Part Three

- Single pixels do not carry that much information
- The spatial structure of the image carries most of the interpretation
- The next part of the course looks at local indicators of spatial structure
  - smoothing to deal with noise
  - understanding image scale
  - texture
  - scale and rotation invariant informative keypoints

Linear Filters (§7)

- Example One
  - Suppose we want to try to smooth an image by replacing each pixel by the average of a 5x5 block centered on the pixel?
- Example Two
  - Suppose we want to smooth as we did in example one, but pixels that are further away count for less?
- Example Three
  - Suppose we want to create an image that illustrates horizontal steps and drops as we go across the image?

Linear Filters (§7)

- Illustrated in 1D

```
[0 0 1 2 2 0 0 0 0]
```

Multiply lined up pairs of numbers and then sum up

Linear Filters (§7)

- Example: smoothing by averaging
  - form the average of pixels in a neighborhood (weights are equal)
- Example: smoothing with a Gaussian
  - form a weighted average of pixels in a neighborhood (weights follow a Gaussian function)
- Example: finding a derivative
  - negative weights on one side, positive ones on the other
Linear Filter Example

- Compute a new image which is an average of 3 by 3 blocks
- $H$ (the kernel)
  
  $$
  H(i', j') = \begin{cases} 
  \frac{1}{9} & i', j' \in [-1, 1] \\
  0 & \text{otherwise}
  \end{cases}
  $$
- Result is
  
  $$
  R_{ij} = \sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} \frac{1}{9} F(i', j')
  $$

Example: Smoothing by Averaging

Smoothing with a Gaussian

- Smoothing with an average actually doesn’t really make sense because points close to the center should count more.
- Also, it does not compare at all well with a defocused lens
  - Most obvious difference is that a single point of light viewed in a defocused lens looks like a fuzzy blob; but the averaging process would give a little square.
- A Gaussian gives a good model of a fuzzy blob

An Isotropic Gaussian

- The picture shows a smoothing kernel proportional to
  
  $$
  \exp \left( - \frac{x^2 + y^2}{2\sigma^2} \right)
  $$
  
  (a reasonable model of a circularly symmetric fuzzy blob)
Smoothing with a Gaussian

Block Averaging

Image Scale

- The difference between a tree in the distance, and its leaves up close, is one of image scale
- An arbitrary image will have multiple arbitrary scales
- Typically we analyze images at various scales
- A good way to think of rescaling an image is to smooth with a Gaussian and sub sample the results.
Filters used as templates
Filter for a dark spot
Positive responses
(negatives values are set to zero)
To visualize negative values, make mean value gray.
Increase contrast by scaling.
(push actual max/min towards display max/min).

Filters used as templates
Filter for a dark bar at 135°
Positive responses. Negatives are clipped at zero.
To visualize negative values, make mean value gray.
Increase contrast by scaling.
(push actual max/min towards display max/min).

Linear Filters (§7)
- Properties
  - Output is a linear function of the input
  - Can represent as a matrix (but usually do not)
- Terminology
  - Array of weights is referred to as the kernel (H)
  - (Sometimes referred to as “mask” or “template”)
- Be aware of two forms
  - (Cross) Correlation (more natural, often what we visualize)
  - Convolution (more fundamental, has some useful mathematical properties)
  - Convolution is correlation by a flipped kernel (if kernel is symmetric, then no difference)
    - (and vice versa)

Convolution vs. Correlation
- Perhaps easiest to think about in just one dimension (time)
- Correlation
  - New signal by moving desired mask/template around.
  - New values follow the mask
- Convolution
  - Filter is the response to a unit bar (or delta function)
  - Convolution gives response to the entire signal
  - Your signal comes to the filter
- Can switch from one to the other by flipping the filter
  - $h_{\text{corr}}(x) = h_{\text{conv}}(-x)$
  - $h_{\text{corr}}(x, y) = h_{\text{conv}}(-x, -y)$
- No difference if filter is symmetric
Image Filtering Preliminaries

• Denote the image by $F$ (to follow the book).

• Represent weights as a second image, $H$ (the kernel).

• Pretend that images are padded to infinity with zeros (so sums don’t need limits).

• To shift a function $f(x,y)$ up and to the right by $(a,b)$
  - $f(x-a, y-b)$

Correlation

• Denote by $\odot$

• Then the definition of discrete 2D correlation is:

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

Puts filter on $(i,j)$