Descriptors
Motivation::
Scene Classification

How to differentiate between scenes?
Naïve-st approach::
Primitive Global descriptors

Color expectancy:  \[ E_i = \frac{1}{N} \sum_{j=1}^{N} P_{ij} \]

Color variance:  \[ \delta_i = \left( \frac{1}{N} \sum_{j=1}^{N} (P_{ij} - E_i)^2 \right)^{\frac{1}{2}} \]

Skewness:  \[ \sigma_i = \left( \frac{1}{N} \sum_{j=1}^{N} (P_{ij} - E_i)^3 \right)^{\frac{1}{3}} \]

Very Simple  Very-very limited
Less but still naive::
Color Histogram

High invariance to many transformation

Limited discriminative power
We understand by now that global descriptors are very limited...

In many applications we need to describe images at a local scale based on local features
Local Descriptors

Motivation::Detection

Would a global descriptor help us find the book (left image) in the right image?
Detection::
General Approach
Detection::
General Approach

1. Find a set of distinctive key-points
Detection::
General Approach

1. Find a set of distinctive key-points
Detection::

General Approach

1. Find a set of distinctive key-points
Detection:
General Approach

1. Find a set of distinctive key-points
2. Define a region around each keypoint
Detection::

General Approach

1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Compare regions
Visual Descriptors

**Detection::**

**General Approach**

1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Compare regions

Different region appearance for different keypoints

Not a problem
Detection::

General Approach

1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Compare regions

Different region appearance for matching keypoints

Problem
Visual Descriptors

The process illustrates 3 basic challenges in vision:

What to look at? (Keypoints)

How to find it? (detectors)

How to describe it? (descriptors)
Visual Descriptors: are descriptions of the visual features of the contents in images or video. They describe elementary characteristics that are hopefully invariant over different appearances.

What we need?
1. Description which is invariant to various transformations
2. Description which is distinctive for different keypoints
3. Description which is local, therefore robust to occlusion and clutter
Global vs. Local descriptors

• The **global** image descriptor is being computed on the entire image

• The **local** image description is founded on the premise that images can be characterized by attributes computed on regions of the image
The perfect local descriptor...

Will be invariant to:

1. Rotation
2. Translation
3. Scale changes
4. Illumination
5. Noise

Other transformations...
**Simplest descriptor**: list of intensities within a patch

- Write regions as vectors
Visual Descriptors:: Local descriptors

Simplest descriptor:: list of intensities within a patch

- Write regions as vectors
- Match keypoints by measuring distance between vectors

What is it going to be invariant to?
Visual Descriptors:: Local descriptors

**Simplest descriptor::** list of intensities within a patch

- Write regions as vectors
- Match keypoints by measuring distance between vectors

What is it going to be invariant to?

Not much...
Small shifts can affect matching score a lot
Visual Descriptors:: Local descriptors

Simplest descriptor:: list of intensities within a patch

Solution1:: Histogram of intensities

What is it going to be invariant to?
Visual Descriptors:: Local descriptors

**Simplest descriptor::** list of intensities within a patch

**Solution 2::** Histogram of gradients
Local Desc.: HOG (Histogram of Oriented Gradients)

1. Partition an image into small squared cells
2. Compute a histogram of oriented gradients in each cell
3. Return a descriptor for each cell

HOG Visual Descriptors:: Local descriptors

8x8 cell size
Visual Descriptors:: Local descriptors

Local (Detector &) Desc.:: SIFT (Scale Invariant Feature Transform)

- Detect and describe local features in images
- Select keypoints that are stable in scale space
Recap:: Multi scale signal representation:: Scale space

\[ L(x, y; 0) = f(x, y) \]
\[ L(x, y; 1) \]
\[ L(x, y; 16) \]
\[ L(x, y; 256) \]
Recap:: Multi scale signal representation:: Bandpass Pyramids

Gaussian pyramid

\[ G^{(0)} \]

\[ G^{(1)} = (W * G^{(0)}) \downarrow \]

\[ G^{(2)} = (W * G^{(1)}) \downarrow \]

Us-sample

Down-sample

W

W

W

Laplacian pyramid

\[ P^{(1)} \sim G^{(1)} - W * G^{(1)} \]

\[ P^{(2)} = G^{(2)} \]

\[ P^{(0)} \sim G^{(0)} - W * G^{(0)} \]
Visual Descriptors:: Local descriptors

**SIFT Detector**

1. **Scale-space extremum detection**: Candidate keypoints are defined as maxima and minima of the DoG applied in scale space

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

\[ D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma) \]
SIFT Detector

2. Keypoint localization:: Compute a taylor expansion of the Dog around the candidate keypoint

\[ D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \]

\[ \hat{x} = - \frac{\partial^2 D}{\partial x^2} \frac{1}{\partial x} \frac{\partial D}{\partial x}. \]

*The initial approach was just to locate each keypoint at the location and scale of the candidate keypoint, which resulted in poor performances*
SIFT Detector

2.1 Reject low-contrast keypoints :: Discard of low-contrast keypoints by checking the value of the second order Taylor expansion at the offset

\[ D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x}. \]
2.2 Reject edge-like keypoints: We do that by solving the eigenvalues of the Hessian matrix.

Edge points are poorly located

Edge points have a much larger principal curvature across the edge than along the edge

Finding Principal curvatures amounts to finding the eigenvalues of the Hessian

\[
H = \begin{bmatrix}
D_{xx} & D_{xy} \\
D_{xy} & D_{yy}
\end{bmatrix}
\]

\[
\begin{align*}
\text{Tr}(H) &= D_{xx} + D_{yy} = \alpha + \beta, \\
\text{Det}(H) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta
\end{align*}
\]

\[
\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r+1)^2}{r}
\]
SIFT Detector

3. Orientation assignment: Compute gradient for each pixel in a region around the keypoint, and create a histogram of gradients (36 angle bins)

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

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

- Each gradient is weighted by its magnitude
- The peak of the histogram defines a dominant orientation
- If more than one orientations exist (peaks that are within 80% of the highest peak), additional keypoints are created (same location, scale)
SIFT Descriptor:

1. Find the blurred image of the closest scale
2. Rotate the region by the dominant orientation
3. Divide the region into a 4x4 grid
4. Create a histogram for each cell (8 angels)
5. Normalize the vector

Visual Descriptors:: Local descriptors

(8 angels) x (4x4 grid) = 128 dimension descriptor
**Visual Descriptors:: Local descriptors**

**SIFT Example**

- Original image
- SIFT Detector
- SIFT keypoints
- SIFT Descriptor
- SIFT descriptors
SIFT Summary

1. **Scale Invariance**: keypoints are located using scale-space representation

2. **Rotation Invariance**: keypoints are aligned with dominant orientation

3. **Illumination Invariance**: Sift descriptor vector is normalized
Self-Similarity::

Images of the same object often **do not share** common image properties (e.g. color, texture)

Images of the same object **do share** a similar geometric layout of local internal self-similarities
Self-Similarity::

Self similarity descriptor overlook image properties and aims to extract the pattern that is shared between all appearances of the same object.
Local “Self-Similarity descriptor”::

1. Extract a patch $P_q$ (typically 5x5) around pixel $q$
1. Extract a region $R_q$ (typically 40x40) around pixel $q$
Local “Self-Similarity descriptor”::

3. Compare $P_q$ with region $R_q$ using sum of squared differences

$$S_q(x, y) = \exp \left( - \frac{SSD_q(x, y)}{\max \left( \text{var}_{\text{noise}}, \text{var}_{\text{auto}}(q) \right)} \right)$$

Image patch  
Image region  
Correlation surface
Local “Self-Similarity descriptor”::

4. Transform correlation surface $S_{q}(x, y)$ into log-polar coordinates centered at $q$, and partitioned into 80 bins (20 angles, 4 radial intervals). Select the maximal correlation value in each bin.
Local “Self-Similarity descriptor”::

5. Flatten & normalize to get the final descriptor
Visual Descriptors:: Local descriptors

Matching global ensembles of local descriptors::

1. compute the local self similarity descriptors densely throughout F and G (compute every 5\textsuperscript{th} pixel)

2. Filter out non-informative descriptors
   - descriptors that do not catch any local self-similarity
   - descriptors that contain high self similarity everywhere
Matching global ensembles of local descriptors:

- All the local descriptors in F form together a global “ensemble of descriptors”
- A good match of F in G corresponds to finding a similar ensemble of descriptors in G

3. Find a good match using an “ensemble matching” algorithm