

Computer Vision SDML Book Club

July 18, 2025



TEXTS IN COMPUTER SCIENCE

Computer Vision

Algorithms and Applications
Second Edition



Richard Szeliski

 Springer

Chapter 2 contents

- Geometric primitives and transformations
- Photometric image formation
- The digital camera

Geometric primitives and transformations

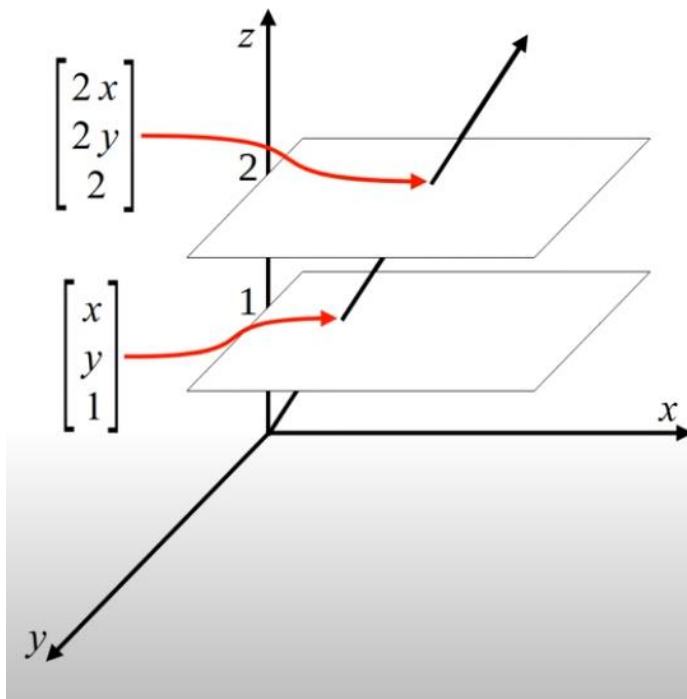
- Geometric primitives and transformations
 - Primitives: points, lines, and planes
 - 2D transformations
 - 3D transformations
 - 3D rotations
 - 3D to 2D projections
 - Lens distortions
- Photometric image formation
- The digital camera

Homogeneous coordinates [1]

- Homogeneous coordinates are also called projective coordinates
- Adds an extra dimension (2D has 3 coordinates, 3D has 4 coordinates)
 - Representation is no longer unique
- Benefits
 - In addition to usual rotation and scaling, now translation and projection can also be represented by a matrix multiplication
 - Combinations of transformations can use fast chained matrix multiplications
 - Points at infinity can be represented and manipulated with finite coordinates

Homogeneous coordinates [2]

- <https://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/>
- https://youtu.be/S7zli1qWKfo?si=r4CEDaG_Z0yDvg6j



In homogeneous coordinates, a point and its scaled versions are same

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} \quad w \neq 0$$

They're the same point on different z planes

$$\begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

If z component is zero then the point is at infinity

Primitives

- 2D points – (x, y) or $(\tilde{x}, \tilde{y}, \tilde{w})$. Notation $\bar{\mathbf{x}} = (x, y, 1)$
- 2D lines – equation $\bar{\mathbf{x}} \cdot \tilde{\mathbf{l}} = 0$ defines line perpendicular to $\tilde{\mathbf{l}}$
- 2D conics
- 3D points – (x, y, z) or $(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w})$
- 3D planes – equation $\bar{\mathbf{x}} \cdot \tilde{\mathbf{m}} = 0$ defines line perpendicular to $\tilde{\mathbf{m}}$
- 3D lines – less elegant; can use 2 points, 2 points in 2 planes, or Plücker coordinates
- 3D quadrics

2D transformations






Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3 \times 3}$	8	straight lines	

Table 2.1 *Hierarchy of 2D coordinate transformations, listing the transformation name, its matrix form, the number of degrees of freedom, what geometric properties it preserves, and a mnemonic icon. Each transformation also preserves the properties listed in the rows below it, i.e., similarity preserves not only angles but also parallelism and straight lines. The 2×3 matrices are extended with a third $[\mathbf{0}^T \ 1]$ row to form a full 3×3 matrix for homogeneous coordinate transformations.*

Additional 2D transformations

- Stretch/squash
- Planar surface flow
- Bilinear interpolant

3D transformations






Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{3 \times 4}$	3	orientation	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{3 \times 4}$	6	lengths	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{3 \times 4}$	7	angles	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{3 \times 4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{4 \times 4}$	15	straight lines	

Table 2.2 *Hierarchy of 3D coordinate transformations. Each transformation also preserves the properties listed in the rows below it, i.e., similarity preserves not only angles but also parallelism and straight lines. The 3×4 matrices are extended with a fourth $[\mathbf{0}^T \ 1]$ row to form a full 4×4 matrix for homogeneous coordinate transformations. The mnemonic icons are drawn in 2D but are meant to suggest transformations occurring in a full 3D cube.*

3D rotations

3D rotations can be parameterized multiple ways

- Euler angles – product of three rotations around three cardinal axes
 - Generally bad b/c order matters and cannot always move smoothly
- Axis/angle (exponential twist) – a rotation axis $\hat{\mathbf{n}}$ and angle θ
 - Rodrigues' formula:

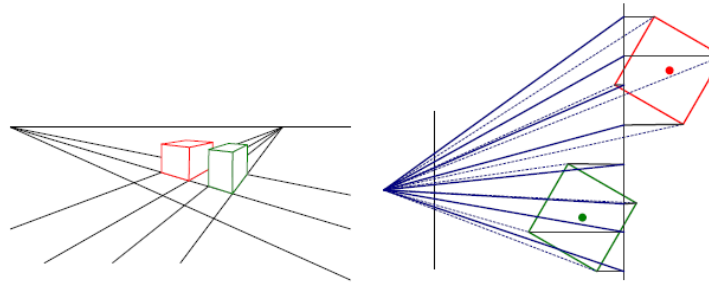
$$\mathbf{R}(\hat{\mathbf{n}}, \theta) = \mathbf{I} + \sin \theta [\hat{\mathbf{n}}]_{\times} + (1 - \cos \theta) [\hat{\mathbf{n}}]_{\times}^2$$

- Unit quaternions – a unit length 4-vector living on the unit sphere
 - \mathbf{q} and $-\mathbf{q}$ represent same rotation, but otherwise unique
 - Easy to compose or invert quaternion rotations
 - Popular for pose and pose interpolation
 - Convenient for spherical linear interpolation (slerp)

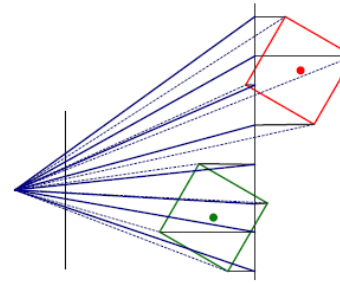
3D to 2D projections [1]

- Orthographic projection – simply drops the z component
- Scaled orthography – equivalent to first projection to local image plane, which can be different for different objects, then perspective projection
- Para-perspective – similar but first projection is along line of sight
- Perspective – dividing points by their z component
- Sometimes two-step first projects to *normalized device coordinates* $(x, y, z) \in [-1, 1] \times [-1, 1] \times [0, 1]$ then rescales to pixels using a *viewport* transformation

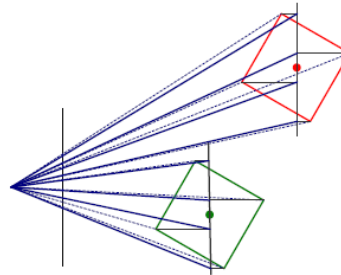
3D to 2D projections [2]



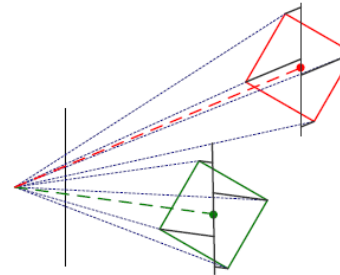
(a) 3D view



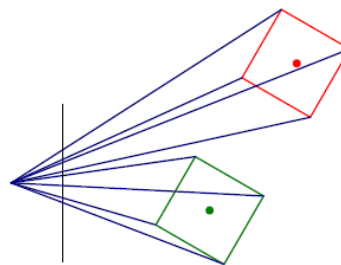
(b) orthography



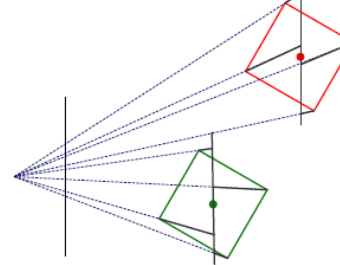
(c) scaled orthography



(d) para-perspective



(e) perspective



(f) object-centered

Camera intrinsics [1]

“Once we have projected a 3D point through an ideal pinhole using a projection matrix, we must still transform the resulting coordinates according to the pixel sensor spacing and the relative position of the sensor plane to the origin”

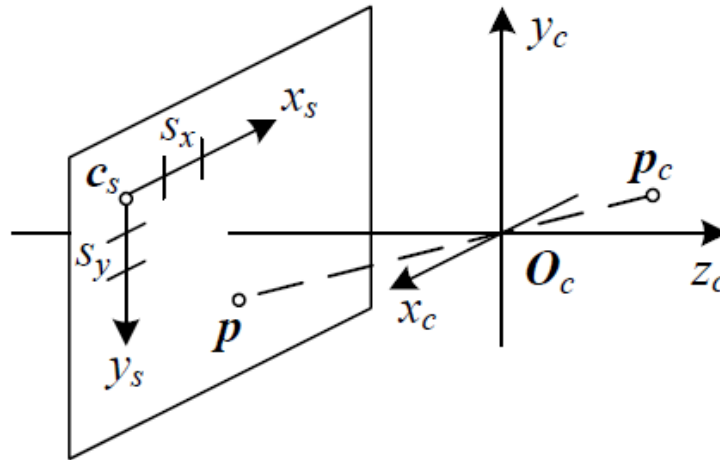


Figure 2.8 *Projection of a 3D camera-centered point \mathbf{p}_c onto the sensor planes at location \mathbf{p} . \mathbf{O}_c is the optical center (nodal point), \mathbf{c}_s is the 3D origin of the sensor plane coordinate system, and s_x and s_y are the pixel spacings.*

Camera intrinsics [2]

- Mapping 2D pixel coordinates to 3D rays:

$$\mathbf{p} = \begin{bmatrix} \mathbf{R}_s & \mathbf{c}_s \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \mathbf{M}_s \bar{\mathbf{x}}_s$$

- “We can therefore write the complete projection between \mathbf{p}_c and a homogeneous version of the pixel address $\tilde{\mathbf{x}}_s$ as:”

$$\tilde{\mathbf{x}}_s = \alpha \mathbf{M}_s^{-1} \mathbf{p}_c = \mathbf{K} \mathbf{p}_c$$

- \mathbf{K} is called the *calibration matrix* and describes the camera *intrinsics*

Additional discussion

- Note that the focal length depends on the units used to measure pixels
 - For a 35mm film camera the width is in mm
 - For digital images it is more convenient to work in pixels
- Camera matrix – a 3×4 matrix combining intrinsics and extrinsics
- Plane plus parallax (projective depth)
- Mapping from one camera to another
- Object-centered projection

Lens distortions

- Radial distortion – coordinates displaced proportional to radial dist.
- Fisheye lenses



(a)



(b)



(c)

Figure 2.13 *Radial lens distortions: (a) barrel, (b) pincushion, and (c) fisheye. The fisheye image spans almost 180° from side-to-side.*

Photometric image formation

- Geometric primitives and transformations
- Photometric image formation
 - Lighting
 - Reflectance and shading
 - Optics
- The digital camera

Lighting

- Point light source
 - Has an intensity and a color spectrum (distribution over wavelengths)
- Area light sources are more complicated
- Environment map – can map complex incident illumination
 - Representations include a collection of cubical faces, a single longitude-latitude map, or the image of a reflecting sphere

Reflectance and shading

- The Bidirectional Reflectance Distribution Function (BRDF)
 - General function of incident direction, reflected direction, and wavelength
 - Can simplify for *isotropic* (no preferred directions)
- Diffuse reflection – uniform in all directions
 - Quantity depends on angle between incident light direction and surface
- Specular reflection – gloss or highlights 180 degrees around normal
- Phong shading – adds term for *ambient illumination*
 - Representations include a collection of cubical faces, a single longitude-latitude map, or the image of a reflecting sphere
- Global illumination – ray tracing and radiosity

Optics [1]

- Real world isn't an ideal pinhole camera
- Focus, aperture, circle of confusion, and depth of field

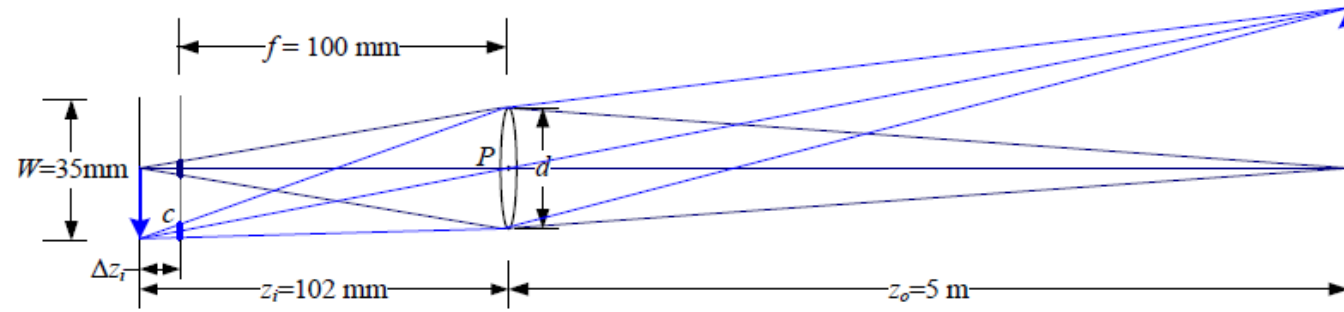


Figure 2.19 A thin lens of focal length f focuses the light from a plane at a distance z_o in front of the lens onto a plane at a distance z_i behind the lens, where $\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}$. If the focal plane (vertical gray line next to c) is moved forward, the images are no longer in focus and the circle of confusion c (small thick line segments) depends on the distance of the image plane motion Δz_i relative to the lens aperture diameter d . The field of view (f.o.v.) depends on the ratio between the sensor width W and the focal length f (or, more precisely, the focusing distance z_i , which is usually quite close to f).

Optics [2]

- Real lenses have five classic geometric aberrations
 - spherical aberration, coma, astigmatism, curvature of field, and distortion
- Chromatic aberration – index of refraction varies by wavelength
 - Different colors focus at slightly different distances
 - Compound lenses reduce aberrations
- Vignetting – brightness decreases towards the edges
 - Natural vignetting decrease related to fourth power of the off-axis angle

The digital camera

- Geometric primitives and transformations
- Photometric image formation
- The digital camera
 - Sensing pipeline
 - Sampling and aliasing
 - Color
 - Compression

Sensing pipeline [1]

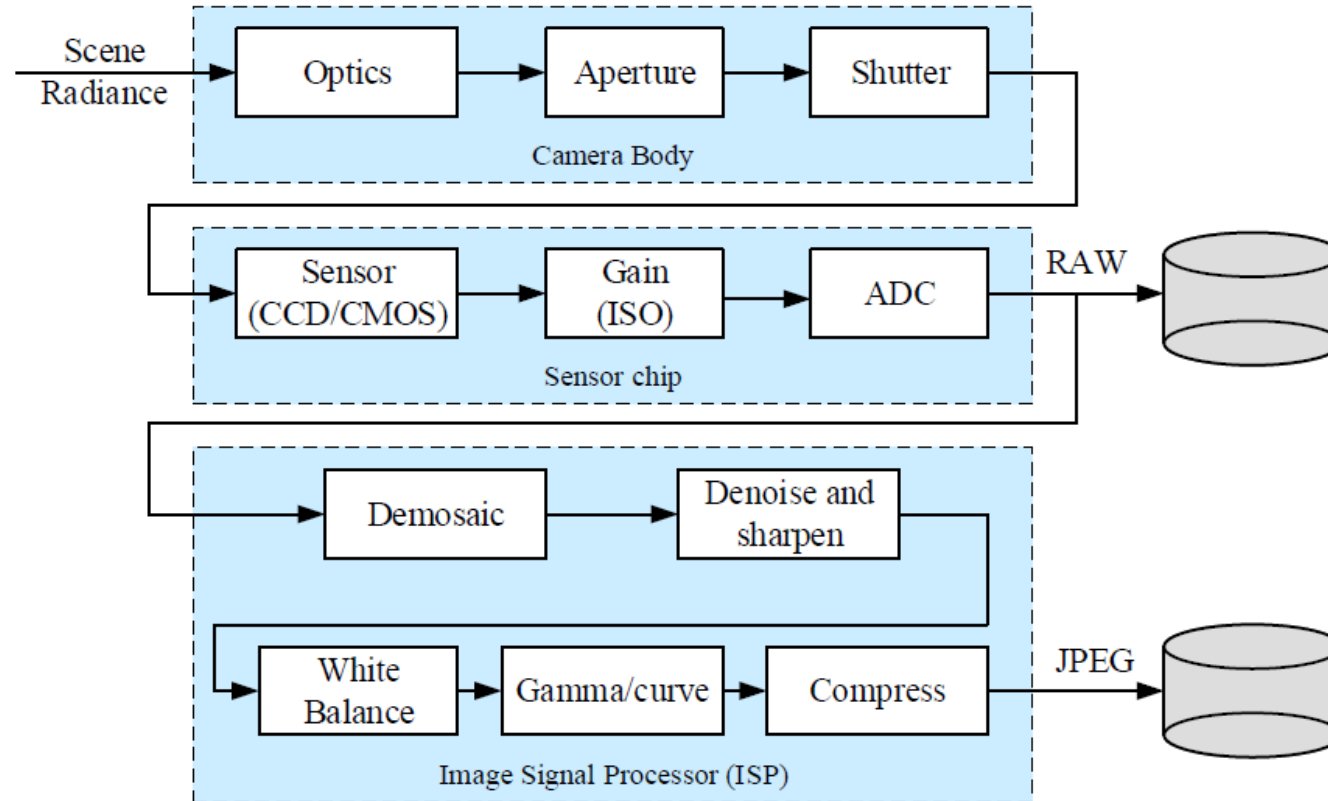


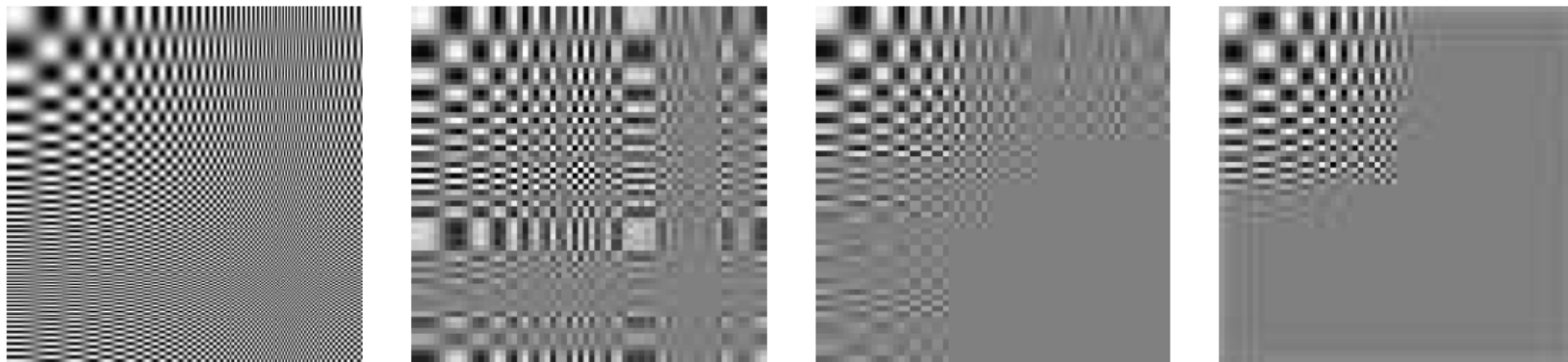
Figure 2.23 *Image sensing pipeline, showing the various sources of noise as well as typical digital post-processing steps.*

Sensing pipeline [2]

- Two main kinds of sensors are CCD and CMOS
- Factors affecting performance
 - Shutter Speed
 - Sampling pitch – spacing between adjacent sensor cells
 - Fill factor – fraction of total area that is active sensing
 - Chip size
 - Analog gain – boosts sensed signal; may be adjusted through ISO setting
 - Sensor noise
 - ADC resolution – resolution (# of bits) and noise level (affects useful bits)
 - Digital post-processing
 - Newer image sensors

Sampling and aliasing

- Digital sampling can create unpleasing aliasing that looks like lower frequencies
- Averaging



(a)

(b)

(c)

(d)

Figure 2.26 Aliasing of a two-dimensional signal: (a) original full-resolution image; (b) downsampled $4 \times$ with a 25% fill factor box filter; (c) downsampled $4 \times$ with a 100% fill factor box filter; (d) downsampled $4 \times$ with a high-quality 9-tap filter. Notice how the higher frequencies are aliased into visible frequencies with the lower quality filters, while the 9-tap filter completely removes these higher frequencies.

Color

- Different wavelengths get represented in RGB values

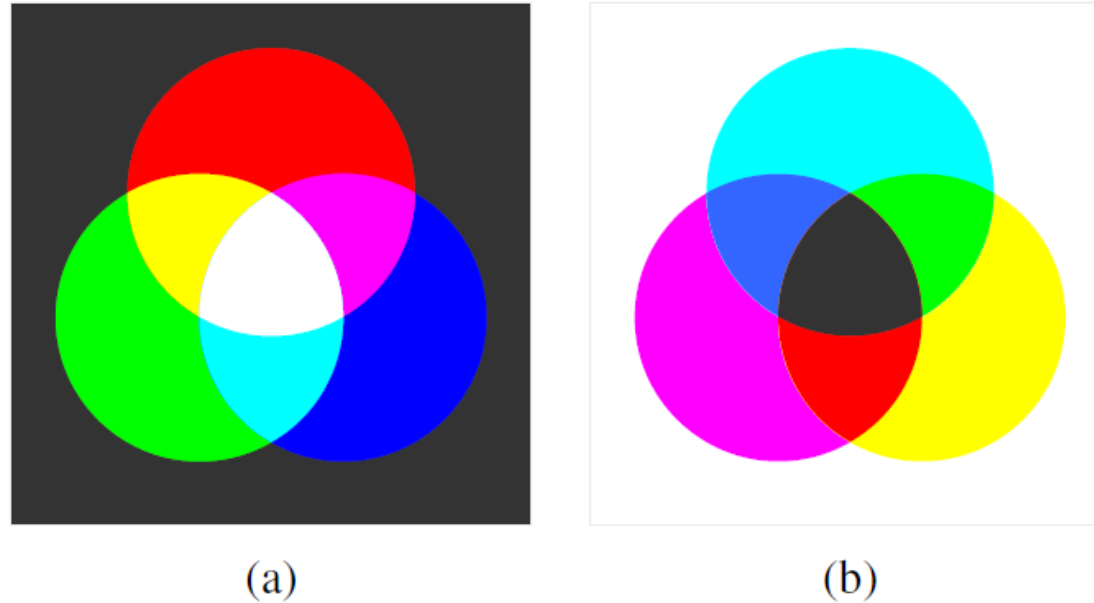


Figure 2.28 *Primary and secondary colors: (a) additive colors red, green, and blue can be mixed to produce cyan, magenta, yellow, and white; (b) subtractive colors cyan, magenta, and yellow can be mixed to produce red, green, blue, and black.*

CIE RGB and XYZ [1]

- XYZ avoided negative values in biological color matching

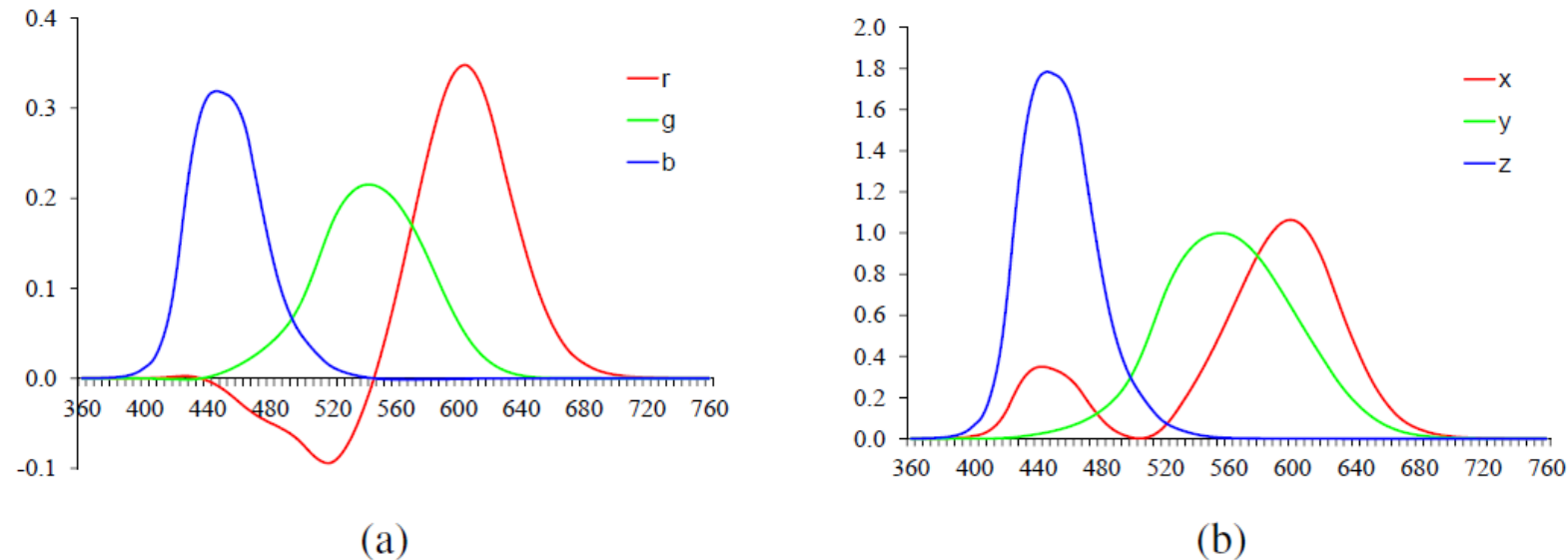


Figure 2.29 Standard CIE color matching functions: (a) $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$ color spectra obtained from matching pure colors to the $R=700.0\text{nm}$, $G=546.1\text{nm}$, and $B=435.8\text{nm}$ primaries; (b) $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ color matching functions, which are linear combinations of the $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$ spectra.

CIE RGB and XYZ [2]

- Transformation from RGB to XYZ:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Y is *luminance*, or perceived relative brightness
- The *chromaticity coordinates* sum to 1

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}, \quad z = \frac{Z}{X + Y + Z}$$

XYZ and $L^*a^*b^*$ color space

- XYZ doesn't predict how people perceive differences in color/luminance

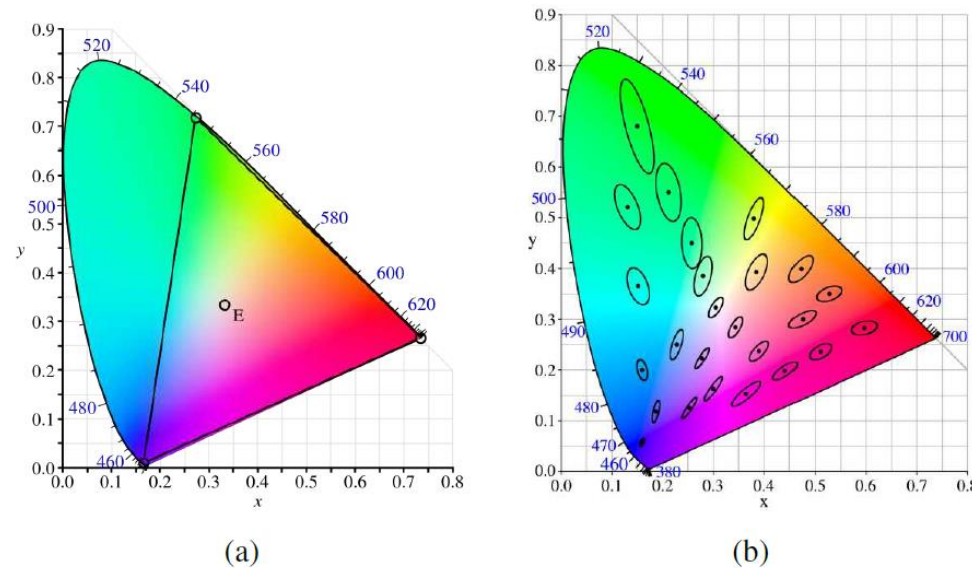


Figure 2.30 CIE chromaticity diagram, showing the pure single-wavelength spectral colors along the perimeter and the white point at E, plotted along their corresponding (x, y) values. (a) the red, green, and blue primaries do not span the complete gamut, so that negative amounts of red need to be added to span the blue–green range; (b) the MacAdam ellipses show color regions of equal discriminability, and form the basis of the Lab perceptual color space.

L*a*b* color space

- L* is called *lightness* $L^* = 116f\left(\frac{Y}{Y_n}\right)$

$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases}$$

- The a* and b* are defined:

$$a^* = 500 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad \text{and} \quad b^* = 200 \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right]$$

Color filter arrays and Color balance

- Most still and video cameras use a *color filter array*, most commonly the *Bayer pattern*

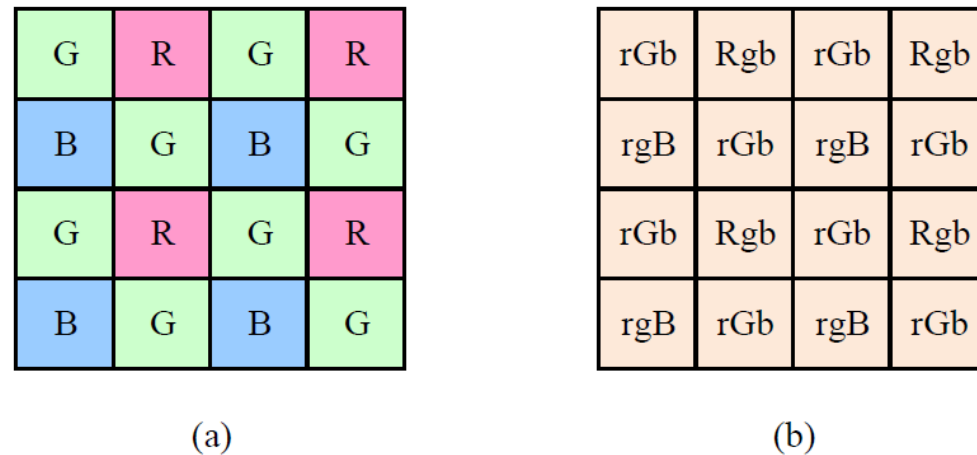


Figure 2.31 Bayer RGB pattern: (a) color filter array layout; (b) interpolated pixel values, with unknown (guessed) values shown as lower case.

- Color balance – compensation for illumination not being pure white

Gamma

- In early TVs, the phosphors responded non-linearly to voltage, so the gamma translated between raw voltage and brightness
- Many calculations require a linear space, so gamma in image values can be problematic

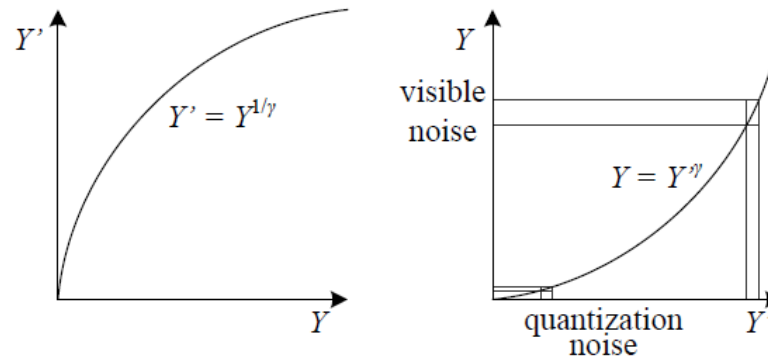


Figure 2.32 Gamma compression: (a) The relationship between the input signal luminance Y and the transmitted signal Y' is given by $Y' = Y^{1/\gamma}$. (b) At the receiver, the signal Y' is exponentiated by the factor γ , $\hat{Y} = Y'^{\gamma}$. Noise introduced during transmission is squashed in the dark regions, which corresponds to the more noise-sensitive region of the visual system.

Other color spaces

- NTSC in US used YIQ and PAL in Europe used YUV
 - Y was a *luma* channel, and the other two were lower resolution color info
- More recent digital video uses YCbCr with different scale factors
- Hue, saturation, value (HSV)
- Yxy
- Color ratios: $r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}$

Compression

- Usually compression starts by converting to YCbCr (or closely related)
- Cb and Cr are often subsampled (JPEG does both horiz. and vert.)
- Discrete cosine transform (DCT)
 - MPEG and JPEG use 8x8 DCT transforms, newer variants use smaller blocks
 - Alternatives use wavelets and lapped transforms
- Coefficients are quantized into small ints, such as with Huffman code
 - The *quality* setting in JPEG mainly controls the quantization step size
- Video also uses block-based motion compensation