# Lecture 4: Stat 238. Winter 2014. B. Bonev, A.L. Yuille

January 15, 2014

## 1 Segmentation as Intermediate-level Vision

Image segmentation is the process of dividing an image into multiple segments (sets of pixels, also known as superpixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Meaningful in the sense that each segment will hopefully correspond to a part/subpart of an object (e.g, person, head) or background region (e.g, sky, road). Easier to analyze in the sense that the number of segments will be much smaller than the number of pixels and each segment would provide a richer description than a single pixel does.

In this lecture we consider segmentation as an intermediate-level computer vision technique. Segmentation goes beyond low-level vision by grouping or dividing regions based on both local and global image properties. It is based on natural properties of the images, such as the assumption that there exist smooth or roughly homogeneous regions and they are compact to some extent. We consider a segmentation method which does not use any object-specific information. Thus, the segments produced serve as an input for a high-level computer vision method. For example, they can be used as candidate regions for object recognition techniques, see Figure 1.

Segments can be both connected or disconnected, and they can form a single partition of the image or they can overlap, if there are multiple hypothesis about the image segmentation. This is the case of hierarchical image segmentation methods, which produce different segmentation levels with increasing (or decreasing) number of segments in each partition.

## 2 Region-based Image Segmentation

There are two main ways to describe and simplify an image: by finding the edges (rapid changes) which separate different regions, or by grouping together the pixels which belong to roughly homogeneous regions, that is, regions which have similar image properties (e.g, colors or texture). Both ways are complementary. Following we describe a simple region-based segmentation method which optionally makes use of edge information. Apart from the described method, many other related methods are available, for instance, Segmentation by Weighted Aggregation (Sharon et al. 2006).

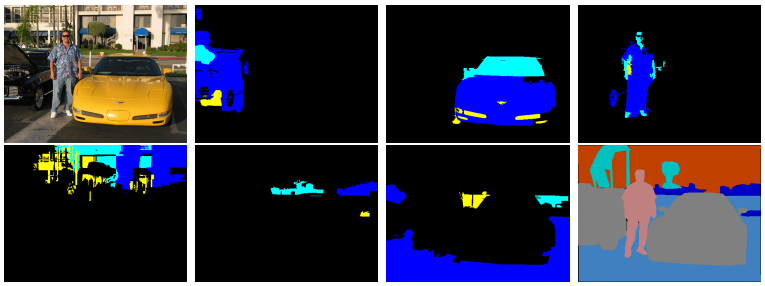


Figure 1: Examples of candidate regions for objects and background regions. Left-to-right and top-to-bottom: image, top three (disconnected) segments for left car, right car, person, building, grass, ground, and the ground truth. Most objects are covered well by two to three segments (except building) which can be used as candidate regions for object recognition.

Segmentation is a particular case of data clustering. As such, the segmentation processes can be classified in those which iteratively divide the image, those which group small segments into bigger ones, and those which do both. The method we describe is based on greedy iterative grouping of regions. Greedy refers to the inability to undo a grouping decision. The starting point of such algorithm could be the raw pixels but they, by themselves, are unable to capture texture properties. Instead, the method starts from small elementary segments produced by the SLIC algorithm.

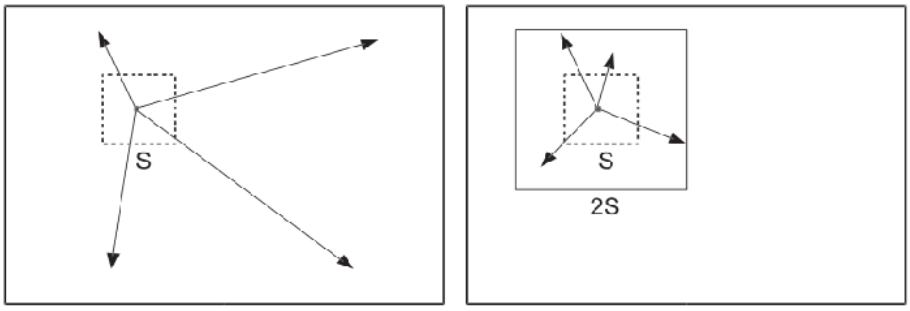
## 3 Simple Linear Iterative Clustering

The Simple Linear Iterative Clustering (SLIC) method, introduced by (R. Achanta et al., PAMI 2012) partitions the image into a set of small segments (or superpixels) which are roughly uniform and have similar sizes. While there are many other methods producing similar outputs, SLIC is simple and fast, and requires the selection of a single parameter.

This algorithm clusters (by k-means) the 5-dimensional $[labxy]$ space, where $lab$ refer to the three components of the CIELAB color space ($l$ is lightness, $a$ and $b$ are color-opponent dimensions) and $xy$ are the pixel coordinates in the image. The method enforces color similarity as well as pixel proximity in this 5D space.

For a desired number $K$ of approximately equally-sized superpixels and an image with $N$ pixels, the approximate size of each super- pixel is therefore $N/K$ pixels. For roughly equally sized superpixels there would be a superpixel center at every grid interval $S = \sqrt{N/K}$. The algorithm starts by setting the cluster centers at regular grid intervals $S$. It is assumed that the pixels associated with a cluster lie within a $2S \times 2S$ area around the center, this is the search area (Figure 2.

Instead of using just an Euclidean distance in the 5D space, a new distance measure $D_s$ is defined in order to control how important is the spatial position




*(a)* standard $k$-means searches the entire image

*(b)* SLIC searches a limited region

Figure 2: Search space for the k-means algorithm. (R. Achanta et al., PAMI 2012)

$x, y$ with respect to $l, a, b$.

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \quad (1)$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (2)$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \quad (3)$$

The greater the value of $m$, the more spatial proximity is emphasized and the more compact the cluster. This value is between 1 and 20 and a reasonable choice is 10.

First, the $K$ cluster centers (seed locations) are moved within a 3 pixels neighbourhood to the lowest gradient position, to avoid placing them at an edge or noisy pixel. Image gradients are computed as

$$G(x, y) = ||\mathbf{I}(x + 1, y) - \mathbf{I}(x - 1, y)||^2 + ||\mathbf{I}(x, y + 1) - \mathbf{I}(x, y - 1)||^2, \quad (4)$$

where $\mathbf{I}(x, y)$ is the $lab$ vector corresponding to the $(x, y)$ pixel.

Then each pixel in the image is associated with the nearest cluster in the search area. After all the pixels are associated with the nearest cluster center, a new center is computed as the average $labxy$ vector of all the pixels belonging to the cluster. Then the process of associating pixels with the nearest cluster center and recomputing the cluster center is iterated until convergence.

The described process does not guarantee connectivity. A postprocess may be applied in order to relabel the few disconnected regions that may be produced.

See sample results of the SLIC algorithm in Figure 3. This segmentation simplifies the image by providing a set of roughly homogeneous segments, respecting most of the edges in the image.

# 4 Hierarchical grouping algorithm

The elementary segments provided by SLIC are an over-segmentation of the image. They are a good starting point for grouping regions with similar statistical properties. By iteratively grouping regions, a hierarchy of segments is produced,

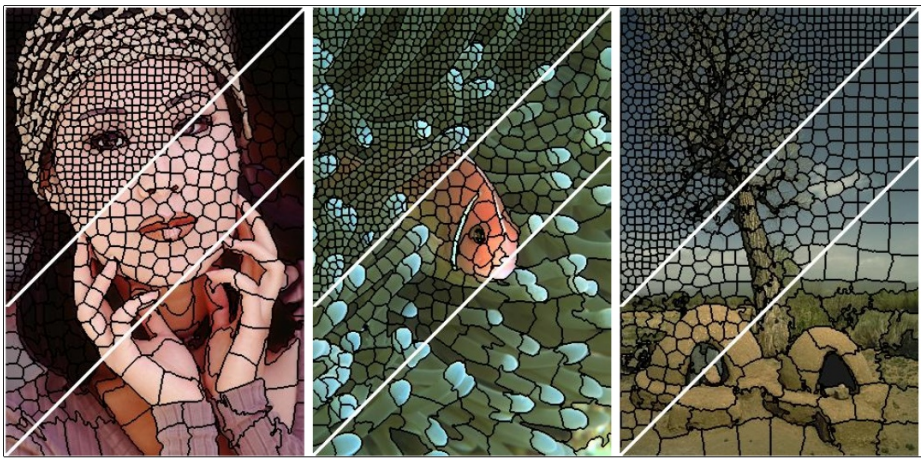


Figure 3: Image segmented into SLIC superpixels of (approximate) size 64, 256, and 1024 pixels. (R. Achanta et al., PAMI 2012)

where the first level is the output of SLIC and the last level is a single segment containing the complete image. See Figure 4.

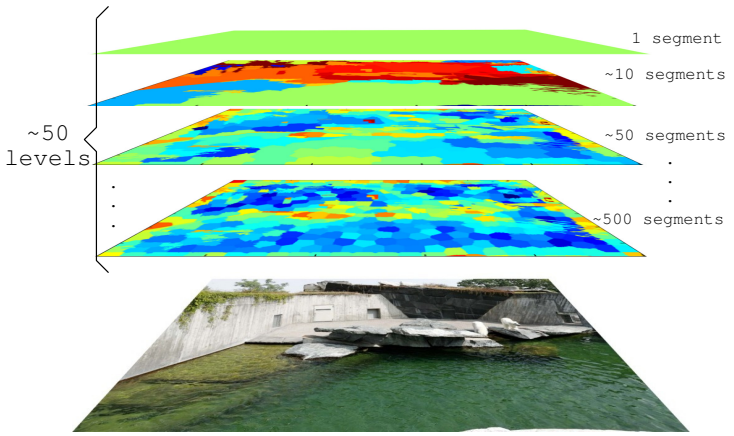



Figure 4: Multiple segmentation levels in a hierarchy. Segments with a good coverage of objects or parts may happen at different levels.

The algorithm starts with the basic set of elementary segments $\mathcal{S}_1$ which are produced by SLIC. Then it builds a segmentation hierarchy $\mathcal{S}_\mathcal{H} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_L$ of $L$ levels by merging segments as follows.

Each segment is described by an *appearance vector*

$$V = (\overline{\mu}, \overline{\sigma}, c_x, c_y, w, h), \tag{5}$$

where $\overline{\mu}, \overline{\sigma}$ are the mean and the standard deviation of the Lab color space components and the first and second gradients, $(l, a, b, \nabla_x, \nabla_y, \nabla_x^2, \nabla_y^2)$. Here $(c_x, c_y, w, h)$ are the centroid of the segment and the dimensions of its bounding

box. These appearance vectors are designed so that they can be computed efficiently recursively for new segments composed by merging.

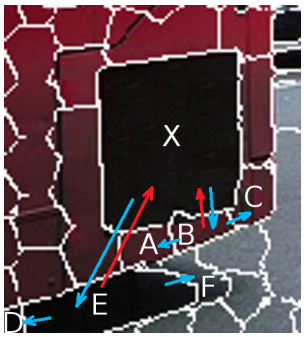



Figure 5: Asymmetric dissimilarity function. The nodes that are less dissimilar to X are B and E (second neighbor) because they don't modify a lot the statistics of X. However, X modifies a lot the statistics of B and E. Both of these nodes have other nodes to which they are much more similar.

We define an asymmetric dissimilarity function $\Delta^A_{i|j}$ between segments which are $1^{st}$ or $2^{nd}$-order neighbors. The appearance term is defined to be

$$\Delta^A_{i|j} = ||V_i - V_{i \bigcup j}||_2$$

This is the change in the appearance vector of region $i$ caused by merging it with region $j$. This function is asymmetric – i.e. $\Delta^A_{i|j} \neq \Delta^A_{j|i}$, see Fig 5. This term will encourage merging neighboring regions which have similar appearance vectors. The asymmetry has the meaning that after merging $i$ and $j$, the statistics of $i$ may be modified in a different degree than the statistics of $j$. Intuitively, each segment has its own preference for merging or not to a neighbour, and it is based on minimizing the change of appearance.

The appearance dissimilarity function is modified by an edge-term ($E_{i,j} \in [0, 1]$) that represents the amount of edge-ness on the boundary between two adjacent regions. This edge term is computed only once. Any fast edge detector can be used: Sobel detector performs well, but slight improvements (1.7%) are obtained by using the Sketch-Token method (Lim, Zitnick, Dollár CVPR 2013). The intuition for this edge modulation is that we penalize the similarity between adjacent regions if there is an edge between them. There is no edge-term between segments in the $2^{nd}$-order neighborhood (this type of merging is allowed to jump between regions) and instead there is a fixed penalty of size 1, which is the maximum value the edge term can take.

This gives an asymmetric dissimilarity function $\Delta_{i|j}$:

$$\begin{aligned} \Delta_{i|j} &= E_{i,j} + \Delta^A_{i|j}, \text{ if } i, j \text{ are } 1-\text{neighbors}, \\ \Delta_{i|j} &= 1 + \Delta^A_{i|j}, \qquad \text{if } i, j \text{ are } 2-\text{neighbors}. \end{aligned} \tag{6}$$

In order to allow some regions to grow "faster" and other to remain small accross different levels of the hierarchy, not all the segments are merged at each

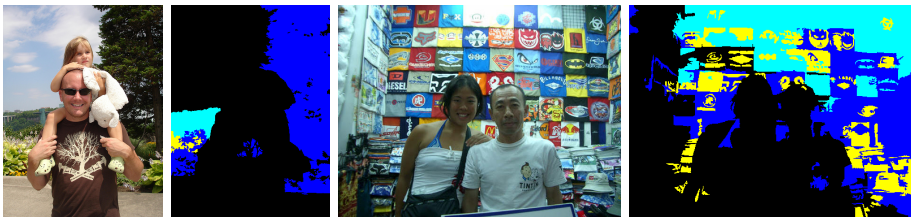


Figure 6: Left: example of vegetation textures captured by a single segment. Right: a coarser texture which failed to be merged in a single segment.

level. Only a fraction of them is merged (a 30%), and the decision of which ones to merge is taken by a ranking approach explained in the next section. These top-ranked segments $i$ are merged to their less dissimilar neighbor $j$, unless they violate the condition $\Delta_{i|j} > 0.9\Delta_{j|i}$. The intuition is that this ranking encourages merging between segments which are most similar, but that merges are rejected in situations where the dissimilarity function between two regions is too asymmetric.

At each level the same procedure is followed: evaluating the dissimilarity function between 1st and 2nd neighbors, ranking them and merging the top-ranked 30% of them. The algorithm finishes when there is only one segment containing the whole image, $\mathcal{S}_L$, producing a hierarchy $\mathcal{S}_\mathcal{H}$ as illustrated in Figure 4.

The algorithm does not have an explicit mechanism for handling textures. However it usually merges textures into a single segment. In the case of textures which are finer than the size the first level $\mathcal{S}_1$ segments, the texture characteristics are captured by the mean and standard deviation of the segments and they are likely to be merged, see Fig. (6)-left. In the case of very coarse textures several big and disconnected segments may be formed, see Fig. (6)-right.

# 5 Segments Ranking by PageRank

This grouping approach requires a criterion for merging segments and a procedure for selecting the order of merging. Given the asymmetric similarity measure between segments (here nodes), a directed graph is defined. Over this graph, the PageRank algorithm (Franceschet, ACM 2011) is used to determine the order of merging.

The use of PageRank is motivated by the desire to start by merging those segments which have similar statistics to their neighbors. PageRank is illustrated in figure (7). It plays a key role in ensuring that our segments take *global properties* of the image into account when merging segments to construct the hierarchy. At each level $l$ of the hierarchy, it selects which segments should be allowed to merge, by taking into account the whole *directed* graph, where the nodes are the segments in $\mathcal{S}_l$ and the weights of the directed edges are given by the asymmetric dissimilarity (6). This strategy also allows some segments to grow "faster", that is, the segments $\mathcal{S}_l$ of a level $l$ may have different sizes. In that way $l$ is not directly related to the size of the segments, as the appearance also determines the size of the segments. In Fig. (8) it can be seen that small

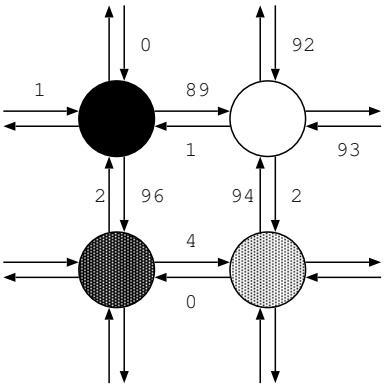



Figure 7: The PageRank algorithm prioritizes merging segments which are very similar to their neighbors. In this example, the white node (representing a segment) has highest ranking and so is selected first for merging. The edge weights denote (asymmetric) similarity between the segments.

salient areas can "survive" together with large segments covering homogeneous areas. This is an advantage when searching for region candidates for objects of different sizes and with large background regions.

More formally, PageRank quantifies the importance of each segment (i.e. graph node) after a sequence of probabilistic transitions over the graph. These probabilistic transitions are encoded by a stochastic matrix. At level $l$, $W_l$ is a stochastic matrix of size $N_l$ where each element is given by the similarity from node $i$ to $j$, normalized by rows so that the outgoing edges form a probability distribution, $\forall i,\ \sum_j^{N_l} w_{ij} = 1,\ i,j \in [1, N_l]$. Given that we work with the dissimilarity $\Delta_{i|j}$ further defined in (6), we define $w_{ij} = 1/(\epsilon + \Delta_{i|j} \sum_{j=1}^{N_l} \Delta_{i|j})^{-1}$ where $\epsilon$ is a constant close to 0 for numerical stability. PageRank returns a ranking of the nodes and we select those nodes with the highest ranks for merging. The merging order over the nodes $i$ does not determine to which neighbor $j$ to merge them. This decision is taken by using the dissimilarity measure $\Delta_{i|j}$ (6).

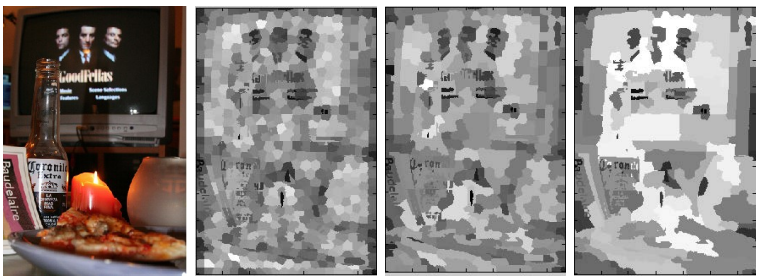


Figure 8: PageRank: at each level we merge first the highest rank (whitest) nodes. Note that the "salient" or different regions have lower rank, while the segments covering homogeneous areas tend to have higher ranks.

# 6 Segments selection

Given a segmentation hierarchy it is difficult to select a single segmentation level as the most appropriate one. Meaningful structures may appear at different levels, as illustrated in Figure 9.

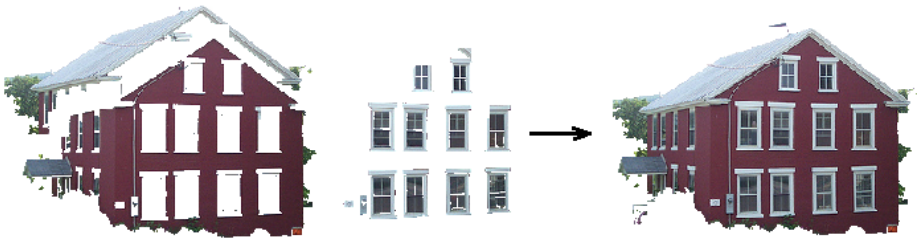


Figure 9: Candidate regions at different scales. The walls of the house (left panel). Each of the windows (center panel). The whole house (right panel).

However, having a single partition of the image may be necessary in some applications. Also, the whole hierarchy contains too many segments, and limiting them to a number of selected segments facilitates further processing of the image.

There are two desirable properties that the selected segments should have: a) be as big as possible while b) being roughly homogeneous. As big as possible means that a big region like a sky is preferred to be described by a single segment instead of many elementary segments. Being roughly homogeneous means that the region statistics should be easy to describe. Not all objects are roughly homogeneous and this property implies that some objects will be composed by several segments. See Figure 10.

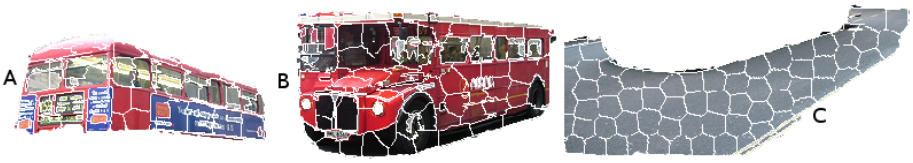



Figure 10: Segments A and B are not homogeneous while C is roughly homogeneous because there are few changes between the statistics of the elementary segments that compose it.

In order to measure the amount of change in a segment $S$, the appearance vectors $V_i$ (5) of the elementary segments $s_i \in S$ can be used. Similarly to (5), here the appearance vector is $V = (\overline{\mu}, \overline{\sigma})$, where $\overline{\mu}, \overline{\sigma}$ are the mean and the standard deviation of $(l, a, b, \nabla_x, \nabla_y, \nabla_x^2, \nabla_y^2)$ (but the $x, y, w, h$ are not included). Imposing a maximum difference threshold $t$ between all pairs of segments $||V_i - V_j|| < t, \forall i, j \in S$ is too restrictive and disallows big segments like sky, which may present gradual changes. A less restrictive criterion is to thresh-

old the differences between 1st and 2nd-order neighbors within the segment:

$$||V_i - V_j|| < t,\ \forall i, j \in S,\ d_G(i, j) \leq 2, \tag{7}$$

where $d_G(i, j)$ is the graph distance between $i, j$ and it has to be 1 or 2, that is, 1st or 2nd neighbors.

Note that this criterion imposes a smooth variation in the statistics of a segment, which makes it possible to describe the segment with linear or polynomial approximations.

In Figure 11 there are two different partitions output by setting two different distance thresholds.

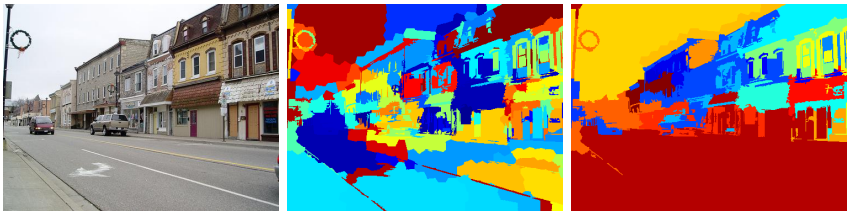


Figure 11: Two partitions resulting from two different threshold values.