VIC-Introduction to Visual Computing
Course 5 : Segmentation

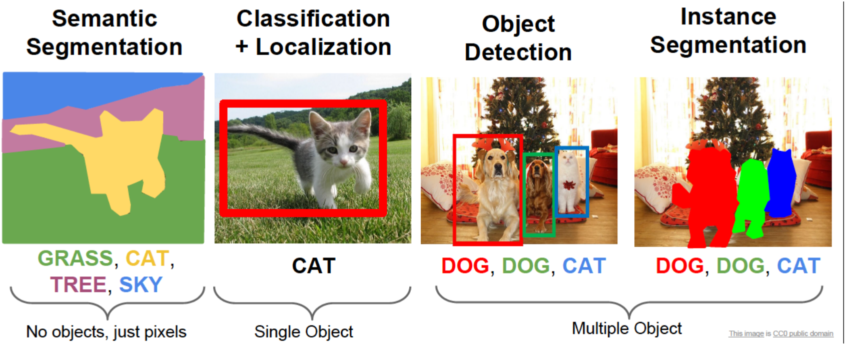
Maria Vakalopoulou and Céline Hudelot

31 janvier 2025

Plan

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Segmentation : Introduction



Segmentation : Introduction

Segmentation divides an image into groups of pixels

- Meaningful regions (coherent objects)
- Linear structures (line, curve, . . .)
- Shapes (circles, ellipses, . . .)

Segmentation : Introduction

Why ?

- Identify groups of pixels that form a set.
- Identify objects of interest in the image.
- Differentiate these objects from the background.
- Assign each pixel a code indicating which object it belongs to.

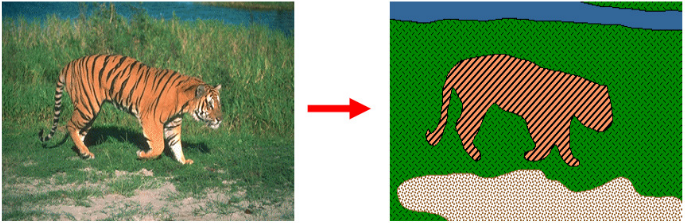
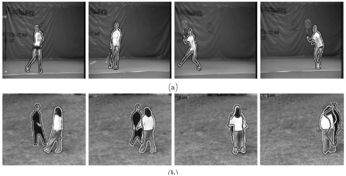


Image Segmentation : Why ?

For decision-making purposes, the *objects* in the image carry information.

- To understand the content of the image.
- To enable consistent post-processing of this content (e.g., object tracking, video surveillance).



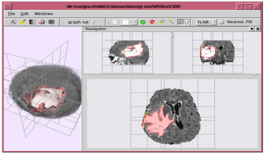
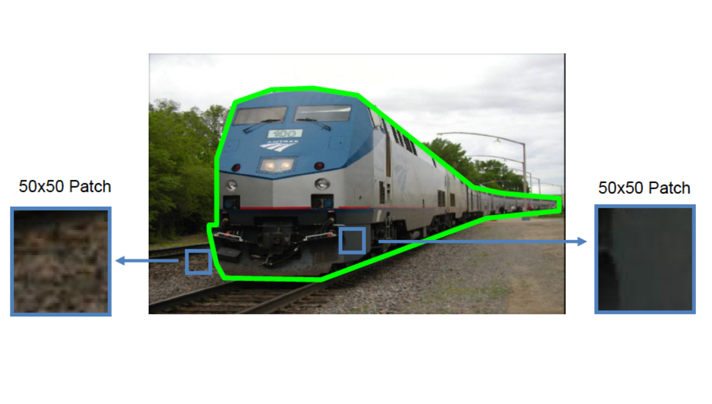


Image Segmentation : Why ?

Segmentation to Guide Feature Extraction

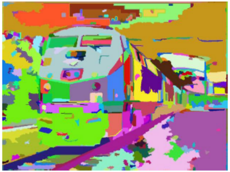


Source : D. Hoeim

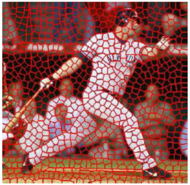
Image Segmentation : Why ?

Segmentation for Efficiency

Grouping pixels with similar appearance to make subsequent processing more efficient.



[Felzenszwalb and Huttenlocher 2004]



[Shi and Malik 2001]

[Hoiem et al. 2005, Mori 2005]

Image Segmentation : Why ?

Segmentation as an End in Itself

Source : D. Hoeim

Image Segmentation : Examples

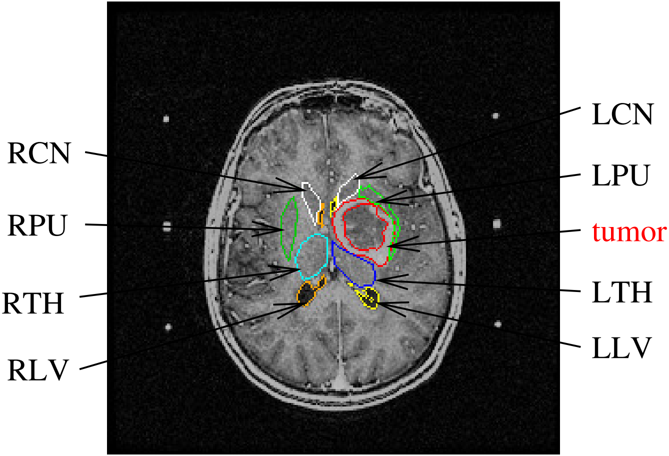
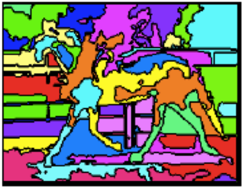
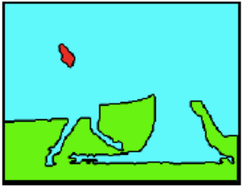


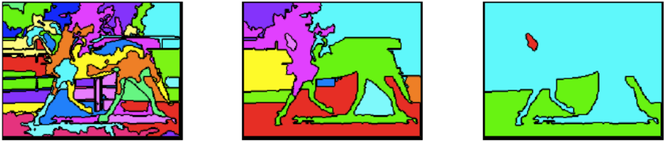
Image Segmentation : Different Types



Oversegmentation



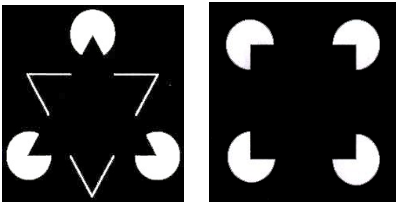
Undersegmentation



Multiple Segmentations

Image Segmentation : A Difficult Problem

Due to optical illusions



Contours illusoires

Un dalmatien ? Le cerveau préfère une information structurée

Image Segmentation : A Difficult Problem

Due to the complexity of reality

Textured background

slide credit Svetlana Lazebnik

Image Segmentation : A Difficult Problem

Image Segmentation

How ?

- An object is a region in the image that has semantic coherence.
- In practice : connected, consistent in color, defined by contours, with a prior shape, etc.

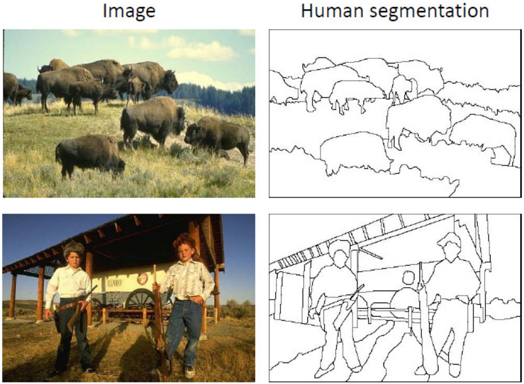


Image Segmentation

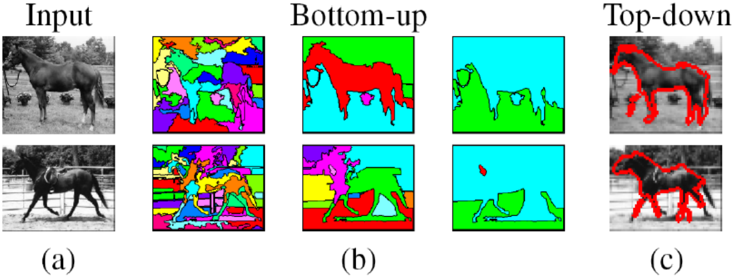
How ?

Segmentation is typically based on :

- Discontinuities, i.e., edges : abrupt changes, boundaries between regions.
- Homogeneous areas : searching for consistent regions according to a criterion (same colors, textures, intensities, motion...).
- Dividing an image into different regions and/or edges.
- Region/closed contour duality :
  - A region is defined by its contour.
  - A contour is a boundary between two regions.

Image Segmentation : Main Approaches

- Bottom-up : grouping elements with similar characteristics.
- Top-down : grouping elements that likely belong to a given class (requires a model).



[Levin and Weiss 2006]

Image Segmentation : Bottom-up Approach

Intuitive Definition

Separating the image into coherent regions (usually spatial and color consistency).

Mathematical Formulation

Partition $S = R_1, R_2, ..., R_n$ of I such that :

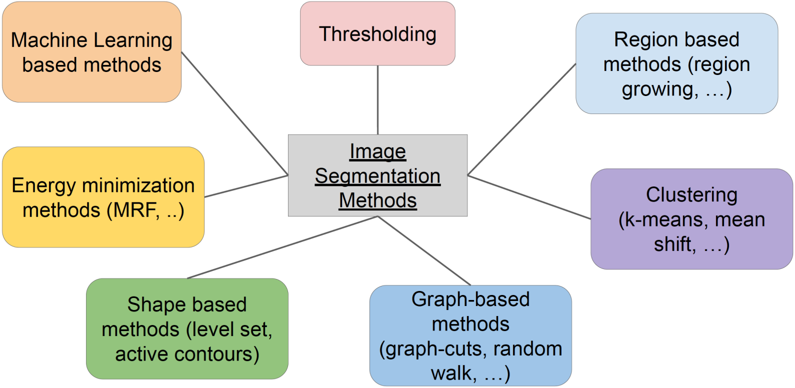
- $I = \bigcup R_i, \forall i \in [1..n]$
- $R_i \cap R_j = \emptyset, \forall i, j \; s.t. \; i \neq j$
- R_i is connected, $\forall i \in [1..n]$
- $\mathcal{P}(R_i)$ is true, where $\mathcal{P}$ is the predicate *the region is homogeneous*.
- $\mathcal{P}(R_i \cup R_j)$ is false, $\forall (i, j)$ adjacent (maximal partition).

Image Segmentation : Introduction

Different Approaches

- Pixel Description Space
  - Thresholding segmentation.
  - Clustering-based segmentation.
- Image Space (2 dual approaches) :
  - Region-based approach :
    - Region growing segmentation.
    - Split-and-Merge method.
  - Contour-based approach :
    - Contour closure segmentation.
    - Active contours.
  - Graph-based segmentation.
  - ...
- Modeling as an Optimization Problem : constrained minimization between the original image and the segmented image.

Image Segmentation : different approaches



Outline

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Thresholding Segmentation

- A simple and very popular method for digital image processing.
- Based on histograms.

Idea

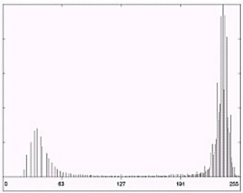
Find the threshold(s) to separate the histogram into parts and thus divide the image into regions.

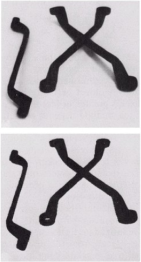
Different Approaches

- Global : a single threshold for the entire image.
- Local : a threshold for a portion of the image.
- Adaptive : a threshold that adjusts according to different parts of the image.

Thresholding Segmentation : Binarization

- Basic thresholding (2 classes) :
  - If $value(pixel) \geq threshold$, then $value(pixel) = 1$.
  - If $value(pixel) < threshold$, then $value(pixel) = 0$.
- The result of thresholding is a binary image.
- Problem : how to choose the threshold correctly ?



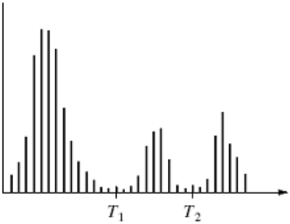


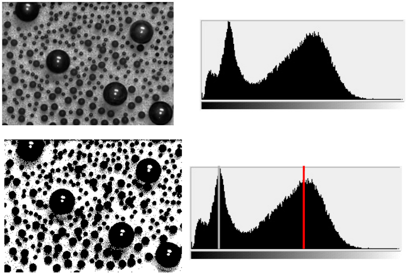
Thresholding Segmentation : Choosing the Threshold

Threshold Selection

- By performing tests.
- Mean value of gray levels.
- Median value between the maximum and minimum gray level.
- Adaptive thresholding : find the threshold automatically.
  1. Choose an initial value for S.
  2. Obtain two pixel groups based on this value.
  3. Compute the mean gray level values for these two groups : μ_1 and μ_2.
  4. Compute a new threshold S such that $S = \frac{1}{2}(\mu_1 + \mu_2)$.
  5. Repeat until S is constant.

Thresholding Segmentation : Multiple Thresholds





Global Thresholding : Problem

Global thresholding does not handle this case !

Solution

Adaptive local thresholding.

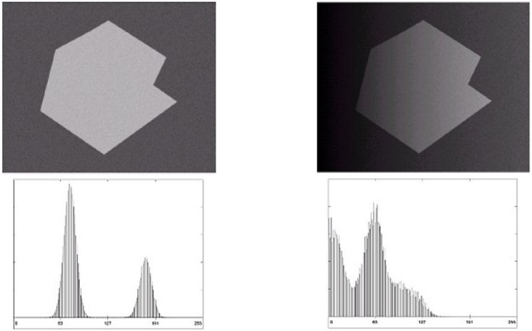


FIGURE – Source : Gonzalez & Woods

Adaptive Local Thresholding

- The image is divided into sub-images, and each is processed with its own threshold.
- The choice of sub-image size is important.
- Before processing each sub-image, check the variance of gray levels to decide if segmentation is needed.

Thresholding Segmentation : Advantages and Disadvantages

Many methods are based on analyzing the modes of the pixel distribution.

Advantages

- Simple and real-time.
- Works well with multi-modal histograms (multiple peaks).

Disadvantages

- The number of classes must be known.
- False elements may appear since spatial components are not considered.

Thresholding Segmentation : Improvements

- Improved thresholding with post-processing integrating spatial analysis.
- Principle :
  1. Locate an isolated mode in a histogram.
  2. Detect corresponding zones by thresholding.
  3. Among the zones contributing to this mode, select the largest connected region.
  4. Return to step 1.

Histogram-Based Methods

Summary

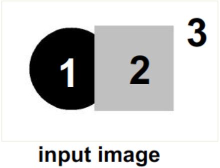
- Simple
- Limited performance, effective if objects are on a uniform background
- No (or little) use of spatial information.
- Extension if the pixel descriptor is a vector rather than a scalar : from histograms to clustering.

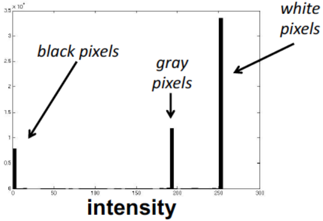
Outline

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Segmentation by Clustering

From thresholding to clustering.

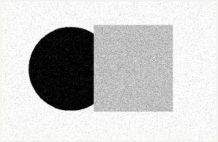




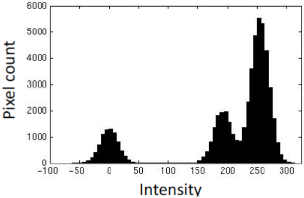
Source : K. Grauman

Segmentation by Clustering

From thresholding to clustering.



Input image

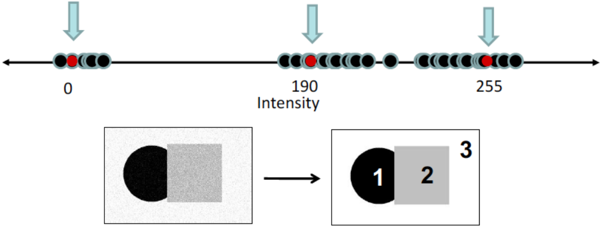


Source : K. Grauman

How to determine the *average* intensities defining the 3 groups ?

Segmentation by Clustering

From thresholding to clustering.



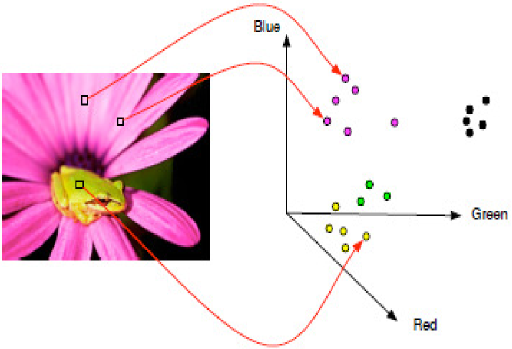
Source : K. Grauman

The best centers c_i are those that minimize the sum of squared differences between all points and their center c_i

$$SSD = \sum_{k} \sum_{x \in C_k} (x - c_k)^2$$

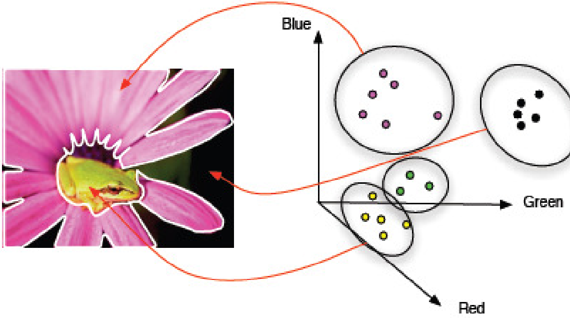
Segmentation by Clustering

Representation of the image in terms of pixel clusters in the pixel description space.



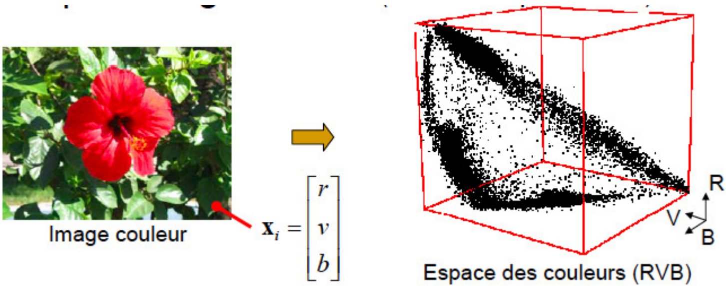
Segmentation by Clustering

Representation of the image in terms of pixel clusters.



Feature Space

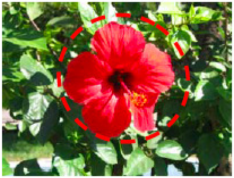
- Representation of data (image) features in a common space.
- Example : color image $\rightarrow$ color features.

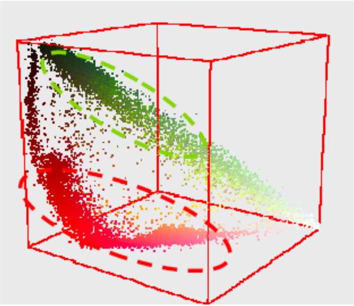


Feature Space

Principle

- Concentration of information in the feature space $\Rightarrow$ meaningful information.
- Several approaches :
  - Simple clustering : K-means.
  - Density estimation in feature space : Mean Shift.





K-means Algorithm

Dividing points into k groups :

- k is given in advance (algorithm parameter).
- The center of a group is defined as the centroid of the elements (pixels) in the group.
- Demonstration of the algorithm : `https://www.naftaliharris.com/blog/visualizing-k-means-clustering/`

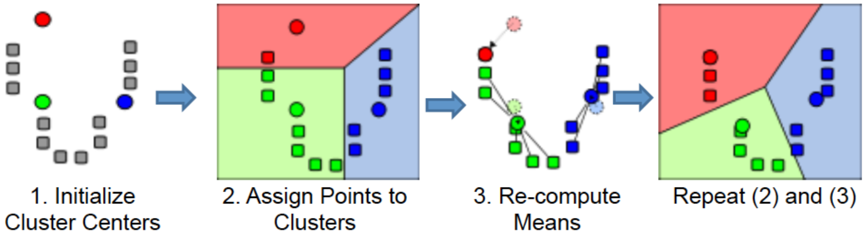
Partitioning Clustering : K-means (MacQueen, 1967) [1]

Principle

- Choose K initial elements as *centroids* $\mu_k, k = 1..K$ of the K groups.
- Each point (or object) x_i is associated with one and only one cluster, the cluster C_l whose centroid μ_l is the closest.
- Recalculate the centroid of each cluster $C_k, k = 1..K$.
- Iterate until stability, i.e., convergence of the criterion (i.e., objects no longer change groups).

1. J. MacQueen, Some methods for classification and analysis of multivariate observations," Proc. of the Fifth Berkeley Symp. On Math. Stat. and Prob., vol. 1, pp. 281-296, 1967.

K-means Algorithm



Partitioning Clustering : K-means

Details

- Initial centers are often chosen randomly :
  - Within the range of definition of x_i
  - From the set of x_i
  - Resulting clusters may vary, i.e., different initializations may lead to different clusters.
- The center is the mean of points in a group.
- *Proximity* is calculated using Euclidean distance, cosine similarity, correlation, etc.
- The algorithm often converges in a few iterations
  - The stopping condition is often changed to *until only a few points change groups*.

K-means Algorithm

Example of algorithm results based on intensity or color.

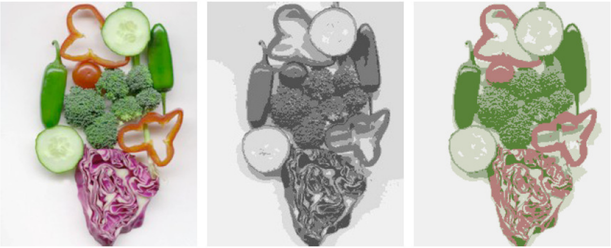


Image source *K-moyennes sur l'intensité* *K-moyennes sur la couleur*

FIGURE – Source : Forsyth & Ponce

Depending on the chosen feature space (intensity, color, texture, intensity and position, ...), pixels can be grouped in different ways.

K-means Algorithm

Influence of the parameter k.

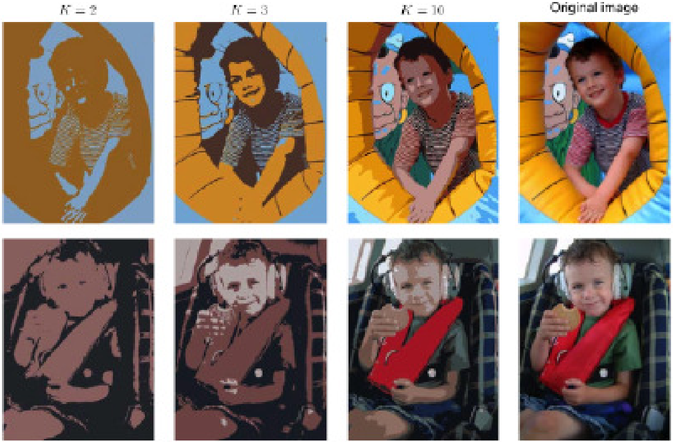


FIGURE – Source : E. Arnaud

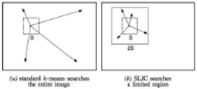
k-means Algorithm for Superpixels : SLIC Approach

Superpixels : Entities that are perceptually meaningful and obtained from a low-level grouping process.

- Feature space : intensity + position
  - Lab color space
  - Limited region (window of size 2S)

$$[l_k, a_k, b_k, x_k, y_k]$$

Fig. 1: Images segmented using SLIC into superpixels of size 64, 256, and 1024 pixels (approximately).



- Distances

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$D = \sqrt{\frac{d_c}{N_c}^2 + \frac{d_s}{N_s}^2}$$

- Initialization = spatial grid

Achanta et al., SLIC Superpixels Compared to State-of-the-art Superpixel Methods, PAMI 2012

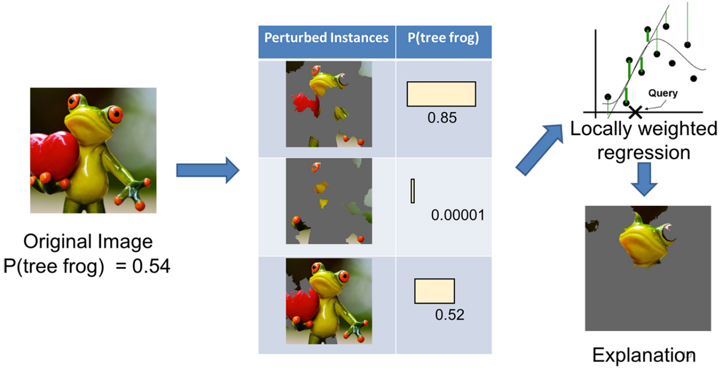
K-means for Superpixels : SLIC Approach

SLIC = Simple Linear Iterative Clustering

Algorithm 1 SLIC superpixel segmentation

```
/* Initialization */
Initialize cluster centers C_k = [l_k, a_k, b_k, x_k, y_k]^T by
sampling pixels at regular grid steps S.
Move cluster centers to the lowest gradient position in a
3 × 3 neighborhood.
Set label l(i) = −1 for each pixel i.
Set distance d(i) = ∞ for each pixel i.
repeat
  /* Assignment */
  for each cluster center C_k do
    for each pixel i in a 2S × 2S region around C_k do
      Compute the distance D between C_k and i.
      if D < d(i) then
        set d(i) = D
        set l(i) = k
      end if
    end for
  end for
  /* Update */
  Compute new cluster centers.
  Compute residual error E.
until E ≤ threshold
```

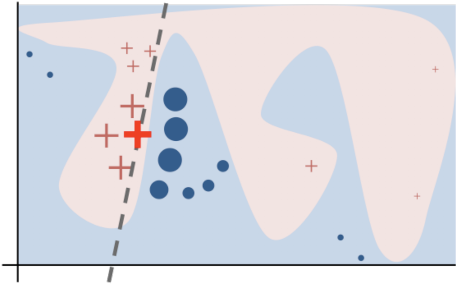
Small Parenthesis : Superpixels in XAI



LIME, Local Interpretable Model-agnostic Explanations (Ribeiro et al. KDD 2016)

One of the the most known contributions in this domain.

- The overall decision boundary is complex
- In the neighborhood of a single decision, the boundary is simple
- A single decision can be explained by auditing the black box around the given instance and learning a ***local*** decision.

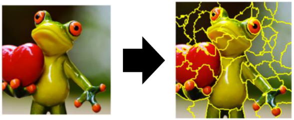


Ribeiro et al - Why Should I Trust You ? : Explaining the Predictions of Any Classifier. KDD 2016 : 1135-1144 [2]

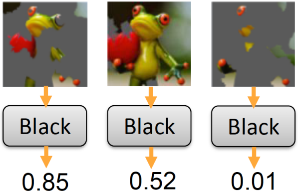
2. https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf

LIME : Intuition by example

- Given a data point you want to explain



- Sample at the nearby : each image is represented as a set of superpixels

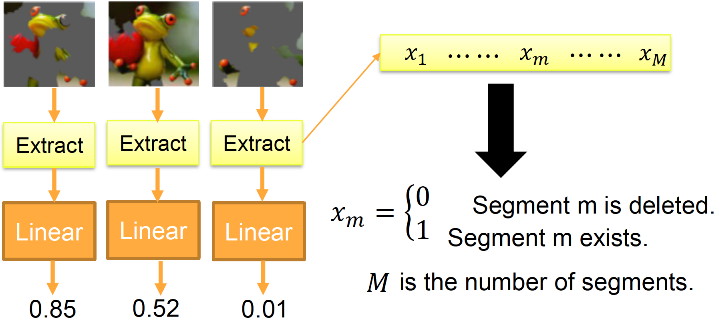


Randomly delete some segments.

Compute the probability of “frog” by black box

LIME : Intuition by example

- Fit with linear (or interpretable model)



LIME : Intuition by example

- Interpret the learned model

Extract

Linear

0.85

$$y = w_1 x_1 + \cdots + w_m x_m + \cdots + w_M x_M$$

$$x_m = \begin{cases} 0 & \text{Segment m is deleted.} \\ 1 & \text{Segment m exists.} \end{cases}$$

M is the number of segments.

If $w_m \approx 0$ $\Rightarrow$ segment m is not related to “frog”

If w_m is positive $\Rightarrow$ segment m indicates the image is “frog”

If w_m is negative $\Rightarrow$ segment m indicates the image is not “frog”

LIME

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

LIME

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

Just collecting samples!

LIME

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$ Lasso “feature” selection
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

Superpixels and Data Augmentation

Many studies use superpixels for data augmentation.

(a) Image 1 (b) SuperpixelGridCut (c) SuperpixelGridMean (d) Image 2 (e) SuperpixelGridMix

Figure 1. (b) and (c) show generated augmented images using the proposed *SuperpixelGridCut* and *SuperpixelGridMean* applied from the original image (a). (e) shows the generated augmented image using the *SuperpixelGridMix* over the original images (a) and (d).

(Hammoudi et al., 2022) SuperpixelGridCut, SuperpixelGridMean, and SuperpixelGridMix Data Augmentation [3]

3. `https://arxiv.org/pdf/2204.08458.pdf`

Superpixel Sampling Networks

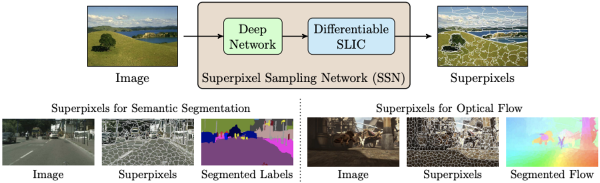


Fig. 1: **Overview of Superpixel Sampling Networks.** A given image is first passed onto a deep network that extracts features at each pixel, which are then used by differentiable SLIC to generate the superpixels. Shown here are a couple of example SSN generated task-specific superpixels for semantic segmentation and optical flow.

https://arxiv.org/abs/1807.10174

Superpixels et transformers

SPFormer: Enhancing Vision Transformer with Superpixel Representation

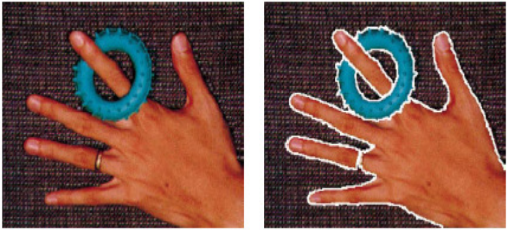
Jieru Mei[1] Liang-Chieh Chen[2] Alan Yuille[1] Cihang Xie[3]
[1]Johns Hopkins University [2]Bytedance [3]UC Santa Cruz

A lot of works on superpixels to tackle the inherent limitations of patch representations in ViTs. `https://arxiv.org/pdf/2401.02931.pdf`

Segmentation by Clustering

Mean-Shift Algorithm

- D. Comaniciu and P. Meer. Mean Shift : A Robust Approach Toward Feature Space Analysis. PAMI 2002.
- `http://courses.csail.mit.edu/6.869/handouts/PAMIMeanshift.pdf`
- Clustering method that does not require prior knowledge of the number of clusters or constraints on their shape.
- Principle : detection of the modes of a probability distribution through a recursive procedure.

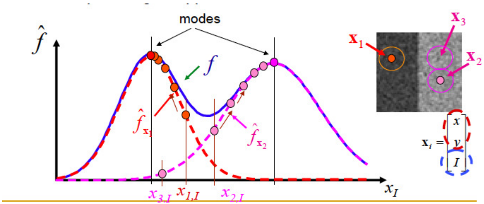


Segmentation by Clustering : Mean-Shift Algorithm

Principle

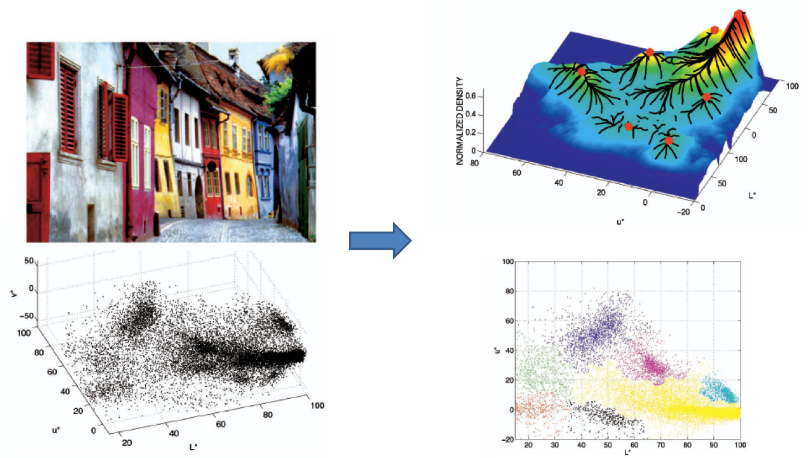
Finding the modes of a probability density function.

- Mode = maximum of a probability density function. $\nabla \hat{f}(x) = 0$.
- A mode characterizes a probability density.
- A region is characterized by a probability density : multiple regions $\rightarrow$ multiple modes.
- Finding which mode a data point belongs to $\rightarrow$ Finding which region the data point belongs to.



Segmentation by Clustering : Mean-Shift Algorithm

The mean shift algorithm searches for modes or density maxima in the feature space.



Segmentation by Clustering : Mean-Shift Algorithm

Principle

- Perform local estimates of the density gradient at data points.
- Move these points along the estimated gradient iteratively.
- Repeat until convergence.
- Stationary points = local maxima of the distribution.
- Points associated with the same stationary point = same cluster.

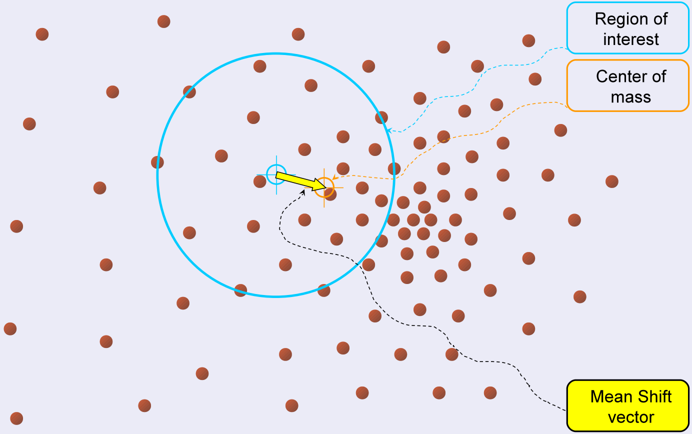
Segmentation by Clustering : Mean-Shift Algorithm

Algorithm

1. Choose a search window size.
2. Select an initial position for the search window.
3. Compute the mean position within the search window (center of data points).
4. Center the search window at this mean position.
5. Repeat steps 3 and 4 until convergence.

Segmentation by Clustering : Mean-Shift Algorithm

Intuitive Description

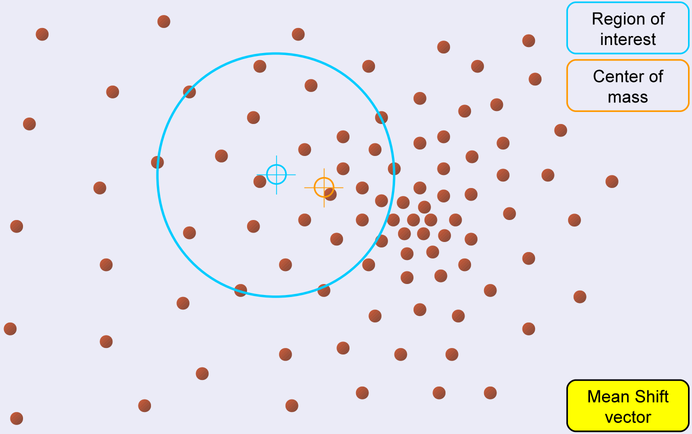


Objective : Find the densest region

Distribution of identical billiard balls

Segmentation par clustering :Algorithme Mean-Shift

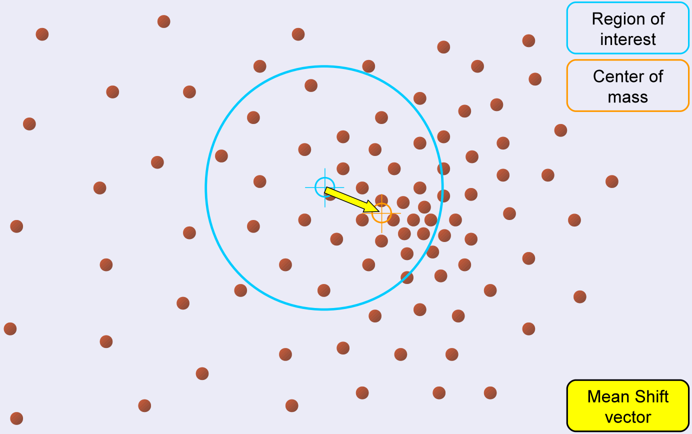
Intuitive Description



Objective : Find the densest region
Distribution of identical billiard balls

Segmentation par clustering :Algorithme Mean-Shift

Intuitive Description

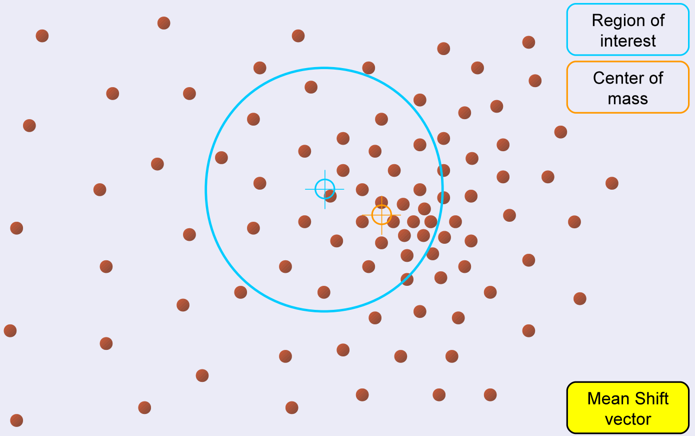


Objective : Find the densest region

Distribution of identical billiard balls

Segmentation par clustering :Algorithme Mean-Shift

Intuitive Description

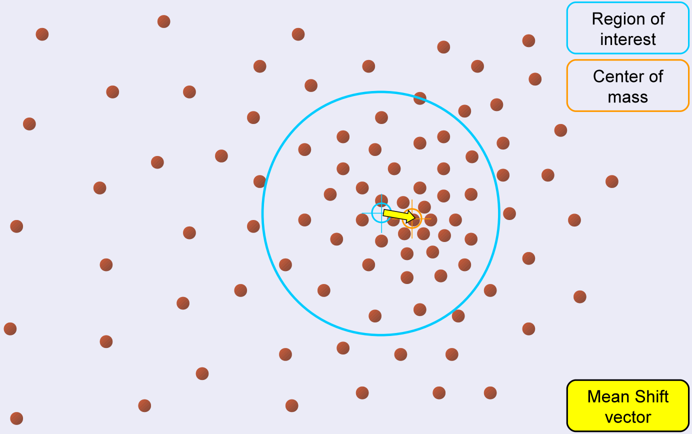


Objective : Find the densest region

Distribution of identical billiard balls

Segmentation par clustering :Algorithme Mean-Shift

Intuitive Description

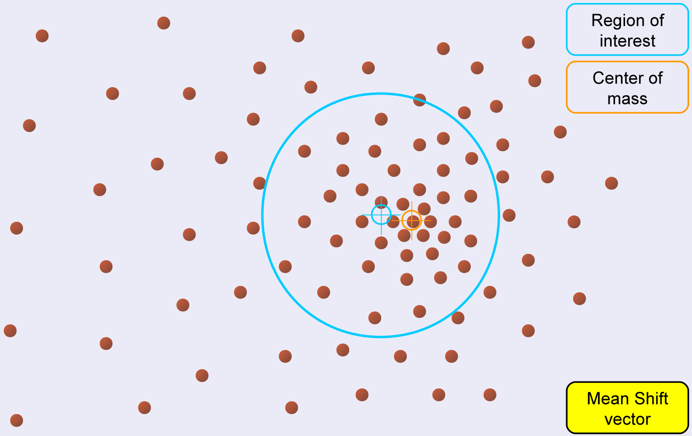


Objective : Find the densest region

Distribution of identical billiard balls

Segmentation par clustering :Algorithme Mean-Shift

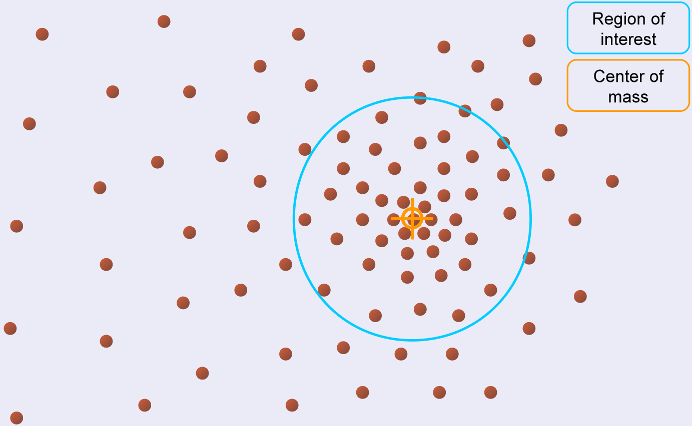
Intuitive Description



Objective : Find the densest region
Distribution of identical billiard balls

Segmentation par clustering :Algorithme Mean-Shift

Intuitive Description



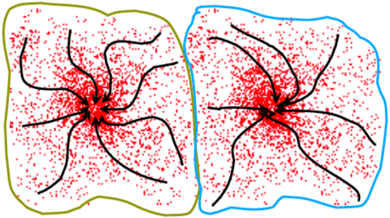
Objective : Find the densest region

Distribution of identical billiard balls

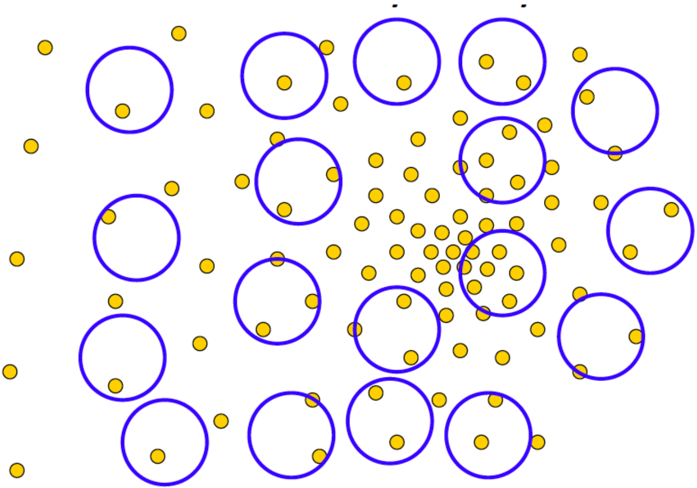
Segmentation by Clustering : Mean-Shift Algorithm

Clustering using mean shift.

- Cluster : all points in the attraction basin of a mode.
- Attraction basin : the region where all trajectories lead to the same mode.

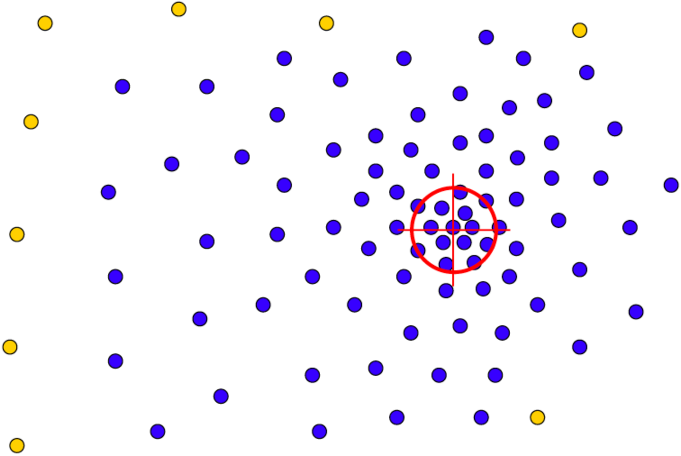


Segmentation by Clustering : Mean-Shift Algorithm



In practice = multiple starting points are processed in parallel.

Segmentation by Clustering : Mean-Shift Algorithm



Clusters = points traversed by windows leading to the same mode.

Segmentation by Clustering : Mean-Shift Algorithm

Mean-Shift Segmentation Algorithm

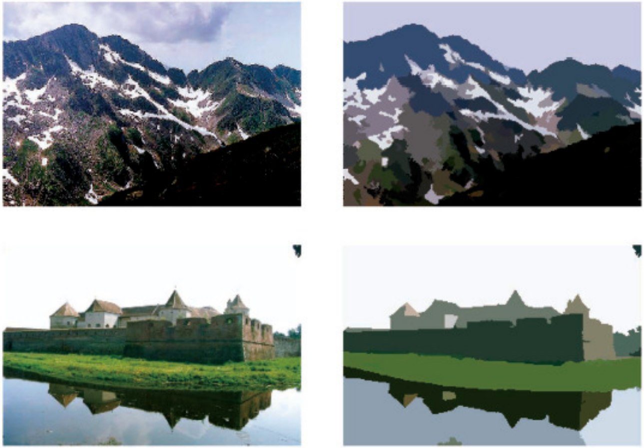
- Determine relevant *features* at each pixel :
  - Color (Lab, HSV...), texture...
  - Position $(x, y) \rightarrow$ ensures spatial consistency !
- Initialize a window at each point (choosing the window size...)
- Apply mean shift on each window until convergence
- Merge trajectories converging to the same mode ("close" peak) to determine attraction basins

Segmentation by Clustering : Mean-Shift Algorithm

Example of segmentation using mean-shift

Segmentation by Clustering : Mean-Shift Algorithm

Example of segmentation using mean-shift



Segmentation by Clustering : Mean-Shift Algorithm

Summary

Advantages

- Does not impose cluster shapes.
- Only 1 parameter (window size).
- Robust to artifacts.
- Many applications : segmentation, filtering, object tracking.

Drawbacks

- Output clusters depend on the window size.
- Computationally expensive.
- Issue with the dimensionality of the feature space.

Segmentation by Clustering : Mean-Shift Algorithm

OpenCV

- Meanshift and Camshift
- `https://docs.opencv.org/4.5.1/d7/d00/tutorial_meanshift.html`

Plan

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Pixel / Region Segmentation

- Thresholding and clustering are operations on pixels.
  - Do not necessarily produce connected regions.
- Region segmentation : preserving connectivity between regions.
  - Using the Divide-and-Conquer principle.
  - Using region growing.

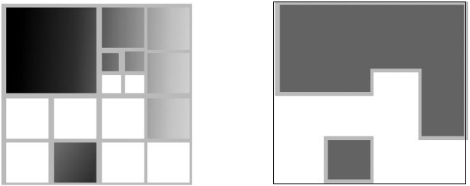
Plan

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Region Segmentation : Divide-and-Conquer

Principle

- *Split and merge* algorithm
- Recursively divide the image into blocks according to a homogeneity predicate, as long as the predicate is not satisfied.
  - Quadtree division : dividing a block results in 4 sub-blocks.
  - The attributes of each sub-block are recalculated.
- Merge adjacent regions for which the union of the pixels satisfies the predicate.



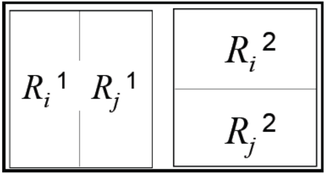
Region Segmentation : Divide-and-Conquer

Division : General Principle

1. Initialization : $R = I$, entire image
2. Application of several divisions δ producing new regions R_i^δ.
3. Choose the subdivision according to the following process : for each set $\{R_i^\delta\}$, count the number of homogeneous sub-regions and select the one with the most.
4. Return to step 2 for each non-homogeneous sub-region.

Region Segmentation : Divide-and-Conquer

Division : Example of Region Partitioning into 2 Parts



Partitions d'une zone : choix entre partition verticale ou horizontale

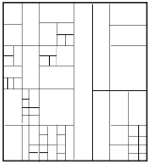
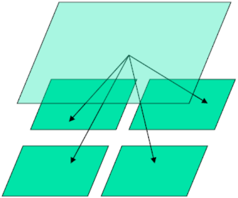


Image partitionnée

Region Segmentation : Divide-and-Conquer

Division Step (Example of the Quadtree)

- The image is stored in a tree (root = full image).
- Each leaf F is recursively subdivided into 4 (quadtree) if it is not sufficiently homogeneous.
- The algorithm continues as long as there are non-homogeneous leaves to divide.



Region Segmentation : Divide-and-Conquer

Region Division or Partition : Example

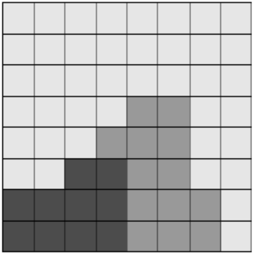
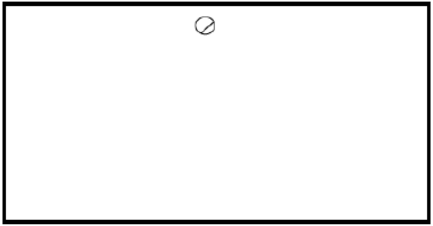


Image originale

Image stockée dans un arbre
Initialement, *arbre racine* = *image complète*



Construction du Quad-tree correspondant

Region Segmentation : Divide-and-Conquer

Region Division or Partition : Example

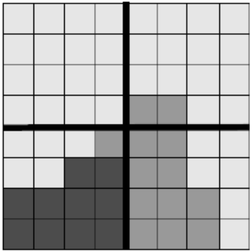
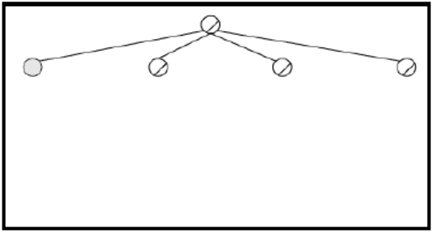


Image originale



Construction du Quad-tree correspondant

Region Segmentation : Divide-and-Conquer

Region Division or Partition : Example

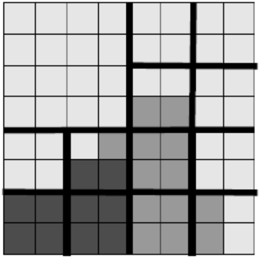
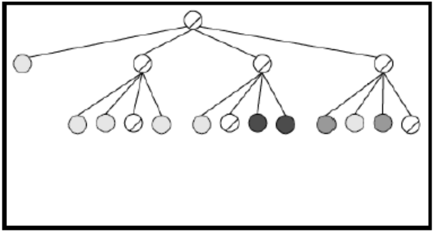


Image originale



Construction du Quad-tree correspondant

Region Segmentation : Divide-and-Conquer

Region Division or Partition : Example

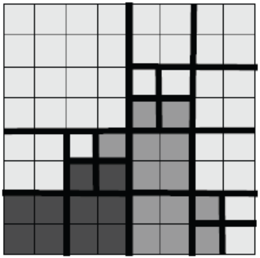
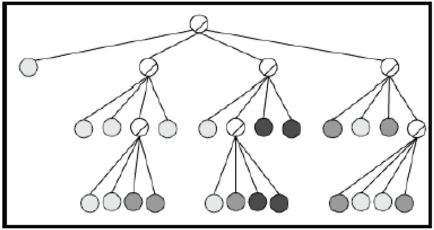


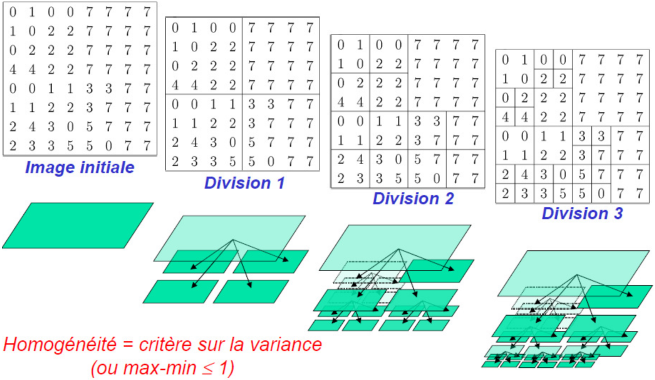
Image originale



Construction du Quad-tree correspondant

Region Segmentation : Divide-and-Conquer

Division Step : Summary



Region Segmentation : Divide-and-Conquer

Segmentation by Division, Partition

Properties

- The geometry of the division directly influences the segmentation result.
- Example : Quadtree method : square regions
- Other types of partitioning exist : triangle, pyramid
- The choice of partitioning depends on the shapes to be segmented.

Region Segmentation : Divide-and-Conquer

Merging : General Principle

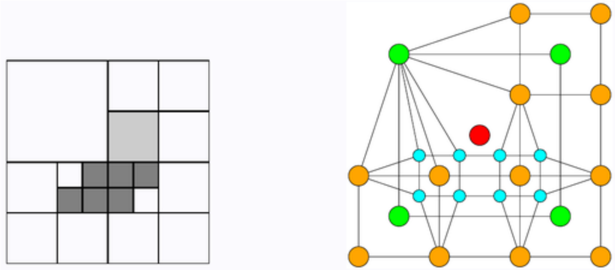
Starting from an initial segmentation obtained by division :

1. Define an adjacency graph where a region is a node and an edge represents an adjacency relation.
2. Define a similarity function between two nodes.
3. Sort all adjacent node pairs in an ordered list.
4. Merge the two best candidates.
5. Update the list and iterate (return to step 3).

Region Segmentation : Divide-and-Conquer

Merging Step : Construction of the *Region Adjacency Graph* (RAG) [Pavlidis, 1977]

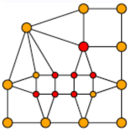
- Each region is a node
- Two regions with a common edge are adjacent.
- Adjacent regions are connected by an edge.
- Each edge is associated with a homogeneity score between the two regions it connects.



Region Segmentation : Divide-and-Conquer

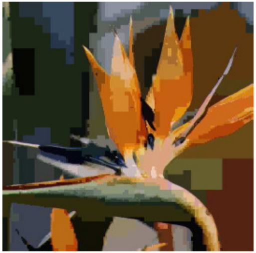
Merging Step :

- Each node of the RAG is examined.
- If one of the neighbors of this node is within a threshold distance for merging, the two nodes merge in the RAG.
- When no node can merge with one of its neighbors, the procedure stops.



Region Segmentation : Divide-and-Conquer

Segmentation Example



Region Segmentation : Divide-and-Conquer

Segmentation Example

Image originale

Split & Merge

Plan

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Region Segmentation : Region Growing

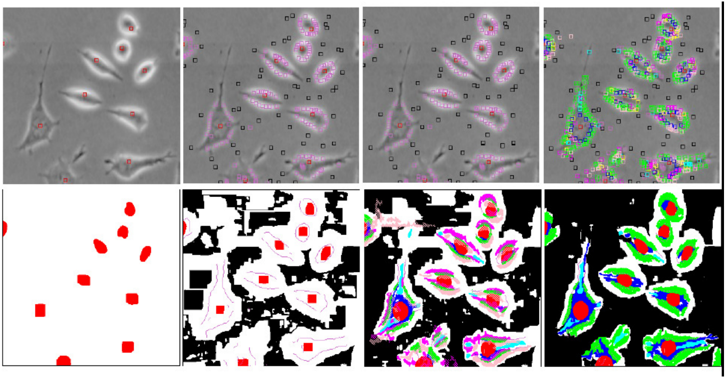
Principle

- Start with a pixel and add neighboring pixels that satisfy a membership criterion :
  - Low variance.
  - Grayscale level meeting a threshold.
  - ...
- The initial pixels are called seeds.
  - Choice of initial pixels can be automatic or semi-automatic (avoiding high-contrast areas).
- The region grows from its seed.
  - A criterion (or predicate) is needed to select the pixels to add.

Region Segmentation : Region Growing

Region Segmentation : Region Growing

Example with multiple seeds.



Region Segmentation : Region Growing

Summary

- Performance is highly dependent on initialization (seeds).
- Often depends on the processing order : the order in which pixels are added to a region influences the result.
- Very simple implementation.
- Fast methods.

Plan

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Plan

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

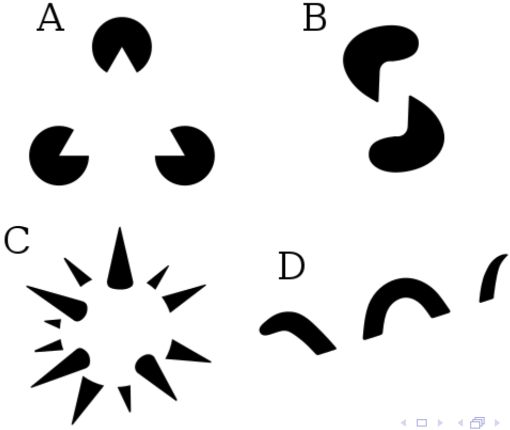
Grouping : Gestalt Approach

Principle

- Another approach to segmentation/grouping inspired by human psychology and the observation of humans and their environment.
- Based on Gestalt studies related to human vision
`http://en.wikipedia.org/wiki/Gestalt_psychology`
`https://www.interaction-design.org/literature/topics/gestalt-principles`
- Importance of context.

Grouping : Gestalt Approach

- Grouping is a key element of visual perception.
- Elements in a collection have properties that arise from relationships.
  - **The whole is greater than the sum of its parts**
  - **Relationships among parts can yield new properties/features**



Grouping : Gestalt Approach

Criteria

- Proximity : nearby objects are grouped.
- Similarity : similar objects are grouped.
- Common fate : objects with similar coherent motion are grouped.
- Common region : objects within the same region are grouped.
- Parallelism : parallel curves or objects are grouped.
- Closure : curves or objects forming closed shapes are grouped.
- Symmetry : curves or objects forming symmetric shapes are grouped.
- Continuity : curves or objects forming a continuous shape are grouped.
- Familiar configuration : curves or objects forming known shapes are grouped.

Grouping : Gestalt Approach

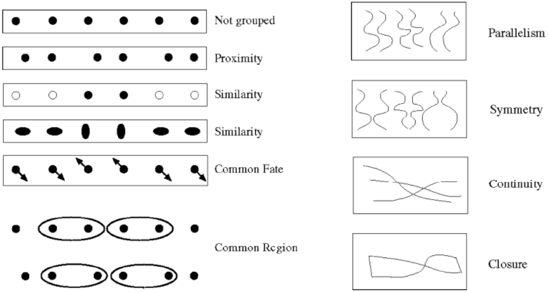


FIGURE – Source : Forsyth & Ponce

Problem : how to integrate these grouping rules into an algorithm ?

Learning Discriminative Models for Segmentation

Ren and Malik. Learning a classification model for Segmentation. ICCV 2003.
`http://ttic.uchicago.edu/~xren/research/iccv2003/`

- Inspired by Gestalt grouping rules to develop a discriminative model for segmentation.
- Idea : Use grouping features to learn what distinguishes good segmentation from bad segmentation.

(a) (b) (c)

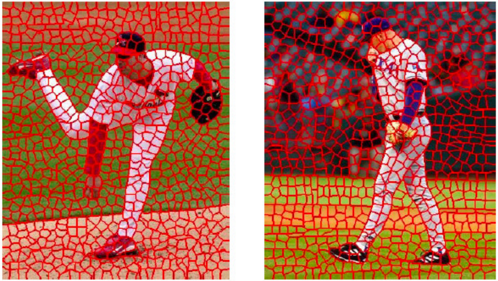
- Good segmentation (provided by a human) (image b) and bad segmentation obtained by random matching between an image and a mask (image c).
- Segmentation problem formulated as a classification problem between good and bad segmentations.

Learning Discriminative Models for Segmentation

Use of the notion of superpixels

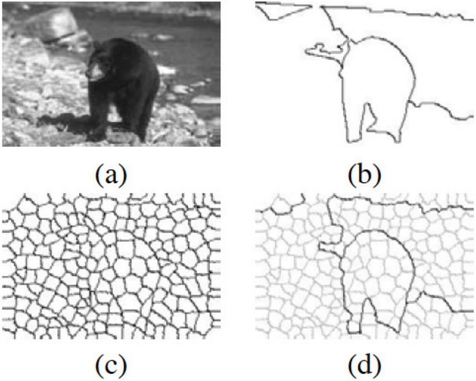
- Entities that have perceptual meaning and are obtained from a low-level grouping process.

"superpixels"



Learning Discriminative Models for Segmentation

Use of superpixels : images are transformed into a map of 200 superpixels and reconstruction of human segmentation.



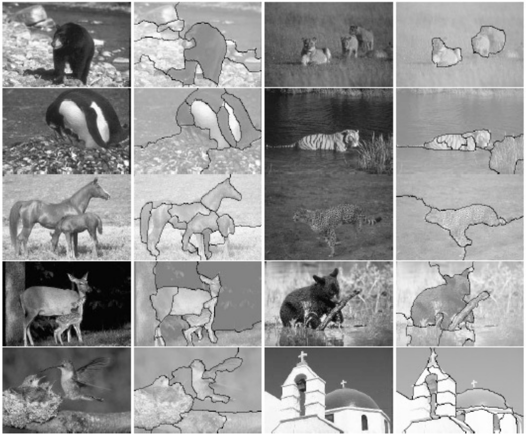
Learning Discriminative Models for Segmentation

Features used for a segment S

- Inter-region contour energy E_{ext}
  Measures the relative contour strength along the segment.
- Intra-region contour energy E_{int}
  Measures the relative contour strength within the segment.
- Inter-region brightness (dis)similarity T_{ext}
  Intensity similarity of a segment with its neighboring segments.
- Intra-region brightness similarity T_{int}
  Homogeneity of intensity within the segment.
- Inter-region texture (dis)similarity B_{ext}
  Texture similarity of a segment with its neighboring segments.
- Intra-region texture similarity B_{int}
  Homogeneity of texture within the segment.
- Curvilinear continuity : C

Learning Discriminative Models for Segmentation

Learning : binary classification problem



Gestalt : In Neural Networks ?

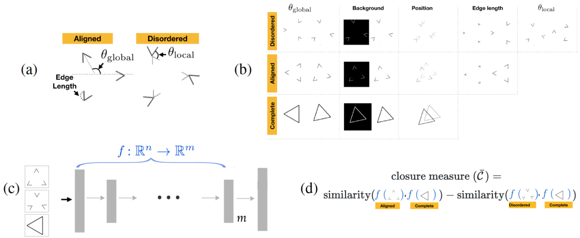


Figure 4: Outline of the stimuli and methodology to test closure in pretrained neural networks. (a) The tested shapes varied in global orientation and the disordered images also varied in local orientation of their elements. (b) Examples of stimulus variation for the three experimental conditions (depicted in the rows), and for five properties (depicted in the columns). (c) Images are fed into a deep ConvNet trained to classify images, where there is one output neuron per class. In most of our simulations, the penultimate layer, with m units is used as a deep embedding of the input image. (d) Computing a closure measure, $\bar{\mathcal{C}}$, where a larger value indicates that the representation of the complete triangle is more similar to the representation of the aligned fragments than to the representation of the disordered fragments. Note that $\bar{\mathcal{C}}$ is an expectation over many image triples, not depicted in the equation.

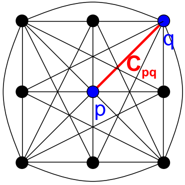
(Kim et al., 2020) Neural Networks Trained on Natural Scenes Exhibit Gestalt Closure [4]

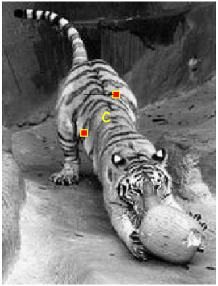
4. https://arxiv.org/pdf/1903.01069.pdf

Outline

1. Segmentation : Introduction
2. Thresholding Segmentation
3. Segmentation by Clustering
   - Feature Space
   - K-means
4. Region Segmentation
   - Divide-and-Conquer
   - Region Growing
5. Other Approaches
   - Grouping
   - Graphs
6. Conclusion

Graph-based approach for segmentation





G = {E,V}

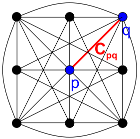
V : vertex
E : edges : liens entre les noeuds

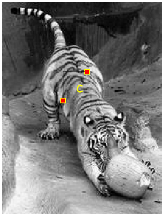
Image = pixels
Similarité entre pixels

Segmentation = partitionnement de graphe

Shi and Malik. Normalized Cuts and Image Segmentation. PAMI 2000.
`http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf`

Graph-based approach for segmentation





G = {E,V}

V : vertex
E : edges : liens entre les noeuds

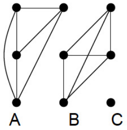
Image = pixels
Similarité entre pixels

Segmentation = partitionnement de graphe

- Each pixel = a node.
- A link between each pair of pixels (or between sufficiently close pairs of pixels).
- Edge weights represent the affinity or similarity between two nodes.

Graph-based approach for segmentation

Segmentation = graph partitioning



- Partitioning the graph into segments.
- Removing edges with low affinity :
  - Similar pixels should belong to the same segments.
  - Dissimilar pixels should be in different segments.

Graph-based approach for segmentation

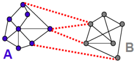
Affinity

- Each pixel is represented by a feature vector $\vec{x}$.
- Defining an appropriate distance in the feature space.
- Converting the distance between two feature vectors into an affinity using a generalized Gaussian kernel.

$$w(i,j) = exp(-\frac{1}{2\sigma^2} dist(\vec{x_i}, \vec{x_j})^2)$$

  - Low σ : groups only very close points.
  - High σ : groups distant points.

Graph-based approach for segmentation



Graph cut

- Set of edges whose removal partitions the graph.
- Cut cost : sum of the weights of the removed edges.

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

  with $A \cap B = \emptyset$

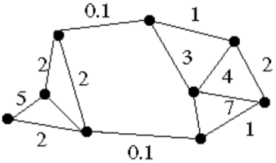
- Solution : minimize $cut(A, B)$!

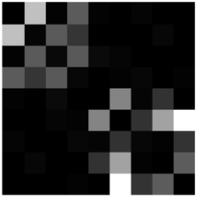
Graph-based approach for segmentation

Minimal cut

Segmentation by searching for the minimal graph cut : various algorithms exist.

Minimum cut example

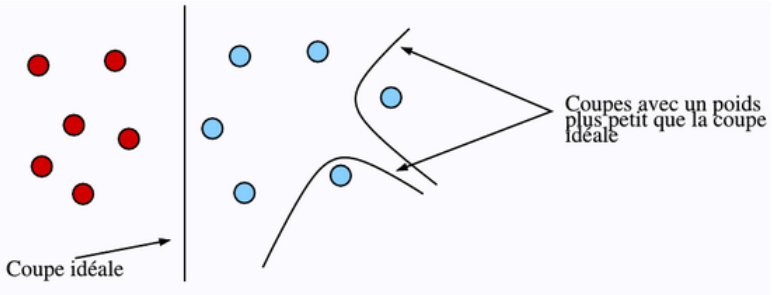




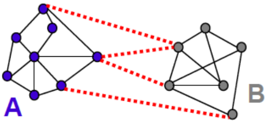
Graph-based approach for segmentation

Minimal cut

Problem : The minimal cut tends to isolate very small components, since $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$ increases with the number of connections between A and B.



Graph-based approach for segmentation



Normalized cut

$w(u, v)$: affinities between nodes u and v
$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

$assoc(A, V) = \sum_{u \in A, v \in V} w(u, v)$
V = full graph = set of vertices.

Numerical solution using matrix diagonalization.

Graph-based approach for segmentation

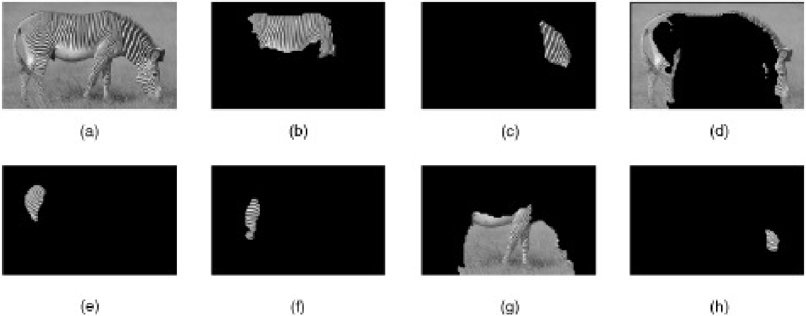
Principle

1. Given an image, construct the weighted graph $G = (V, E)$, where the weight between two nodes measures their similarity.
2. Solve $(D - W)y = \lambda Dy$: smallest eigenvalues.
3. Use the eigenvector corresponding to the second smallest eigenvalue to cut the graph in two.
4. Decide if the partition is sufficient ; otherwise, recursively subdivide the partitions.

Segmentation = global process. The decision on what constitutes a contour is not local.

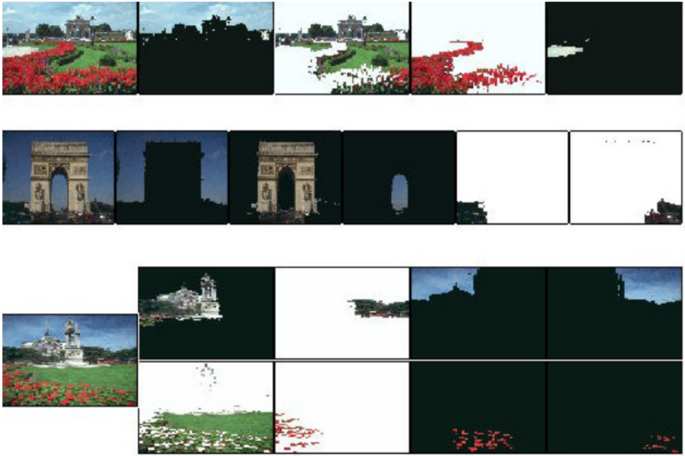
Graph-based approach for segmentation

Example result



Graph-based approach for segmentation

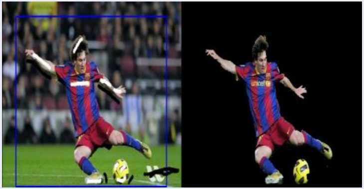
Example result



Graph-based approach for segmentation

OpenCV

GrabCut algorithm : http://docs.opencv.org/trunk/doc/py_tutorials/py_imgproc/py_grabcut/py_grabcut.html



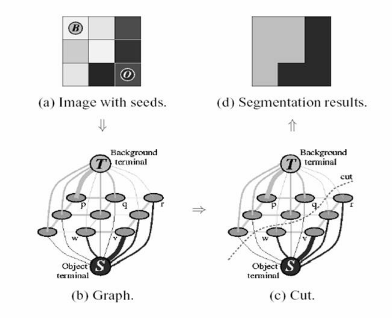


Image Segmentation : Conclusions

- Segmentation involves identifying regions in images.
- Region/contour duality.
- Sometimes useful, sometimes sufficient (sometimes unnecessary).
- Various methods exist (see course outline !).
- No perfect segmentation method.
- Segmentation evaluation is challenging (humans disagree on the "correct" segmentation).
- Defining the purpose of segmentation is key :
  - What exactly are we looking for ?
  - With what level of precision ?
- Warning : Segmentation is not always a prerequisite for recognition : sometimes, good segmentation is recognition !