Lecture 6: CS677

Sept 7, 2017

1

Review

- HW1 due September 12
- Cloud computing: students can get a better personal account at: https://console.cloud.google.com/freetrial?_ga=2.228461851.-722665125.1503520492&page=1
- Previous class
  - Shape from shading and photometric stereo
  - Color perception
  - OpenCV tutorial
- Today's objective
  - Image filtering
  - Image segmentation

Comments on HW1

- Final formulas for vanishing points and lines are given in a slide in Lec4 notes
- These formulas were not derived, just stated
- For HW1, students need to derive these formulas from the basic projection equations
- Answers need not be expressed in the given form, nor derived using concepts of plane or line at infinity; other ways of approaching the solution are equally acceptable.

Next Topics

- We are finished with image formation
- Next major topics of interest
  - Inference of 3-D
    - Single Images (difficult)
    - From multiple views
    - From direct range sensing
  - Detection and recognition of objects
- Image feature extraction and segmentation
  - Useful for both of the above topics
  - This is what we will study next

4

Image Smoothing/Filtering

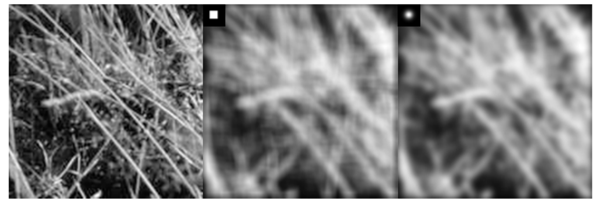
- Reduce effects of noise, texture
- Enhance sharpness of edges
- Linear Filters
- General process
  - Form a new image whose pixels are a weighted sum of original pixel values, using the same set of weights at each point
  - Smoothing by averaging
    - Form the average of pixels in a neighborhood

$$\mathcal{R}_{ij} = \frac{1}{(2k+1)^2} \sum_{u=i-k}^{u=i+k} \sum_{v=j-k}^{v=j+k} \mathcal{F}_{uv}.$$

    - Weights need not be uniform: high value at center, drops off as a function of distance (e.g. as a Gaussian function)
    - An example on next slide

5

Example: Smoothing



Original Image | Average smoothing | Gaussian Smoothing

6

Convolution

- Represent weights as an image, H (sometimes called the **kernel)**
- *Convolving* a filter H with image F gives a new image R given by:

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv}$$

- Note: reversal of sign of dummy variables, *u* and *v*; sum is over all non-zero values
- Convolution commonly written as R = H * F
- Convolution formulation provides many useful properties studied in basic signal processing, we list only a few here
- Convolution is linear and shift invariant
  - Scaling of H or F results in linear scaling of R
  - Shift invariance: translation of signal results in same translation of result
  - Commutative: g*h = h*g; associative (f*(g*h)) = (f*g)*h)
- Edge effects
  - Values undefined at image boundary: pad the image by zeros beyond the border

7

Derivative (Difference) by Convolution

- Consider the convolution filter given below:

$$\mathcal{H} = \left\{ \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{matrix} \right\}$$

- Computes difference in the x-direction
- Following compute derivatives in x and y directions with some influence from neighboring rows and columns (Sobel masks)

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

8

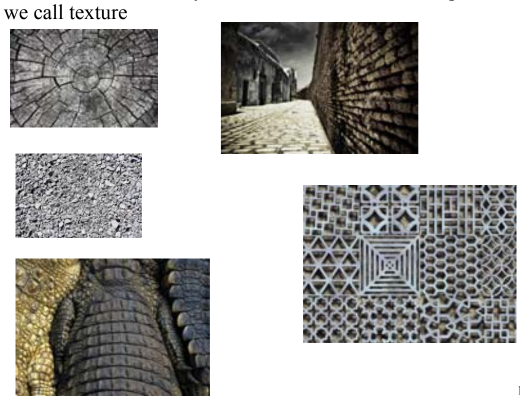
Image Pyramids

- Interesting patterns may occur at different scales
  - e.g. on a zebra, are we interested in hair, stripes or just the neck?
- Applying large filters is expensive
  - We can get same effect by reducing resolution of the image
- Figure shows multiple levels of reduction, resulting in a *pyramid*
  - We can sample alternate pixels (for example) or convolve with a Gaussian of selected σ.

9

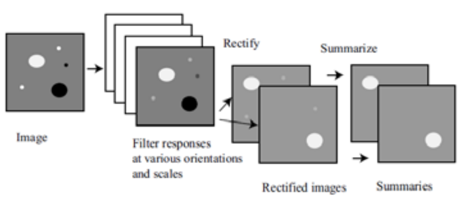
Texture

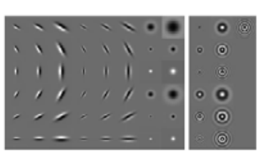
- Different materials may have different characteristic patterns, that we call texture



10

Texture Representation





Filter bank: oriented and isotropic

11

Texture Comments

- Texture seems characterized by repeated patterns
  - However, natural textures are not completely uniform
- What is the size of image patch over which we should compute texture patterns?
- For natural textures, one approach is to compute outputs of various filters
  - Outputs of filters that emphasize different patterns, at different scales and different orientations
- Texture variation (gradient) can be a strong indicator of surface orientation
  - But requires texture to be homogeneous

12

Image Segmentation

- Find boundaries of objects and surfaces in the scene
  - Typically characterized by discontinuities in range/depth of points in the scene (assumes object surfaces are continuous)
  - However, depth information is not readily available; instead we detect intensity (or color) discontinuities which may not always correspond to object boundaries
- Some examples on following slides

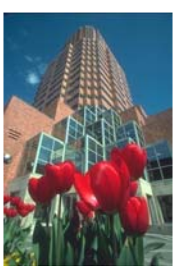
13

Image Segmentation

Example from Berkeley Segmentation Dataset

14

Image Segmentation



Example from Berkeley Segmentation Dataset

15

Image Segmentation: Approaches

- Find points of rapid changes in intensity/color (edge detection), connect them to give boundaries
- Find *regions* of constant properties (intensity, color, texture..)
- We will study region segmentation first (chapter 9); book does it in the reverse order.

16

Region Segmentation (FP 9.3,9.4)

- Segmentation is equivalent to *clustering*
  - ***Cluster***: collection of data samples (pixels)
- Divisive
  - Start with entire data set as a single cluster
  - Split cluster into components that yield largest inter-cluster distance
  - Repeat until distance or clusters become small
- Agglomerative
  - Start with each point being a separate cluster
  - Merge clusters with smallest inter-cluster distance
  - Repeat until distance or clusters become large

17

K-Means Clustering

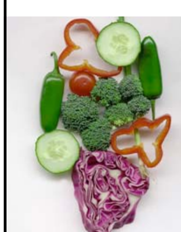
- Choose a fixed number of clusters
- Choose cluster centers and point-cluster allocations to minimize error:

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

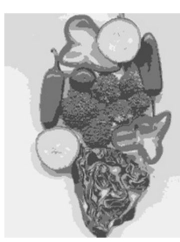
- Too many possible allocations for exhaustive search
- Algorithm
  - Fix cluster centers; allocate points to closest cluster
  - Fix allocation; compute the cluster centers
  - Initial assignment may be random
- *x* can be any *vector* in a space for which we can compute a distance

18

Image



Clusters on intensity

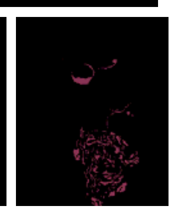


Clusters on color

K-means clustering using intensity only and using color (five clusters)

19

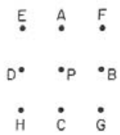
K-means using color, 11 segments.
Left: image with mean values of clusters
Others: Four segments shown, note a segment is not necessarily connected

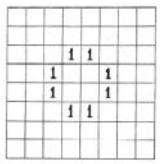


20

Connected Components

- Compute *connected components*
  - 4 or 8 connectivity
  - Connectivity algorithms not covered (but are simple)



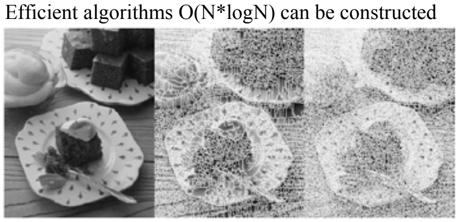


Superpixels

- Goal is to produce segments of similar size with high fidelity to boundaries
- Segments not likely to correspond to objects; objects to be detected in a subsequent stage
- May be useful for various forms of post-processing
- Has become a popular first step in recent years
- Watershed algorithm
- SLIC algorithm (not in book)

Watershed Algorithm

- Imagine intensity of image to represent terrain height
- Find “catchment basins” (lakes) that would form from rain fall
- Find local minima: consider them to be seeds and give a unique label to each
- For any pixel, go in direction of negative gradient, if a seed is reached (or a labeled pixel is reached), take its label.
- Efficient algorithms O(N*logN) can be constructed



On image intensity   On gradient magnitude

Next Class

- Chapter 9, section 9.4
- Chapter 5, sections 5.1, 5.2