Recap of last lecture (big picture)

- **goal**: match query image to a stored library of reference images

- **cannot use raw pixels**:
  - too many
  - too unstable (lighting, shadow, occlusion, scale change, perspective change...)

- **three step plan**:
  1. identify interest points (easy to identify structures): visual words
  2. extract local feature descriptors (characterises interest points): word definitions
  3. match images based by comparing sets of feature descriptors
Recap of last lecture *(edge detection)*
Recap of last lecture \((\text{edge detection})\)

\[ I(x) \]
Recap of last lecture (edge detection)

image $\rightarrow$ Gaussian blur

$I(x) \otimes g_\sigma(x)$
Recap of last lecture (edge detection)

image $\rightarrow$ Gaussian blur $\rightarrow$ differentiate

$I(x) \otimes g_\sigma(x) \rightarrow \frac{d}{dx}$
Recap of last lecture (edge detection)

image $\rightarrow$ Gaussian blur $\rightarrow$ differentiate $\rightarrow$ locate extrema

$I(x) \otimes g_{\sigma}(x) \frac{d}{dx}$
Recap of last lecture (edge detection)

image $\rightarrow$ Gaussian blur $\rightarrow$ differentiate $\rightarrow$ locate extrema $\rightarrow$ threshold

$I(x) \otimes g_\sigma(x) \quad \frac{d}{dx}$
Recap of last lecture (edge detection)

image $\rightarrow$ Gaussian blur $\rightarrow$ differentiate $\rightarrow$ locate extrema $\rightarrow$ threshold

$I(x) \otimes g_\sigma(x) \frac{d}{dx}$
convolution
lowpass filter

frequency
Recap of last lecture (edge detection)

Image $I(x)$

- Gaussian blur $\otimes g_\sigma(x)$
- Differentiate $\frac{d}{dx}$
- Locate extrema
- Threshold
Recap of last lecture (edge detection)

image $\rightarrow$ Gaussian blur $\rightarrow$ differentiate $\rightarrow$ locate extrema $\rightarrow$ threshold

$I(x) \otimes g_\sigma(x)$

convolution lowpass filter

$\frac{d}{dx}$

convolution highpass filter

$\propto$ $\propto$ frequency $\propto$ $\propto$
Recap of last lecture (edge detection)

image $\rightarrow$ Gaussian blur $\rightarrow$ differentiate $\rightarrow$ locate extrema $\rightarrow$ threshold

$I(x) \otimes g_\sigma(x) + \frac{d}{dx}$

convolution lowpass filter

highpass filter

$X$

= bandpass filter
Recap of last lecture *(edge detection)*

Image → Gaussian blur → differentiate → locate extrema → threshold

\[ I(x) \xrightarrow{\otimes g_\sigma(x)} \frac{d}{dx} \]

Convolution
Lowpass filter
Highpass filter

\[ X \]

Frequency

\[ \text{bandpass filter} \]

Linear operations are commutative and associative
Recap of last lecture (edge detection)

linear operations are commutative and associative
Recap of last lecture (edge detection)

linear operations are commutative and associative

\[ I(x) \circ \frac{d}{dx} \circ \bigotimes g_\sigma(x) \]

image \(\rightarrow\) Gaussian blur \(\rightarrow\) differentiate \(\rightarrow\) locate extrema \(\rightarrow\) threshold

\[ I(x) \circ \bigotimes g_\sigma(x) \rightarrow \frac{d}{dx} \bigotimes g_\sigma(x) \]

Convolution lowpass filter \(\rightarrow\) highpass filter \(\rightarrow\) bandpass filter
Recap of last lecture (edge detection)

- Image \( \rightarrow \) Gaussian blur \( \rightarrow \) differentiate \( \rightarrow \) locate extrema \( \rightarrow \) threshold.

\[
I(x) \quad \otimes g_\sigma(x) \quad \frac{d}{dx} \quad \text{lowpass filter} \quad \text{highpass filter}
\]

\[
\text{convolution} \quad \text{convolution}
\]

\[
\text{linear operations are commutative and associative}
\]

- Image \( \rightarrow \) blur with derivative of Gaussian

\[
I(x) \quad \otimes \frac{d}{dx} g_\sigma(x)
\]
Recap of last lecture (edge detection)

linear operations are commutative and associative

image $\rightarrow$ blur with derivative of Gaussian

$$I(x) \quad \otimes \quad \frac{d}{dx} g_\sigma(x)$$
Today

• Is there a correct level of blur for edge detection?

• Edge detection in 2D
  – Canny edge detection
  – Marr Hildreth edge detection

• Moving beyond edges
  – why edges aren’t sufficient for computer vision
Discrete approximation of the Gaussian filter

N = number of pixels in 1D image

N = number of pixels in 1D image

Continuous Gaussian

Discrete approx

Trade-off between comp. complexity and high-frequency artifacts
Truncation of the Gaussian filter

N = number of pixels in 1D image
K = truncated approximation

peak height

continuous Gaussian

peak height
1000

discrete approx

1 2 3 4 5 6 7 8 9 ...K

Trade-off between comp. complexity and high-frequency artifacts
Truncation of the Gaussian filter

N = number of pixels in 1D image
K = truncated approximation

⇒ Trade-off between computational complexity and high-frequency artifacts
Convolution

\[ G_{\sigma}(x, y) \]

\[ \otimes \]

\[ I(x, y) \]

\[ S(x, y) = G_{\sigma}(x, y) \otimes I(x, y) \]
$S(x, y) = G_\sigma(x, y) \otimes I(x, y)$
Is this the right thing to do?

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) \]
Is this the right thing to do?

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v) \]
Convolution

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v) \]
Convolution

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v) \]

convolution = correlation when the filter is symmetric

e.g. isotropic Gaussian
Convolution

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v) I(u, v) \]
Convolution

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v) \]
Convolution

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v) I(u, v) \]
Convolution

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v) I(u, v) \]
What’s the computational cost of this naïve approach?

\[
S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v)
\]
What’s the computational cost of this naïve approach?

The computational cost of this approach is given by:

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v) \]

The computational cost is then:

\[ \text{computational cost} = NMK^2 \]

NB. \[ K = 2n+1 \] in your notes.
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) \]

![Filter](image)

\[ \otimes \]

![Image](image)

\[ I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v) \]
Can we speed this up when the filter is separable?

\[
G_\sigma(x, y) = g_\sigma(x)g_\sigma(y)
\]

\[
\begin{array}{ccc}
  a & b & c \\
  d & e & f \\
  g & h & i \\
\end{array} = \begin{array}{ccc}
  a' & b' & c' \\
  a & b & c \\
  b' & d & e \\
  c' & c & e \\
\end{array}
\]

\[
I(x, y)
\]

\[
S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v G_\sigma(x - u, y - v)I(u, v)
\]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[
\begin{array}{ccc}
  a & b & c \\
  d & e & f \\
  g & h & i \\
\end{array}
\]

\[
\begin{array}{ccc}
  a' & b' & c' \\
  b' & d' & e' \\
  c' & d' & e' \\
\end{array}
\]

\[
\begin{array}{ccc}
  a & a & b & c \\
  d & e & f \\
  g & h & i \\
\end{array} \otimes
\begin{array}{ccc}
  a' & a' & b' & c' \\
  b' & b' & d' & e' \\
  c' & c' & e' & f \\
\end{array}
\]

\[ I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u \sum_v g_\sigma(x - u)g_\sigma(y - v)I(u, v) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = \sum_u g_\sigma(x - u) \sum_v g_\sigma(y - v) I(u, v) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x) g_\sigma(y) \]

\[ I(x, y) \odot g(y) \odot I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \otimes g(y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \otimes G_\sigma(x, y) = \begin{pmatrix} a' & b' & c' \\ a & b & c \\ a & b & c \end{pmatrix} \otimes \begin{pmatrix} a' & a & b & c' \\ a & b & c & d \\ a & b & c & d \end{pmatrix} \]

\[ = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \]

\[ \rightarrow g(y) \otimes I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_{\sigma}(x, y) = g_{\sigma}(x)g_{\sigma}(y) \]

\[ I(x, y) = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \]

\[ G_{\sigma}(x, y) \otimes I(x, y) = \begin{pmatrix} a' & a & b & c \\ b' & d & e & f \\ c' & g & h & i \end{pmatrix} \]

\[ S(x, y) = G_{\sigma}(x, y) \otimes I(x, y) = g_{\sigma}(x) \otimes g_{\sigma}(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \text{ filter} \]

\[ g(y) \otimes I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \otimes g(y) \otimes I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \quad = \quad \begin{pmatrix} c' \\ b' \\ a' \end{pmatrix} \quad \begin{pmatrix} g(y) \otimes I(x, y) \end{pmatrix} \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \ast g(y) \ast I(x, y) = S(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \quad \overset{\otimes}{\longrightarrow} \quad g(y) \otimes I(x, y) \quad \overset{\otimes}{\longrightarrow} \quad S(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \Rightarrow g(y) \otimes I(x, y) \Rightarrow S(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[
\begin{array}{cccc}
  a & b & c \\
  d & e & f \\
  g & h & i \\
\end{array}
\]

\[
\begin{array}{cccc}
  a' & b' & c' \\
  a' & a & b & c \\
  b' & b & d & e \\
  c' & c & d & e \\
\end{array}
\]

\[ g(y) \otimes I(x, y) \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[
\begin{array}{ccc}
  a & b & c \\
  d & e & f \\
  g & h & i \\
\end{array}
\]
\[
\begin{array}{ccc}
  a' & b' & c' \\
  b' & c' & d' \\
  c' & d' & e' \\
\end{array}
\]

\[ \triangleright \]

\[ I(x, y) \otimes g(y) \otimes I(x, y) = S(x, y) \]

\[ G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_{\sigma}(x, y) = g_{\sigma}(x)g_{\sigma}(y) \]

\[ I(x, y) \quad \Rightarrow \quad g(y) \otimes I(x, y) \quad \Rightarrow \quad S(x, y) \]

\[ S(x, y) = G_{\sigma}(x, y) \otimes I(x, y) = g_{\sigma}(x) \otimes g_{\sigma}(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ \begin{array}{ccc}
  \text{image} & \otimes & \text{filtered image} \\
  I(x, y) & \rightarrow & S(x, y) \\
 \end{array} \]

\[ S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]
Can we speed this up when the filter is separable?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[
\begin{array}{ccc}
\text{filter} & & \\
\begin{array}{ccc}
 a & b & c \\
 d & e & f \\
 g & h & i \\
\end{array} & \Rightarrow & \\
\begin{array}{ccc}
 a' & b' & c' \\
 b' & d & e \\
 c' & c & f \\
\end{array}
\end{array}
\]

\[
I(x, y) \quad \otimes \quad g(y) \otimes I(x, y) \quad \Rightarrow \quad S(x, y)
\]

\[
S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y)
\]
What is the new computational cost?

\( G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \)

\[
\begin{array}{cccc}
  a & b & c \\
  d & e & f \\
  g & h & i \\
\end{array} \quad \mapsto \quad \begin{array}{cccc}
  c' & b' & a' \\
  a' & b' & c' \\
  b' & c' & a' \\
\end{array}
\]

\[
N \text{ by } M \text{ pixel image} \quad \mapsto \quad \text{K dimensional vectors} \quad \mapsto \quad \text{filtered image}
\]

\[
S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y)
\]
What is the new computational cost?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

\[ I(x, y) \otimes G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \]

new computational cost is \( = 2N MK \)
What is the new computational cost?

\[ G_\sigma(x, y) = g_\sigma(x)g_\sigma(y) \]

old computational cost was = \( NMK \)

new computational cost is = \( 2NMK \)

\( S(x, y) = G_\sigma(x, y) \otimes I(x, y) = g_\sigma(x) \otimes g_\sigma(y) \otimes I(x, y) \)

new computational cost is = \( 2NMK \)

old computational cost was = \( NMK^2 \)
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Which way do you think the edge is moving?
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Aperture problem
Problem with edge-features

Edges can only tell us about the component of motion perpendicular to the edge
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?
What about corners?