Similarity and affine invariant point detectors and descriptors

Frédéric SUR
LORIA, École des Mines de Nancy & INRIA, France
sur@loria.fr

with ASIFT material from
Jean-Michel MOREL
CMLA, École Normale Supérieure de Cachan, France
Guoshen YU
CMAP, École Polytechnique, France

CVPR11 tutorial - Tools and Methods for Image Registration - June 24, 2011
Feature extraction in images (especially Interest Point detection) is the very first step of many Computer Vision applications, e.g.:

- photography stitching,
- object recognition,
- stereovision,
- pose estimation,
- structure from motion,
- robot localization
- ...

Second step: define correspondences between Interest Points (i.e. pairs of IP which are images of the same physical 3D point).
Why Interest *Points*?

Features = **points** instead of edges/lines.

Attneave 1954

"Information is concentrated along contours and is further concentrated at those points on a contour at which its direction changes most rapidly."
Why Interest *Points*?

Features = **points** instead of edges/lines.

Robustness to the *aperture problem*.

* Local method (vs global method): robust to occlusions/clutter and to local deformations.
Interest Points and invariant descriptors

**Feature** = interest point + descriptor of a local patch.


→ local descriptor to ease IP matching.

**Feature** = interest point + descriptor of a local patch.

source: http://www.cs.ubc.ca/~lowe/keypoints/ + Vedaldi’s VLFEAT.

→ local descriptor to ease IP matching.

Desirable properties

IP and descriptor contents should be “invariant” to:

- **illumination change**
- and to some **geometric transformations**.

→ Which ones?

i.e. Which geometric transformations are likely to map the query image to the test image?
Outline

1. The **two-view geometry**
   → requirements on geometric invariance of IP/descriptors.

2. Invariant **Interest Points**
   → How to define an *Interest Point*?

3. Invariant **Descriptors**
   → How to describe the patch around each IP?

4. **Matching** IP: limitations

5. A new way to attain affine invariance: **viewpoint simulation**
   → Morel and Yu’s ASIFT.
Viewpoint invariance?

Invariance to central projection in the pinhole camera.

→ **unreachable** when the structure of the underlying object is unknown (problems e.g. with self-occlusions.)
The pinhole camera model

\[ P \]

- \( C \): optical center
- \( \mathcal{P} \): image plane
- \( f \): focal length

\[ d(C, \mathcal{P}) \]

\[
\begin{pmatrix}
    u \\
    v \\
    1
\end{pmatrix}
\approx
\begin{pmatrix}
    f\alpha & f\gamma & u_0 \\
    0 & f\beta & v_0 \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    x \\
    y \\
    z
\end{pmatrix}
\]
Pinhole camera: a *homography* maps corresponding planes.
Invariance to homographies

Additional hypothesis: underlying object locally planar.

→ invariance to homographies
(a square is mapped to any quadrilateral).

Oxford’s House

Drawback: homography = 8 d.o.f.
The affine camera model

Pinhole camera:
\[
\begin{pmatrix}
    u \\
v \\
1
\end{pmatrix} \approx
\begin{pmatrix}
    \alpha & \gamma & u'_0 \\
0 & \beta & v'_0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
x \\
v \\
z/f
\end{pmatrix}
\]

Particular case: \( z/f = c \) constant (weak-perspective)
e.g. telephoto lens.

Affine camera:
\[
\begin{aligned}
u &= ax + by + u_0 \\
v &= cy + v_0
\end{aligned}
\]
Image of a plane: affine camera

Affine camera: an affine transformation maps corresponding planes.
Planar homography: 8 d.o.f.

→ 1st order approximate (or affine camera model):

**affine transformation** (6 d.o.f.)

(a square is mapped to any parallelogram.)

Do not forget: **local** patches.
Invariance to similarity transformations

Still simpler: **similarity** (4 d.o.f.: zoom + rotation + translation) (a square is mapped to any square.)
Wish-list

Interest Point in an image \( I \) (remember Attneave)

= a point which is exceptional from its neighbourhood.

We would like **Interest Points**...

- which are detected at the same location, whatever the transformation on the images;
- coming with a region whose description is invariant to the chosen group of transformations.

Transformations = **affine** or **similarity** transformations.
Interest point detection: Harris-Stephens

**Canonical Example:** (Harris-Stephens 1988)

\[ A(x, y) = \sum_{u,v} w_{x,y}(u, v) \begin{pmatrix} \partial_x I(u, v)^2 & \partial_x I(u, v) \partial_y I(u, v) \\ \partial_x I(u, v) \partial_y I(u, v) & \partial_y I(u, v)^2 \end{pmatrix} \]

\( A \) is an empirical covariance matrix of \( \nabla I \) localized at \((x, y)\).

→ interest point (corner) if \( A \) has two “large” eigenvalues.

Cornerness: \( C(x, y) = \det(A) - \kappa \text{Trace}(A)^2 \).

**Advantage:** invariant to rotations.

\( w = \) isotropic Gaussian kernel.

**Limitation:** not invariant to scale change. (\( w \) ? derivatives ?)

→ **Idea 1:** define the scale of an image.

→ **Idea 2:** define the characteristic scale of a point.
Interest point detection: Harris-Stephens

**Canonical Example:** (Harris-Stephens 1988)

\[ A(x, y) = \sum_{u,v} w_{x,y}(u, v) \begin{pmatrix} \partial_x I(u, v)^2 & \partial_x I(u, v)\partial_y I(u, v) \\ \partial_x I(u, v)\partial_y I(u, v) & \partial_y I(u, v)^2 \end{pmatrix} \]

\( A \) is an empirical covariance matrix of \( \nabla I \) localized at \((x, y)\).

\[ \rightarrow \text{interest point (corner) if } A \text{ has two "large" eigenvalues.} \]

Cornerness: \( C(x, y) = \det(A) - \kappa \text{Trace}(A)^2. \)

**Advantage:** invariant to rotations.
\( w = \) isotropic Gaussian kernel.

**Limitation:** not invariant to scale change. (\( w \) ? derivatives ?)

\[ \rightarrow \text{Idea 1: define the scale of an image.} \]

\[ \rightarrow \text{Idea 2: define the characteristic scale of a point.} \]
How to simulate scale changes?

**Scale-space theory**: scale change = convolution with Gaussian kernels:

\[ I_t = G_t \ast I \quad \text{where} \quad G_t(x) = \frac{1}{2\pi t} e^{-\frac{|x|^2}{2t}}. \]

Scale-space and scale change (Witkin 1983, Koenderink 1984)

\[ G_t \ast SI = S(G_{s^2t} \ast I) \]

where \( S = \) zoom change: \( SI(x) = I(sx) \)

Means: Gaussian convolution simulates scale change.
Normalized derivatives

**Question**: How to compute derivatives in scale-space?

**Nice property**: $\partial_x I_t = (\partial_x G_t) * I$.

**But**: Convolution does not simulate scale change for derivatives.
\[
\partial_x G_t * SI \neq S(\partial_x G_{s^2 t} * I)
\]

**Solution** (Lindeberg’90s):
replace $\partial_x G_t$ by the normalized derivative $t^{1/2} \partial_x G_t$
\[
\rightarrow \text{ then equality holds.}
\]
i.e. replace $\partial_x I_t$ by $\partial'_x I_t = t^{1/2} (\partial_x G_t) * I$,
$\partial_{xy} I_t$ by $\partial'_{xy} I_t = t (\partial_{xy} G_t) * I$

**Scale-space and scale change**

If $I$ (or any function of the derivatives) has an extremum in scale space at $(x_0, y_0, t)$, then $SI$ (or any function of the derivatives) has an extremum in scale-space at $(sx_0, sy_0, s^2 t)$. 
Normalized derivatives

**Question**: How to compute derivatives in scale-space?

**Nice property**: \( \partial_x l_t = (\partial_x G_t) * l \).

**But**: Convolution does not simulate scale change for derivatives. 
\[ \partial_x G_t * S l \neq S (\partial_x G_{s^2 t} * l) \]

**Solution** (Lindeberg’90s):
replace \( \partial_x G_t \) by the **normalized derivative** \( t^{1/2} \partial_x G_t \)
\[ \rightarrow \] then equality holds.
i.e. replace \( \partial_x l_t \) by \( \partial'_x l_t = t^{1/2} (\partial_x G_t) * l \),
\( \partial_{xy} l_t \) by \( \partial'_{xy} l_t = t (\partial_{xy} G_t) * l \)

**Scale-space and scale change**
If \( l \) (or any function of the derivatives) has an extremum in scale space at \((x_0, y_0, t)\), then \( S l \) (or any function of the derivatives) has an extremum in scale-space at \((s x_0, s y_0, s^2 t)\).
Automatic characteristic scale selection: example

Radius: \( r, 2r, 3r, 4r \)

Graphs of \( \sqrt{t} \mapsto \Delta'I_t(x, y) \) where \( x, y \) is the center of the circles.

\( \rightarrow \) Note the \( \sqrt{t} \) of the (sharp) maximum: \( r', 2r', 3r', 4r' \).
Scale-invariant Harris detector: (Mikolajczyk-Schmid 2004)
Use the normalized derivatives to define $A(x, y, \sqrt{t})$.

$\rightarrow$ scale-adapted cornerness $C(x, y, \sqrt{t})$

**IP+scale:** – track $(\hat{x}, \hat{y})$ maximizing $C$ across scales,
– characteristic scale $\sqrt{t}$ minimizes or maximizes $\Delta'I_t$. 

source: Tuytelaars - Mikolajczyk 2008
Another way of defining IP: the Hessian matrix

IP: strong gradient changes in two directions

= **blob detection**: interest point if “large values” of the Hessian matrix:

\[ H(x, y, t) = \begin{pmatrix} \partial'_{xx}I_t & \partial'_{xy}I_t \\ \partial'_{xy}I_t & \partial'_{yy}I_t \end{pmatrix} \]

How to quantify “large values”?

→ Det of Hessian: \( \text{DoH}(x, y, t) = \det(H(x, y, t)) \)
(used in Bay et al’s SURF)

→ Laplacian of Gaussian: \( \text{LoG}(x, y, t) = \text{Trace}(H(x, y, t)) \)

→ Difference of Gaussian: \( \text{DoG}(x, y, t) = \frac{2}{h} \left( I_{t+h}(x, y) - I_t(x, y) \right) \)
so:
\[ \text{DoG}(x, y, t) = \frac{2}{h} \left( G_{t+h} - G_t \right) * I(x, y) \]
(used in Lowe’s SIFT)

Motivation: \( 1/2 \cdot \Delta I_t(x, y) = \partial_t I_t(x, y) \) (heat diffusion equation)

+ **scale-selection**:

Interest Point if

\( (x, y, \sqrt{t}) = \text{argmax}_{x,y} \) and \( \text{argmax/min}_t f(x, y, \sqrt{t}) \)
Another way of defining IP: the Hessian matrix

IP: strong *gradient changes* in two directions

= **blob detection**: interest point if “large values” of the Hessian matrix:

\[
\mathbf{H}(x, y, t) = \begin{pmatrix}
\partial'_{xx}l_t & \partial'_{xy}l_t \\
\partial'_{xy}l_t & \partial'_{yy}l_t
\end{pmatrix}
\]

How to quantify “large values”?

→ Det of Hessian: \( \text{DoH}(x, y, t) = \det(\mathbf{H}(x, y, t)) \)

(used in Bay et al’s SURF)

→ Laplacian of Gaussian: \( \text{LoG}(x, y, t) = \text{Trace}(\mathbf{H}(x, y, t)) \)

→ Difference of Gaussian: \( \text{DoG}(x, y, t) = \frac{2}{h} (l_{t+h}(x, y) - l_t(x, y)) \)

so: \( \text{DoG}(x, y, t) = \frac{2}{h} (G_{t+h} - G_t) * I(x, y) \) (used in Lowe’s SIFT)

Motivation: \( 1/2 \cdot \Delta l_t(x, y) = \partial_t l_t(x, y) \) (heat diffusion equation)

+ **scale-selection**:

**Interest Point** if

\[
(x, y, \sqrt{t}) = \arg\max_{x, y} \text{ and } \arg\max_{t} f(x, y, \sqrt{t})
\]
Another way of defining IP: the Hessian matrix

IP: strong gradient changes in two directions
= **blob detection**: interest point if “large values” of the Hessian matrix:
\[
H(x, y, t) = \begin{pmatrix}
\partial'_{xx} l_t & \partial'_{xy} l_t \\
\partial'_{yx} l_t & \partial'_{yy} l_t
\end{pmatrix}
\]

How to quantify “large values”?
→ Det of Hessian: \( \text{DoH}(x, y, t) = \det(H(x, y, t)) \)
(used in Bay et al’s SURF)
→ Laplacian of Gaussian: \( \text{LoG}(x, y, t) = \text{Trace}(H(x, y, t)) \)
→ Difference of Gaussian: \( \text{DoG}(x, y, t) = \frac{2}{\partial^2} (l_{t+h}(x, y) - l_t(x, y)) \)
so: \( \text{DoG}(x, y, t) = \frac{2}{\partial^2} (G_{t+h} - G_t) * l(x, y) \)
(used in Lowe’s SIFT)

Motivation: \( 1/2 \cdot \Delta l_t(x, y) = \partial_t l_t(x, y) \)
(heat diffusion equation)

+ **scale-selection**:
**Interest Point** if
\[(x, y, \sqrt{t}) = \arg\max_{x, y} \text{ and } \arg\max/\min_t f(x, y, \sqrt{t}) \]
Adapting to affine invariance (outline)

**Idea**: Harris’ $A$ matrix covariant with affine transformations.

**Iterative algorithm**: Harris-Affine (Mikolajczyk-Schmid 2004)

1. Detect multiscale Harris-Laplace points
2. Warp patch so that $A$ is rectified into unit matrix
3. Back to step 1 on the warped image, until convergence.

source: Tuytelaars - Mikolajczyk 2008
Example: Harris-Affine

source: Tuytelaars - Mikolajczyk 2008
Another family: intensity based regions

Affine geometric normalization of contrasted regions.

Level Line Descriptors:
(Morel et al. 2000-2008)

MSER
(Matas et al. 2002)

See also IBR/EBR (Tuytelaars - Van Gool 2004)
**Summary**: We have defined a patch around an IP, covariant with a group of geometric transformations.

**Requirement**: Concise description, invariant to *contrast changes*.
Normalizing the photometry

**Idea 1:** Normalize the grey-levels by: \( \frac{I(x,y)-\mu}{\sigma} \)
(with: \( \mu \) average gray level in the patch & \( \sigma \): standard deviation)
Descriptor = a subset of normalized grey-levels from the patch.
\( \rightarrow \) invariant to affine contrast changes.

**Problem:** not robust to small drifts of the localization of the IP +
problem with quantization?

**Remark:** The gradient *direction* is invariant to contrast changes

\[ \nabla(g \circ I)(x,y) = g'(I(x,y)) \cdot \nabla I(x,y) \]

**Idea 2:** Descriptor = statistics over the *direction of the gradient.*
Normalizing the photometry

**Idea 1:** Normalize the grey-levels by: \( \frac{I(x, y) - \mu}{\sigma} \)
(with: \( \mu \) average gray level in the patch & \( \sigma \): standard deviation)
Descriptor = a subset of normalized grey-levels from the patch.
\( \rightarrow \) invariant to affine contrast changes.

**Problem:** not robust to small drifts of the localization of the IP +
problem with quantization?

**Remark:** The gradient *direction* is invariant to contrast changes

\[ \nabla (g \circ I)(x, y) = g'(I(x, y)) \cdot \nabla I(x, y) \]

**Idea 2:** Descriptor = statistics over the *direction of the gradient*. 
Canonical example: Lowe’s SIFT descriptors

**Patches** = square centered at IP and size proportional to the detected scale.

**Orientation** = dominant direction of the gradient.

Each gradient votes in a direction histogram.
Vote weighted by the norm of the gradient.
Global normalization → each SIFT descriptor is invariant to affine contrast changes.
**Summary**: Now, we have:

1. similarity or affine invariant interest points
2. a descriptor of a (covariant) patch around.

**Question**: How to build correspondences from two images?
A popular algorithm

**Point matching** between images 1 and 2:

1. Associate each point from image 1 to the nearest neighbour in image 2.
   (in the sense of a distance between descriptors)

2. RANSAC to keep only correspondences consistent with a realistic motion of the camera.
   (affine transformation, homography, or fundamental matrix)
Example (1)

SIFT
Lowe 1999-2004
22 matches
Example (1)

MSER
Matas et al. 2002
13 matches
Example (1)

Hessian Affine
Mikolajczyk and Schmid
2002
9 matches
Example (1)

Harris Affine
Mikolajczyk and Schmid
2002
5 matches
Example (1)

ASIFT
Morel and Yu 2009
123 matches
Example (2)

SIFT - 14 matches
Example (2)

MSER - 6 matches
Example (2)

Hessian Affine - 3 matches
Example (2)

Harris Affine - 0 match
Example (2)

Moreels-Perona 2007: “We also find that no detector/descriptor combination performs well with viewpoint changes of more than 25–30°”. (ASIFT was not tested)
CSI: Colorado Springs - Where is the skull?

Hans Holbein, *The Ambassadors*, 1533
The painting was probably hung up by a grand staircase.

The painting was likely to be seen from the side, slantwise.
A clue... and the solution!

The painting was probably hung up by a grand staircase.

→ The painting was likely to be seen from the side, slantwise.
A clue... and the solution!

The painting was probably hung up by a grand staircase.

→ The painting was likely to be seen from the side, slantwise.
A clue... and the solution!

The painting was probably hung up by a grand staircase.
→ The painting was likely to be seen from the side, slantwise.

Instead of normalizing all parameters, it is possible to simulate a part of them to make easier (e.g.) SIFT matching.
Parametrization of affine transformations

From Singular Value Decomposition:

\[ A = USV \]

where \( U \), \( V \) orthogonal matrices, \( S \) diagonal matrix.

Consequence:

\[ A = \lambda \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \]

**Remark 1**: already used in Lepetit-Fua 2006 to simulate views of a planar patch (exhaustive sampling).

**Remark 2**: Harris-Affine is a way to avoid exhaustive simulation.
With the affine camera model:

\[
A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = H_\lambda R_1(\psi)T_t R_2(\phi) = \lambda \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}
\]

- \( \phi \): *longitude* angle between optical axis and a fixed vertical plane.
- \( \theta = \arccos(1/t) \): *latitude* angle between optical axis and the normal to the image plane. **Tilt** \( t > 1 \iff \theta \in [0^\circ, 90^\circ] \).
- \( \psi \): rotation angle of camera around optical axis.
- \( \lambda \): *zoom* parameter.
Morel and Yu’s ASIFT

\[
A = \lambda \begin{bmatrix}
\cos \psi & -\sin \psi \\
\sin \psi & \cos \psi 
\end{bmatrix}
\begin{bmatrix}
t & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
\cos \phi & -\sin \phi \\
\sin \phi & \cos \phi 
\end{bmatrix}
\]

Morel & Yu: generate a discrete subset of every possible

\[
l_{t,\phi} = \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} R(\phi)(I)
\]

Yields:

Since SIFT is invariant to zoom (\(\lambda\)) + rotation (\(\psi\)):

\[
\{ \text{SIFT from every } l_{t,\phi} \} = \text{Affine SIFT}
\]

→ ASIFT is affine invariant.
Parameter sampling precision

**Remark 1:** cf scale sampling in scale-space based detectors.

**Remark 2:** – the larger $t$ (slanted view), the finer the discretization of $\phi$ and $t$.

– longitude angle $\phi \in [0, \pi)$ since:

$$R_1(\psi)T_tR_2(\phi + \pi) = R_1(\psi + \pi)T_tR_2(\phi).$$
The ASIFT algorithm

**Data**: two images $I$ and $I'$.

1. **Generate** $I_{t,\phi}$’s and $I'_{t',\phi'}$’s via the “dense” sampling of $t, \phi$.

2. **Extract SIFT features** from generated images.

3. **Match SIFT features**: for each pair from step 1, match each feature from $I_{t,\phi}$ to its nearest neighbour (NN) in $I'_{t',\phi'}$.

4. **Keep the five pairs** of simulated views yielding the largest set of correspondences.

6. Discard possible false correspondences: epipolar RANSAC.

**Output**: a set of corresponding Interest Points.

**Remark**: multi-resolution scheme in original ASIFT

$\rightarrow$ comparing two images is just $\simeq 2 - 3 \times$ longer than with SIFT.
**Experiment (1)**

Images by Matas et al.

Number of correct matches:

ASIFT (top)—254;
SIFT (middle)—10;
MSER (bottom)—22.
Parkings.
Number of correct matches:
ASIFT (top)—78;
SIFT (middle)—0;
MSER (bottom)—0.
Experiment (3)

image 1

image 2

(for reference)

ASIFT: 94 matches. (SIFT, Harris-Affine, Hessian-Affine, MSER fail)
## Conclusion

Broad classification of the mentioned IP:

<table>
<thead>
<tr>
<th>Invariance</th>
<th>Rotation</th>
<th>Similarity</th>
<th>$\simeq$ Affine</th>
<th>Affine</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Harris</td>
<td>SIFT, SURF</td>
<td>MSER, Harris-Affine</td>
<td>ASIFT</td>
</tr>
<tr>
<td></td>
<td>Laplace-Harris</td>
<td></td>
<td>LLD, IBR/EBR</td>
<td></td>
</tr>
</tbody>
</table>

Make your choice depending on the problem:

- nature of the images? (well contrasted shapes? planar objects?)
- small (e.g. video) or extreme viewpoint change?
- computational time?
Selected references (1)


**About scale-space and scale selection:**


Selected references (2)

**Interest Points and Descriptors:**


Intensity based descriptors:


Comprehensive comparisons/surveys:


Viewpoint Simulation:


Selected references (6)

**Softwares:**

- Lowe’s SIFT:
  http://www.cs.ubc.ca/spider/lowe/keypoints/
- Vedaldi’s VLFEAT (MSER, SIFT, etc.):
  http://www.vlfeat.org/
- Mikolajczyk et al.’s FeatureSpace (MSER, SIFT, Harris/Hessian-Affine and more):
  http://www.featurespace.org
- Matas et al.’s MSER:
  http://cmp.felk.cvut.cz/~wbsdemo/demo/
- Morel and Yu’s ASIFT on IPOL (code, demo, try your images):
  http://www.ipol.im/pub/algo/my_affine_sift/