Agenda

• Steerable filters (recap)
• Harris corner detector
• Binomial filters
Steerable filters

**Definition**: Steerable filters are a class of filters where a filter of arbitrary orientation is synthesized as a linear combination of a set of basis filters [Freeman and Adelson, 1991].
Steerable filters

**Definition**: Steerable filters are a class of filters where a filter of arbitrary orientation is synthesized as a linear combination of a set of basis filters [Freeman and Adelson, 1991].

*second derivative of Gaussian*
Steerable derivatives of Gaussian
Steerable filter architecture

\[
\left( \sum_i k_i(\theta)B_i \right) \ast I \equiv \sum_i k_i(\theta)I_i^B
\]
Derivatives of Gaussian frequency spectrum

\[ G_n(x) \leftrightarrow (j\omega)^n \hat{G}(\omega) \]

normalized magnitude spectrum
Agenda

- Steerable filters (recap)
- Harris corner detector
- Binomial filters
Motivation

“What stuff in one image matches with stuff in another?”
Motivation

“What stuff in one image matches with stuff in another?”
What makes a good feature?

**Repeatability**
Same feature can be found in other images despite geometric and photometric transformations.

**Saliency**
Each feature is distinctive.

**Compactness and efficiency**
Many fewer features than image pixels.

**Locality**
Feature occupies a relatively small area of the image; robust to clutter and occlusion.
# Detectors

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moravec</td>
<td>[Moravec ‘77]</td>
</tr>
<tr>
<td>Hessian</td>
<td>[Beaudet ‘78]</td>
</tr>
<tr>
<td>Harris/Forstner</td>
<td>[Harris ‘88, Forstner ‘87]</td>
</tr>
<tr>
<td>Laplacian</td>
<td>[Lindeberg ‘98]</td>
</tr>
<tr>
<td>DoG</td>
<td>[Lowe 1999]</td>
</tr>
<tr>
<td>Harris-/Hessian-Laplace</td>
<td>[Mikolajczyk &amp; Schmid ‘01]</td>
</tr>
<tr>
<td>Harris-/Hessian-Affine</td>
<td>[Mikolajczyk &amp; Schmid ‘04]</td>
</tr>
<tr>
<td>EBR and IBR</td>
<td>[Tuytelaars &amp; Van Gool ‘04]</td>
</tr>
<tr>
<td>MSER</td>
<td>[Matas ‘02]</td>
</tr>
<tr>
<td>Salient Regions</td>
<td>[Kadir &amp; Brady ‘01]</td>
</tr>
<tr>
<td>FAST</td>
<td>[Rosten et al. ‘2010]</td>
</tr>
</tbody>
</table>

Many Others…
# Detectors

<table>
<thead>
<tr>
<th>Detector</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moravec</td>
<td>Moravec ‘77</td>
</tr>
<tr>
<td>Hessian</td>
<td>Beaudet ‘78</td>
</tr>
<tr>
<td>Harris/Forstner</td>
<td>Harris ‘88, Forstner ‘87</td>
</tr>
<tr>
<td>Laplacian</td>
<td>Lindeberg ‘98</td>
</tr>
<tr>
<td>DoG</td>
<td>Lowe 1999</td>
</tr>
<tr>
<td>Harris-/Hessian-Laplace</td>
<td>Mikolajczyk &amp; Schmid ‘01</td>
</tr>
<tr>
<td>Harris-/Hessian-Affine</td>
<td>Mikolajczyk &amp; Schmid ‘04</td>
</tr>
<tr>
<td>EBR and IBR</td>
<td>Tuytelaars &amp; Van Gool ‘04</td>
</tr>
<tr>
<td>MSER</td>
<td>Matas ‘02</td>
</tr>
<tr>
<td>Salient Regions</td>
<td>Kadir &amp; Brady ‘01</td>
</tr>
<tr>
<td>FAST</td>
<td>Rosten et al. ‘2010</td>
</tr>
</tbody>
</table>

Many Others…
Harris corner?
Harris detector: Intuition

- Analyze local variation of signal.
- “Corner”: Shifting window in any direction yields a large change in appearance.
Harris detector: Intuition

- Analyze local variation of signal.
- "Corner": Shifting window in any direction yields a large change in appearance.
Harris detector: Intuition

- Analyze local variation of signal.
- "Corner": Shifting window in **any direction** yields a **large change** in appearance.

"flat" region: no change in all directions

"edge": no change along the edge direction
Harris detector: Intuition

- Analyze local variation of signal.
- "Corner": Shifting window in any direction yields a large change in appearance.

"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions
Harris detector: Derivation

Variation of intensity for the shift \((\Delta x, \Delta y)\):

\[
E(\Delta x, \Delta y) = \sum_{x,y} w(x, y) [I(x, y) - I(x + \Delta x, y + \Delta y)]^2
\]
Harris detector: Derivation

Variation of intensity for the shift \((\Delta x, \Delta y)\):

\[
E(\Delta x, \Delta y) = \sum_{x,y} w(x, y)[I(x, y) - I(x + \Delta x, y + \Delta y)]^2
\]

For nearly constant patches:

\[
E(\Delta x, \Delta y) \approx 0
\]
Harris detector: Derivation

Variation of intensity for the shift \((\Delta x, \Delta y)\):

\[
E(\Delta x, \Delta y) = \sum_{x,y} w(x, y)[I(x, y) - I(x + \Delta x, y + \Delta y)]^2
\]

For “edge” patches:

\[
E(\Delta x, \Delta y)
\] variation large along one orientation
Harris detector: Derivation

Variation of intensity for the shift \((\Delta x, \Delta y)\):

\[
E(\Delta x, \Delta y) = \sum_{x,y} w(x, y)[I(x, y) - I(x + \Delta x, y + \Delta y)]^2
\]

For “corner” patches:

\[
E(\Delta x, \Delta y)
\]

variation large along two orientations
Review: Taylor series

\[ f(x + u, y + v) = f(x, y) + f_x(x, y)u + f_y(x, y)v \]

\[ + \frac{1}{2} \left[ f_{xx}(x, y)u^2 + f_{xy}(x, y)uv + f_{yy}(x, y)v^2 \right] \]

\[ + \text{h.o.t.} \]
Review: Taylor series

\[ f(x + u, y + v) \approx f(x, y) + f_x(x, y)u + f_y(x, y)v \]

\[ + \frac{1}{2} [f_{xx}(x, y)u^2 + f_{xy}(x, y)uv + f_{yy}(x, y)v^2] \]

+ h.o.t.
Harris detector: Derivation

\[ E(\Delta x, \Delta y) = \sum_{x,y} [I(x, y) - I(x + \Delta x, y + \Delta y)]^2 \]
Harris detector: Derivation

\[
E(\Delta x, \Delta y) = \sum_{x,y} [I(x, y) - I(x + \Delta x, y + \Delta y)]^2
\]

Taylor series expansion

\[
\approx \sum_{x,y} [I(x, y) - I(x, y) - I_x(x, y)\Delta x - I_y(x, y)\Delta y]^2
\]
Harris detector: Derivation

\[ E(\Delta x, \Delta y) = \sum_{x,y} [I(x, y) - I(x + \Delta x, y + \Delta y)]^2 \]

Taylor series expansion

\[ \approx \sum_{x,y} [I(x, y) - I(x, y) - I_x(x, y) \Delta x - I_y(x, y) \Delta y]^2 \]
Harris detector: Derivation

\[ E(\Delta x, \Delta y) = \sum_{x,y} [I(x, y) - I(x + \Delta x, y + \Delta y)]^2 \]

Taylor series expansion

\[ \approx \sum_{x,y} [I(x, y) - I(x, y) - I_x(x, y)\Delta x - I_y(x, y)\Delta y]^2 \]

expand

\[ = \sum_{x,y} [I_x(x, y)^2\Delta x^2 + 2I_x(x, y)I_y(x, y)\Delta x\Delta y + I_y(x, y)^2\Delta y] \]
Harris detector: Derivation

\[ E(\Delta x, \Delta y) = \sum_{x,y} [I(x, y) - I(x + \Delta x, y + \Delta y)]^2 \]

- Taylor series expansion

\[ \approx \sum_{x,y} [I(x, y) - I(x, y) - I_x(x, y)\Delta x - I_y(x, y)\Delta y]^2 \]

- Expand

\[ = \sum_{x,y} [I_x(x, y)^2\Delta x^2 + 2I_x(x, y)I_y(x, y)\Delta x\Delta y + I_y(x, y)^2\Delta y] \]

- Rewrite as matrix
Harris detector: Derivation

\[ E(\Delta x, \Delta y) = (\Delta x \quad \Delta y) M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \]

where

\[ M = \sum_{x,y} w(x,y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \]

\( M \) captures the variation of the gradients within the local patch

Thursday, February 9, 2012

M is generally a positive definite matrix

Note: Windowing function reintroduced and products of derivatives NOT second derivatives
M: second-order moment, covariance matrix or scatter matrix
Back to intuition

1. Treat gradient vectors as a set of 2D points, $(I_x, I_y)$.
2. Compute scatter matrix/ellipse.
3. Analyze scatter matrix/ellipse shape.
Intuition

“Flat”

“Edge”

“Corner”

notice distribution of gradients vary for different types of patches
Intuition

“Flat”

\[ \lambda_1 \approx \lambda_2 \]
small

“Ix

“Iy

“Edge”

\[ \lambda_1 \gg \lambda_2 \]

“Corner”

\[ \lambda_1 \approx \lambda_2 \]
large

Eigenvalues, \( \lambda_1 \) and \( \lambda_1 \), of scatter matrix, \( M \), quantify variation in the principle directions (i.e., eigenvectors of \( M \)) of the data.

Thursday, February 9, 2012
Corner response function

\[ r = \det \mathbf{M} - k(\text{trace } \mathbf{M})^2 \]

where

\[ \det \mathbf{M} = ? \quad \text{trace } \mathbf{M} = ? \]

\( k \) is empirically set constant: 0.04 - 0.06
Corner response function

\[ r = \det M - k (\text{trace } M)^2 \]

where

\[ \det M = \lambda_1 \lambda_2 \quad \text{trace } M = ? \]

\( k \) is empirically set constant: 0.04 - 0.06
Corner response function

\[ r = \det \mathbf{M} - k (\text{trace } \mathbf{M})^2 \]

where

\[ \det \mathbf{M} = \lambda_1 \lambda_2 \quad \text{trace } \mathbf{M} = \lambda_1 + \lambda_2 \]

\[ k \text{ is empirically set constant: } 0.04 - 0.06 \]
Classification of image patches

Classification of image points using eigenvalues of $\mathbf{M}$ (or $r$)
Classification of image points using eigenvalues of $\mathbf{M}$ (or $r$)

- $E$ is almost constant in all directions
- $\lambda_1$ and $\lambda_2$ are small
- $|r|$ small

Thursday, February 9, 2012
Classification of image patches

Classification of image points using eigenvalues of $\mathbf{M}$ (or $r$)

- $E$ increases about the horizontal orientation
  - $\lambda_2 \gg \lambda_1$
  - $r < 0$

- $E$ is almost constant in all directions
  - $\lambda_1$ and $\lambda_2$ are small
  - $|r|$ small

- $E$ increases about the vertical orientation
  - $\lambda_1 \gg \lambda_2$
  - $r < 0$
Classification of image patches

Classification of image points using eigenvalues of $\mathbf{M}$ (or $r$)

- $E$ increases about the horizontal orientation
  - $\lambda_2 \gg \lambda_1$
  - $r < 0$

- $E$ is almost constant in all directions
  - $\lambda_1$ and $\lambda_2$ are small
  - $|r|$ small

- $E$ increases in all directions
  - $\lambda_1 \approx \lambda_2$
  - $\lambda_1$ and $\lambda_2$ are large,
    - $r > 0$ and $|r| >> 0$

- $E$ increases about the vertical orientation
  - $\lambda_1 >> \lambda_2$
  - $r < 0$
Harris corner example
Extension to video

Feature detectors operating at multiple scales
Agenda

- Steerable filters (recap)
- Harris corner detector
- Binomial filters
Binomial filter

- Class of smoothing filter (\(\approx\) Gaussian).
- Attenuates high-frequencies and retains low frequencies.
- Filter set generated by a single filter combined with itself.
Starting point

• Generating filter: \( B_1 = \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix} \)

• Binomial filter set:

\[
\left\{ B_R = \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix} \ast \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix} \ast \ldots \ast \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix} \right\}
\]

• By the Central-Limit Theorem, the shape of the Binomial filters tend to a Gaussian as \( R \) increases.

Thursday, February 9, 2012

Notice that the binomial filters are closed under convolution.

Sum of iid random variables tend to a Gaussian (or normal distribution) as the number of random variables gets larger.
Binomial filter: Spatial domain

\[ n \]

\[ n \neq 0 \]
Binomial filter: Spatial domain

\[ B_2 = 0.5^2 [1 1]^2 \]
Equivalence to ...

\[
\begin{array}{cccccc}
\mathbf{B}_1 & 1 & 1 \\
\mathbf{B}_2 & 1 & 2 & 1 \\
\mathbf{B}_3 & 1 & 3 & 3 & 1 \\
\mathbf{B}_4 & 1 & 4 & 6 & 4 & 1 \\
\mathbf{B}_5 & 1 & 5 & 10 & 10 & 5 & 1 \\
\mathbf{B}_6 & 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
\end{array}
\]

...
| B<sub>1</sub> | 1 1          |
| B<sub>2</sub> | 1 2 1        |
| B<sub>3</sub> | 1 3 3 1      |
| B<sub>4</sub> | 1 4 6 4 1    |
| B<sub>5</sub> | 1 5 10 10 5 1|
| B<sub>6</sub> | 1 6 15 20 15 6 1 |

...
Equivalence to ...

\[\begin{array}{c}
B_1 \\
B_2 \\
B_3 \\
B_4 \\
B_5 \\
B_6
\end{array}\]

\[
\begin{array}{cccc}
1 & 1 \\
1 & 2 & 1 \\
1 & 3 & 3 & 1 \\
1 & 4 & 6 & 4 & 1 \\
1 & 5 & 10 & 10 & 5 & 1 \\
1 & 6 & 15 & 20 & 15 & 6 & 1 \\
\end{array}
\]

\[\begin{array}{c}
Pascal's Triangle \\
\end{array}\]

\[
\binom{n}{r}
\]

nth row
rth element

Thursday, February 9, 2012
Binomial derivatives

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
...
Binomial derivatives

-1 1
-1 0 1
-1 -1 1 1
-1 -2 0 2 1
-1 -3 -2 2 3 1
-1 -4 -5 0 5 4 1
...

Thursday, February 9, 2012
Frequency domain

$B_2 = (1 \ 2 \ 1)$ (not normalized)

DTFT:
Frequency domain

\[ \mathbf{B}_2 = (1 \ 2 \ 1) \quad \text{(not normalized)} \]

DTFT:
\[ \hat{\mathbf{B}}(\omega) = \sum_{n=-\infty}^{\infty} \mathbf{B}_2[n]e^{-i\omega n} \]

Note: using i instead of j for complex number representation
Frequency domain

\[ B_2 = (1 \ 2 \ 1) \quad \text{(not normalized)} \]

**DTFT:**

\[ \hat{B}(\omega) = \sum_{n=-\infty}^{\infty} B_2[n] e^{-i\omega n} \]

\[ = \sum_{n=-\infty}^{\infty} \left( \delta[n + 1] + 2\delta[n] + \delta[n + 1] \right) e^{-i\omega n} \]

Note: using \( i \) instead of \( j \) for complex number representation
Frequency domain

\[ \mathbf{B}_2 = \begin{pmatrix} 1 & 2 & 1 \end{pmatrix} \] (not normalized)

DTFT:

\[
\hat{\mathbf{B}}(\omega) = \sum_{n=-\infty}^{\infty} \mathbf{B}_2[n] e^{-i\omega n}
\]

\[
= \sum_{n=-\infty}^{\infty} \left( \delta[n+1] + 2\delta[n] + \delta[n+1] \right) e^{-i\omega n}
\]

\[
= \sum_{n=-\infty}^{\infty} \delta[n+1] e^{-i\omega n} + \sum_{n=-\infty}^{\infty} 2\delta[n] e^{-i\omega n} + \sum_{n=-\infty}^{\infty} \delta[n+1] e^{-i\omega n}
\]

Thursday, February 9, 2012

Note: using \( i \) instead of \( j \) for complex number representation
Frequency domain

\[ \hat{\mathbf{B}}(\omega) = \sum_{n=-\infty}^{\infty} \mathbf{B}_2[n] e^{-i\omega n} \]

\[ = \sum_{n=-\infty}^{\infty} (\delta[n + 1] + 2\delta[n] + \delta[n + 1]) e^{-i\omega n} \]

\[ = \sum_{n=-\infty}^{\infty} \delta[n + 1] e^{-i\omega n} + \sum_{n=-\infty}^{\infty} 2\delta[n] e^{-i\omega n} + \sum_{n=-\infty}^{\infty} \delta[n + 1] e^{-i\omega n} \]

\[ = 2 + e^{-i\omega} + e^{i\omega} \]

\[ \mathbf{B}_2 = (1 \ 2 \ 1) \quad \text{(not normalized)} \]
Frequency domain

\[ \mathbf{B}_2 = (1 \ 2 \ 1) \quad \text{(not normalized)} \]

**DTFT:**

\[
\hat{\mathbf{B}}(\omega) = \sum_{n=-\infty}^{\infty} B_2[n] e^{-i\omega n}
\]

\[
= \sum_{n=-\infty}^{\infty} (\delta[n + 1] + 2\delta[n] + \delta[n + 1]) e^{-i\omega n}
\]

\[
= \sum_{n=-\infty}^{\infty} \delta[n + 1] e^{-i\omega n} + \sum_{n=-\infty}^{\infty} 2\delta[n] e^{-i\omega n} + \sum_{n=-\infty}^{\infty} \delta[n + 1] e^{-i\omega n}
\]

\[
= 2 + 2 \cos(\omega) \quad \text{Euler’s formula}
\]
Binomial filter
frequency domain

$\omega$

0
Binomial filter
frequency domain

\[ B_2 = \frac{1}{4^1}[1 \ 2 \ 1]^1 \]
2D Binomial filter

- 2D Binomial filters are constructed by 1D horizontal and vertical filters.

\[ B_{2D}^2 = B_x^R \ast B_y^R \]

Example:

\[ B_{2D}^2 = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix} \ast \frac{1}{4} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \]
2D Binomial filter smoothing example
2D Binomial filter smoothing example