Features and Matching

CSE P576

Vitaly Ablavsky
Correspondence Problem

- A basic problem in Computer Vision is to establish matches (correspondences) between images
- This has many applications: rigid/non-rigid tracking, object recognition, image registration, structure from motion, stereo...
Feature Detectors

A variety of feature detectors and descriptors can be used to analyze, describe and match images: (a) point-like interest operators (Brown, Szeliski, and Winder 2005) © 2005 IEEE; (b) region-like interest operators (Matas, Chum, Urban et al. 2004) © 2004 Elsevier; (c) edges (Elder and Goldberg 2001) © 2001 IEEE; (d) straight lines (Sinha, Steedly, Szeliski et al. 2008) © 2008 ACM.

Corners/Blobs

Regions

Edges

Straight Lines
Feature Descriptors

Image Patch

Shape Context

SIFT

Learned Descriptors

A: Feature network
B: Metric network
C: MatchNet in training
Features and Matching

- Feature detectors
  - Canny edges, Harris corners, DoG, MSERs
- Feature descriptors
  - Image patches, invariance, SIFT, learned features
Edge Detection

• One of the first algorithms in Computer Vision
Edge Detection

- Consider edge detection for a 1D signal $I(x)$

![Signal graph](image1)

![Differentiated signal graph](image2)

- Naive approach: look for maxima/minima in $I'(x)$

What’s the problem?
Edge Detection

- Solution: start by smoothing the image to remove noise

\[ I(x) = \text{image} \]
\[ k(x) = \text{kernel} \]
\[ s(x) = I(x) \ast k(x) \]
\[ s'(x) = \text{smoothed derivative} \]

Edges are found by thresholding the smoothed derivative
2D Edge Detection

- Smooth image and convolve with $[-1 \ 1]$.

2D gradient: $\nabla I = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$
2D Edge Detection

- Look at the magnitude of the smoothed gradient \(|\nabla I|\)

\[
|\nabla I| = \sqrt{g_x^2 + g_y^2}
\]

- Non-maximal suppression (keep only points where \(|\nabla I|\)
is a maximum in directions \(\pm \nabla I\))

[ Canny 1986 ]
The next step is to find the gradient of the smoothed image $S(x, y)$ at every pixel:

$$\nabla S = \nabla \left( G_\sigma \ast I \right) = \begin{bmatrix} \frac{\partial (G_\sigma \ast I)}{\partial x} & \frac{\partial (G_\sigma \ast I)}{\partial y} \end{bmatrix}$$

The following example shows $|\nabla S|$ for a fruity image:

(a) Original image (b) Edge strength
2D Edge Detection

- Threshold the gradient magnitude with two thresholds: $T_{\text{high}}$ and $T_{\text{low}}$
- Edges start at edge locations with gradient magnitude $> T_{\text{high}}$
- Continue tracing edge until gradient magnitude falls below $T_{\text{low}}$

[ Canny 1986 ]
Edges + Segmentation

- Segmentation is subjective [Martin, Fowlkes, Tal, Malik 2001]
Image Structure

• What kind of structures are present in the image locally?

0D Structure: not useful for matching

1D Structure: edge, can be localised in one direction, subject to the “aperture problem”

2D Structure: corner, or interest point, can be localised in both directions, good for matching

Edge detectors find contours (1D structure), Corner or Interest point detectors find points with 2D structure.
Local SSD Function

- Consider the sum squared difference (SSD) of a patch with its local neighbourhood

\[
\text{SSD} = \sum_{\mathcal{R}} |I(x) - I(x + \Delta x)|^2
\]
Local SSD Function

- Consider the local SSD function for different patches

High similarity locally

High similarity along the edge

Clear peak in similarity function
Harris Corners

- Harris corners are peaks of a local similarity function
Harris Corners

- We will use a first order approximation to the local SSD function

\[ \text{SSD} = \sum_{\mathcal{R}} |I(x) - I(x + \Delta x)|^2 \]
Without loss of generality, we will assume a grayscale 2-dimensional image is used. Let this image be given by \( I \). Consider taking an image patch \( (x, y) \in W \) (window) and shifting it by \((\Delta x, \Delta y)\). The sum of squared differences (SSD) between these two patches, denoted \( f \), is given by:

\[
f(\Delta x, \Delta y) = \sum_{(x_k, y_k) \in W} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2
\]

\(I(x + \Delta x, y + \Delta y)\) can be approximated by a Taylor expansion. Let \( I_x \) and \( I_y \) be the partial derivatives of \( I \), such that

\[
I(x + \Delta x, y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y
\]

This produces the approximation

\[
f(\Delta x, \Delta y) \approx \sum_{(x, y) \in W} (I_x(x, y)\Delta x + I_y(x, y)\Delta y)^2,
\]

which can be written in matrix form:

\[
f(\Delta x, \Delta y) \approx \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix},
\]

where \( M \) is the structure tensor,

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

computations of \( I_x^2, I_x I_y, \) etc. are per-pixel

For \( x \ll y \), one has \( \frac{x^2}{x+y} = x \cdot \frac{1}{1+x/y} \approx x \). In this step, we compute the smallest eigenvalue of the structure tensor using that approximation:

\[
\lambda_{\text{min}} \approx \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\det(M)}{\text{tr}(M)}
\]

with the trace \( \text{tr}(M) = m_{11} + m_{22} \).

Another commonly used Harris response calculation is shown as below,

\[
R = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 = \det(M) - k \cdot \text{tr}(M)^2
\]

where \( k \) is an empirically determined constant; \( k \in [0.04, 0.06] \).

Credit: https://en.wikipedia.org/wiki/Harris_corner_detector
Harris Corners

- Corners matched using correlation

99 inliers

89 outliers

Difference of Gaussian

- DoG = centre-surround filter

- Find local-maxima of the centre surround response

Non-maximal suppression:
These points are maxima in a 10 pixel radius
Difference of Gaussian

- DoG detects blobs at scale that depends on the Gaussian standard deviation(s)

Note: DOG \approx \text{Laplacian of Gaussian}

\[
\text{red} = [1 \ -2 \ 1] * g(x; 5.0)
\]
\[
\text{black} = g(x; 5.0) - g(x; 4.0)
\]
Detection Scale

- Smoothing standard deviations determine scale of detected features, e.g., edge detection in cloth

\[ \sigma = 1 \quad \sigma = 5 \]

- Many algorithms use multi-scale architectures to get around this problem
- e.g., Scale-Invariant Feature Transform “SIFT”
MSERS

- Maximally Stable Extremal Regions

- Find regions of high contrast using a watershed approach

MSERS are stable (small change) over a large range of thresholds

[ Matas et al 2002 ]
• Try the **Interest Point Extractor** section in Project 1
• `corner_function` : Devise a corner strength function
• `find_local_maxima` : Find interest points as maxima of the corner strength function
Corner Matching

- A simple approach to correspondence is to match corners between images using normalised correlation or SSD
Breaking Correlation

- Correlation/SSD works well when the images are quite similar (e.g., tracking in frames of a video)
- However, it is easily broken by simple image transforms, e.g.,

![Original](image1) ![Rotation](image2) ![Scale](image3)

- These transformations are very common in imaging, so we would like feature matching to be **invariant** to them
Local Coordinate Frame

- One way to achieve invariance is to use local coordinate frames that follow the surface transformation.
Detecting Scale/Orientation

- A common approach is to detect a local scale and orientation for each feature point

\[
\begin{align*}
H &= \begin{bmatrix}
    D_{xx} & D_{xy} \\
    D_{xy} & D_{yy}
\end{bmatrix}, \\
\text{Tr}(H) &> 10 \\
\text{Det}(H) &> 10
\end{align*}
\]

e.g., extract Harris at multiple scales and align to the local gradient
Detecting Scale/Orientation

- Patch matching can be improved by using scale/orientation and brightness normalisation.

Sampling at a coarser scale than detection further improves robustness.
Panorama Alignment
Wide Baseline Matching

- Patch-based matching works well for short baselines, but fails for large changes in scale, rotation or 3D viewpoint.

What factors cause differences between these images?
Wide Baseline Matching

- We would like to match patches despite these changes

What features of the local patch are invariant?
Scale Invariant Feature Transform

- A detector and descriptor designed for object recognition

- SIFT features are invariant to translation, rotation and scale and slowly varying under perspective and 3D distortion
- Variants widely used in object recognition, image search etc.

[ Lowe 1999 ]
Scale Invariant Feature Transform

- Scale invariant detection and local orientation estimation
- **Edge based** representation that is robust to local shifting of edges (parallax and/or stretch)

[ vlfeat.org ]
SIFT Detection

- Convolve with centre-surround Laplacian/DoG filter

- Find all maxima at all scales in a Laplacian Pyramid
Scale Selection

- A DOG (Laplacian) Pyramid is formed with multiple scales per octave

Detections are local maxima in a 3x3x3 scale-space window
Scale Selection

Figure 1.4. Schematic three-dimensional illustration of the scale-space representation of a one-dimensional signal.
Scale Selection

- Maximising the DOG function in scale as well as space performs scale selection
Orientation Selection

- To select a local orientation, build a histogram over orientation

Selected orientation is peak in this histogram
SIFT Descriptor

- We selected a scale and orientation at each detection,
- Now need descriptor to represent the local region in a way robust to parallax, illumination change etc.
Simple + Complex Cells in V1

- Neuroscientists have investigated the response of cells in the primary visual cortex

- “Complex Cells” in V1 respond over a range of positions but are highly sensitive to orientation

[ Hubel and Wiesel ]
SIFT Descriptor

- Describe local region by distribution (over angle) of gradients

Each descriptor: $4 \times 4$ grid $\times$ 8 orientations $= 128$ dimensions
SIFT Recap

- **Detector**: find points that are maxima in a DOG pyramid
- Compute local orientation from gradient histogram
- This establishes a local coordinate frame with scale/orientation
- **Descriptor**: Build histograms over gradient orientations (8 orientations, 4x4 grid)
- Normalise the final descriptor
SIFT Matching

- Extract SIFT features from an image

Each image might generate 100’s or 1000’s of SIFT descriptors
SIFT Matching

• Goal: Find all correspondences between a pair of images

• Extract and match all SIFT descriptors from both images
SIFT Matching

• Each SIFT feature is represented by 128 numbers
• Feature matching becomes task of finding a nearby 128-d vector
• Nearest-neighbour matching:

$$NN(j) = \arg \min_i |x_i - x_j|, \ i \neq j$$

• Linear time, but good approximation algorithms exist
• e.g., Best Bin First K-d Tree [Beis Lowe 1997], FLANN (Fast Library for Approximate Nearest Neighbours) [Muja Lowe 2009]
SIFT Matching

- Feature matching returns a set of noisy correspondences
- To get further, we will have to know something about the **geometry** of the images
Shape Context

- Useful for matching with contours

Descriptor is log polar histogram

[ Belongie Malik 2000 ]
Choosing Features

- The best choice of features is usually application dependent

Shape context?  SIFT?  Something else?
Learning Descriptors

- Descriptor design as a learning (embedding) problem

[Winder Brown 2007]
Learning Descriptors

- Deep networks for descriptor learning

Patch labels

Image labels, also learns interest function

[A: Feature network
  Bottleneck
  Pool4
  Conv4
  Conv3
  Conv2
  Pool1
  Conv1
  Pool0
  Conv0
  Preprocessing]

[B: Metric network
  FC3 + Softmax
  FC2
  FC1]

[C: MatchNet in training
  Cross-Entropy Loss
  Metric network]

[MatchNet
Han et al 2015 ]
[DELF
Noh et al 2017]
Project 1

• You can now complete Project 1 — Descriptors and Matching and Testing and Improving Feature Matching sections.
Next Lecture

- Planar Geometry, Camera Models, RANSAC