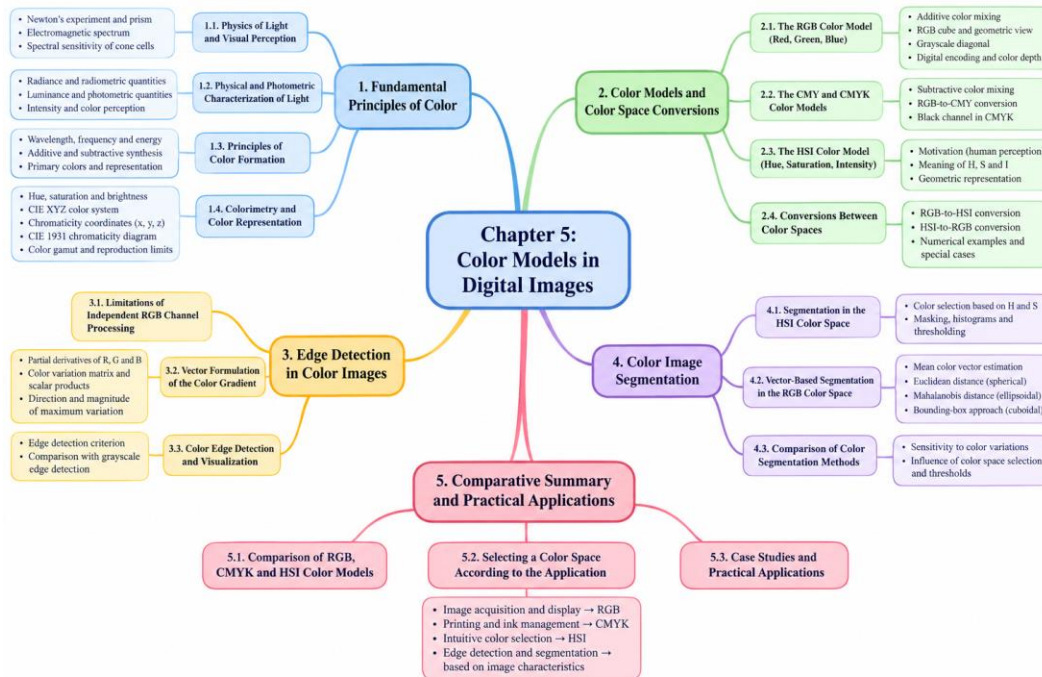


Chapter 5: Color Models in Digital Images

Dr. MAAMAR BEKHOUCHE

m.bekhouche@centre-univ-mila.dz

2026-2027



Learning Objectives

After studying this chapter, the reader should be able to explain the physical basis of color perception, distinguish radiometric from photometric quantities, describe common color models, perform basic color-space conversions, and apply color information to edge detection and image segmentation.

- Explain how visible light and the human visual system contribute to color perception.
- Interpret RGB, CMY/CMYK, HSI, and CIE XYZ representations and identify their appropriate uses.

- Convert normalized RGB values to CMY and HSI, including important special cases.
- Describe vector-based color gradients and explain why color edges may be missed by grayscale methods.
- Design simple color-segmentation strategies using thresholds, Euclidean distance, Mahalanobis distance, or bounding boxes.

Chapter Overview

Color is a multidimensional property of an image rather than a single brightness value. This chapter moves from the physics and perception of light to standardized colorimetry, digital color models, color-space conversion, and two major processing tasks: edge detection and segmentation. Equations are included as editable Microsoft Word equations. Figure placeholders identify useful illustrations to insert during final course-layout work.

1. Fundamental Principles of Color

1.1. Physics of Light and Visual Perception

1.1.1. Newton's Experiment and the Dispersion of Light Through a Prism

In the seventeenth century, Isaac Newton demonstrated that white light can be separated into a continuous spectrum by a glass prism. The prism refracts different wavelengths by different amounts, producing colors commonly described as violet, blue, green, yellow, orange, and red. A second prism or a suitable optical arrangement can recombine the dispersed components into approximately white light. The experiment established that a prism separates components already present in white light rather than creating color from nothing.

The visible spectrum is continuous; named colors are convenient regions rather than sharply separated physical bands. The perceived color of an object also depends on the illumination spectrum, the object's reflectance, and the observer.

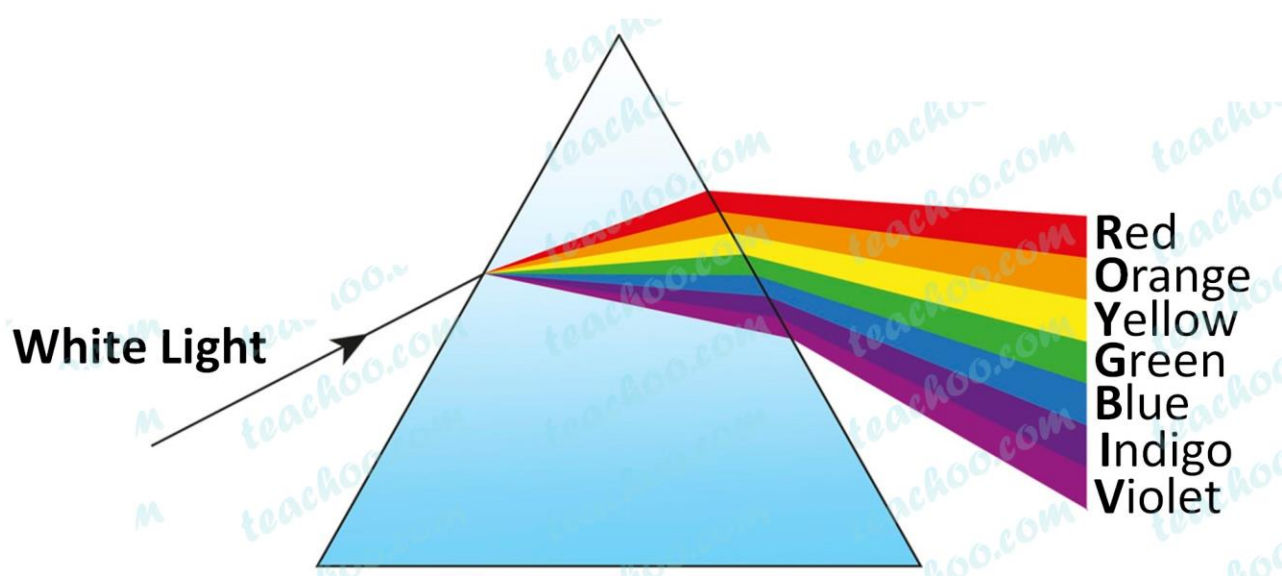


Figure 1: Dispersion of white light through a prism

1.1.2. The Electromagnetic Spectrum and the Visible Light Range

Visible light is a small portion of the electromagnetic spectrum, between ultraviolet and infrared radiation. For many introductory applications, the visible range is approximated as 380–700 nm, although quoted limits vary with observer sensitivity and experimental conditions. Wavelength is not a complete description of perceived color in every circumstance: different spectral distributions can produce the same visual response, a phenomenon known as metamerism.

Table 1: Approximate visible-spectrum regions

Color		Wavelength (nm)	Frequency (THz)	Photon energy (eV)
Infrarouge		> 780	< 405	< 1.6
rouge		~ 625-740	~ 480-405	~ 1.6 - 2.0
orange		~ 590-625	~ 510-480	~ 2.0 - 2.1
jaune		~ 565-590	~ 530-510	~ 2.1 - 2.2
vert		~ 520-565	~ 580-530	~ 2.2 - 2.4
bleu		~ 446-520	~ 690-580	~ 2.4 - 2.8
violet		~ 380-446	~ 790-690	~ 2.8 - 3.2
ultraviolet		< 380	> 790	> 3.3

These intervals are approximate and overlap perceptually. The color seen from a source is determined by the spectral power distribution and the response of the visual

system, not by a one-to-one mapping between every wavelength and a unique perceived color.

1.1.3. Spectral Sensitivity of the Human Eye's Cone Cells

The retina contains rods and cones. Rods are highly sensitive under low illumination and contribute strongly to night vision, whereas cones support color discrimination in brighter conditions. Human color vision is commonly described using three cone classes with short-, medium-, and long-wavelength sensitivity (S, M, and L). Their broad, overlapping response curves provide the biological basis for trichromatic color matching.

The three cone responses are not identical to the RGB channels used by a display. RGB is an engineering representation chosen to reproduce colors through three primaries; cone sensitivities are physiological response functions. The visual system also performs spatial and chromatic comparisons, so color appearance depends on context, adaptation, and surrounding colors.

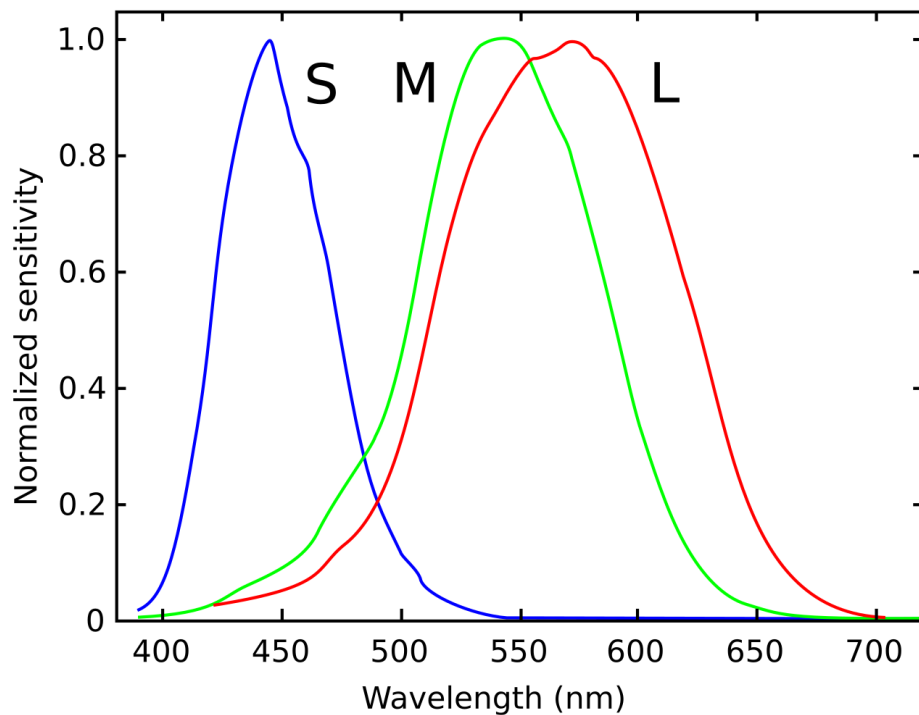


Figure 2 : Relative spectral sensitivities of S-, M-, and L-cones

1.2. Physical and Photometric Characterization of Light

1.2.1. Radiance and Radiometric Quantities

Radiometry measures optical radiation in physical energy units. Radiant energy is measured in joules, radiant flux (power) in watts, and irradiance in watts per square metre. Radiance describes power travelling from a surface per projected area and per solid angle; it is especially useful for describing light transport in imaging systems.

For a surface with spectral radiance, the wavelength-dependent quantity is often written as spectral radiance. The precise units depend on whether the spectral variable is wavelength or frequency. Radiometric quantities do not directly encode the non-uniform sensitivity of human vision.

$$L \text{ (radiance)} = \frac{d}{dA \cos \theta d\Omega} \quad (1)$$

Here, d is a differential radiant flux element, dA is surface area, θ is the angle between the surface normal and the viewing direction, and $d\Omega$ is a differential solid angle. In formal radiometry the numerator is written as differential radiant flux $d^2\Phi$; the compact expression emphasizes the projected-area and solid-angle normalization.

1.2.2. Luminance and Photometric Quantities

Photometry weights optical radiation according to the standard photopic sensitivity of human vision. Luminous flux is measured in lumens, illuminance in lux (lumens per square metre), and luminance in candelas per square metre. Luminance is therefore not interchangeable with radiance: two sources with equal radiant power can produce different perceived brightness if their spectra differ.

$$L_v = K_m \int V(\lambda) L_e(\lambda) d\lambda \quad (2)$$

In this expression, $V(\lambda)$ is the photopic luminous-efficiency function and K_m is the maximum luminous efficacy constant, conventionally 683 lm/W for monochromatic radiation near 555 nm. The equation illustrates the spectral weighting used to derive a photometric quantity from spectral radiance.

1.2.3. Intensity and Color Perception

The word intensity is used differently across fields. In image processing, it often means a scalar brightness-like value associated with a pixel; in radiometry, radiant intensity is power per solid angle. When discussing color models, the HSI component I is a

particular scalar summary of RGB components and should not be confused with a complete physical measurement of light.

1.3. Principles of Color Formation

1.3.1. Relationship Between Wavelength, Frequency, and Photon Energy

For electromagnetic radiation in vacuum, wavelength and frequency are related by the speed of light. Photon energy increases with frequency and decreases with wavelength. These relationships help explain why violet photons have greater energy than red photons, although photon energy alone does not determine perceived brightness.

$$c = \lambda \nu \quad (3)$$

$$E = h\nu = \frac{hc}{\lambda} \quad (4)$$

Here c is the speed of light in vacuum, λ is wavelength, ν is frequency, h is Planck's constant, and E is the energy of a single photon. In a material medium, propagation speed and wavelength change with refractive index, while frequency remains constant across a stationary interface.

1.3.2. Additive and Subtractive Color Synthesis

Additive synthesis combines emitted light. Red, green, and blue primaries are commonly used by displays; combining all three at suitable levels can produce white, while reducing all three to zero produces black. Subtractive synthesis combines absorptive colorants or filters. Cyan, magenta, and yellow inks ideally absorb complementary portions of incident white light; combining them reduces the reflected light.

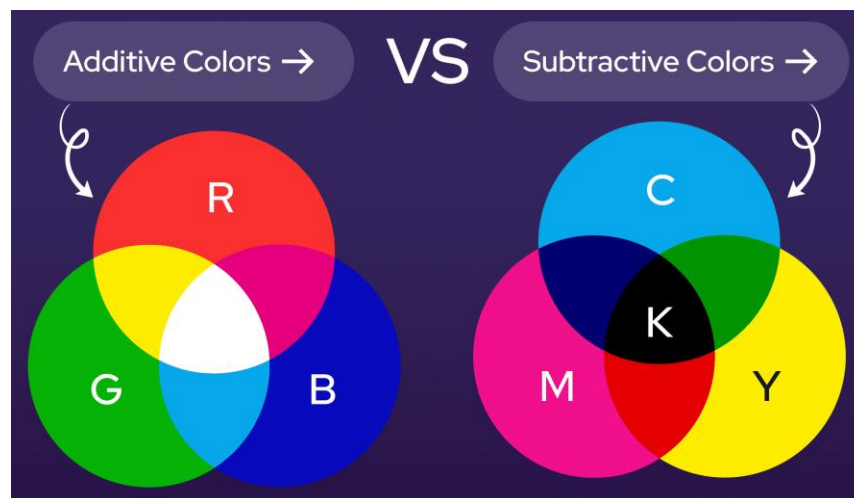


Figure 3 : Additive RGB and subtractive CMY color mixing

1.3.3. Primary Colors and Their Dependence on the Color Representation System

Primary colors are not universal physical categories. They are selected according to the purpose and gamut of a representation system. RGB primaries are suited to emitted-light devices; CMY(K) is suited to printing; and perceptual or device-independent systems such as HSI and CIE XYZ serve other purposes. No finite set of primaries reproduces every color visible to humans.

1.4. Colorimetry and Color Representation

1.4.1. Hue, Saturation, and Brightness

Hue describes the dominant chromatic family, such as red or green. Saturation describes the strength or purity of chromatic color relative to a neutral color. Brightness is a perceptual attribute related to how much light appears to be emitted or reflected; lightness is a related attribute judged relative to a reference white. Different models define their components differently, so HSI intensity is not exactly the same as perceptual brightness or CIE lightness.

1.4.2. The CIE XYZ Color System and Tristimulus Values

The International Commission on Illumination (CIE) defined the XYZ tristimulus system to represent color matches using three non-negative coordinates for ordinary visible stimuli. The color-matching functions are mathematical standards derived from color-matching experiments. Y is associated with luminance under the specified conditions, while X and Z help encode chromatic information. XYZ is device-independent in a way that raw device RGB values are not, but a complete color specification still requires a reference white and viewing conditions.

$$X = k \int S(\lambda) \bar{x}(\lambda) d\lambda \quad (5)$$

$$Y = k \int S(\lambda) \bar{y}(\lambda) d\lambda \quad (6)$$

$$Z = k \int S(\lambda) \bar{z}(\lambda) d\lambda \quad (7)$$

$S(\lambda)$ denotes the spectral stimulus, and $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ are the CIE color-matching functions. The scaling constant k depends on the chosen normalization. These integrals show that tristimulus values summarize a spectral distribution through standardized weighting functions.

1.4.3. Chromaticity Coordinates (x, y, z)

Chromaticity coordinates separate the relative proportions of XYZ from their overall magnitude. They are obtained by normalizing the tristimulus values by their sum. If $X + Y + Z$ is zero, as for an ideal zero stimulus, chromaticity is undefined.

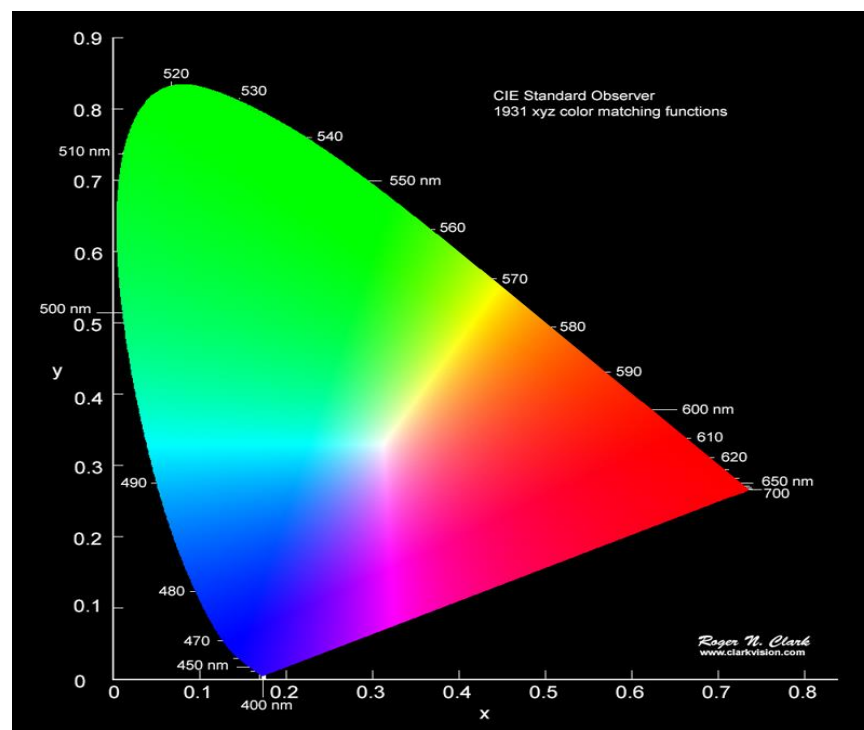
$$x = \frac{X}{X+Y+Z}, y = \frac{Y}{X+Y+Z}, z = \frac{Z}{X+Y+Z} \quad (8)$$

$$x + y + z = 1 \quad (9)$$

Because the three coordinates sum to one, only two are independent. The two-dimensional (x, y) plane is therefore commonly used for chromaticity plots.

1.4.4. The CIE 1931 Chromaticity Diagram

The CIE 1931 xy chromaticity diagram plots chromaticity independently of luminance. Its curved boundary represents monochromatic spectral colors, while the straight lower segment joins the extreme ends of the visible spectrum. A point inside the diagram can represent a mixture of spectral colors. The diagram is useful for comparing gamuts, but equal geometric distances on the xy diagram do not correspond to equal perceived color differences.



Insert Figure 4: CIE 1931 xy chromaticity diagram

1.4.5. Color Gamut and the Limits of Color Reproduction

A color gamut is the set of colors a device or color space can represent under specified conditions. A display gamut is constrained by its primaries and signal limits; a printer gamut depends on inks, paper, illumination, and printing process. Colors outside a target gamut must be mapped or clipped when converting between spaces. Gamut mapping can preserve hue, lightness, saturation, or visual relationships, but usually cannot preserve every attribute simultaneously.

Table 2: Colorimetric terms used in this chapter

Term	Meaning	Typical use
Radiance	Radiant power per projected area and solid angle	Optical systems and light transport
Luminance	Photometrically weighted light per projected area and solid angle	Display and viewing measurements
Tristimulus values	Three weighted quantities describing a color match	CIE XYZ colorimetry
Chromaticity	Relative proportions of tristimulus values	Comparing hue-related gamut coverage
Gamut	Set of colors reproducible by a system	Display, print, and conversion workflows

2. Color Models and Color Space Conversions

2.1. The RGB Color Model (Red, Green, Blue)

2.1.1. Principle of Additive Color Mixing

RGB represents a color using the red, green, and blue channel values. In a normalized representation, each component lies between 0 and 1. The triple (0, 0, 0) represents black and (1, 1, 1) represents the maximum neutral output for the chosen encoding. Equal channel values form a neutral axis, although the displayed result depends on the transfer function, white point, and calibration of the device.

$$\mathbf{C} = (\mathbf{R}, \mathbf{G}, \mathbf{B}), \mathbf{R}, \mathbf{G}, \mathbf{B} \in [0, 1] \quad (10)$$

2.1.2. The RGB Cube and Geometric Representation

The RGB cube is a convenient geometric representation in which the three axes correspond to R, G, and B. Its eight corners represent black, white, the three primaries, and the three secondary colors (cyan, magenta, yellow). A point inside the cube represents a possible encoded color under the assumed channel ranges.

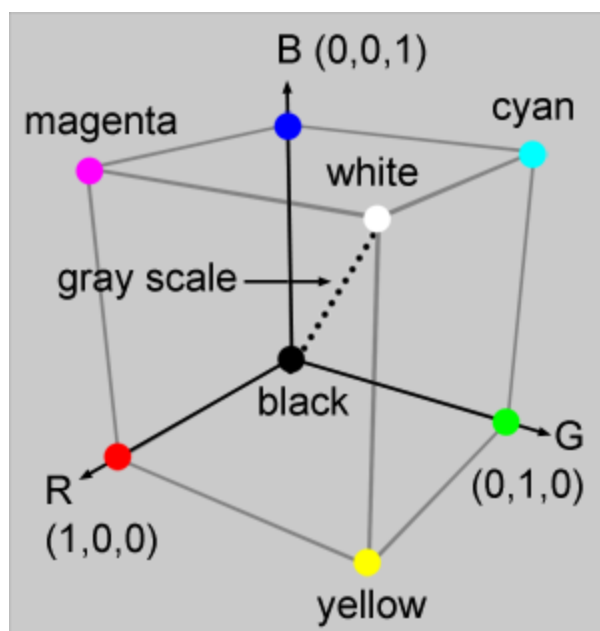


Figure 5: RGB color cube with primary and secondary vertices

2.1.3. The Grayscale Diagonal

The line from black to white through the RGB cube is the grayscale diagonal. Points on this line have equal channel values, $R = G = B$. Under a common linear-light RGB representation, equal components produce neutral chromaticity; in encoded sRGB, equal code values still produce neutral colors, but their numerical values are not proportional to emitted light because sRGB applies a nonlinear transfer function.

$$R = G = B = g \quad (11)$$

2.1.4. Digital Color Encoding and Color Depth

For an n -bit channel, there are 2^n possible code values. An RGB image with 8 bits per channel has 24 bits per pixel and can encode $2^{24} = 16,777,216$ channel combinations, before accounting for alpha or other metadata. This number is a count of code combinations, not a guarantee that all colors are distinguishable or physically reproducible. In practice, sRGB values are nonlinear encodings; many image-processing equations assume normalized linear-light values and therefore require linearization first.

$$N = 2^n \quad (\text{levels per channel}) \quad (12)$$

$$N_{\text{RGB}} = 2^{3n} \quad (13)$$

Table 3: Common digital RGB encodings

Encoding	Bits per channel	Total RGB bits per pixel	Number of RGB code combinations
----------	------------------	--------------------------	---------------------------------

Low-depth RGB	5 or 6 (varies by channel)	16 bits in common 5-6-5 format	65,536
True color	8	24	16,777,216
High precision	10	30	1,073,741,824
Floating point	Often 16 or 32-bit float	Implementation-dependent	Not usefully summarized as a simple integer code count

2.2. The CMY and CMYK Color Models

2.2.1. Principle of Subtractive Color Mixing

CMY describes cyan, magenta, and yellow colorants that subtract light from a reflecting substrate. Cyan ideally absorbs red, magenta absorbs green, and yellow absorbs blue. Real inks are imperfect, so their spectral absorption overlaps and their combinations do not behave as ideal mathematical complements.

2.2.2. RGB-to-CMY Conversion

For normalized, idealized RGB and CMY values, the basic conversion is obtained by taking the complement of each channel. This relation is a pedagogical model and is not a complete color-managed conversion from a calibrated RGB space to a printer profile.

$$C = 1 - R, \quad M = 1 - G, \quad Y = 1 - B \quad (14)$$

Example: if a normalized RGB triplet is (0.20, 0.60, 0.90), the idealized CMY triplet is (0.80, 0.40, 0.10). This calculation illustrates complementing the channel values; a real print workflow should use an ICC profile and a rendering intent.

2.2.3. Introduction of the Black Channel in CMYK

CMYK adds a black (K) channel. Black ink improves dark detail, reduces the amount of colored ink required for neutral shadows, and can improve economy and stability. A simple textbook conversion first estimates K as the minimum of C, M, and Y, then computes residual chromatic components. Production printing systems use device-specific black generation and under-color removal, so the simple equations should not replace an ICC-managed workflow.

$$K = \min(C, M, Y) \quad (15)$$

$$C' = \frac{C - K}{1 - K}, \quad M' = \frac{M - K}{1 - K}, \quad Y' = \frac{Y - K}{1 - K} \quad (16)$$

The residual equations apply when K is less than 1. For pure black (K = 1), the denominator becomes zero; handle that special case directly by setting C' = M' = Y' =

0. Some conventions store K as a separate channel and use different scaling or black-generation strategies.

Table 4: RGB, CMY, and CMYK at a glance

Model	Representation	Typical application	Important limitation
RGB	Three additive channels	Cameras, displays, web images	Device-dependent unless color-managed
CMY	Three subtractive colorants	Conceptual printing model	Ideal complement equations ignore ink behavior
CMYK	C, M, Y plus black	Printing and press workflows	Conversion depends on printer, paper, ink, and profile

2.3. The HSI Color Model (Hue, Saturation, Intensity)

2.3.1. Motivation: Adapting Color Representation to Human Perception

RGB is convenient for hardware but mixes chromatic and brightness information across three channels. HSI separates a hue-like descriptor from saturation and intensity, making it useful for color selection and some image-processing tasks. HSI is not perfectly perceptually uniform, and its numerical behavior differs from HSV, HSL, and other similarly named models. A method should specify its exact convention.

2.3.2. Meaning of Hue, Saturation, and Intensity

Hue H is an angle describing the color family; saturation S measures the departure from a neutral gray; intensity I is the arithmetic mean of the RGB components in the common HSI definition. Hue is unstable or undefined for gray pixels because saturation approaches zero.

$$I = \frac{R + G + B}{3} \quad (17)$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \quad (18)$$

For normalized RGB, S is normally in $[0, 1]$. If $R + G + B = 0$, the pixel is black and the formula is undefined; define $S = 0$ by convention. Hue is computed from channel relationships and then wrapped to the interval $[0^\circ, 360^\circ)$.

2.3.3. Geometric Representation and Interpretation

A common geometric interpretation places intensity along a vertical axis and hue around a circular axis, with saturation increasing outward from the neutral axis. The geometry is often visualized as a double cone or related solid, but the exact shape

depends on the chosen color model. The representation helps users select a hue range while controlling how vivid or bright the selected pixels should be.

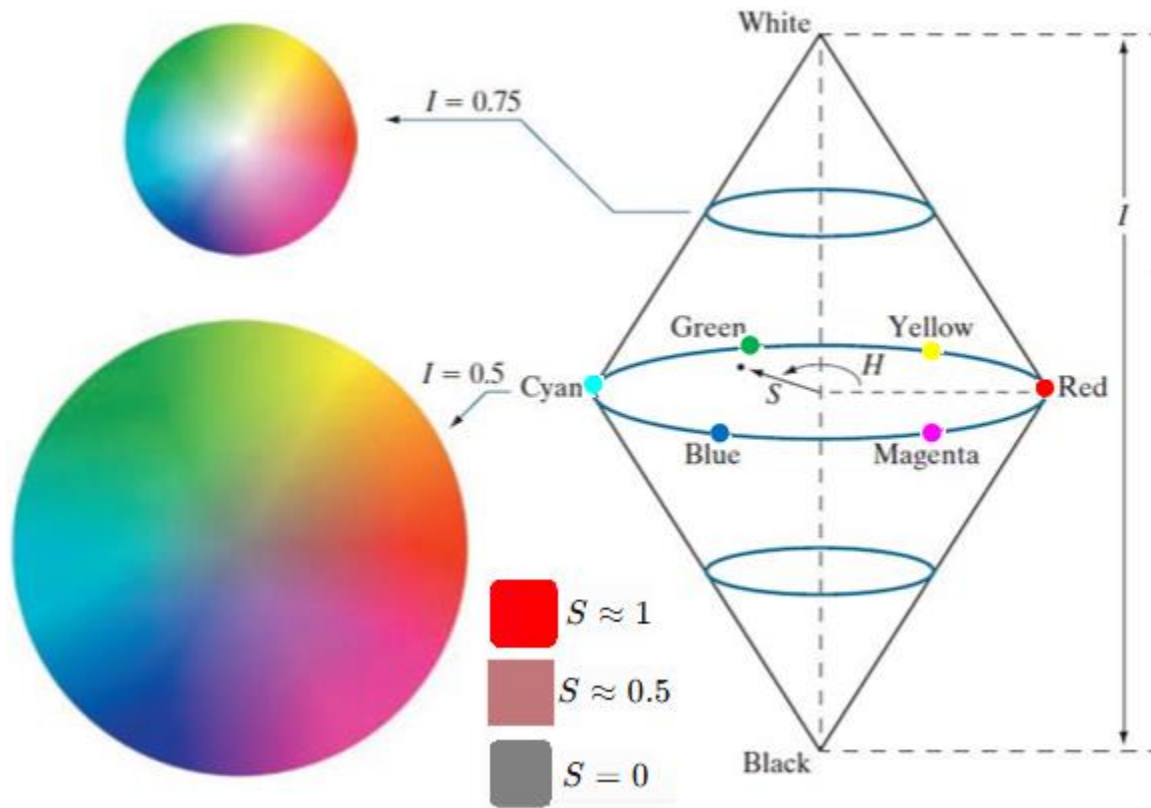


Figure 06: HSI geometric representation

2.4. Conversions Between Color Spaces

2.4.1. RGB-to-HSI Conversion

Assume normalized RGB values and the common HSI convention. Hue is calculated using an arccosine expression. The result is adjusted according to whether B is greater than G so that hue covers the full circle.

$$\theta = \cos^{-1} \left\{ \frac{1}{2} [(R - G) + (R - B)] / \sqrt{(R - G)^2 + (R - B)(G - B)} \right\} \quad (19)$$

$$H = \theta \text{ if } B \leq G; \quad H = 360^\circ - \theta \text{ if } B > G \quad (20)$$

The arccosine input should be clipped numerically to $[-1, 1]$ to prevent floating-point round-off from producing an invalid value. When $R = G = B$, the denominator and hue interpretation are degenerate; set H to a conventional value such as 0° and $S = 0$.

2.4.2. HSI-to-RGB Conversion

The inverse conversion is piecewise because hue determines which RGB sector is active. Let H be in degrees, S and I be normalized, and use the conventional HSI equations below. First reduce H modulo 360° . In each sector, compute two channels and recover the third from the intensity constraint.

$$0^\circ \leq H < 120^\circ: B = I(1 - S), R = I[1 + \frac{S \cos H}{\cos(60^\circ - H)}], G = 3I - (R + B) \quad (21)$$

$$120^\circ \leq H < 240^\circ: H' = H - 120^\circ, R = I(1 - S), G = I[1 + \frac{S \cos H'}{\cos(60^\circ - H')}], B = 3I - (R + G) \quad (22)$$

$$240^\circ \leq H < 360^\circ: H' = H - 240^\circ, G = I(1 - S), B = I[1 + \frac{S \cos H'}{\cos(60^\circ - H')}], R = 3I - (G + B) \quad (23)$$

These equations assume a particular HSI definition. Implementations should verify that the output lies in the valid range, allowing only small numerical tolerance. If the chosen convention uses radians, convert angles consistently.

2.4.3. Numerical Examples and Handling Special Cases

1. RGB to CMY conversion

for $RGB = (0.20, 0.60, 0.90)$,

the complement equations give : $CMY = (0.80, 0.40, 0.10)$.

2. RGB to HSI conversion

Suppose we have the following RGB color:

$$R = 100, \quad G = 150, \quad B = 200$$

The RGB values range from 0 to 255.

Step 1: Normalize the RGB values

Divide each component by 255:

$$r = \frac{100}{255} = 0.3922 \quad g = \frac{150}{255} = 0.5882 \quad b = \frac{200}{255} = 0.7843$$

Step 2: Calculate the Intensity I

$$I = \frac{r + g + b}{3} = \frac{0.3922 + 0.5882 + 0.7843}{3}$$

$$I = 0.5882$$

On a scale from 0 to 255, this is $I = 150$.

Step 3: Calculate the Saturation S

$$S = 1 - \frac{3}{r + g + b} \min(r, g, b)$$

Since $\min(r, g, b) = 0.3922$,

$$S = 1 - \frac{3(0.3922)}{1.7647}$$

$$S = 0.3333$$

Step 4: Calculate the Hue H

Step 4: Calculate the Hue H

First, calculate the angle θ :

$$\theta = \cos^{-1} \left(\frac{\frac{1}{2}[(r - g) + (r - b)]}{\sqrt{(r - g)^2 + (r - b)(g - b)}} \right)$$

Substituting the values:

$$\theta = \cos^{-1} \left(\frac{-0.2941}{\sqrt{0.0381 + 0.0229}} \right)$$

$$\theta \approx 130.1^\circ$$

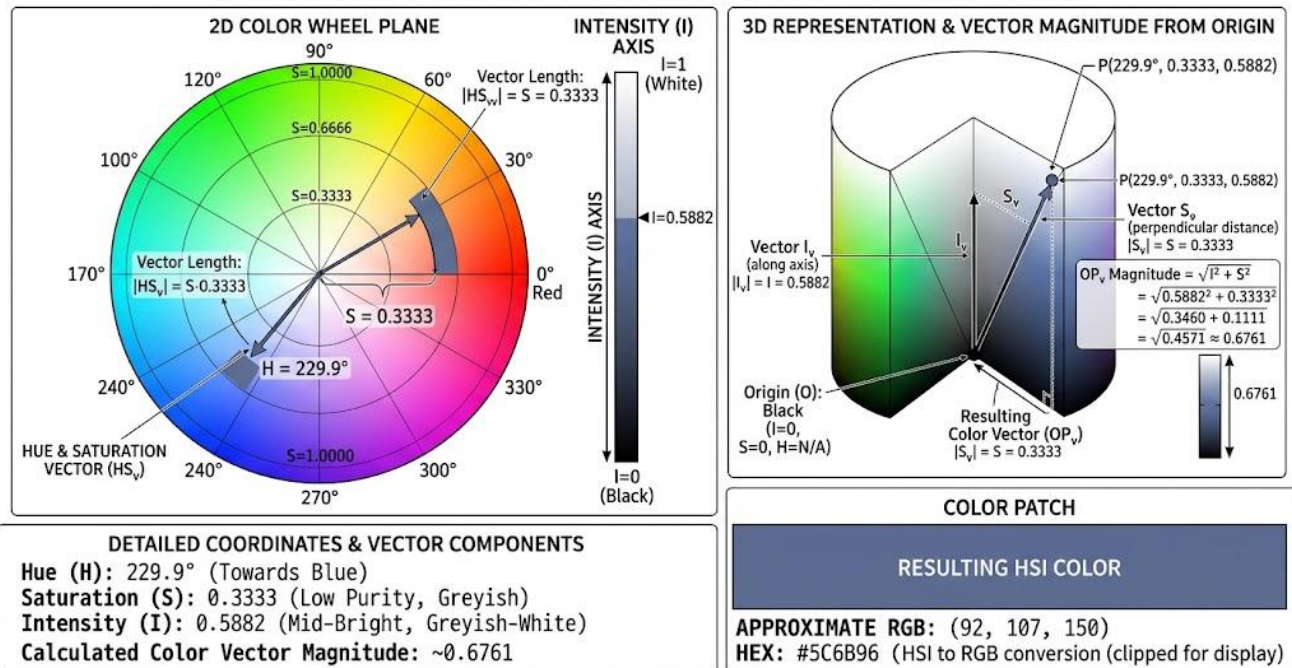
Because $b > g$, the hue is: $H = 360^\circ - \theta$

$$H \approx 229.9^\circ$$

Therefore, the final HSI representation is approximately:

$$(H, S, I) = (229.9^\circ, 0.3333, 0.5882)$$

HSI COLOR MODEL REPRESENTATION (H=229.9, S=0.3333, I=0.5882)



3. HSI to RGB conversion

Now convert the same HSI values back to RGB:

$$H = 229.9^\circ, \quad S = 0.3333, \quad I = 0.5882$$

Since $180^\circ \leq H < 300^\circ$, use the blue-sector equations.

Step 1: Calculate the blue component b

$$b = I(1 - S) = 0.5882(1 - 0.3333)$$

$$b \approx 0.3922$$

Step 2: Calculate the red component r

$$r = I \left(1 + \frac{S \cos H}{\cos(60^\circ - H)} \right)$$

Substituting:

$$r \approx 0.7843$$

Step 3: Calculate the green component g

$$g = 3I - (r + b) = 3(0.5882) - (0.7843 + 0.3922)$$

$$g \approx 0.5882$$

Step 4: Convert the normalized values to the 0–255 range

Multiply each component by 255:

$$R = 255r \approx 200$$

$$G = 255g \approx 150$$

$$B = 255b \approx 100$$

Table 5: Common color-conversion edge cases

Case	Issue	Recommended handling
Black RGB = (0,0,0)	HSI saturation formula denominator is zero	Set S = 0 and choose H = 0° by convention
Neutral gray R = G = B	Hue is not meaningful	Set S = 0; hue may be set to 0°
B > G in RGB-to-HSI	Hue is in the lower half of the circle	Use H = 360° – θ
K = 1 in CMYK complement conversion	Residual-channel denominator is zero	Set C' = M' = Y' = 0
Floating-point round-off	Invalid inverse cosine input	Clamp the input to [–1,1]

3. Edge Detection in Color Images

3.1. Limitations of Independent RGB Channel Processing

A color edge is a spatial location where the color vector changes significantly. Applying a grayscale edge detector independently to R, G, and B and then combining the results can be unreliable: each channel may respond differently, noise may create inconsistent edges, and a color transition can be strong even when its grayscale brightness remains constant. Conversely, a large change in one channel does not always correspond to a perceptually strong color difference.

A simple luminance conversion collapses a three-dimensional signal to one scalar. This can be useful for speed or when brightness boundaries are the target, but it discards chromatic information. A robust color-edge method should define how changes across channels are combined.

3.2. Vector Formulation of the Color Gradient

3.2.1. Partial Derivatives of the R, G, and B Components

Represent an image at location (x,y) by a color vector. Spatial derivatives describe how each channel changes along horizontal and vertical directions.

$$f(x,y) = [R(x,y), G(x,y), B(x,y)]^T \quad (24)$$

$$f_x = [R_x, G_x, B_x]^T, \quad f_y = [R_y, G_y, B_y]^T \quad (25)$$

The partial derivatives may be estimated with finite differences or standard gradient kernels such as Sobel. For example, a central difference estimates the derivative along x from neighboring pixels. Derivatives amplify high-frequency noise, so smoothing or robust estimation is often needed before edge strength is evaluated.

$$R_x(x,y) \approx \frac{R(x+1,y) - R(x-1,y)}{2} \quad (26)$$

3.2.2. Color Variation Matrix and Scalar Products

A useful local representation is a 2×2 color variation matrix formed from the inner products of the channel-derivative vectors. It summarizes how much color changes along each spatial direction and how those changes are coupled.

$$J = [f_x^T f_x \quad f_x^T f_y; f_y^T f_x \quad f_y^T f_y] \quad (27)$$

The matrix J is symmetric and positive semidefinite. Its eigenvalues indicate the strength of color variation along orthogonal directions. The largest eigenvalue corresponds to the direction of maximum local variation, while the associated eigenvector gives that spatial direction. This is related to the structure-tensor formulation used in corner and orientation analysis.

$$\lambda_{\max} = \frac{J_1}{2} + \sqrt{\frac{J_1^2}{4} - J_{12}^2} \quad (28)$$

Here J_1 and J_2 denote the diagonal entries and J_{12} the off-diagonal entry. In practice, derivative products are often averaged over a neighborhood to reduce noise and make the estimate more stable.

3.2.3. Direction and Magnitude of Maximum Color Variation

The principal eigenvector of J gives the direction in the image plane where the color vector changes most rapidly. The square root of the largest eigenvalue can be used as a scalar edge-strength measure. A threshold then distinguishes candidate edges from

weaker variations; non-maximum suppression and connectivity-based hysteresis can further refine the result.

$$E(x, y) = \sqrt{\lambda} \quad (29)$$

3.3. Color Edge Detection and Visualization

3.3.1. Edge Detection Criterion

A practical color-edge pipeline typically includes optional smoothing, derivative estimation for all channels, calculation of a combined edge-strength measure, non-maximum suppression, and thresholding. Thresholds should be selected with respect to image noise, contrast, and the intended application rather than copied blindly from grayscale settings.

- Smoothing reduces small-scale noise but may blur narrow boundaries.
- The color-gradient measure combines information from all channels.
- Non-maximum suppression narrows broad responses to thin edge traces.
- Single or dual thresholds can distinguish strong edges from weak connected edges.

3.3.2. Comparison with Grayscale Edge Detection

Table 6: Grayscale and color edge detection

Aspect	Grayscale edge detection	Color edge detection
Input	One intensity or luminance channel	Multiple channels or a color vector
Chromatic boundaries	May miss boundaries with similar luminance	Can preserve boundaries caused by color change
Noise	Noise affects the scalar signal	Channel noise and inter-channel differences must be managed
Complexity	Usually lower	Higher due to vector or multi-channel computation
Best use	Brightness-defined boundaries	Boundaries defined by brightness and/or chromatic change

Neither method is universally superior. Grayscale methods are appropriate when the target boundary is primarily a luminance change or computational simplicity is important. Color methods are useful when region identity depends on chromatic differences. Evaluation should match the task, and a color edge map should not be interpreted as a semantic boundary detector.

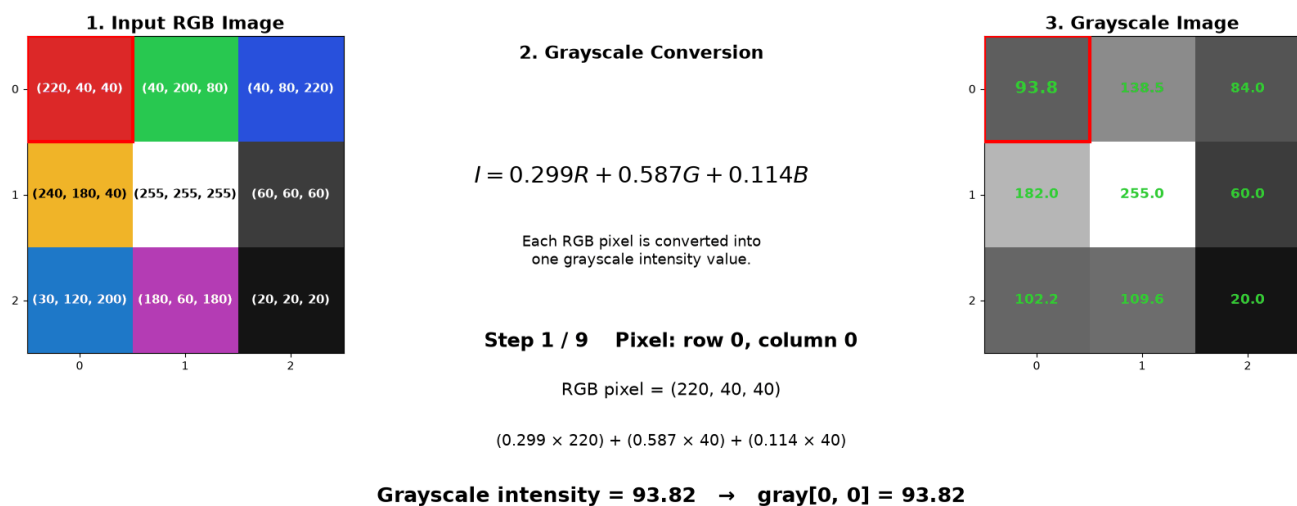


Figure 07: Comparison of grayscale and color edge maps

4. Color Image Segmentation

Image segmentation partitions an image into regions that share a property of interest. Color segmentation uses chromatic information to separate objects or materials from their surroundings. It is often effective when object colors are distinctive, but it can fail under changing illumination, shadows, reflections, camera white-balance changes, or overlapping object and background colors.

4.1. Segmentation in the HSI Color Space

4.1.1. Color Selection Based on Hue and Saturation

HSI can make color selection more intuitive because hue represents a color family and saturation indicates chromatic strength. A basic mask selects pixels whose hue lies within a target interval and whose saturation exceeds a minimum. Hue is circular: a red interval may cross the $0^\circ/360^\circ$ boundary and therefore require two intervals.

$$M(x,y) = 1 \text{ if } H_{\min} \leq H(x,y) \leq H_{\max} \text{ and } S(x,y) \geq S_{\min}; \text{ otherwise } 0 \quad (30)$$

Intensity may be included when the target is expected to lie within a brightness range. However, intensity thresholds can reject valid pixels under shadows or accept unrelated objects under similar lighting. Hue should generally be ignored or down-weighted for nearly neutral pixels because hue is unstable when saturation is very low.

4.1.2. Masking, Histograms, and Thresholding

A color histogram summarizes how frequently channel or color-space values occur. Histograms can guide threshold selection, especially when the target occupies a

distinguishable range. A binary mask can be refined with morphological operations such as opening to remove isolated noise and closing to fill small gaps. These operations should be selected according to object scale and shape; they do not replace a well-defined color model.

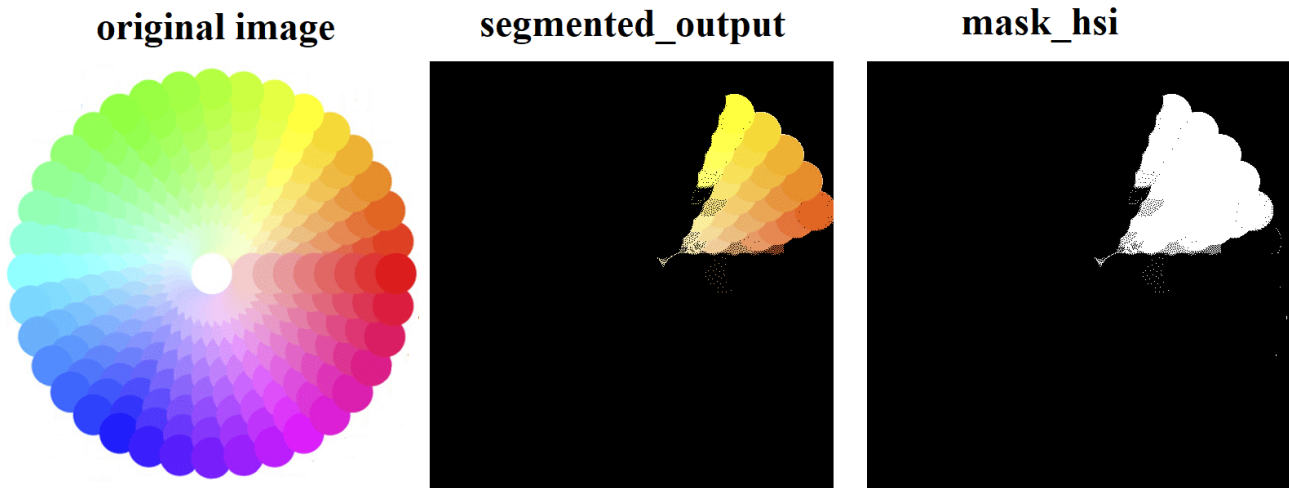


Figure 08: HSI thresholding and binary mask workflow

Exemple numérique : Segmentation dans l'espace colorimétrique HSI

Prenons une image contenant des pommes rouges et vertes. Nous voulons extraire les pixels rouges en utilisant le modèle HSI (Hue, Saturation, Intensity).

L'avantage du modèle HSI est qu'il permet de sélectionner une couleur principalement à partir de sa teinte H, tout en utilisant la saturation (S) pour éviter les pixels peu colorés.

1. Principe de la segmentation

On définit les conditions suivantes pour sélectionner les pixels rouges :

$$H \in [0^\circ, 20^\circ] \cup [340^\circ, 360^\circ]$$
$$S \geq 0.4, \quad I \geq 0.2$$

- H : teinte, qui permet de reconnaître la famille de couleurs.
- S : saturation, qui permet de distinguer les couleurs vives des gris.
- I : intensité, qui permet d'exclure les pixels trop sombres.

Un pixel est considéré comme rouge s'il satisfait toutes ces conditions.

2. Exemple avec quatre pixels

Supposons que nous ayons les valeurs HSI suivantes :

Pixel	H	S	I	Décision
P_1	10°	0,80	0,60	Rouge sélectionné
P_2	180°	0,75	0,50	Rejeté : teinte verte/cyan
P_3	5°	0,20	0,70	Rejeté : saturation faible
P_4	350°	0,90	0,10	Rejeté : intensité faible

Pour P_1 :

- 10° appartient à l'intervalle rouge.
- $0,80 \geq 0,4$.
- $0,60 \geq 0,2$.

Donc P_1 est sélectionné.

Pour P_4 , la teinte et la saturation conviennent, mais $I = 0,10 < 0,2$. Le pixel est donc rejeté.

Code Python complet

Ce programme utilise une image `HSI.png` placée dans le dossier `img_05`. OpenCV convertit d'abord l'image RGB en HSI-like via HSV : pour illustrer les seuils HSI classiques, on utilise ici les canaux Hue, Saturation et Value, où Value joue le rôle d'une mesure de luminosité. Ce n'est pas une conversion mathématiquement identique à HSI, mais une approximation pratique pour la segmentation.

```

import cv2
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path

# 1. Load the original image
image_path = Path("img_05") / "HSI.png"
img_bgr = cv2.imread(str(image_path))

if img_bgr is None:
    raise FileNotFoundError(
        f"Image not found: {image_path.resolve()}"
    )

# Convert BGR to RGB for display
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# 2. Convert BGR to HSV
# OpenCV uses H: 0-179, S: 0-255, V: 0-255
hsv = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2HSV)

# 3. Define red color ranges
# Red wraps around the hue boundary, so use two intervals
lower_red_1 = np.array([0, 102, 51])
upper_red_1 = np.array([10, 255, 255])

lower_red_2 = np.array([170, 102, 51])
upper_red_2 = np.array([179, 255, 255])

# 4. Create binary masks
mask_1 = cv2.inRange(hsv, lower_red_1, upper_red_1)
mask_2 = cv2.inRange(hsv, lower_red_2, upper_red_2)

red_mask = cv2.bitwise_or(mask_1, mask_2)

# 5. Segment the red regions
segmented = cv2.bitwise_and(
    img_rgb, img_rgb, mask=red_mask
)

# 6. Display the results
plt.figure(figsize=(12, 5))

plt.subplot(1, 3, 1)
plt.imshow(img_rgb)
plt.title("Original Image")
plt.axis("off")

plt.subplot(1, 3, 2)
plt.imshow(red_mask, cmap="gray")
plt.title("Red Color Mask")
plt.axis("off")

plt.subplot(1, 3, 3)
plt.imshow(segmented)
plt.title("Red Color Segmentation")
plt.axis("off")

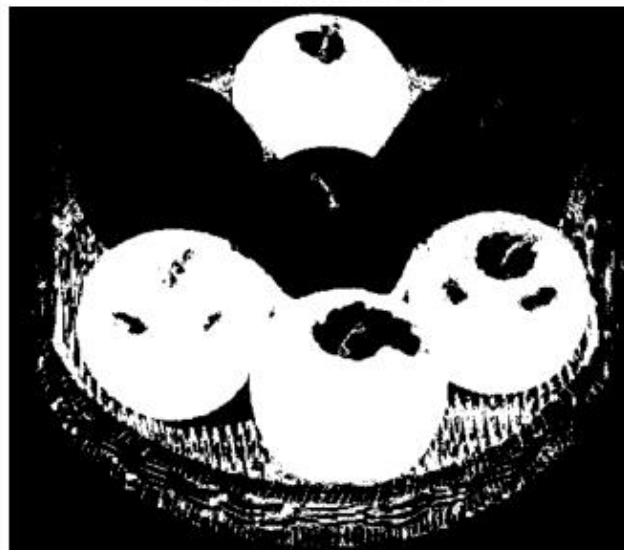
plt.tight_layout()
plt.show()

```

Original Image



Red Color Mask



Red Color Segmentation



4.2. Vector-Based Segmentation in the RGB Color Space

4.2.1. Estimation of the Mean Color Vector

Let a set of representative target pixels have RGB vectors c_i . Their mean color vector estimates the center of the target color distribution. In a simple implementation, samples may be selected manually from a region known to belong to the object.

$$\mu = \frac{1}{N} \sum_i c_i \quad (31)$$

The mean is useful when the target colors form a compact cluster. If the samples contain outliers or multiple distinct colors, a single mean may not represent the target well. In that case, robust estimates, multiple prototypes, or a learned classifier may be more appropriate.

4.2.2. Euclidean Distance and the Spherical Model

Euclidean distance measures the straight-line distance between a pixel color vector c and the target mean μ . Thresholding this distance defines a sphere in the three-dimensional color space. The method is simple and fast, but it treats variation in all channel directions equally and is sensitive to channel scaling.

$$dE(c, \mu) = \sqrt{(c - \mu)^T (c - \mu)} \quad (32)$$

$$M(c) = 1 \text{ if } dE(c, \mu) \leq \tau; \text{ otherwise } 0 \quad (33)$$

The threshold τ controls the trade-off between coverage and false positives. A small τ selects only colors close to the prototype; a large τ includes more variation but may merge background colors with the target.

4.2.3. Mahalanobis Distance and the Ellipsoidal Model

The Mahalanobis distance accounts for the covariance of target colors. Directions with high observed variance are treated as less unusual than directions with low variance. The resulting decision region is ellipsoidal rather than spherical.

$$dM(c, \mu) = \sqrt{(c - \mu)^T \Sigma^{-1} (c - \mu)} \quad (34)$$

$$\Sigma = \frac{1}{N-1} \sum_i (c_i - \mu)(c_i - \mu)^T \quad (35)$$

The covariance matrix Σ must be estimated from enough representative samples. If it is singular or poorly conditioned, regularization may be required, for example by adding a small positive value to its diagonal. Mahalanobis distance is more flexible than

Euclidean distance but depends on reliable covariance estimates and can be less intuitive to tune.

4.2.4. Bounding-Box Approach and the Cuboidal Model

A bounding-box model independently constrains each color channel. For each component, the algorithm estimates a minimum and maximum acceptable value from target samples, possibly expanded by a margin. The resulting region is a cuboid aligned with the RGB axes. It is computationally efficient but ignores correlations between channels.

$$R_{\min} \leq R \leq R_{\max}, G_{\min} \leq G \leq G_{\max}, B_{\min} \leq B \leq B_{\max} \quad (36)$$

Simple Numerical Example: Vector-Based Segmentation in RGB

In RGB color segmentation, each pixel is represented by a three-dimensional vector:

Vector-Based Segmentation in the RGB Color Space

We represent each pixel by a 3D color vector $x = [R, G, B]$ and classify it based on its similarity to a target color (or color model).
Below are simple numerical examples using four common methods.

1. Estimation of the Mean Color Vector

Idea
Compute the average color of the target region and classify pixels that are close to this mean color.

Example
Suppose the target (leaf) region has 5 sample pixels:

Pixel	R	G	B
1	52	160	48
2	60	170	55
3	55	165	50
4	48	158	45
5	58	168	52

Mean color vector:

$$\bar{x} = \frac{1}{5} \sum_{i=1}^5 x_i$$

$$\bar{x} = [54.6, 164.2, 50.0]$$

Classification rule:
If $d(x, \bar{x}) = \sqrt{(R - 54.6)^2 + (G - 164.2)^2 + (B - 50.0)^2}$
 $\leq T \rightarrow$ target (leaf)
else $\rightarrow$ background

Result (illustrative)

2. Euclidean Distance and the Spherical Model

Idea
Measure the Euclidean distance between the pixel color and the reference color (or mean color). Pixels within a certain radius are classified as target. The model assumes a spherical distribution in RGB space.

Example
Reference color (leaf): $x_e = [50, 160, 50]$
For a pixel $x = [60, 170, 55]$:

$$d(x, x_e) = \sqrt{(60 - 50)^2 + (170 - 160)^2 + (55 - 50)^2}$$

$$d = \sqrt{100 + 100 + 25} = \sqrt{225} = 15$$

Decision rule:
If $d \leq T$ (e.g., $T = 30$) $\rightarrow$ target
 $\rightarrow$ background

Result (illustrative)

3. Mahalanobis Distance and the Ellipsoidal Model

Idea
Use the Mahalanobis distance, which takes into account the correlation between color channels. The model assumes an ellipsoidal distribution in RGB space.

Example
Mean color: $\mu = [54.6, 164.2, 50.0]$
Covariance matrix (computed from samples):

$$\Sigma = \begin{bmatrix} 120 & 30 & 10 \\ 30 & 80 & 15 \\ 10 & 15 & 60 \end{bmatrix}$$

For a pixel $x = [60, 170, 55]$,

$$d_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \approx 2.14$$

Decision rule:
If $d_M \leq T$ (e.g., $T = 3$) $\rightarrow$ target
 $\rightarrow$ background

Result (illustrative)

4. Bounding-Box Approach and the Cuboidal Model

Idea
Define a rectangular range (box) in RGB space. Pixels whose R, G, B values fall inside the box are classified as target.

Example
Target color range (leaf):
• $R \in [40, 70]$
• $G \in [140, 180]$
• $B \in [30, 70]$

For a pixel $x = [55, 165, 50]$:
Since $40 \leq 55 \leq 70$, $140 \leq 165 \leq 180$, $30 \leq 50 \leq 70$
 $\rightarrow$ Pixel is inside the box (target).

Decision rule:
If $R_{\min} \leq R \leq R_{\max}$ and $G_{\min} \leq G \leq G_{\max}$ and $B_{\min} \leq B \leq B_{\max}$
 $\rightarrow$ target
else $\rightarrow$ background

Result (illustrative)

$$\mathbf{x} = [R, G, B]^T$$

1. Estimation of the Mean Color Vector

Suppose we select three pixels from a leaf:

Pixel	R	G	B
P_1	50	150	50
P_2	60	160	40
P_3	55	155	45

$$\boldsymbol{\mu} = \begin{bmatrix} (50 + 60 + 55)/3 \\ (150 + 160 + 155)/3 \\ (50 + 40 + 45)/3 \end{bmatrix} = \begin{bmatrix} 55 \\ 155 \\ 45 \end{bmatrix}$$

The mean color vector is:

Therefore, the reference color is $\boldsymbol{\mu} = [55, 155, 45]^T$.

2. Euclidean Distance and the Spherical Model

We compare a new pixel $\mathbf{x} = [60, 150, 50]^T$ with the mean color vector.

The Euclidean distance is: $d(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{(R - 55)^2 + (G - 155)^2 + (B - 45)^2}$

Substituting the values:

$$d = \sqrt{(60 - 55)^2 + (150 - 155)^2 + (50 - 45)^2}$$

$$d = \sqrt{25 + 25 + 25} = \sqrt{75}$$

$$d \approx 8.66$$

Suppose the threshold is $T = 20$.

- If $d \leq 20$, classify the pixel as a leaf.
- If $d > 20$, classify it as background.

Since $8.66 < 20$, the pixel belongs to the leaf.

Geometrically, all accepted pixels lie inside a sphere centered at the mean color vector.

3. Mahalanobis Distance and the Ellipsoidal Model

Unlike Euclidean distance, Mahalanobis distance accounts for the variability and correlation of the RGB components.

For a simple numerical example, suppose the covariance matrix is:

$$\Sigma = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

The Mahalanobis distance is: $d_M = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}$

For our pixel, the RGB difference is $[5, -5, 5]^T$. Therefore:

$$d_M = \sqrt{\frac{25}{100} + \frac{25}{100} + \frac{25}{100}}$$

$$d_M \approx 0.866$$

Suppose the threshold is $T = 2$. Because $0.866 < 2$, the pixel is classified as a leaf.

The accepted colors form an ellipsoid when the RGB components have different variances or correlations. In our simplified example, the covariance matrix is proportional to the identity matrix, so the ellipsoid is actually a sphere.

4. Bounding-Box Approach and the Cuboidal Model

Instead of calculating a distance, we define acceptable intervals for each RGB component.

Suppose the target leaf color is described by:

$$40 \leq R \leq 70 \quad 140 \leq G \leq 170 \quad 30 \leq B \leq 60$$

Consider the pixel: $\mathbf{x} = [60, 150, 50]^T$

Check each component:

- $40 \leq 60 \leq 70 \checkmark$
- $140 \leq 150 \leq 170 \checkmark$
- $30 \leq 50 \leq 60 \checkmark$

All three conditions are satisfied. The pixel belongs to the leaf.

Now consider $[80, 150, 50]^T$. Its red component exceeds 70, so it is classified as background.

Geometrically, the accepted colors form a cuboid in RGB space.

4.3. Comparison of Color Segmentation Methods

4.3.1. Sensitivity to Color Variations

Table 7: Comparison of RGB color-distance models

Method	Main principle	Decision criterion	Geometric model
Mean Color Vector	Estimate the average RGB color of the target region.	Compute the mean RGB vector from sample pixels.	Reference point
Euclidean Distance	Measure the straight-line distance between a pixel and the reference color.	$d(\mathbf{x}, \boldsymbol{\mu}) \leq T$	Sphere
Mahalanobis Distance	Measure color distance while accounting for variances and correlations between RGB components.	$d_M(\mathbf{x}, \boldsymbol{\mu}) \leq T$	Generally an ellipsoid
Bounding-Box Approach	Define an acceptable range independently for each RGB component.	$R_{\min} \leq R \leq R_{\max}, G_{\min} \leq G \leq G_{\max}, \text{ and } B_{\min} \leq B \leq B_{\max}.$	Cuboid

4.3.2. Influence of Color Space Selection and Thresholds

Color-space choice affects how target variation is represented. RGB is convenient and widely available, but channel changes are not perceptually uniform. HSI can make hue-based selection easier, while CIE-based spaces can support colorimetric comparisons when the image has been properly color-managed. Thresholds should be validated against representative examples that include expected lighting and background variation.

For robust applications, consider white-balance correction, illumination normalization, spatial constraints, morphology, or a learned segmentation model. A color threshold is a baseline method, not a guarantee of object recognition.

SPHERICAL, ELLIPSOIDAL, AND CUBOIDAL COLOR DECISION REGIONS

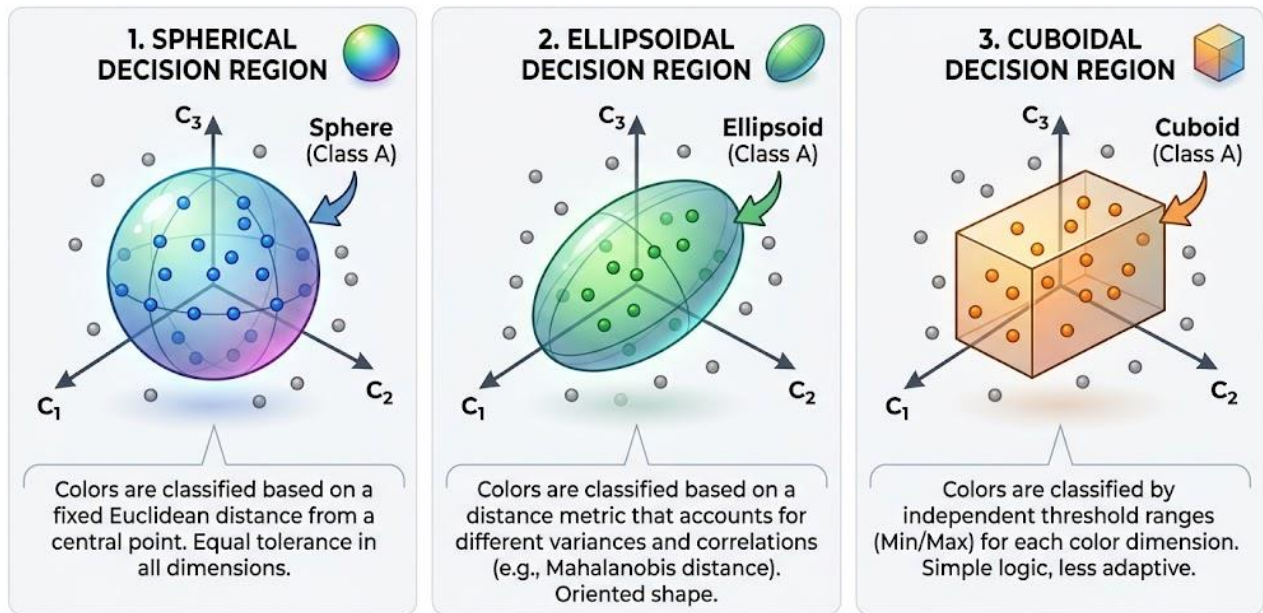


Figure 09: Spherical, ellipsoidal, and cuboidal color decision regions]

5. Comparative Summary and Practical Applications

5.1. Comparison of RGB, CMYK, and HSI Color Models

The three models solve different problems. RGB is suited to image acquisition and display because most cameras and screens operate with three-channel representations. CMYK is designed around printing with subtractive colorants and a separate black channel. HSI reorganizes color information into components that can be easier to interpret for selection and manipulation.

Table 8: Comparative summary of RGB, CMYK, and HSI

Criterion	RGB	CMYK	HSI
Underlying principle	Additive light	Subtractive colorants	Perceptually motivated decomposition
Components	Red, green, blue	Cyan, magenta, yellow, black	Hue, saturation, intensity
Typical domain	Capture, display, web	Printing and ink management	Color selection and image analysis
Main strength	Direct fit to common imaging hardware	Represents printing channels and black ink	Separates hue-like and brightness-like attributes
Main limitation	Device-dependent; channels mix color and brightness	Device/profile dependent; ink behavior is nonlinear	Not fully perceptually uniform; hue unstable near gray

5.2. Selecting a Color Space According to the Application

Image Acquisition and Display: RGB

RGB is the natural choice when working with camera outputs, display buffers, and most standard digital images. For color-critical work, identify the RGB space (for example, sRGB), transfer function, and white point rather than assuming that every RGB triplet has the same physical meaning.

Printing and Ink Management: CMYK

CMYK is appropriate for preparing separations for a known printing process. The conversion should use the destination printer or press profile, paper characteristics, black generation, and a suitable rendering intent. The idealized RGB-to-CMY equations are educational and should not be used as a substitute for color management.

Intuitive Color Selection and Manipulation: HSI

HSI can be useful for selecting a color family or adjusting saturation and intensity separately. It is often suitable for classroom demonstrations and rule-based segmentation. For high-quality editing, perceptual uniform spaces such as CIELAB or more modern appearance models may be preferable, depending on the task.

Edge Detection and Image Segmentation: Selection Based on Image Characteristics and the Chosen Method

Use grayscale or luminance-based edge detection when the relevant boundaries are primarily brightness transitions. Use color-gradient methods when chromatic boundaries matter. Use HSI thresholds when a target can be described by hue and saturation, or use vector-distance models when representative target-color samples are available. The best choice depends on the image, illumination variability, required precision, and computational budget.

5.3. Case Studies and Practical Applications

Case Study 1: Detecting a Colored Object

A simple object detector can convert an RGB image to HSI, apply hue and saturation thresholds, and refine the mask with morphological opening and closing. The method is suitable when the object's color differs clearly from the background. It should be tested on images with shadows, highlights, and background objects of similar color.

No numerical performance is claimed here; evaluation requires a labeled dataset and a stated metric.

Case Study 2: Color-Aware Edge Detection

Consider two neighboring regions whose luminance is similar but whose hues differ. A grayscale detector may return a weak response because the scalar intensity changes little. A vector color-gradient detector can detect the boundary because the RGB vector changes. The reverse issue is also possible: channel noise can create a strong vector response without a meaningful object boundary, so smoothing and threshold selection remain important.

Case Study 3: Color-Based Segmentation in Agricultural Images

Color thresholds can provide a first-pass mask for leaves, fruit, or visible disease symptoms when color is distinctive under controlled acquisition. However, symptom colors can overlap with shadows, soil, other plant parts, and illumination changes. A practical workflow should standardize acquisition where possible, inspect sample distributions, compare HSI thresholding with vector-distance baselines, and validate the mask against manually annotated ground truth.

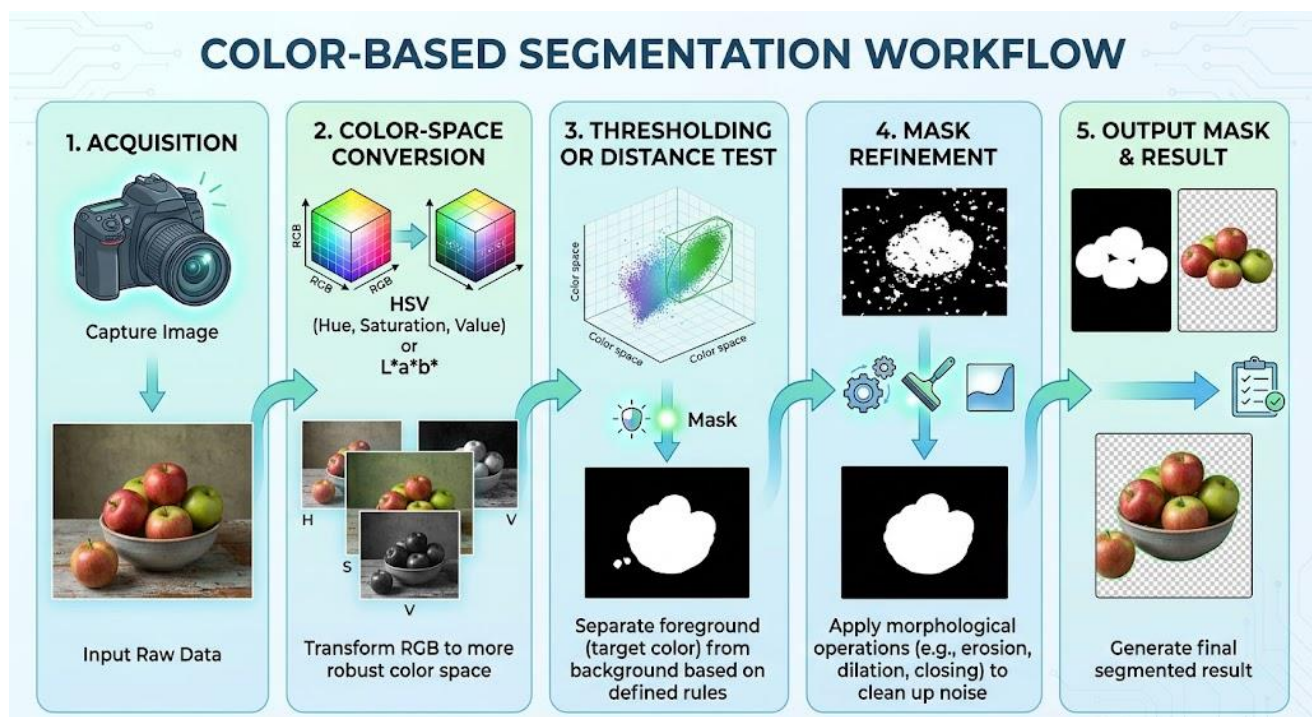


Figure 10: Example workflow for color-based segmentation

Suggested illustration: acquisition, color-space conversion, thresholding or distance test, mask refinement, and output mask.

Practical Checklist

- Identify whether the task concerns emitted light, printed colorants, perceptual selection, or quantitative color comparison.
- Confirm channel ranges, bit depth, color profile, and whether RGB values are linear-light or gamma-encoded.
- Normalize values before applying formulas that assume components lie in $[0, 1]$.
- Handle undefined hue and zero denominators explicitly.
- Choose edge or segmentation thresholds using representative images, not isolated examples.
- Report the color space, conversion convention, thresholds, preprocessing, and evaluation procedure so the method can be reproduced.

Chapter Summary

Color in digital images arises from the interaction of spectral light, surfaces, imaging devices, and human vision. Radiometry describes optical energy physically, whereas photometry weights that energy according to visual sensitivity. CIE XYZ and chromaticity coordinates support standardized color description, while gamut limits the colors that a device can reproduce.

RGB is widely used for acquisition and display, CMYK is tailored to printing, and HSI offers a convenient separation of hue, saturation, and intensity. Color-space conversions require careful treatment of ranges and special cases. Color edge detection benefits from a vector view of channel variation, while color segmentation can be based on HSI thresholds or geometric models in RGB space. In all cases, the representation and thresholds should be selected according to the application and validated with representative data.

Review Questions

- Why does a prism separate white light into a spectrum, and why is the visible spectrum not divided into perfectly discrete color bands?
- Distinguish radiance from luminance and give one application for each.
- Explain the difference between additive RGB mixing and subtractive CMY(K) mixing.

- What information is retained by CIE chromaticity coordinates, and what information is removed by normalization?
- Why are equal RGB components associated with the grayscale diagonal?
- State two reasons that an RGB-to-CMY complement conversion is not sufficient for production printing.
- Why is hue unstable for nearly gray pixels in HSI?
- How can a color-gradient method detect a boundary that a grayscale method misses?
- Compare Euclidean and Mahalanobis color distances in terms of their decision regions.
- Which color representation would you choose for display, print preparation, and hue-based segmentation? Justify each choice.

Suggested Practical Exercise

Use a small RGB image and perform the following tasks: (1) display the original image and its R, G, and B channels; (2) convert normalized RGB to CMY using the idealized complement equations; (3) compute HSI and visualize hue and saturation; (4) create a binary mask using a chosen hue and saturation interval; (5) compare a grayscale edge map with a color-gradient edge map; and (6) test Euclidean and bounding-box segmentation using a manually selected target region. Record the thresholds and discuss failure cases. Do not report accuracy unless a reference mask is available.

Notation and Formula Reference

Table 9: Principal symbols

Symbol	Meaning
Λ	Wavelength
N	Frequency
C	Speed of light in vacuum
H	Planck's constant
X, Y, Z	CIE XYZ tristimulus values
x, y, z	CIE chromaticity coordinates
R, G, B	Red, green, and blue components
C, M, Y, K	Cyan, magenta, yellow, and black printing components
H, S, I	Hue, saturation, and intensity in HSI
M	Mean target color vector
Σ	Color covariance matrix
T	Segmentation threshold
J	Color variation / structure tensor matrix

References for Further Reading

The following standard references provide background for the concepts covered in this chapter. Consult the edition available through your institution and follow the citation style required for the course.

- Gonzalez, R. C., and Woods, R. E. Digital Image Processing. Pearson.
- Fairchild, M. D. Color Appearance Models. Wiley.
- Wyszecki, G., and Stiles, W. S. Color Science: Concepts and Methods, Quantitative Data and Formulae. Wiley.
- CIE. Colorimetry, CIE 15. International Commission on Illumination.
- Poynton, C. A Technical Introduction to Digital Video. Wiley.