2D Image Processing
Feature Descriptors

Prof. Didier Stricker

Kaiserlautern University

http://ags.cs.uni-kl.de/

DFKI – Deutsches Forschungszentrum für Künstliche Intelligenz

http://av.dfki.de
Overview

- Previous lectures:
  - Feature extraction
    - Gradient/edge
    - Points (Kanade-Tomasi + Harris)
    - Blob (incl. scale)

- Today:
  - Feature Descriptors

- Why?
  - Point matching
  - Region and image matching
  - Goals: object recognition, tracking, …
Matching with Features

Problem:
- For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor!
Feature descriptors

- We know how to detect good points
- Next question: How to match them?

Answer: Come up with a descriptor for each point, find similar descriptors between the two images
Feature descriptors

- We know how to detect good points
- Next question: How to match them?

- Lots of possibilities (this is a popular research area)
  - Simple option: match square windows around the point
  - State of the art approach: SIFT
Invariance vs. discriminability

**Invariance:**
- Descriptor shouldn’t change even if image is transformed

**Discriminability:**
- Descriptor should be highly unique for each point
Image transformations

- Geometric
  - Rotation
  - Scale
  - Affine…

- Photometric
  - Intensity change
Invariance

- Most feature descriptors are designed to be invariant to
  - Translation
  - 2D rotation
  - Scale

- They can usually also handle
  - Limited 3D rotations (SIFT works up to about 60 degrees)
  - Limited affine transformations (some are fully affine invariant)
  - Limited illumination/contrast changes
How to achieve invariance

Need both of the following:

1. Make sure your detector is invariant
2. Design an invariant feature descriptor

- Simplest descriptor: a single pixel value
  - What’s this invariant to?
  - Is this unique?

- Next simplest descriptor: a square window of pixels
  - Is this unique?
  - What’s this invariant to?

- Let’s look at some better approaches…
Comparing two patches

- L1 – Sum of Absolute Differences (SAD)
- L2 – Sum of Squared Differences (SSD)
- Cross-Correlation
Common window-based approaches

- **Sum of Absolute Differences (SAD) – L1 norm**
  \[
  \sum_{(i,j) \in W} |I_1(i,j) - I_2(x + i, y + j)|
  \]

- **Sum of Squared Differences (SSD) – L2 norm**
  \[
  \sum_{(i,j) \in W} (I_1(i,j) - I_2(x + i, y + j))^2
  \]
Common window-based approaches

- Normalized Cross-Correlation (NCC)

\[
\frac{\sum_{(i,j) \in W} I_1(i,j) \cdot I_2(x + i, y + j)}{\sqrt{\sum_{(i,j) \in W} I_1^2(i,j) \cdot \sum_{(i,j) \in W} I_2^2(x + i, y + j)}}
\]

- Zero-Mean Normalized Cross-Correlation (ZNCC)

\[
\frac{\sum_{(i,j) \in W} (I_1(i,j) - \overline{I_1}(i,j)) \cdot (I_2(x + i, y + j) - \overline{I_2}(x + i, y + j))}{\sqrt{\sum_{(i,j) \in W} (I_1(i,j) - \overline{I_1}(i,j))^2 \cdot \sum_{(i,j) \in W} (I_2(x + i, y + j) - \overline{I_2}(x + i, y + j))^2}}
\]
Recall NCC

- Write regions as vectors:
  - A → \( \mathbf{a} \)
  - B → \( \mathbf{b} \)

\[
\text{NCC} = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|}
\]

\(-1 \leq \text{NCC} \leq 1\)
Cross-Correlation

\[ CC(P_1, P_2) = \frac{1}{N} \sum_{i=1}^{N} P_1[i] P_2[i]. \]

- **Output in range:**
  \[ +1 \rightarrow -1 \]
- **Not invariant to changes in** a, b

**Affine photometric transformation:**

\[ I \rightarrow aI + b \]

- Original Patch and Intensity Values
- Brightness Decreased, \( CC = 0.99998895629 \)
- Contrast increased, \( CC = 0.969868160814 \)
(Zero Mean) Normalized Cross-Correlation

Make each patch zero mean:

\[ \mu = \frac{1}{N} \sum_{x,y} I(x,y) \]
\[ Z(x, y) = I(x, y) - \mu \]

Then make unit variance:

\[ \sigma^2 = \frac{1}{N} \sum_{x,y} Z(x, y)^2 \]
\[ ZN(x, y) = \frac{Z(x, y)}{\sigma} \]

Affine photometric transformation:

\[ I \rightarrow aI + b \]

Original Patch and Intensity Values

Brightness Decreased, CC = 0.99998895629

Contrast increased, CC = 0.969868160814
Example with NCC
Example with NCC

left image band

right image band

cross correlation
Example with NCC

Why is it not as good as before?
SIFT – Scale Invariant Feature Transform
Orientation assignment: Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

David Lowe IJCV 2004

This figure shows just 8x8 of the 16x16 squares

Adapted from slide by David Lowe
Orientation assignment

- Create histogram of gradient directions, within a region around the keypoint, at selected scale:

\[ L(x,y,\sigma) = G(x,y,\sigma) \ast I(x,y) \]

\[ m(x, y) = \sqrt{\left( (L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2 \right)^{\frac{1}{2}}} \]

\[ \theta(x, y) = \arctan \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \]

- Histogram entries are weighted by (i) gradient magnitude and (ii) a Gaussian function with \( \sigma \) equal to 1.5 times the scale of the keypoint.
Orientation assignment

\( D_x \)

\( D_y \)

\( M \)

\( \Theta \)
Orientation assignment

- Keypoint location = extrema location
- Keypoint scale is scale of the DOG image
Orientation assignment

- Gaussian image (at closest scale, from pyramid)
- Gradient magnitude
- Gradient orientation
Orientation assignment

\[ \sigma = 1.5 \times \text{scale of the keypoint} \]
Orientation assignment

weighted gradient magnitude

weighted orientation histogram. Each bucket contains sum of weighted gradient magnitudes corresponding to angles that fall within that bucket.

36 buckets
10 degree range of angles in each bucket, i.e.

0 \leq \text{ang} < 10 \ : \ \text{bucket 1}
10 \leq \text{ang} < 20 \ : \ \text{bucket 2}
20 \leq \text{ang} < 30 \ : \ \text{bucket 3} \ldots
Orientation assignment

weighted gradient magnitude

weighted orientation histogram.

gradient orientation

Orientation of keypoint is approximately 25 degrees

Note: accurate peak position is determined by fitting
Orientation assignment

- There may be multiple orientations:

  - In this case, generate duplicate keypoints, one with orientation at 25 degrees, one at 155 degrees.

  - Design decision: you may want to limit the number of peaks to two.
Keypoint Descriptor
Keypoint Descriptor (cont’d)

1. Take a 16x16 window around detected interest point.

2. Divide into a 4x4 grid of cells.

3. Compute histogram in each cell.

16 histograms x 8 orientations = 128 features
Keypoint Descriptor (cont’d)

Each histogram entry is weighted by
(i) gradient magnitude and
(ii) a Gaussian function with $\sigma$ equal to 0.5 times the width of the descriptor window.
Keypoint Descriptor (cont’d)

Partial Voting: distribute histogram entries into adjacent bins (i.e., additional robustness to shifts)

- Each entry is added to all bins, multiplied by a weight of 1-d, where d is the distance from the bin it belongs to.
Keypoint Descriptor (cont’d)

- Descriptor depends on two main parameters:
  1. number of orientations $r$
  2. $n \times n$ array of orientation histograms

\[ r n^2 \text{ features} \]

SIFT: $r=8$, $n=4$
128 features
Keypoint Descriptor (cont’d)

Invariance to linear illumination changes:
- Normalization to **unit length** is sufficient.

![Diagram showing image gradients and keypoint descriptor with 128 features]
Keypoint Descriptor (cont’d)

Non-linear illumination changes:
- Saturation affects gradient magnitudes more than orientations
- Threshold entries to be no larger than 0.2 and renormalize to unit length

128 features
Robustness to viewpoint changes

- Match features after random change in image scale and orientation, with 2% image noise, and affine distortion.
- Find nearest neighbor in database of 30,000 features.

Additional robustness can be achieved using affine invariant region detectors.
Distinctiveness

- Vary size of database of features, with 30 degree affine change, 2% image noise.
- Measure % correct for single nearest neighbor match.
Matching SIFT features

Given a feature in $I_1$, how to find the best match in $I_2$?

1. Define distance function that compares two descriptors.
2. Test all the features in $I_2$, find the one with minimal distance.
Matching SIFT features (cont’d)

- What distance function should we use?
  - Use $\text{SDD}(f_1, f_2) = \sum_{i=0}^{N-1} (f_{1i} - f_{2i})^2$ (i.e., sum of squared differences)
  - Can give good scores to very ambiguous (bad) matches
Matching SIFT features (cont’d)

- A better distance measure is the following:
  - $\text{SSD}(f_1, f_2) / \text{SSD}(f_1, f_2')$
    - $f_2$ is best SSD match to $f_1$ in $I_2$
    - $f_2'$ is 2nd best SSD match to $f_1$ in $I_2$
Matching SIFT features (cont’d)

- Accept a match if $\frac{\text{SSD}(f_1, f_2)}{\text{SSD}(f_1, f_2')} < t$
- $t=0.8$ has given good results in object recognition.
  - 90% of false matches were eliminated.
  - Less than 5% of correct matches were discarded
SIFT Steps – Review

(1) Scale-space extrema detection
   - Extract scale and rotation invariant interest points (i.e., keypoints).

(2) Keypoint localization
   - Determine location and scale for each interest point.
   - Eliminate “weak” keypoints

(3) Orientation assignment
   - Assign one or more orientations to each keypoint.

(4) Keypoint descriptor
   - Use local image gradients at the selected scale.


Cited > 10,000 times
SIFT flow
Maxima in D
Remove low contrast
Remove edges
SIFT descriptor

[Image of Albert Einstein with SIFT keypoints]
SIFT – Scale Invariant Feature Transform

- Empirically found to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and robust to moderate *affine* transformations

Scale = 2.5
Rotation = 45°

---

1 D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. Accepted to IJCV 2004
Summary of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
HOG – Histogram of Oriented Gradients
2d Global Detector
Dalal and Triggs, CVPR 2005

- 3-D Histogram of Oriented Gradients (HOG) as descriptors
- Linear SVM for runtime efficiency
- Tolerates different poses, clothing, lighting and background
- Currently works for fully visible upright persons

Slide from Sminchisescu
Window-based models: Building an object model

Simple holistic descriptions of image content
- grayscale / color histogram
- vector of pixel intensities
Window-based models: Building an object model

- Pixel-based representations sensitive to small shifts

- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation
Window-based models: Building an object model

- Consider edges, contours, and (oriented) intensity gradients
Window-based models: Building an object model

- Consider edges, contours, and (oriented) intensity gradients

- Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Contrast-normalization: try to correct for variable illumination
Histograms

Gradient histograms measure the orientations and strengths of image gradients within an image region

- Global descriptor for the complete body
- Very high-dimensional
  - Typically ~4000 dimensions
HOG: use in detection, tracking…

Very promising results on challenging data

Phases
1. Learning Phase
2. Detection Phase
Descriptor

1. Compute gradients on an image region of 64x128 pixels

2. Compute histograms on ‘cells’ of typically 8x8 pixels (i.e. 8x16 cells)

3. Normalize histograms within overlapping blocks of cells (typically 2x2 cells, i.e. 7x15 blocks)

4. Concatenate histograms
Gradients

- Convolution with [-1 0 1] filters
- No smoothing
- Compute gradient magnitude + direction
- Per pixel: color channel with greatest magnitude \(\rightarrow\) final gradient
Cell histograms

- 9 bins for gradient orientations (0-180 degrees)
- Filled with magnitudes
- Interpolated trilinearly:
  - Bilinearly into spatial cells
  - Linearly into orientation bins
Linear and Bilinear interpolation for subsampling

Linear:

\[ y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0} \]

Bilinear:

\[
f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad \text{where} \quad R_1 = (x, y_1),
\]

\[
f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad \text{where} \quad R_2 = (x, y_2).
\]

\[
f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).
\]
Histogram interpolation example

θ = 85 degrees
Distance to bin centers
- Bin 70 → 15 degrees
- Bin 90 → 5 degrees
Ratios: 5/20 = 1/4, 15/20 = 3/4

Distance to cell centers
- Left: 2, Right: 6
- Top: 2, Bottom: 6
Ratio Left-Right: 6/8, 2/8
Ratio Top-Bottom: 6/8, 2/8
Ratios:
- 6/8 * 6/8 = 36/64 = 9/16
- 6/8 * 2/8 = 12/64 = 3/16
- 2/8 * 6/8 = 12/64 = 3/16
- 2/8 * 2/8 = 4/64 = 1/16
Blocks

- Overlapping blocks of 2x2 cells

- Cell histograms are concatenated and then normalized
  - Note that each cell has several occurrences with different normalization in final descriptor

- Normalization
  - Different norms possible
  - We add a normalization epsilon to avoid division by zero
Blocks

- Gradient magnitudes are weighted according to a Gaussian spatial window

- Distant gradients contribute less to the histogram
Final Descriptor

- Concatenation of blocks:

- Visualization:

![Images of people and corresponding feature maps]
Linear SVM for pedestrian detection using the HOG descriptor
Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG’s over detection window → Linear SVM → Person/ non-person classification
Histogram of gradient orientations

- Orientation

Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG’s over detection window → Linear SVM → Person/ non-person classification

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
Engineering

- Developing a feature descriptor requires a lot of engineering
  - Testing of parameters (e.g. size of cells, blocks, number of cells in a block, size of overlap)
  - Normalization schemes (e.g. L1, L2-Norms etc., gamma correction, pixel intensity normalization)

- An extensive evaluation of different choices was performed, when the descriptor was proposed

- It’s not only the idea, but also the engineering effort
Thank you!