

Deep Learning for Computer Vision

Image Segmentation

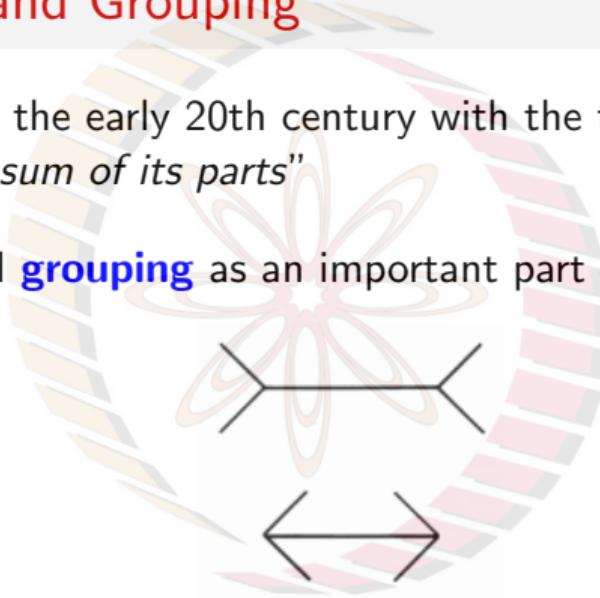
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad



Human Vision: Gestalt and Grouping

- Gestalt theory emerged in the early 20th century with the following main belief: "*The whole is greater than the sum of its parts*"
- Gestalt theory emphasized **grouping** as an important part of understanding human vision.



- The famous Muller-Lyer illusion above illustrates the Gestalt belief - humans tend to see things as groups and not individual components.

Source: David Forsyth

Human Vision: Gestalt and Grouping

- Gestalt theory proposes various factors in images which can lead to grouping:



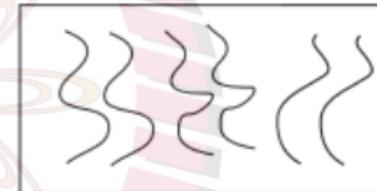
Not grouped



Proximity



Similarity



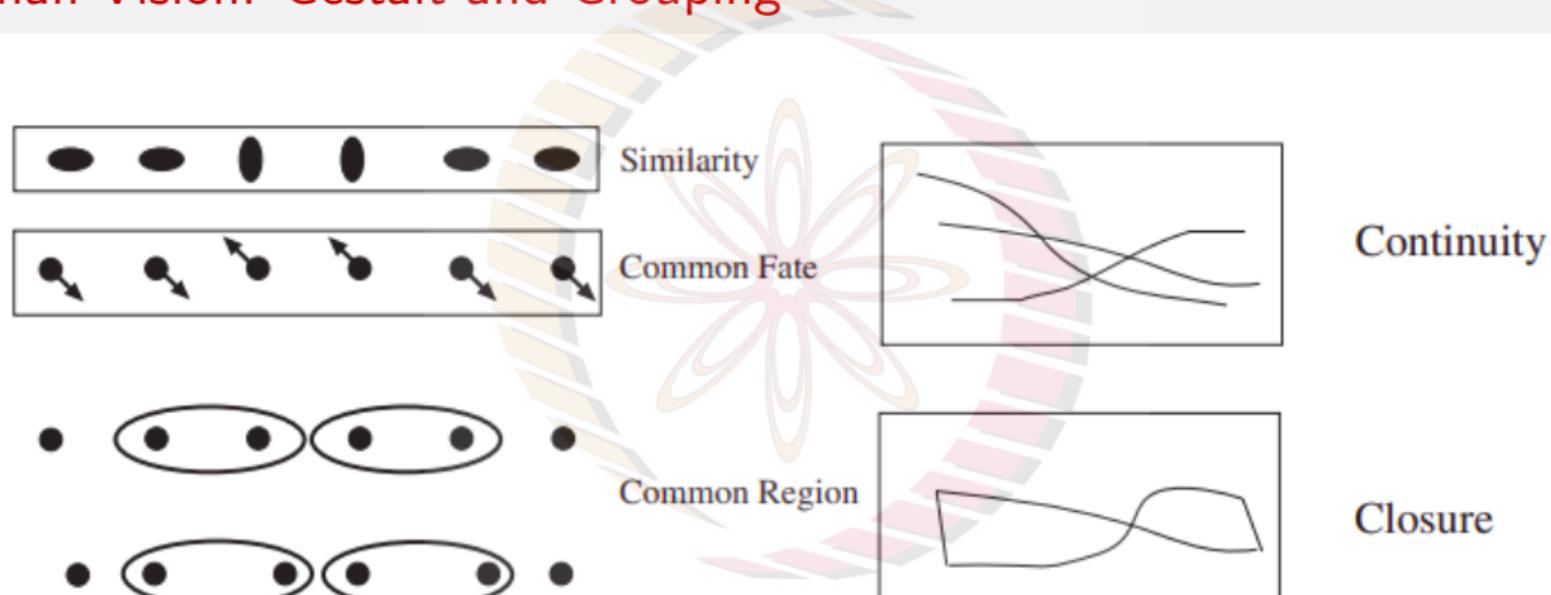
Parallelism



Symmetry

Credit: David Forsyth

Human Vision: Gestalt and Grouping



NPTEL

Credit: David Forsyth

Human Vision: Gestalt and Grouping

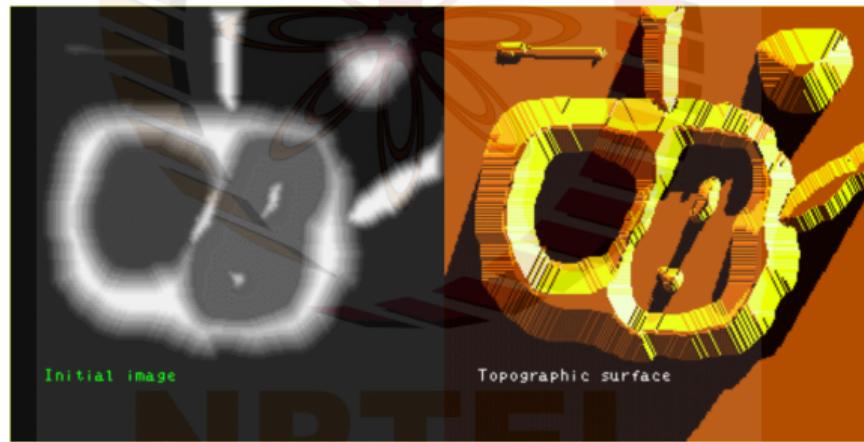
- Gestalt theory is fairly descriptive - loose set of rules to explain why some elements can be grouped together in an image
- However, rules are insufficiently defined to be directly used to form algorithmic tools for grouping objects in images

Further Reading

- Chapter 15.1, Forsyth, *Computer Vision: A Modern Approach*

Watershed Segmentation Method

- An early method for image segmentation (1979)
- Segments an image into several "catchment basins" or "regions"
- Any grayscale image can be interpreted as a 3D topological surface

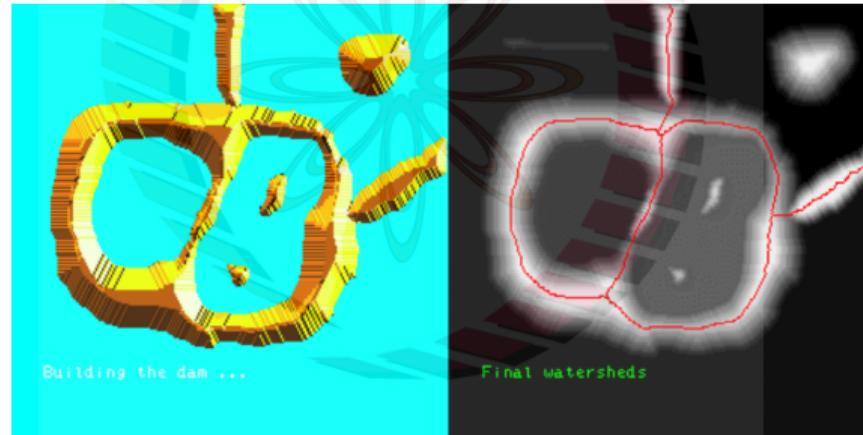


- Image can be segmented into regions where rainwater would flow into the same lake

Credit: S Beucher

Watershed Segmentation Method

- Flood the landscape from local minima and prevent merging of water from different minima

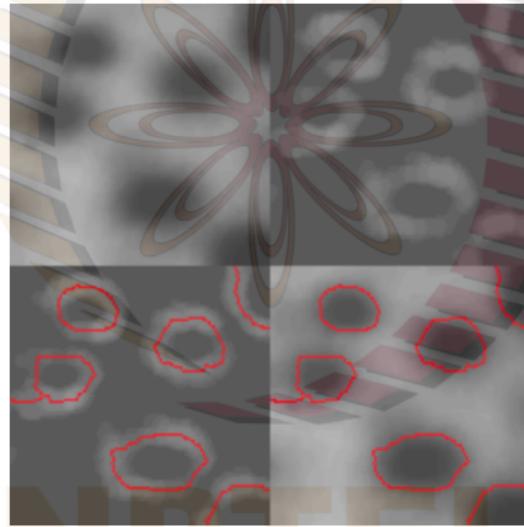


- Results in partitioning the image into **catchment basins** and **watershed lines**

Credit: S Beucher

Watershed Segmentation Method

Generally applied on image gradients instead of applying directly on images

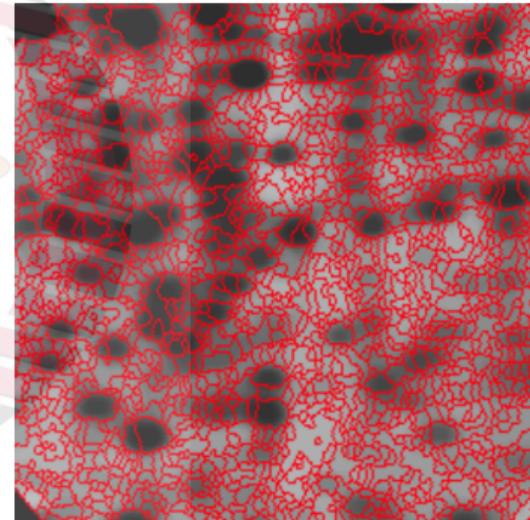


(Top left) Original image; (Top right) Gradient image; (Bottom left) Watersheds of gradient image; (Bottom right) Final segmentation output

Credit: S Beucher

Watershed Segmentation Method

- In practice, often leads to over-segmentation due to noise and irregularities in image
- Hence usually used as part of an interactive system, where user marks "centers" of each component, on which flooding is done



Further Reading

- Chapter 5.2.1, Szeliski, *Computer Vision: Algorithms and Applications*

Categories of Methods: Region Splitting and Merging

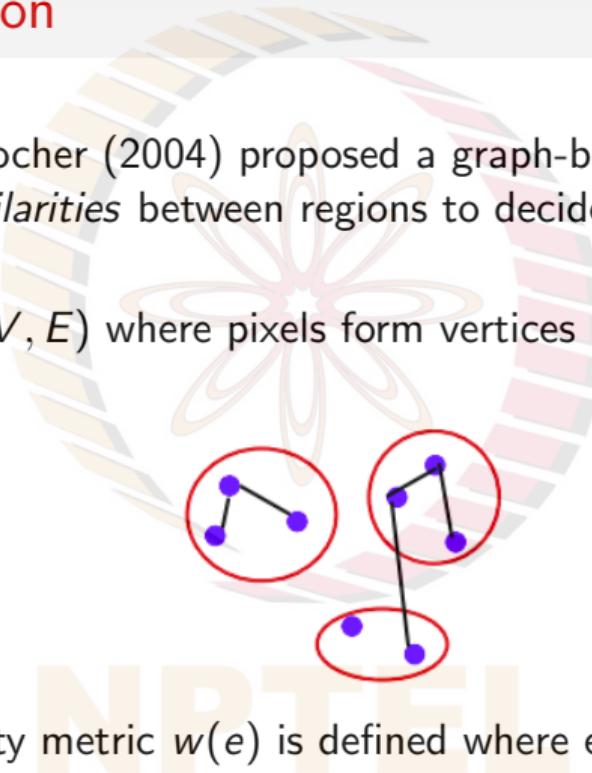
- **Region splitting methods** involve splitting the image into successfullly finer regions.
 - We'll discuss one such method in the upcoming slides
- **Region merging methods** successively merge pixels into groups based on various heuristics such as color differences
 - Figure on right shows an image segmented into such *superpixels*
 - Generally used as preprocessing step to higher-level segmentation algorithms



Image Credit: Achanta et al. SLIC Superpixels

Graph-based Segmentation

- Felzenszwalb and Huttenlocher (2004) proposed a graph-based segmentation algorithm which uses *relative dissimilarities* between regions to decide which ones to merge (region-merging method)
- An image = graph $G = (V, E)$ where pixels form vertices V and edges E lie between adjacent pixels



- A pixel-to-pixel dissimilarity metric $w(e)$ is defined where edge $e = (v_1, v_2)$ and v_1, v_2 are two pixels. This measures, for instance, intensity differences between N_8 neighbors.

Graph-based Segmentation

- For a region C , its **internal difference** is defined as the largest edge weight in the region's minimum spanning tree:

$$\text{Int}(C) = \max_{e \in MST(C)} w(e)$$

NPTEL

Graph-based Segmentation

- For a region C , its **internal difference** is defined as the largest edge weight in the region's minimum spanning tree:

$$\text{Int}(C) = \max_{e \in \text{MST}(C)} w(e)$$

- The **minimum internal difference** between two adjacent regions is defined as ($\tau(C)$ is a manually chosen region penalty):

$$\text{MInt}(C_1, C_2) = \min \left(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2) \right)$$

NPTEL

Graph-based Segmentation

- For a region C , its **internal difference** is defined as the largest edge weight in the region's minimum spanning tree:

$$\text{Int}(C) = \max_{e \in \text{MST}(C)} w(e)$$

- The **minimum internal difference** between two adjacent regions is defined as ($\tau(C)$ is a manually chosen region penalty):

$$\text{MInt}(C_1, C_2) = \min \left(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2) \right)$$

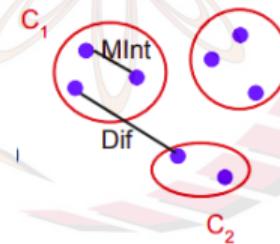
- For any two adjacent regions with at least one edge connecting their vertices, difference between these two regions = minimum weight edge connecting these two regions

$$\text{Dif}(C_1, C_2) = \min_{e=(v_1, v_2) | v_1 \in C_1, v_2 \in C_2} w(e)$$

Graph-based Segmentation

- A predicate $D(C_1, C_2)$ for any two regions C_1 and C_2 is defined as:

$$D(C_1, C_2) = \begin{cases} \text{true,} & \text{if } \text{Dif}(C_1, C_2) > \text{MInt}(C_1, C_2) \\ \text{false,} & \text{otherwise} \end{cases}$$

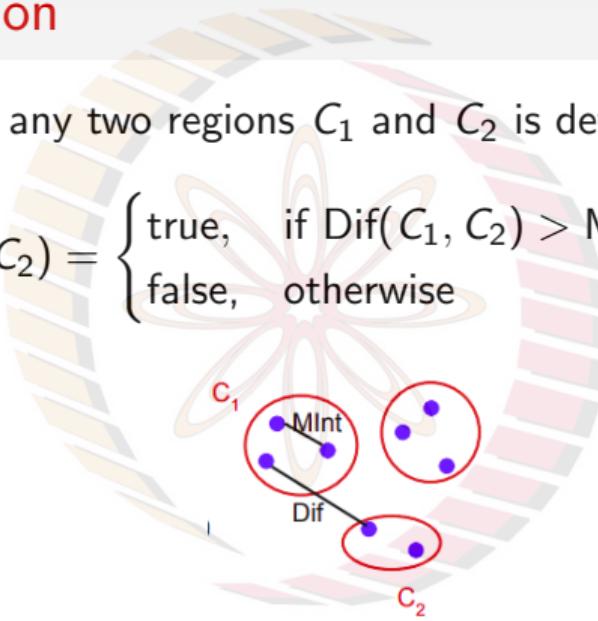


NPTEL

Graph-based Segmentation

- A predicate $D(C_1, C_2)$ for any two regions C_1 and C_2 is defined as:

$$D(C_1, C_2) = \begin{cases} \text{true,} & \text{if } \text{Dif}(C_1, C_2) > \text{MInt}(C_1, C_2) \\ \text{false,} & \text{otherwise} \end{cases}$$



- For any two regions, if the predicate D evaluates to **false**, regions are merged. Else, regions are considered separate.
- \Rightarrow This algorithm merges any two regions whose difference is smaller than minimum internal difference of these two regions.

Graph-based Segmentation



Graph-based merging segmentation using N_8 pixel neighborhood

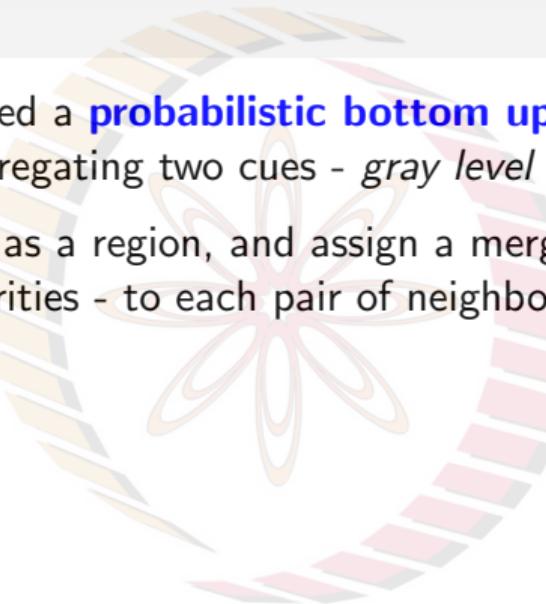
Further Reading

- Chapter 5.2.4, Szeliski, *Computer Vision: Algorithms and Applications*

NPTEL

Probabilistic Aggregation

- Alpert *et al.* (2007) proposed a **probabilistic bottom up merging algorithm** for image segmentation based on aggregating two cues - *gray level similarity* and *texture similarity*
- Initially consider each pixel as a region, and assign a merging likelihood p_{ij} - based on intensity and texture similarities - to each pair of neighboring regions



NPTEL

Probabilistic Aggregation

- Alpert *et al.* (2007) proposed a **probabilistic bottom up merging algorithm** for image segmentation based on aggregating two cues - *gray level similarity* and *texture similarity*
- Initially consider each pixel as a region, and assign a merging likelihood p_{ij} - based on intensity and texture similarities - to each pair of neighboring regions
- Given a graph $G^{[s-1]} = (V^{[s-1]}, E^{[s-1]})$, $G^{[s]}$ is constructed by selecting subset of seed nodes $C \subset V^{[s-1]}$, we merge nodes/regions if they are *strongly coupled* to regions in C . Strong coupling is defined as:

$$\frac{\sum_{j \in C} p_{ij}}{\sum_{j \in V} p_{ij}} > \text{threshold} \quad (\text{usually set to 0.2})$$

NPTEL

Probabilistic Aggregation

- Alpert *et al.* (2007) proposed a **probabilistic bottom up merging algorithm** for image segmentation based on aggregating two cues - *gray level similarity* and *texture similarity*
- Initially consider each pixel as a region, and assign a merging likelihood p_{ij} - based on intensity and texture similarities - to each pair of neighboring regions
- Given a graph $G^{[s-1]} = (V^{[s-1]}, E^{[s-1]})$, $G^{[s]}$ is constructed by selecting subset of seed nodes $C \subset V^{[s-1]}$, we merge nodes/regions if they are *strongly coupled* to regions in C . Strong coupling is defined as:

$$\frac{\sum_{j \in C} p_{ij}}{\sum_{j \in V} p_{ij}} > \text{threshold} \quad (\text{usually set to 0.2})$$

- Once a segmentation is identified at a coarser level, assignments are propagated to their finer level "children", followed by further coarsening

Credit: Szeliski

Probabilistic Aggregation

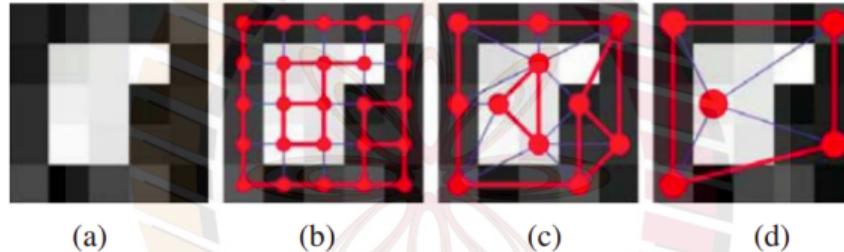


Figure 5.15 Coarse to fine node aggregation in segmentation by weighted aggregation (SWA) (Sharon, Galun, Sharon *et al.* 2006) © 2006 Macmillan Publishers Ltd [Nature]: (a) original gray-level pixel grid; (b) inter-pixel couplings, where thicker lines indicate stronger couplings; (c) after one level of coarsening, where each original pixel is strongly coupled to one of the coarse-level nodes; (d) after two levels of coarsening.

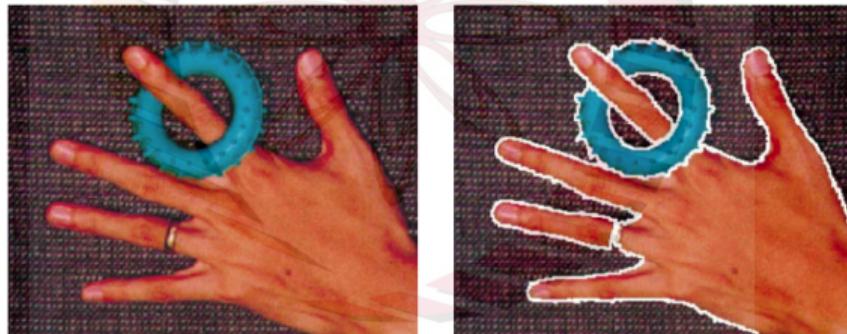
Further Reading

- Chapter 5.2.5, Szeliski, *Computer Vision: Algorithms and Applications*

Image Credit: Szeliski

Mean Shift Segmentation

- A mode-finding technique based on non-parametric density estimation
- Feature vectors of each pixel in the image are assumed to be samples from an unknown probability distribution

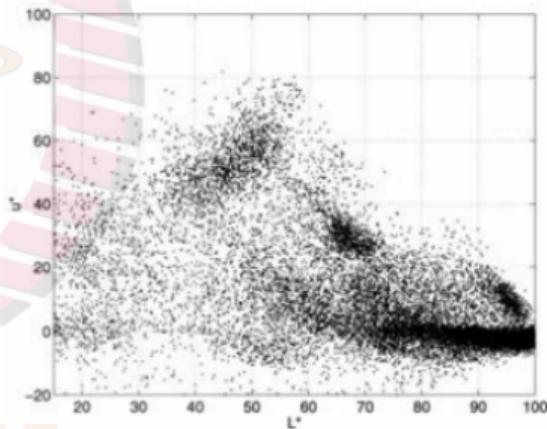


- We estimate p.d.f. using non-parametric estimation and find its *modes*.
- Image is segmented pixel-wise by considering every set of pixels which climb to the same mode as a consistent segment.

Image Credit: Szeliski

Mean Shift: Example

- Consider an example image below on the left. The graph on the right shows the distribution of $L^* u^*$ features of each pixel (in the $L^* u^* v^*$ /CIELUV space¹)



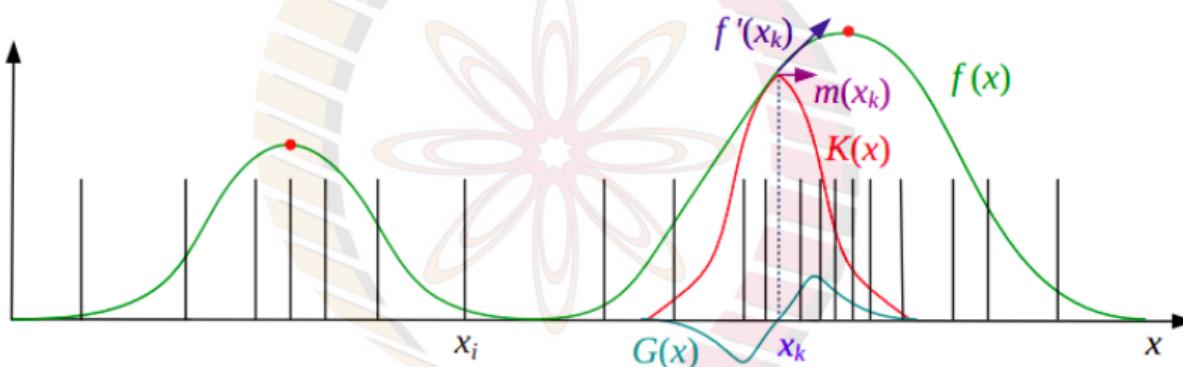
- Our aim is to obtain modes of distribution on the right, without actually explicitly computing the density function! How to do this?

Image Credit: Comaniciu and Meer

¹<https://en.wikipedia.org/wiki/CIELUV>

Mean Shift: Example

- 1D visualization as an example, to illustrate the mode finding approach.



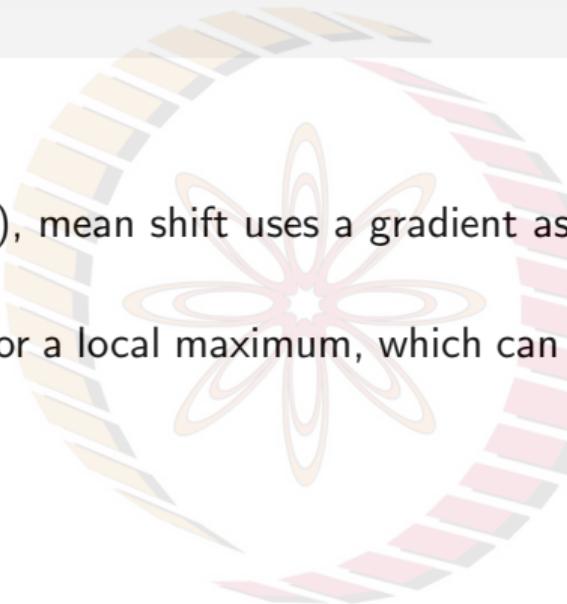
- Estimate the density function by convolving the data with kernel of width h , where k is the kernel function:

$$f(\mathbf{x}) = \sum_i k\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right)$$

Image Credit: Szeliski

Mean Shift: Example

- To find the modes (peaks), mean shift uses a gradient ascent method with multiple restarts.
- First, we pick a guess y_0 for a local maximum, which can be a random input data point x_i .



NPTEL

Mean Shift: Example

- To find the modes (peaks), mean shift uses a gradient ascent method with multiple restarts.
- First, we pick a guess \mathbf{y}_0 for a local maximum, which can be a random input data point \mathbf{x}_i .
- Then, we calculate the gradient of the density estimate $f(\mathbf{x})$ at \mathbf{y}_0 and take an ascent step in that direction.

$$\nabla f(\mathbf{x}) = \sum_i (\mathbf{x}_i - \mathbf{x}) g\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right)$$

where $g(\cdot) = -k'(\cdot)$, the derivative of kernel k

Mean Shift: Example

- The gradient of the density function can be re-written as:

$$\nabla f(\mathbf{x}) = \left[\sum_i G(\mathbf{x} - \mathbf{x}_i) \right] m(\mathbf{x}) \text{ where } m(\mathbf{x}) = \frac{\sum_i \mathbf{x}_i G(\mathbf{x} - \mathbf{x}_i)}{\sum_i G(\mathbf{x} - \mathbf{x}_i)} - \mathbf{x}$$

$$\text{where } G(\mathbf{x} - \mathbf{x}_i) = g\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right)$$

NPTEL

Mean Shift: Example

- The gradient of the density function can be re-written as:

$$\nabla f(\mathbf{x}) = \left[\sum_i G(\mathbf{x} - \mathbf{x}_i) \right] m(\mathbf{x}) \text{ where } m(\mathbf{x}) = \frac{\sum_i \mathbf{x}_i G(\mathbf{x} - \mathbf{x}_i)}{\sum_i G(\mathbf{x} - \mathbf{x}_i)} - \mathbf{x}$$

$$\text{where } G(\mathbf{x} - \mathbf{x}_i) = g\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right)$$

- How? Homework!
- The vector $m(\mathbf{x})$ is known as the **mean shift**, the difference between \mathbf{x} and the gradient weighted mean of the neighbors around \mathbf{x} .

NPTEL

Mean Shift: Example

- The gradient of the density function can be re-written as:

$$\nabla f(\mathbf{x}) = \left[\sum_i G(\mathbf{x} - \mathbf{x}_i) \right] m(\mathbf{x}) \text{ where } m(\mathbf{x}) = \frac{\sum_i \mathbf{x}_i G(\mathbf{x} - \mathbf{x}_i)}{\sum_i G(\mathbf{x} - \mathbf{x}_i)} - \mathbf{x}$$

$$\text{where } G(\mathbf{x} - \mathbf{x}_i) = g\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right)$$

- How? Homework!
- The vector $m(\mathbf{x})$ is known as the **mean shift**, the difference between \mathbf{x} and the gradient weighted mean of the neighbors around \mathbf{x} .
- Current estimate of the mode \mathbf{y}_k at iteration k is replaced by its locally weighted mean:

$$\mathbf{y}_{k+1} = \mathbf{y}_k + m(\mathbf{y}_k) = \frac{\sum_i \mathbf{x}_i G(\mathbf{y}_k - \mathbf{x}_i)}{\sum_i G(\mathbf{y}_k - \mathbf{x}_i)}$$

Mean Shift Segmentation

- Relies on selecting a suitable kernel width h
- Above description strictly color based, however, better results can be obtained by working with feature vectors which include both color and location

Readings

- Chapter 5.3.2, Szeliski, *Computer Vision: Algorithms and Applications*

NPTEL

Normalized Cuts for Segmentation

- **Region-splitting method** where a graph representing pixels in an image is successively split into parts
- Edge weights between pixels in graph measure their similarity



- Graph split into two parts by finding and deleting a **cut-set** with minimum sum of weights i.e., a **min-cut**

Image Credit: K Shafique

Normalized Cuts

- **Min-cut** is defined as the sum of all weights being cut:

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

where A and B are two disjoint subsets of V (set of all vertices)

NPTEL

Normalized Cuts

- **Min-cut** is defined as the sum of all weights being cut:

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

where A and B are two disjoint subsets of V (set of all vertices)

- Using min-cut criterion directly can result in trivial solutions such as isolating a single pixel.
- This paved the way for the formulation of a **Normalized Cut**, defined as:

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

Credit: Szeliski

Normalized Cuts

- We define $\text{assoc}(A, V) = \text{assoc}(A, A) + \text{assoc}(A, B)$ as the sum of all weights associated with vertices in A where:

$$\text{assoc}(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- While computing an optimal normalized cut is NP-complete, there exist approximate solutions (Shi and Malik, 2000).

Readings

- Chapter 5.4, Szeliski, *Computer Vision: Algorithms and Applications*
- Shi and Malik, Normalized Cuts and Image Segmentation, IEEE TPAMI 2000.

Credit: Szeliski

Normalized Cuts

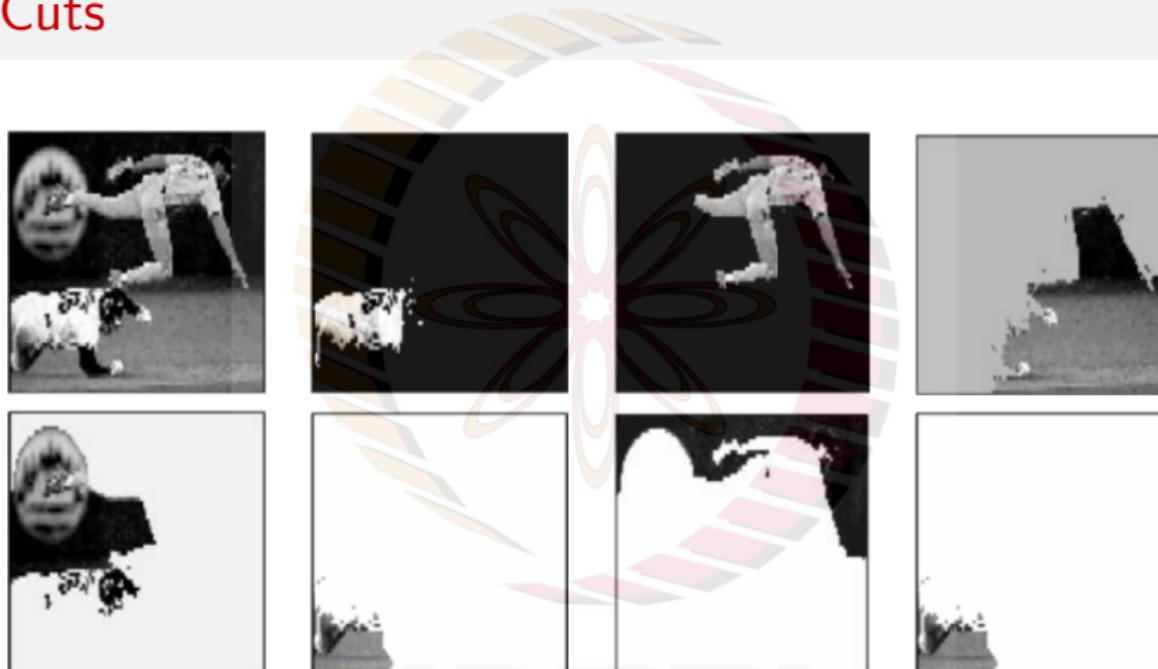


Image components returned by Normalized cuts algorithm

Image Credit: Shi and Malik

Image Segmentation: Other Methods

- k-Means clustering
- Markov Random Fields and Conditional Random Fields
- Many more...

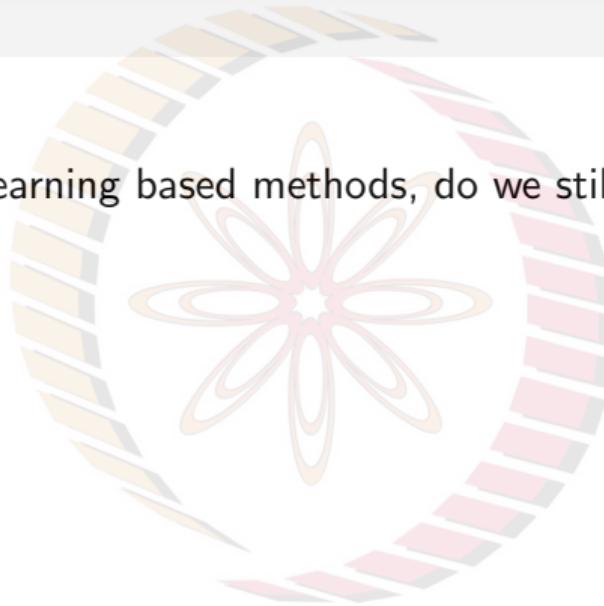
Further Information

- Chapter 5, Szeliski, *Computer Vision: Algorithms and Applications*

NPTEL

Do we need these?

- With the advent of deep learning based methods, do we still need these methods?



NPTEL

Do we need these?

- With the advent of deep learning based methods, do we still need these methods?
- Yes, to an extent. These classical segmentation methods actually inspired early versions of deep learning based methods for object detection and semantic segmentation (we will see this later)
- First R-CNN work (object detection method) used a version of min-cut segmentation method known as CPMC (Constrained Parametric Min Cuts) to generate region proposals for foreground segments



Beyond Images: Segmentation for Video

- **Shot boundary detection** - a key problem in video segmentation: Divide a video into collection of *shots*, each taken from a single sequence of camera



- Another interesting problem is **motion segmentation**, where the aim is to detect and isolate motion in the video. Examples: a person running, a car moving, etc.

Further Information

- Chapter 15.2, 17.1.4, Forsyth, *Computer Vision: A Modern Approach*

Image Credit: M Gygli

Homework

Readings

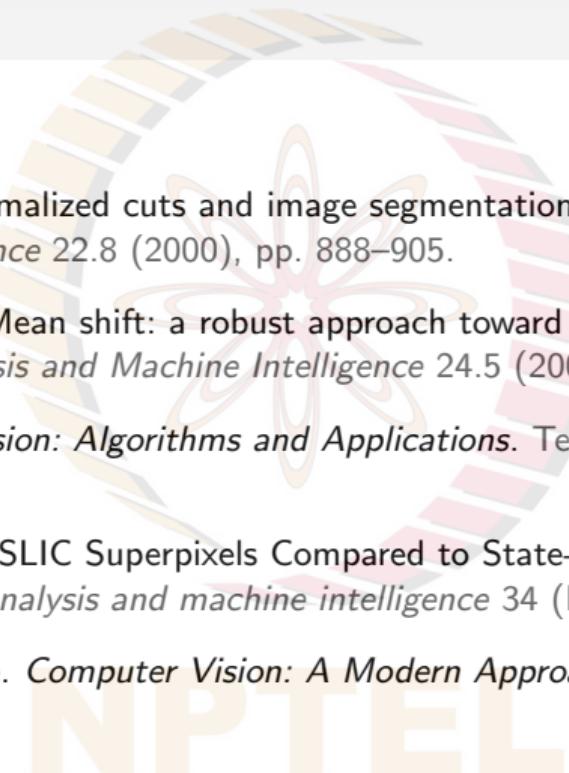
- Chapter 5, Szeliski, *Computer Vision: Algorithms and Applications*
- Chapter 15, 17.1.4, Forsyth, *Computer Vision: A Modern Approach*

Questions

- Derive the final expression for gradient of the kernel density function used in the mean shift method

NPTEL

References

- 
-  Jianbo Shi and J. Malik. "Normalized cuts and image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905.
 -  D. Comaniciu and P. Meer. "Mean shift: a robust approach toward feature space analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 603–619.
 -  Richard Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. London: Springer-Verlag, 2011.
 -  Radhakrishna Achanta et al. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods". In: *IEEE transactions on pattern analysis and machine intelligence* 34 (May 2012).
 -  David Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. 2 edition. Boston: Pearson Education India, 2015.