Lecture Notes

Computer Vision and Pattern Recognition 8820
Spring 2026

1 Background

1.1 Pinhole Camera and Perspective Foreshortening

Pin hole camera → approximation (formed at the focal plane of the camera) this doesnt hold true always.

Perspective Projection → used by humans (the model for approximation)

$$\sqrt{\left(\frac{fx_1}{z} - \frac{fx_2}{z}\right)^2 + \left(\frac{fy_1}{z} - \frac{fy_2}{z}\right)^2} = \frac{f}{z}\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

Where, f is the camera focal length, z is the depth of the points from the camera, and (x_1, y_1), (x_2, y_2) are their coordinates in the camera frame.

1.2 Digital Image and Discretisation

In a 2D sensor (CCD Camera), the image intensity is sampled at discrete points in image plane. Image plane is discretised into cells and pixels (picture element)

An Image is essentially a 2D array (matrix) of cells and pixels The pixel coordinates (i, j) refer to center of the cell. The pixel size essnetialy the image resolution can be measured in terms of pixel per unit area.

Discretisation of the image plane is equivalent to spatial sampling.

Sampling Density/Resolution Pixels/Unit area

Higher Sampling Density leads to higher image resolution

For a image plane of a given area → smaller pixel size implies higher resolution Digital Image → Discretisation of spatial coordinates (sampling)

Discretization of intensity values (Quantization)

Quantization is the discretization of intensity values.

If each pixel is represented using n bits, then the intensity values range from 0 to $2^n - 1$.

For $n = 8$,

$$\therefore \text{ gray-level intensity values} \in \{0, 1, \ldots, 255\}.$$

1.3 Levels of Computation

1. **Pixel Level Computation (contrast enhancement)**
   - Output pixel (assume gray scale) Location of pixel or intensity of the pixel (understood in context) (spatial and gray scale intensity can mean both)
   - Each pixel is processed independently (parallelism can be applied) (massively data parallel operations)
2. **Local Level Computation (Parallelizable but not easy) (edge)**
   - Output Pixel Value = f(input pixel value, values of pixels in the neighborhood of the input pixel) (overlapping neighbourhood)
3. **Global Level Computation (distance of gray levels within image)**
   - Performed on all pixels in image
   - Histogram computation
4. **Object Level Computation**
   - This is performed after a semantic entity is extracted from the image (image region)
     - Region Size
     - Region Shape
     - Region Intensity / Texture

2 Binary Images

2.1 Background

1 pixel → Object Pixel

0 pixel → non object pixel (background pixel)

Shape or Geometry Conveyed (Eg: Silhouette of the person)

Operations on bits (very efficient) (robotic sorting)

How would you create a binary image?

Binary Image (Created by thresholding a gray scale image)

2.2 Binary Image Formation (Image Taken)

$$B(i,j) = \text{threshold}\big(F(i,j)\big)$$

$$\text{threshold}\big(F(i,j)\big) = \begin{cases} 1, & \text{if } F(i,j) > T \\ 0, & \text{otherwise} \end{cases}$$

white objects against dark background

Note. Selection of **T** is based on **domain knowledge**

3 Geometric Properties of Interests

3.1 Object Size

Note. Assumption: image contains one object.

1. Object Size.
   - $$A = \sum_{i=1}^{n} \sum_{j=1}^{m} B(i,j) \tag{2}$$
2. Centroid
   - $$X_C = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} j \cdot B(i,j) \tag{3}$$
   - $$Y_C = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} i \cdot B(i,j) \tag{4}$$
3. Orientation of Object
   - Normalising the coordinates based on the centroid

     $$x' = j - x_c$$

     $$y' = i - y_c$$
   - Centralised coordinates wrt to centroid

     $$a = \sum_{i=1}^{n} \sum_{j=1}^{m} [x'(i,j)]^2 B(i,j) \tag{5}$$

4 Moments

4.1 Object Orientation

To compute the second-order moments a, b, and c, we start with the equation:

$$\tan(2\theta) = \frac{b}{a - c}$$

$$a = \sum_{i=1}^{n} \sum_{j=1}^{m} [X'(i,j)]^2 \cdot B(i,j) \tag{6}$$

$$b = 2 \sum_{i=1}^{n} \sum_{j=1}^{m} [X'(i,j)][Y'(i,j)] \cdot B(i,j) \tag{7}$$

$$c = \sum_{i=1}^{n} \sum_{j=1}^{m} [Y'(i,j)]^2 \cdot B(i,j) \tag{8}$$

Consider the following expression for X^2:

$$X^2 = \frac{1}{2}(a+c) + \frac{1}{2}(a-c)\cos(2\theta) + \frac{1 \cdot b}{2}\sin(2\theta) \tag{9}$$

Here, X^2 represents the moment of inertia.

Substitution Step: Now, we substitute $2\theta_1$ and $2\theta_2$ into the expression for X^2. The value of X^2 is minimized for one of them (let's call it $2\theta_1$) and maximized for the other (denoted as $2\theta_2$).

Thus, the axis of orientation is the one that minimizes X^2.

Elongation: The elongation ratio is given by:

$$E = \frac{X_{\max}}{X_{\min}}$$

where $X_{\max}$ and $X_{\min}$ are the maximum and minimum moments of inertia.

For a **sphere** the elongation is:

$$E = 1$$

4.2 Projections

$$H[i] = \sum_{j=1}^{m} B(i,j) \tag{10}$$

$$V[j] = \sum_{i=1}^{m} B(i,j) \tag{11}$$

Where, H and V are compact representations of $B(i,j)$.

5 Topological Definitions

5.1 Path

A sequence of pixels where successive pixels are neighbours

4 path $\rightarrow$ successive pixels are 4 neighbours

8 path $\rightarrow$ successive pixels are 8 neighbours

5.2 Foreground

The set of pixels with value 1 denoted by S.

5.3 Connectivity

A pixel, $p \in S$ connected to pixel $q \in S$ if there exists a path from p to q consisting entirely of pixels in S (foreground pixels)

4 connectivity 4 path

8 connectivity 8 path

5.4 Relations

- p is trivially connected to p (reflexive)
- p is connected to q $\iff$ q is connected to p (symmetric)
- if p is connected to q and q is connected to r then p is connected to r (transitive)

A subset of pixels on S in which each pixel is connected to all other pixels in the subset.

4 conncected component

8 connected component

The connectivity relation partitions S into connected components

Definition of a partition of a set S into components S1, S2 and S3

- $S_i = S$
- $S_i \cap S_j = \phi \forall$ i and j

6 Image Segmentation

Let $\bar{S}$ = complement of S

$\bar{S}$ = set of 0 - pixels $\because$ S is foreground

The set of of all pixels connected components of $\bar{S}$ that gave some pixels on the image border are referred to as the background.

The other components of $\bar{S}$ are referred as holes.

Holes are semantically tied to the geometry of the object.

(related to the object)

Background is semantically not related to the object

6.1 Criteria/Caveat

Different criteria need to be used for S and $\bar{S}$ in terms of connectivity

0 1 0

1 0 1

0 1 0

4 4 connected comp in S

$S\prime_{int} = S' - S'_{boundary}$

Surrounded / Contains Relation

A set of pixels T is said to be surrounded a set of pixels S if a 4 path from any pixel of S to the image border must intersect T

If T surrounds S then, S is contained in T

7 Connected Component Labelling

The process by which one assigns a unique label to to each of the connected components

Components in S

Components in S

All the pixels within the same component will be assigned the same label

7.1 Recursive

Recursive CCL Algorithm

- Scan the image until an unlabelled 1 pixel is found and assign it a new label L.
- Assign recursively the label L to all of its neighbouring one pixels...
- Halt the recursion when no unlabelled 1 pixels are found .
- Repeat 1 to 3 until no more unlabelled one pixel are found

Note. Check relation of CCL with flood fill

7.2 Iterative

Sequential

Iterative CCL Algorithm

Image Scan

1. Raster Scan (left to right, top to bottom array scan) 2. Serpentine Scan (lawn mover scan)

4CC

1. *Raster scan the image* 2. *If the given pixel p is a 1 pixel* 1. *if only of u or l has a label then assign that assign label to p* 2. *Both u and l have the same label then assign that label to p* 3. *pixel u and l have different labels then assign p the label of u and enter the two labels in an equivalence table* 4. *Create a new label and assign it to pixel p* 5. *Continue with the raster scan until no more un-labeled 1 pixel exist.* 3. *Raster scan the image and for each 1 pixel assign it a single unique label from its equivalence class.*

8 Boundary Extraction

Binary Image → CCL → Size Filtering

Size Filtering

Replace all components when area ¡ t with all other 0' s

(below the size of smallest object or interests)

Euler Number

Computed for various objects

E = n.components - n.holes

Boundary Extraction for a Component

1. Perform a raster scan to find the starting pixel of that component

1. Current pixel ← starting pixel

b ⟵ 4 neighbour to the west of current pixel

1. Enumerate the 8 neighbour of current pixel starting with b in clockwise order. Let these be n1, n2 ... n8 (where n_1 = b)

1. Determine the smallest i such that n_i belongs to the component

1. Append the current pixel to the contour list (also called a boundary list) 2. Update current $pixel = n_i b \leftarrow n_i - 1$ 3. Repeat Step 3 to 6 until current pixel comes back to starting pixel

Boundary Properties

1. Perimeter (8 connected) 1. Trace the boundary list 2. if the sucessive pixels are 4 neighbours add 1 3. if the successive pixels are diagonal neighbour add $\sqrt{2}$ 2. Compactness of the component

$Compactness = Perimeter^2/Area$

The smaller the compactness the more compact the object

of all 2d shapes the circle has the smallest compactness →

$$Circle = 4\pi^2 r^2/\pi r^2$$

9 Skeletonization of a 2D Shape

Activity Analysis ↔ Animation

A 2D shape in the current case is represented as a *binary component*.

9.1 Distance Measure Properties

A distance function $d(p, q)$ satisfies the following properties:

- **Non-negativity**
$$d(p, q) \geq 0$$
- **Identity of indiscernibles**
$$d(p, q) = 0 \iff p = q$$
- **Symmetry**
$$d(p, q) = d(q, p)$$
- **Triangle inequality**
$$d(p, r) \leq d(p, q) + d(q, r)$$

9.2 Distance Measures in a 2D Digital Image

Let
$$p = (i_1, j_1), \quad q = (i_2, j_2)$$

9.2.1 Euclidean Distance

$$d_E(p, q) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$$

9.2.2 Chessboard Distance

$$d_C(p, q) = \max\left(|i_1 - i_2|,\ |j_1 - j_2|\right)$$

9.3 Isodistance Contours

The geometric locust of all points (pixels) that are at a fixed distance from a given point (pixel)

Euclidean distance contour is a circle

Skeletonization Cont.

The distance transform (DT) of all pixels in component S is the minimum values from the background $\bar{S}$ for each pixel in S. The distance transform DT image DT(i, j) gives the minimum distance of each pixel (i, j) from $\bar{S}$.

9.4 Distance Parallel Algorithm

$$DT^{\circ}(i,j) = B(i,j)$$
$$DT^{n}(i,j) = DT^{\circ} + min(DT^{n-1}(u,v)$$

(u,v) = set of pixels that $d((i,j),(u,v)) = 1$ unit neighbourhood of (i, j)

d = euclidean, manhattan (u, v) = 4 neighbour of (i, j)

d = chessboard

(u,v) = 8 neighbours of (i, j)

Check for convergence at end $DT^{n}(i,j) \neq DT^{n-1}(i,j)$ (hence not converged)

9.5 Skeleton or the medial axis transform *MAT(i, j)*

$$\text{MAT}(i,j) = \begin{cases} 1, & \text{if } DT(i,j) \geq DT(u,v) \text{ for all } (u,v) \in N(i,j) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

MAT$(i,j) \longleftarrow$ represents the skeleton of the original compound.
Given $MAT(i,j)$ and the DT value for each non zero pixel in $MAT(i,j)$ then one reconstruct the original component (using reverse distance propagation).
Halt when,

$$DT(i,j) = DT(i,j)^{n-1} \ \forall \text{ i, j} \tag{13}$$

10 Region Analysis in Grayscale Images

10.1 Formal Definition of image segmentation

Given an image I, compute a partition $R_1, R_2, ..., R_n$ such that,

$$\bigcup_{i=1}^{n} R_i - \text{exhaustive} \tag{14}$$

$$R_i \cap R_j = \phi \text{ if i and j } - \text{ mutually exclusive/nonoverlapping} \tag{15}$$

Let P be a predicate that incorporates some homogeneity notion

$$P(R_i) = 1 \tag{16}$$

$$P(R_i \cup R_j) = 0 \text{ for i } \neq \text{ j} \tag{17}$$

- The optimum partition is one for which n is the minimum. (partition into fewest regions)
- Homogeneity criteria needs to be quantified

10.2 Image Segmentation (Semantic Image Segmentation)

Partition the image into regions that correspond to distinct objects

Assign semantic label to each region - semantic segmentation

Two Approaches

1. Region based segmentation (similarity)
    - Group together or cluster all pixels that belong to a single object - clustering
2. Edge Based Segmentation (dissimilarity)
    - Detect edge pixels and construct region boundaries or region contours and identify regions - edge detector

11 Modal Analysis

11.1 Iterative Threshold Selection

1. Determine an initial estimate for T = avg of intensity values of the pixels in the image.
2. Partition the image into two classes R_1 and R_2 using T.
3. Compute the mean gray levels μ_1 and μ_2 of the pixels in R_1 and R_2 respectively.

$$T_{new} = \frac{\mu_1 + \mu_2}{2}$$

4. If $T_{new} = T$ halt else update $T \longleftarrow T_{new}$ and repeat step 2 to 4

Note. For **step 4** we can also do

$$\alpha \cdot \mu_1 + \beta \cdot \mu_2$$

where α and β are the fraction of pixels in R_1 and R_2 $(\alpha + \beta) = 1$

11.2 Non-Uniform Background (variable thresholding)

- Capture the variation in background intensity
- Eliminate the variation in background intensity from the pixel values $\longrightarrow$ normalization
- Threshold based on normalized pixel values

Assumptions

1. Variation in background intensity values is systematic.
2. Systematic Variation in background intensity values can be captured by a simple function. bilinear and biquadratic
3. Assume that background pixels dominates the foreground pixels. Number of background pixels exceed the number of foreground pixels which essential employs that variation in image intensity values can be attributed to the variation in background intensity values

11.3 Bilinear Regression

$$\frac{1}{N^2} \sum_{x_i} \sum_{y_i} |F(x_i, y_i) - F'(x_i, y_i)| \tag{18}$$

Compute the normalized pixel values

$$F_n = F(x, y) - F'(x, y)$$

Map $(F_n)_{min} \ldots (F_n)_{max} \implies (0, \text{max gray level})$

use iterative thresholding to threshold the normalized image

11.4 Model Fitting

$$F_L(x, y) = Ax + By \tag{19}$$

linear

$$Q(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F \tag{20}$$

biquadratic

$$R(x, y) = Ax^3 + Bx^2 + Cxy^2 + Dy^3 + Ex^2 + Fxy + Gy^2 + Hx + Iy + J \tag{21}$$

bicubic

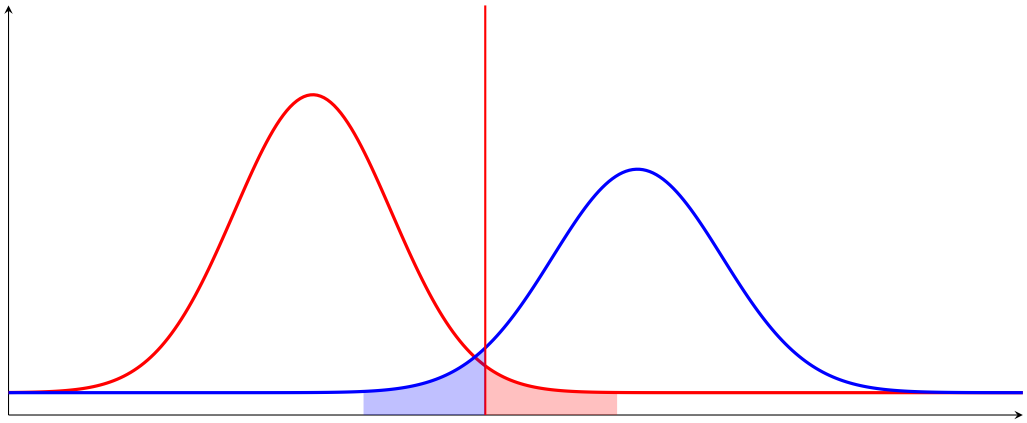
As the number of coeff increases so does the computational complexity increases.

1. Pro's
   - Less fitting error (higher fitting accuracy)
2. Con's
   - Computational complexity
   - overfitting

12 Adaptive thresholding

1. Divivde the image into sub images
2. Perform thresholding independently in each of subimage (eg iterative thresholding)
3. Combine the threshold results.

Resulting mosiac image will have inconsistency (artifacts).



13 Combine histogram with partial information

13.1 Dual threhsolding with region growing

Algorithm

1. Select threshold T_1 and T_2
2. Parititon the image into 3 pixel classes
   $R_1 \longleftarrow$ graylevel $\leq T_1$ - gaurantee foreground pixel
   $R_2 \longleftarrow$ graylevel $\geq T_2$ - gaurantee background pixel
   $R_3 \longleftarrow T_1 <$ graylevel $< T_2$ - could be either foreground or background
3. Visit each pixel in class R_3. If the pixel has a neighbour in class R_1 then reassign that pixel to R_1
4. Repeat step 3 until no more R_3 pixels are reassigned.

Tries to reduce the **misclassification** by using spatial coherence

14 Region Representation Schemes

Note. Not an Exhuastive List

1. **Array representations**

   - **Label Array Representation**

   $$L(i,j) = \alpha \text{ if pixel } (i,j) \text{ in image } I(i,j) \text{ has the label } \alpha \text{ belongs to region with label } \alpha \tag{22}$$

   - **Bitmap Representation**
     Each label α has an associated bitmap β^α defined as

     $$\beta^\alpha(i,j) = \begin{cases} 1 & \text{if pixel } (i,j) \text{ in image } I \text{ belongs to class } \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

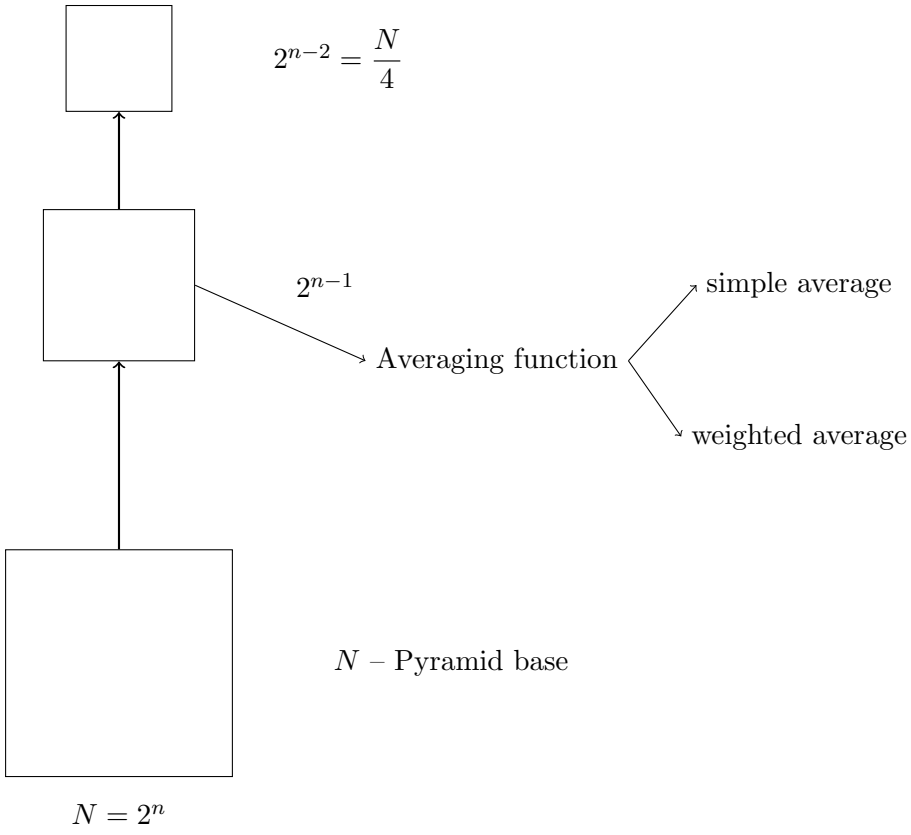
     The bitmap β^α is referred to as the *mask* for class α.

2. **Heirarchical Representation**
   Represent image and regions at multiple levels of representations.

   - Fine analysis at higher resolution (detached)
   - Coarse Analysis at lower resolution

   Only a subset of the image is analyzed at higher resolution - **Focus of Attention** (FOA)
   FOA is Dynamic

   - **Image Pyramid**


$2^{n-2} = \frac{N}{4}$
2^{n-1}
Averaging function
simple average
weighted average
N – Pyramid base
$N = 2^n$

15 QuadTree

Each node has 4 children or no children (leafnode)

Roof node represents the entire image
A leaf node in the quad tree represents homogenous region otherwise the node is split into 4 child nodes

OctTree

3-Dimensional representations

16 Picture tree

- leaf node represent a homogenous region
- a non leaf node is split into child nodes based on containment
- not a regular structure unlike quad and oct tree

17 Region Adjacency Graph (RAG)

Scene Graph (RAG is more specialized variant)

- Nodes: Represent Regions
- Arcs: Represent Common Boundary between two regions

Note. Nodes and Arc possess attribtues

Node Attributes: Region Attributes

- Geometric Attributes/Features
- Spectral Attributes (avg gray level, texture, color etc . . .)

ARC Attributes: Contour Attributes

-
-

RAG Generation

Given a label array $L(i, j)$ one can generate an RAG

1. Raster Scan the L(i,j)
2. Consider pixel (i,j)
   - If node with label L(i,j) exists then update the node in the RAG to include pixel(i, j)
   - else create a new node with label L(i, j) cosnsiting of pixel (i, j)
3. Examine the neighbour pixel (i,j)
   - if an arc exists between (i, j) and its neighbours region then update it
   - else create a new arc
4. Continue with raster scan until all pixels are exhausted

Given the bitmaps B^{α} (i, j)

1. Determine the connected components of $B^{\alpha}(i, j)$ for each α
2. Create a node for each connected component for all α
3. Trace the boundary of each cc to generate the arcs between the nodes

18 Over Segmentated Image

Need to start with a over segemented image because otherwise you will end up improper merge ...

19 Merge based segementation

Over segement the image
Create initial RAG for the over segemented image
visit each node in the RAG
a. conside the adjacent nodes
b. if the current node and its adjacent node satisfies the merge criteria then merge the nodes and update RAG
4. Repeat Step 4 unitl no more merges are possible

20 Node Selection

- Random selection of the node
- Order the nodes based on same criterion and select the best node "best" node - longest region (smallest region)
- list of all pairs of adjacent nodes and select the best pair for merging

21 Mean Based

Boundary Contrast Based, Variance Based or Mean Based
$|\mu_i - \mu_j|$

22 Split and Merge Segmentation using RAG

Merge Algorithm Parameters

- Selection of nodes for merging
- Merge Criterion
- Stating point over segmented image
- An improper merge cannot be undone

Split Based Algorithm

1. Start with an initial RAG
2. Consider a node for splitting
3. If the node satisfies the split criterion then split it and update RAG
4. Perform steps 2 and 3 until no more splits are possible

- Initial RAG - Undersegmented Image
- Undersegmented Image - Binary Thresholding + CCL

Splitting Criteria for a node

- Variance test performed in the pixels within the region/component/node
- how to split a node
  - edge detection within region
  - contour following
  - split the region along the contours
  - split the region along predefined boundary (quad tree)

approxmiate the region by fitting to a func
(homogeneity test) low mean square error then good fit

Parameters

- splitting criterion
  - Variance test

- Function Fitting test (bi linear and quadratic)

- Avg Fitting error
- Pixels where fitting error $\gg$ avg fitting error

- Candidate along which to split the region
- Where to split the region
- Node Selection

Split where the fit error is higher (not only tells when not to split but also where to split)

Combined Split and Merge Algorithm

1. Start with an initial RAG
2. Go through a split phase
   - Select a node that satisfies the split criterion and split it and update RAG.
3. Merge Phase
   - Select a pair of adjacent node
   - If they satisfy the merge criteria, then merge them and update the RAG
4. Perform the split and merge phase alternatively
5. Halt when no more splits or merge are possible

Maintain a list of previously visited k solution states

if the current split/merge results in a previously visited state then reject it and select another node

23 Iterative Model Fitting (Segementation Technique) with Region Growing

Basic Premise

1. Segmented image can be modeled as a partition of regions approximated by a parametric function (constant, bilinear, biquadratic, bicubic).
2. Fitting a function:
$$f = f(x, y, a, m) = \sum_{i+j \leq m} a_{ij} x^i y^j$$
where x, y are variables and a, m are parameters.
3. For a bilinear fit:
$$f(x, y) = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{02}y^2 + a_{11}xy$$

Fitting Error

The fitting error over a region R is defined as:

$$X^2(R, a, m) = \sum_{(x,y) \in R} [F(x, y) - f(x, y, a, m)]^2 \tag{24}$$

where $F(x, y)$ is the image intensity function.

For a given region R, determine the coefficients a and model order m that minimize the error X^2.

Algorithm

1. Determine seed regions in the image — the most homogeneous core of each region.
   - Use a 5×5 or 7×7 window.
   - Perform a bilinear fit and compute the fitting error.
   - Select the center of the window where the error is $< T_1$.
2. For each seed region, perform the following:
   (a) For a given model order, expand the current window (region growing) while keeping the same $\underline{a}$ and $\underline{m}$. Compute the fitting error (model extrapolation).
      - Repeat step (a) until the fitting error $> T_2$.
   (b) Refit using the same m to obtain a new a, then repeat step (a).
   If the refitting error $> T_3$, increase the model order and repeat.
   (c) Repeat steps (a) and (b) until the maximum model order is reached.

24 Edge Detection

Edge
Location of $\underbrace{\text{significant change}}_{\text{rate of change}}$ in a "local image property"
Local Image property - gray level, color, texture, motion

$$f(t) - \frac{df(t)}{dt} \tag{25}$$

$$f_\circ = \underbrace{f(t)}_{\text{signal}} + \underbrace{n(t)}_{\text{noise}} \tag{26}$$

$$\frac{df_\circ(t)}{dt} = \frac{df(t)}{dt} + \frac{dx(t)}{dt} \tag{27}$$

$$f(t) = A_S \sin(2\pi f_S t)$$

$$n(t) = A_n \sin(2\pi f_n t)$$

$$f_\circ(t) = A_S \sin(2\pi f_S t) + A_n \sin(2\pi f_n t)$$

where, $f_n \gg f_s$

$$SNR = \frac{A_s^2}{A_n^2}$$

Change detection is inherently noisy
Change detector is preceded by filtering or smoothing for noise removed

Noise samples $n_1, n_2 \ldots n_k$

$\sigma^2 \longleftarrow NoisePower$

$Var(n_i = \sigma^2)$

$n_s = \frac{n_1+n_2+\ldots+n_k}{k}$ $var(n_s) = \sigma^2/k$

Smoothing - Avg Direction