Vision Systems

Lecture 10

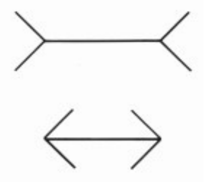
Part 1

Image Segmentation

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.

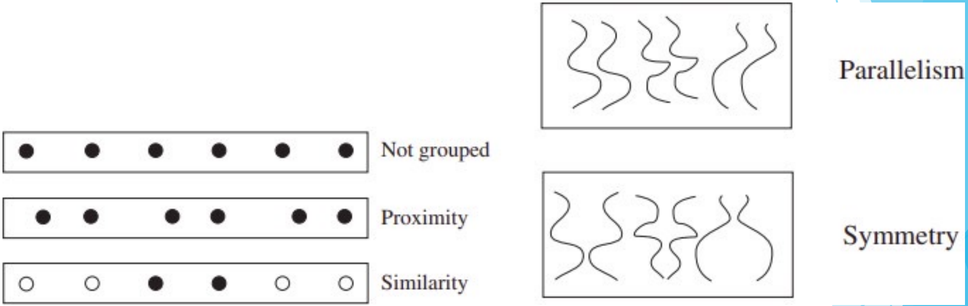


- The famous Muller-Lyer illusion above illustrates the Gestalt belief - humans tend to see things as groups and not individual components.

Source: David Forsyth

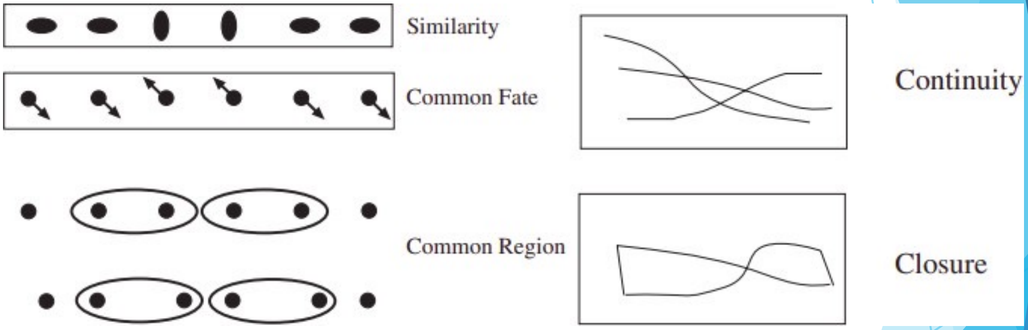
Human Vision: Gestalt and Grouping

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



Credit: David Forsyth

Human Vision: Gestalt and Grouping



Credit: David Forsyth

Human Vision: Gestalt and Grouping

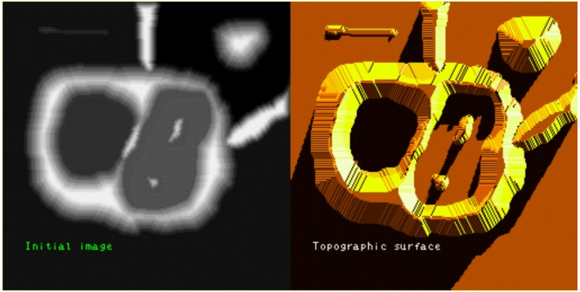
- Gestalt theory is fairly descriptive - loose set of rules to explain why some elements can be grouped together in an image
- However, rules are insufficiently defined to be directly used to form algorithmic tools for grouping objects in images

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

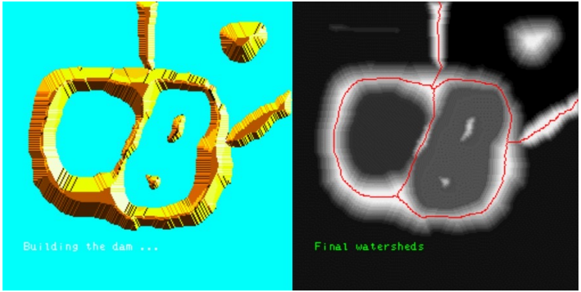


- Image can be segmented into regions where rainwater would flow into the same lake

Credit: S Beucher

Watershed Segmentation Method

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

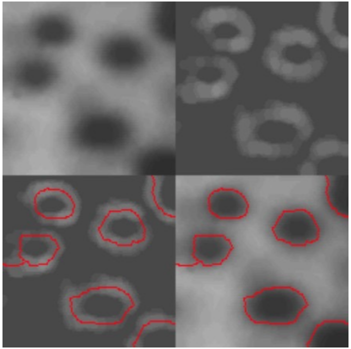


- Results in partitioning the image into **catchment basins** and **watershed lines**

Credit: S Beucher

Watershed Segmentation Method

Generally applied on image gradients instead of applying directly on images

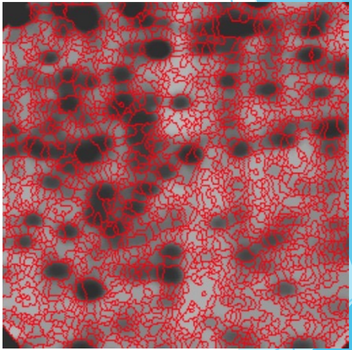


(Top left) Original image; *(Top right)* Gradient image; *(Bottom left)* Watersheds of gradient image; *(Bottom right)* Final segmentation output

Credit: S Beuche

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 successfully 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

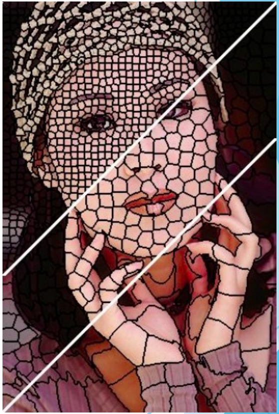
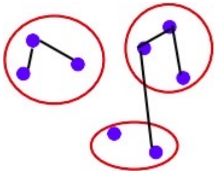


Image Credit: Achanta et al. SLIC Superpixels

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:

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

Graph-based Segmentation

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

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

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

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

Graph-based Segmentation

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

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

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

$$\mathrm{MInt}(C_1, C_2) = \min \ \mathrm{Int}(C_1) + \tau(C_1), \mathrm{Int}(C_2) + \tau(C_2)$$

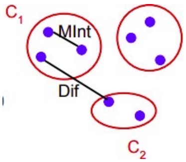
- For any two adjacent regions with at least one edge connecting their vertices, difference between these two regions = minimum weight edge connecting these two regions

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

Graph-based Segmentation

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

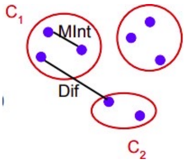
$$D(C_1, C_2) = \begin{cases} \text{true}, & \text{if } \mathrm{Dif}(C_1, C_2) > \mathrm{MInt}(C_1, C_2) \\ \text{false}, & \text{otherwise} \end{cases}$$



Graph-based Segmentation

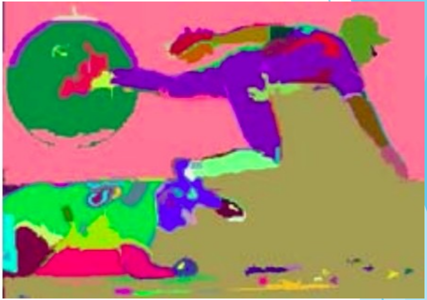
- A predicate $D(C_1, C_2)$ for any two regions C_1 and C_2 is defined as:

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



- For any two regions, if the predicate D evaluates to **false**, regions are merged. Else, regions are considered separate.
- $\Longrightarrow$ 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*

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

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 *pij* - 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:

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

Probabilistic Aggregation

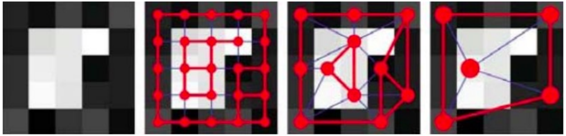
- Alpert *et al.* (2007) proposed a **probabilistic bottom up merging algorithm** for image segmentation based on aggregating two cues - *gray level similarity* and *texture similarity*
- Initially consider each pixel as a region, and assign a merging likelihood *pij* - based on intensity and texture 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:

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

- 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



(a) (b) (c) (d)

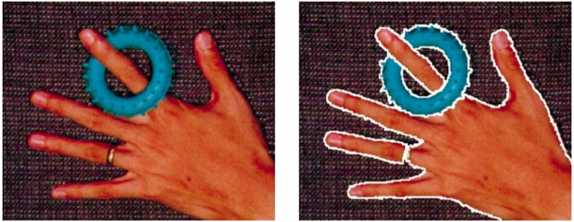
Figure 5.15 Coarse to fine node aggregation in segmentation by weighted aggregation (SWA) (Sharon, Galun, Sharon *et al.* 2006) © 2006 Macmillan Publishers Ltd [Nature]: (a) original gray-level pixel grid; (b) inter-pixel couplings, where thicker lines indicate stronger couplings; (c) after one level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

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

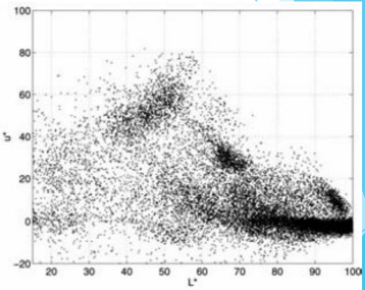


- We estimate p.d.f. using non-parametric estimation and find its *modes*.
- Image is segmented pixel-wise by considering every set of pixels which climb to the same mode as a consistent segment.

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 space[1])



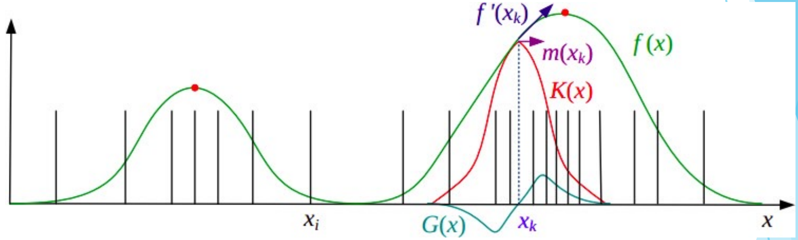
- Our aim is to obtain modes of distribution on the right, without actually explicitly computing the density function! How to do this?

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 $\mathbf{y}_0$ for a local maximum, which can be a random input data point $\mathbf{x}_i$.

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 Segmentation

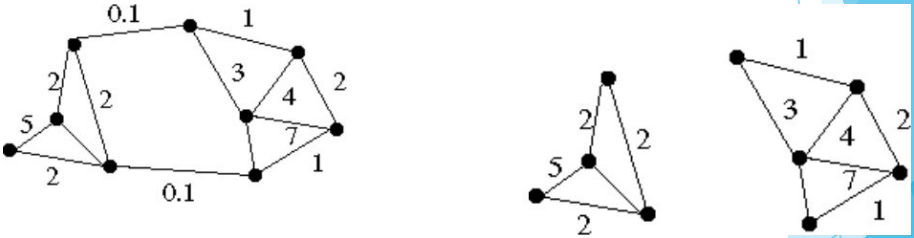
- Relies on selecting a suitable kernel width h
- Above description strictly color based, however, better results can be obtained by working with feature vectors which include both color and location

Readings

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

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)

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

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

- 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

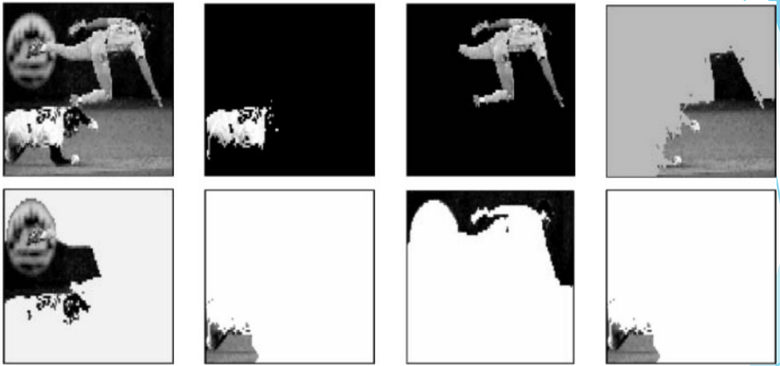


Image components returned by Normalized cuts algorithm

Image Credit: Shi and Malik

Image Segmentation: Other Methods

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

Further Information

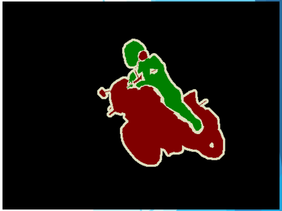
Chapter 5, Szeliski, *Computer Vision: Algorithms and Applications*

Do we need these?

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

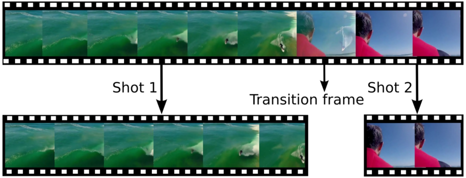
Do we need these?

- 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

Part 1 - 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

Part 2

Human Visual System

Human Visual System: Visual Pathway

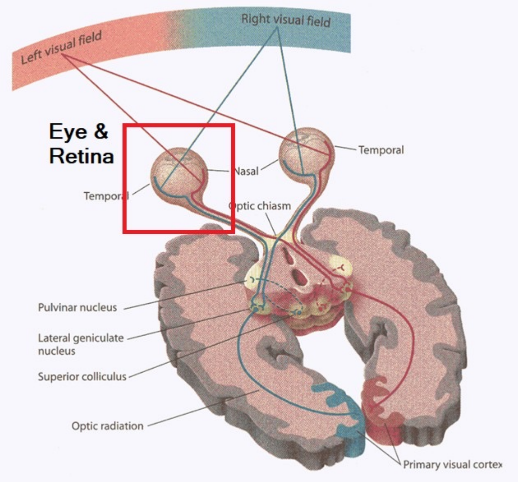


Image Source: Rafael Redondo [6]

Light Visible to Human Eye

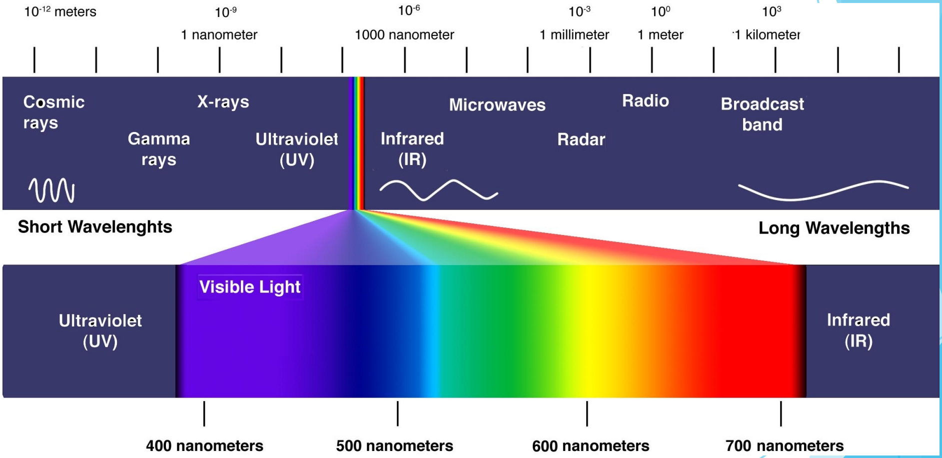
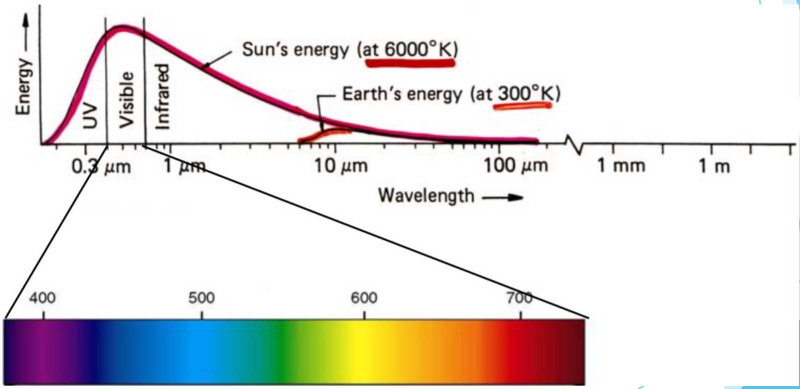


Image Source: www.astronomersgroup.org

Light Visible to Human Eye

Our vision appears to be optimized for receiving the most abundant spectral radiance our star emits



The Retina

The Retina = Photoreceptors + Image Filtering

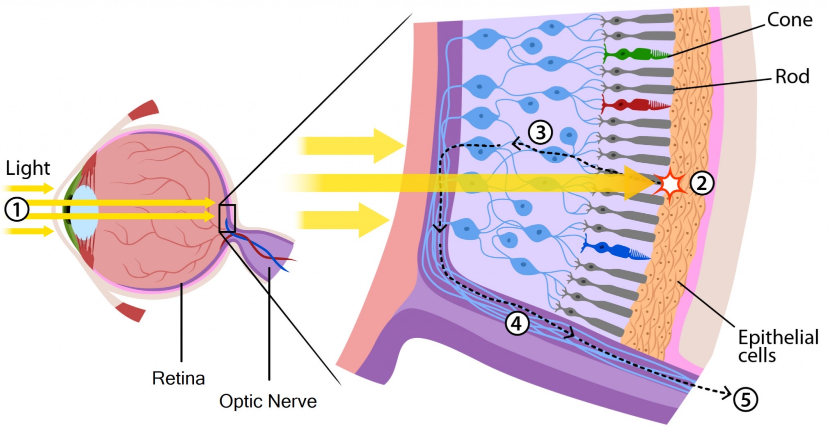


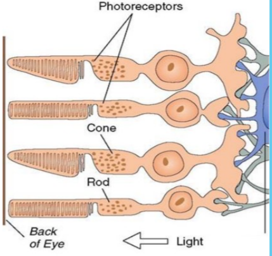
Image Source: mymacularjournal.com

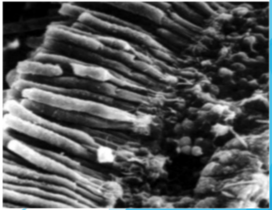
Photoreceptors in the Retina

Two Types:

- **Rods:** Sensitive to intensity, but not color; form blurred images

- **Cones:** Color sensitive, form sharp images, require many photons. Three types, each maximally sensitive to one of three different wavelengths.





Coding of Light by Rods and Cones

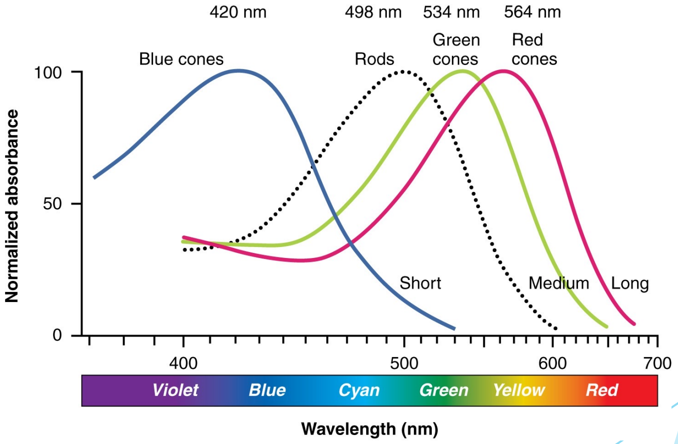
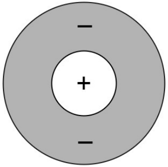


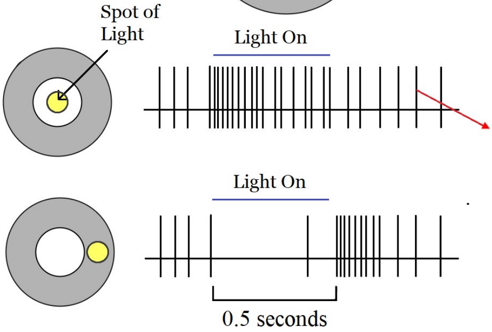
Image Source: Michael C (StackExchange)

Image Filtering in Space and Time in the Retina

On center,
Off surround
cell

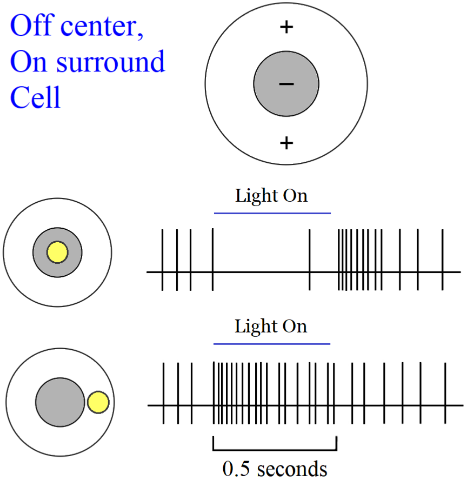


This space-time filter is also called the cell's "receptive field"

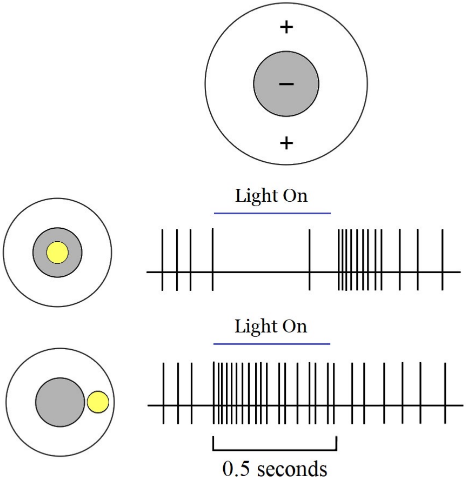


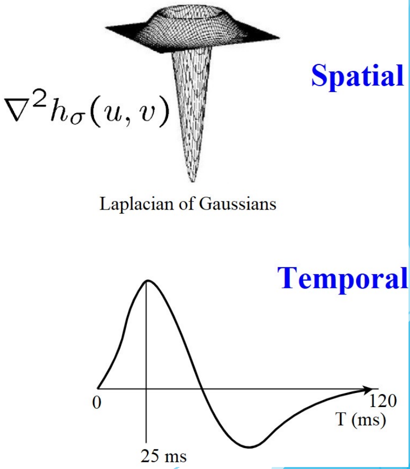
Neuron responds with electrical pulses known as Spikes or Action Potentials

Image Filtering in Space and Time in the Retina

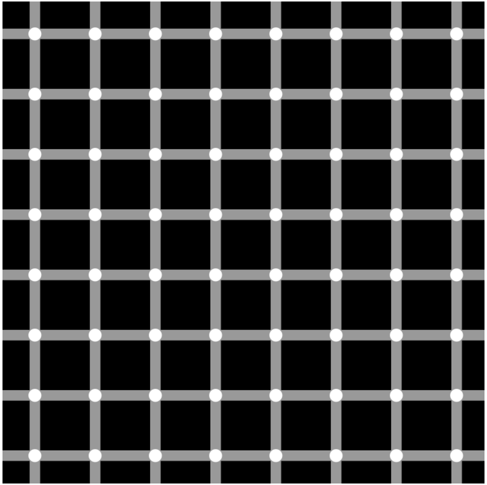


Retina takes Spatial and Temporal Derivatives





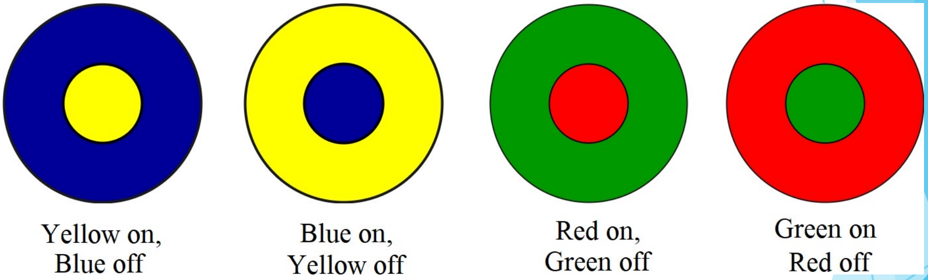
Your Retinal Filters at Work



Black dots or white dots?

Retina also takes Derivatives in Color Space

"Color-opponent" processing



Visual consequence: **Negative afterimage** - An image is seen after a portion of the retina is exposed to an intense visual stimulus (colors complementary to those of stimulus)

The Visual Pathway: LGN

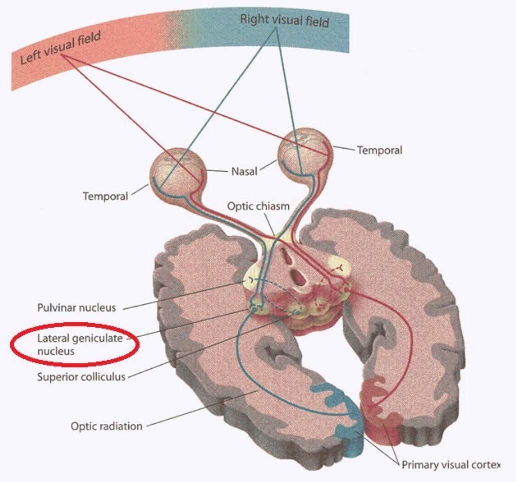


Image Source: Rafael Redondo [6]

- LGN receptive fields similar to retinal (center-surround, on-off)
- Thought to be a relay but receives massive feedback from cortex

The Visual Pathway: V1

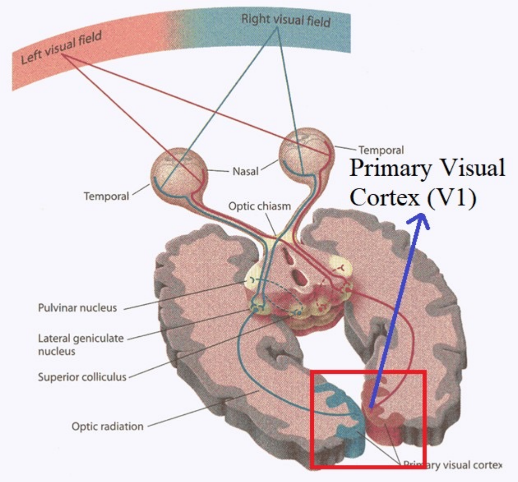
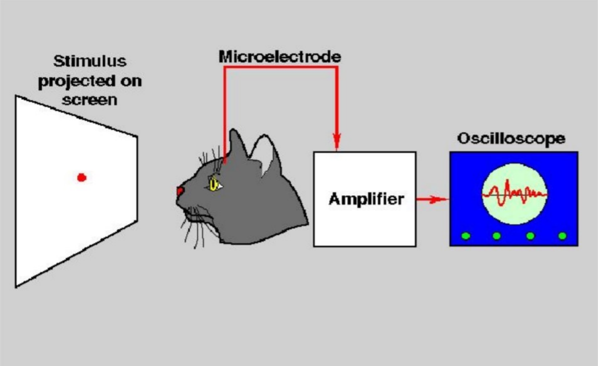


Image Source: Rafael Redondo [6]

A Tale of Two Receptive Fields

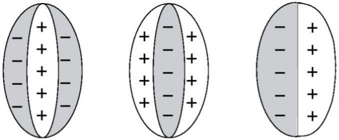
Recall: David Hubel and Torsten Wiesel were the first to characterize V1 receptive fields by recording from a cat viewing stimuli on a screen



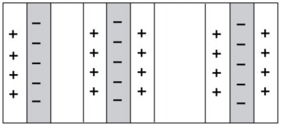
In 1981, they received a Nobel prize in physiology and medicine for their work

Simple and Complex Cell Receptive Fields

Receptive fields



“Bar” detectors “Edge” detector



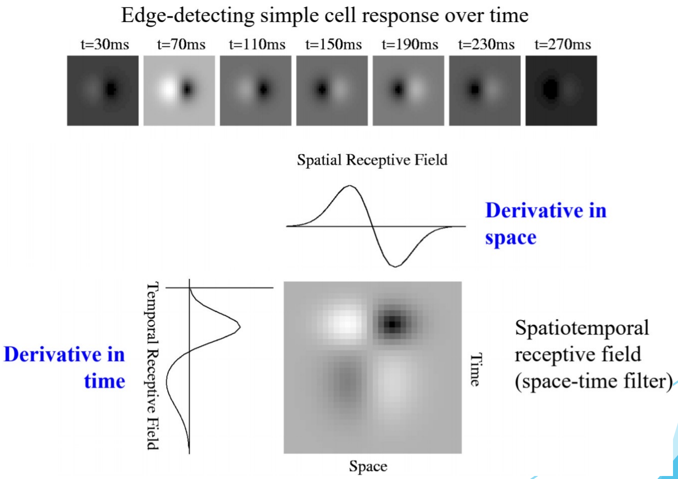
Position-invariant “bar” detector

- **Simple Cells:**
  Detect oriented bars and edges at a specific location

- **Complex Cells:**
  Sensitive to orientation but invariant to position

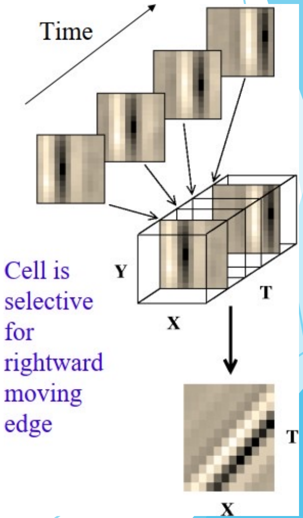
Cortical Cells Compute Derivatives

Spatial derivative is orientation-sensitive



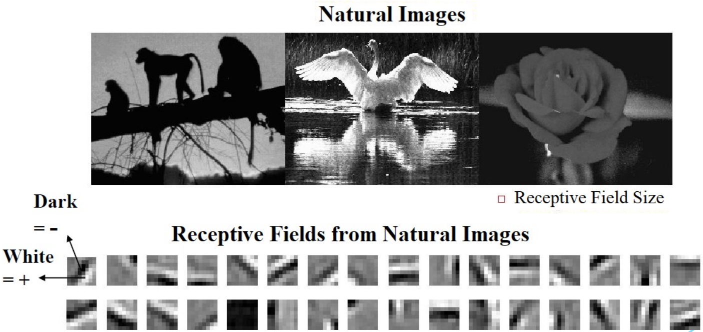
Direction Selectivity of Some Cortical Cells

Oriented derivative in X-T space!



Oriented Filters and Natural Images

- **Goal:** Learn independent filters whose linear combination best represents natural images
- Optimal set of such filters are oriented and localized to specific regions of image



See Olshausen and Field 1996, Rao and Ballard 1999 for more details

Dorsal and Ventral Pathways in the Visual Cortex

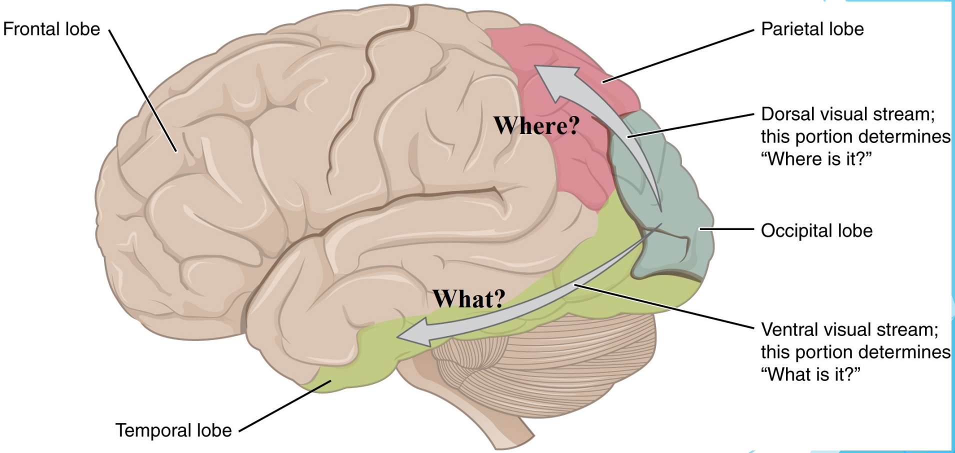


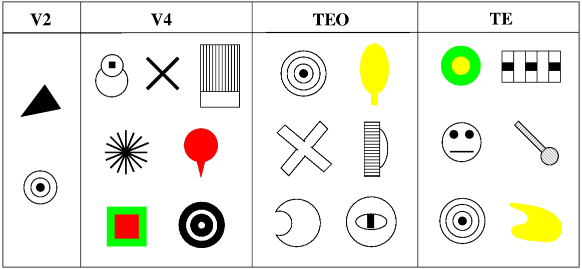
Image Source: [Rice University OpenStax]

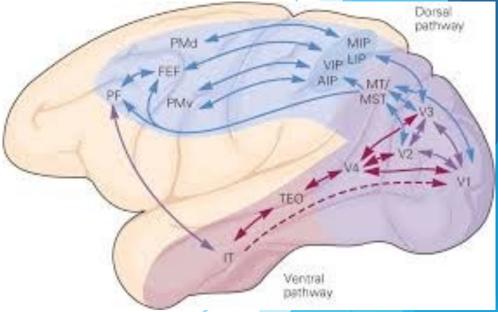
Visual Cortex is Hierarchically Organized: ”What” Pathway

Object Pathway: V1 → V2 → V4 → TEO → TE

Cells respond to more and more complex stimuli as we go higher up

Example Receptive Fields

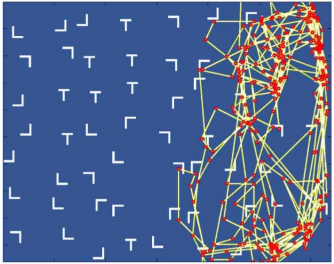




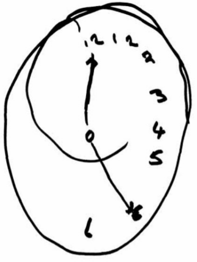
"Where" Pathway

V1 → V2 → MT → MST → Posterior Parietal Cortex

- Cells respond to more and more complex forms of motion and spatial relationships
- Damage to right parietal cortex may result in spatial hemi-neglect - patient behaves as if the left part of the visual world doesn't exist



Eye movements only to right part of the screen



Only right side of clock drawn

Image Source: [Scholarpedia - Hemineglect]

The Visual Processing Hierarchy

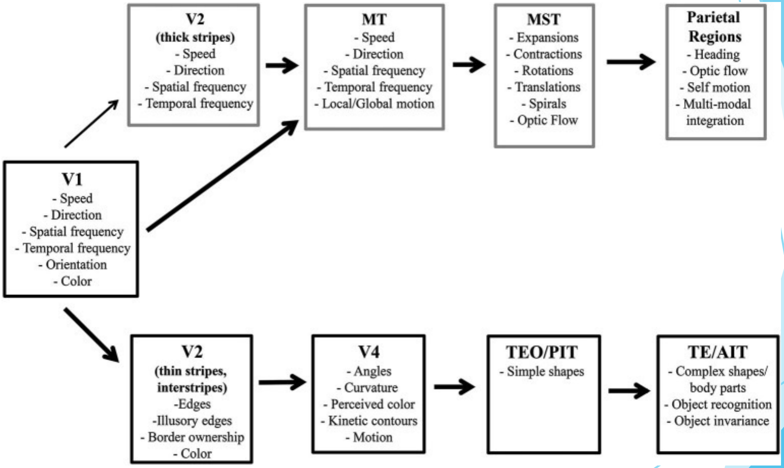


Image Source: Perry, Fallah 2014

Readings

Summary of Human Visual System

Lecture Notes of Majumder, UCI on Visual Perception

If you'd like to know more…

Chapter on Vision by Martin A. Fischler and Oscar Firschein in Intelligence: The Eye, the Brain, and the Compute

Nobel laureate David Hubel's book: Eye, Brain, and Vision

The Joy of Visual Perception by Peter K. Kaiser (Web Book)

Lecture 8 of UWash's CS455: Computer Vision (Rao, 2009)

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.

References

- Torsten N. Weisel David H. Hubel. “Effects of Monocular Deprivation in Kittens”. In: *Naunyn Schmiedebergs Arch Exp Pathol Pharmakol* 248 (1964), pp. 492–497.
- David C. Van Essen and Jack L. Gallant. “Neural mechanisms of form and motion processing in the primate visual system”. In: *Neuron* 13 (1994), pp. 1–10.
- Bruno A. Olshausen and David J. Field. “Natural image statistics and efficient coding.”. In: *Network* 7 2 (1996), pp. 333–9.
- Rajesh Rao and Dana Ballard. “Predictive Coding in the Visual Cortex: a Functional Interpretation of Some Extra-classical Receptive-field Effects”. In: *Nature neuroscience* 2 (Feb. 1999), pp. 79–87.
- Carolyn Jeane Perry and Mazyar Fallah. “Feature integration and object representations along the dorsal stream visual hierarchy”. In: *Frontiers in computational neuroscience* 8 (2014), p. 84.
- Rafael Redondo. “New contributions on image fusion and compression based on space-frecuency representations”. In: (July 2020).