Outline

Outline

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**

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

- **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

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

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

Image Derivatives

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

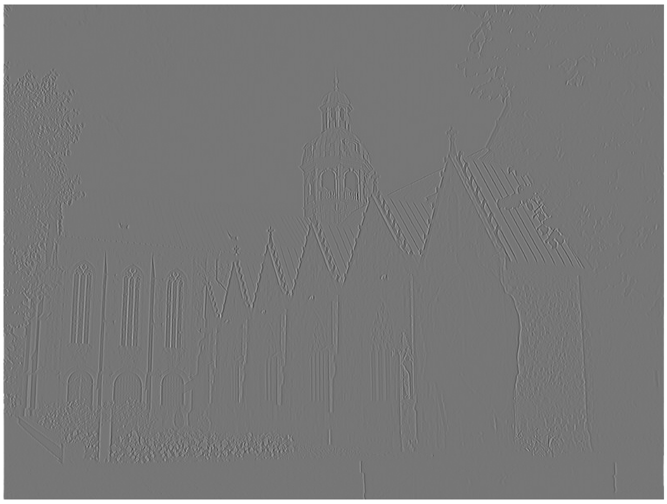
$$I_X(x, y) := I(x, y) - I(x-1, y)$$
$$I_Y(x, y) := I(x, y) - I(x, y-1)$$

or symmetric

$$I_X(x, y) := 2I(x, y) - I(x-1, y) - I(x+1, y)$$
$$I_Y(x, y) := 2I(x, y) - I(x, y-1) - (x, y-2)$$

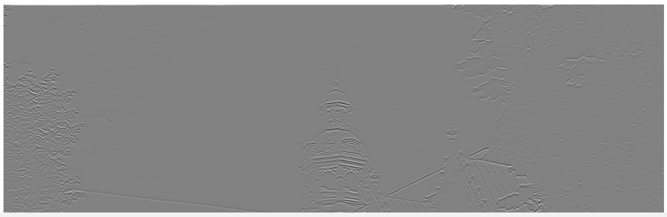
Image Derivatives / Example

derivative in x-direction:



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:

$$(w * I)(x, y) := \sum_{x', y'} w(x - x', y - y') I(x', y')$$

- Examples:

$$I_{\text{smooth}} = w * I$$

$$I_X(x, y) := I(x, y) - I(x-1, y) = \begin{pmatrix} 1 & -1 \end{pmatrix} * I$$

$$I_Y(x, y) := I(x, y) - I(x, y-1) = \begin{pmatrix} 1 \\ -1 \end{pmatrix} * I$$

$$\text{or } I_X(x, y) := 2I(x, y) - I(x-1, y) - I(x+1, y) = \begin{pmatrix} -1 & 2 & -1 \end{pmatrix} *$$

$$I_Y(x, y) := 2I(x, y) - I(x, y-1) - (x, y+1) = \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix} * I$$

Convolutions / Associativity

- Convolutions are associative:

$$I * (J * K) = (I * J) * K$$

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

$$\begin{pmatrix} -1 & 2 & -1 \end{pmatrix} * w = \begin{pmatrix} -0.007 & 0.002 & 0.017 & 0.002 & -0.007 \\ -0.033 & 0.008 & 0.077 & 0.008 & -0.033 \\ -0.054 & 0.077 & 0.128 & 0.077 & -0.054 \\ -0.033 & 0.008 & 0.077 & 0.008 & -0.033 \\ -0.007 & 0.002 & 0.017 & 0.002 & -0.007 \end{pmatrix}$$

Outline

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: $[N] := \{1, \ldots, N\}$.

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

gradient magnitude:

original (grayscale):

overlay with 500 interest points:

Laplacian of Gaussian and Difference of Gaussian

Further simple interestingness measures:

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

$$i(x,y) = (((D_X * D_X + D_Y * D_Y) * G) * I)(x,y)$$

  - uses second order information

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

$$i(x,y) = ((G_{\sigma_1} - G_{\sigma_2}) * I)(x,y), \quad \sigma_1 \neq \sigma_2$$

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

$$((D_X * D_X + D_Y * D_Y) * G_\sigma) * I \approx \frac{\sigma}{\Delta\sigma}((G_{\sigma+\Delta\sigma} - G_{\sigma-\Delta\sigma}) * I)$$

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.,

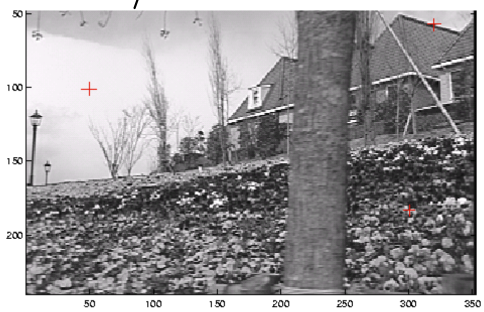
$$w(x, y) := \begin{cases} 1, & \text{if } |x - x_0| < 3 \text{ and } |y - y_0| < 3 \\ 0, & \text{else} \end{cases}$$

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

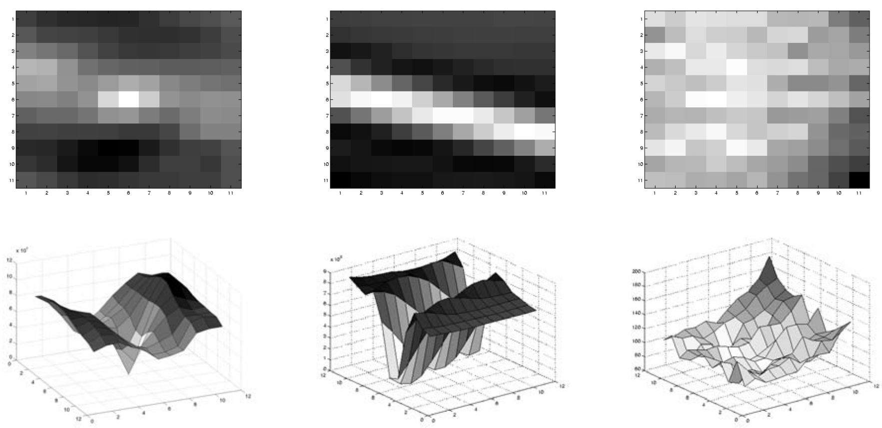
- A point is easy to identify, if its minimum in the **autocorrelation surface** is pronounced:

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

Harris Corner Detector / Autocorrelation Surface



(a)



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:

$$\begin{aligned} H(0,0;w) &\approx 2\sum_{x,y} w(x,y) \nabla I|_{(x,y)} \nabla I|_{(x,y)}^T, \quad \text{for } \frac{\partial^2 I}{\partial^2(x,y)} := 0 \\ &= 2w * \begin{pmatrix} (I_X)^2 & I_X I_Y \\ I_X I_Y & (I_Y)^2 \end{pmatrix}, \end{aligned}$$

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

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

Note: $I * J(x,y) := \sum_{x',y'} I(x-x', y-y')J(x',y')$ **convolution** of two images.

Harris Corner Detector

use SVD to assess steepness

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

and define interestingness measure:

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

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

$$i_{\text{Triggs}}(x,y) := \sigma_2 - \alpha\sigma_1, \qquad \alpha := 0.05$$

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

- the larger $\sigma_{1:2}$, the steeper the autocorrelation surface E.
- Harris and Brown avoid computing σ_1, σ_2 explicitly (which requires computing a square root).

Harris Corner Detector / Algorithm

procedure INTERESTPOINTS-HARRIS($I \in \mathbb{R}^{N \times M}$; $w \in \mathbb{R}^{-K:K \times -L:L}$, $\alpha \in \mathbb{R}$)

$I_X := D_X * I$

$I_Y := D_Y * I$

$I_X^2 := I_X \cdot I_X$

$I_Y^2 := I_Y \cdot I_Y$

$I_X I_Y := I_X \cdot I_Y$

$A := w * I_X^2$ ▷ compute $H(x,y) = \begin{pmatrix} A(x,y) & C(x,y) \\ C(x,y) & B(x,y) \end{pmatrix}$

$B := w * I_Y^2$

$C := w * I_X I_Y$

$i := A \cdot B - C \cdot C - \alpha(A+B) \cdot (A+B)$

return i

- 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)

a) original, b) Harris corners, c) DoG interest points

[Sze11, p. 213]

Interest Points at Different Scales (SIFT Detector)

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

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

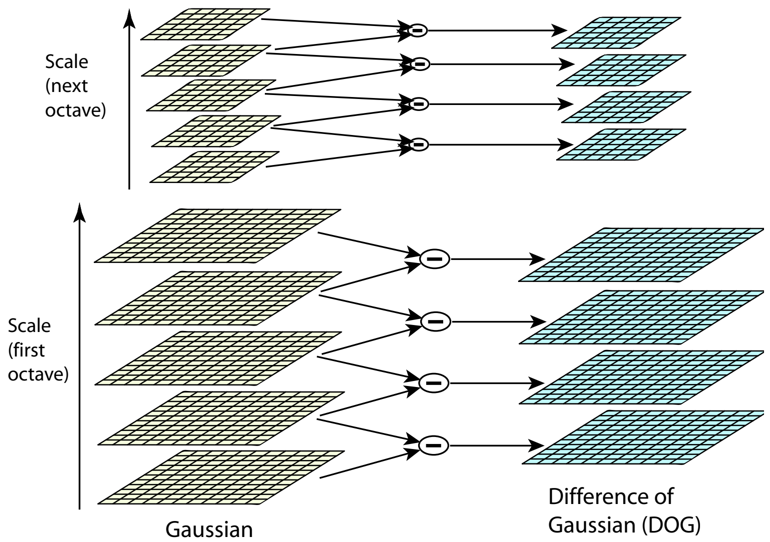
where

$$\sigma_1 > \sigma_2 > \cdots > \sigma_S$$

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 ocatve)

Interest Points at Different Scales (SIFT Detector)



(a)

[Sze11, p. 216]

Non-Maximum Suppression

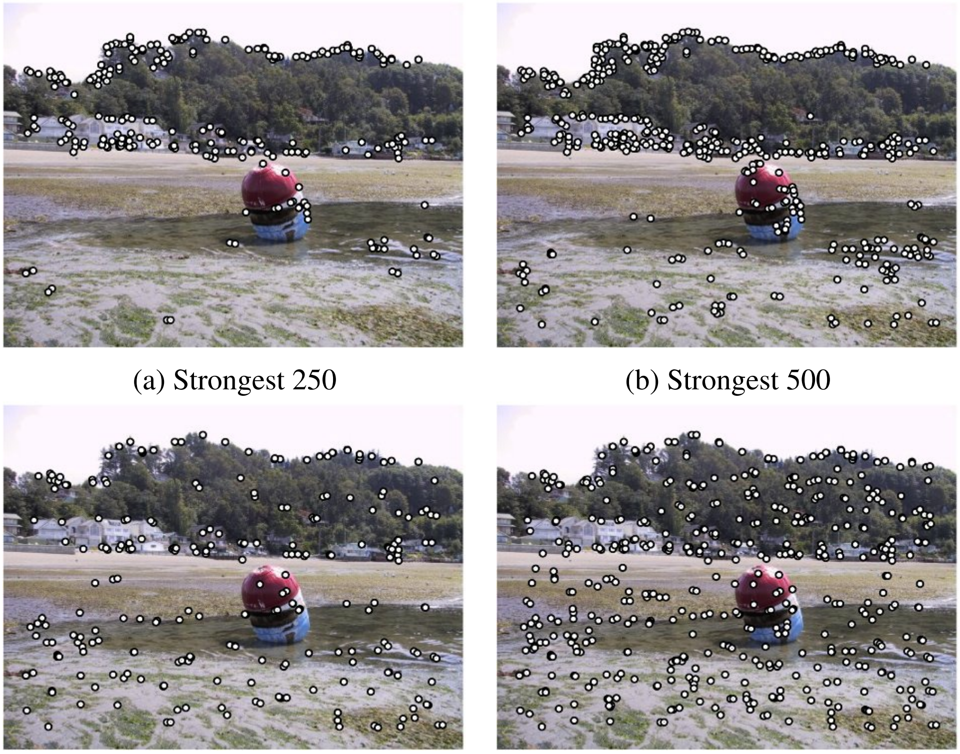
- Often neighbors of interest points have similar high interestingness, yielding redundant close-by interest points.
- Keep only interest points that are **local maxima** in their neighborhood:

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

with **neighborhood**

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

Non-Maximum Suppression / Example



(a) Strongest 250 (b) Strongest 500

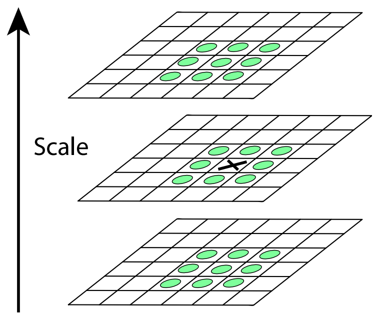
(c) ANMS 250, $r = 24$ (d) ANMS 500, $r = 16$

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

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
- 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**

Histograms / Intensities

- represent interest point (x, y) by its B-dimensional **intensity histogram features** $\phi(x, y)$:

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

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

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

for intensities $I(x, y)$ in range $[0, I_{\max}]$.

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

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

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

- sometimes called **trilinear counting**.

Histograms / Gradient Directions

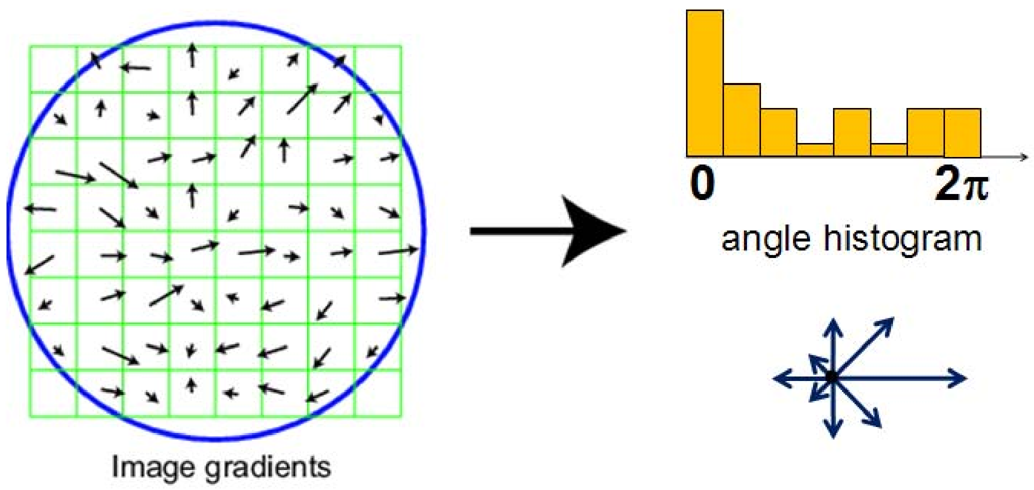
- represent interest point (x, y) by its B-dimensional **gradient directions histogram features** $\phi(x, y)$:

$$\begin{aligned}
\phi(x,y)_b :=& |\{(x',y') \in \mathcal{N}(x,y) \mid d(x',y') \in \text{bin}_b\}|, \quad b = 0, \dots, B-1 \\
d(x,y) :=& \tan^{-1}((D_Y * I)(x,y)/(D_X * I)(x,y)) \\
\text{bin}_b :=& [\frac{b}{B}2\pi, \frac{b+1}{B}2\pi[
\end{aligned}$$

- variant: weight gradients by their magnitude:

$$\phi(x,y)_b := \sum_{(x',y')\in\mathcal{N}(x,y), d(x',y')\in \text{bin}_b} (D_X * I)(x',y')^2 + (D_Y * I)(x',y')^2$$

Histograms / Gradients / Example



[Sze11, p. 217]

Block Descriptors

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

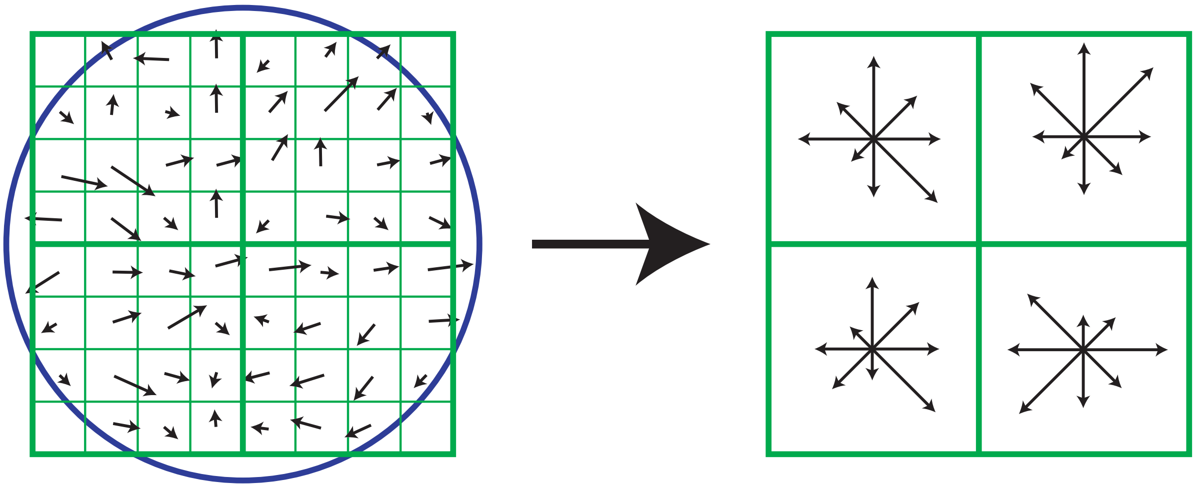
$$\phi(x,y) := \bigoplus_{(x',y')\in\mathcal{C}(x,y)} \phi'(x',y')$$

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

- 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



[Low04, p. 15]

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):

$$\psi := -d(x, y)$$

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

$$\begin{aligned} I_{\text{bi}}(x, y) :=& (1-(x-\lfloor x\rfloor))(1-(y-\lfloor y\rfloor)) && I(\lfloor x\rfloor, \lfloor y\rfloor) \\ &+ (x-\lfloor x\rfloor)(1-(y-\lfloor y\rfloor)) && I(\lceil x\rceil, \lfloor y\rfloor) \\ &+ (1-(x-\lfloor x\rfloor))(y-\lfloor y\rfloor) && I(\lfloor x\rfloor, \lceil y\rceil) \\ &+ (x-\lfloor x\rfloor)(y-\lfloor y\rfloor) && I(\lceil x\rceil, \lceil y\rceil) \end{aligned}$$

(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
  - 16×16, partitioned into 16 blocks a 4×4
- block features:
  - gradient directions
  - weighted by a Gaussian of the distance to the interest point
- block feature aggregation:
  - smoothly counted histograms
  - 8 bins
- $\rightsquigarrow$ feature vector $\phi \in \mathbb{R}^{128}$
- normalization in 3 steps:

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

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)**

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)

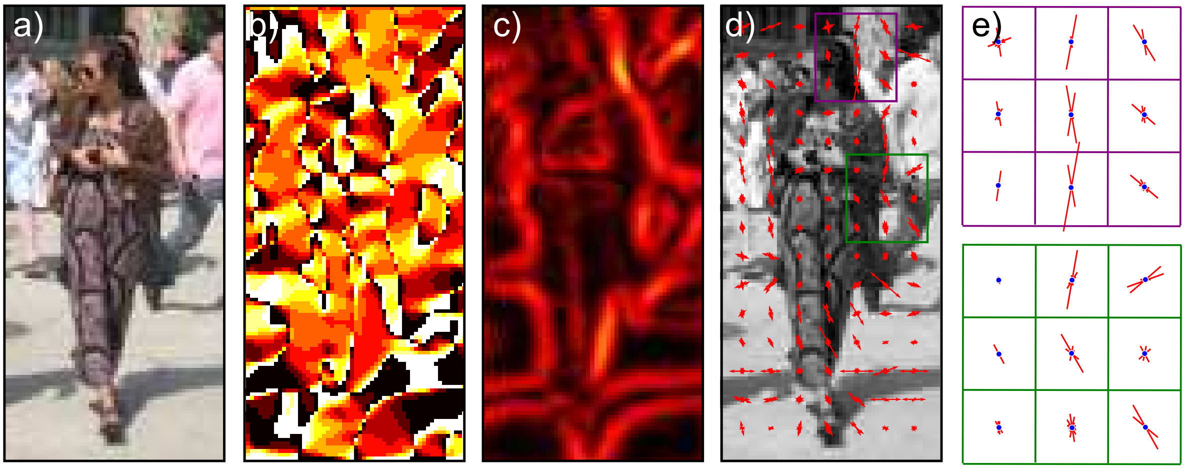


Figure 13.17 HOG descriptor. a) Original image. b) Gradient orientation, quantized into nine bins from 0 to 180^o. c) Gradient magnitude. d) Cell descriptors are 9D orientation histograms that are computed within 6×6 pixel regions. e) Block descriptors are computed by concatenating 3×3 blocks of cell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

[Pri12, p. 343]

Outline

Settings, Assumptions, Distances

Two settings:

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

$$d(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}) := d'(\phi(x_1, y_1), \phi(x_2, y_2)) = ||\phi(x_1, y_1) - \phi(x_2, y_2)||_2$$

- 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.,

$$\begin{aligned} d(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}) :=& \alpha d'(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}) + \beta d'(\phi(x_1, y_1), \phi(x_2, y_2)) \\ =& \alpha || \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} - \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} ||_2 + \beta ||\phi(x_1, y_1) - \phi(x_2, y_2)||_2 \end{aligned}$$

Simple methods

To match two sets P and Q of interest points:

- match interest points by **distance threshold**

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

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

- match interest points by **nearest neighbor**

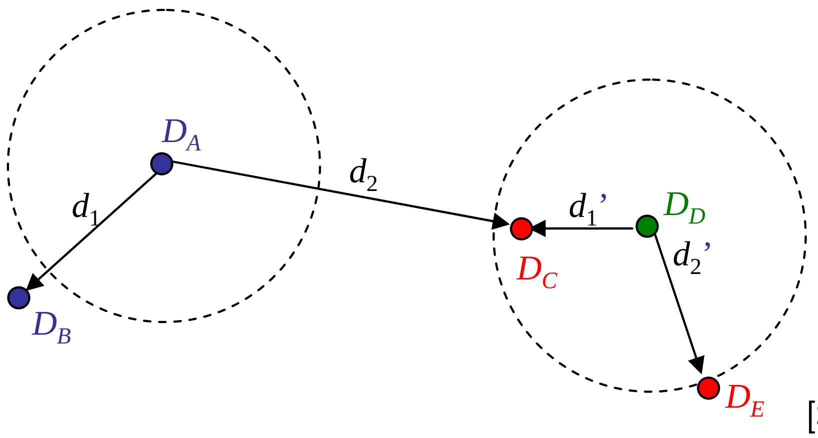
$$p \sim q :\Leftrightarrow q = \arg\min_{q \in Q} d(p, q)$$

Nearest Neighbor Distance Ratio

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

$$p \sim q :\Leftrightarrow i)\ q = \arg\min_{q \in Q} d(p,q) \text{ and}$$

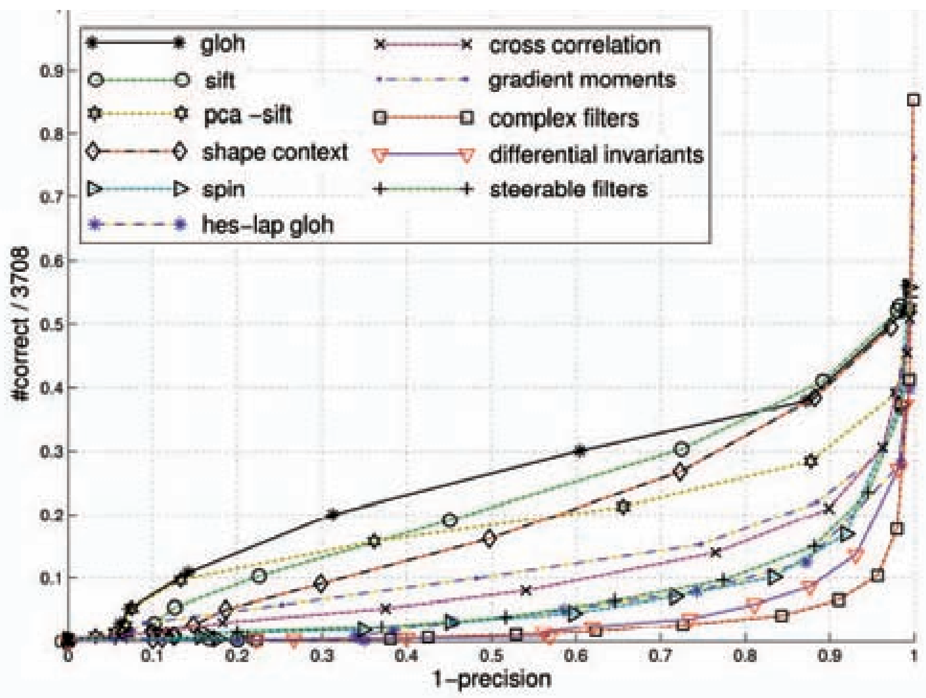
$$ii)\ \mathrm{NNDR}(p,q) := \frac{d(p,q)}{d(p,q')} < \mathrm{NNDR}_{\min}, \quad q' := \arg\min_{q' \in Q \setminus \{q\}} d(p,q')$$



[Sze11, p. 228]

Comparison of Different Descriptors & Matchings

a) fixed threshold:



[Sze11, p. 229]

b) nearest neighbor:

Mutual Nearest Neighbors

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

$$p \sim q :\Leftrightarrow i) \; q = \arg\min_{q \in Q} d(p, q) \text{ and}$$
$$ii) \; p = \arg\min_{p \in P} d(p, q)$$

- 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

$$:\Leftrightarrow \quad i) \; p_{v+1} = \arg\min_{q \in P_{v+1}} d(p_v, q), \quad v = 1, \ldots, V-1 \text{ an}$$
$$ii) \; p_1 = \arg\min_{q \in P_V} d(p_1, q)$$

Outline

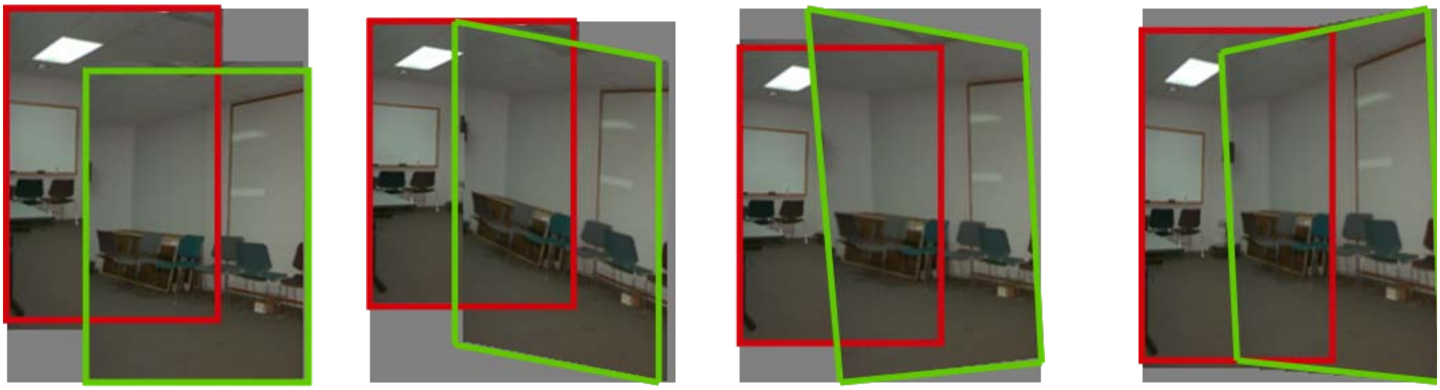
Image Stitching

- 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

[Sze11, p. 312]

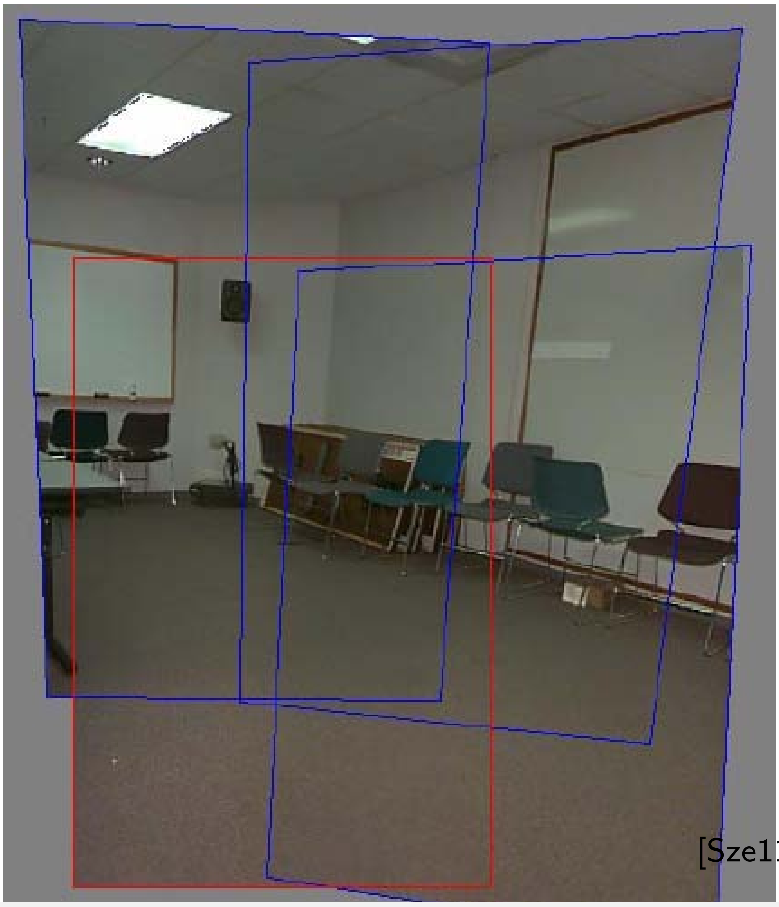
Image Stitching / Different Transforms



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

[Sze11, p. 425]

Image Stitching / Example



[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 positons

- 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.
International journal of computer vision, 60(2):91–110, 2004.

Simon JD Prince.
Computer vision: models, learning, and inference.
Cambridge University Press, 2012.

Richard Szeliski.
Computer vision: algorithms and applications.
Springer Science & Business Media, 2011.