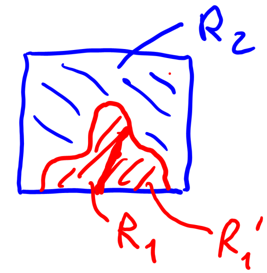
9 Image Segmentation

9.1 Image Segmentation

Decomposition of scene into its components

- Key step in image analysis and object-based coding
- Correspondence to physical objects (ideal)



Spatial segmentation

- Into different regions

Temporal segmentation

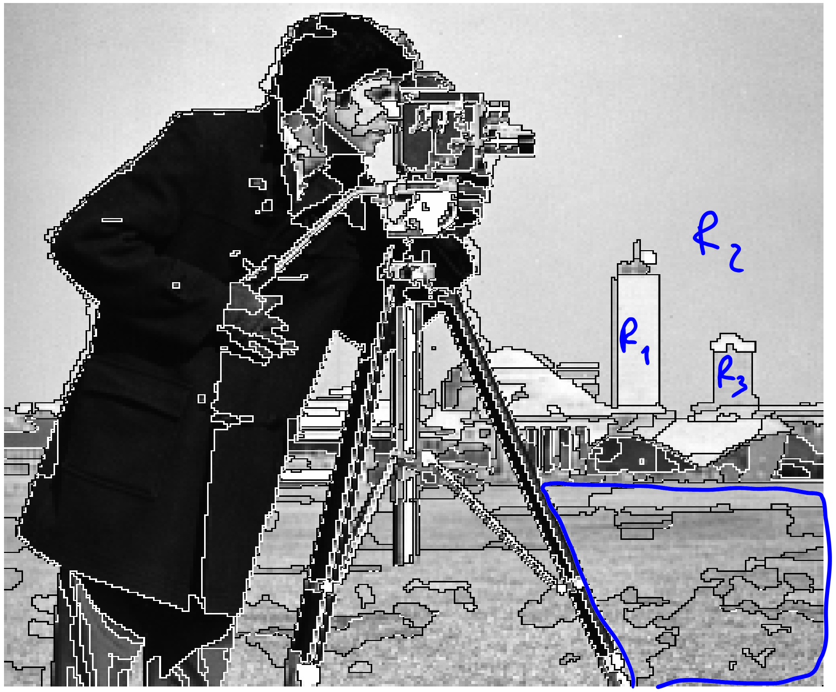
- Shot detection

Requirements for complete segmentation

- **Connectivity:** each region (segment) consists of connected image points
- **Completeness:** union of all regions yields complete image
- **Homogeneity:** each region is homogeneous under given criterion
- **Closeness:** combining two segments gives inhomogeneous region

Result of segmentation process is also called **partition**

Partition Examples



Cluster-Based Segmentation

Assumptions

- Each pixel is assigned a set of features (color, gradient, texture, etc.)
- Feature space reveals significant cumulative clusters

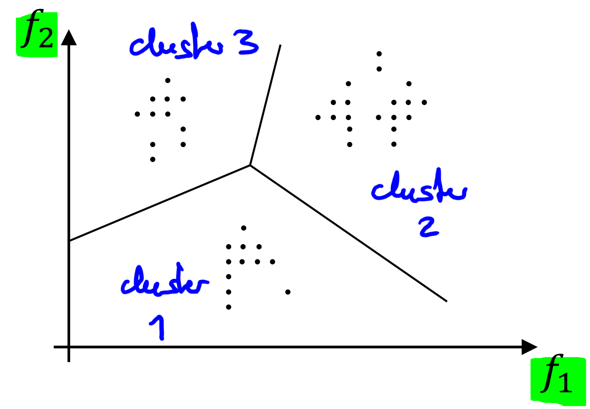
1 **Supervised** segmentation (classification)

- Class prototypes are known (e.g. pdfs)

a priori knowledge available

2 **Unsupervised** segmentation (cluster analysis)

- Neither class prototypes nor number of classes are available



Example: 2 features, 3 classes

Special case: **thresholding**

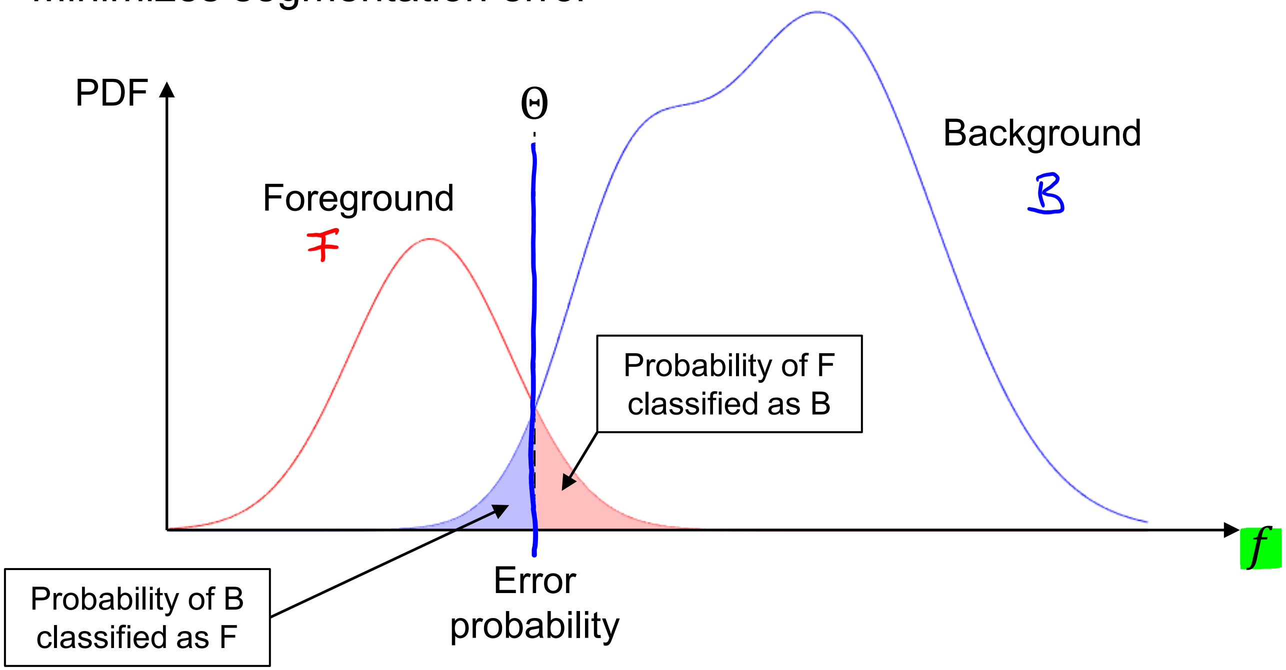
- Only one feature (e.g. luminance)
- Only two classes: object (foreground) vs. background

9.2 Thresholding

Key issue: threshold Θ selection

Supervised thresholding

- Example: MAP (Maximum a Posteriori) estimator
  - Minimizes segmentation error



Unsupervised Thresholding

Idea: find Θ that minimizes **within-class variance**

- Operates directly on gray level histogram
- Assumes bimodal distribution (foreground vs background)

(*) $$\boxed{\sigma^2_{\mathrm{wcv}}(\Theta) = \underbrace{\omega_F(\Theta)}_{\text{prob. of foreground}}\sigma^2_F(\Theta) + \underbrace{\omega_B(\Theta)}_{\text{prob. of background}}\sigma^2_B(\Theta)}$$

where

$$\omega_{F/B}(\Theta) = \frac{N_{F/B}(\Theta)}{N}$$

$$\omega_F(\Theta) + \omega_B(\Theta) = 1$$

Issues

- Requires exhaustive search
- Computing variances can be computationally expensive

Total variance does not change

- Equivalent solution: maximize **between-class variance**

$$\sigma^2_{\mathrm{bcv}}(\Theta) = \overbrace{\sigma^2}^{\text{total var.}} - \sigma^2_{\mathrm{wcv}}(\Theta)$$

Unsupervised Thresholding

Between-class variance

$$\sigma_{\text{bcv}}^2(\Theta) = \sigma^2 - \sigma_{\text{wcv}}^2(\Theta) =$$

$$= \overbrace{\left[\left(\frac{1}{N}\sum_{\boldsymbol{n}} f^2[\boldsymbol{n}]\right) - \mu^2\right]} - \overbrace{\frac{N_F}{N}\left[\left(\frac{1}{N}\sum_{\boldsymbol{n}\in F} f^2[\boldsymbol{n}]\right) - \mu_F^2\right] - \frac{N_B}{N}\left[\left(\frac{1}{N}\sum_{\boldsymbol{n}\in B} f^2[\boldsymbol{n}]\right) - \mu_B^2\right]}^{(*)} =$$

$$= -\mu^2 + \frac{N_F}{N}\mu_F^2 + \frac{N_B}{N}\mu_B^2 + \frac{1}{N}\cancelto{0}{\left(\sum_{\boldsymbol{n}} f^2[\boldsymbol{n}] - \frac{N_F}{N}\sum_{\boldsymbol{n}\in F} f^2[\boldsymbol{n}] - \frac{N_B}{N}\sum_{\boldsymbol{n}\in B} f^2[\boldsymbol{n}]\right)}$$

$$\omega_F(\Theta) + \omega_B(\Theta) = 1$$

$$\omega_F(\Theta)\mu_F(\Theta) + \omega_B(\Theta)\mu_B(\Theta) = \mu$$

$$\boxed{\sigma_{\text{bcv}}^2(\Theta) = \omega_F(\Theta)\omega_B(\Theta)\left(\mu_F(\Theta) - \mu_B(\Theta)\right)^2}$$

only mean required,
no variance !

Unsupervised Thresholding

θ

Search for threshold that maximizes between-class variance

- This automatic threshold selection is known as **Otsu's algorithm**

Efficient recursive computation of $N_{F/B}$ and $\mu_{F/B}$:

$$\left.\boxed{\begin{aligned}
N_F(\Theta+1) &= N_F(\Theta) + n_\Theta \\
N_B(\Theta+1) &= N_B(\Theta) - n_\Theta \\
\mu_F(\Theta+1) &= \frac{\mu_F(\Theta)N_F(\Theta) + \Theta n_\Theta}{N_F(\Theta+1)} \\
\mu_B(\Theta+1) &= \frac{\mu_B(\Theta)N_B(\Theta) - \Theta n_\Theta}{N_B(\Theta+1)}
\end{aligned}}\right\} (**)$$

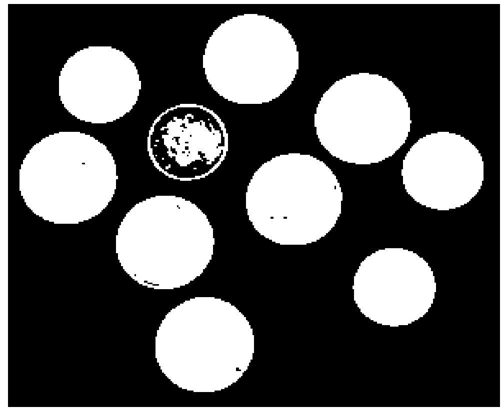
calculate σ_{bcv}^2 using (**)

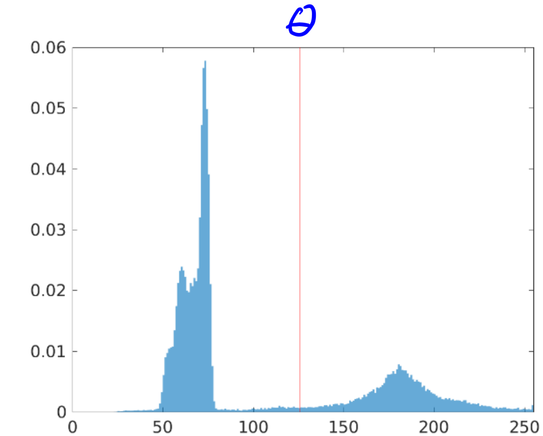
→ select max.

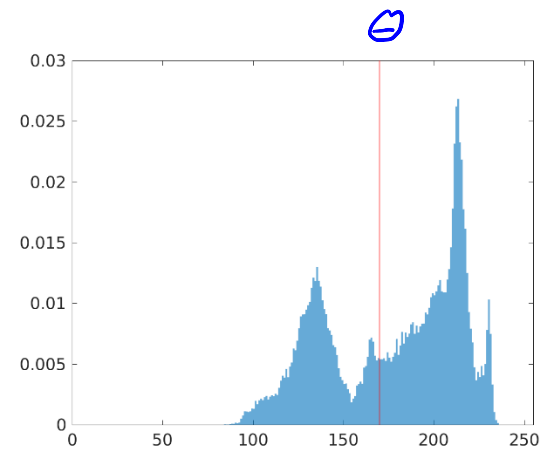
n_Θ - Θ^{th} bin of image histogram (number of pixels with luminance equal to Θ)

Otsu's Thresholding Examples

Original image | Thresholded image | Histograms



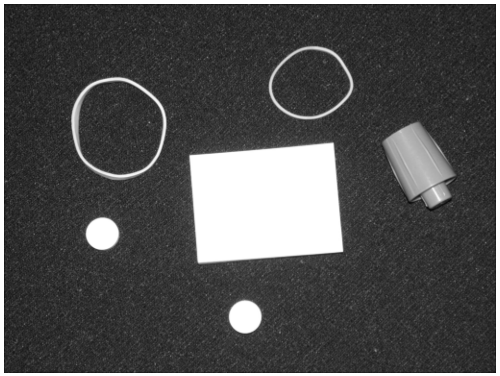


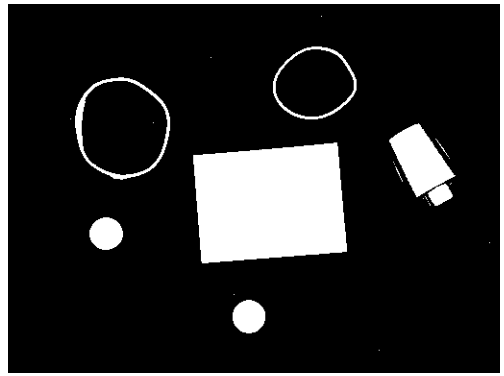


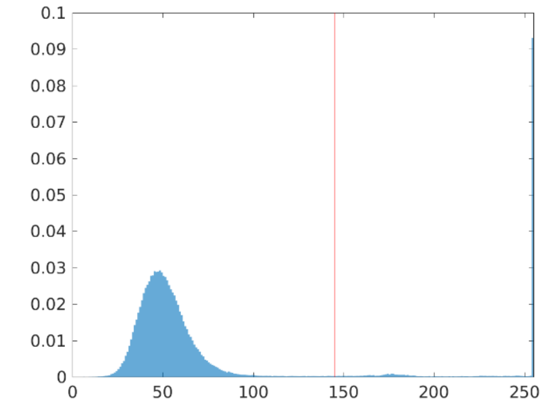
$\Rightarrow$ use morphological filters for smoothing binary mask !

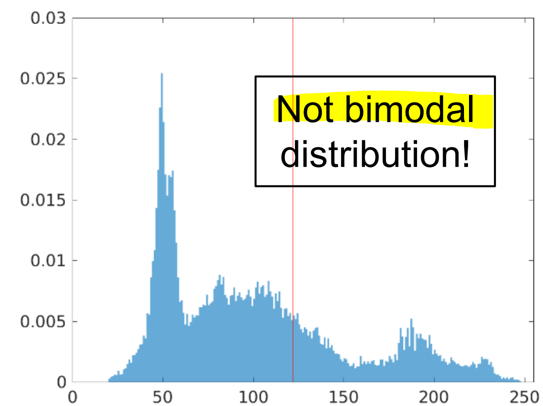
Otsu’s Thresholding Examples

Original image | Thresholded image | Histograms







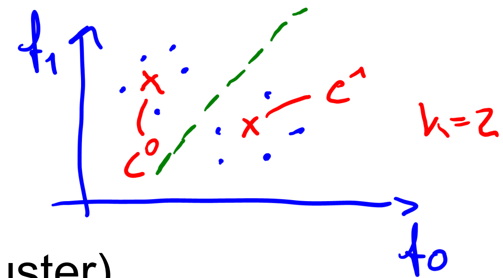


9.3 Classification

For each pixel (m, n) we have L features

- Feature vector $\boldsymbol{f}[m,n] = \left[f_0[m,n], f_1[m,n], \dots, f_{L-1}[m,n]\right]^{\mathrm{T}}$

2D example, $L=2$



We have K clusters (a priori knowledge)

- Cluster centroids $\boldsymbol{c}^{(k)} = \left[c^{(0)}, c^{(1)}, \dots, c^{(K-1)}\right]^{\mathrm{T}}$ (k^{th} cluster)

Each pixel is assigned the **best fitting** cluster S according to

$$\boxed{S[m,n] = \underset{k}{\operatorname{argmin}} \sum_{l=1}^{L} \left| f_l[m,n] - c_l^{(k)} \right|^P = \underset{k}{\operatorname{argmin}} \left\| \boldsymbol{f}[m,n] - \boldsymbol{c}^{(k)} \right\|_P}$$

- If P=2 (Euclidean norm) we have **nearest neighbor classification**

Chroma Keying

movie „Matrix“

Color is more powerful feature than luminance

- 3D space vs. 1D space

Classification with only **two classes**

- Take picture in front of green screen (or blue, or orange…)

c^0

Taken picture

Classification

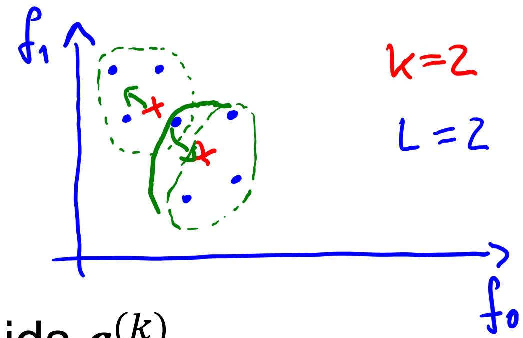
Merging

http://techteacherslog.net http://www.ralph-dte.eu https://i.ytimg.com/vi/4bkENVYNHHs/maxresdefault.jpg

9.4 K-means Clustering (unsupervised case)

Simple **unsupervised** learning algorithm

- Extension of nearest neighbor classification with **unknown** cluster centroids (clustering problem)



Assumption: number of classes K known a priori

Start with arbitrarily chosen set of K cluster centroids $\boldsymbol{c}^{(k)}$

Step 1

1) Nearest neighbor classification (assign each pixel to its nearest cluster)
2) Re-compute cluster centroids

$$\boxed{\boldsymbol{c}_{\text{new}}^{(k)} = \frac{1}{N_k} \sum_{\substack{(m,n)\in \\ \text{cluster } k}} \boldsymbol{f}[m,n] \qquad k = 0,1,\ldots,K-1}$$

Mean value of all features assigned to class k

N_k – number of feature vectors assigned to cluster k

3) Go back to step 1) (or terminate if centroids don't change anymore)

cf. LBG in VQ [IVC lecture]

K-means Clustering Example

L = 1 , only grey level

Original | 2 clusters | 3 clusters | 4 clusters

K-means Clustering

Automatically finds clusters that minimize squared classification error

- Important for **image quantization** (cluster centroids)

Example (8-bit luminance depth, i.e. 256 gray levels)

Original	4 clusters (2 bits)	8 clusters (3 bits)	16 clusters (4 bits)

9.5 Bayesian Classification

Based on minimization of Bayes risk $R(\hat{S})$

- Risk defined as expected cost value

$$R(\hat{S}) = E[C(\hat{S}, S)] = \iint_{S,\boldsymbol{f}} C(\hat{S}, S) P(S, \boldsymbol{f}) dS d\boldsymbol{f} = \iint_{S,\boldsymbol{f}} C(\hat{S}, S) \underbrace{P(S|\boldsymbol{f}) P(\boldsymbol{f})}_{\text{Bayes rule}} dS d\boldsymbol{f}$$

estimated segmentation

true segmentation

Feature probability $P(\boldsymbol{f})$ ("observed signal") is class-independent

- The same for all classes → no influence on minimization

$$R(\hat{S}|\boldsymbol{f}) = \int_S C(\hat{S}, S) P(S|\boldsymbol{f}) dS$$

Bayesian classification

$$\boxed{\hat{S} = \underset{S}{\operatorname{argmin}}\, R(\hat{S}|\boldsymbol{f}) = \underset{S}{\operatorname{argmin}} \left[\int_S C(\hat{S}, S) P(S|\boldsymbol{f}) dS \right]} \quad (*)$$

Maximum a Posteriori (MAP)

Cost function: $C(\hat{S},S) = 1 - \delta(\hat{S},S)$

$\delta(\hat{S},S) = \begin{cases} 1 & \text{if } \hat{S}=S \\ 0 & \text{else} \end{cases}$ $\quad c(\hat{S},S) = \begin{cases} 0 & \text{if } \hat{S}=S \\ 1 & \text{if else} \end{cases}$

Bayes risk: $R_{\mathrm{MAP}}(\hat{S}|\boldsymbol{f}) \overset{(*)}{=} \int_S \left(1-\delta(\hat{S},S)\right) P(S|\boldsymbol{f})dS = 1 - \underbrace{P(\hat{S}|\boldsymbol{f})}_{\to \max}$ $\to \min$

Probability a posteriori $P(\hat{S}|\boldsymbol{f})$

- Probability of assigning $\hat{S}$ given feature $\boldsymbol{f}$ (usually not known)

Bayes' rule: $\boxed{P(S|\boldsymbol{f}) = \dfrac{P(\boldsymbol{f}|S)P(S)}{p(\boldsymbol{f})}}$

does not affect maximization

Thomas Bayes
(1701-1761)

MAP classification: $\boxed{\hat{S}_{\mathrm{MAP}} = \underset{S}{\operatorname{argmax}}\, P(S|\boldsymbol{f}) = \underset{S}{\operatorname{argmax}}[P(\boldsymbol{f}|S)P(S)]}$

Bayesian Classification

Difference between MAP and nearest neighbor classification

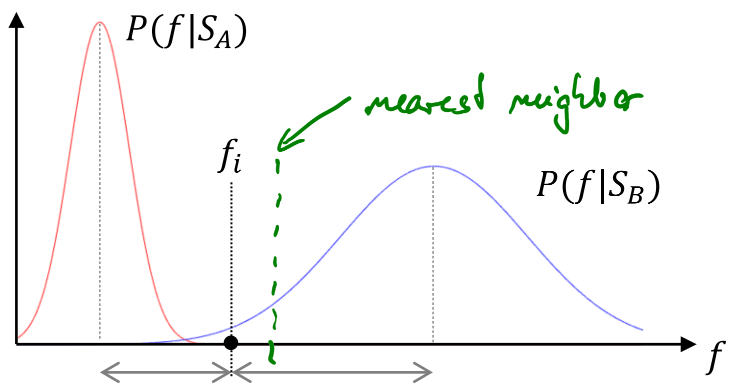
- Decision not necessarily for the nearest cluster

Instead, the following is considered

1 - Clusters a priori probability ($P(S)$)
    - Which cluster is more likely?

2 - Feature likelihood within a specific class $P(\boldsymbol{f}|S)$
    - Given a cluster, which feature vector is more likely?

Example (1D)



Nearest neighbor assigns class A (it is closer)

MAP assigns class B (it is more likely)

Bayesian Classification

If classes are equally probable then MAP is reduced to **Maximum Likelihood (ML)** classifier

$$\boxed{\hat{S}_{\mathrm{ML}} = \underset{S}{\operatorname{argmax}}[P(\boldsymbol{f}|S)\cancel{P(S)}] = \underset{S}{\operatorname{argmax}}\, P(\boldsymbol{f}|S)}$$

constant

Other cost functions are also possible

Disadvantage of classification and clustering methods discussed so far

- Operate on features only (no spatial relation between pixels considered)

9.6 Region-Based Segmentation

Idea: Incorporate knowledge about topological structure of partition (especially neighbor relations)

Region: Group of connected pixels with similar properties

(=> still incorporate features)

Principles: 1 Similarity

- Feature differences / variance

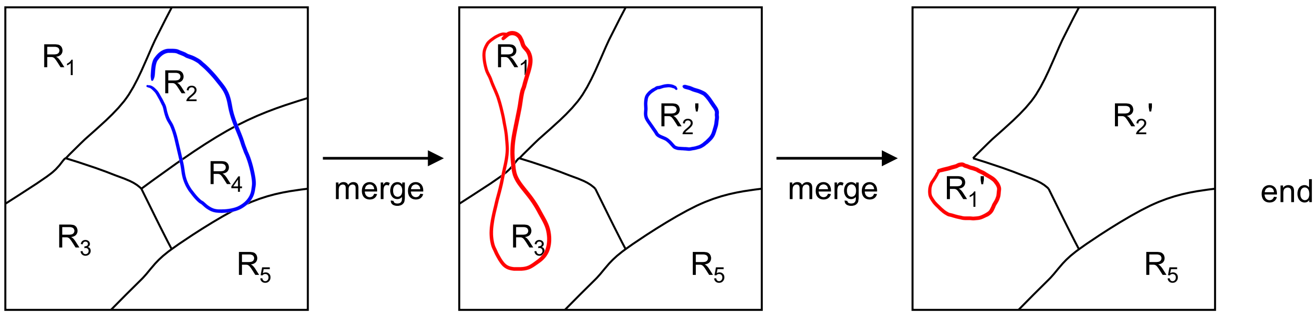
2 Spatial proximity new here!

- Euclidean distance
- Compactness of a region

Region Growing

Starts with a set of **seed** points

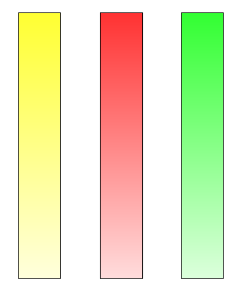
- Expansion (growth) by pixels with similar features
- Adjacent similar regions are merged together



→ bottom-up segmentation

Properties

- Fast and conceptually simple
- Sensitive to noise
- Gradient problem →



Similarity Measures for Two Regions

Given mean μ and variance σ^2 of two neighboring regions R_i and R_j

1 **Absolute deviation** of mean value

$$d(R_i, R_j) = \left|\mu(R_i) - \mu(R_j)\right|$$

- Simple to calculate, does not account for region variances

2 **Variance coherence**

$$d(R_i, R_j) = \sigma^2(R_i \cup R_j) - \frac{\sigma^2(R_i) + \sigma^2(R_j)}{2}$$

(μ is included implicitly)

- Extensible to higher order statistics

3 **Likelihood ratio**

$$d(R_i, R_j) = \frac{\sigma^2(R_i \cup R_j)^{N_i + N_j}}{\sigma^2(R_i)^{N_i} \cdot \sigma^2(R_j)^{N_j}}$$

(includes reliability of estimates)

- Considers region size N

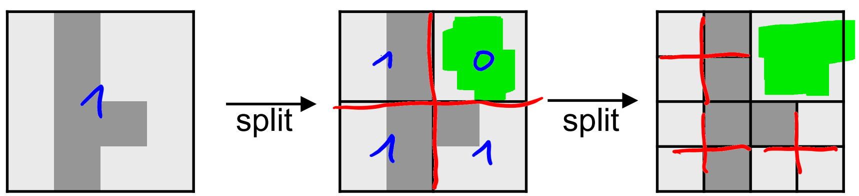
Region Splitting

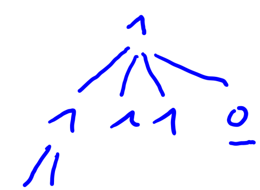
Split image into disjoint regions

- Check each region for homogeneity, if not homogeneous keep splitting

Example: quad-tree decomposition

quadtree





→ top-down segmentation

Problems:

- How to optimally split a region into homogeneous sub-regions?
- Requires knowledge about number of sub-regions and location of region boundaries (e.g. by edge detection)

Split & Merge Segmentation

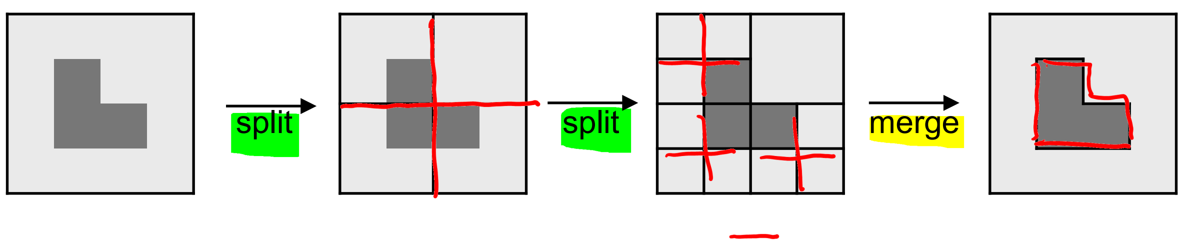
Combination of agglomerative and divisive region operations

Splitting (e.g. quad-tree)

- Keep splitting until all blocks fulfill homogeneity criterion

Merging

- Merge all neighboring block which are sufficiently similar



also used in compression for video data

Disadvantage: region borders often exhibit "staircase" character

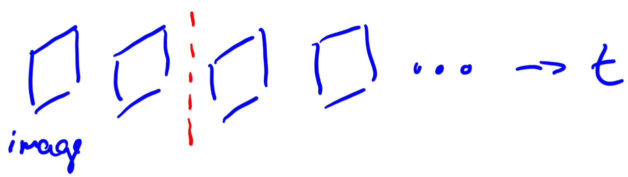
9.5 Temporal Segmentation of Video

Detection of scene cuts and smooth scene transitions (fading)

- Usually as preprocessing for video structuring applications (news server, media abstraction for messaging)

Scene cut assumption

- Content changes significantly between two consecutive frames or over a number of frames



Analysis of suitable (global) image feature over time

- Color distribution, motion, texture descriptors, etc.
- Often audio track is analyzed in parallel to improve segmentation results (e.g. silence detection)

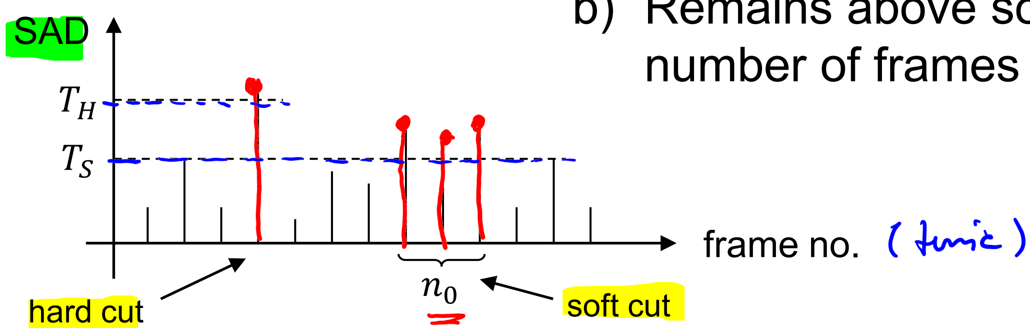
Shot Detection Using Feature Histograms

Shot detection

- Detect scene cuts and segment video into a number of temporal **consistent** video sequences (**shots**)

Histogram of suitable image feature (e.g. color)

- Analyze sum of absolute histogram differences between subsequent video frames
- Scene cut detected if:
  a) Sum excedes hard threshold T_H (hard cut)
  b) Remains above soft threshold T_S for a certain number of frames n_0 (soft cut or fading)



Sensitive to (sudden) illumination changes → use additional features

Shot Detection Using Motion Analysis

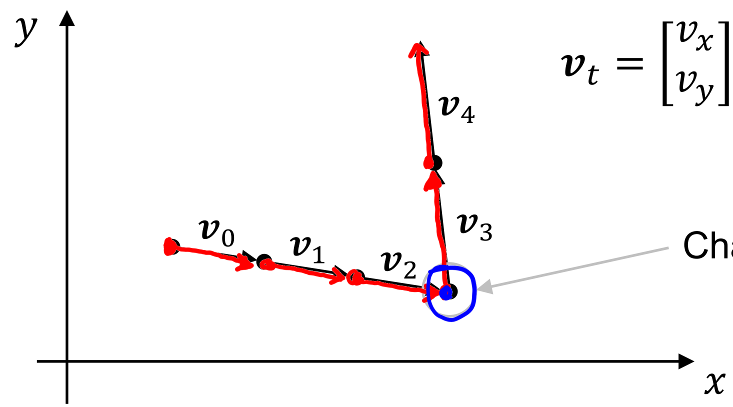
Change in global motion usually corresponds to new scene

- E.g. start and end of camera pan (= horizontal movement)

Detection of shot boundaries

- Significant change in motion trajectory

Example:



$$v_t = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

Global translational motion between consecutive frames

Can be improved by considering higher order motion

- Rotation, zoom, affine models, etc.