Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Computer Vision
Segmentation

WS 2019/2020

Prof. Dr. Simone Frintrop

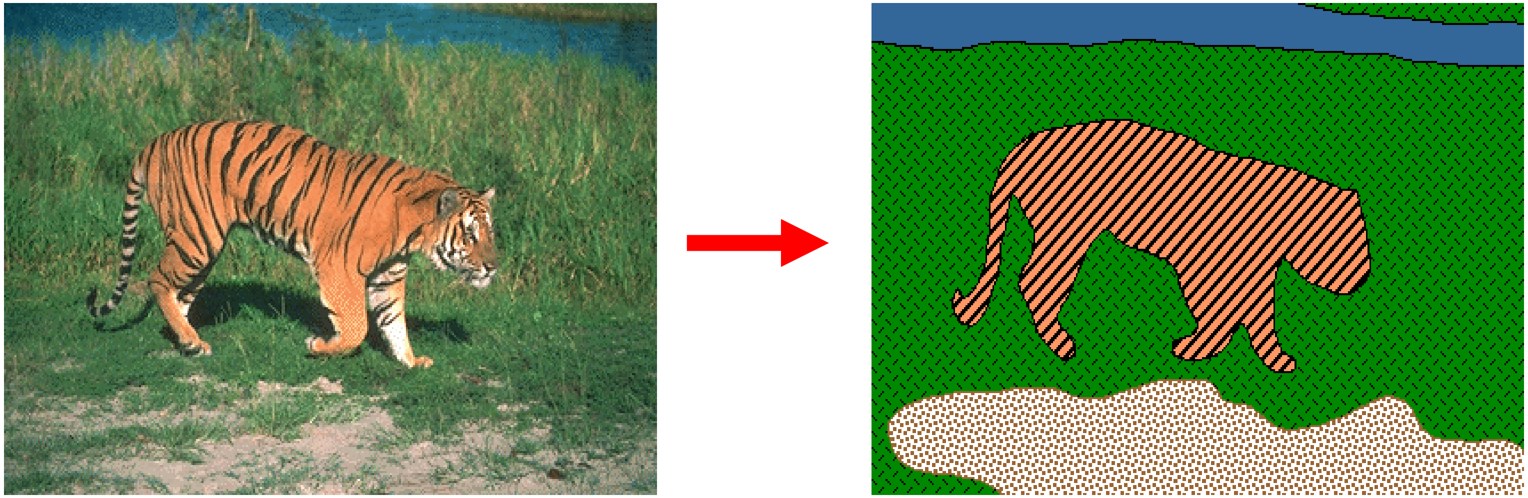
Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Content

- Segmentation: Motivation, definition, and applications
- Region Growing
- K-means for segmentation
- Superpixels: SLIC
- Adaptive Seeding for Superpixels and Supervoxels

What is Segmentation?

- Segmentation subdivides an image into its constituent regions (Gonzalez/Woods)
- Segmentation belongs to the oldest and hardest problems in computer vision



- Q: Why do we segment images?
- A: To obtain a simpler representation that is easier and faster to analyze

[Images: Kristen Grauman]

What is Segmentation?

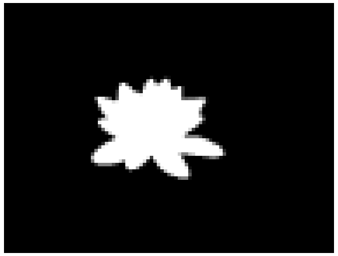
According to Klette 2014:

- Segmentation partitions an image into regions called

Segments S_i, $i = 1, \ldots, n$, such that

1. $S_i \neq \emptyset$, for any $i \in \{1, \ldots, n\}$
2. $\bigcup_{i=1}^{n} S_i = \Omega$
3. $S_i \cap S_j = \emptyset$, for all $i, j \in \{1, \ldots, n\}$ with $i \neq j$

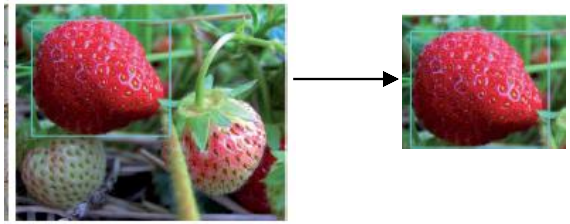
Note: Definition does not fit to all segmentation methods, because sometimes the regions can overlap. It fits especially well to superpixel methods.



Why segmentation?

Segmentation as preprocessing for many applications:

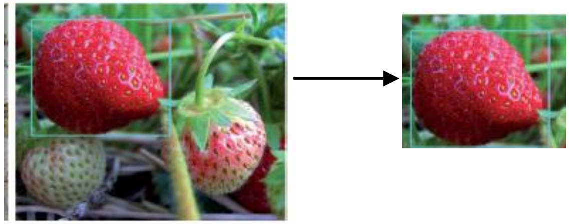
- Photo editing:



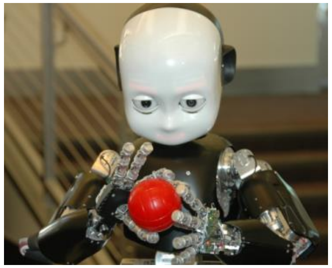
Why segmentation?

Segmentation as preprocessing for many applications:

- Photo editing:



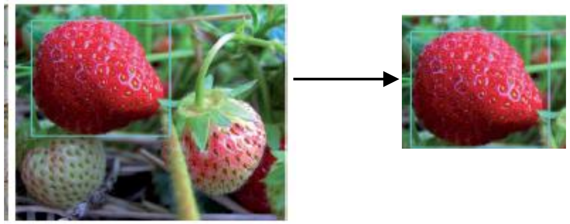
- Robotics:



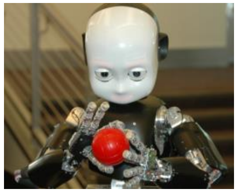
Why segmentation?

Segmentation as preprocessing for many applications:

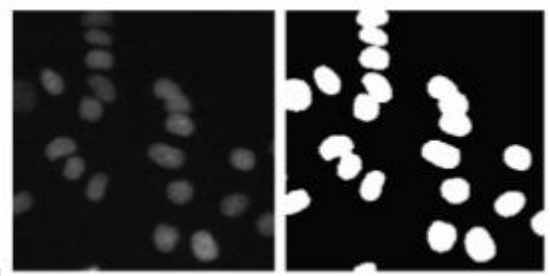
- Photo editing:

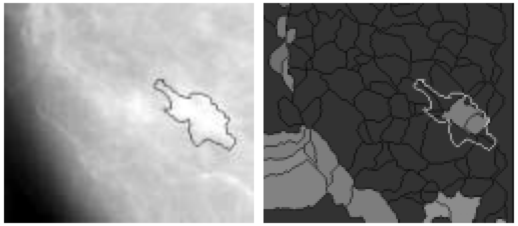


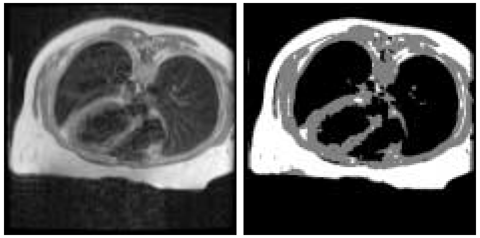
- Robotics:



- Medical image processing (e.g. X-Ray, MRI, CT, …):
  - Diagnosis (e.g. tumor detection)
  - Computer-aided surgery
  - Quantification (number of cancer cells, size of tumor)

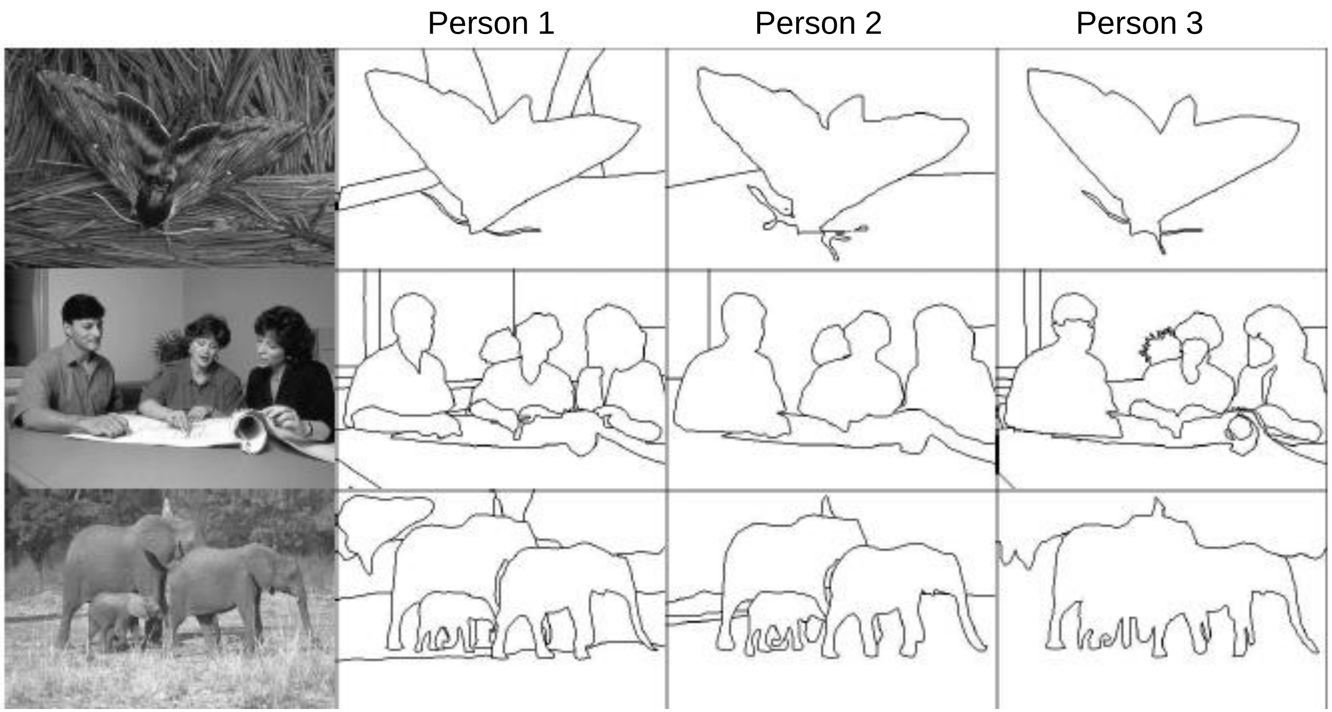






What is "good"?

- Q: How do we know if a segmentation is "good"?
- A: Compare with human performance:
- **The Berkeley Segmentation Dataset and Benchmark** (https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/)

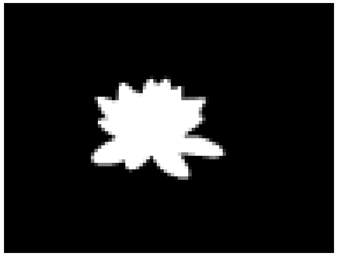


1000 images
12000 segmentations
30 subjects

[Martin et al. ICCV 2001]

Level of Detail

- The required level of detail depends on the application

- Is this what we want?
- Or should every leaf be a segment?

- Usually, some parameters can be adapted to obtain more or less segments

Over- vs Undersegmentation

- Oversegmentation:
  get more segments than desired
- Undersegmentation:
  get less segments than desired
- It is very hard (and often impossible)
  to obtain the optimum between the two


Human segmentation

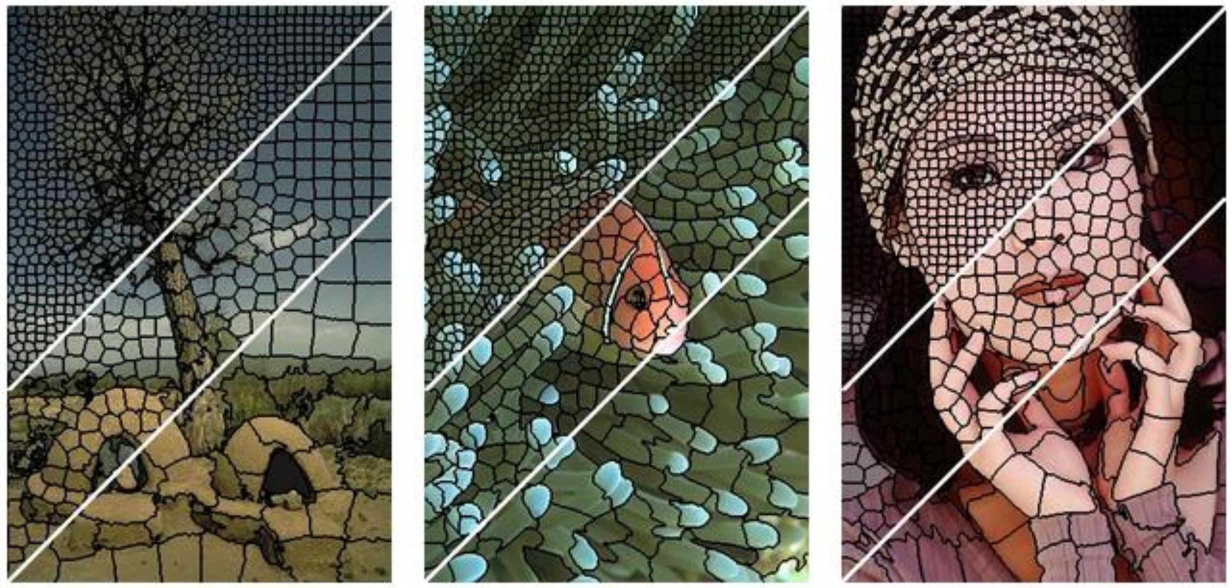

Oversegmentation


Undersegmentation

[Sigut et al. 2015]

Superpixels

- Superpixels are a different name for the segments obtained by a segmentation algorithm
- Usually, they
  - come from a strong over-segmentation (they resemble “large pixels”)
  - have a coherent, perceptually uniform appearance (color, texture)
- Example: SLIC superpixels (Achanta et al. 2010):



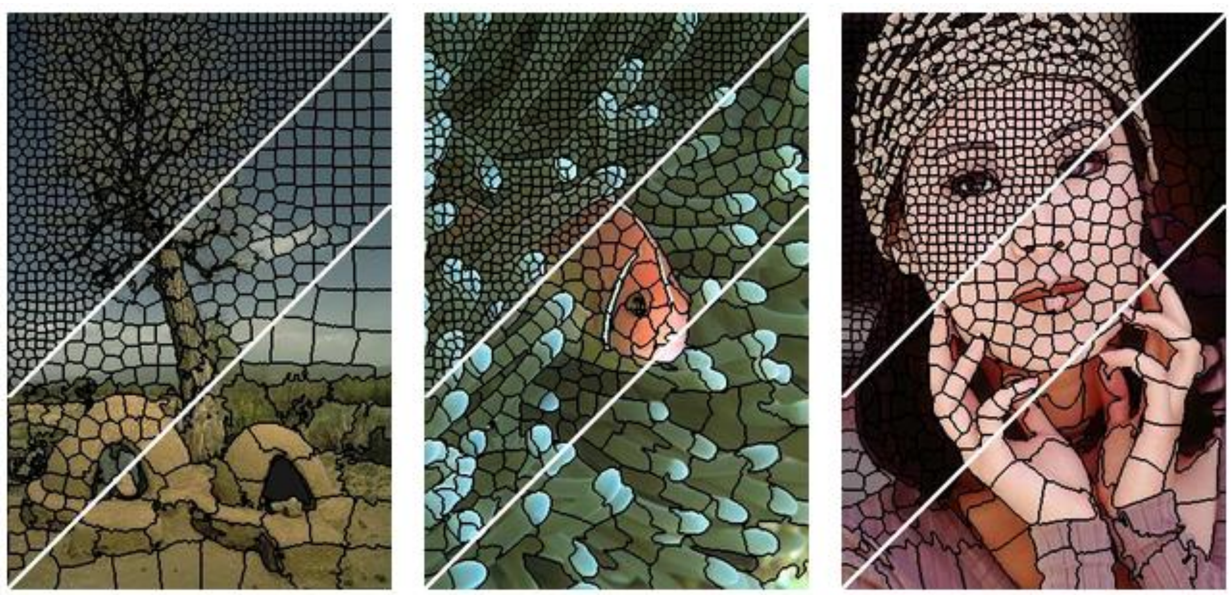
[Achanta 2010]

Superpixels

Advantage of superpixels:

Reduce computational complexity.

Much less superpixels than pixels to operate on



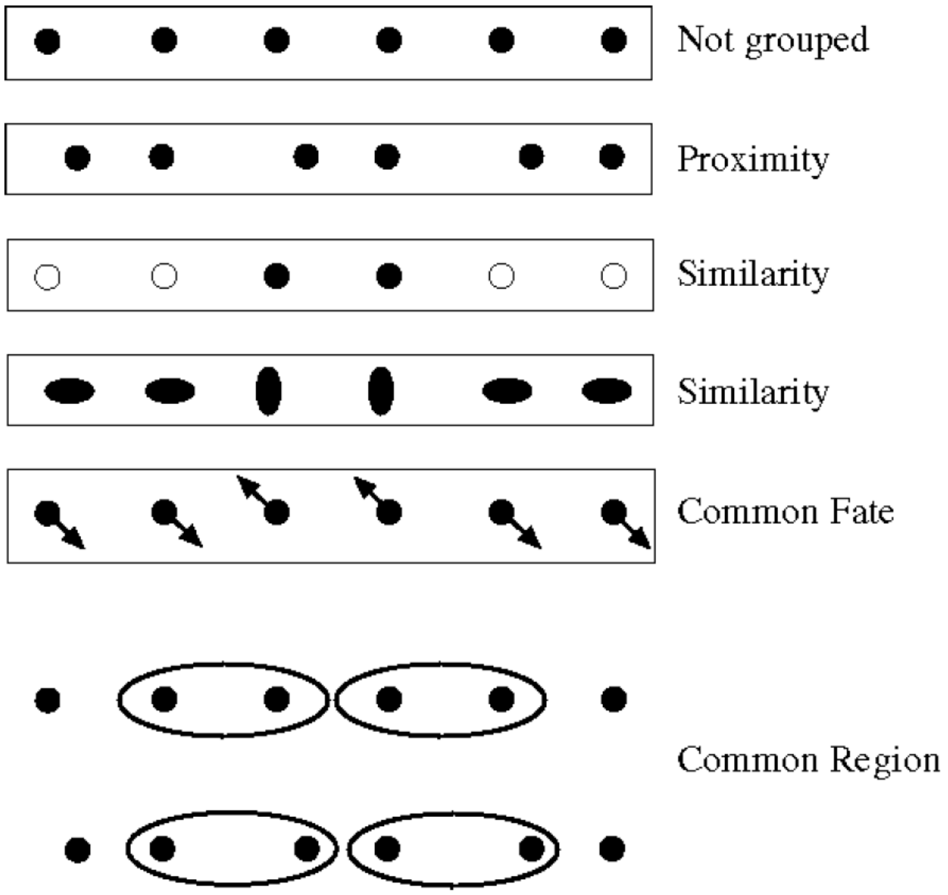
[Achanta 2010]

Segmentation in Human Vision

Human Vision:

- **Segmentation processes** exist on all levels of the visual system. They bundle parts of the visual input to perceptually coherent regions
- **Gestalt principles** group regions to larger regions

Gestalt Factors



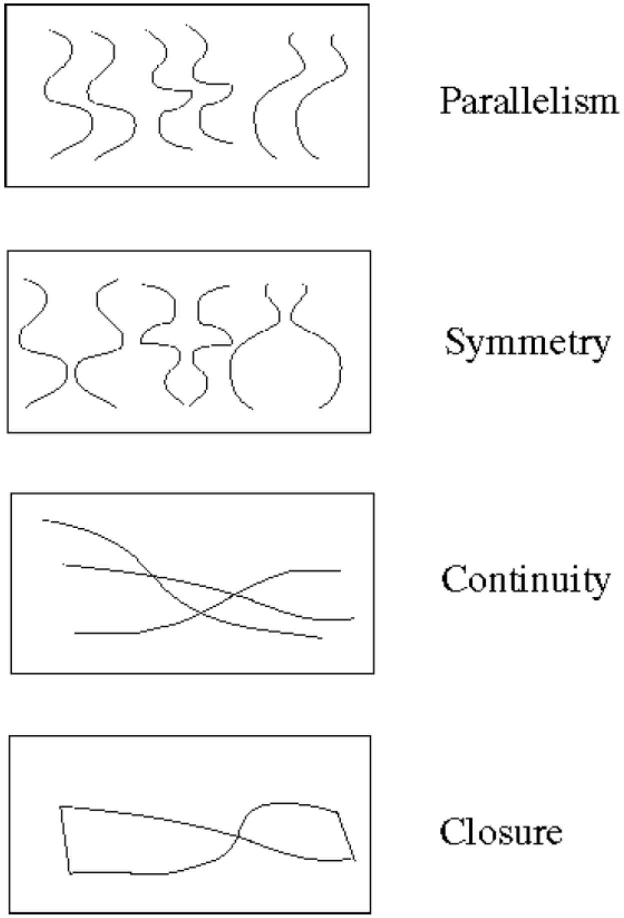
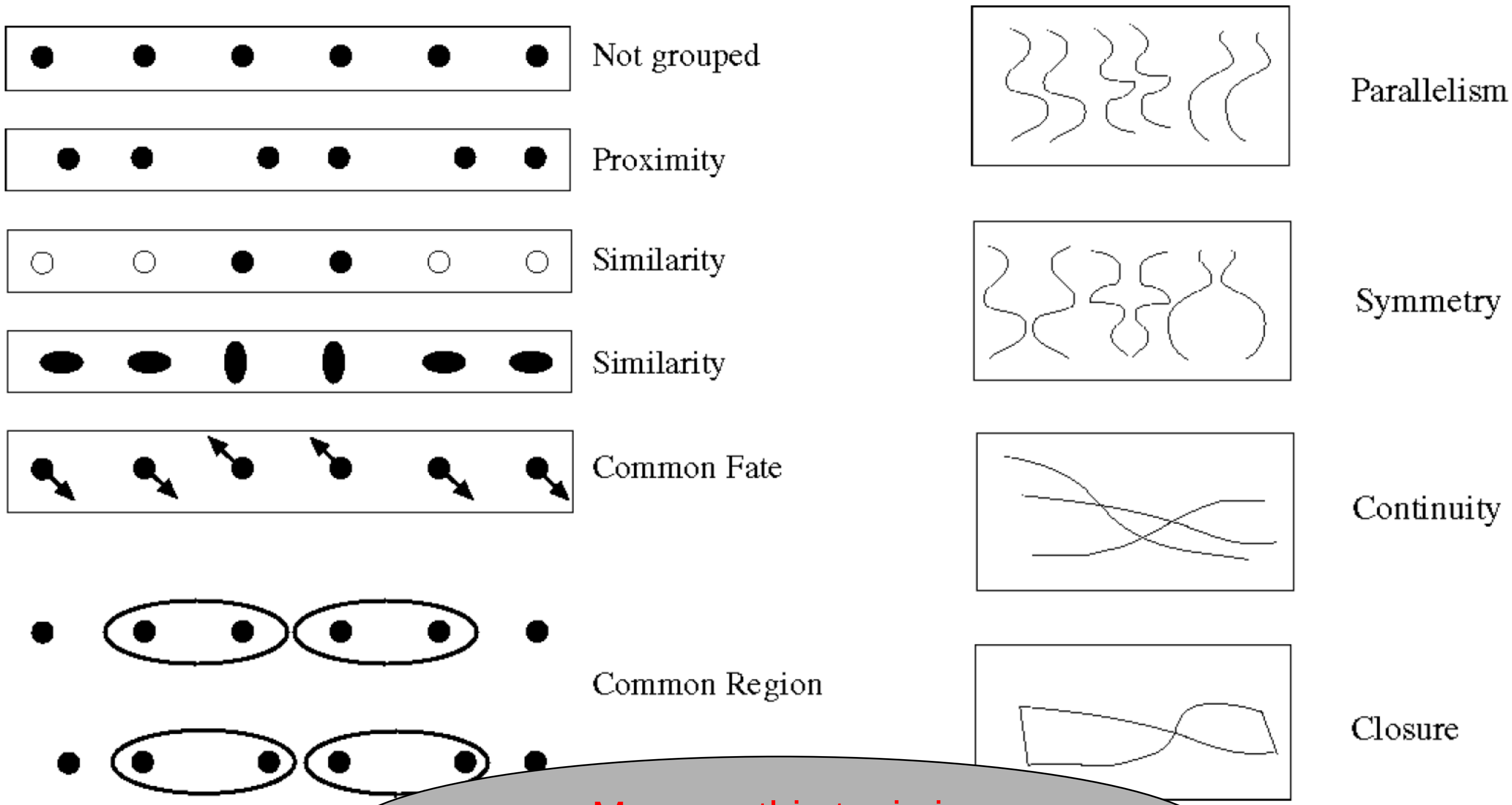


Image source: Forsyth & Ponce

Gestalt Factors



More on this topic in
“Computer Vision 2” Lecture

Image source: Forsyth & Ponce

Basics of Segmentation

- **Key elements in segmentation: proximity and similarity:**
  base segmentation on similarity of neighboring pixels
- Similarity wrt: intensity, color, texture, depth, …

- Pixels within one segment are more similar to each other than to pixels from neighboring segments
- Strong dissimilarities between neighboring segments (edges) are indicators for borders

Many segmentation approaches exist:

1 Applications
2 Thresholding
3 Clustering methods
4 Compression-based methods
5 Histogram-based methods
6 Edge detection
7 Dual clustering method
8 Region-growing methods
9 Partial differential equation-based methods
- 9.1 Parametric methods
- 9.2 Level set methods
- 9.3 Fast marching methods

10 Variational methods
11 Graph partitioning methods
- 11.1 Markov random fields
  - 11.1.1 Supervised image segmentation using MRF and MAP
  - 11.1.2 Optimization algorithms
    - 11.1.2.1 Iterated conditional modes/gradient descent
    - 11.1.2.2 Simulated annealing (SA)
    - 11.1.2.3 Alternative algorithms
  - 11.1.3 Unsupervised image segmentation using MRF and expectation maximization
  - 11.1.4 Disadvantages of MAP and EM based image segmentation
  - 11.1.5 Implementations of MRF-based image segmentation

12 Watershed transformation
13 Model based segmentation
14 Multi-scale segmentation
- 14.1 One-dimensional hierarchical signal segmentation
- 14.2 Image segmentation and primal sketch

15 Semi-automatic segmentation
16 Trainable segmentation
17 Other methods

Wikipedia: Image Segmentation

Region Growing

- Region Growing (or Seed Growing) iteratively groups pixels (seeds) into larger regions

Idea:

- Start from some seeds
- Grow regions by appending similar neighbors

1	2	1	1	0	0	0	0
1	1	9	8	7	0	0	0
1	1	9	9	0	0	0	0
2	1	8	7	1	2	1	1
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1

Seed

For each neighbor:
Similar enough?

Region Growing

- Region Growing (or Seed Growing) iteratively groups pixels (seeds) into larger regions

Idea:

- Start from some seeds
- Grow regions by appending similar neighbors

1	2	1	1	0	0	0	0
1	1	9	8	7	0	0	0
1	1	9	9	0	0	0	0
2	1	8	7	1	2	1	1
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1

Repeat recursively

For each neighbor:
Similar enough?

Region Growing

- Region Growing (or Seed Growing) iteratively groups pixels (seeds) into larger regions

Idea:

- Start from some seeds
- Grow regions by appending similar neighbors

1	2	1	1	0	0	0	0
1	1	9	8	7	0	0	0
1	1	9	9	0	0	0	0
2	1	8	7	1	2	1	1
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1

Final result

Region Growing

- Region Growing (or Seed Growing) iteratively groups pixels (seeds) into larger regions

Idea:

- Start from some seeds
- Grow regions by appending similar neighbors

Three questions:

- How to determine the seeds?
- How to define the similarity?
- When to stop growing?

Region Growing

Question 1:

- How to determine the seeds?

- Depends on the problem

Possibilities:

- Find maxima on intensity profile
- Select seeds evenly spaced on a grid
- Choose random seeds

Region Growing

Question 2:

- How to define the similarity?

- Depends on the problem

Possibilities:

- Intensity similarity
- Color similarity (in some color space)
- Texture similarity (e.g. gradient histograms)

Region Growing

Question 3:

- When to stop growing?

- Threshold on the similarity measure (compared with the seed)

Additionally possible to consider:

- Similarity of new pixel to average of region so far
- Size of region
- Shape of region

Region Growing

Region Growing algorithm:

- Select *k* seed points S_k
- For each seed point S_k:
  - Add all neighboring pixels N_i to S_k if the value of N_i is similar to S_k according to some similarity measure
  - Repeat until no pixels are added

whether the resulting segments should be allowed to overlap or not depends on the application

A recursive algorithm can be found in (Tönnies 2012)

Application

Extracting salient regions from saliency map for finding object candidates

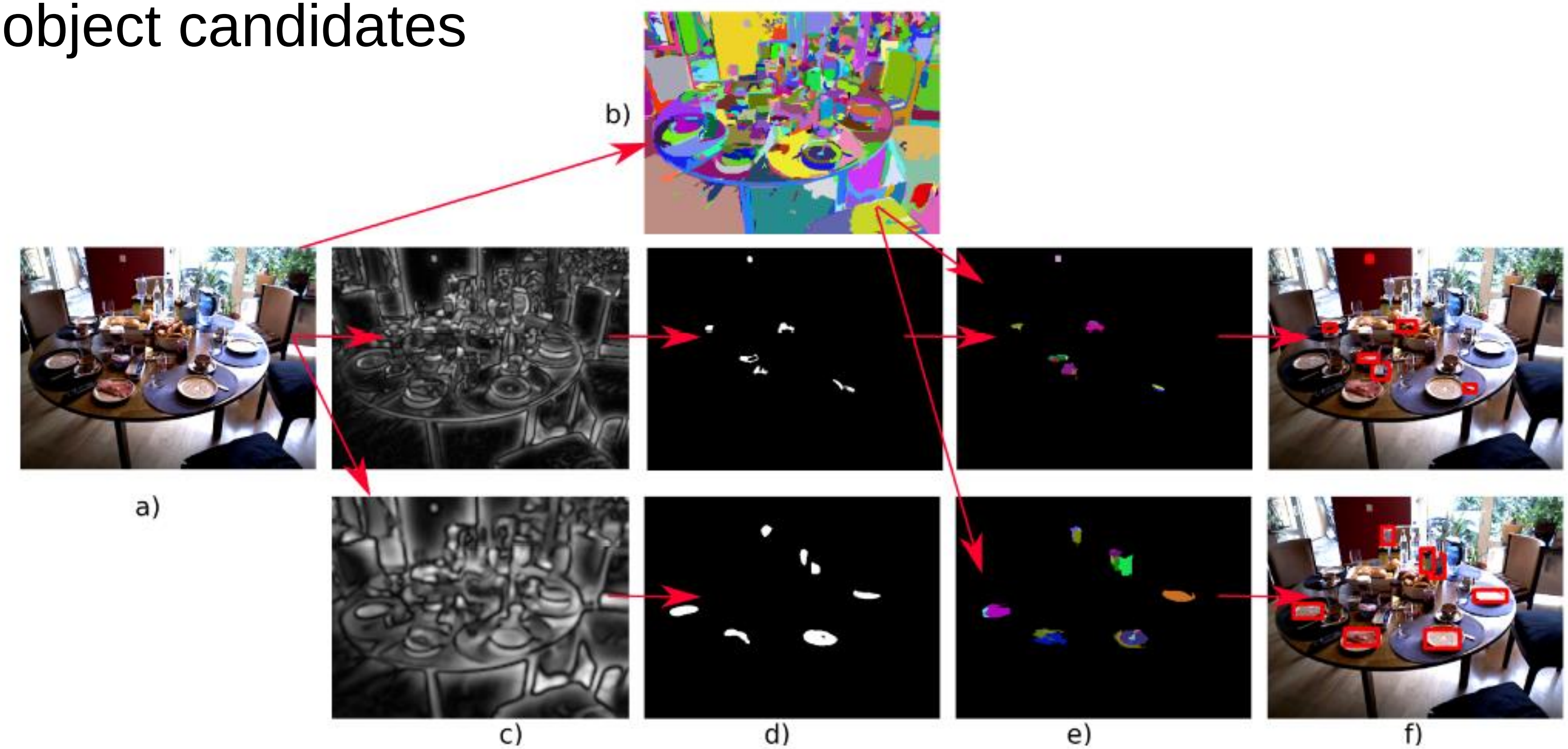


Fig. 2: Object candidate generation: a) original image, b) superpixel segmentation c) for octaves 1 and 2, their specific saliency maps, d) salient blobs obtained by region-growing, e) combining superpixels into object candidates with help of salient blobs, and e) bounding boxes of the object candidates.

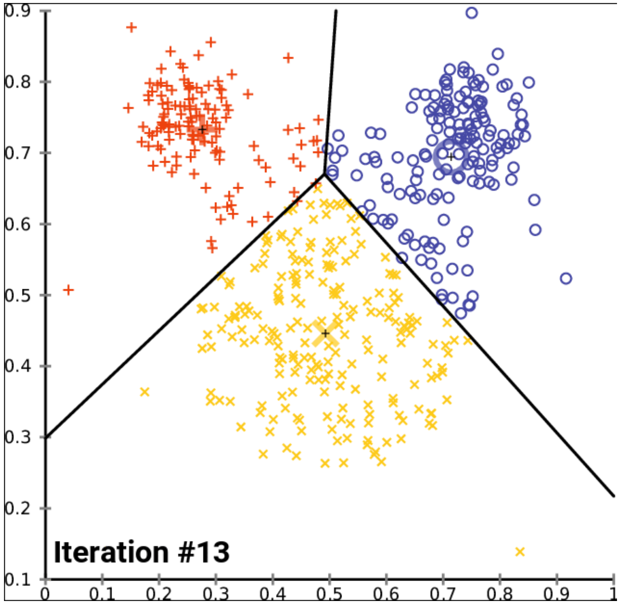
[Horbert et al, ICRA 2015]

Segmentation as Clustering

Clustering: group data points into clusters

K-means: cluster n data points into k clusters

Every cluster has a cluster center (the prototype)



Demo:

https://en.wikipedia.org/wiki/K-means_clustering#/media/File:K-means_convergence.gif

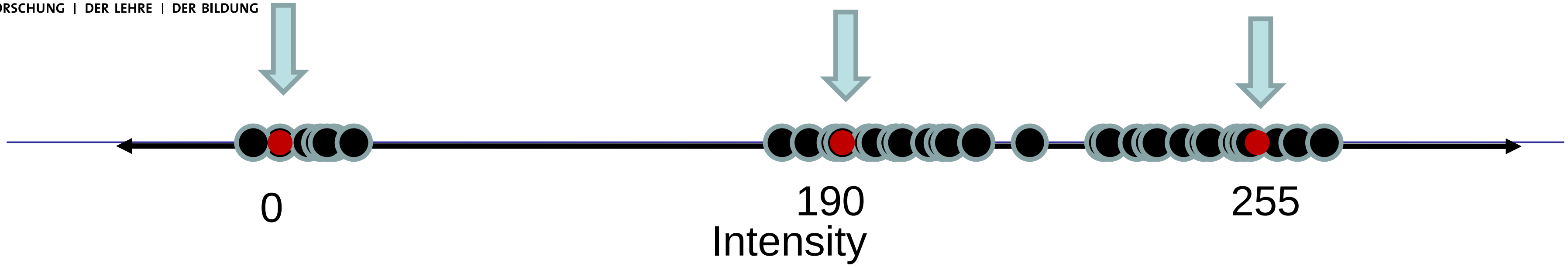
http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

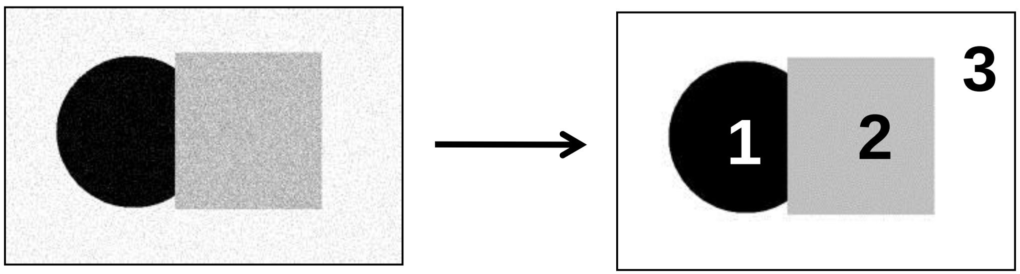
K-Means Clustering

K-Means Algorithm:

- randomly initialize the *k* cluster centers,
- iterate between two steps:
  - **Assignment:** Assign each data point to the nearest cluster (compute distance to the cluster centers and choose closest center)
  - **Update:** Recompute cluster centers
- Repeat until convergence

- Properties
  - Will always converge to *some* solution
  - Can be a “local minimum”

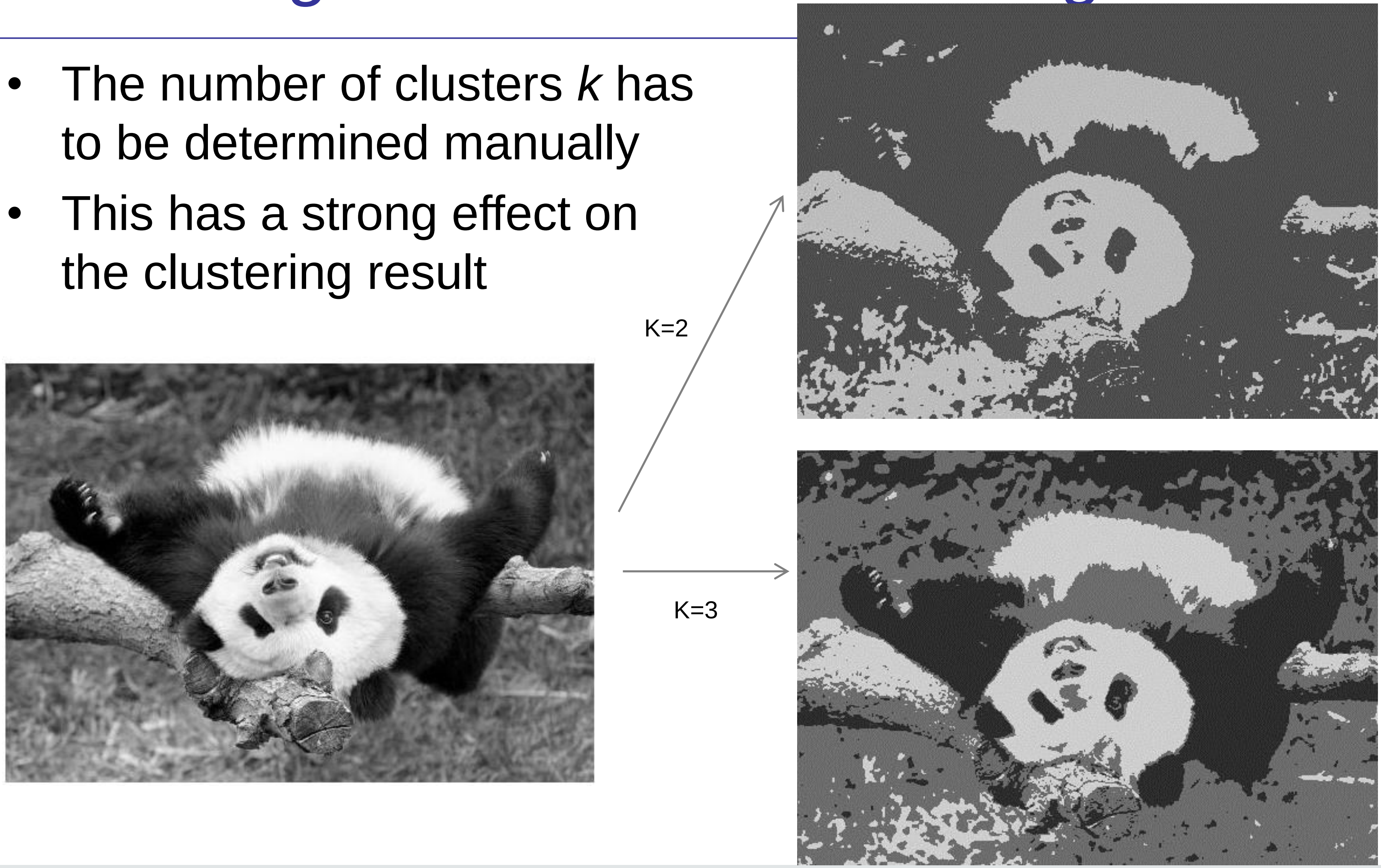




choose three “centers” as the representative intensities, and label every pixel according to which of these centers it is nearest to.

Segmentation as Clustering

- The number of clusters *k* has to be determined manually
- This has a strong effect on the clustering result



Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

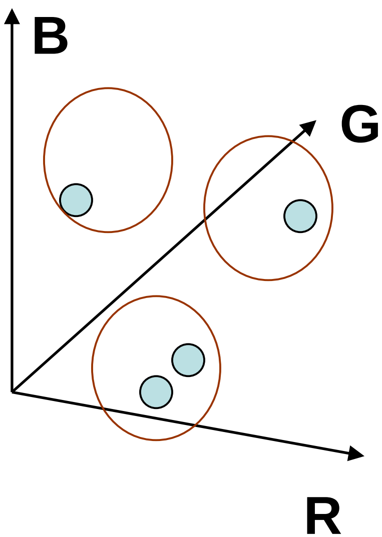
- Grouping pixels based on intensity similarity

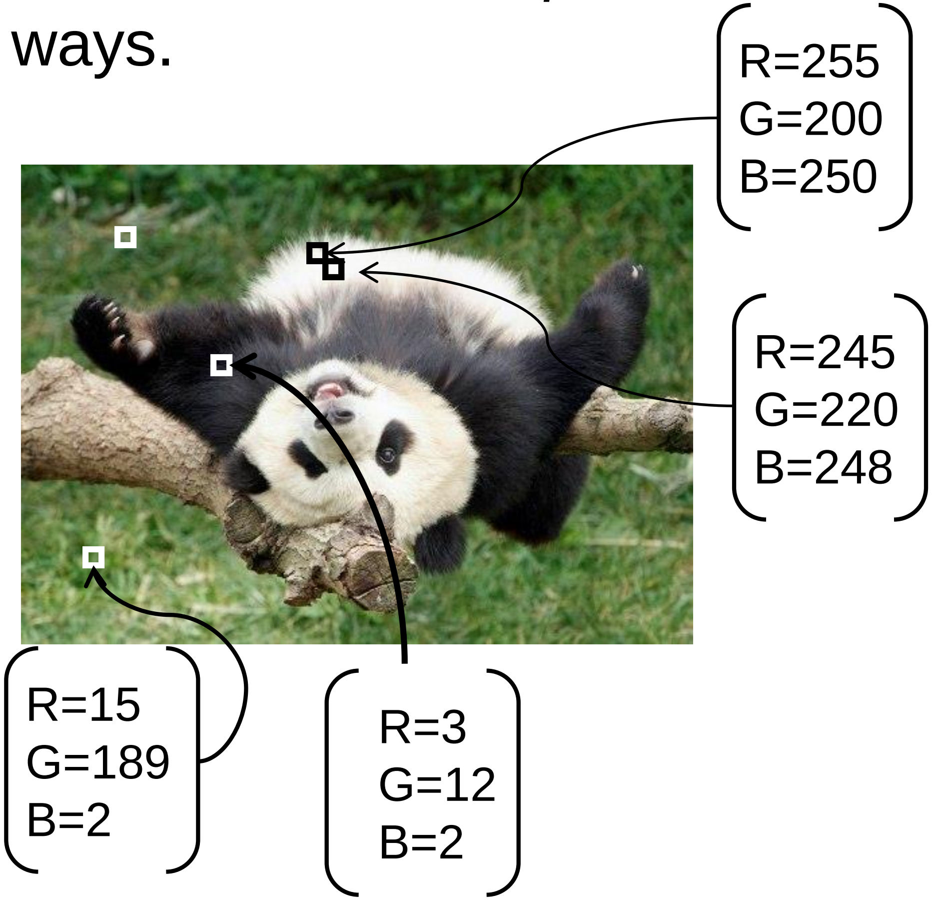
- Feature space: intensity value (1D)

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on color similarity



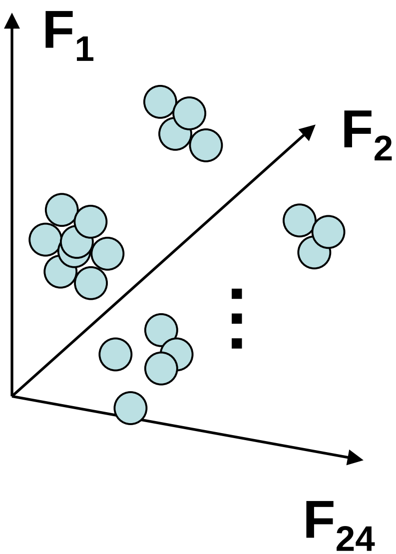


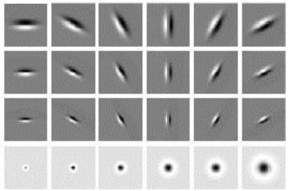
- Feature space: color value (3D)

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on texture similarity



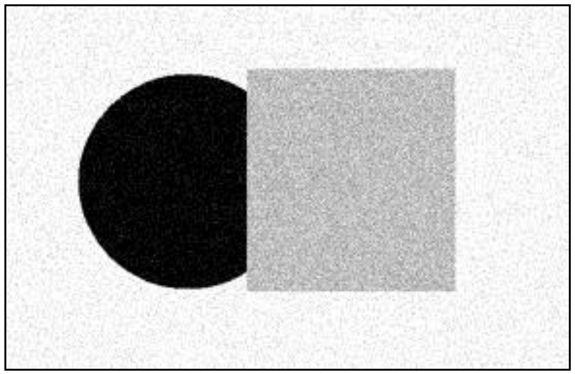


Filter bank of 24 filters

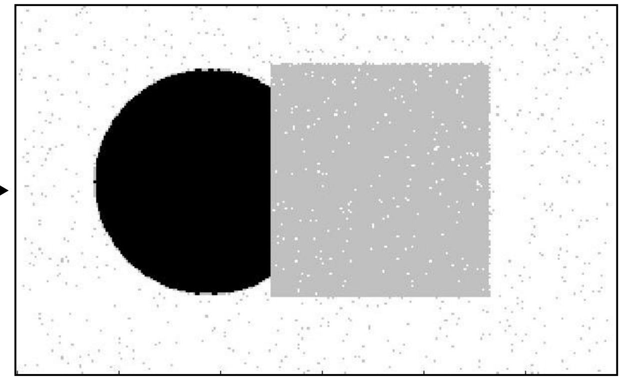
- Feature space: filter bank responses (e.g., 24D)

Spatial coherence

- Assigning a cluster label per pixel may yield outliers:



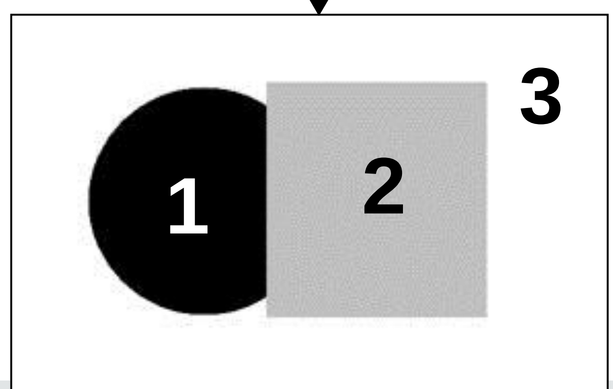
Original



Labeled by cluster center's intensity

?

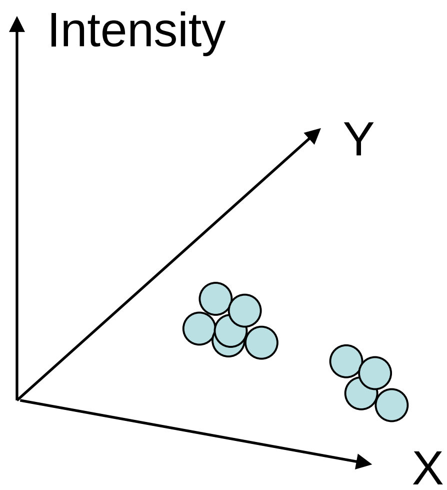
- How can we ensure they are spatially smooth?

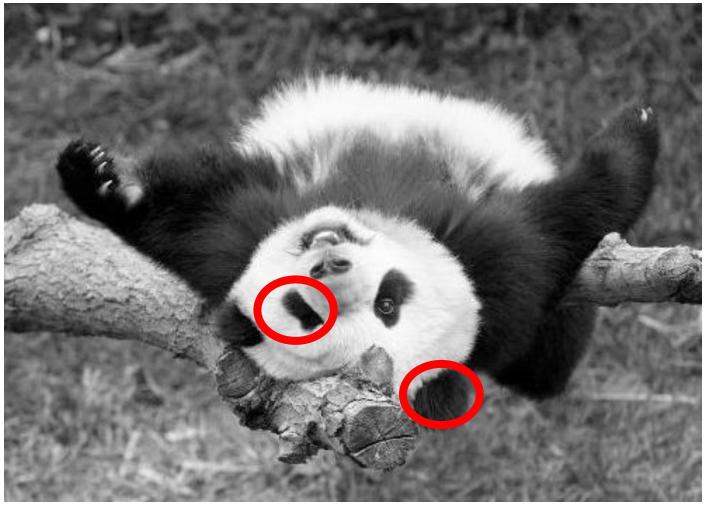


Spatial coherence

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on *intensity+position* similarity



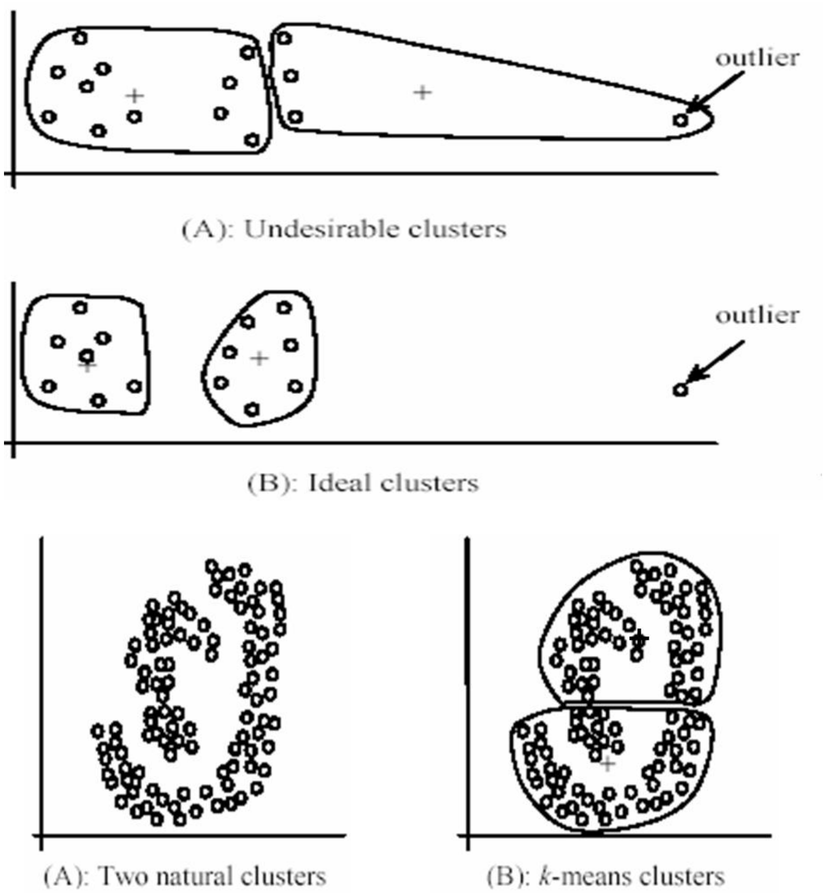


⇒ Simple way to encode both *similarity* and *proximity.*

Summary K-Means

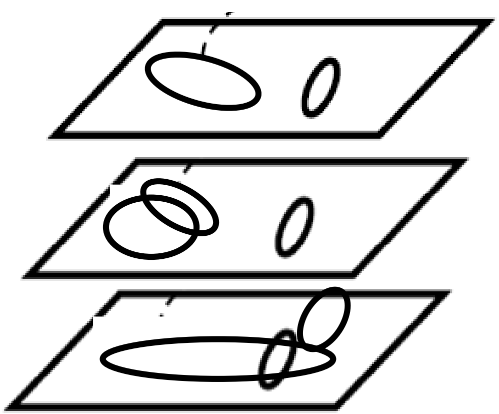
- Pros
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error

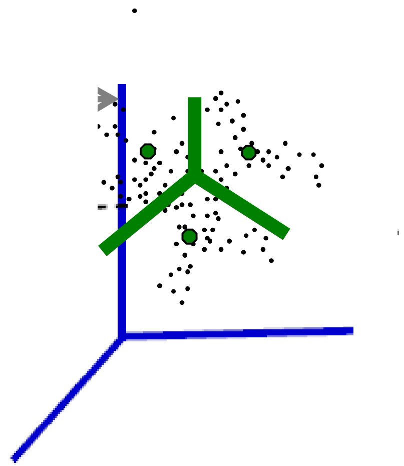
- Cons/issues
  - Setting k?
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters only



Application of K-means

Generate visual words: Map high-dimensional descriptors (e.g. SIFT) to tokens/words by quantizing the feature space





Descriptor space

- **Determine which word to assign to each new image region by finding the closest cluster center.**

Nister, David, and Henrik Stewenius. "Scalable recognition with a vocabulary tree." *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*.IEEE, 2006.

K-means – still up to date

Google Scholar

k-means

Articles

About 18.900 results (**0,12** sec)

Any time
Since 2020
Since 2019
Since 2016
Custom range...

Sort by relevance
Sort by date

include patents
include citations

Create alert

Multiple kernel k-means with incomplete kernels
X Liu, X Zhu, M Li, L Wang, E Zhu, T Liu... - IEEE transactions on ..., 2019 - ieeexplore.ieee.org
Existing MKC algorithms cannot efficiently address the situation where some rows and columns of base kernel matrices are absent. This paper proposes two simple yet effective algorithms to address this issue. Different from existing approaches where incomplete kernel ...
Cited by 44 Related articles All 18 versions

Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms
S Ahmadian, A Norouzi-Fard, O Svensson... - SIAM Journal on ..., 2019 - SIAM
Clustering is a classic topic in optimization with **k-means** being one of the most fundamental such problems. In the absence of any restrictions on the input, the best-known algorithm for **k-means** in Euclidean space with a provable guarantee is a simple local search heuristic ...
Cited by 79 Related articles All 8 versions

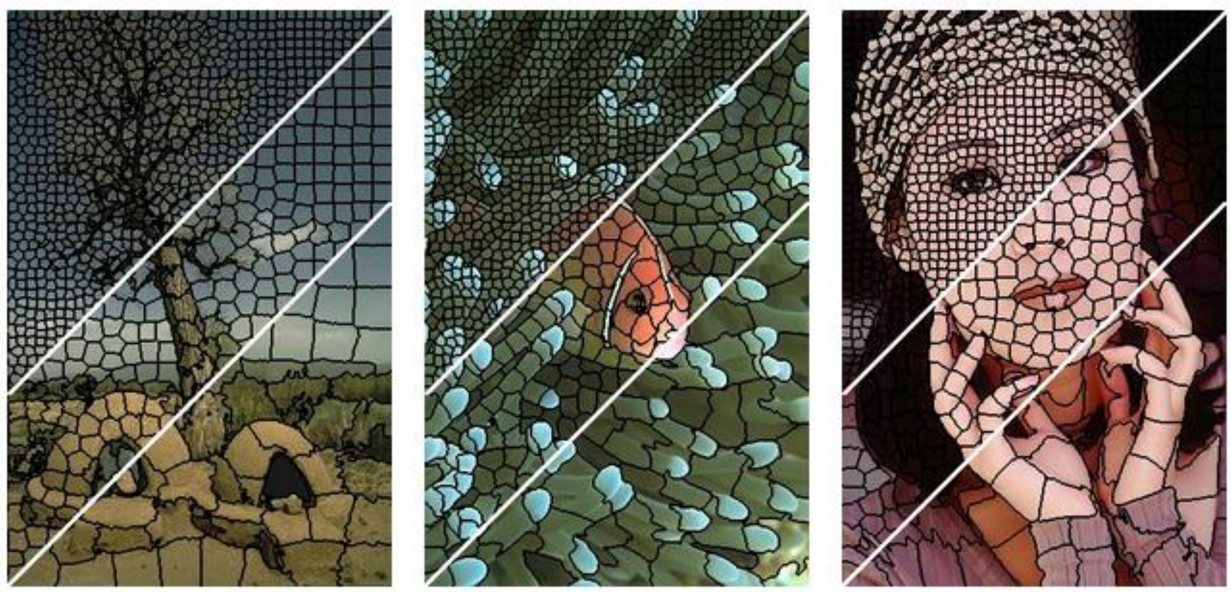
Local Search Yields Approximation Schemes for k-Means and k-Median in Euclidean and Minor-Free Metrics
V Cohen-Addad, PN Klein, C Mathieu - SIAM Journal on Computing, 2019 - SIAM
We give the first polynomial-time approximation schemes (PTASs) for the following problems:(1) uniform facility location in edge-weighted planar graphs;(2) k-median and **k-means** in edge-weighted planar graphs; and (3) **k-means** in Euclidean space of bounded ...
Cited by 68 Related articles All 9 versions

Local Search Yields a PTAS for k-Means in Doubling Metrics
Z Friggstad, M Rezapour, MR Salavatipour - SIAM Journal on Computing, 2019 - SIAM
The most well-known and ubiquitous clustering problem encountered in nearly every branch of science is undoubtedly **k-means**: given a set of data points and a parameter k, select k centers and partition the data points into k clusters around these centers so that the sum of ...
Cited by 68 Related articles All 7 versions

Superpixels

Remember

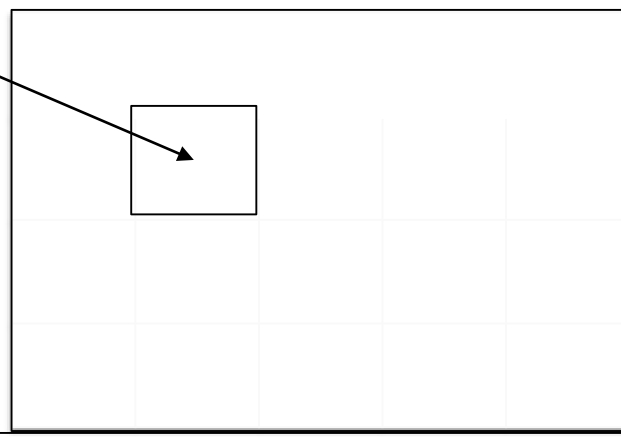
- Superpixels are a different name for the segments obtained by a segmentation algorithm
- Usually, they
  - come from a strong over-segmentation (they resemble “large pixels”)
  - have a coherent, perceptually uniform appearance (color, texture)
- Example: SLIC superpixels (Achanta et al. 2010):



[Achanta 2010]

SLIC Superpixels

- SLIC: **Simple Linear Iterative Clustering**
- SLIC superpixels are based on a local version of k-means
- SLIC is **very fast** to compute and **memory-efficient**
- Apply k-means to a local region on a regularly sampled grid:



Radhakrishna Achanta

	Citations	year
SLIC superpixels compared to state-of-the-art superpixel methods R Achanta, A Shaji, K Smith, A Lucchi, P Fua, S Süsstrunk IEEE transactions on pattern analysis and machine intelligence 34 (11), 2274 ...	2464 *	2012

SLIC Superpixels

Algorithm:

- Initiate k cluster centers, sampled regularly on grid
- Move centers to point of minimal gradient intensity
- Repeat until convergence:
  - Assign each pixel to its nearest cluster centers
  - Adjust cluster center to mean color and position

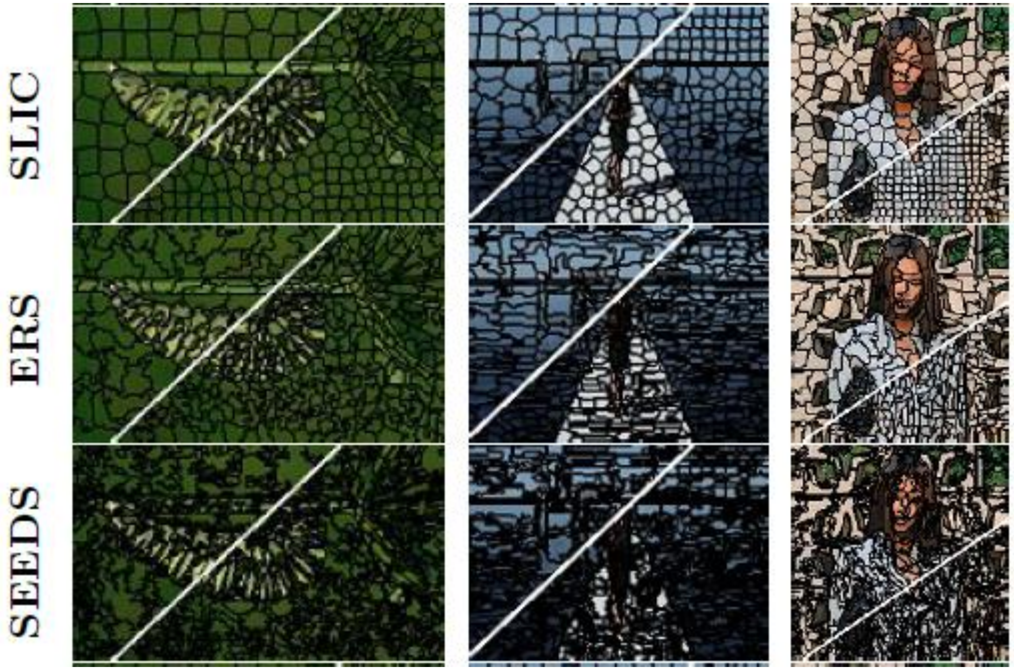
SLIC Superpixels

Number of superpixels: 500

Number of superpixels: 100

Superpixel Evaluation

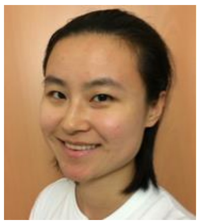
- Superpixel Evaluation Framework: http://davidstutz.de/projects/superpixel-benchmark/
- (Stutz, Hermans, Leibe: CVIU 2018)
- Evaluation of 28 superpixel algorithms on 5 datasets



„we recommend 6 algorithms for use in practice, thereby covering a wide range of application scenarios: **ETPS** [84], **SEEDS** [80], **ERS** [47], **CRS** [78, 79], **ERGC** [68] and **SLIC** [74]."

Adaptive Superpixel Seeding

Use prior map to determine superpixel resolution



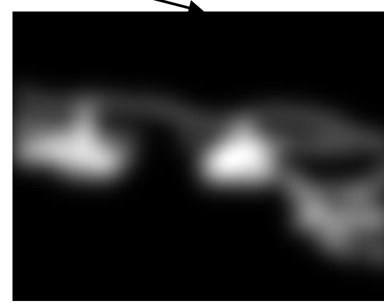
Ge Gao

Christian Wlms

Saliency prior

Edge prior

IROS 2017



GCPR 2017

- Wilms/Frintrop: Edge Adaptive Seeding for Superpixel Segmentation, Proc. of the German Conference on Pattern Recognition (GCPR), 2017
- IROS 2017: Gao, Lauri, Zhang, Frintrop: Saliency-guided Adaptive Seeding for Supervoxel Segmentation

Summary

- Segmentation methods partition the image into segments
- The literature on segmentation is huge, the task is still considered unsolved. Which algorithm to choose depends highly on the application
- We have seen two simple but effective algorithms: region growing and k-means
- More sophisticated methods will be covered in the summer term lecture “Computer Vision 2”

Primary Literature

Gonzalez/Woods, 4th ed, 2018

Region growing:

- Toennies, Klaus D. "Segmentation: Principles and Basic Techniques." *Guide to Medical Image Analysis*. Springer London, 2012. 171-209

K-means

- Bishop, Christopher M. "Pattern recognition." *Machine Learning* 128 (2006)

Secondary Literature

- Martin, David, et al. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics." *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 2. IEEE, 2001.
- Achanta, Radhakrishna, et al. „SLIC superpixels". No. EPFL-REPORT-149300. 2010.
- Horbert, Esther, et al. "Sequence-level object candidates based on saliency for generic object recognition on mobile systems." *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015.
- Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007.
- Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012): 2274-2282.
- Jose Sigut, Francisco Fumero, Omar Nuñez: *Over- and Under-Segmentation Evaluation based on the Segmentation Covering Measure,* WSCG 2015 Conference on Computer Graphics, Visualization and Computer Vision
- Tönnies, Klaus D. *Grundlagen der Bildverarbeitung*. Vol. 11. München: Pearson Studium, 2005.