AIMS Computer Vision
Lecture 4.2: Co-variant detectors
HT 2018
Andrea Vedaldi

For slides and up-to-date information:
http://www.robots.ox.ac.uk/~vedaldi/teach.html

Local features: contours

Local features: regions

Local features: interest points
Selecting points in a space co-variant manner

Goal: detect the same interest points, regardless of the translation

Suppose that an oracle shows you a template of the point
Detection amounts to searching points that match the template
In the second image, the same location are selected:
this detector is covariant to translations

Now suppose that the oracle provides two templates
▶ Which one would you pick?

Selecting points in a space co-variant manner

This patch matches at a set of discrete locations.
**Suitable** to detect a sparse set of interest points.

This patch matches entire rows of the image.
**Cannot** be used to define interest points.

The aperture problem

An image patch that does not contain sufficient structure (e.g. a uniform patch or an edge) does not allow fixing two components of the translation

Anisotropic structure: good interest point
Isotropic structure: bad interest point
Good patches to match

Pick a patch at a point \((tx, ty)\) and compare it to a patch slightly translated

The patch can be localized in \(x\) and \(y\) \(\iff E\) changes rapidly in \(x\) and \(y\)

\[
E(\delta tx, \delta ty) = \sum_{x,y \in W} [I(x + \delta tx, y + \delta ty) - I(x + tx, y + t + y)]^2
\]

Harris' cornerness function is defined as

\[
Harris(x, y) = \det M(x, y) - k(\text{tr } M(x, y))^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2
\]

Intuition

- the first term is large if both \(I I\) are
- the second term further attenuates the case when only one eigenvalue is large

The structure tensor \(M(x,y)\) is the second order approximation of \(E(\delta tx, \delta ty)\)

\[
E(\delta tx, \delta ty) = \sum_{x,y \in W} [I(x + \delta tx, y + \delta ty) - I(x + tx, y + t + y)]^2
\]

\[
\approx [\delta tx, \delta ty] \left( \sum_{x-tx, y-ty \in W} \left[ \frac{\partial I(x, y)}{\partial x} I(x, y) \right] \left[ \frac{\partial I(x, y)}{\partial y} I(x, y) \right] \right) \left[ \begin{array}{c} \delta tx \\ \delta ty \end{array} \right]
\]

\(E\) varies rapidly in \(x\) and \(y\) if \(M\) eigenvalues are both large; specifically

- two large eigenvalues \(\implies\) interest point
- one large eigenvalue \(\implies\) edge
- no large eigenvalue \(\implies\) uniform region

Harris cornerness (and similar measures) can be localised in space

- However, small translations of the same patch are also be localised (with a corresponding shift)

The solution is to run **non-maxima suppression**

- For each small neighbour of the image, pick the maximum response

Harris detector
**Scale and space covariance: blobs**

- [Courtesy Svetlana Lazebnik]

**Edge detection**

- [Courtesy S. Seitz]

**Edge detection, take 2**

- [Courtesy S. Seitz]

**From two edges to a “top hat” (blob)**

- [Courtesy S. Seitz]

**Spatial selection:** The magnitude of the Laplacian response will achieve a maximum at the centre of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob.
Scale-invariant feature detection

**Goal:** independently detect corresponding regions in *scaled* versions of the same image

Needs a selection mechanism that is co-variant with image rescaling

Scale selection

Search **characteristic scale** of a blob by

- convolving it with Laplacians at several scales
- looking for the maximum response

However, Laplacian response *decays as scale increases*:

Why does this happen?

Scale normalization

The response of a *derivative of a Gaussian filter* to a *perfect step edge* decreases as $1/\sigma$

Compensating

- Gaussian derivative: $\Rightarrow$ multiplying the filter by $\sigma$
- Laplacian (Gaussian second derivative) $\Rightarrow$ multiply by $\sigma^2$
Effect of scale normalization after applying a Laplacian of increasing $\sigma$ input signal with normalised Laplacian maximum

Laplacian of Gaussian: A circularly symmetric operator suitable for blob detection in 2D:

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Requires proper normalization similarly to 1D case

The 2D Laplacian is given by

$$\nabla^2_{\text{norm}} g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) \propto (x^2 + y^2 - 2\sigma^2)e^{-\frac{x^2+y^2}{2\sigma^2}}$$

For a binary circle of radius $r$, the Laplacian achieves a maximum at $\sigma = r / \sqrt{2}$
Characteristic scale

Scale of an image that produces a peak of the Laplacian response:

\[ \text{Characteristic scale} \]


Scale selection

The characteristic scale is **scale co-variant**

Relation between characteristic scales: \( s \times s_1 = s_2 \)

LoG detector (SIFT)

The normalized Laplacian of Gaussian response \( F(x,y,s) \) can be interpreted as a "blobbiness" measure

\[ \text{LoG}(x, y, \sigma) = (I \ast \nabla_{\text{norm}} g_\sigma)(x, y, \sigma) \]

Similar to Harris, run **non-maxima suppression in this volume** to detect blob in scale and space

Harris-Laplace detector

Combine **scale-selection** with a **spatial detector** run at multiple resolution to obtain a detector which is **co-variant to translation and scale**

Harris points computed at multiple scales by resizing the image

Scale selected points at maximum of Laplacian
**2D transformation models**

**Similarity**
(translation, scale, rotation)

**Affine**
(anisotropic scaling, skew)

**Projective**
(homography, perspective)

---

**Viewpoint covariant detection**

**Characteristic scales** (size of region)
- Lindeberg and Garding ECCV 1994
- Lowe ICCV 1999
- Mikolajczyk and Schmid ICCV 2001

**Affine covariance** (shape of region)
- Baumberg CVPR 2000
- Matas et al BMVC 2002 ← maximally stable regions
- Mikolajczyk and Schmid ECCV 2002
- Schaffalitzky and Zisserman ECCV 2002
- Tuytelaars and Van Gool BMVC 2000
- Mikolajczyk et al., IJCV 2005

---

**Example: Harris-affine**

View 1

View 2 (a)

View 2 (b)

View 2 (c)

Not the same region!

---

**Example of affine covariant regions**

**Maximally Stable Regions** (MSR)
- Segment using watershed algorithm, and track connected components as threshold value varies
- An MSR is detected when the area of the component is stationary
- See Matas et al. BMVC 2002

(first image)
(second image)
Maximally stable regions

Example: Maximally stable regions

Maximally stable: The “best” threshold is the one for which the region changes the least.

Shape-adaptation

Similar to scale-selection, this method can be combined with most detectors

The idea is to look for a linear transformation that makes the patch look “isotropic”

▶ “isotropic” patch = the eigenvalues of the structure tensor $M$ are equal

$$M = \sum_{x,y} \left[ \frac{\partial_x I(x,y)}{\partial_y I(x,y)} \right] \left[ \frac{\partial_y I(x,y)}{\partial_x I(x,y)} \right]$$

Residual rotation

▶ Both MSER and shape-adaptation are based on making an ellipse round

▶ This still leaves the rotation of the patch undetermined
Estimation of the dominant orientation
- extract gradient orientation
- histogram over gradient orientation
- peak in this histogram
- Rotate patch in dominant direction

Fixing the rotation

Example of affine covariant regions
1000+ regions per image
- Shape adapted regions
- Maximally stable regions

A region's size and shape are not fixed
They automatically adapt to the image intensity to cover the same physical surface
I.e. they correspond to the same physical surface region

Co-variant detection, invariant descriptor
- Extract elliptical viewpoint covariant regions
  - Shape Adapted regions
  - Maximally Stable regions

Normalization 1: map ellipse to circle
Normalization 2: orientate by dominant direction

Represent each region by SIFT descriptor (128-vector) [Lowe 1999]
- see Mikolajczyk and Schmid CVPR 2003 for a comparison of descriptors

Viewpoint invariant description

Harris-affine
Descriptors – SIFT [Lowe’99]

SIFT is a distribution of the gradient over an image patch
- 4x4 location grid and 8 orientations (128 dimensions)
- very good performance in image matching
  [Mikolaczyk and Schmid’03]

Image patch → Gradient → 3D histogram

Extract affine regions → Normalize regions → Eliminate rotation → Compute appearance descriptors → Match