Computer Vision

4. Interest Points

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany
Outline

1. Smoothing, Image Derivatives, Convolutions
2. Edges, Corners, and Interest Points
3. Image Patch Descriptors
4. Interest Point Matching
5. A Simple Application: Image Stitching
Outline

1. Smoothing, Image Derivatives, Convolutions

2. Edges, Corners, and Interest Points

3. Image Patch Descriptors

4. Interest Point Matching

5. A Simple Application: Image Stitching
Smoothing / Blurring / Averaging

- **Smoothing**: Replace each pixel by the weighted average of its surrounding patch:

\[ I_{\text{smooth}}(x, y; w) := \sum_{\Delta x, \Delta y} w(-\Delta x, -\Delta y) I(x + \Delta x, y + \Delta y) \]

\[ = \sum_{x', y'} w(x - x', y - y') I(x', y') \]

- **padding** with 0 at the image boundaries.

- **example**: box kernel

\[ w_{-2:2,-2:2}(\Delta x, \Delta y) := \frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \]

- **Gaussian smoothing**: smoothing with a Gaussian kernel.
Gaussian Kernels

- Precomputed weights: (clipped) Gaussian density values

\[
\tilde{w}(\Delta x, \Delta y) := \begin{cases} 
\mathcal{N}(\sqrt{\Delta x^2 + \Delta y^2}; 0, \sigma^2), & \text{if } |\Delta x| \leq K, |\Delta y| \leq K \\
0, & \text{else}
\end{cases}
\]

\[
w(\Delta x, \Delta y) := \frac{\tilde{w}(\Delta x, \Delta y)}{\sum_{\Delta x', \Delta y'} \tilde{w}(\Delta x', \Delta y')}
\]

- clipped: small support, window size $K$.

- example ($K = 2, \sigma^2 = 1$):

\[
w_{-2:2,-2:2} := \begin{pmatrix}
0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
0.013 & 0.060 & 0.098 & 0.060 & 0.013 \\
0.022 & 0.098 & 0.162 & 0.098 & 0.022 \\
0.013 & 0.060 & 0.098 & 0.060 & 0.013 \\
0.003 & 0.013 & 0.022 & 0.013 & 0.003
\end{pmatrix}
\]

Note: \( \mathcal{N}(x; \mu, \sigma^2) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \).
Blurring / Example

original:

blurred by $G(K = 5, \sigma = 1)$:
Blurring / Example

original:

blurred by $G(K = 5, \sigma = 10)$:
Blurring / Example

original:  

blurred by $G(K = 50, \sigma = 1)$:
Blurring / Example

original:

blurred by $G(K = 50, \sigma = 10)$:
Image Derivatives

- **Image Derivative**: How does the intensity values change in x or y direction?

\[
I_X(x, y) := I(x, y) - I(x - 1, y)
\]
\[
I_Y(x, y) := I(x, y) - I(x, y - 1)
\]

or symmetric

\[
I_X(x, y) := 2I(x, y) - I(x - 1, y) - I(x + 1, y)
\]
\[
I_Y(x, y) := 2I(x, y) - I(x, y - 1) - (x, y - 2)
\]
Image Derivatives / Example

original (grayscale):

derivative in x-direction:
Image Derivatives / Example

original (grayscale):

derivative in y-direction:
Convolutions

- Smoothing, Image Derivatives and further operations such as filtering can be represented by a convolution: an image where each pixel \((x, y)\) represents the weighted sum around \((x, y)\) in image \(I\) weighted with \(w\):

\[
(w \ast I)(x, y) := \sum_{x', y'} w(x - x', y - y')I(x', y')
\]

- Examples:

\[
\begin{align*}
I_{\text{smooth}} &:= w \ast I \\
I_X(x, y) &:= I(x, y) - I(x - 1, y) \\
I_Y(x, y) &:= I(x, y) - I(x, y - 1) \\
\text{or } I_X(x, y) &:= 2I(x, y) - I(x - 1, y) - I(x + 1, y) \\
I_Y(x, y) &:= 2I(x, y) - I(x, y - 1) - (x, y + 1)
\end{align*}
\]
Convolutions / Associativity

- Convolutions are associative:

\[ I \ast (J \ast K) = (I \ast J) \ast K \]

- Example:
  First smooth an image with Gaussian \( w \) from slide 2, then compute its x-derivative with \( \begin{pmatrix} -1 & 2 & -1 \end{pmatrix} \):
  \( \leadsto \) just convolve with \( \begin{pmatrix} -1 & 2 & -1 \end{pmatrix} \ast w \)

\[
\begin{pmatrix} -1 & 2 & -1 \end{pmatrix} \ast w =
\begin{pmatrix}
-0.007 & 0.002 & 0.017 & 0.002 & -0.007 \\
-0.033 & 0.008 & 0.077 & 0.008 & -0.033 \\
-0.054 & 0.077 & 0.128 & 0.077 & -0.054 \\
-0.033 & 0.008 & 0.077 & 0.008 & -0.033 \\
-0.007 & 0.002 & 0.017 & 0.002 & -0.007 \\
\end{pmatrix}
\]
Outline

1. Smoothing, Image Derivatives, Convolutions

2. Edges, Corners, and Interest Points

3. Image Patch Descriptors

4. Interest Point Matching

5. A Simple Application: Image Stitching
Edges, Corners, and Interest Points

- Good candidates for points that are easy to recognize and match in two images are:
  - Points on edges
  - Corners

  *I.e.*, points with sudden intensity changes.

- Two stage approach: given an image $I \in \mathbb{R}^{N \times M}$,
  1. Compute an interestingness measure $i \in \mathbb{R}^{N \times M}$ for points,
  2. Select a useful set of points $p_1, \ldots, p_K \in [N] \times [M]$
    - With high interestingness measure
    - Not too close to each other.

- Many names: corners, interest points, keypoints, salient points, ...

Note: $[N] \coloneqq \{1, \ldots, N\}$. 
Gradient Magnitude (Canny Edge Detector)

- Simply use the **magnitude of the gradient** as interestingness measure:

\[
i(x, y) = \sqrt{(D_X * I)(x, y)^2 + (D_Y * I)(x, y)^2}
\]

- \(D_X, D_Y\): differentiation kernels, e.g.,

\[
D_X := \begin{pmatrix}
-1 & 2 & -1
\end{pmatrix},
\quad D_Y := \begin{pmatrix}
-1 \\
2 \\
-1
\end{pmatrix}
\]
Gradient Magnitude / Example

original (grayscale):

gradient magnitude:
Gradient Magnitude / Example

original (grayscale):

overlay with 500 interest points:
Laplacian of Gaussian and Difference of Gaussian

Further simple interestingness measures:

- **Laplacian of Gaussian (LoG):**

\[
i(x, y) = (((D_X \ast D_X + D_Y \ast D_Y) \ast G) \ast I)(x, y)
\]

  - uses second order information

- **Difference of two Gaussians (DoG):**

\[
i(x, y) = ((G_{\sigma_1} - G_{\sigma_2}) \ast I)(x, y), \quad \sigma_1 \neq \sigma_2
\]

  - uses variations at different scales
  - often interpreted as limit of Laplacian of Gaussians

\[
((D_X \ast D_X + D_Y \ast D_Y) \ast G_{\sigma}) \ast I \approx \frac{\sigma}{\Delta \sigma} ((G_{\sigma+\Delta \sigma} - G_{\sigma-\Delta \sigma}) \ast I)
\]
Harris Corner Detector

- Represent a corner by its patch surrounding it, represent such a patch by a weight function $w : [N] \times [M] \rightarrow \mathbb{R}$, i.e.,
  
  $$w(x, y) := \begin{cases} 
1, & \text{if } |x - x_0| < 3 \text{ and } |y - y_0| < 3 \\
0, & \text{else}
\end{cases}$$

  for a rectangular patch of size 5 centered around $(x_0, y_0)$.

- A point is easy to identify, if its minimum in the autocorrelation surface is pronounced:

  $$E(\Delta x, \Delta y; w) := \sum_{x, y} w(x, y)(I(x + \Delta x, y + \Delta y) - I(x, y))^2$$
Harris Corner Detector / Autocorrelation Surface

Note: left to right: flower bed, roof edge, cloud.

[Sze11, p. 187]
Harris Corner Detector

\[ E(\Delta x, \Delta y; w) := \sum_{x,y} w(x, y)(I(x + \Delta x, y + \Delta y) - I(x, y))^2 \]

with Hessian at minimum:

\[ H(0, 0; w) \approx 2 \sum_{x,y} w(x, y) \nabla I|_{(x,y)} \nabla I|_{(x,y)}^T, \quad \text{for } \frac{\partial^2 I}{\partial^2 (x, y)} := 0 \]

\[ = 2w \ast \begin{pmatrix} (I_X)^2 & I_X I_Y \\ I_X I_Y & (I_Y)^2 \end{pmatrix}, \]

\[ I_X(x, y) := I(x + 1, y) - I(x, y) \approx \frac{\partial I}{\partial x}(x, y) \]

\[ I_Y(x, y) := I(x, y + 1) - I(x, y) \approx \frac{\partial I}{\partial y}(x, y) \]

Note: \( I \ast J(x, y) := \sum_{x',y'} I(x - x', y - y')J(x', y') \) convolution of two images.
Harris Corner Detector

use SVD to assess steepness

\[ H = U \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} U^T, \quad \sigma_1 \geq \sigma_2 \geq 0, \quad UU^T = I \]

and define interestingness measure:

\[ i_{\text{Shi-Tomasi}}(x, y) := \sigma_2 \]

\[ i_{\text{Harris}}(x, y) := \sigma_1 \sigma_2 - \alpha (\sigma_1 + \sigma_2)^2 = \det H - \alpha \text{trace}(H)^2, \quad \alpha := 0.06 \]

\[ i_{\text{Triggs}}(x, y) := \sigma_2 - \alpha \sigma_1, \quad \alpha := 0.05 \]

\[ i_{\text{Brown}}(x, y) := \sigma_1 \sigma_2 / (\sigma_1 + \sigma_2) = \det H / \text{trace}(H) \]

- the larger \( \sigma_{1:2} \), the steeper the autocorrelation surface \( E \).
- Harris and Brown avoid computing \( \sigma_1, \sigma_2 \) explicitly (which requires computing a square root).
Harris Corner Detector / Algorithm

1: procedure INTERESTPOINTS-HARRIS($I \in \mathbb{R}^{N \times M}$; $w \in \mathbb{R}^{K \times K \times L \times L}$, $\alpha \in \mathbb{R}$)
2: $I_X := D_X \ast I$
3: $I_Y := D_Y \ast I$
4: $I_X^2 := I_X \cdot I_X$
5: $I_Y^2 := I_Y \cdot I_Y$
6: $I_X I_Y := I_X \cdot I_Y$
7: $A := w \ast I_X^2$ \hspace{1cm} \triangledown \text{compute } \begin{pmatrix} A(x, y) & C(x, y) \\ C(x, y) & B(x, y) \end{pmatrix}$
8: $B := w \ast I_Y^2$
9: $C := w \ast I_X I_Y$
10: $i := A \cdot B - C \cdot C - \alpha (A + B) \cdot (A + B)$
11: return $i$

\triangledown $D_X, D_Y$: differentiation kernels, e.g.,

$D_X := \begin{pmatrix} -1 & 2 & -1 \end{pmatrix}$, $D_Y := \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$.

Note: $\cdot$ denotes the element/pixelwise product.
Harris Corner Detector / Example

(a) (b) (c)

Figure 4.8: Sample image (a) and two different interest operator responses: (b) Harris; (c) DoG.
The circle sizes and colors indicate the scale at which each interest point was detected.
Notice how the two detectors tend to respond at complementary locations.

The steps in the basic auto-correlation-based keypoint detector are summarized in Algorithm 4.1.

Figure 4.8 shows the resulting interest operator responses for the classic Harris detector as well as the DoG detector discussed below.

Adaptive non-maximal suppression (ANNS).

While most feature detectors simply look for local maxima in the interest function, this can lead to an uneven distribution of feature points across the image, e.g., points will be denser in regions of higher contrast. To mitigate this problem, Brown et al. (2005) only detect features that are both local maxima and whose response value is significantly (10%) greater than than of all of its neighbors within a radius \( r \) (Figure 4.9c–d). They devise an efficient way to associate suppression radii with all local maxima by first sorting them by their response strength, and then creating a second list sorted by decreasing suppression radius (see (Brown et al. 2005) for details). A qualitative comparison of selecting the top \( n \) features vs. ANMS is shown in Figure 4.9.

Measuring repeatability.

Given the large number of feature detectors that have been developed in computer vision, how can we decide which ones to use? Schmid et al. (2000) were the first to propose measuring the repeatability of feature detectors, which they define as the frequency with which keypoints detected in one image are found within \( \epsilon \) (say \( \epsilon = 1.5 \)) pixels of the corresponding location in a transformed image. In their paper, they transform their planar images by applying rotations, scale changes, illumination changes, viewpoint changes, and adding noise. They also measure the information content available at each detected feature point, which they define as the entropy of a set of rotationally invariant local grayscale descriptors. Among the techniques they survey, they find that the improved (Gaussian derivative) version of the Harris operator with \( \sigma_d = 1 \) (scale of the derivative Gaussian) and \( \sigma_i = 2 \) (scale of the integration Gaussian) works best.

[Sze11, p. 213]
Interest Points at Different Scales (SIFT Detector)

- Interest points also can be identified at different scales in parallel:

\[ i(p, s) := (G_{\sigma_{s+1}} * I - G_{\sigma_s} * I), \quad s \in [S] \]

where

\[ \sigma_1 > \sigma_2 > \cdots > \sigma_S \]

where \( S \in \mathbb{N} \) is the number of scale levels

- Often scale levels are grouped by octaves:
  - each octave is represented by a downsampling by a factor 2
  - scales within an octave are \( \sigma_s := 2^{s/S_o} \sigma \) (with \( S_o \) the number of scale levels within an octave)
Interest Points at Different Scales (SIFT Detector)

In addition to dealing with scale changes, most image matching and object recognition algorithms need to deal with (at least) in-plane image rotation. One way to deal with this problem is to design algorithms that need to be invariant to scale and rotation.

Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce images separated by a constant factor in scale space (i.e., doubling of \( \sigma \)). The relationship between neighboring Gaussians can be understood from the heat diffusion equation, where the second derivative with respect to scale is approximately equal to the Laplacian of a Gaussian function at each detected Harris point (in a multi-scale pyramid).

While Lowe's Scale Invariant Feature Transform (SIFT) performs well in practice, it is not perfect. This is because it relies on detecting Harris corners, which have maxima that are a continuous function of the image, for some ellipse with intermediate elongation. There can be two maxima near each end of the ellipse. As the locations of critical points for more circular positive central regions of maxima are a continuous function of the image, for some ellipse with intermediate elongation, there will be two maxima near each end of the ellipse. As the locations of critical points for more circular positive central regions of maxima are a continuous function of the image, for some ellipse with intermediate elongation, there will be two maxima near each end of the ellipse. As the locations of critical points for more circular positive central regions of maxima are a continuous function of the image, for some ellipse with intermediate elongation, there will be two maxima near each end of the ellipse. As the locations of critical points for more circular positive central regions of maxima are a continuous function of the image, for some ellipse with intermediate elongation, there will be two maxima near each end of the ellipse.

Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with a red circle) to its eight neighbors in the current image and nine neighbors in the scale above and below. The Laplacian of a Gaussian function at each detected Harris point (in a multi-scale pyramid) is evaluated. This shows that the normalization of the Laplacian with the factor \( \sigma \) is no different than for the start of the previous octave, while computation is greatly reduced.

This shows that when the difference-of-Gaussian function has scales differing by a constant factor in scale space, shown stacked in the left column. We choose to divide each octave into a number of intervals, so that final extrema detection or localization for even significant differences in scale, such as \( k \) times the octave, can be eliminated following the first few checks.

We must produce \( 2^m - 1 \) images in the stack of blurred images for each octave, so that final extrema detection or localization for even significant differences in scale, such as \( k \) times the octave, can be eliminated following the first few checks. An important issue is to determine the frequency of sampling in the image and scale domain.

The normalization of the Laplacian with the factor \( \sigma \) is no different than for the start of the previous octave, while computation is greatly reduced.

An efficient approach to construction of \( D \) images in the stack of blurred images for each octave, so that final extrema detection or localization for even significant differences in scale, such as \( k \) times the octave, can be eliminated following the first few checks.
Non-Maximum Suppression

- Often neighbors of interest points have similar high interestingness, yielding redundant close-by interest points.

- Keep only interest points that are **local maxima** in their neighborhood:

\[
i'(p) := \begin{cases} 
i(p), & \text{if } i(p) > i(p') \forall p' \in N(p) \\
0, & \text{else}
\end{cases}, \quad p \in [N] \times [M]
\]

with **neighborhood**

\[
N_K(p) := \{ p' \in [N] \times [M] \mid |p_x - p'_x| \leq K, |p_y - p'_y| \leq K, p' \neq p \}
\]

**rectangular**

\[
N_K(p) := \{ p' \in [N] \times [M] \mid ||p - p'|| \leq K, p' \neq p \}
\]

**circular**
Non-Maximum Suppression / Example

(a) Strongest 250

(b) Strongest 500

(c) ANMS 250, $r = 24$

(d) ANMS 500, $r = 16$

Note: ANMS = adaptive non-maximum suppression; see the book for details.

[Sze11, p. 214]
Non-Maximum Suppression / At Different Scale

- Non-Maximum Suppression also can be extended to work on interest points at different scale:

\[ N_K(p, s) := \{(p', s') \in [N] \times [M] \times [S] \mid |p_x - p'_x| \leq K, |p_y - p'_y| \leq K, \\
|s - s'| \leq 1, p' \neq p \} \]

[Sze11, p. 216]
SIFT Interest Points

SIFT refines interest points by further steps:

- non-maximum suppression at different scale
- localization of interest points at sub-pixel granularity
Outline

1. Smoothing, Image Derivatives, Convolutions

2. Edges, Corners, and Interest Points

3. Image Patch Descriptors

4. Interest Point Matching

5. A Simple Application: Image Stitching
Image Patch Descriptors

- Which properties from a patch to extract?
  - grayscale intensities, color intensities, gradient directions

- Which patches to extract?
  - orientation of the patch w.r.t. the image frame
  - offset of the patch w.r.t. the interest point (cells)
Histograms

- the most simple patch:
  - a square centered on the interest point
Histories

- the most simple patch:
  - a square centered on the interest point

- properties:
  - most simple: grayscale intensities of the pixels

- how to represent?
  - as a matrix or a vector
Histories

- the most simple patch:
  - a square centered on the interest point

- properties:
  - most simple: grayscale intensities of the pixels

- how to represent?
  - as a matrix or a vector
    - is affected by rotations
Histograms

- the most simple patch:
  - a square centered on the interest point

- properties:
  - most simple: grayscale *intensities* of the pixels

- how to represent?
  - as a matrix or a vector
    - is affected by rotations
  - by some scalar properties (mean, standard deviation)
Histograms

- the most simple patch:
  - a square centered on the interest point

- properties:
  - most simple: grayscale **intensities** of the pixels

- how to represent?
  - as a matrix or a vector
    - is affected by rotations
  - by some scalar properties (mean, standard deviation)
    - represents only little information
Histograms

- the most simple patch:
  - a square centered on the interest point

- properties:
  - most simple: grayscale intensities of the pixels

- how to represent?
  - as a matrix or a vector
    - is affected by rotations
  - by some scalar properties (mean, standard deviation)
    - represents only little information
  - by its histogram
Histograms

- the most simple patch:
  - a square centered on the interest point

- properties:
  - most simple: grayscale **intensities** of the pixels
    - is affected by global intensity fluctuations
  - **gradient directions**

- how to represent?
  - as a matrix or a vector
    - is affected by rotations
  - by some scalar properties (mean, standard deviation)
    - represents only little information
  - by its **histogram**
represent interest point \((x, y)\) by its \(B\)-dimensional **intensity histogram features** \(\phi(x, y)\):

\[
\phi(x, y)_b := \left| \{(x', y') \in \mathcal{N}(x, y) \mid I(x', y') \in \text{bin}_b\} \right|, \quad b = 0, \ldots, B - 1
\]

\[
\text{bin}_b := \left[ \frac{b}{B} I_{\text{max}} , \frac{b + 1}{B} I_{\text{max}} \right]
\]

\[
\mathcal{N}(x, y) := \{(x', y') \in [N] \times [M] \mid |x' - x| < K, |y' - y| < K\}
\]

for intensities \(I(x, y)\) in range \([0, I_{\text{max}}]\).
Histograms / Smoothed Counting

▶ To avoid non-continuous changes if a value crosses bin boundaries, values can be counted
  ▶ in both closest bins,
  ▶ antiproportional to their distance from the bin center

\[
\text{binc}_b := \frac{b + 0.5}{B} I_{\text{max}}
\]

\[
\text{bin}_b := \sum_{(x', y') \in \mathcal{N}(x,y)} \max(0, 1 - \frac{|I(x', y') - \text{binc}_b|}{I_{\text{max}}/B})
\]

▶ sometimes called trilinear counting.
Histograms / Gradient Directions

- represent interest point \((x, y)\) by its \(B\)-dimensional \textbf{gradient directions histogram features} \(\phi(x, y)\):

\[
\phi(x, y)_b := |\{(x', y') \in \mathcal{N}(x, y) \mid d(x', y') \in \text{bin}_b\}|, \quad b = 0, \ldots, B - 1
\]

\[
d(x, y) := \tan^{-1}((D_Y * I)(x, y)/(D_X * I)(x, y))
\]

\[
\text{bin}_b := \left[\frac{b}{B}2\pi, \frac{b + 1}{B}2\pi\right]
\]
Histogmams / Gradient Directions

▶ represent interest point \((x, y)\) by its \(B\)-dimensional gradient directions histogram features \(\phi(x, y)\):

\[
\phi(x, y)_b := |\{(x', y') \in \mathcal{N}(x, y) \mid d(x', y') \in \text{bin}_b\}|, \quad b = 0, \ldots, B - 1
\]

\[
d(x, y) := \tan^{-1}((D_Y * I)(x, y)/(D_X * I)(x, y))
\]

\[
\text{bin}_b := \left[\frac{b}{B} 2\pi, \frac{b + 1}{B} 2\pi\right]
\]

▶ variant: weight gradients by their magnitude:

\[
\phi(x, y)_b := \sum_{(x', y') \in \mathcal{N}(x, y), d(x', y') \in \text{bin}_b} (D_X * I)(x', y')^2 + (D_Y * I)(x', y')^2
\]
A dominant orientation estimate can be computed by creating a histogram of all the gradient orientations (weighted by their magnitudes and/or after thresholding out small gradients), and then finding the significant peaks in this distribution (Lowe 2004). Descriptors that are rotationally invariant (Schmid and Mohr 1997), but such descriptors have poor discriminability, i.e., they map different looking patches to the same descriptor.

A better method is to estimate a dominant orientation at each detected keypoint. Once the local orientation and scale of a keypoint have been estimated, a scaled and oriented patch around the detected point can be extracted and used to form a feature descriptor (Figures 4.10 and 4.17).

The simplest possible orientation estimate is the average gradient within a region around the keypoint. If a Gaussian weighting function is used (Brown et al. 2005), this average gradient is equivalent to a first order steerable filter §3.2.1, i.e., it can be computed using an image convolution with the horizontal and vertical derivatives of Gaussian filter (Freeman and Adelson 1991). In order to make this estimate more reliable, it is usually preferable to use a larger aggregation window (Gaussian kernel size) than the detection window size (Brown et al. 2005). The orientations of the square boxes shown in Figure 4.10 were computed using this technique.

Sometimes, however, the averaged (signed) gradient in a region can be small and therefore an unreliable indicator of orientation. A more reliable technique is to look at the histogram of orientations computed around the keypoint. Lowe (2004) computes a 36-bin histogram of edge orientations weighted by both gradient magnitude and Gaussian distance to the center, finds all peaks within 80% of the global maximum, and then computes a more accurate orientation estimate using a 3-bin parabolic fit (Figure 4.12).
Block Descriptors

- Describe an interest point not just by features of the surrounding patch, but by the features of several neighboring patches (blocks, cells):

\[
\phi(x, y) := \bigoplus_{(x', y') \in C(x, y)} \phi'(x', y')
\]

\[
C(x, y) := \{x + c\Delta X, y + d\Delta Y \mid c, d \in \{-C, \ldots, C\}\}
\]

- Often a simple partition of a large \((2C + 1)(2K + 1) \times (2C + 1)(2K + 1)\) patch is used \((\Delta X = \Delta Y = 2K + 1)\).

- Features have dimensions \((2C + 1)^2 B\).

Note: \((x_1, \ldots, x_N) \oplus (y_1, \ldots, y_M) := (x_1, \ldots, x_N, y_1, \ldots, y_M)\) concatenation.
Block Descriptors

Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

6.1 Descriptor representation

Figure 7 illustrates the computation of the keypoint descriptor. First the image gradient magnitudes and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation. For efficiency, the gradients are precomputed for all levels of the pyramid as described in Section 5. These are illustrated with small arrows at each sample location on the left side of Figure 7.

A Gaussian weighting function with $\sigma$ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point. This is illustrated with a circular window on the left side of Figure 7, although, of course, the weight falls off smoothly. The purpose of this Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are far from the center of the descriptor, as these are most affected by misregistration errors.

The keypoint descriptor is shown on the right side of Figure 7. It allows for significant shift in gradient positions by creating orientation histograms over 4x4 sample regions. The figure shows eight directions for each orientation histogram, with the length of each arrow corresponding to the magnitude of that histogram entry. A gradient sample on the left can shift up to 4 sample positions while still contributing to the same histogram on the right, thereby achieving the objective of allowing for larger local positional shifts.

It is important to avoid all boundary affects in which the descriptor abruptly changes as a sample shifts smoothly from being within one histogram to another or from one orientation to another. Therefore, trilinear interpolation is used to distribute the value of each gradient sample into adjacent histogram bins. In other words, each entry into a bin is multiplied by a weight of $1 - d$ for each dimension, where $d$ is the distance of the sample from the central value of the bin as measured in units of the histogram bin spacing.
Align Patches by the Gradient Direction of the Interest Point

- Extract features from the image rotated by
  - the negative gradient direction at the interest point
  - around the interest point

(afterwards the gradient at the interest point \((x, y)\) points towards positive \(x\)-direction):

\[
\psi := - \, d(x, y)
\]

\[
R_\psi(x', y') := \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos \psi & - \sin \psi \\ \sin \psi & \cos \psi \end{pmatrix} \left( \begin{pmatrix} x' \\ y' \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} \right)
\]

\[
l_{bi}(x, y) := (1 - (x - \lfloor x \rfloor))(1 - (y - \lfloor y \rfloor)) \, l(\lfloor x \rfloor, \lfloor y \rfloor)
+ (x - \lfloor x \rfloor)(1 - (y - \lfloor y \rfloor)) \, l(\lfloor x \rfloor, \lceil y \rceil)
+ (1 - (x - \lfloor x \rfloor))(y - \lfloor y \rfloor) \, l(\lceil x \rceil, \lfloor y \rfloor)
+ (x - \lfloor x \rfloor)(y - \lceil y \rceil) \, l(\lceil x \rceil, \lceil y \rceil)
\]

(bilinear interpolation)
SIFT descriptors

- patches:
  - extract from the scaled image the interest point has been detected on
  - align patch by the gradient direction of the interest point
  - $16 \times 16$, partitioned into 16 blocks a $4 \times 4$

- block features:
  - gradient directions
  - weighted by a Gaussian of the distance to the interest point

- block feature aggregation:
  - smoothly counted histograms
  - 8 bins

- $\mapsto$ feature vector $\phi \in \mathbb{R}^{128}$

- normalization in 3 steps:

$$
\phi'_i := \phi_i / \|\phi\|_2, \\
\phi''_i := \min(0.2, \phi'_i), \\
\phi'''_i := \phi''_i / \|\phi''\|_2
$$
Image Descriptors

To describe a whole image (not just a patch), two main approaches are used:

1. Concatenate patch descriptors of equally spaced “interest points”
   1.1 e.g., used in **Histograms of Oriented Gradients (HoG)**
Image Descriptors

To describe a whole image (not just a patch), two main approaches are used:

1. Concatenate patch descriptors of equally spaced “interest points”
   1.1 e.g., used in **Histograms of Oriented Gradients (HoG)**

2. **Bag of words descriptors**:
   2.1 compute interest points and their descriptors for a set of images
   2.2 discretize the descriptors
      ▶ e.g., clustering in $K$ clusters using k-means
   2.3 represent each image by the $K$ cluster frequencies of their interest point descriptors
Histograms of Oriented Gradients (HoG)

Figure 13.17 HOG descriptor. a) Original image. b) Gradient orientation, quantized into nine bins from 0 to 180°. c) Gradient magnitude. d) Cell descriptors are 9D orientation histograms that are computed within 6 × 6 pixel regions. e) Block descriptors are computed by concatenating 3 × 3 blocks of cell descriptors. The block descriptors are normalized. The final HOG descriptor consists of the concatenated block descriptors.

[Pri12, p. 343]
Outline

1. Smoothing, Image Derivatives, Convolutions

2. Edges, Corners, and Interest Points

3. Image Patch Descriptors

4. Interest Point Matching

5. A Simple Application: Image Stitching
Settings, Assumptions, Distances

Two settings:

- match interest points in different scenes
  - goal: detect similar objects
    (object identification)
  - coordinates of the points do not matter

\[
d((x_1, y_1), (x_2, y_2)) := d'(\phi(x_1, y_1), \phi(x_2, y_2)) = ||\phi(x_1, y_1) - \phi(x_2, y_2)||_2
\]

- match interest points in two views of the same scene
  - goal: detect corresponding points in different views of the same scene
    (required for SLAM)
  - coordinates of corresponding points also should be close,

\[
d((x_1, y_1), (x_2, y_2)) := \alpha d'(\phi(x_1, y_1), \phi(x_2, y_2)) + \beta d'(\phi(x_1, y_1), \phi(x_2, y_2)) + \beta ||\phi(x_1, y_1) - \phi(x_2, y_2)||_2
\]
Settings, Assumptions, Distances

Two settings:

- match interest points in different scenes
  - goal: detect similar objects (object identification)
  - coordinates of the points do not matter
    \[ d((x_1, y_1), (x_2, y_2)) := d'(\phi(x_1, y_1), \phi(x_2, y_2)) = ||\phi(x_1, y_1) - \phi(x_2, y_2)||_2 \]

- match interest points in two views of the same scene
  - goal: detect corresponding points in different views of the same scene (required for SLAM)
  - coordinates of corresponding points also should be close, e.g.,
    \[ d((x_1, y_1), (x_2, y_2)) := \alpha d'(\phi(x_1, y_1), \phi(x_2, y_2)) + \beta d'(\phi(x_1, y_1), \phi(x_2, y_2)) \]
    \[ = \alpha || (x_1, y_1) - (x_2, y_2) ||_2 + \beta ||\phi(x_1, y_1) - \phi(x_2, y_2)||_2 \]
Simple methods

To match two sets $P$ and $Q$ of interest points:

- match interest points by **distance threshold**

\[ p \sim q : \iff d(p, q) < d_{\text{max}}, \quad p \in P, q \in Q \]

- distance threshold $d_{\text{max}}$ can be estimated from known matches and non-matches
Simple methods

To match two sets $P$ and $Q$ of interest points:

- match interest points by **distance threshold**

  $p \sim q : \Leftrightarrow d(p, q) < d_{\text{max}}, \quad p \in P, q \in Q$

- distance threshold $d_{\text{max}}$ can be estimated from known matches and non-matches

- match interest points by **nearest neighbor**

  $p \sim q : \Leftrightarrow q = \arg \min_{q \in Q} d(p, q)$
Nearest Neighbor Distance Ratio

- match interest points by nearest neighbor distance ratio (NNDR)

\[ p \sim q \iff i) \ q = \arg \min_{q \in Q} d(p, q) \text{ and } \]

\[ ii) \ \text{NNDR}(p, q) := \frac{d(p, q)}{d(p, q')} < \text{NNDR}_{\text{min}}, \quad q' := \arg \min_{q' \in Q \setminus \{q\}} d(p, q') \]

![Diagram showing Nearest Neighbor Distance Ratio](image-url)
Comparison of Different Descriptors & Matchings

a) fixed threshold:

---

[Sze11, p. 229]
Comparison of Different Descriptors & Matchings

b) nearest neighbor:

![Graph showing the performance of different feature descriptors compared with different matching strategies.](image)

[Sze11, p. 229]
Comparison of Different Descriptors & Matchings

c) nearest neighbor distance ratio:

![Graph showing the comparison of different matchings strategies](image.png)

[Szel11, p. 229]
Mutual Nearest Neighbors

- match interest points if they **mutually** are nearest neighbors

\[ p \sim q \iff \begin{align*}
    &i) \quad q = \arg \min_{q \in Q} d(p, q) \quad \text{and} \\
    &\quad \quad \quad \quad \quad \quad \quad \quad \quad ii) \quad p = \arg \min_{p \in P} d(p, q)
    \end{align*} \]

- also for more than two views \( P_1, P_2, \ldots, P_V \) (called **closed chains**)

\((p_1, p_2, \ldots, p_V)\) corresponding tuple

\[ \iff \quad \begin{align*}
    &i) \quad p_{v+1} = \arg \min_{q \in P_{v+1}} d(p_v, q), \quad v = 1, \ldots, V - 1 \quad \text{and} \\
    &\quad \quad \quad \quad \quad \quad \quad \quad \quad ii) \quad p_1 = \arg \min_{q \in P_V} d(p_1, q)
    \end{align*} \]
Outline

1. Smoothing, Image Derivatives, Convolutions

2. Edges, Corners, and Interest Points

3. Image Patch Descriptors

4. Interest Point Matching

5. A Simple Application: Image Stitching
Image Stitching

- join several images depicting overlapping parts of the same real scene to one large image
Image Stitching

- join several images depicting overlapping parts of the same real scene to one large image

- algorithm:
  1. detect interest points in all images and extract their descriptors
  2. match interest points between every two images
  3. form a tree linking the best matching image pairs
  4. estimate a similarity transform between each two such images
  5. transform all images to joint coordinates
  6. average overlapping image regions

also called panography
Image Stitching

- join several images depicting overlapping parts of the same real scene to one large image

- algorithm:
  1. detect interest points in all images and extract their descriptors
  2. match interest points between every two images
  3. form a tree linking the best matching image pairs
  4. estimate a similarity transform between each two such images
  5. transform all images to joint coordinates
  6. average overlapping image regions

- also called **panography**
Image Stitching / Example

Figure 6.3: A simple panograph consisting of 3 images automatically aligned with a translational model and then averaged together.

6.1.2 Application: Panography

One of the simplest (and most fun) applications of image alignment is a special form of image stitching called panography. In a panograph, images are translated and optionally rotated and scaled before being blended with simple averaging (Figure 6.3). This process mimics the photographic collages created by artist David Hockney, although his compositions use an opaque overlay model, being created out of regular photographs.

In most of the examples seen on the Web, the images are aligned by hand for best artistic effect. However, it is also possible to use feature matching and alignment techniques to perform the registration automatically (Nomura et al. 2007, Zelnik-Manor and Perona 2007).

Consider a simple translational model. We want all the corresponding features in different images to line up as best as possible. Let $t_j$ be the location of the $j$th image coordinate frame in the global composite frame, and $x_{ij}$ be the location of the $i$th matched feature in the $j$th image. In order to align the images, we wish to minimize the least squares error

$$E_{PLS} = \sum_{ij} \| t_j + x_{ij} - x_i \|_2,$$

(6.12)

where $x_i$ is the consensus (average) position of feature $i$ in the global coordinate frame. (An alternative approach is to register each pair of overlapping images separately, and to then compute a consensus location for each frame—see Exercise 6.2.)
Image Stitching / Different Transforms

(a) translation [2 dof]  (b) affine [6 dof]  (c) perspective [8 dof]  (d) 3D rotation [3+ dof]

[Sze11, p. 425]
Figure 6.11: Four images taken with a hand-held camera registered using a 3D rotation motion model (Szeliski and Shum 1997). Notice how the homographies, rather than being arbitrary, have a well defined keystone shape whose width increases away from the origin, which is due to the interaction of the rotation matrix and the finite focal length in the calibration matrix.

(Sze11, p. 330)
Summary

- Small intensity fluctuations can be damped by smoothing, intensity changes can be captured by image derivatives, both being convolutions.

- Interest points are found as maxima of an interestingness measure,
  - gradient magnitude, Laplacian of Gaussian (LoG), Different of two Gaussians (DoG)
  - Harris corners:
    - large eigenvalues of the Hessian
    - can be approximated efficiently: $\det H - \alpha (\text{trace} H)^2$
  - SIFT:
    - detected interest points at different scale
    - several further tweaks

- non-maximum suppressions: ignore large values in the vicinity of a maximum
Summary (2/3)

- Interest points are characterized by local image information (descriptors)

- Descriptors often describe several patches (blocks/cells)

- Patches are described by histograms

- Histograms usually do not count pixel intensities, but gradient directions

- Descriptors sometimes
  - align patches with the orientation of the gradient at the interest point
  - weight gradient directions by their
    - gradient magnitude and/or
    - distance of the location to the interest point

- Common descriptors:
  - SIFT descriptors, Histogram of Gradients (HoG)
Summary (3/3)

- Whole images can be described two ways:
  - by the descriptors on a fixed grid of “interest points”
  - by the cluster frequencies of descriptors of variably located interest points

Both is useful, e.g. for image classification.

- Interest points are matched by their descriptors
  - for geometric tasks: also by their positions

- To match interest points, nearest neighbors are used
  - with a maximal distance threshold to avoid wrong matches
    - e.g. of points occluded in one view
  - Nearest Neighbor Distance Ratio
  - mutual nearest neighbors, closed chains in multiple views.

- Corresponding points can be used for
  - image stitching
  - SLAM, camera auto-calibration, ...
Further Readings

- Interest points and patch descriptors: [Pri12, ch. 13], [Sze11, ch. 4].
- Image stitching: [Sze11, ch. 9].
References

David G. Lowe.
Distinctive image features from scale-invariant keypoints.

Simon JD Prince.
*Computer vision: models, learning, and inference*.

Richard Szeliski.
*Computer vision: algorithms and applications*.