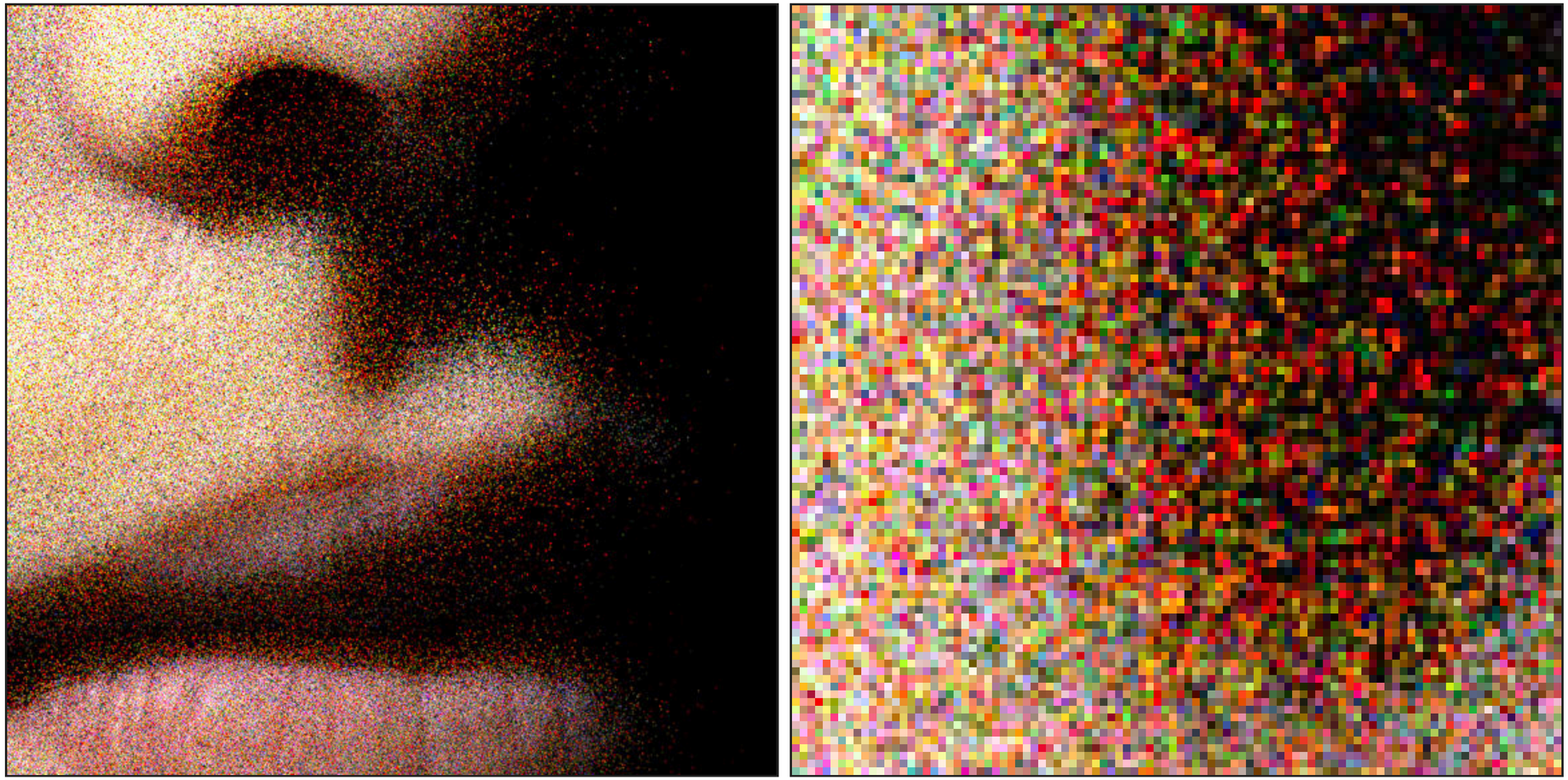
- skin material with 3 wavelengths





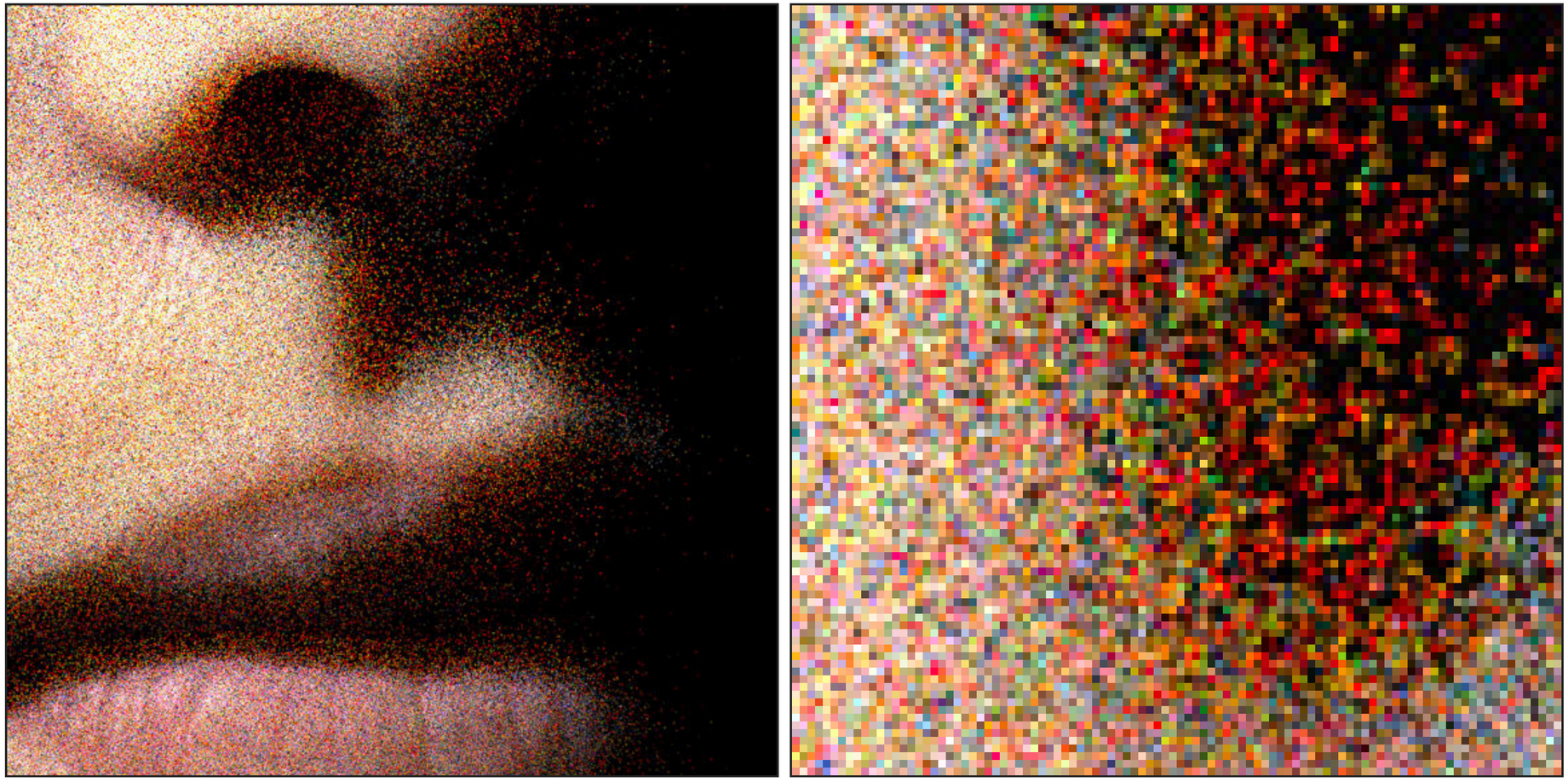
Hero wavelength image comparison 4spp

- skin material with 4 wavelengths



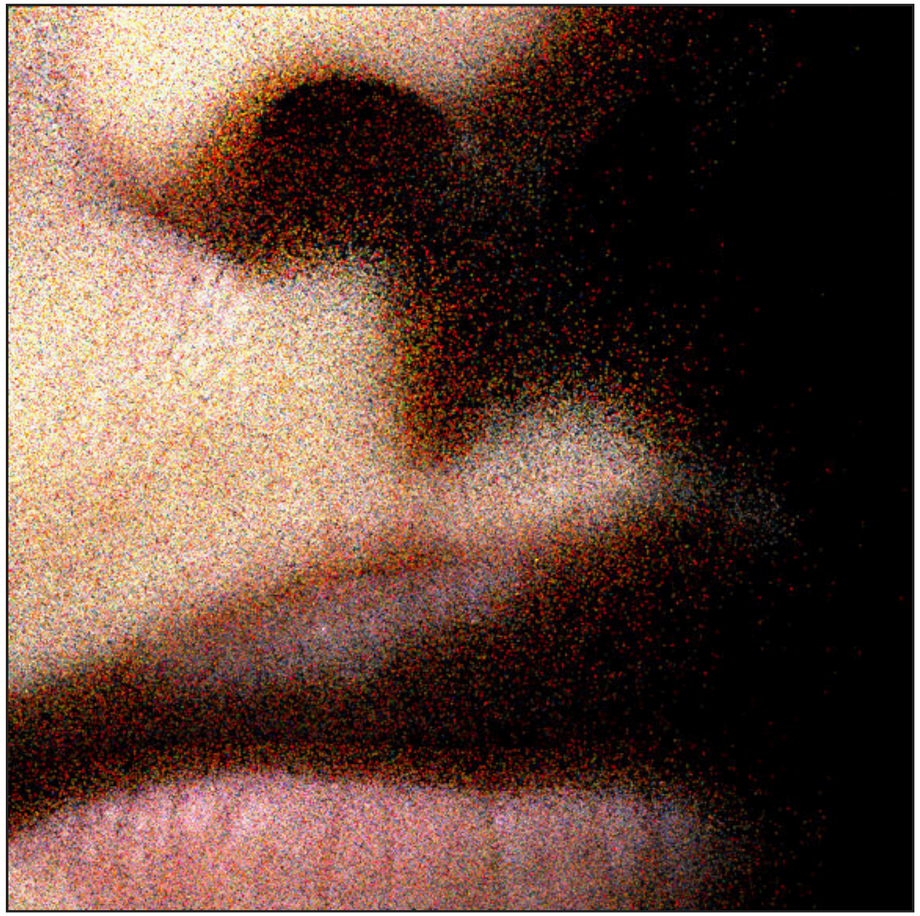
Hero wavelength image comparison 4spp

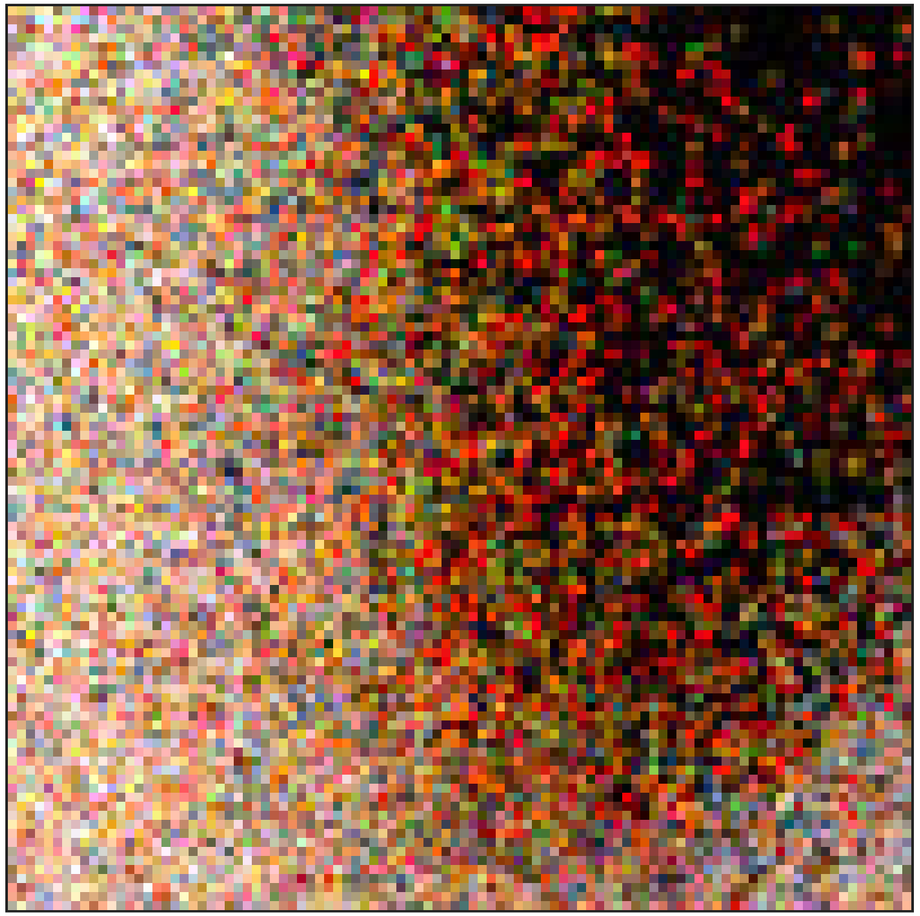
- skin material with 5 wavelengths



Hero wavelength image comparison 4spp

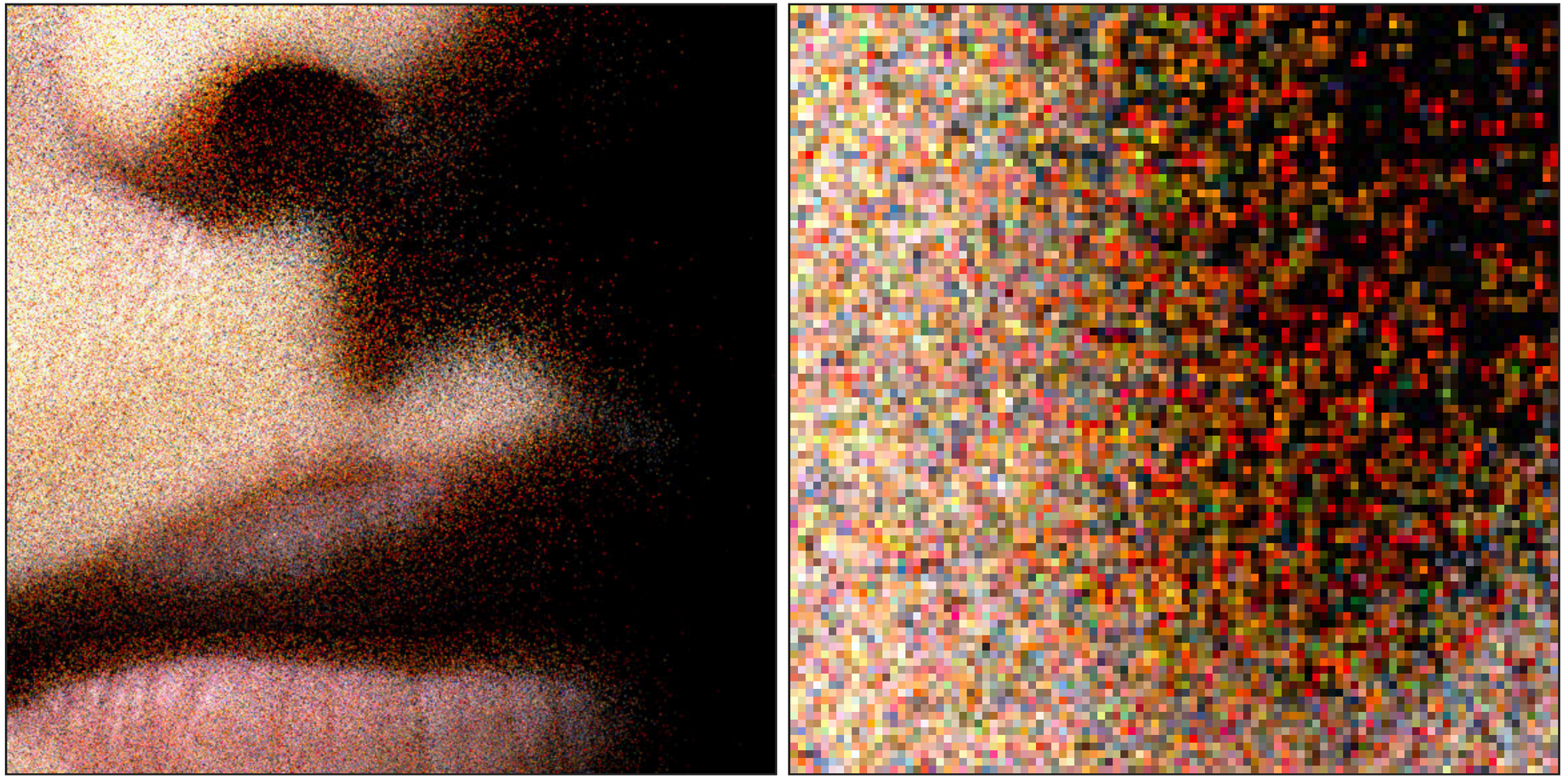
- skin material with 6 wavelengths





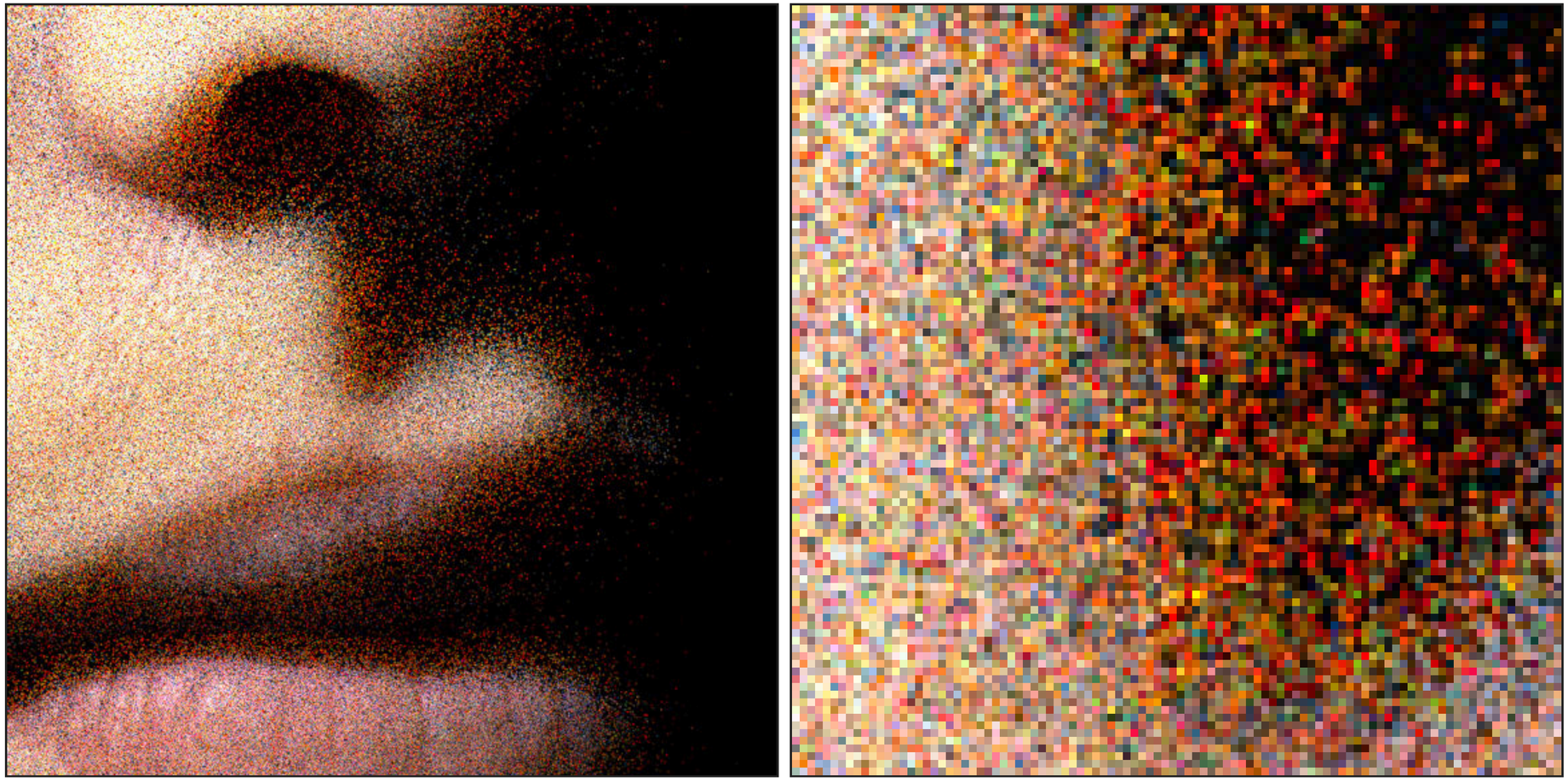
Hero wavelength image comparison 4spp

- skin material with 7 wavelengths



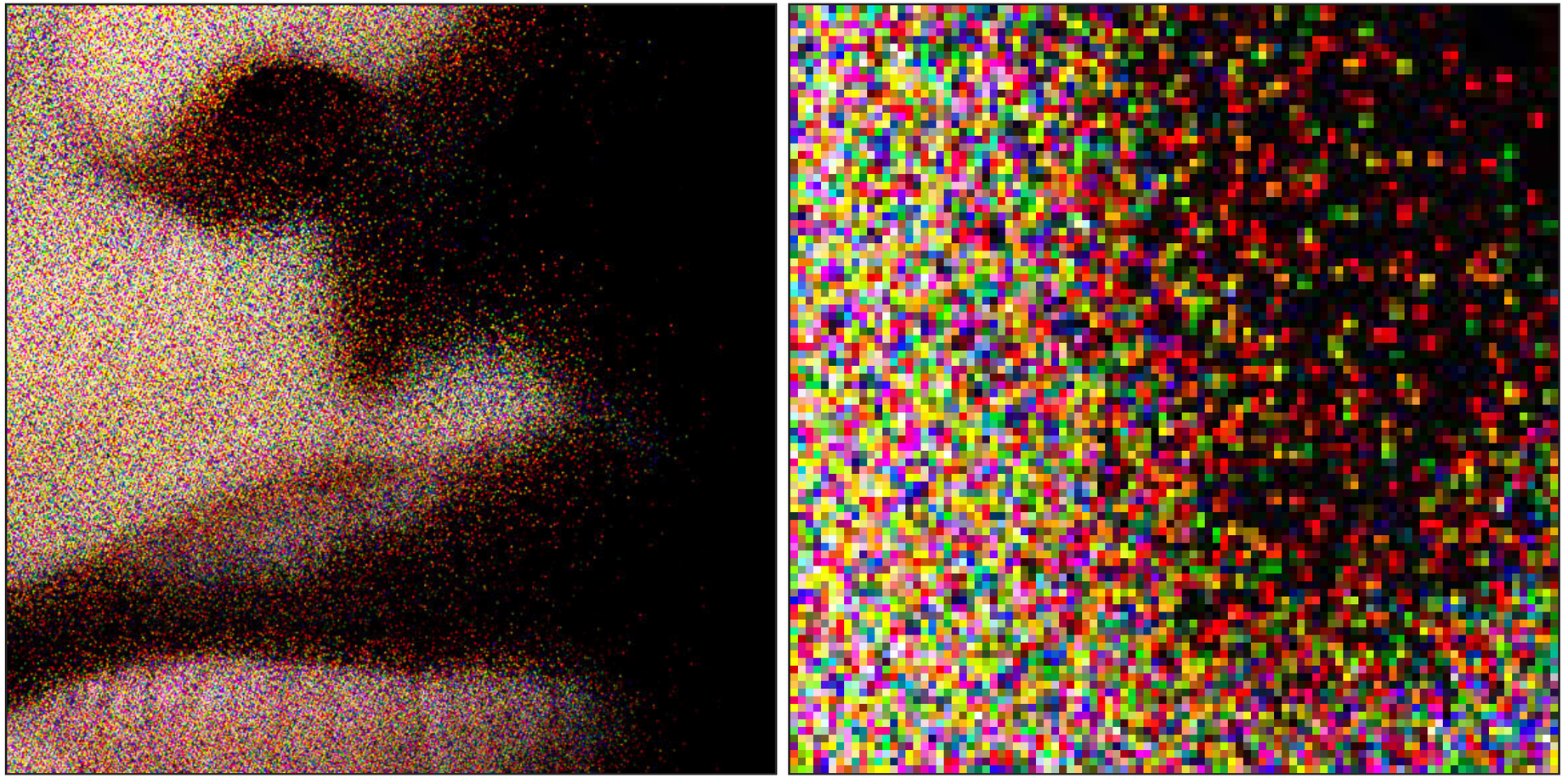
Hero wavelength image comparison 4spp

- skin material with 8 wavelengths



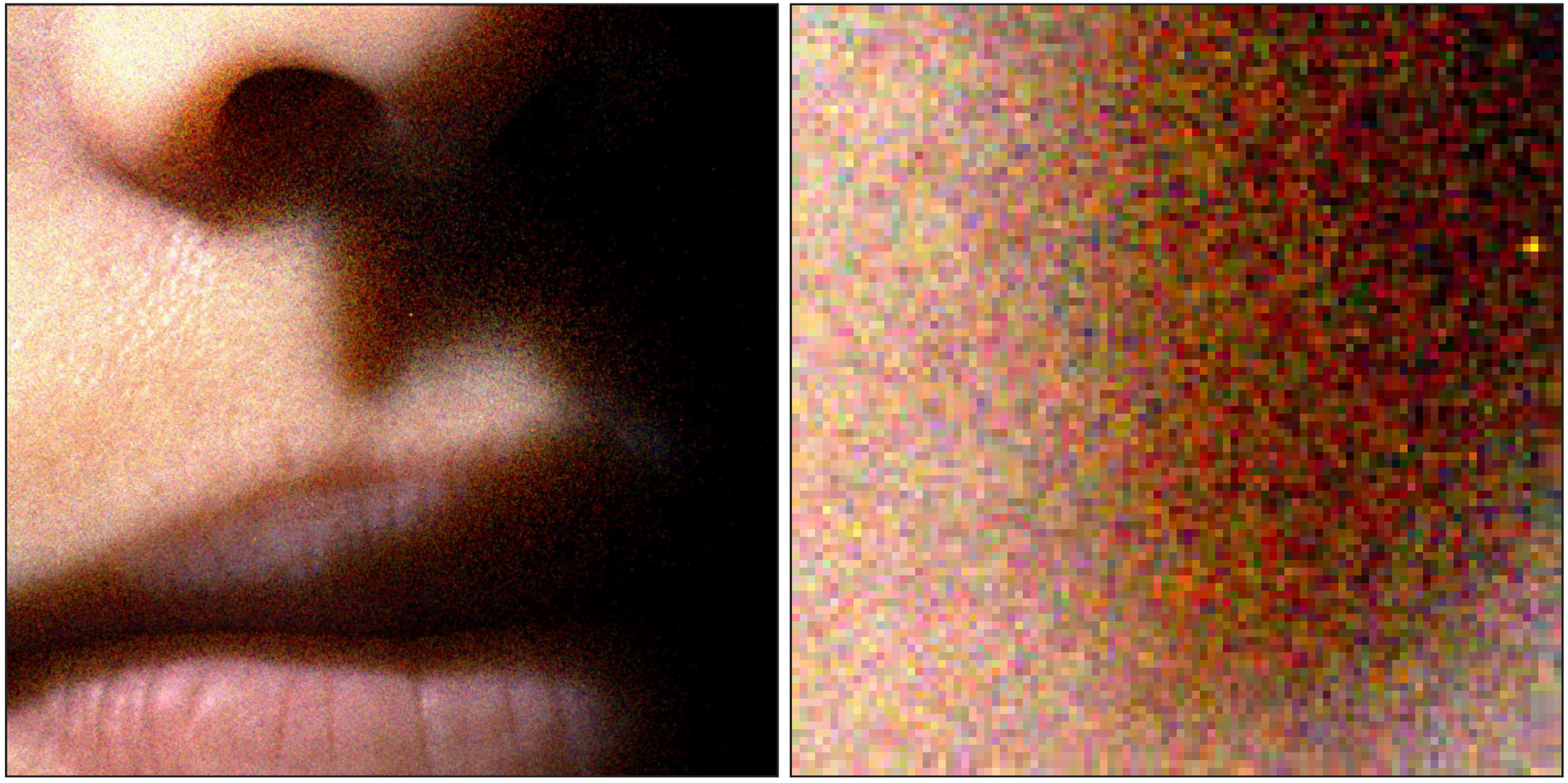
Hero wavelength image comparison 4spp

- readjust your eyes! single wavelength again:



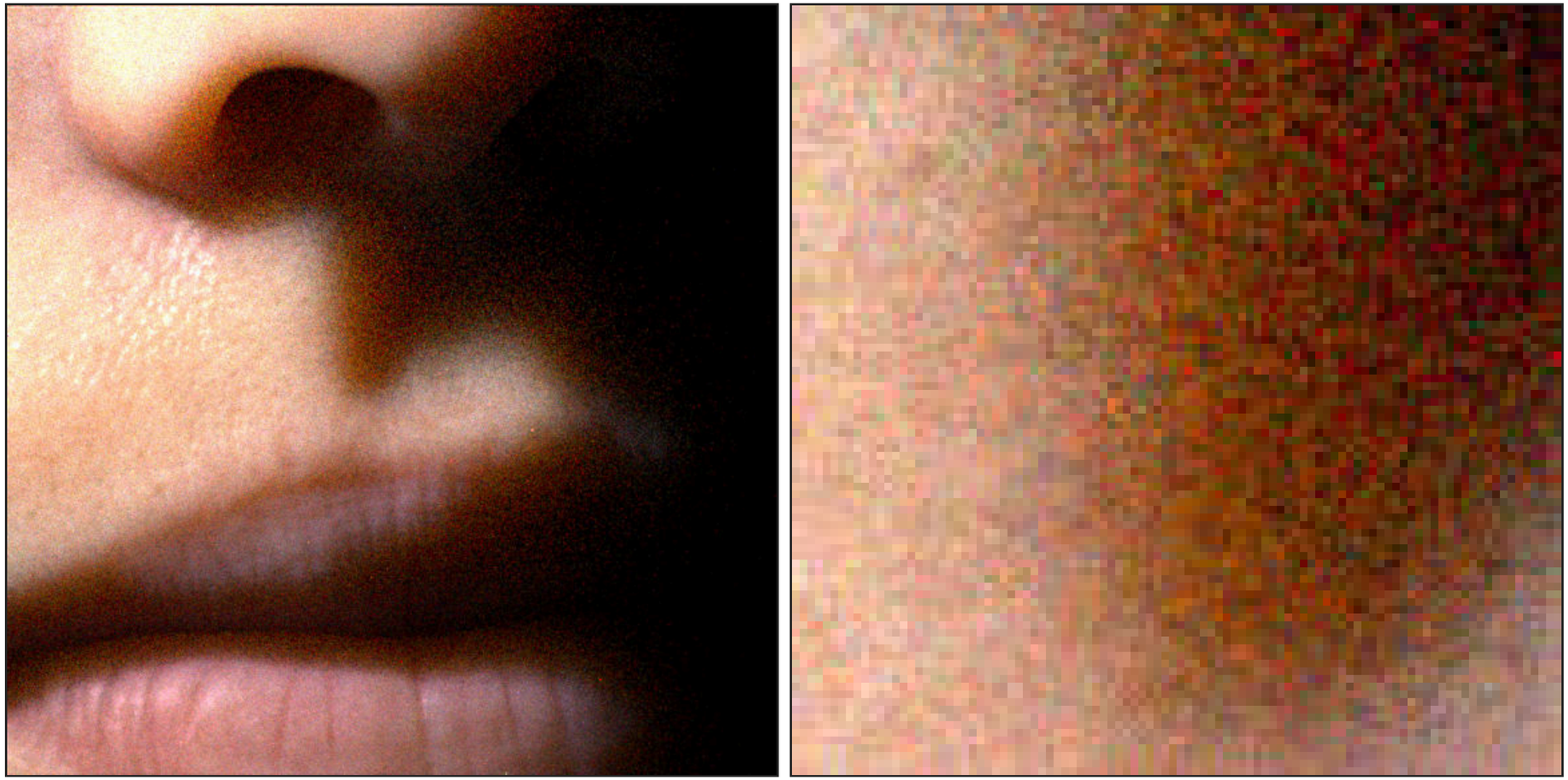
Hero wavelength image comparison 64spp

- skin material with 1 wavelength



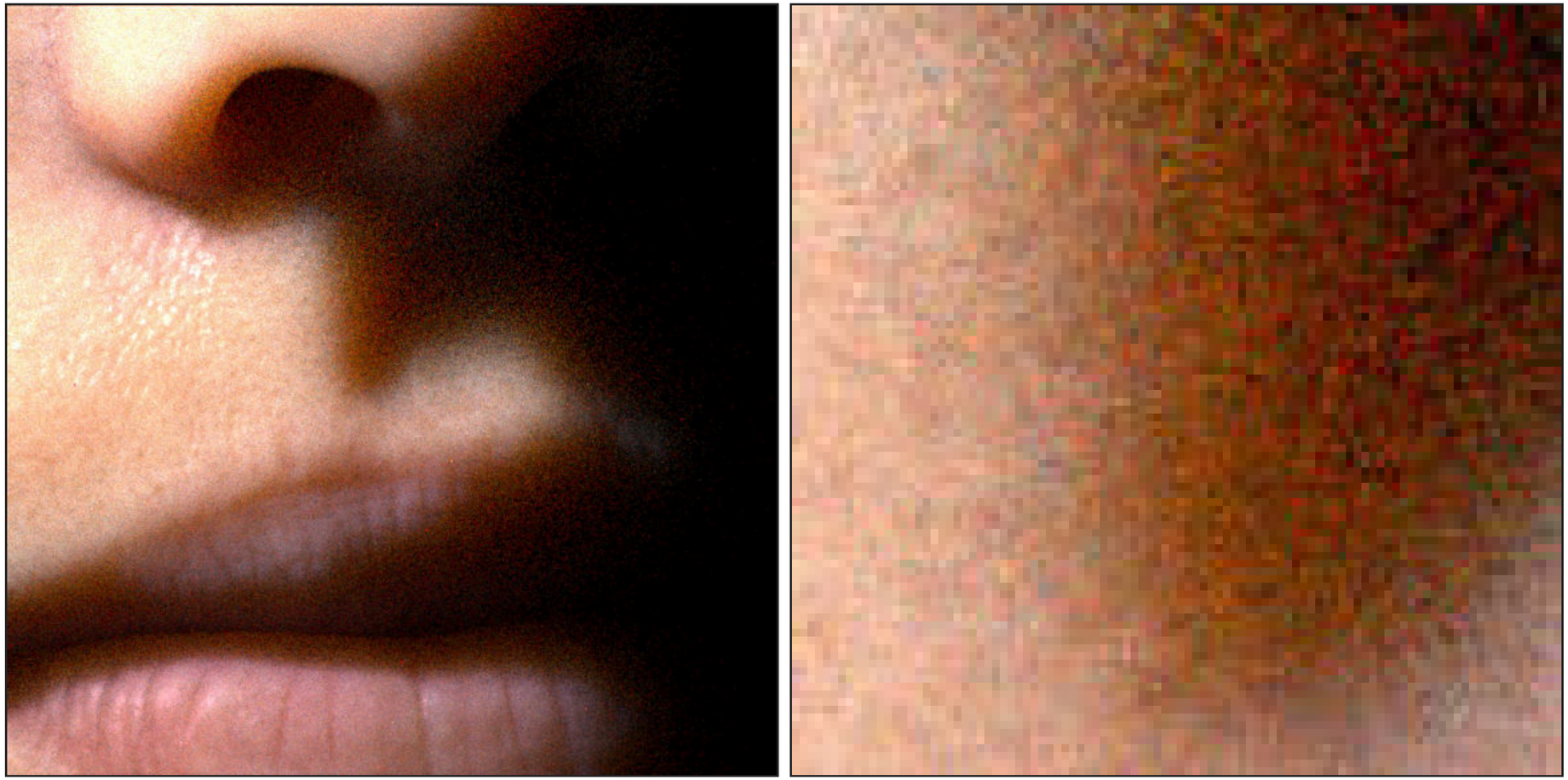
Hero wavelength image comparison 64spp

- skin material with 4 wavelengths (SSE)



Hero wavelength image comparison 64spp

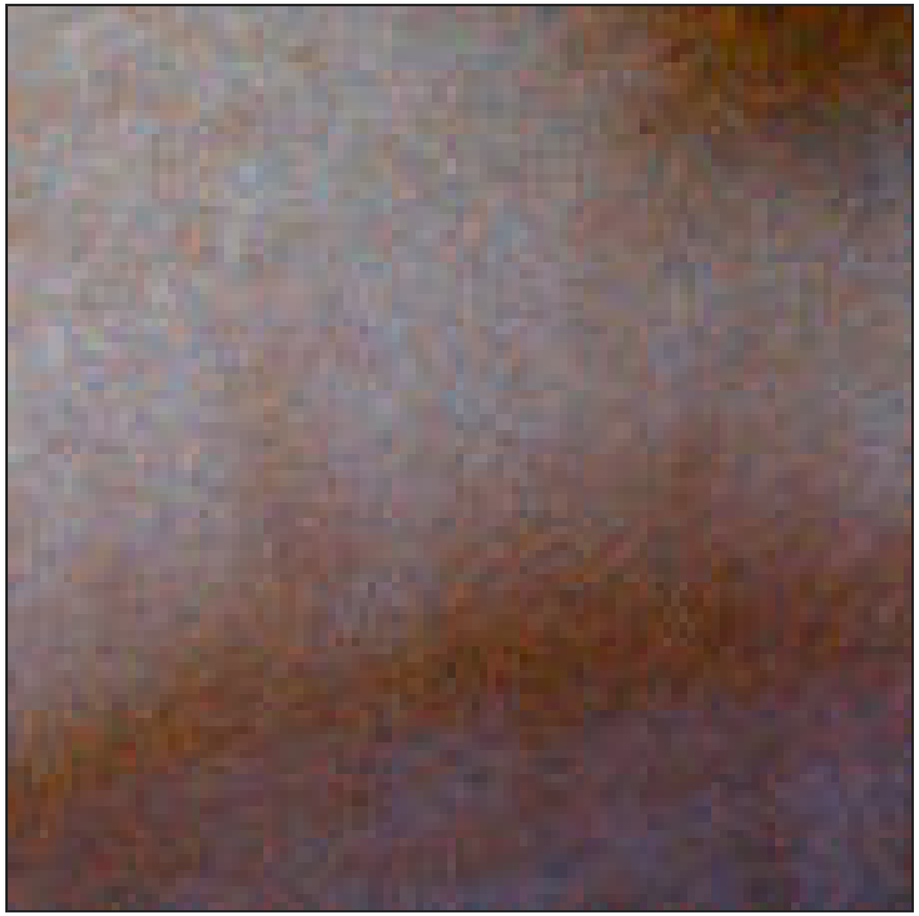
- skin material with 8 wavelengths (AVX)



Hero wavelength vs. denoising in post

- can we apply off-the shelf image denoising to chromaticity channels only?

1 wavelength

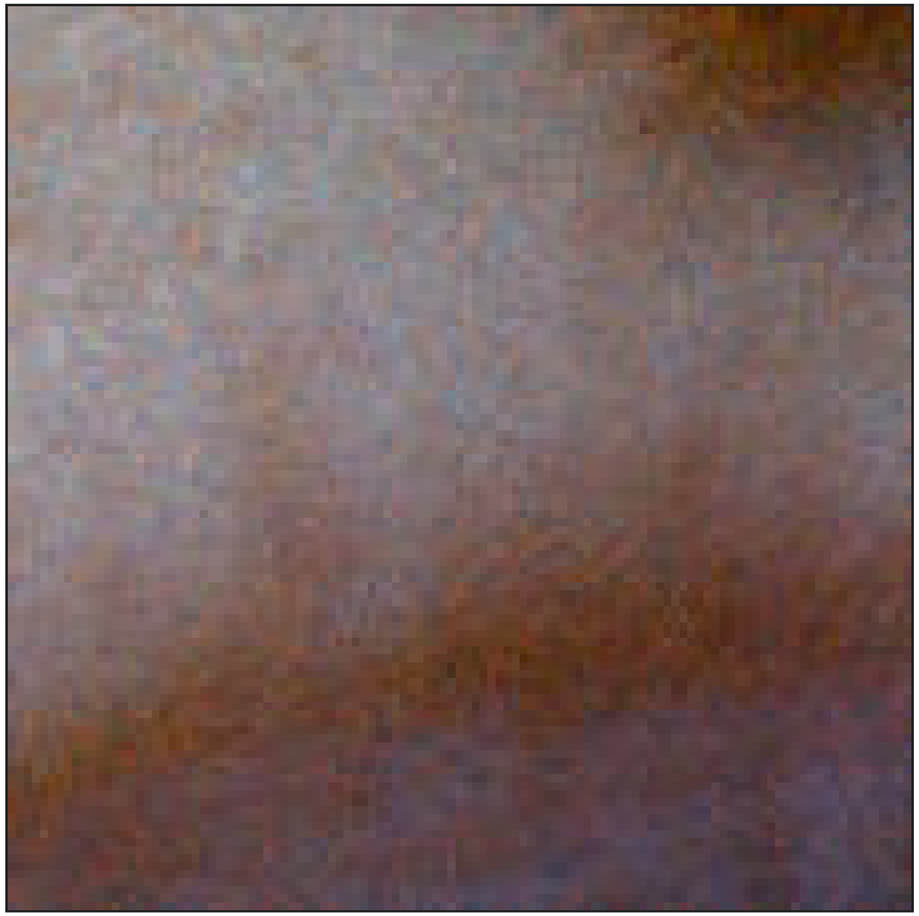


8 wavelengths

Hero wavelength vs. denoising in post

- works increasingly better at higher sample counts (fast preview?)

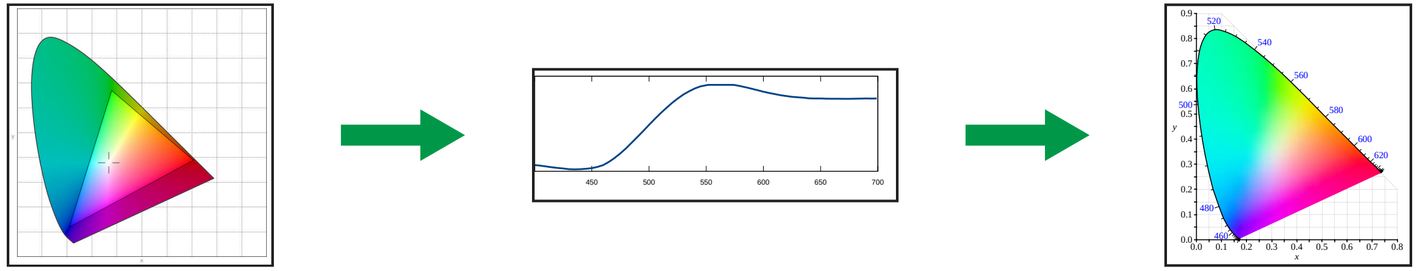
1 wavelength + Lab denoising

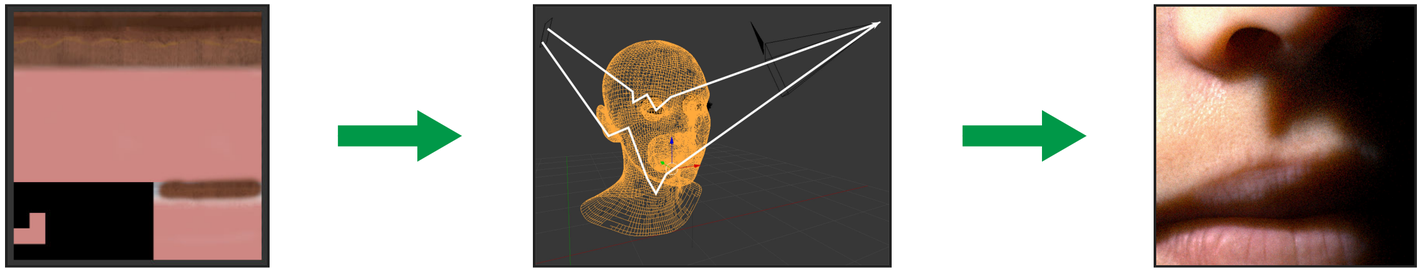


8 wavelengths

Colour pipeline

- well defined input colour space for rgb textures
  - upsampling rgb to spectra (lights and Bsdf parameters)
  - map to gamut of valid reflectances
- light transport in spectral domain
- map to output gamut for final frames (render is all colours!)

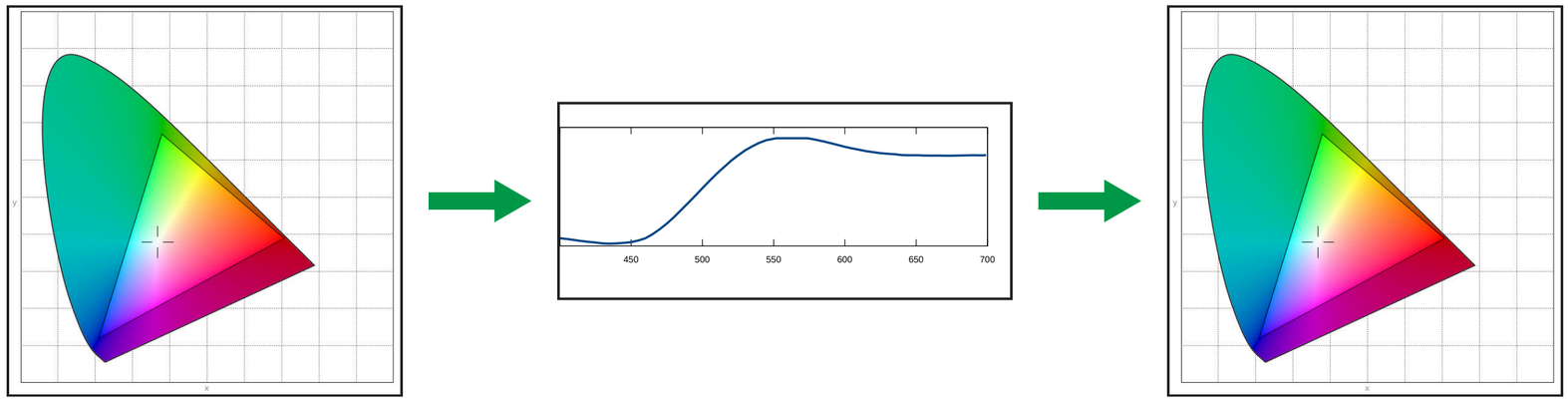




Upsampling RGB to spectrum

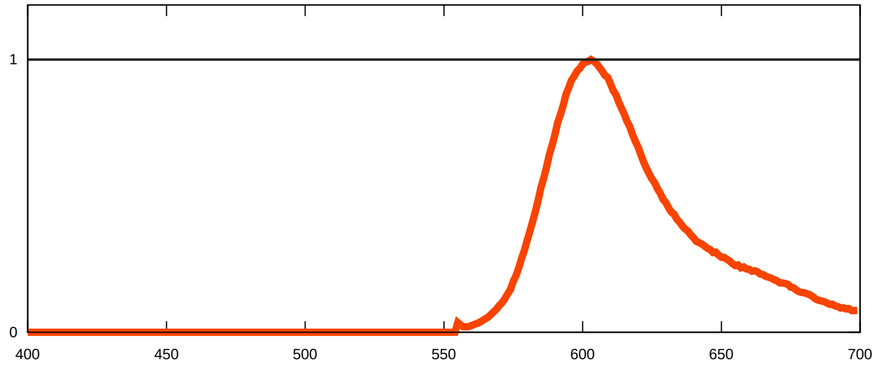
there are various approaches, common constraints:

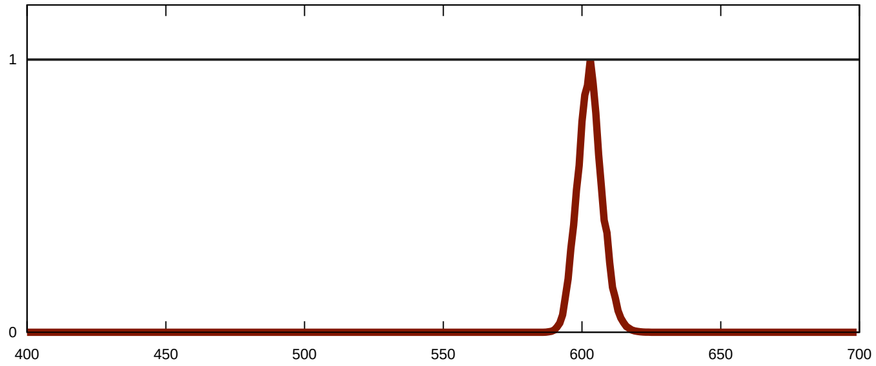
- "same colour":
  - input XYZ to match output spectrum lit by illuminant E seen by XYZ CMF
- smooth spectra
- non-negative spectra
- reflectances bounded by one
  - not a constraint for emission spectra



Gamut of reflectances, revisited

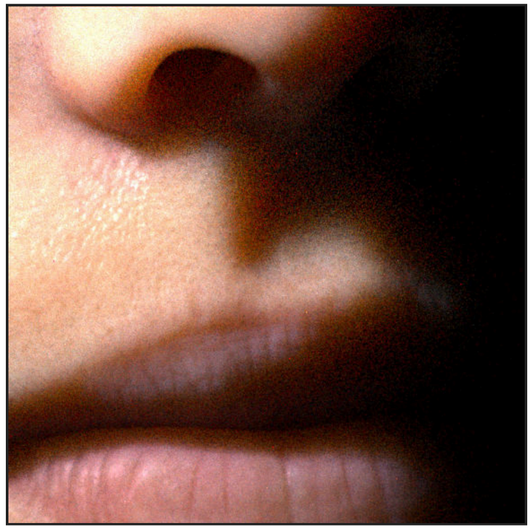
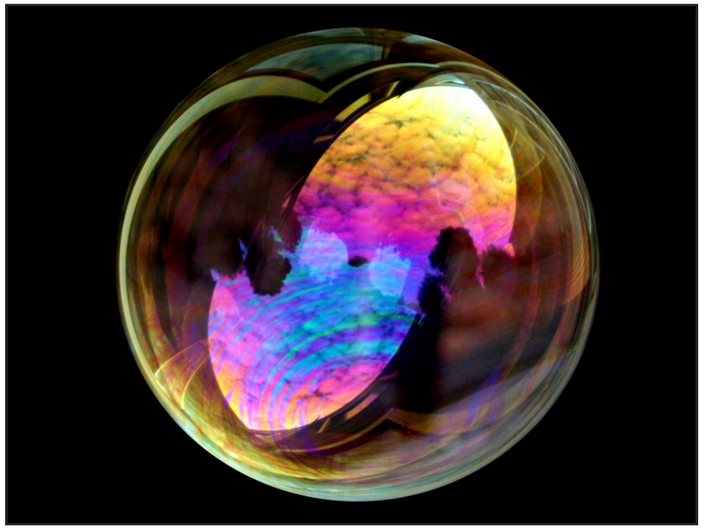
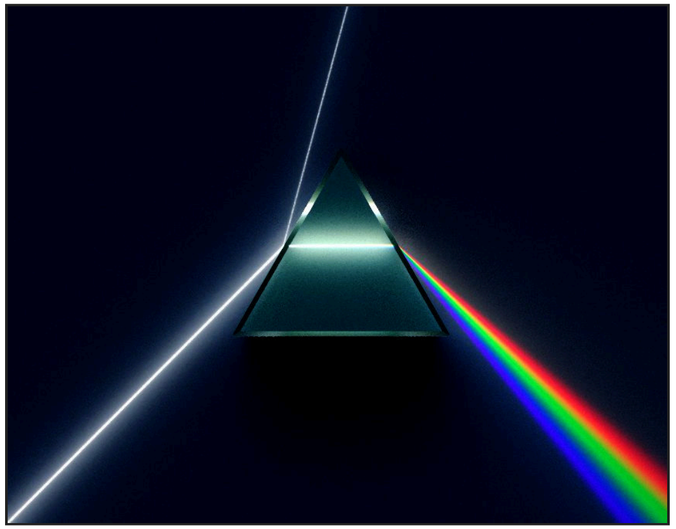
- limits the look of things
- overly saturated colours require fluorescence
  - how do you do bidirectional path construction?
  - may need to do clever colour grading instead
- for us more a feature: avoids radioactive glow





Spectral importance sampling

- chromatic extinction (skin)
  - actually see Disney paper this afternoon:
    Spectral and Decomposition Tracking for Rendering Heterogeneous Volumes
- wavelength dependent Bsdf (interference, iridescence, Rayleigh scattering)
- okay, dispersion

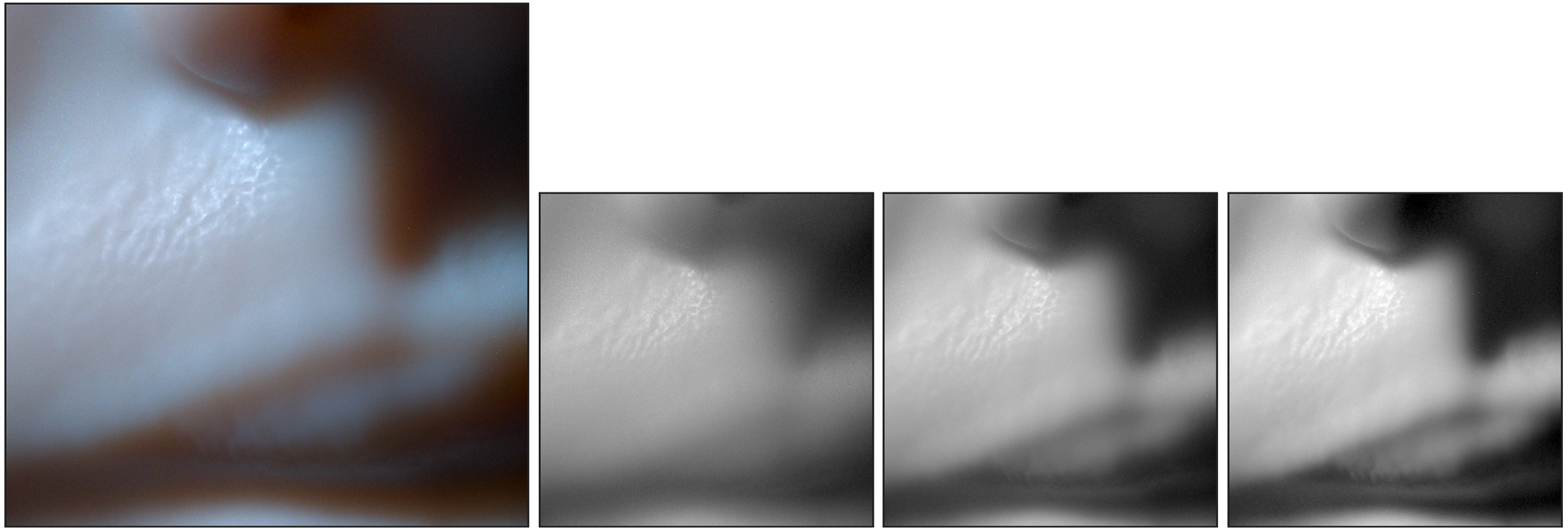




Chromatic extinction

- useful for skin: model red blur in SSS while keeping surface colour neutral

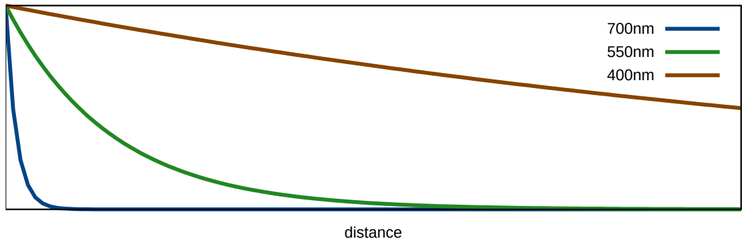
Chromatic extinction

- useful for skin: model red blur in SSS while keeping surface colour neutral



Chromatic extinction

- transmittance $\tau(\lambda, t)$ depends on wavelength through $\mu_t(\lambda)$



- Monte Carlo sample weight:

$$X = \frac{f(\mathbf{X})}{p(\mathbf{X})} = \frac{\tau(\lambda, t)}{\underbrace{\mu_t(\lambda) \cdot \tau(\lambda, t)}_{\to 0!}}$$

- unbounded if sampling the path with a different λ!
- solution: split and sample per wavelength
  - often implemented in RGB renderers, too (for 3 "wavelengths")
  - with MIS combination *exactly* what hero wavelength sampling does!

Photometric units

- design tools for lighters:
  - keep perceived brightness but change colour
  - predictable results (much like in the real world)
- got very good feedback on this from *War for the planet of the apes*

Conclusion: spectral rendering is here to stay

- precise colour reproduction
  - match camera and lights
  - match indirect by correct transport
- spectral rendering
  - Monte Carlo simulation of λ
  - control colour noise using hero wavelengths
- colour management
  - input, output, and reflectance gamut
- good importance sampling
  - especially chromatic media (skin!)

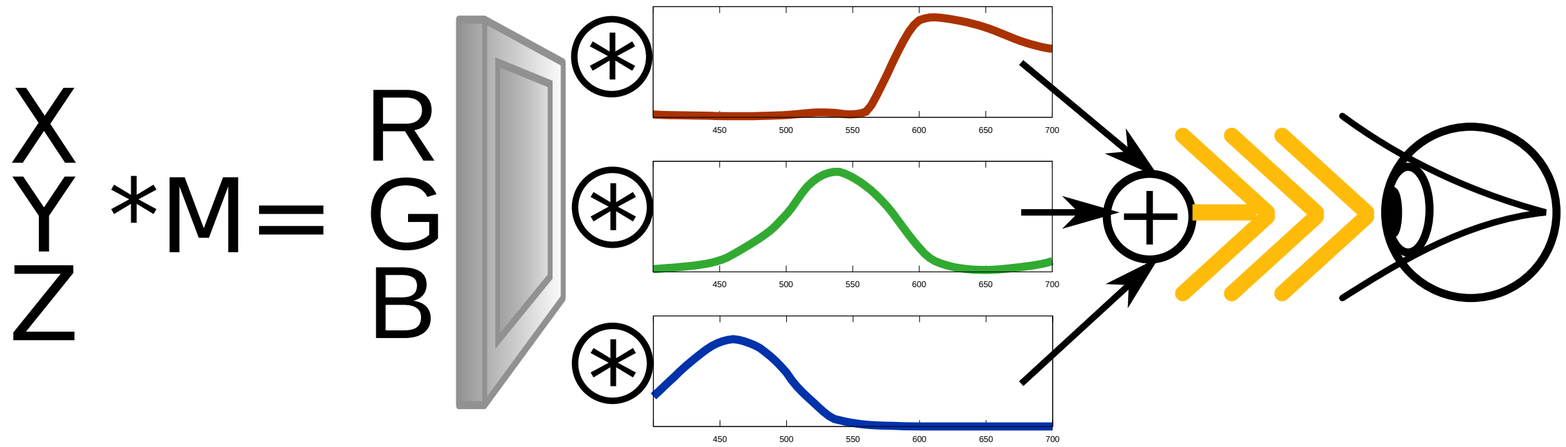
Thank you for listening!

thanks to the wikihuman project for the digital emily asset used in the skin renders

backup slides

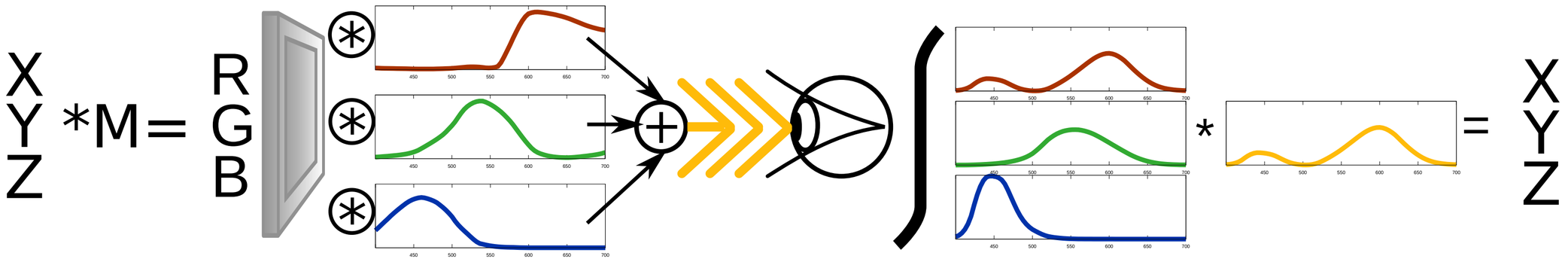
How do we reproduce colours for the eye?

- multiply monitor response curves by RGB values
- light will be projected onto XYZ CMF



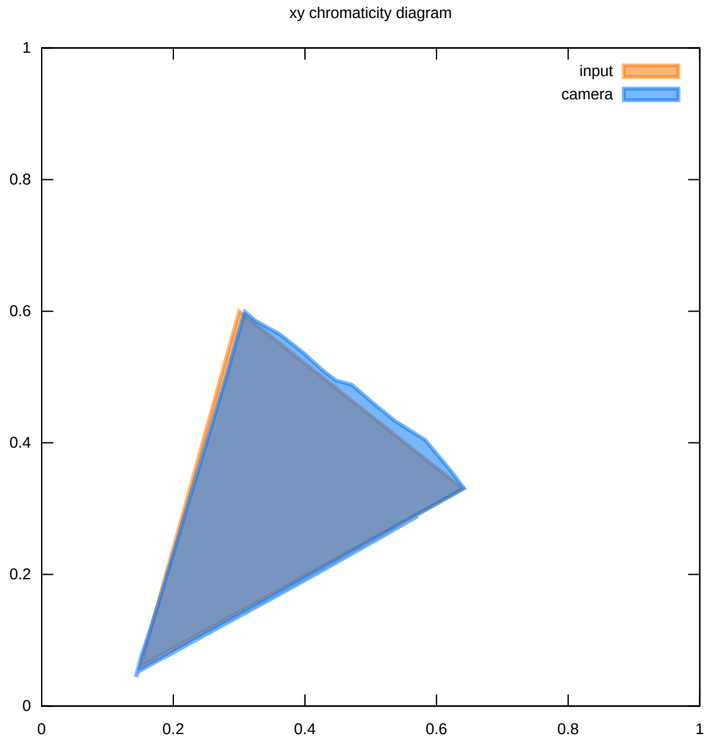
How do we reproduce colours for the eye?

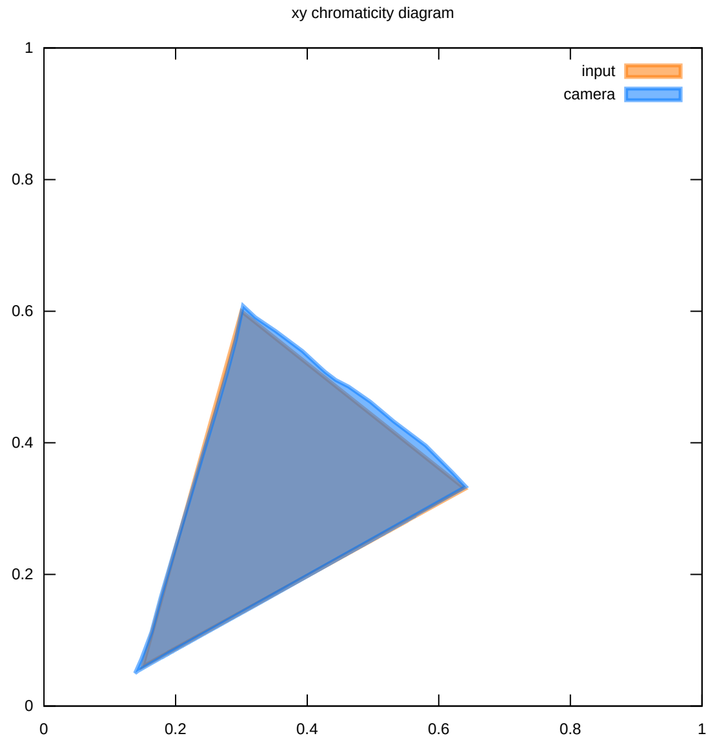
- full pipeline is linear!
- change R on RGB input:
  - multiply by spectrum $\bar{r}(\lambda)$ and CMF $\bar{x}(\lambda)$
  - $X = \int \bar{x}(\lambda) \cdot \bar{r}(\lambda) \cdot R \cdot d\lambda = R \cdot \underbrace{\int \bar{x}(\lambda) \cdot \bar{r}(\lambda) \cdot d\lambda}_{\text{const!}}$
- XYZ to RGB through linear matrix

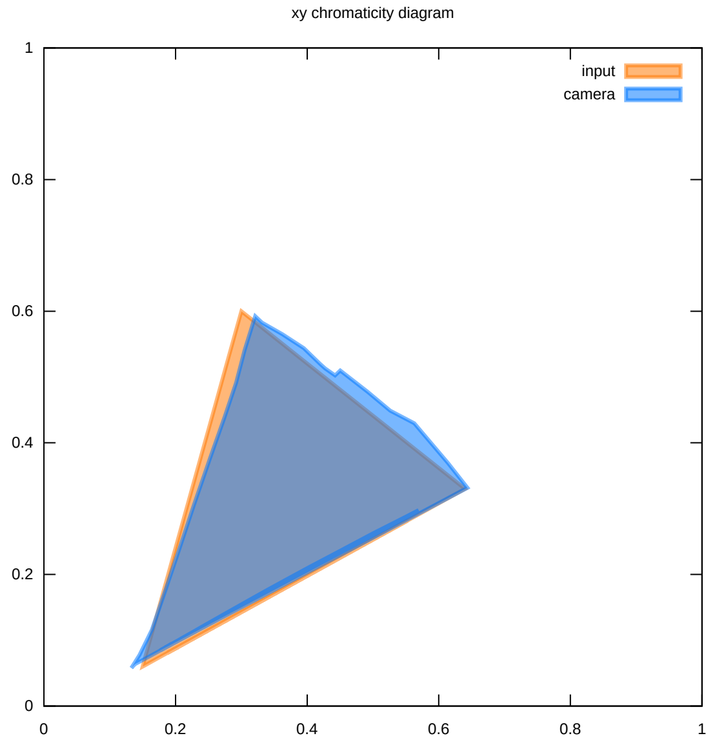


Colours within rec709

- round tripping:
  - scene (spectral) → camera (RGB) → matrix profile → output colour (XYZ)
- as seen by Canon 5DmII, Nikon D70, Kodak DCS 420

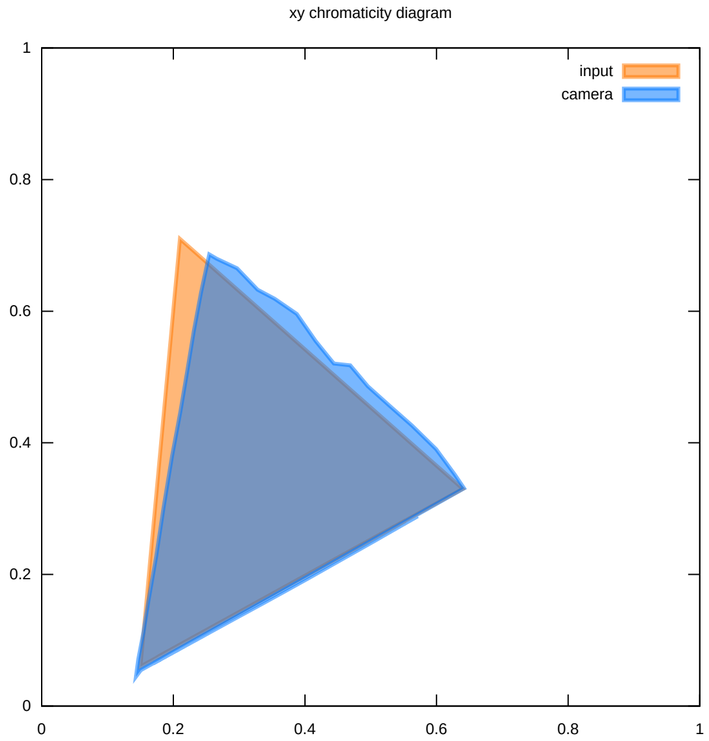


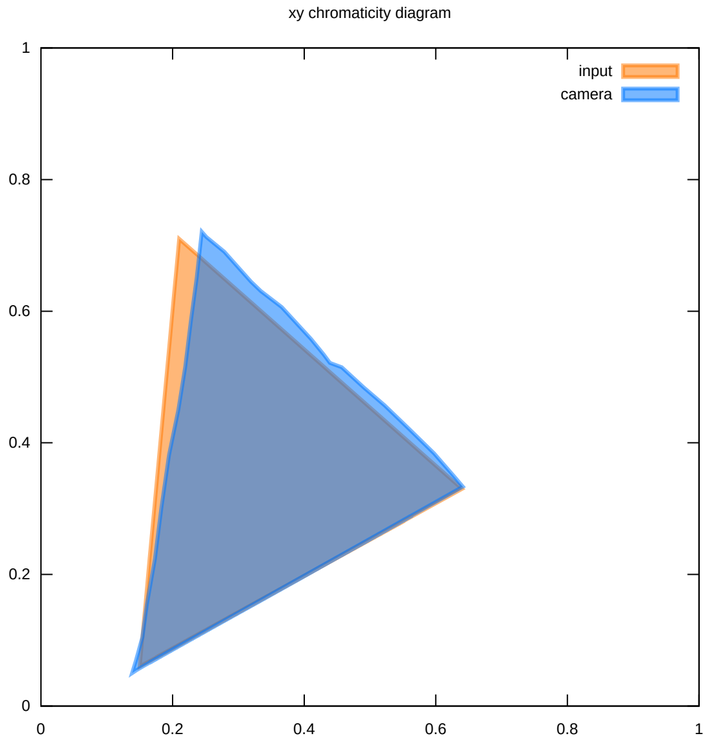


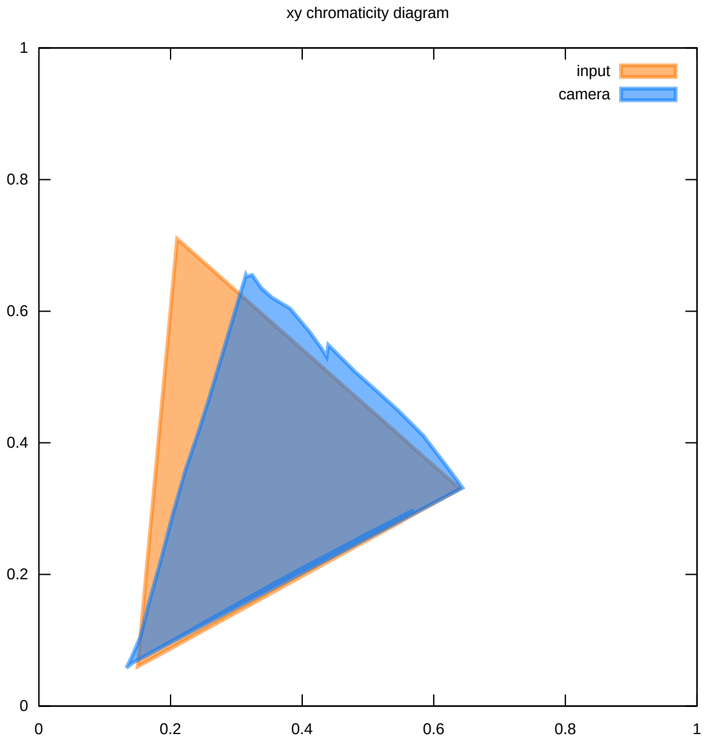


Colours within AdobeRGB

- round tripping:
  - scene (spectral) → camera (RGB) → matrix profile → output colour (XYZ)
- as seen by Canon 5DmII, Nikon D70, Kodak DCS 420

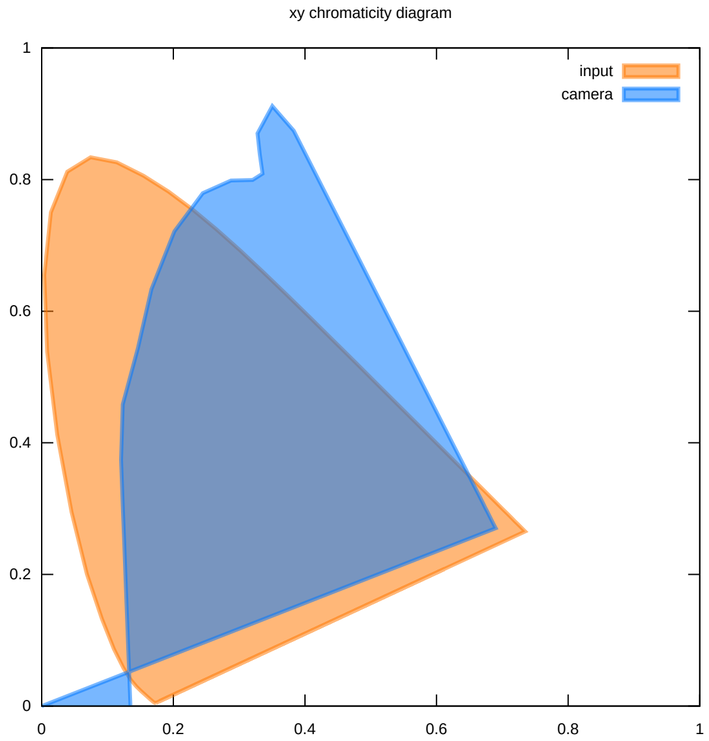


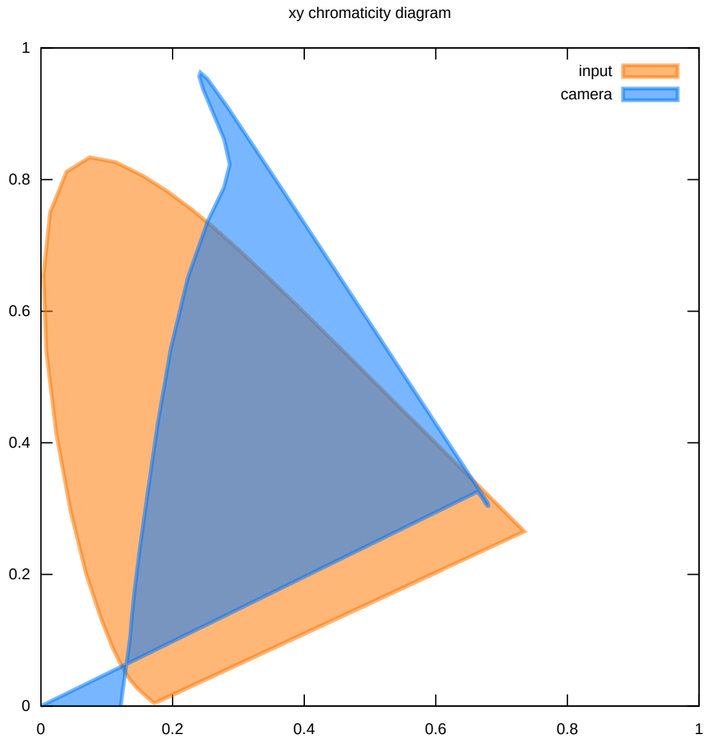


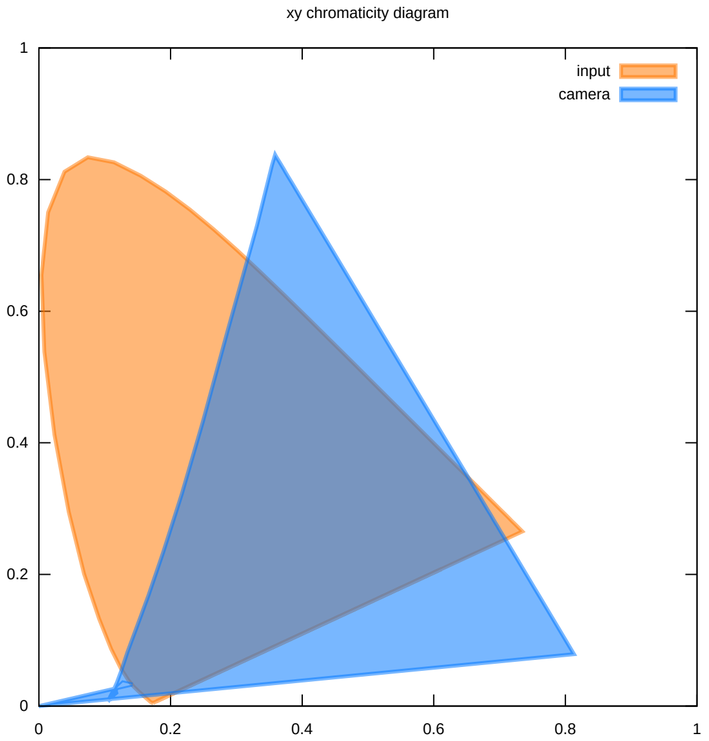


Colours within the spectral locus

- round tripping:
  - scene (spectral) → camera (RGB) → matrix profile → output colour (XYZ)
- note that rendering using XYZ CMF would yield ground truth!





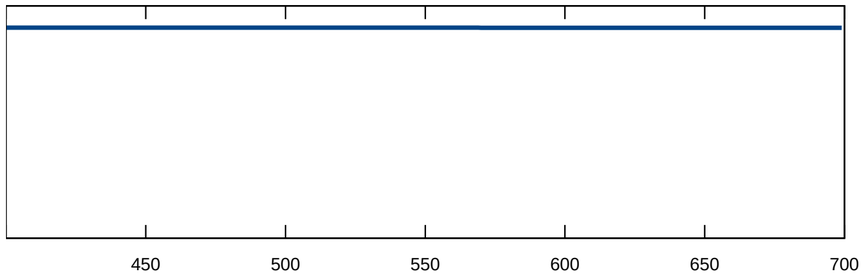


Upsampling RGB to spectrum

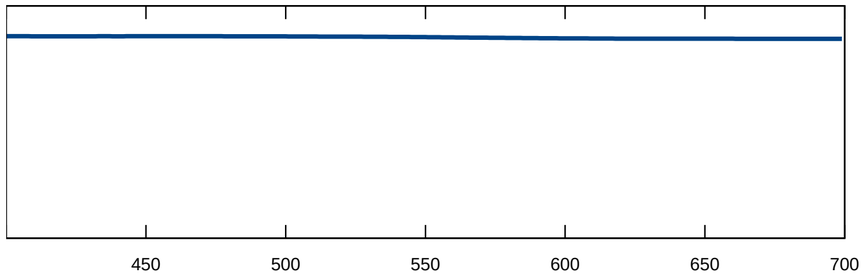
- [Smits 1999] for reflectances
  - optimise for your input colour space
  - use finer resolution than original paper
- [Meng et al. 2015] for reflectances and emission
  - works for all of XYZ, independent of input colour space
  - do gamut mapping as separate step (for reflectances)
- Manuka
  - the others optimise for illuminant E and XYZ colour matching functions
  - optimise specificly for current camera responsivity
  - constraint on emission spectra
    - a D65 spectrum maps to itself after round trip

Indirect lighting, reflectance in rec709 (1.00, 0.95, 0.90)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

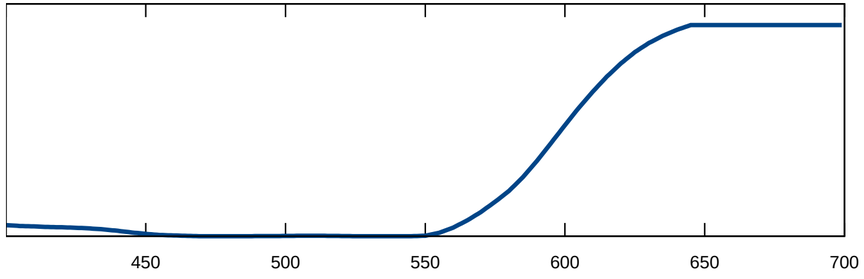


illuminant E white (whitepoint does not match rec709)

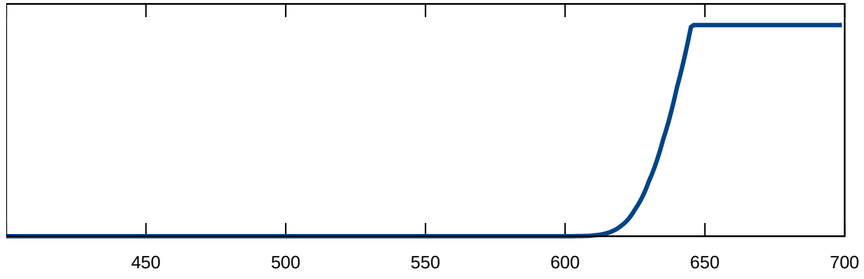


Indirect lighting, reflectance in rec709 (0.97, 0.01, 0.00)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

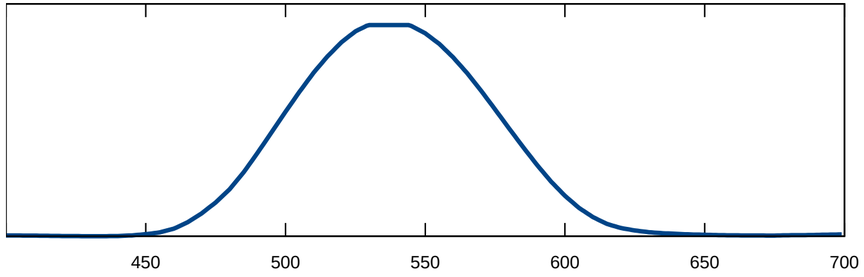


most saturated red (gamut mapped to valid reflectance)

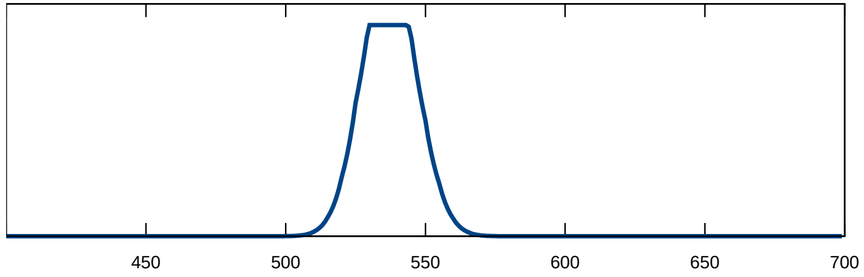


Indirect lighting, reflectance in rec709 (0.00, 0.86, 0.00)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

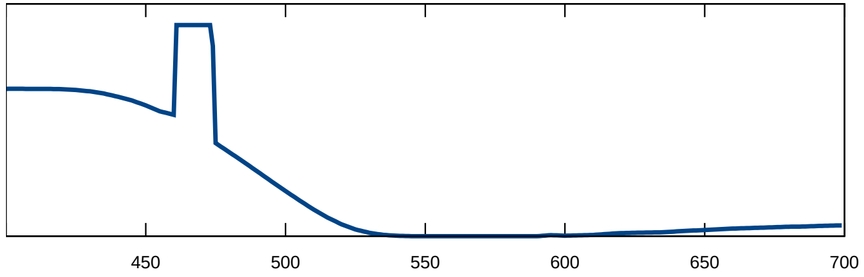


saturated green (gamut mapped to valid reflectance)

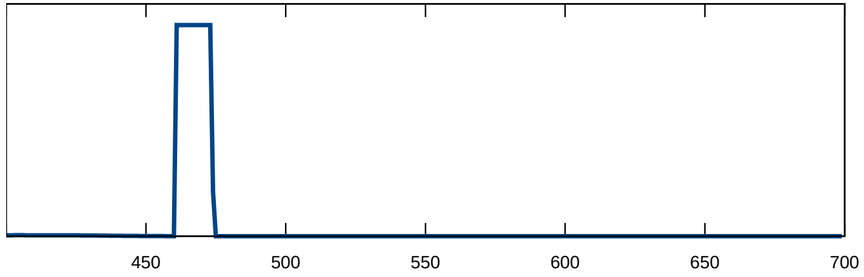


Indirect lighting, reflectance in rec709 (0.00, 0.00, 0.68)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

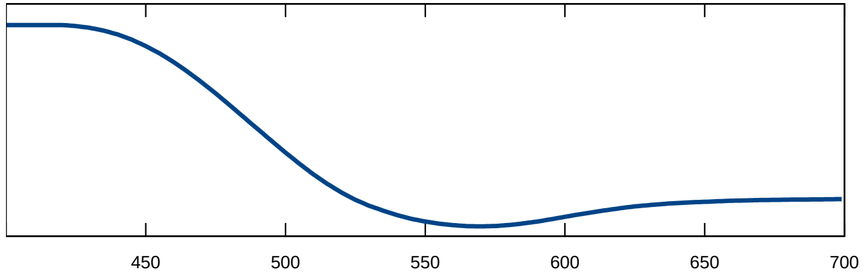


most saturated blue (gamut mapped to valid reflectance)

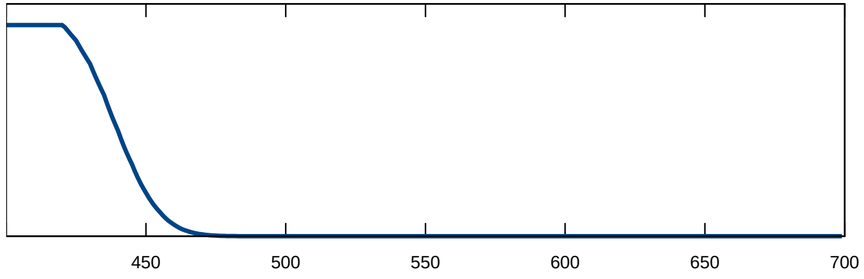


Indirect lighting, reflectance in rec709 (0.11, 0.09, 0.85)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

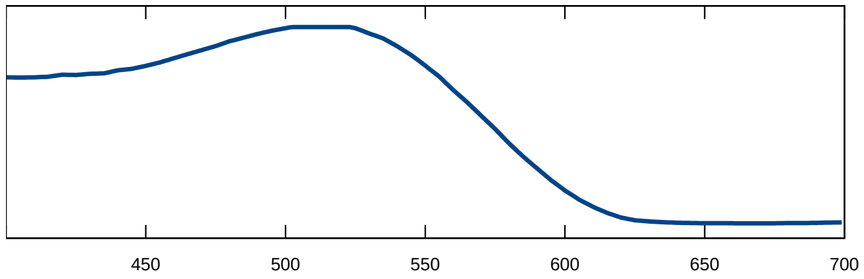


softer blue to reduce peaks

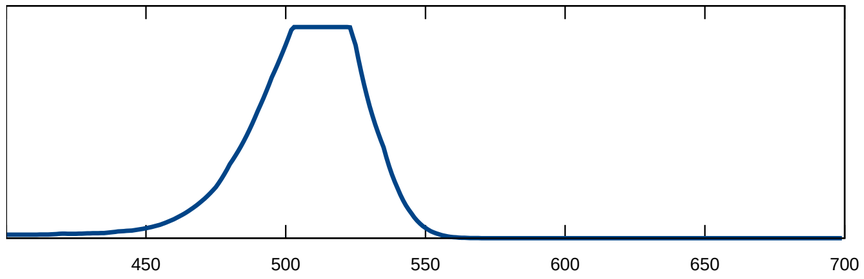


Indirect lighting, reflectance in rec709 (0.00, 0.81, 0.78)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															



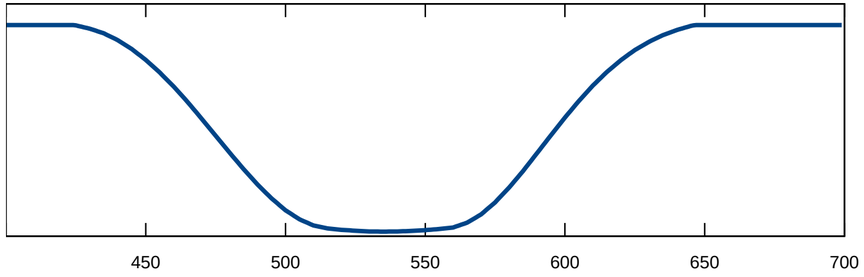
cyan

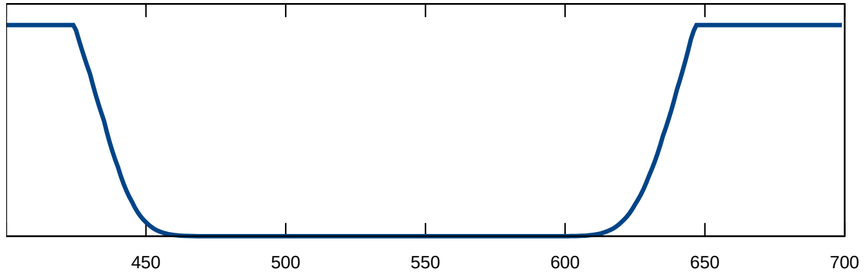


Indirect lighting, reflectance in rec709 (1.00, 0.00, 0.74)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

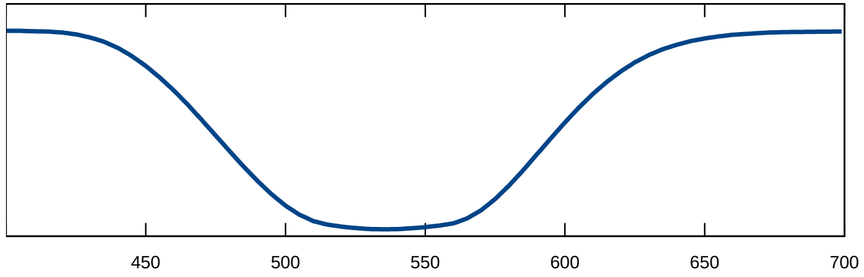
magenta



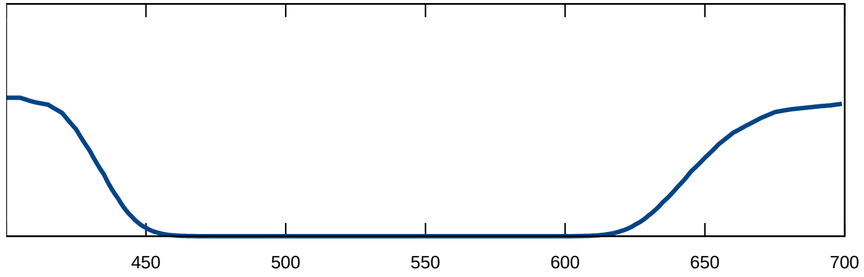


Indirect lighting, reflectance in rec709 (0.96, 0.00, 0.72)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															



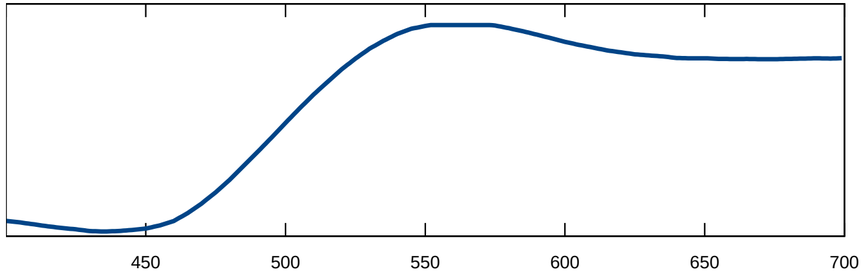
magenta
(not saturated)

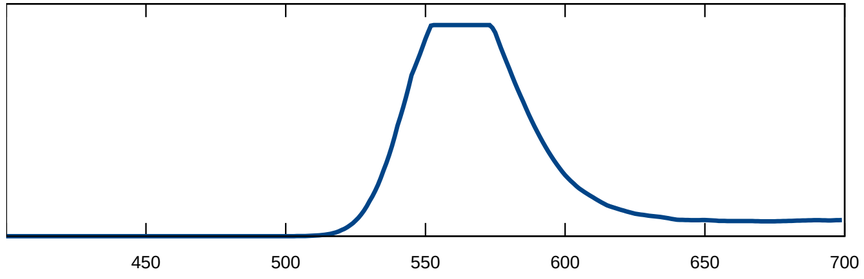


Indirect lighting, reflectance in rec709 (1.00, 0.89, 0.00)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

yellow





Indirect lighting, reflectance in rec709 (0.93, 0.06, 0.00)

#bounces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
XYZ															
ACES															
ACEScg															
Adobe RGB															
rec709															
spectral															

measured spectrum