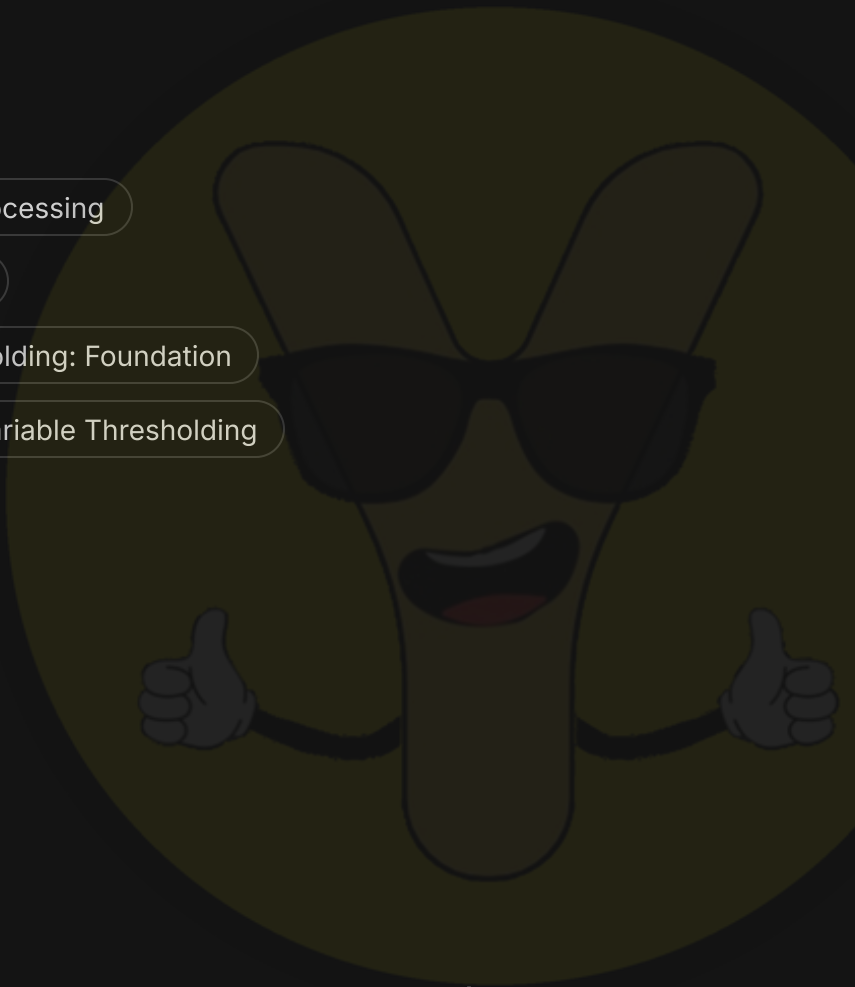
AKTU · 3RD YEAR · SEMESTER 6 · BCDS061

Image Analytics

Unit 3 — Colour Image Processing and Segmentation

- Colour fundamentals
- Colour models
- Pseudo colour image processing
- Basics of full colour image processing
- Colour transformation
- Smoothing and sharpening (colour images)
- Image Segmentation based on colour
- Active Contours: Snakes and Level Sets
- Noise in colour images
- Colour image compressions
- Thresholding: Foundation
- Basic Global thresholding
- Optimum Global Thresholding using Otsu's Method
- Multiple Thresholds
- Variable Thresholding



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Colour Fundamentals & Perceptual Attributes

TRISTIMULUS VALUES & CHROMATICITY

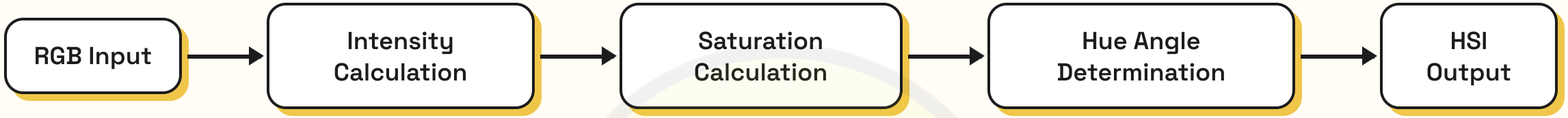
Colour perception relies on human eye cone receptors sensitive to Red (630nm), Green (530nm), and Blue (450nm). Chromaticity defines the quality of light measured by hue and saturation.

- Luminance Y represents the actual amount of radiant energy (brightness) perceived by an observer.
- Tristimulus values X, Y, Z denote the amounts of primary red, green, and blue needed to match any spectral colour.
- Chromaticity coordinates: $x = \frac{X}{X+Y+Z}$, $y = \frac{Y}{X+Y+Z}$, $z = \frac{Z}{X+Y+Z}$ where $x + y + z = 1$.
- CIE Chromaticity Diagram plots y versus x, defining all visible spectral hues along its boundary.

Standard Colour Models Comparison

Model	Type	Primary Application	Key Characteristics
RGB	Additive	Monitors, Cameras, Displays	Red, Green, Blue primaries; normalized inside unit cube $[0,1]^3$.
CMY / CMYK	Subtractive	Color Printing, Hardcopy Devices	Cyan, Magenta, Yellow, Key (Black); absorbs light spectrum.
HSI / HSV	Perceptual	Image Processing, Human Vision	Hue (color tint), Saturation (purity), Intensity (brightness).
YCbCr	Luminance-Chrominance	Video Compression (JPEG, MPEG)	Y is luminance, Cb and Cr are blue-difference and red-difference.

RGB to HSI Model Conversion Flow



Mathematical mapping from normalized RGB coordinates to HSI perceptual color space.

Pseudo Colour Image Processing

Assigning colors to monochrome images based on specific intensity transformation functions.

- Objective: Enhance human visual interpretation of monochrome images, as humans distinguish thousands of colors but only around 30 gray levels.
- Intensity Slicing: Divides grayscale range into discrete intervals, mapping each interval to a specific color code.
- Gray-Level-to-Color Transformations: Passes grayscale values through three parallel transformation functions $f_R(r), f_G(r), f_B(r)$.
- Frequency Domain Filtering: Splits monochrome image into low, mid, and high spatial frequency bands assigned to R, G, B channels.

Basics of Full Colour Image Processing

Processing vector-valued colour images using scalar channel-by-channel or direct vector operations.

- Vector Representation: Pixel value at coordinate (x, y) is a vector $c(x, y) = [R(x, y), G(x, y), B(x, y)]^T$.
- Scalar Processing: Process each RGB component independently using scalar algorithms: $g_i(x, y) = f[c_i(x, y)]$.
- Vector Processing: Processes color vectors directly as unified entities: $g(x, y) = T[c(x, y)]$.
- Equivalence Condition: Scalar and vector processing are equivalent ONLY IF the operation is linear and independent across channels.

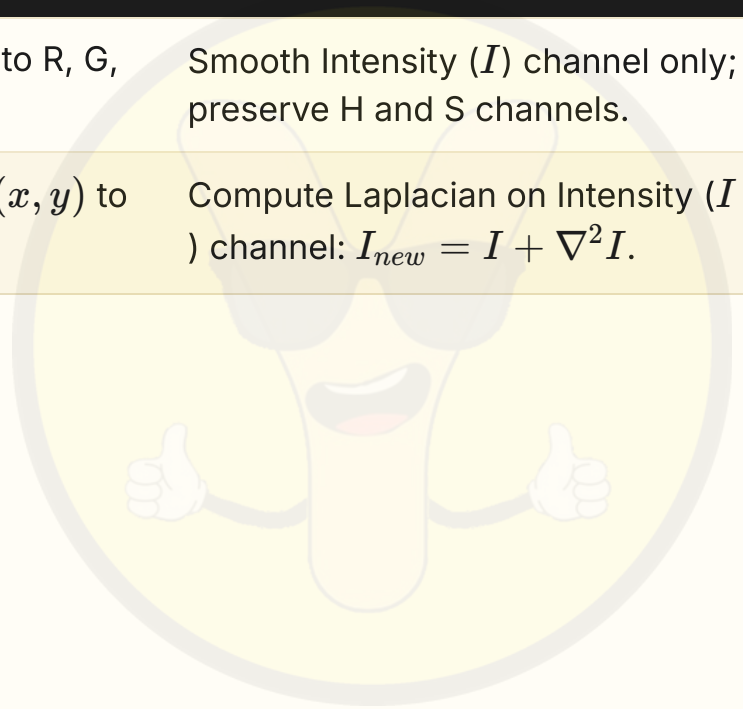
Colour Transformations

Modifying component values based strictly on individual pixel colour values.

- General Formulation: $s_i = T_i(r_1, r_2, ..., r_n)$ where r_i are input color components and s_i are output components.
- Color Complements: Generates color negative; in RGB space, $s_i = k - r_i$ where k is maximum intensity value (e.g., 255).
- Color Slicing: Highlights a specific region of color in space: $s_i = 0.5$ if $\sum |r_j - a_j| > D_0$ else retains original value.
- Tone and Color Corrections: Adjusts tonal range (shadows, midtones, highlights) and white balance using lookup tables.

Spatial Filtering: Smoothing vs Sharpening

Operation	RGB Space Method	HSI Space Method	Key Consideration
Smoothing	Apply spatial smoothing mask to R, G, B channels independently.	Smooth Intensity (I) channel only; preserve H and S channels.	Smoothing HSI intensity preserves hue integrity without color bleeding.
Sharpening	Apply Laplacian operator $\nabla^2 c(x, y)$ to R, G, B channels.	Compute Laplacian on Intensity (I) channel: $I_{new} = I + \nabla^2 I$.	Vector gradient computation avoids artifacting at sharp color transitions.



Colour Image Segmentation

RGB SPACE VS HSI SPACE SEGMENTATION

Partitioning a colour image into homogeneous regions based on colour similarity metrics.

- HSI Segmentation: Uses Hue (H) and Saturation (S) thresholds as a color mask, while Intensity (I) is ignored.
- RGB Vector Distance: Measures Euclidean distance $D(z, a) = \sqrt{(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2}$.
- Segmentation Threshold: A pixel z belongs to class a if $D(z, a) \leq D_0$, forming a bounding sphere or ellipsoid in RGB space.

Active Contours: Snakes & Level Sets

Deformable models driven by internal and external forces to delineate object boundaries.

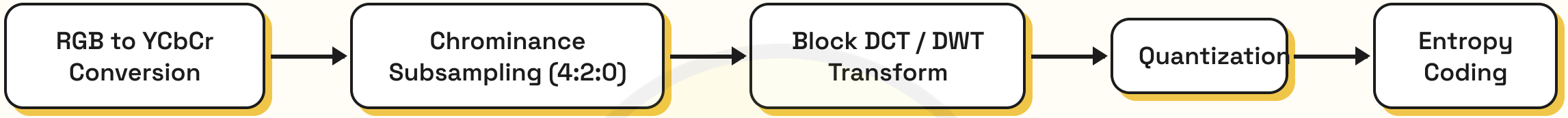
- Snakes (Parametric): Curves $v(s) = (x(s), y(s))$ minimizing energy $E^*_{snake} = \int_0^1 (E_{int}(v(s)) + E_{ext}(v(s)))ds$.
- Internal Energy (E_{int}): Controls smoothness and elasticity: $E_{int} = \frac{1}{2}(\alpha|v'(s)|^2 + \beta|v''(s)|^2)$.
- External Energy (E_{ext}): Attracts contour towards edges, derived from image gradients $-\|\nabla I(x, y)\|^2$.
- Level Sets (Geometric): Represents boundary as zero level set of higher-dimensional function $\phi(x, y, t) = 0$.
- Advantage of Level Sets: Handles topological changes (splitting and merging of boundaries) naturally without re-parameterization.

Noise in Colour Images

Characteristics and suppression of noise artifacts across colour components.

- Channel Corruption: Noise can affect individual channels independently (e.g., sensor noise) or across all channels correlatedly.
- Noise Spreading: Transforming a noisy RGB image to HSI spreads noise non-linearly across Hue and Saturation components.
- Vector Median Filters (VMF): Selects pixel vector minimizing sum of distances to all neighboring vector pixels inside a window.
- VMF Formula: $\vec{x}_{VMF} = \arg\min_{\vec{x}_k \in W} \sum_{j=1}^{N} \|\vec{x}_k - \vec{x}_j\|$, effective for impulse noise reduction.

Colour Image Compression Pipeline



Standard JPEG compression pipeline exploiting psycho-visual chrominance redundancy.

Foundation of Image Thresholding

IMAGE THRESHOLDING

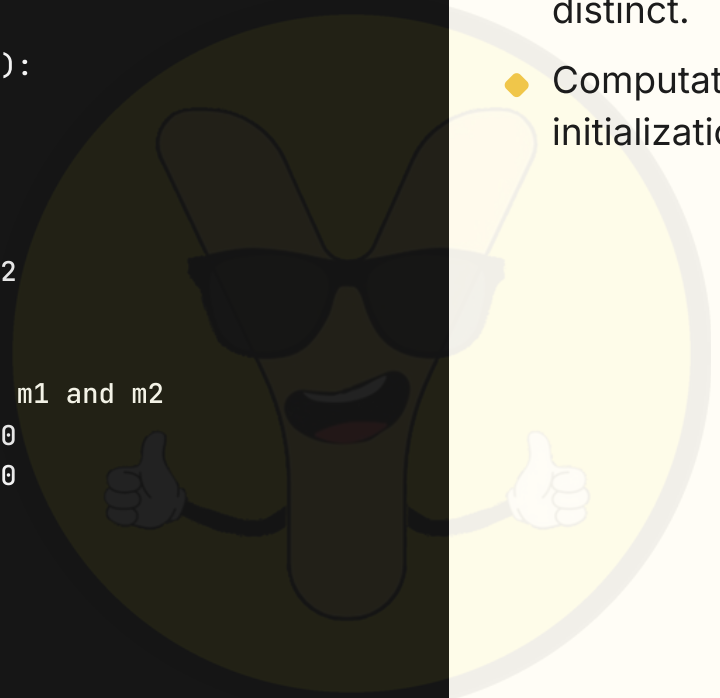
Segmentation technique converting grayscale or color image into binary image by comparing intensity values against threshold T.

- Mathematical formulation: $g(x, y) = 1$ if $f(x, y) > T$, else $g(x, y) = 0$.
- Bimodal Histogram: Ideal histogram containing two dominant peaks corresponding to background and foreground object.
- Role of Illumination: Non-uniform illumination shifts local pixel intensities, rendering global thresholding ineffective.

Iterative Basic Global Thresholding Algorithm

```python
import numpy as np

def basic_global_threshold(image, delta_T=0.5):
    # Step 1: Initialize T as mean intensity
    T = np.mean(image)
    done = False
    while not done:
        # Step 2: Segment image into G1 and G2
        G1 = image[image > T]
        G2 = image[image ⩽ T]
        # Step 3: Compute average intensities m1 and m2
        m1 = np.mean(G1) if len(G1) > 0 else 0
        m2 = np.mean(G2) if len(G2) > 0 else 0
        # Step 4: Compute new threshold
        T_new = 0.5 * (m1 + m2)
        # Step 5: Check convergence
        if abs(T - T_new) < delta_T:
            done = True
        T = T_new
    return T
```



- Guaranteed convergence when histogram peaks are distinct.
- Computationally simple and parameter-free initialization.

Optimum Global Thresholding (Otsu's Method)

Non-parametric, unsupervised method to find optimal threshold by maximizing between-class variance.

- Normalized Probabilities: $p_i = \frac{n_i}{M \times N}$, where $\sum_{i=0}^{L-1} p_i = 1$.
- Cumulative Probabilities: $\omega_0(k) = \sum_{i=0}^{k} p_i$, $\omega_1(k) = 1 - \omega_0(k)$.
- Class Means: $\mu_0(k) = \frac{1}{\omega_0(k)} \sum_{i=0}^{k} i p_i$, $\mu_1(k) = \frac{1}{\omega_1(k)} \sum_{i=k+1}^{L-1} i p_i$.
- Global Mean: $\mu_T = \sum_{i=0}^{L-1} i p_i$.
- Between-Class Variance: $\sigma_B^2(k) = \frac{[\mu_T \omega_0(k) - \mu(k)]^2}{\omega_0(k)(1-\omega_0(k))}$.
- Optimal Threshold k^*: Select $k^* = \arg\max_{0 \le k \le L-1} \sigma_B^2(k)$.

Worked Example: Otsu's Method Calculation

Calculate the optimal threshold k^* for a 3-bit image (levels 0-7) with histogram counts $h = [8, 7, 6, 5, 2, 4, 3, 5]$ and total pixels $N = 40$.

1. Step 1: Compute normalized probabilities $p_i = [0.20, 0.175, 0.15, 0.125, 0.05, 0.10, 0.075, 0.125]$.
2. Step 2: Calculate Global Mean $\mu_T = \sum i \cdot p_i = 3.10$.
3. Step 3: Evaluate between-class variance $\sigma_B^2(k)$ for candidate thresholds $k \in [0, 6]$.
4. Step 4: For $k = 2$: $\omega_0 = 0.525$, $\omega_1 = 0.475$, $\mu_0 = 0.905$, $\mu_1 = 5.526$.
5. Step 5: Compute $\sigma_B^2(2) = 0.525 \cdot 0.475 \cdot (0.905 - 5.526)^2 \approx 5.32$, which is the maximum variance.

Optimal threshold level $k^* = 2$.

Multiple vs Variable (Adaptive) Thresholding

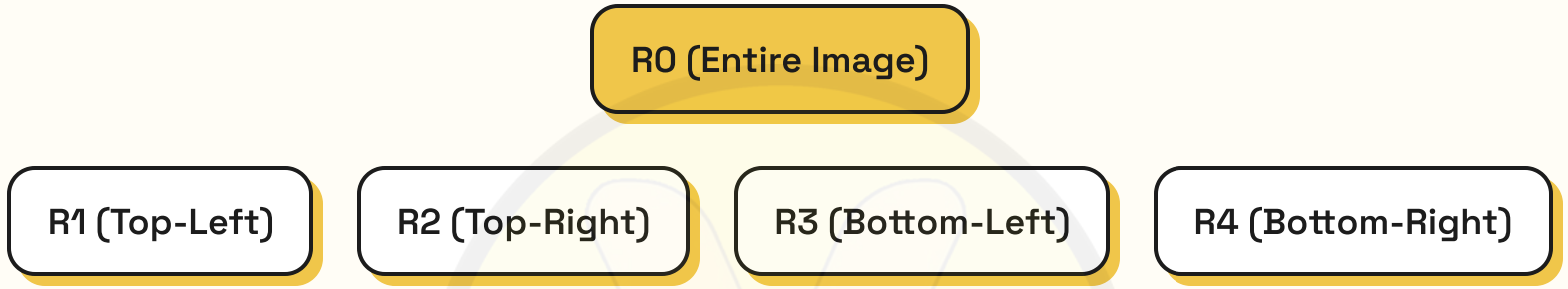
Feature	Multiple Thresholding	Variable (Adaptive) Thresholding
Definition	Uses multiple thresholds $T_1, T_2, ..., T_n$ to segment into multiple classes.	Threshold $T(x, y)$ varies dynamically across pixel coordinates (x, y).
Formulation	$g(x, y) = a$ if $f(x, y) \leq T_1$; b if $T_1 < f(x, y) \leq T_2$; etc.	$g(x, y) = 1$ if $f(x, y) > T(x, y)$ else 0.
Threshold Basis	Global histogram multimodal analysis.	Local neighborhood statistics (local mean m_{xy}, std dev σ_{xy}).
Best Used For	Images containing distinct object classes of varying intensities.	Images with non-uniform illumination or uneven background lighting.

Region-Based Segmentation: Region Growing

Group pixels or sub-regions into larger regions based on predefined similarity criteria.

- Seed Selection: Initial pixels selected manually or automatically based on feature criteria (e.g., local intensity extrema).
- Growing Criterion: Unallocated neighbor pixel p is added to region R_i if $|f(p) - \mu_{R_i}| \leq T_R$.
- Stopping Rules: Process halts when no unassigned neighboring pixels meet the similarity condition.
- Advantages: Simple concept, yields well-connected regions with smooth boundaries.
- Disadvantages: Computationally expensive, highly sensitive to noise and seed point selection.

Region Splitting and Merging Quadtree Structure



Quadtree decomposition: Iteratively split non-uniform regions into quadrants until predicate P(Ri)=TRUE, then merge adjacent uniform regions.

Exam Corner: Important Questions

Q1 1. Explain RGB, CMY, and HSI colour models with relevant conversion formulas from RGB to HSI space.

Q2 2. Differentiate between scalar and vector processing in colour images. Why is scalar processing not always equivalent to vector processing?

Q3 3. Derive the mathematical expression for Otsu's optimum global thresholding method and state its steps.

Q4 4. Compare active contour models: Parametric Snakes vs Level Set Methods in terms of energy minimization and handling topology changes.

Q5 5. Explain the algorithm for Region Splitting and Merging with the help of a Quadtree diagram structure.

Q6 6. Write short notes on: (a) Pseudo Colour Image Processing, (b) Vector Median Filter for colour noise reduction.

Unit 3 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 4 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

