Lecture 2
Linear filters
• Proposition 1. The primary task of early vision is to deliver a small set of useful measurements about each observable location in the plenoptic function.

• Proposition 2. The elemental operations of early vision involve the measurement of local change along various directions within the plenoptic function.

• Goal: to transform the image into other representations (rather than pixel values) that makes scene information more explicit.
RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT’S STRIATE CORTEX

BY D. H. HUBEL* AND T. N. WIESEL*

From the Wilmer Institute, The Johns Hopkins Hospital and University, Baltimore, Maryland, U.S.A.

Receptive field of a cell in the cat’s cortex

Responses to an oriented bar
Outline

• Linear filtering
• Fourier Transform
• Human spatial frequency sensitivity
• Phase
• Sampling and Aliasing
• Spatially localized analysis
Filtering

We want to remove unwanted sources of variation, and keep the information relevant for whatever task we need to solve.
Linear filtering

For a linear system, each output is a linear combination of all the input values:

\[ f[m,n] = \sum_{k,l} h[m,n,k,l]g[k,l] \]

In matrix form:

\[ f = H g \]
Linear filtering

In vision, many times, we are interested in operations that are spatially invariant. For a linear spatially invariant system:

\[ f[m,n] = h \otimes g = \sum_{k,l} h[m - k, n - l]g[k,l] \]
Linear filtering

\[ f[m,n] = h \otimes g = \sum_{k,l} h[m - k, n - l] g[k, l] \]

Linear system:

Input:

\[ g[m] \]

Output?

\[ f[m = 0] = \sum_{k} h[-k] g[k] \]

\[ f[m = 1] = \sum_{k} h[1 - k] g[k] \]

\[ f[m = 2] = \sum_{k} h[2 - k] g[k] \]

\[ f[m=0]=\sum_{k} h[-k] g[k] = -2 \]

\[ f[m=1]=\sum_{k} h[1-k] g[k] = -4 \]

\[ f[m=2]=\sum_{k} h[2-k] g[k] = 0 \]
Linear filtering

For a linear spatially invariant system

$$f[m, n] = I \otimes g = \sum_{k,l} h[m - k, n - l]g[k, l]$$

m=0 1 2 ...

<table>
<thead>
<tr>
<th>111</th>
<th>115</th>
<th>113</th>
<th>112</th>
<th>111</th>
<th>112</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>135</td>
<td>138</td>
<td>137</td>
<td>139</td>
<td>145</td>
<td>146</td>
<td>149</td>
</tr>
<tr>
<td>191</td>
<td>201</td>
<td>217</td>
<td>220</td>
<td>103</td>
<td>59</td>
<td>68</td>
</tr>
<tr>
<td>195</td>
<td>205</td>
<td>216</td>
<td>222</td>
<td>113</td>
<td>68</td>
<td>69</td>
</tr>
<tr>
<td>199</td>
<td>203</td>
<td>223</td>
<td>228</td>
<td>108</td>
<td>68</td>
<td>71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-1</th>
<th>2</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
</tbody>
</table>

$$g[m,n] \otimes h[m,n] = f[m,n]$$
Borders

From Szeliski, Computer Vision, 2010
Impulse

\[ f[m,n] = I \odot g = \sum_{k,l} h[m-k, n-l] g[k, l] \]
Shifts

\[ f[m,n] = I \otimes g = \sum_{k,l} h[m-k, n-l]g[k,l] \]
It is linear, but not a spatially invariant operation. There is not convolution.
Rectangular filter

\[ g[m,n] \otimes h[m,n] = f[m,n] \]
Rectangular filter

\[ g[m,n] \odot h[m,n] = f[m,n] \]
Rectangular filter

\[ g[m,n] \times h[m,n] = f[m,n] \]
Sharpening

original

Sharpened original

2.0

0.33

==
Sharpening example

Original

Sharpened (differences are accentuated; constant areas are left untouched).
Sharpening

before

after
A taxonomy of useful filters

• Impulse, Shifts,
• Blur
  – Rectangular blur (see artifacts)
  – Gaussian
  – Bilateral exponential
  – Asymmetrical filter: motion blur
• Edges
  – [-1 1]
  – Derivative filter
  – Derivative of a gaussian
  – Oriented filters
  – Gabor filter
  – Quadrature filters: phase and magnitude.
  – Elongated edges: filling gaps...
Linear blur occurs under many natural situations

(from Fergus et al, 2007)

This is not a Gaussian kernel...
Linear blur occurs under many natural situations
Linear blur occurs under many natural situations
Linear blur occurs under many natural situations.
Gaussian filter

\[ G(x, y; \sigma) = \frac{1}{2\pi \sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]
Gaussian filter

Dali
Some desirable properties for a blur kernel

- Positivity: \( h(m) \geq 0 \)
- Symmetry: \( h(m) = h(-m) \)
- Unimodality: \( h(m) \geq h(m+1) \) for \( m \geq 0 \)
- Normalized: \( \sum h(m) = 1 \)
- Equal contribution: \( \sum h(2m) = \sum h(2m+1) \)

Some kernels that verify this are:

\[
\begin{bmatrix}
\frac{1}{2} & \frac{1}{2} \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4}
\end{bmatrix}
\]
\[
\frac{\partial I}{\partial x} \approx I(x, y) - I(x - 1, y)
\]

\[
g[m,n] \times [-1, 1] = h[m,n] = f[m,n]
\]
\([-1 \ 1]^T\)

\[
\begin{bmatrix}
  g[m,n] \\
  \hline
  \times
  \hline
  [-1, 1]^T
  \hline
  =
  \\
  h[m,n]
\end{bmatrix}
\]

\[
f[m,n]
\]
Differential Geometry Descriptors

$I(x,y)$
Finding edges in the image

Image gradient:

$$\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

Approximation image derivative:

$$\frac{\partial I}{\partial x} \approx I(x, y) - I(x - 1, y)$$

Edge strength

$$E(x, y) = |\nabla I(x, y)|$$

Edge orientation:

$$\theta(x, y) = \angle \nabla I = \arctan \frac{\partial I/\partial y}{\partial I/\partial x}$$

Edge normal:

$$n = \frac{\nabla I}{|\nabla I|}$$
Differential Geometry Descriptors

If we think of the image as a continuous function

\[ I(x,y) \]

\[ \nabla I = \left( \frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y} \right) \]

Image gradient:

Directional gradient:

\[ |u| = 1 \]

\[ u^T \nabla I = \cos(\alpha) \frac{\partial I(x,y)}{\partial x} + \sin(\alpha) \frac{\partial I(x,y)}{\partial y} \]

Laplacian:

\[ \nabla^2 I = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \]

Problem: \( dI/dx \) might not be defined around discontinuities.
Gaussian derivative

\[ g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

\[ \frac{\partial g(x, y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]
\[ g(x, y) = \frac{1}{2\pi \sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

\[
\frac{\partial I(x, y)}{\partial x} \otimes g(x, y) = \frac{\partial I(x, y) \otimes g(x, y)}{\partial x} = I(x, y) \otimes \frac{\partial g(x, y)}{\partial x}
\]

\[
\frac{\partial g(x, y)}{\partial x} = \frac{-x}{2\pi \sigma^4} e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]
\[ g_x(x, y) = \frac{\partial g(x, y)}{\partial x} = \frac{-x}{2\pi \sigma^4} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

\[ g_y(x, y) = \frac{\partial g(x, y)}{\partial x} = \frac{-x}{2\pi \sigma^4} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

The smoothed directional gradient is a linear combination of two kernels

\[ u^T \nabla g \otimes I = \left( \cos(\alpha) g_x(x, y) + \sin(\alpha) g_y(x, y) \right) \otimes I(x, y) = \]

Any orientation can be computed as a linear combination of two filtered images

\[ = \cos(\alpha) g_x(x, y) \otimes I(x, y) + \sin(\alpha) g_y(x, y) \otimes I(x, y) = \]

\[ = \cos(\alpha) + \sin(\alpha) = \]

Steereability of gaussian derivatives, Freeman & Adelson 92
Laplacian

\[
g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]

\[
\nabla^2 I \otimes g = \left( \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \right) \otimes g(x, y)
\]

\[
\nabla^2 I \otimes g = I \otimes \nabla^2 g
\]

\[
\nabla^2 g(x, y) = \left( \frac{x^2 + y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) g(x, y)
\]
Laplacian
Outline

• Linear filtering
• Fourier Transform
• Human spatial frequency sensitivity
• Phase
• Sampling and Aliasing
• Spatially localized analysis
Linear image transformations

• In analyzing images, it’s often useful to make a change of basis.

\[ \vec{F} = U \vec{f} \]

Transformed image

Fourier transform, or Wavelet transform, or Steerable pyramid transform

Vectorized image

\[ = U \]
Self-inverting transforms

\[ \vec{F} = U \vec{f} \quad \text{↔} \quad \vec{f} = U^{-1} \vec{F} \]

Same basis functions are used for the inverse transform

\[ \vec{f} = U^{-1} \vec{F} \]

\[ = U^+ \vec{F} \]

U transpose and complex conjugate
An example of such a transform: the Discrete Fourier transform

Forward transform

\[ F[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f[k,l] e^{-\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)} \]

Inverse transform

\[ f[k,l] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F[m,n] e^{\pm \pi i \left( \frac{km}{M} + \frac{ln}{N} \right)} \]
Fourier transform visualization

$$F[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f[k,l]e^{-\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)}$$
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of $x, y$ for some fixed $u, v$. We get a function that is constant when $(ux + vy)$ is constant. The magnitude of the vector $(u, v)$ gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.
Here $u$ and $v$ are larger than in the previous slide.
And larger still...

\[ e^{-\pi(ux + vy)} \]

\[ e^{\pi(ux + vy)} \]
Why is the Fourier domain particularly useful?

- Linear, space invariant operations are just diagonal operations in the frequency domain.

- Ie, linear convolution is multiplication in the frequency domain.
Fourier transform of convolution

Consider a (circular) convolution of $g$ and $h$

$$f = g \odot h$$

In the transform domain, this just modulates the transform amplitudes

$$F[m, n] = DFT(g \odot h)$$

$$= G[m, n]H[m, n]$$
Fourier transform of convolution

\( f = g \otimes h \)

\[
F[m, n] = DFT(g \otimes h)
\]

\[
F[m, n] = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sum_{k,l} g[u-k, v-l]h[k,l]e^{-\pi i \left( \frac{um}{M} + \frac{vn}{N} \right)}
\]

\[
= \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sum_{k,l} g[u-k, v-l]e^{-\pi i \left( \frac{um}{M} + \frac{vn}{N} \right)}h[k,l]
\]

\[
= \sum_{\mu=-k}^{M-k-1} \sum_{\nu=-l}^{N-l-1} \sum_{k,l} g[\mu, \nu]e^{-\pi i \left( \frac{(\mu+k)m}{M} + \frac{(\nu+l)n}{N} \right)}h[k,l]
\]

\[
= \sum_{k,l} G[m, n]e^{-\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)}h[k,l]
\]

\[
= G[m, n]H[m, n]
\]
Analysis of a simple sharpening filter

\[ F[m] = \sum_{k=0}^{M-1} f[k] e^{-\pi i \left( \frac{km}{M} \right)} \]

\[ = 2 - \frac{1}{3} \left( 1 + 2 \cos \left( \frac{\pi m}{M} \right) \right) \]
#1: Range [0, 1]
Dims [256, 256]

#2: Range [8.5e-006, 1.7]
Dims [256, 256]
#1: Range [0, 1]
Dims [256, 256]

#2: Range [1.39e-005, 5.88]
Dims [256, 256]
2094

#1: Range [0, 1]
Dims [256, 256]

#2: Range [8.7e-005, 19]
Dims [256, 256]
4052.
8056.
28743

#1: Range [0, 1]
Dims [256, 256]

#2: Range [0.00109, 146]
Dims [256, 256]
49190.
Now, an analogous sequence of images, but selecting Fourier components in descending order of magnitude.
Figure 6

#1: Range [0, 1]
Dims [256, 256]

#2: Range [0.108, 0.676]
Dims [256, 256]
Figure 7

#1: Range [0, 1]
Dims [256, 256]

#2: Range [5.04e-06, 0.788]
Dims [256, 256]
Figure 8

#1: Range [0, 1]
Dimensions [256, 256]

#2: Range [2.62e-05, 0.934]
Dimensions [256, 256]
Figure 9

#1: Range [0, 1]
Dims [256, 256]

#2: Range [5.05e-005, 1.09]
Dims [256, 256]
Figure 10

#1: Range [0, 1]
Dims [256, 256]

#2: Range [8.78e-008, 1.22]
Dims [256, 256]
Figure 11

#1: Range [0, 1]
Dims [256, 256]

#2: Range [4.79e-005, 1.27]
Dims [256, 256]
513

Figure 13

#1: Range [0, 1]
Dims [256, 256]

#2: Range [1.76e-005, 1.26]
Dims [256, 256]
Figure 14

#1: Range [0, 1]
Dims [256, 256]

#2: Range [2.24e-005, 1.28]
Dims [256, 256]
2049

Figure 15

#1: Range [0, 1]
Dims [256, 256]

#2: Range [0.000347, 1.27]
Dims [256, 256]
4097

Figure 16

#1: Range [0, 1]
    Dims [256, 256]

#2: Range [0.000592, 1.23]
    Dims [256, 256]
16385

Figure 18

#1: Range [0, 1]
Dims [256, 256]

#2: Range [0.000365, 1.1]
Dims [256, 256]
Figure 19

#1: Range [0, 1]
Dims [256, 256]

#2: Range [0.0248, 1.03]
Dims [256, 256]
Some important Fourier Transforms
Bracewell’s pictorial dictionary of Fourier transform pairs

<table>
<thead>
<tr>
<th>Name</th>
<th>Signal</th>
<th>Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>impulse</td>
<td>$\delta(x)$</td>
<td>$1$</td>
</tr>
<tr>
<td>shifted impulse</td>
<td>$\delta(x - u)$</td>
<td>$e^{-j\omega u}$</td>
</tr>
<tr>
<td>box filter</td>
<td>$\text{box}(x/a)$</td>
<td>$a \text{sinc}(a\omega)$</td>
</tr>
<tr>
<td>tent</td>
<td>$\text{tent}(x/a)$</td>
<td>$a \text{sinc}^2(a\omega)$</td>
</tr>
<tr>
<td>Gaussian</td>
<td>$G(x; \sigma)$</td>
<td>$\frac{\sqrt{2\pi}}{\sigma} G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>Laplacian of Gaussian</td>
<td>$(\frac{x^2}{\sigma^2} - \frac{1}{\sigma^2})G(x; \sigma)$</td>
<td>$-\frac{\sqrt{2\pi}}{\sigma} \omega^2 G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>Gabor</td>
<td>$\cos(\omega_0 x)G(x; \sigma)$</td>
<td>$\frac{\sqrt{2\pi}}{\sigma} G(\omega \pm \omega_0; \sigma^{-1})$</td>
</tr>
<tr>
<td>unsharp mask</td>
<td>$(1 + \gamma)\delta(x) - \gamma G(x; \sigma)$</td>
<td>$(1 + \gamma) - \frac{\sqrt{2\pi}}{\sigma \gamma} G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>windowed sinc</td>
<td>$\frac{r \cos(x/(aW))}{\text{sinc}(x/a)}$</td>
<td>(see Figure 3.29)</td>
</tr>
</tbody>
</table>

**Table 3.2** Some useful (continuous) Fourier transform pairs: The dashed line in the Fourier transform of the shifted impulse indicates its (linear) phase. All other transforms have zero phase (they are real-valued). Note that the figures are not necessarily drawn to scale but are drawn to illustrate the general shape and characteristics of the filter or its response. In particular, the Laplacian of Gaussian is drawn inverted because it resembles more a “Mexican hat”, as it is sometimes called.
Some important Fourier Transforms

Image

Magnitude FT
Some important Fourier Transforms

Image

Magnitude FT
The Fourier Transform of some important images

Image

Log(1+Magnitude FT)
How to interpret a Fourier Spectrum

- Horizontal orientation
- Vertical orientation
- 45 deg.
- 0 \( f_{\text{max}} \)
- \( f_x \) in cycles/image

Low spatial frequencies

High spatial frequencies

Log power spectrum
Fourier Amplitude Spectrum

A

B

C

fx(cycles/image pixel size)
Fourier transform magnitude
Masking out the fundamental and harmonics from periodic pillars
Outline

• Linear filtering
• Fourier Transform
• **Human spatial frequency sensitivity**
• Phase
• Sampling and Aliasing
• Spatially localized analysis
Although this is a complex system, tools from linear systems analysis can provide some useful insights...

From M. Lewicky
Figure 1. Stimulus presentation scheme. The stimuli were originally calibrated to be seen at a distance of 150 cm in a 19" display.
Contrast Sensitivity Function

From:
http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/channels/channels.html
Contrast Sensitivity Function

A demo of human contrast sensitivity as a function of spatial frequency. Frequency rises from left to right at a constant rate. Contrast drops from bottom to top at a constant rate. The bars are visible further up for middle frequencies, showing these are more salient to the human visual system.
Contrast Sensitivity Function

Blackmore & Campbell (1969)

Maximum sensitivity

~ 6 cycles / degree of visual angle
Laplacian

An illusion by Vasarely, left, and a bandpass filtered version, right.

Figure 1.2:  
a) Schema of the horizontal cell layer of the retina.  
b) RC analog network.

Neuromorphic circuits
Human Visual Perception

Blur image → Spatial frequency channels → Sharp image

Low spatial frequency

Medium spatial frequency

High spatial frequency
Hybrid Images

Oliva & Schyns
Hybrid Images
Hybrid Images
Outline

• Linear filtering
• Fourier Transform
• Human spatial frequency sensitivity
• Phase
• Sampling and Aliasing
• Spatially localized analysis
Phase and Magnitude

• Curious fact
  – all natural images have about the same magnitude transform
  – hence, phase seems to matter, but magnitude largely doesn’t

• Demonstration
  – Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?
This is the magnitude transform of the cheetah picture.
This is the phase transform of the cheetah pic.
This is the magnitude transform of the zebra pic
This is the phase transform of the zebra pic
Reconstruction with zebra phase, cheetah magnitude
Reconstruction with cheetah phase, zebra magnitude
Phase and Magnitude

Image with cheetah phase (and zebra magnitude)

Image with zebra phase (and cheetah magnitude)
Randomizing the phase
Outline

• Linear filtering
• Fourier Transform
• Human spatial frequency sensitivity
• Phase
• Sampling and Aliasing
• Spatially localized analysis
Sampling

Continuous world

Pixels
Sampling
Sampling
Sampling
What will be the best sampling pattern in 2D?

Images from: http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/retina/retina.html
The Fourier transform of a sampled signal

$$F(\text{Sample}_{2D}(f(x,y))) = F \left( f(x,y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i,y-j) \right)$$

$$= F(f(x,y))^* F \left( \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i,y-j) \right)$$

$$= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F(u-i,v-j)$$

Sampling function

FT(Sampling function)

Sampled image

Downsampling

FT(sampled image)
Sampling function

FT(Sampling function)

Sampled image

Downsampling

FT(sampled image)
Sampling function

FT(Sampling function)

Sampled image

Downsampling

FT(sampled image)
Sampling without smoothing. Top row shows the images, sampled at every second pixel to get the next.
Sampling with smoothing. Top row shows the images. We get the next image by smoothing the image with a Gaussian with sigma 1 pixel, then sampling at every second pixel to get the next.

<table>
<thead>
<tr>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image 1" /></td>
<td><img src="image2.png" alt="Image 2" /></td>
<td><img src="image3.png" alt="Image 3" /></td>
<td><img src="image4.png" alt="Image 4" /></td>
<td><img src="image5.png" alt="Image 5" /></td>
</tr>
</tbody>
</table>
Sampling with smoothing. Top row shows the images. We get the next image by smoothing the image with a Gaussian with sigma 1.4 pixels, then sampling at every second pixel to get the next.

256x256   128x128   64x64   32x32   16x16
Outline

• Linear filtering
• Fourier Transform
• Human spatial frequency sensitivity
• Phase
• Sampling and Aliasing
• Spatially localized analysis
What is a good representation for image analysis?

• Fourier transform domain tells you “what” (textural properties), but not “where”.

• Pixel domain representation tells you “where” (pixel location), but not “what”.

• Want an image representation that gives you a local description of image events—what is happening where.