Visual Computing:

Image features

Prof. Marc Pollefeys
Correlation
(e.g. Template-matching)

\[ I' = \sum_{j=-k}^{k} \sum_{i=-k}^{k} K(i, j)I(x+i, y+j) \]

Convolution
(e.g. point spread function)

\[ I' = \sum_{j=-k}^{k} \sum_{i=-k}^{k} K(i, j)I(x-i, y-j) \]
Template matching

• Problem: locate an object, described by a template $t(x,y)$, in the image $s(x,y)$

• Example
Template matching (cont.)

- Search for the best match by minimizing mean-squared error
  
  \[ E(p,q) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} \left[ s(x,y) - t(x-p, y-q) \right]^2 \]

  \[ = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |s(x,y)|^2 + \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |t(x,y)|^2 - 2 \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} s(x,y) \cdot t(x-p, y-q) \]

- Equivalently, maximize area correlation

  \[ r(p,q) = s(x,y) \star t(x-p, y-q) = s(p,q) \cdot t(p, q) \]

  \[ \text{area correlation} \]

- Area correlation is equivalent to convolution of image \( s(x,y) \)
  with impulse response \( t(-x,-y) \)
Template matching (cont.)

- From Cauchy-Schwarz inequality

\[ r(p,q) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} s(x,y) \cdot t(x-p, y-q) \leq \sqrt{\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |s(x,y)|^2} \cdot \sqrt{\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |t(x,y)|^2} \]

- Equality, iff \( s(x, y) = \alpha \cdot t(x - p, y - q) \) with \( \alpha \geq 0 \)

- Blockdiagram of template matcher

- Remove mean before template matching to avoid bias towards bright image areas
Edge detection

- Idea (continous-space): Detect local gradient

\[ |\text{grad}(f(x, y))| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2} \]

- Digital image: use finite differences instead

lots of slides borrowed from Bernd Girod
# Edge detection filters

<table>
<thead>
<tr>
<th>Filter</th>
<th>Matrix 1</th>
<th>Matrix 2</th>
</tr>
</thead>
</table>
| Prewitt  | \[
\begin{pmatrix}
  -1 & 0 & 1 \\
  -1 & 0 & 1 \\
  -1 & 0 & 1 \\
\end{pmatrix}
\] | \[
\begin{pmatrix}
  -1 & -1 & -1 \\
  0 & 0 & 0 \\
  1 & 1 & 1 \\
\end{pmatrix}
\] |
| Sobel    | \[
\begin{pmatrix}
  -1 & 0 & 1 \\
  -2 & 0 & 2 \\
  -1 & 0 & 1 \\
\end{pmatrix}
\] | \[
\begin{pmatrix}
  -1 & -2 & -1 \\
  0 & 0 & 0 \\
  1 & 2 & 1 \\
\end{pmatrix}
\] |
| Roberts  | \[
\begin{pmatrix}
  [0] & 1 \\
  -1 & 0 \\
\end{pmatrix}
\] | \[
\begin{pmatrix}
  [1] & 0 \\
  0 & -1 \\
\end{pmatrix}
\] |
Prewitt operator example

Original *Bridge 220x160*

magnitude of image filtered with

\[
\begin{bmatrix}
1 & 0 & 1 \\
1 & [0] & 1 \div \\
1 & 0 & 1 \\
\end{bmatrix}
\]

magnitude of image filtered with

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & [0] & 0 \div \\
1 & 1 & 1 \\
\end{bmatrix}
\]
Prewitt operator example

Original *Bridge* 220x160

magnitude of image filtered with

\[
\begin{bmatrix}
1 & 0 & 1 \\
1 & [0] & 1 \div \\
1 & 0 & 1
\end{bmatrix}
\]

magnitude of image filtered with

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & [0] & 0 \div \\
1 & 1 & 1
\end{bmatrix}
\]
Prewitt operator example

Original Bridge
220x160

magnitude of
image filtered with

\[
\begin{bmatrix}
1 & 0 & 1 \\
1 & [0] & 1^\downarrow \\
1 & 0 & 1
\end{bmatrix}
\]

magnitude of
image filtered with

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & [0] & 0^\downarrow \\
1 & 1 & 1
\end{bmatrix}
\]
Prewitt operator example (cont.)

Original *Billsface* 310x241

**log magnitude of** image filtered with

\[
\begin{bmatrix}
1 & 0 & 1 \\
1 & 0 & 1 \\
1 & 0 & 1 \\
\end{bmatrix}
\]

**log magnitude of** image filtered with

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{bmatrix}
\]
Prewitt operator example (cont.)

Original *Billsface* 310x241

- Log magnitude of image filtered with:
  - 1 0 1
  - 1 [0] $\frac{1}{2}$
  - 1 0 1
- Log magnitude of image filtered with:
  - 1 1 1
  - 0 [0] $\frac{1}{2}$
  - 1 1 1
Prewitt operator example (cont.)

Original *Billsface* 310x241

log magnitude of image filtered with

\[
\begin{bmatrix}
1 & 0 & 1 \\
1 & 0 & 1 \\
1 & 0 & 1
\end{bmatrix}
\]

log magnitude of image filtered with

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
1 & 1 & 1
\end{bmatrix}
\]
Prewitt operator example (cont.)

log sum of squared horizontal and vertical gradients

different thresholds
Sobel operator example

log sum of squared horizontal and vertical gradients

different thresholds
Roberts operator example

Original *Bill's face* 309x240

log magnitude of image filtered with

\[
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]

log magnitude of image filtered with

\[
\begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]
Roberts operator example

Original *Bill's face*  
309x240

log magnitude of image filtered with \[
\begin{bmatrix}
1 & 0 \\
0 & 1^\top
\end{bmatrix}
\]

log magnitude of image filtered with \[
\begin{bmatrix}
0 & 1 \\
1 & 0^\top
\end{bmatrix}
\]
Roberts operator example

Original *Bill's face*  
309x240

log magnitude of image filtered with \[
\begin{bmatrix}
1 & 0 \\
0 & 1^\circ
\end{bmatrix}
\]

log magnitude of image filtered with \[
\begin{bmatrix}
0 & 1 \\
1 & 0^\circ
\end{bmatrix}
\]
Roberts operator example (cont.)

log sum of squared diagonal gradients

different thresholds
Laplacian operator

- Detect discontinuities by considering second derivative

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

- Isotropic (rotationally invariant) operator

- Zero-crossings mark edge location

- Discrete-space approximation by convolution with 3x3 impulse response

\[
\begin{pmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{pmatrix}
\text{ or }
\begin{pmatrix}
1 & 1 & 1 \\
1 & -8 & 1 \\
1 & 1 & 1
\end{pmatrix}
\]
1-d illustration of 2\textsuperscript{nd} derivative edge detector

Edge profile $f(x)$

$f'(x)$

$f''(x)$

Detect zero crossing
Zero crossings of Laplacian

- Sensitive to very fine detail and noise ➔ blur image first
- Responds equally to strong and weak edges ➔ suppress “edges” with low gradient magnitude
Laplacian of Gaussian

- Blurring of image with Gaussian and Laplacian operator can be combined into convolution with Laplacian of Gaussian (LoG) operator

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

- Continuous function and discrete approximation

\[
\sigma = 1.4
\]

\[
\sigma = 1
\]
Zero crossings of LoG

\[ \sigma = 1.4 \]

\[
\begin{align*}
\text{w/o Gaussian} & \\
\sigma = 3 & \\
\sigma = 6 & \\
\end{align*}
\]
Zero crossings of LoG – gradient-based threshold

w/o Gaussian

σ = 1.4

σ = 3

σ = 6
Canny edge detector

1. Smooth image with a Gaussian filter
2. Compute magnitude and angle of gradient (Sobel, Prewitt . . .)
   \[
   M(x, y) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}
   \]
   \[
   \alpha(x, y) = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)
   \]
3. Apply nonmaxima suppression to gradient magnitude image
4. Double thresholding to detect strong and weak edge pixels
5. Reject weak edge pixels not connected with strong edge pixels

[Canny, IEEE Trans. PAMI, 1986]
Canny nonmaxima suppression

- Quantize edge normal to one of four directions: horizontal, -45°, vertical, +45°
- If $M(x,y)$ is smaller than either of its neighbors in edge normal direction $\Rightarrow$ suppress; else keep.
Canny thresholding and suppression of weak edges

- Double-thresholding of gradient magnitude

  \[
  \text{Strong edge: } M(x, y) \geq \theta_{\text{high}} \\
  \text{Weak edge: } \theta_{\text{high}} > M(x, y) \geq \theta_{\text{low}}
  \]

- Typical setting: \( \frac{\theta_{\text{high}}}{\theta_{\text{low}}} = 2...3 \)
- Region labeling of edge pixels
- Reject regions without strong edge pixels

[Canny, IEEE Trans. PAMI, 1986]
Canny edge detector

\[ \sigma = 1.4 \]

\[ \sigma = 3 \]

\[ \sigma = 6 \]
Canny edge detector

\[ \sigma = 1.4 \]
Hough transform

- Problem: fit a straight line (or curve) to a set of edge pixels
- Hough transform (1962): generalized template matching technique
- Consider detection of straight lines \( y = mx + c \)

![Diagram](image)
Hough transform (cont.)

- Subdivide \((m,c)\) plane into discrete “bins,” initialize all bin counts by 0.
- Draw a line in the parameter space \(m,c\) for each edge pixel \(x,y\) and increment bin counts along line.
- Detect peak(s) in \((m,c)\) plane.
Hough transform (cont.)

- Alternative parameterization avoids infinite-slope problem

\[ y = 0 \rightarrow \cos \theta = \frac{\rho}{x} \]
\[ x = 0 \rightarrow \cos(90 - \theta) = \frac{\rho}{y} \]
\[ \downarrow \]
\[ x \cos \theta + y \sin \theta = \rho \]
Hough transform (cont.)

- Alternative parameterization avoids infinite-slope problem

\[ x \cos \theta + y \sin \theta = \rho \]
Hough transform Example A

Original image

Courtesy: P. Salembier
Hough transform Example B

Original image

Courtesy: P. Salembier
Hough transform Example C

Original image

Courtesy: P. Salembier
Hough transform example

Original IC image (256x256)

Edge detection (Prewitt)

94 deg

-110 pixel
Circle detection by Hough transform

- Find circles of fixed radius $r$

- For circles of undetermined radius, use 3-d Hough transform for parameters $(x_0, y_0, r)$
Example: circle detection by Hough transform

Original *blood* image

Prewitt edge detection
Detecting corner points

- Many applications benefit from features localized in \((x, y)\)
- Edges well localized only in one direction \(\rightarrow\) detect corners

Desirable properties of corner detector
- Accurate localization
- Invariance against shift, rotation, scale, brightness change
- Robust against noise, high repeatability
How can we mathematically define corners?

- Local displacement sensitivity
  \[ S(\Delta x, \Delta y) = \sum_{(x,y)\in\text{window}} \left[ f(x, y) - f(x + \Delta x, y + \Delta y) \right]^2 \]

- Linear approximation for small \( \Delta x, \Delta y \)
  
  \[ f(x + \Delta x, y + \Delta y) \approx f(x, y) + f_x(x, y) \Delta x + f_y(x, y) \Delta y \]

\[ S(\Delta x, \Delta y) \approx \sum_{(x,y)\in\text{window}} \left[ \begin{pmatrix} f_x(x, y) & f_y(x, y) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right]^2 \]

\[ = (\Delta x \ \Delta y) \sum_{(x,y)\in\text{window}} \begin{pmatrix} f_x^2(x, y) & f_x(x, y) f_y(x, y) \\ f_x(x, y) f_y(x, y) & f_y^2(x, y) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \]

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

- Iso-sensitivity curves are ellipses since \( v^T M v = cte \)
Feature point extraction

$$SSD \approx \Delta^\top M \Delta$$

Find points for which the following is large

$$\min \Delta^\top M \Delta \text{ for } \|\Delta\| = 1$$

i.e. maximize eigenvalues of $M$
Keypoint detection

Often based on eigenvalues $\lambda_1$, $\lambda_2$ of “structure matrix” (aka ”normal matrix” aka “second-moment matrix”)

$$M = \begin{bmatrix}
\sum_{(x,y)\in\text{window}} f_x^2(x,y) & \sum_{(x,y)\in\text{window}} f_x(x,y)f_y(x,y) \\
\sum_{(x,y)\in\text{window}} f_x(x,y)f_y(x,y) & \sum_{(x,y)\in\text{window}} f_y^2(x,y)
\end{bmatrix}$$

$f_x(x,y)$ – horizontal image gradient
$f_y(x,y)$ – vertical image gradient

Measure of “cornerness”

$$C(x,y) = \det(M) - k \cdot (\text{trace}(M))^2$$

$$= \lambda_1\lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2$$

[Harris, Stephens, 1988]
Contour plot of Harris cornerness

\[ C(x, y) = \det(M) - k \cdot (\text{trace}(M))^2 \]
\[ = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 \]
Keypoint Detection: Input
Harris cornerness
Thresholded cornerness
Local maxima of cornerness
Superimposed keypoints
Better localization of corners

• Give more importance to central pixels by using Gaussian weighting function

\[
M = \sum_{(x,y) \in \text{window}} G(x-x_o, y-y_o, \sigma) \begin{bmatrix} f_x^2(x,y) & f_x(x,y)f_y(x,y) \\ f_x(x,y)f_y(x,y) & f_y^2(x,y) \end{bmatrix}
\]

\( \sigma = 0.7 \)

• Compute subpixel localization by fitting parabola to \textit{cornerness} function
Robustness of Harris corner detector

- Invariant to brightness offset: $f(x,y) \rightarrow f(x,y) + c$
- Invariant to shift and rotation
- Not invariant to scaling

[Schmid, 2000]
Lowe’s SIFT features

Recover features with position, orientation and scale

(Lowe, ICCV99)
Position

- Look for strong responses of DoG filter (Difference-Of-Gaussian)
- Only consider local maxima

\[ \text{DOG}(x, y) = \frac{1}{k} e^{-\frac{x^2+y^2}{(k\sigma)^2}} - e^{-\frac{x^2+y^2}{\sigma^2}} \]

\[ k = \sqrt{2} \]
Scale

- Look for strong responses of DoG filter (Difference-of-Gaussian) over scale space
- Only consider local maxima in both position and scale
- Fit quadratic around maxima for subpixel accuracy
Orientation

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)
Minimum contrast and “cornerness”

Figure 5: This figure shows the stages of keypoint selection. (a) The 233×189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principle curvatures.
SIFT descriptor

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions
Matas et al.’s maximally stable regions

- Look for extremal regions

Next week:
Fourier transform

Video