Lecture 5
Image Features

Christopher Brunner
AMME4710 – Computer Vision and Image Processing
Semester 2, 2013

By the end of this lecture you will be able to:

• Recognize the importance of extracting features from images in computer vision.
• Identify the main kinds of image features and their properties.
• Understand the intuition behind the main approaches to detecting features
Resources

• Readings
  – Szeliski, Computer Vision, Ch. 4
    http://szeliski.org/Book/

• Additional material (optional)

• Slides credits
  – A. Torralba, S. Seitz, T. Darrell, K. Grauman, R. Fergus
1.4 Fundamental Steps in Digital Image Processing

Outputs of these processes generally are images

CHAPTER 6
Color image processing

CHAPTER 7
Wavelets and multiresolution processing

CHAPTER 8
Compression

CHAPTER 9
Morphological processing

CHAPTER 5
Image restoration

CHAPTER 10
Segmentation

CHAPTER 3 & 4
Image filtering and enhancement

CHAPTER 11
Representation & description

CHAPTER 2
Image acquisition

CHAPTER 12
Object recognition

FIGURE 1.23
Fundamental steps in digital image processing. The chapter(s) indicated in the boxes is where the material described in the box is discussed. (in the book)
Automatic mosaicing

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html
Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]
Recognition of specific objects, scenes

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

Kristen Grauman
Very hard case

1985, age 12

2002, age 30

“How the Afghan Girl was Identified by Her Iris Patterns” (J. Daugman, 2002)
Harder still?

NASA Mars Rover images
Answer below (look for tiny colored squares…)

NASA Mars Rover images with SIFT feature matches
Figure by Noah Snavely
More motivation...

• Feature points are used also for:
  – Image alignment (e.g., mosaics)
  – 3D reconstruction
  – Motion tracking
  – Object recognition
  – Indexing and database retrieval
  – Robot navigation
  – Other ...
Outline

• Introduction
• Corners (Harris detector)
• Edges (Canny edge detector)
• Lines (RANSAC, Hough transform)
• Invariant Descriptors (SIFT)
• Matching (ICP)
INTRODUCTION
How do we build a panorama?

• We need to match (align) images
• Global methods sensitive to occlusion, lighting, parallax effects. So look for local features that match well.
• How would you do it by eye?
Matching with Features

• Detect feature points in both images

• Find corresponding pairs
Matching with Features

• Detect feature points in both images
• Find corresponding pairs
• Use these pairs to align images
Matching with Features

• Problem 1:
  – Detect the *same* point *independently* in both images

We need a repeatable detector
Matching with Features

• Problem 2:
  – For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor
Figure 4.3: *Image pairs with extracted patches below. Notice how some patches can be localized or matched with higher accuracy than others.*
Selecting Good Features

• What’s a “good feature”?  
  – Satisfies brightness constancy—looks the same in both images  
  – Has sufficient texture variation  
  – Does not have too much texture variation  
  – Corresponds to a “real” surface patch—see below:

  – Does not deform too much over time
Advantages of local features

Locality
  – features are local, so robust to occlusion and clutter

Distinctiveness:
  – can differentiate a large database of objects

Quantity
  – hundreds or thousands in a single image

Efficiency
  – real-time performance achievable

Generality
  – exploit different types of features in different situations
Models of Image Change

• Geometry
  – Rotation
  – Similarity (rotation + uniform scale)
  – Affine (scale dependent on direction)
    valid for: orthographic camera, locally planar object

• Photometry
  – Affine intensity change ($l \rightarrow a \cdot l + b$)
Invariant local features

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, ...

Feature Descriptors
CORNERS
HARRIS DETECTOR
Finding Corners

- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive


Source: Lana Lazebnik
Corners as distinctive interest points

- We should easily recognize the point by looking through a small window.
- Shifting a window in any direction should give a large change in intensity.

“flat” region: no change in all directions
“edge”: no change along the edge direction
“corner”: significant change in all directions

Source: A. Efros
Harris Detector formulation

Change of intensity for the shift \([u,v]\):

\[ E(u,v) = \sum_{x,y} w(x,y) \left[ I(x+u, y+v) - I(x, y) \right]^2 \]

Window function  
Shifted intensity  
Intensity

Window function \(w(x,y)\) = 
1 in window, 0 outside  
Gaussian

Source: R. Szeliski
Harris Detector formulation

Expanding $I(x,y)$ in a Taylor series expansion, we have, for small shifts $[u,v]$, a bilinear approximation:

$$E(u, v) \approx [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

where $M$ is a $2 \times 2$ matrix computed from image derivatives:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Sum over image region – area we are checking for corner

Gradient with respect to $x$, times gradient with respect to $y$

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$
Harris Detector formulation

where $M$ is a $2 \times 2$ matrix computed from image derivatives:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I^2_x & I_x I_y \\ I_x I_y & I^2_y \end{bmatrix}$$

Sum over image region – area we are checking for corner

Gradient with respect to $x$, times gradient with respect to $y$

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$
What does this matrix reveal?

First, consider an axis-aligned corner:
What does this matrix reveal?

First, consider an axis-aligned corner:

\[
M = \left[ \begin{array}{cc}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2 \\
\end{array} \right] = \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2 \\
\end{bmatrix}
\]

This means dominant gradient directions align with x or y axis.

If either \( \lambda \) is close to 0, then this is not a corner, so look for locations where both are large.

What if we have a corner that is not aligned with the image axes?

Slide credit: David Jacobs
General Case

Since $M$ is symmetric, we have

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

We can visualize $M$ as an ellipse with axis lengths determined by the eigenvalues and orientation determined by $R$.
Classification of image points using eigenvalues of $M$:

- $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions.
- $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; $E$ increases in all directions.
- $\lambda_1 \gg \lambda_2$
Harris Detector: Mathematics

Measure of corner response:

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

This expression does not require computing the eigenvalues.

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

\((k - \text{empirical constant, } k = 0.04-0.06)\)
Harris Detector: Mathematics

- $R$ depends only on eigenvalues of $M$
- $R$ is large for a corner
- $R$ is negative with large magnitude for an edge
- $|R|$ is small for a flat region
Harris corner detector

• Algorithm steps
  1. Compute Gaussian derivatives at each pixel
  2. Compute second moment matrix $M$ in a Gaussian window around each pixel
  3. Compute corner response function $R$
  4. Find points with large corner response $(R > \text{threshold})$
  5. Take the points of local maxima of $R$
Harris Detector: Workflow

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.
Harris Detector: Workflow

Compute corner response $R$
Harris Detector: Workflow

Find points with large corner response: $R > \text{threshold}$
Harris Detector: Workflow

Take only the points of local maxima of $R$
Harris Detector: Workflow
Harris Detector: Properties

• Rotation invariance

Ellipse rotates but its shape (i.e. eigenvalues) remains the same

*Corner response R is invariant to image rotation*
Harris Detector: Properties

- Partial invariance to additive and multiplicative intensity changes
  - Only derivatives are used => invariance to intensity shift $l \rightarrow l + b$
  - Intensity scale: $l \rightarrow a \cdot l$ Because of fixed intensity threshold on local maxima, only partial invariance to multiplicative intensity changes.
Harris Detector: Properties

• Not invariant to image scale

All points will be classified as edges  

Corner!
EDGES
CANNY EDGE DETECTOR
What is an edge?
What is an edge?

Depth discontinuity
Material change
Texture boundary
Paint
What is an edge?

Kanizsa triangle
Edge detection

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels – can compress a lot of visual information by using just the edges.

- **Ideal:** artist’s line drawing (but artist is also using object-level knowledge)

Source: D. Lowe
Gradients and edges (Forsyth, ch. 8)

- Points of sharp change in an image are interesting:
  - change in reflectance
  - change in object
  - change in illumination
  - noise

- Sometimes called edge points

- General strategy
  - determine image gradient
  - now mark points where gradient magnitude is particularly large wrt neighbours (ideally, curves of such points).

Forsyth, 2002
Image gradient

- The gradient of an image:
  \[ \nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \]

- \[ \nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right] \]
  \[ \nabla f = \left[ 0, \frac{\partial f}{\partial y} \right] \]

The gradient points in the direction of most rapid increase in intensity
- How does this direction relate to the direction of the edge?

The gradient direction is given by
\[ \theta = \tan^{-1} \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \]

The *edge strength* is given by the gradient magnitude
\[ ||\nabla f|| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \]

Source: Steve Seitz
Designing an edge detector

- Criteria for an “optimal” edge detector:
  - **Good detection**: the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges).
  - **Good localization**: the edges detected must be as close as possible to the true edges.
  - **Single response**: the detector must return one point only for each true edge point; that is, minimize the number of local maxima around the true edge.

Source: L. Fei-Fei
Canny edge detector

• This is probably the most widely used edge detector in computer vision.
• Theoretical model: step-edges corrupted by additive Gaussian noise.
• Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of signal-to-noise ratio and localization.
• MATLAB: edge(image, ‘canny’)


Source: L. Fei-Fei
Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin multi-pixel wide “ridges” down to single pixel width

Source: D. Lowe, L. Fei-Fei
Non-maximum suppression

At q, we have a maximum if the value is larger than those at both p and at r. (Interpolate to get these values.)

Source: D. Forsyth
Example

• original image (Lena, 1972)
Example

Norm (magnitude) of the gradient
Example

thresholding
Example

Non-maximum suppression
Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
   – Thin multi-pixel wide “ridges” down to single pixel width
4. Linking of edge points

Source: D. Lowe, L. Fei-Fei
Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).

Source: D. Forsyth
Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
   - Thin multi-pixel wide “ridges” down to single pixel width
4. Linking of edge points
   - Hysteresis thresholding: use a higher threshold to start edge curves and a lower threshold to continue them

Source: D. Lowe, L. Fei-Fei
Hysteresis thresholding

- Use a high threshold to start edge curves and a low threshold to continue them
  - Reduces *drop-outs*

Source: S. Seitz
Hysteresis thresholding

original image

high threshold
(strong edges)

low threshold
(weak edges)

hysteresis threshold

Source: L. Fei-Fei
Effect of $\sigma$ (Gaussian kernel spread/size)

The choice of $\sigma$ depends on desired behavior

- large $\sigma$ detects large scale edges
- small $\sigma$ detects fine features

Source: S. Seitz
Auto-correlation surfaces