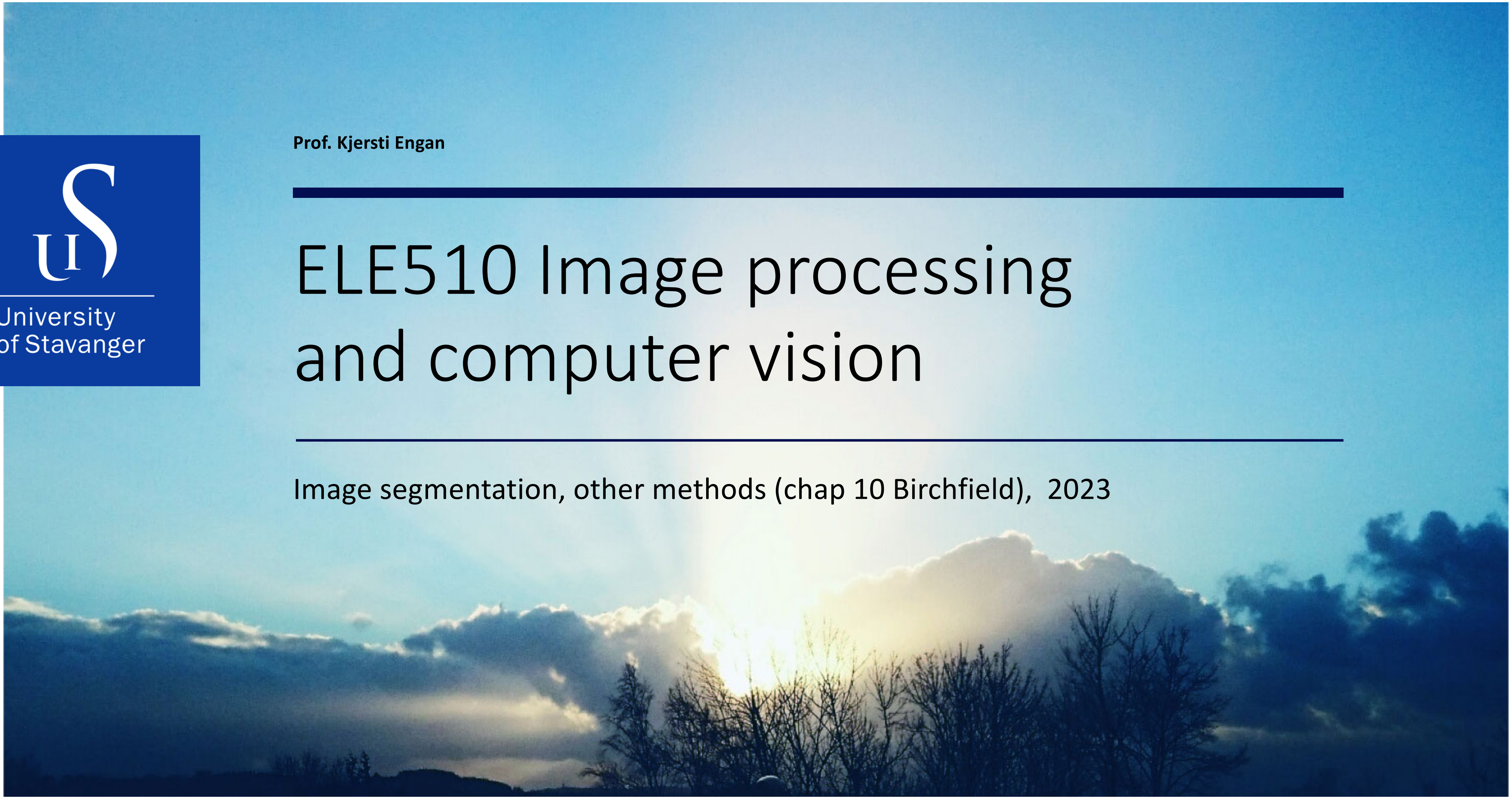
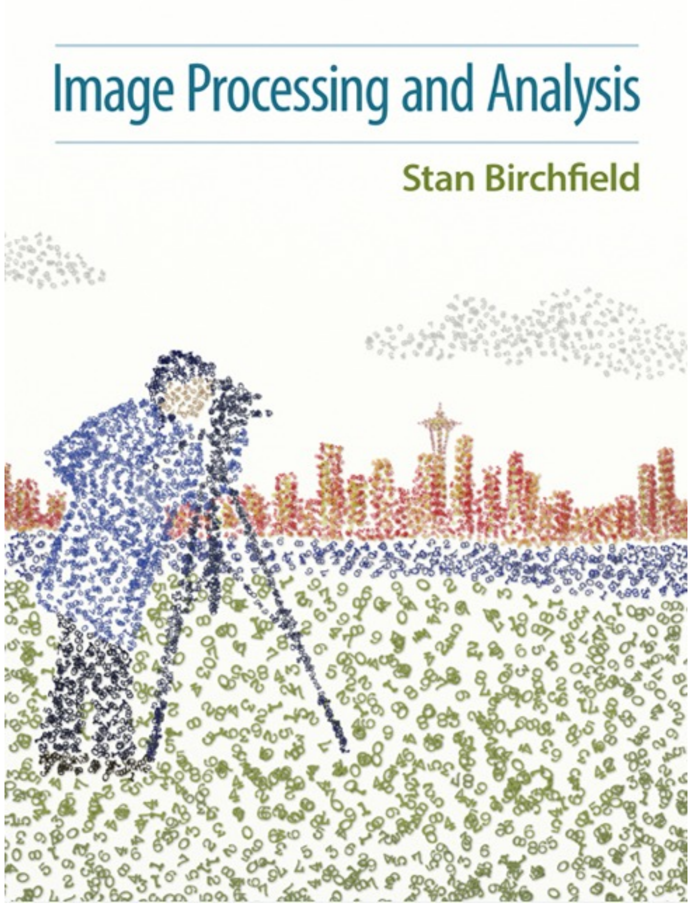


University of Stavanger

Prof. Kjersti Engan

ELE510 Image processing and computer vision

Image segmentation, other methods (chap 10 Birchfield), 2023

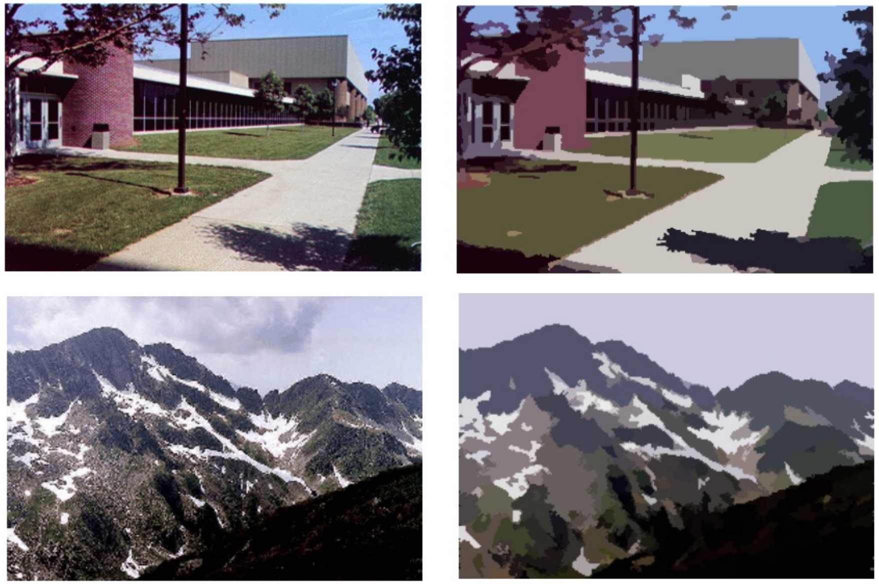


Parts in course presentations is material from Cengage learning. It can be used for teaching, and it can be share it with students on access controlled web-sites (Canvas) for use in THIS course. **But it should not be copied or distributed further in any way** (by you or anybody).

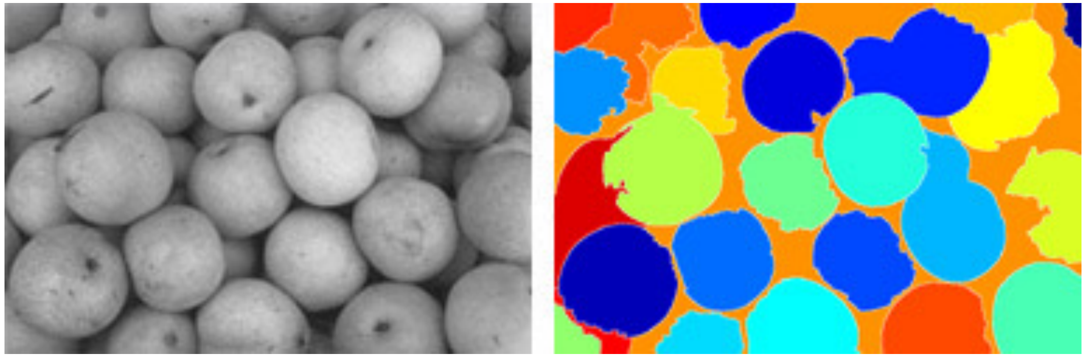
Segmentation – other* methods

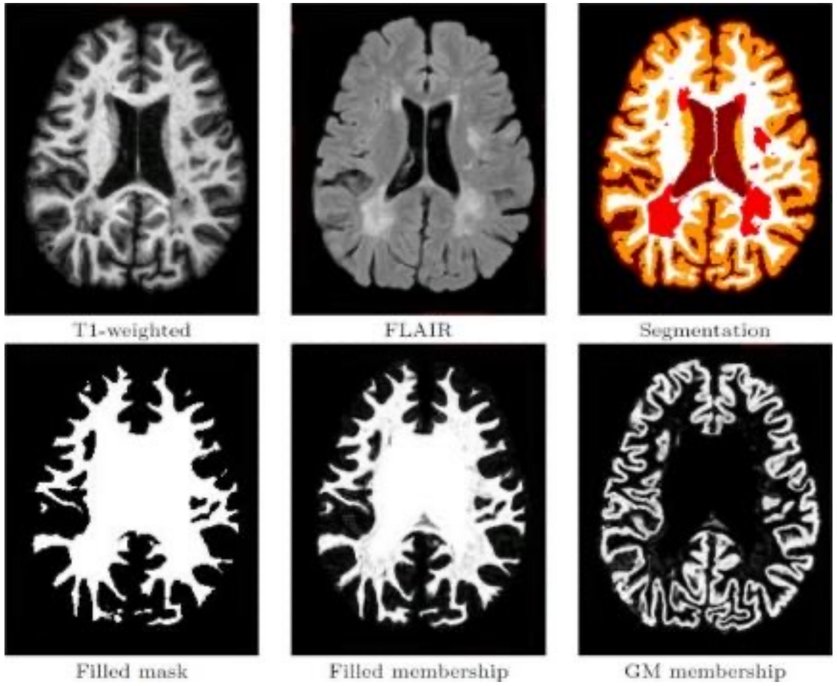
Three points from the topic:

1. Segmentation by region growing
2. Segmentation by watershed
3. Segmetnation by mean-shift



* - other than thresholding





T1-weighted

FLAIR

Segmentation

Filled mask

Filled membership

GM membership

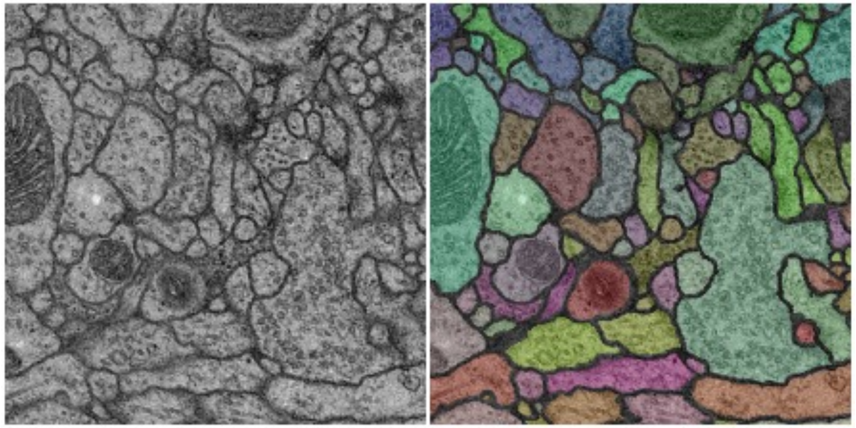
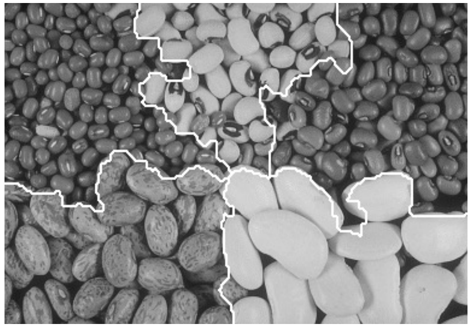


Image segmentation methods, examples

- Thresholding
  - Global / local
  - Otsu’s method
- Region based
  - Region growing
  - Split and merge
- Morphological segmentation
  - Watershed
- Mean-shift segmentation
- Graph based methods *(not a part of this course)*
- Deformable models (*not a part of this course*)
  - Level set
  - Active contours (snakes)

- And more …….

(10.3) Other image segmentation methods

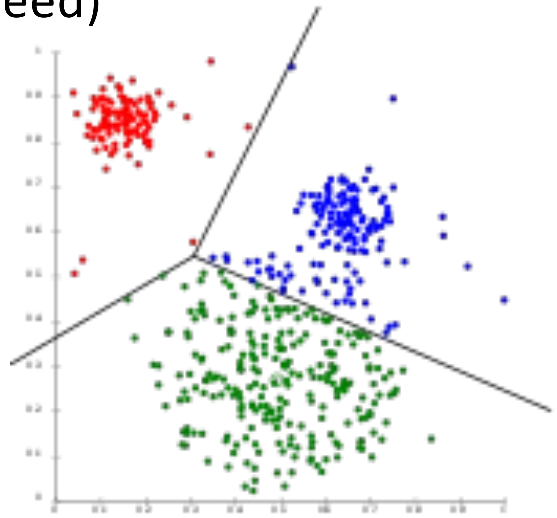


- Sometimes thresholding algorithms will not work.
  - Ex. Segmenting a chessboard pattern from white background,
  - segment texture regions with similar intensities.
- It is often important to take the neighbourhood into consideration.

- Possible approach: associate each pixel (or small region) with MORE than the intensity value, for example a feature vecotor containing the intensity level as one feature, local texture descriptors, edges (from gradient image) as others etc.
- Feature vectors can be fed to a machine learning/classifier algorithm and clustered/classified as objects/background/belonging to diffent classes.

Segmentation and clustering

- Unsupervised image segmentation is a clustering problem
- Clustering = “group similar things”
- Culstering approaches
  - Agglomerative clustering / region growing (start with a seed)
  - K-means (specify number of cluster in advance)
  - Mean – shift (computational expensive)



Splitting and merging of regions

- Let us define the segmentation of an image *I* as a set of regions R_1, …, R_n such that every pixel is in exactly one region:

$$
\begin{aligned}
I &= \bigcup_{i=1}^{n} \mathcal{R}_i && \text{(covers entire image)} \\
\mathcal{R}_i \cap \mathcal{R}_j &= \emptyset \quad \text{for all} \quad i \neq j && \text{(non-overlapping)}
\end{aligned}
$$

- In the classic view of the problem of segmentation, a predicate $h(R_i)$ measures the **homogeneity** of a region.

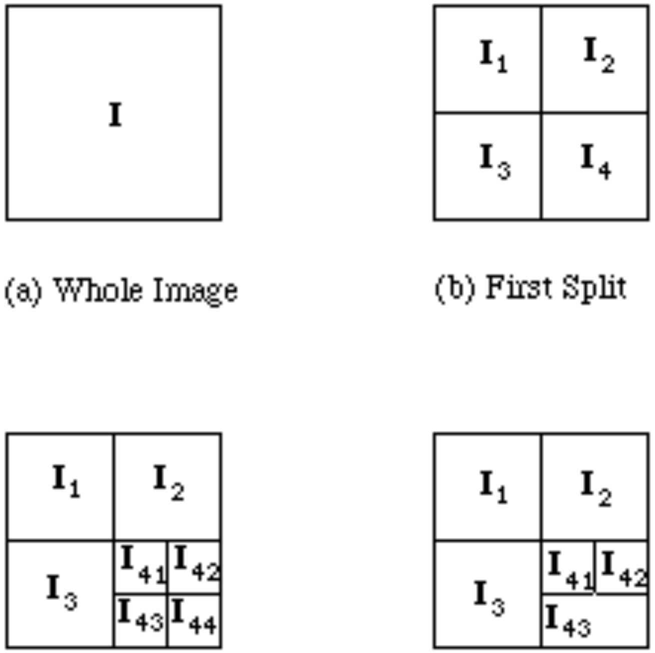
$$
\begin{aligned}
h(\mathcal{R}_i) &= \text{TRUE for all } i && \text{(each region is homogeneous)} \\
h(\mathcal{R}_i \cup \mathcal{R}_j) &= \text{FALSE for all adjacent } \mathcal{R}_i, \mathcal{R}_j, i \neq j && \text{(adjacent regions are different)}
\end{aligned}
$$

- **Merging** algorithms (agglomerative clustering) begin with each pixel as a separate region, then recursively merge adjacent regions whenever they are similar to each other.
- **Splitting** algorithms (divisive clustering) begin with the entire image as a single region, then recursively split regions whenever they are found to be nonhomogeneous.

- The “Split-and-merge” algorithm combines these two ideas.

Split-and-merge

- Start with the whole image as one region, or a predefined block grid.
- Test if a feature's variation in the block is greater than a threshold (homogeneity h(Ri)< threhold). If yes, split the block into four (quad-tree splitting).
- Continue testing until no more regions are splitted.
- Test if adjacent regions have feature value in the same range. f yes -> merge into one region. The order of which regions are tested first will affect the result.
- Continue until no more regions are merged.


I
I_1 I_2 I_3 I_4
I_1 I_2 I_3 I_{41} I_{42} I_{43} I_{44}
I_1 I_2 I_3 I_{41} I_{42} I_{43}

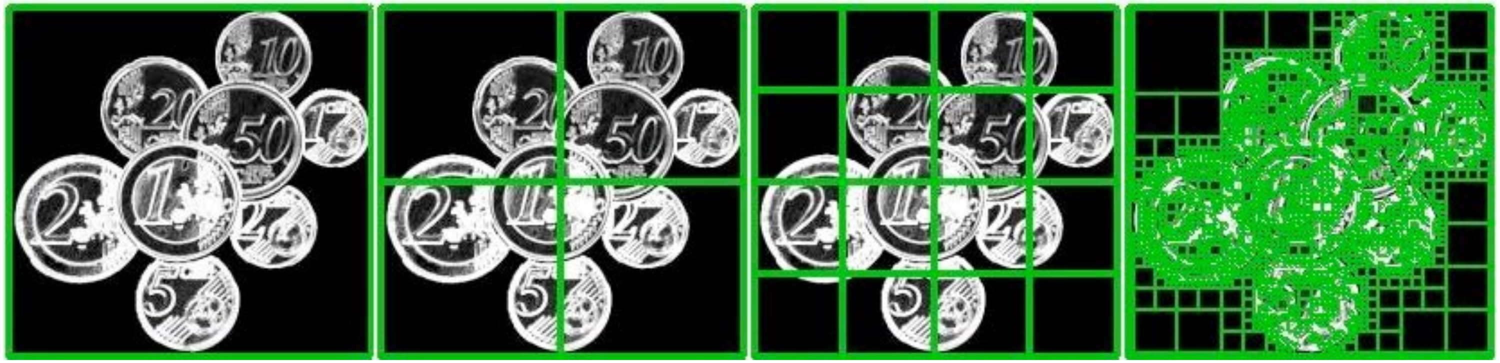
(a) Whole Image

(b) First Split

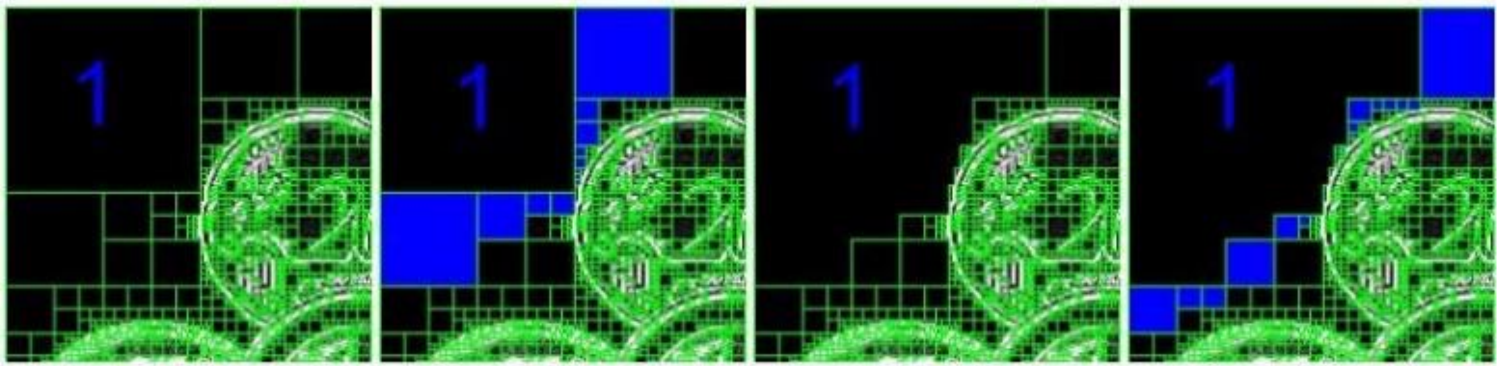
(c) Second Split

(d) Merge

Split and merge



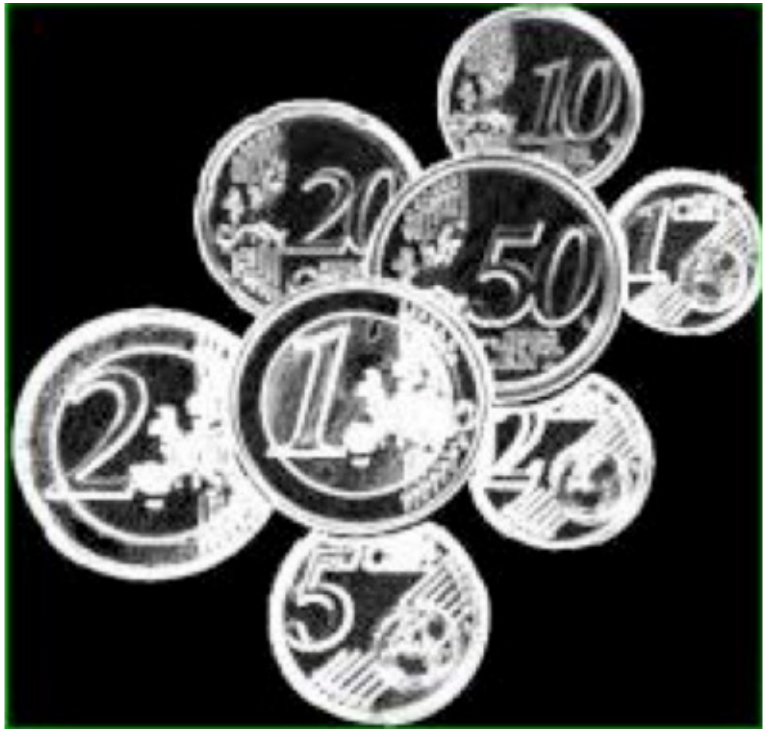
Split (in four) as long as feature variance inside block is large

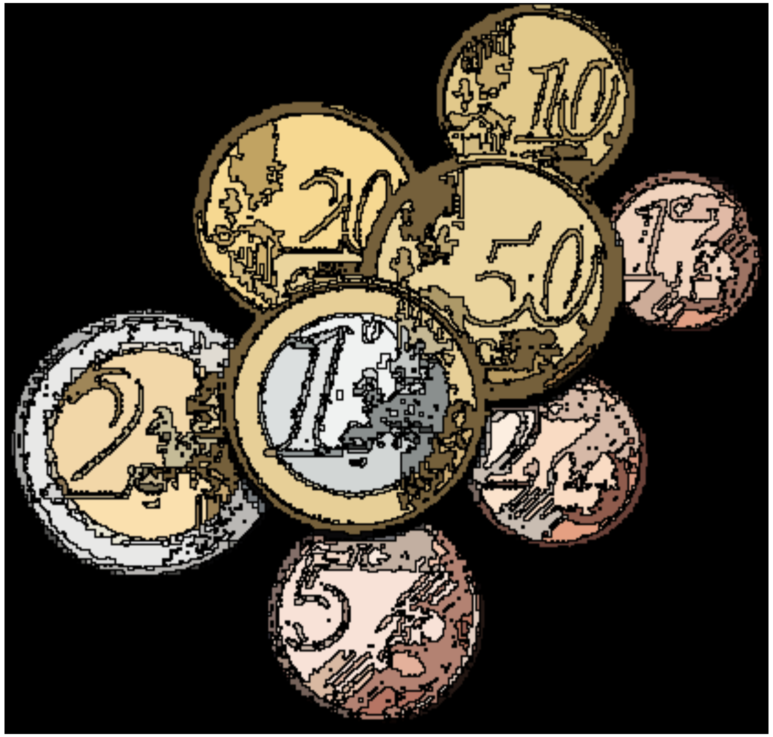


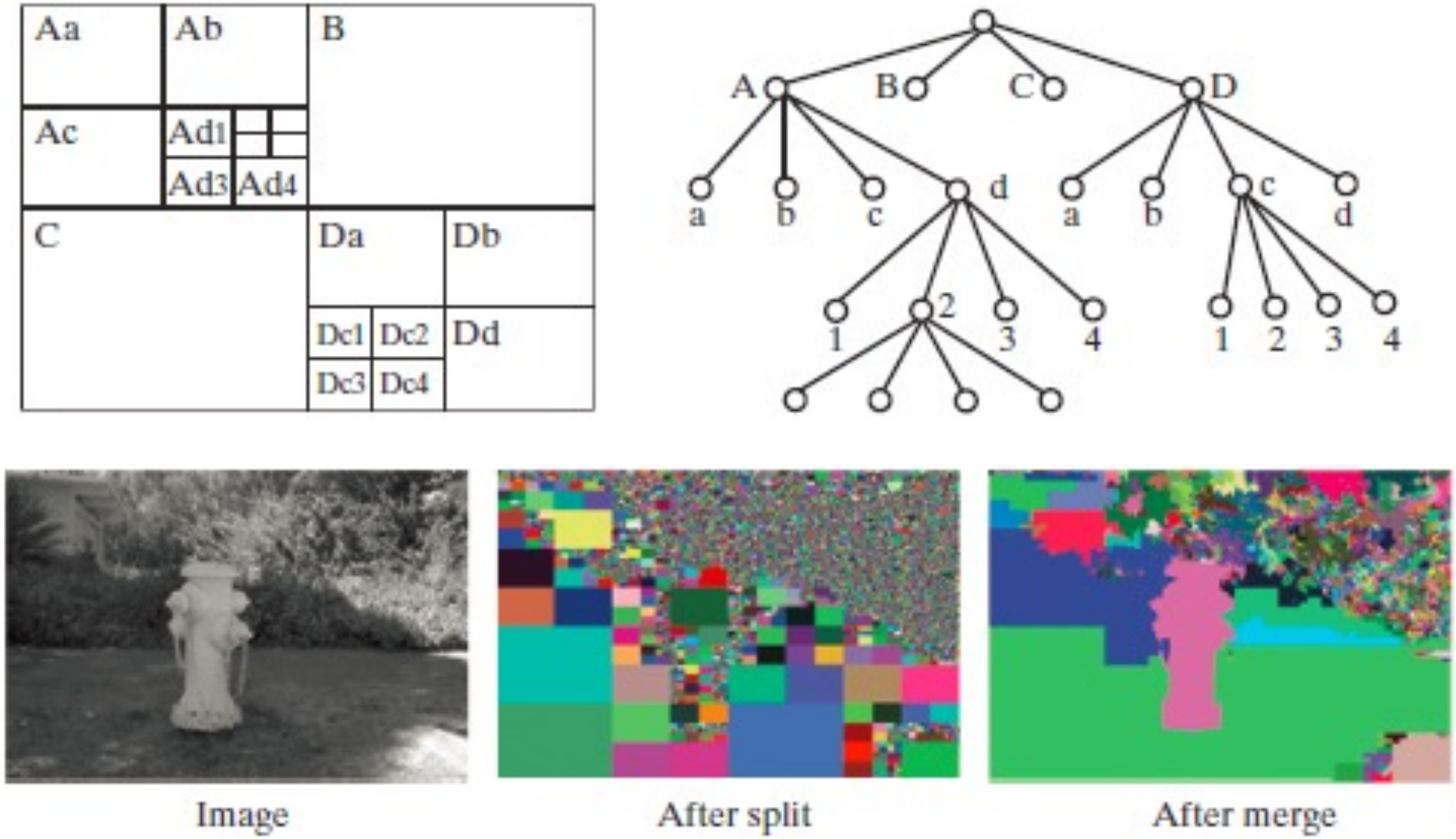
Merge adjacent blocks with similar value of feature (mean intensity)

From: developpez.com

Final result, split and merge







Image

After split

After merge

Stan Birchfield

Figure 10.22 Top: The quad-tree data structure used in splitting. Bottom: The split-and-merge algorithm applied to a grayscale image. The algorithm is able to find the fire hydrant and most of the ground, though it oversegments the textured background.