Course Reader
Lecture 2

Camera Models, Projective Geometry
and Colour Models

Introduction to Image Analysis
ARTORG Center for Biomedical Engineering Research
University of Bern

Spring 2026

This reader covers all examinable material from Lecture 2.
Key definitions, derivations, and conceptual results are highlighted.

Contents

1 Introduction: From Scene to Image **2**

2 Camera Obscura and Historical Context **2**

3 The Pinhole Camera Model **2**
3.1 Core Geometry . . . 2
3.2 Key Observations . . . 3
3.3 Worked Example . . . 3

4 Homogeneous Coordinates and the Camera Matrix **3**
4.1 Why Homogeneous Coordinates? . . . 3
4.2 Projection as Matrix Multiplication . . . 4
4.3 The Full Camera Matrix . . . 4
4.4 Intrinsic vs. Extrinsic Parameters . . . 4
4.5 Why Homogeneous Coordinates Matter Algorithmically . . . 5

5 Perspective Geometry and Vanishing Points **5**
5.1 Why Parallel Lines Converge . . . 5
5.2 Types of Perspective . . . 5
5.3 The Horizon Line . . . 6
5.4 Applications in Medical Imaging . . . 6
5.5 Worked Example: Vanishing Point Computation . . . 6

6 Lenses, Sensors, and the Imaging Pipeline **6**
6.1 From Pinhole to Lens . . . 6
6.2 Key Trade-offs Introduced by Lenses . . . 7
6.3 Digital Sensors: CCD and CMOS . . . 7
6.4 Bayer Pattern and Demosaicing . . . 7

7 Human Vision and Colour Perception **7**
7.1 Photoreceptors: Rods and Cones . . . 8
7.2 Trichromatic Theory and Metamerism . . . 8
7.3 Colour Blindness . . . 8

8 Colour Spaces **8**
8.1 RGB: The Additive Colour Model . . . 8
8.2 HSV / HSI: Separating Colour from Brightness . . . 9
8.3 CIE LAB: Perceptual Uniformity . . . 9
8.4 CIE Chromaticity and Colour Gamuts . . . 10
8.5 CMYK and YCbCr . . . 10
8.6 Gamma Correction . . . 11
8.7 White Balance . . . 11

9 Choosing the Right Colour Space **11**

10 Summary of Key Equations **12**

11 Practice Questions **12**

12 Recommended Reading **13**

1 Introduction: From Scene to Image

Every digital image is the end product of a chain of physical processes. Understanding that chain is essential for interpreting what algorithms see and for correcting the artifacts they encounter.

1. **Illumination.** A light source (sun, lamp, X-ray tube, fluorescence excitation laser) emits photons towards the scene.
2. **Interaction with matter.** Photons reflect off surfaces, scatter inside tissue, or pass through transparent media. The spectral composition and intensity of the outgoing light encode information about the scene.
3. **Optics.** The camera's aperture and lens collect a subset of the outgoing photons and focus them onto the sensor plane. The optics determine the field-of-view, depth-of-field, and geometric distortions.
4. **Sensor.** A detector (CCD or CMOS photo-diode array) converts photon energy into electrical charge at each pixel site.
5. **Digitisation.** An analog-to-digital converter (ADC) maps the continuous voltage at each pixel site to a discrete integer value. The resulting 2D array of integers is the digital image.

This lecture focuses on step 3 (the geometric projection model) and on how colour—a property of light's spectral content—is represented once the image reaches software.

2 Camera Obscura and Historical Context

The **camera obscura** (Latin: "dark chamber") is the simplest imaging device and the direct ancestor of every photographic camera.

Principle. Light from an illuminated outdoor scene passes through a tiny hole in the wall of a darkened room. Because each ray travels in a straight line, an **inverted** image of the scene forms on the opposite wall.

History. The phenomenon was described by the Chinese philosopher Mozi (c. 400 BC) and systematically studied by Ibn al-Haytham (c. 1000 AD) in *Kitāb al-Manāẓir* (Book of Optics). Renaissance artists exploited the camera obscura for realistic drawing. Adding a lens to the aperture increases brightness (see Section 6), and adding a chemical or electronic sensor yields a camera.

The camera obscura motivates the *pinhole camera model*, the starting point for all projection geometry in computer vision.

3 The Pinhole Camera Model

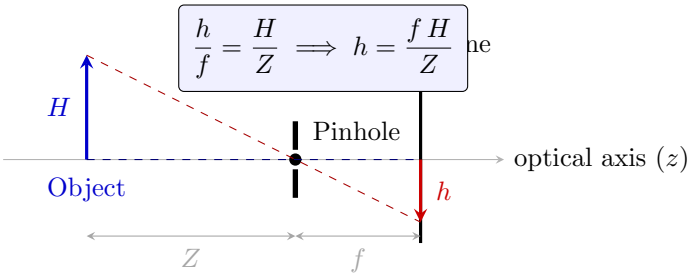
3.1 Core Geometry

Pinhole Camera Model

Place the camera centre (pinhole) at the origin. An image plane lies at distance f (the *focal length*) along the positive z-axis. A world point $\mathbf{P} = (X, Y, Z)^\top$ with $Z > 0$ projects to image coordinates:

$$\boxed{x = f\,\frac{X}{Z}, \qquad y = f\,\frac{Y}{Z}} \tag{1}$$

This is the **virtual image plane** convention (image in front of the pinhole), which produces an upright image.



3.2 Key Observations

- **Non-linearity.** The mapping $(X, Y, Z) \mapsto (fX/Z,\ fY/Z)$ involves division by Z. This prevents us from writing it directly as a matrix–vector product in Euclidean coordinates.
- **Depth-dependent magnification.** Image size $h = fH/Z$: objects farther away (Z large) appear smaller.
- **Loss of depth.** All 3D points along a ray through the pinhole project to the *same* 2D point. Depth information is irrecoverably lost in a single view.
- **Inversion.** On the *real* image plane (behind the pinhole at $z = -f$), the image is inverted in both x and y (equivalent to a 180° rotation about the optical axis). The *virtual* image plane convention ($z = +f$) eliminates this negation—it is the standard in all computer-vision frameworks (OpenCV, PyTorch3D, etc.).

3.3 Worked Example

Example: Pinhole Projection

A building of height $H = 30\,\text{m}$ stands at distance $Z = 100\,\text{m}$ from a pinhole camera with $f = 50\,\text{mm} = 0.05\,\text{m}$.
Image height: $h = fH/Z = 0.05 \times 30/100 = 0.015\,\text{m} = 15\,\text{mm}$.
If the sensor has pixel pitch $5\,\mu\text{m}$, the building spans $15\,\text{mm}/5\,\mu\text{m} = 3000$ pixels.

4 Homogeneous Coordinates and the Camera Matrix

4.1 Why Homogeneous Coordinates?

The problem. The projection (1) is non-linear in (X, Y, Z) because of the $1/Z$ division. We cannot write it as $\mathbf{p} = M\mathbf{P}$ for any fixed matrix M.

The insight. If we represent both input and output in *homogeneous coordinates* and defer the final division, the projection becomes a **linear** map (matrix multiplication). The non-linearity is absorbed into a single, universal normalisation step.

Homogeneous Coordinates

- A 3D Euclidean point (X, Y, Z) is *lifted* to the 4-vector $\widetilde{\mathbf{P}} = (X, Y, Z, 1)^\top \in \mathbb{P}^3$.
- A 2D image point (x, y) is represented as $\widetilde{\mathbf{p}} = (x, y, 1)^\top$, or equivalently any non-zero scalar multiple $(kx, ky, k)^\top$.
- To **recover Euclidean coordinates**, divide by the last component:

$$(u, v, w)^\top \longrightarrow \left(\frac{u}{w},\ \frac{v}{w}\right).$$

4.2 Projection as Matrix Multiplication

Choose $\lambda = Z$. Then:

$$Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}_{\text{simple}}} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\Pi_0} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2}$$

The entire right-hand side is a **matrix–vector product**—no division appears. The non-linearity is isolated in the final normalisation of the homogeneous output $(fX, fY, Z)^\top$ to the Euclidean point $(fX/Z,\ fY/Z)$.

The pipeline in three steps:

4.3 The Full Camera Matrix

Camera Projection Equation

$$\boxed{\lambda\,\widetilde{\mathbf{p}} = \mathbf{P}\,\widetilde{\mathbf{X}} = \mathbf{K}\,[\mathbf{R} \mid \mathbf{t}]\,\widetilde{\mathbf{X}}} \tag{3}$$

where:

- $\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ is the 3×3 **intrinsic matrix**: focal lengths f_x, f_y (in pixels), principal point (c_x, c_y), skew s (usually 0).
- $[\mathbf{R} \mid \mathbf{t}]$ is the 3×4 **extrinsic matrix**: rotation $\mathbf{R} \in \mathrm{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ mapping world to camera coordinates.
- $\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$ is the 3×4 **camera matrix** (or projection matrix), with 11 degrees of freedom (a 3×4 matrix up to scale).

4.4 Intrinsic vs. Extrinsic Parameters

Intrinsics K (camera properties)	**Extrinsics** $[\mathbf{R} \mid \mathbf{t}]$ (camera pose)
Focal length(s) f_x, f_y	Rotation $\mathbf{R}$ (3 angles: pitch, yaw, roll)
Principal point (c_x, c_y)	Translation $\mathbf{t} = (t_x, t_y, t_z)^\top$
Pixel aspect ratio f_x/f_y	6 degrees of freedom total
Skew s (usually 0)	
Fixed for a given camera/lens	**Change** with every new camera placement
Determined by *camera calibration* (checkerboard)	Determined by *pose estimation* (PnP, SLAM, SfM)

Practical Importance

Surgical navigation: Calibrate the endoscope ($\mathbf{K}$), then track its pose ($[\mathbf{R} \mid \mathbf{t}]$) in real-time to overlay surgical plans on the live video.
3D reconstruction: Both intrinsics and extrinsics are needed to triangulate 3D points from

2D correspondences in multiple views.

4.5 Why Homogeneous Coordinates Matter Algorithmically

- **Composition.** Multiple transforms (rotation, translation, intrinsics) are composed by matrix multiplication *before* being applied to any point—cheap and exact.
- **Batch efficiency.** A cloud of N points is projected by a single matrix product (a GPU-friendly operation).
- **Calibration.** Because the projection is linear in the entries of $\mathbf{P}$, camera calibration from 2D–3D correspondences reduces to a **linear least-squares** problem (Direct Linear Transform, DLT).
- **Projective geometry.** Many geometric relationships (collinearity, cross-ratio, homographies) have elegant linear forms in homogeneous coordinates that would require cumbersome case splits in Euclidean coordinates.

5 Perspective Geometry and Vanishing Points

5.1 Why Parallel Lines Converge

Under perspective projection, **3D parallel lines appear to converge** in the image. This is the most visible consequence of the $1/Z$ non-linearity.

Mathematical derivation. Consider two parallel rails at positions $(d_L, 0, t)$ and $(d_R, 0, t)$ for $t \in [0, \infty)$:

$$x_L = f\,\frac{d_L}{t}, \qquad x_R = f\,\frac{d_R}{t}.$$

As $t \to \infty$, both $x_L \to 0$ and $x_R \to 0$. The two distinct 3D lines converge to the **same image point**: the **vanishing point**.

Vanishing Point

Lines parallel in 3D along direction $\mathbf{d} = (d_x, d_y, d_z)^\top$ converge in the image to:

$$\boxed{\mathbf{v} = \mathbf{K} \begin{pmatrix} d_x/d_z \\ d_y/d_z \\ 1 \end{pmatrix}} \tag{4}$$

The vanishing point depends only on the **direction** of the lines and the camera intrinsics, *not* on the lines' 3D positions.

5.2 Types of Perspective

- **One-point perspective.** One set of receding parallel lines → one vanishing point (e.g., looking straight down a road).
- **Two-point perspective.** Two sets of parallels → two VPs on the horizon (e.g., a building corner viewed at an angle).
- **Three-point perspective.** Three mutually orthogonal sets → three VPs (e.g., looking up at a skyscraper—the vertical edges also converge).

The number of vanishing points visible in an image depends on how many distinct sets of parallel lines are present *and* on the camera orientation relative to those directions.

5.3 The Horizon Line

Horizon Line

The **horizon line** is the locus of *all* vanishing points for horizontal 3D directions. Key properties:

- When the camera is level, the horizon passes through the principal point (c_x, c_y).
- Tilting the camera up shifts the horizon *down* in the image (and vice versa).
- The horizon's position encodes the camera's height and tilt angle.

5.4 Applications in Medical Imaging

Perspective geometry governs every optical modality:

- **Endoscopes:** Full perspective projection; understanding it is essential for 3D reconstruction from endoscopic video.
- **Optical microscopes:** At very small working depths relative to magnification, perspective effects are negligible → the **orthographic (parallel) projection** approximation is valid.
- **X-ray / CT:** Fan-beam or cone-beam geometry—a different projection model (not perspective but related).

5.5 Worked Example: Vanishing Point Computation

Example

Camera intrinsics:

$$\mathbf{K} = \begin{pmatrix} 500 & 0 & 320 \\ 0 & 500 & 240 \\ 0 & 0 & 1 \end{pmatrix}$$

. Lines are parallel along $\mathbf{d} = (1, 0, 2)^\top$.
Vanishing point:

$$\mathbf{v} = \begin{pmatrix} 500 & 0 & 320 \\ 0 & 500 & 240 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/2 \\ 0/2 \\ 1 \end{pmatrix} = \begin{pmatrix} 500 \cdot 0.5 + 320 \\ 500 \cdot 0 + 240 \\ 1 \end{pmatrix} = \begin{pmatrix} 570 \\ 240 \\ 1 \end{pmatrix}.$$

So the VP is at pixel $(570, 240)$—to the right of the image centre, on the horizon.

Exam Tip — Vanishing Points

Be able to: (a) explain *why* parallel lines converge (cite the $1/Z$ factor); (b) compute a VP given $\mathbf{K}$ and $\mathbf{d}$; (c) distinguish one-, two-, and three-point perspective; (d) state what the horizon line is.

6 Lenses, Sensors, and the Imaging Pipeline

6.1 From Pinhole to Lens

A pinhole transmits too little light. Lenses focus light from each scene point onto a single sensor point, allowing a larger aperture.

Thin Lens Equation

$$\boxed{\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i}} \tag{5}$$

f = focal length, d_o = object distance, d_i = image (sensor) distance.

6.2 Key Trade-offs Introduced by Lenses

- **Depth-of-field (DOF).** Only one depth is perfectly in focus. Out-of-focus points blur into a *circle of confusion*. A large aperture (small f-number) yields a **shallow** DOF; a small aperture yields a **deep** DOF but admits less light.
- **Aperture and f-number.** $N = f/D$ where D is the aperture diameter. $N = 1.4$ is "fast" (lots of light, shallow DOF); $N = 16$ is "slow" (less light, deep DOF).
- **Lens aberrations:**
  - *Spherical:* Rays through the lens edge focus at a different point than rays through the centre → edge blur.
  - *Chromatic:* Different wavelengths refract differently → colour fringing.
  - *Vignetting:* Light fall-off towards image corners.
  - *Distortion:* Barrel or pincushion bending of straight lines.

6.3 Digital Sensors: CCD and CMOS

Both CCD and CMOS sensors convert photons into electrons at each pixel site. The charge is then read out and digitised. Key differences:

	CCD	**CMOS**
Readout	Sequential (shift register)	Per-pixel amplifier
Speed	Slower	Faster (row-parallel)
Noise	Lower (uniform readout)	Higher (fixed-pattern noise)
Power	Higher	Lower
Rolling shutter	No (global shutter)	Often yes

6.4 Bayer Pattern and Demosaicing

Bayer Colour Filter Array

Each pixel records **one** colour channel through a colour filter. The standard **Bayer RGGB** pattern tiles 2×2 blocks: 50% green, 25% red, 25% blue—mirroring human cone sensitivity.

Demosaicing is the reconstruction of full RGB at every pixel site:

- **Bilinear:** Average neighbouring same-colour pixels. Fast, but may produce colour fringing at sharp edges.
- **Edge-directed:** Interpolate along detected edge directions rather than across them. Higher quality, higher cost.

7 Human Vision and Colour Perception

7.1 Photoreceptors: Rods and Cones

Photoreceptors

Rods ∼120 million per eye. Extremely light-sensitive; responsible for **scotopic** (night) vision. No colour discrimination. Peak sensitivity ∼507 nm.

Cones ∼6 million per eye. Require bright light; responsible for **photopic** (day) vision and colour. Three types:

- **L-cones** (Long λ): peak ∼560 nm (red/yellow), ∼64% of cones.
- **M-cones** (Medium λ): peak ∼530 nm (green), ∼32% of cones.
- **S-cones** (Short λ): peak ∼420 nm (blue/violet), ∼2% of cones.

7.2 Trichromatic Theory and Metamerism

Young–Helmholtz trichromatic theory. The brain compares *ratios* of L, M, S cone activations to perceive colour. Any perceived colour is fully determined by a triple (l, m, s) of cone responses.

Metamerism

Two physically different light spectra $S_1(\lambda) \neq S_2(\lambda)$ that produce **identical** L, M, S cone responses are called **metamers**:

$$\int S_1(\lambda)\,\ell(\lambda)\,d\lambda = \int S_2(\lambda)\,\ell(\lambda)\,d\lambda, \quad \text{(and similarly for } m, s\text{)}.$$

Metamerism is why **RGB displays work**: they need only match cone responses, not reproduce the full physical spectrum.

7.3 Colour Blindness

Missing or defective cones cause colour vision deficiencies (X-linked, affecting ∼8% of males):

Type	Missing cones	Prevalence (males)
Protanopia	L-cones (red-blind)	∼1%
Deuteranopia	M-cones (green-blind)	∼1%
Tritanopia	S-cones (blue-blind)	Very rare

Design implication: In medical imaging, avoid red-only or red-green-only colour maps. Use colourblind-safe palettes (viridis, Okabe–Ito).

8 Colour Spaces

8.1 RGB: The Additive Colour Model

RGB

A 3D cube with axes Red, Green, Blue:

- 8-bit: each channel $\in [0, 255]$; float: $\in [0, 1]$.
- Black $(0, 0, 0)$; White $(255, 255, 255)$; Red $(255, 0, 0)$.
- Additive mixing: $R+G$ = Yellow, $G+B$ = Cyan, $R+B$ = Magenta, $R+G+B$ = White.
- Grayscale conversion (Rec. 601): $L = 0.299R + 0.587G + 0.114B$.

Properties and limitations:

- **Device-dependent.** Same RGB triple looks different on different displays due to differing primaries and white points.
- **Not perceptually uniform.** Equal ΔRGB does *not* correspond to equal perceived colour difference.
- **High channel correlation** in natural images $\rightarrow$ redundant for compression.

When to use: deep learning (CNNs), image I/O, display rendering, PNG/JPEG storage.

Exam Tip — BGR vs. RGB

OpenCV loads images in **BGR** channel order. Always convert with `cv2.cvtColor(img, cv2.COLOR_BGR2RGB)` before displaying with matplotlib or using libraries expecting RGB.

8.2 HSV / HSI: Separating Colour from Brightness

HSV Colour Space

Hue H Colour type; angle on colour wheel, 0°–360°. Red $\approx 0°$, Green $\approx 120°$, Blue $\approx 240°$.
Saturation S Purity: 0 (grey) to 1 (pure colour). $S = (V - \min(R, G, B))/V$.
Value V Brightness: 0 (black) to 1 (max brightness). $V = \max(R, G, B)$.
Geometry: a cylinder—hue is the angle, saturation is the radius, value is the height.

Advantages:

- **Hue is stable** under moderate illumination changes (V changes but H often does not).
- **Intuitive colour queries:** "find all red objects" $\rightarrow$ threshold H near 0°.
- Cleanly separates chromatic from achromatic information.

Caveats:

- HSV is *not* perceptually uniform.
- Hue is undefined when $S = 0$ (achromatic).
- Hue wraps: red straddles 0° and 360° (in OpenCV: $H \in [0, 180]$, so red is near 0 *and* near 170–180).

When to use: colour-based segmentation (skin, tissue, dyes), object tracking, colour correction.

Case Study: Tissue Detection in Surgery

In endoscopic video, tissue hue is relatively stable ($H \approx 0°$–$25°$ in OpenCV's 0–180 range). Saturation thresholds remove white drapes and dark shadows. Typical thresholds: $H \in [0, 20]$, $S \in [30, 255]$, $V \in [80, 255]$.

8.3 CIE LAB: Perceptual Uniformity

CIE LAB

L^* **(Lightness)** 0 (black) to 100 (white); non-linear mapping matching human brightness perception.
a^* Green ($-$) to Red ($+$) opponent axis; roughly -128 to $+127$.
b^* Blue ($-$) to Yellow ($+$) opponent axis; roughly -128 to $+127$.

Perceptual Colour Difference ΔE^*

$$\boxed{\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}} \tag{6}$$

ΔE^*	Perception
< 1	Imperceptible
≈ 2	Just noticeable
> 10	Obviously different

Equal Euclidean distances in LAB $\approx$ equal perceived colour differences.

Conversion chain:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{M} \begin{pmatrix} R_{\text{lin}} \\ G_{\text{lin}} \\ B_{\text{lin}} \end{pmatrix} \quad (\mathbf{M} \text{ depends on primaries \& white point; } R_{\text{lin}} \text{ is gamma-decoded}) \tag{7}$$

$$L^* = 116\, f\left(\frac{Y}{Y_n}\right) - 16, \quad a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right], \quad b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \tag{8}$$

where $f(t) = t^{1/3}$ for $t > (6/29)^3$ and is linear otherwise. (X_n, Y_n, Z_n) is the reference white point.

Device-independent: Based on CIE XYZ tristimulus values modelling the physical human visual response, not any particular display.

When to use: measuring colour differences, stain normalisation in digital pathology, colour segmentation robust to illumination, display/print calibration.

8.4 CIE Chromaticity and Colour Gamuts

The **CIE xy chromaticity diagram** shows all colours visible to the human eye as a horseshoe-shaped region. Within it, colour spaces (sRGB, Adobe RGB, Rec. 2020) occupy triangular subregions called **gamuts**—the range of colours a device can reproduce.

Key facts:

- sRGB covers only ~35% of visible colours.
- Medical monitors may use wider gamuts (Display P3, Rec. 2020).
- A colour outside a device's gamut must be *gamut-mapped* (clipped or compressed) for display.

8.5 CMYK and YCbCr

Space	Model	Key Properties
CMYK	Subtractive: Cyan, Magenta, Yellow, Key (Black). Inks *absorb* light.	Smaller gamut than RGB. K added because impure CMY inks yield brown, not black. Used in printing.
YCbCr	Y=luminance, C_b=blue-diff. chroma, C_r=red-diff. chroma.	Exploits lower human sensitivity to chroma detail. Chroma subsampling (4:2:0) saves ~50% bandwidth. Used in JPEG, H.264/H.265.

Chroma subsampling: 4:4:4 (full), 4:2:2 (halve horizontal chroma), 4:2:0 (halve both dims—standard in JPEG and video).

8.6 Gamma Correction

Raw sensor output is **linear** w.r.t. photon count. Human vision is **non-linear** (Weber–Fechner law: sensitivity $\propto 1/\text{intensity}$).

Gamma Encoding

$$\boxed{V_{\text{out}} = V_{\text{in}}^{1/\gamma}, \qquad \gamma \approx 2.2 \text{ (sRGB)}.} \tag{9}$$

Gamma encoding compresses bright values and expands dark values, producing a perceptually balanced allocation of bits.

Important rule: Always **gamma-decode** before any linear signal-processing operation (averaging, denoising, compositing). Failing to do so applies operations in a non-linear space and produces incorrect results.

8.7 White Balance

Different light sources have different colour temperatures (daylight ≈6500 K, tungsten ≈3000 K). The sensor records the illuminant colour cast; **white balance** corrects it by scaling R, G, B so that a white object appears white.

Common method: the **grey-world assumption**—the average scene colour should be neutral grey, so scale each channel so its mean equals the global mean.

Critical in pathology: stain colours must be reproducible across different scanners and sites; white-balance errors propagate into stain-normalisation algorithms.

9 Choosing the Right Colour Space

Exam Tip — Colour-Space Selection

There is no universally "best" colour space. Be ready to justify your choice:

Space	Best for	Key property
RGB	Deep learning, I/O, display	Device-native, simple
HSV	Colour segmentation, tracking	Separates hue from brightness
LAB	Colour measurement, stain norm.	Perceptually uniform
CMYK	Printing	Subtractive model
YCbCr	Compression (JPEG, video)	Luminance/chroma separation

10 Summary of Key Equations

Concept	Equation
Pinhole projection	$x = f\,X/Z, \quad y = f\,Y/Z$
Camera matrix	$\lambda\widetilde{\mathbf{p}} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\widetilde{\mathbf{X}}$
Intrinsic matrix	$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$
Vanishing point	$\mathbf{v} = \mathbf{K}(d_x/d_z,\, d_y/d_z,\, 1)^\top$
Thin lens equation	$1/f = 1/d_o + 1/d_i$
Grayscale (Rec. 601)	$L = 0.299R + 0.587G + 0.114B$
HSV value/saturation	$V = \max(R,G,B),\ S = (V - \min)/V$
CIE LAB ΔE^*	$\sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$
Gamma encoding	$V_{\text{out}} = V_{\text{in}}^{1/\gamma},\ \gamma \approx 2.2$

11 Practice Questions

Camera Models & Projective Geometry

1. A point $\mathbf{P} = (3, -1, 5)$ is projected through a pinhole camera with $f = 100\,\text{px}$ and principal point at the origin. What are the image coordinates?
2. Why is pinhole projection non-linear, and how do homogeneous coordinates resolve this?
3. Camera: $\mathbf{K} = \begin{pmatrix} 500 & 0 & 320 \\ 0 & 500 & 240 \\ 0 & 0 & 1 \end{pmatrix}$. Find the vanishing point for direction $\mathbf{d} = (1, 0, 2)^\top$.
4. Explain the difference between intrinsic and extrinsic parameters. Which change when you move the camera?
5. What is depth-of-field, and how does the aperture size affect it?
6. Describe the Bayer pattern. Why is there twice as much green as red or blue?
7. A building of height 20 m is at distance 80 m from a pinhole camera with $f = 35\,\text{mm}$. How tall is the image on the sensor?

Colour Models

8. What is metamerism, and why does it enable RGB displays?
9. A pixel has RGB $(200, 50, 50)$. Compute V and S in HSV.
10. Why is CIE LAB preferred over RGB for measuring colour differences? What does $\Delta E^* < 1$ mean?
11. Explain gamma correction. Why must you gamma-decode before linear operations?

12. Which colour space for: (a) endoscopic tissue segmentation by colour, (b) JPEG compression, (c) stain reproducibility measurement? Justify each.
13. What is chroma subsampling 4:2:0, and why does it not noticeably degrade image quality?
14. A pathology scanner at site A and one at site B produce different RGB values for the same tissue. Name two factors that could cause this and suggest a correction strategy.

12 Recommended Reading

- Szeliski, *Computer Vision: Algorithms and Applications* (2nd ed.), Chapter 2 (Image Formation).
- Gonzalez & Woods, *Digital Image Processing*, Chapter 6 (Colour Image Processing).
- Hartley & Zisserman, *Multiple View Geometry in Computer Vision*, Chapter 6 (Camera Models) [advanced].