Scale-space and its applications

Dmitry Chetverikov

Eötvös Lóránd University, Hungarian Academy of Sciences

SSIP 2015

23rd Summer School on Image Processing

9 July - 18 July 2015 • Szeged, Hungary

Scale-space and its applications

1. Scale-space
   - Scale-space and diffusion
2. Image features in scale-space
   - Scale selection
3. Affine-invariant features
   - Corner-like features
   - Blob-like features

Outline

1. Scale-space
   - Scale-space and diffusion
2. Image features in scale-space
   - Scale selection
3. Affine-invariant features
   - Corner-like features
   - Blob-like features

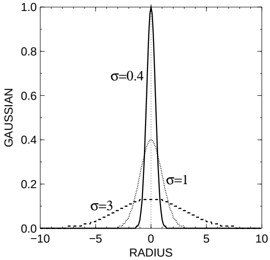
Scale-space

$$g(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) * f(\mathbf{x}) \qquad f, g : \text{input/output image}$$

- Gauss filter

$$G(\mathbf{x}, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|\mathbf{x}|^2}{2\sigma^2}\right)$$

- $|\mathbf{x}| = \sqrt{x^2 + y^2}$: distance from filter center



- Larger $\sigma \Rightarrow$ stronger smoothing, fewer details
- Scale-space $g(\mathbf{x}, \sigma)$
  - image sequence parameterised by scale σ
  - image representation with controllable degree of detail
  - Witkin (1983), Koenderink (1984), Lindeberg (1994), ...

Example of scale-space

$\sigma = 0$ (original image)

$\sigma = 1$

$\sigma = 2$

$\sigma = 4$

$\sigma = 8$

$\sigma = 16$

source: Wikipedia

Scale-space creation by diffusion

- Diffusion process: transfer of heat and matter
  - spatial differences decrease
  - matter concentration equalises
- Relation between scale-space and diffusion:
  - diffusion generates scale-space
- General diffusion equation:

$$\frac{\partial g}{\partial t} = \nabla \left(D \nabla g \right),$$

where $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$ is the gradient operator

  - $D = D(\mathbf{x}, t)$ is the diffusion coefficient, t time

Homogeneous diffusion

- diffusion coefficient D
  - does not depend on co-ordinates x, y
  - but can depend on time t
- Homogeneous diffusion equation:

$$\frac{\partial g}{\partial t} = D \Delta g,$$

$$\text{where} \quad \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad \text{is the Laplace operator}$$

- Solution of diffusion equation:

$$g(\mathbf{x}, t) = \frac{1}{2\pi\sigma^2(t)} \exp\left(-\frac{|\mathbf{x}|^2}{2\sigma^2(t)}\right) * g(\mathbf{x}, 0),$$

$$\text{where} \quad \sigma(t) = \sqrt{2Dt}$$

Gradual elimination of details

- Image details
  - lines, edges, corners, blobs
- As scale grows, **new details do not appear**
- Details disappear or merge
  $\rightarrow$ forming specific **tree structure**
- Information content of image gradually decreases

The minimum-maximum principle

- In principle, scale-space can be built by other filters, as well
- **Minimum-maximum principle**: basic theoretical requirement (axiom) for scale-space
  - local minima must not deepen
  - local maxima must not grow
- Gradual elimination of details in consequence of this principle
- Facilitates **structural analysis** of image
  - theoretical ground for analysis of details and their relations

Significance of the Gaussian filter

- Why to prefer this filter?
- Min-max principle: natural result of diffusion
  $\rightarrow$ valid for Gaussian filter, as well
- **Scale-invariance principle**: the other basic theoretical requirement
  - to be discussed later
- In continuous case, **only Gaussian filter** conforms with the two principles
  - in discrete case, only polynomial filter

Discrete scale-space 1/2

- Use simple parameterisation of Gaussian filter:

$$G(\mathbf{x}, \xi) = \frac{1}{2\pi\xi} \exp\left(-\frac{|\mathbf{x}|^2}{2\xi}\right)$$

- 1D diffusion equation

$$\frac{\partial g(x, \xi)}{\partial \xi} = \frac{\partial^2 g(x, \xi)}{\partial x^2}$$

- After discretisation, we obtain iterative solution

$$g_{n,\xi+1} = \Delta\xi g_{n+1,\xi} - (1 - 2\Delta\xi) g_{n,\xi} + \Delta\xi g_{n-1,\xi},$$

$$\text{where} \quad \begin{aligned} n+1 &\Rightarrow x + \Delta x, \\ \xi + 1 &\Rightarrow \xi + \Delta\xi \end{aligned}$$

Discrete scale-space 2/2

- The above iterative process satisfies the two basic conditions if

$$\Delta\xi \le \frac{1}{4} \qquad \text{(Lindeberg, 1994)}$$

- Usually, scale step $\Delta\xi = \frac{1}{4}$ is selected
- Then we have simple iterative solution

$$g_{n,\xi+1} = \frac{1}{4} g_{n+1,\xi} + \frac{1}{2} g_{n,\xi} + \frac{1}{4} g_{n-1,\xi}$$

  - i.e., application of filter $\frac{1}{4}[1, 2, 1]^T$

- Similar solution in 2D case

Outline

1 Scale-space
- Scale-space and diffusion

2 Image features in scale-space
- Scale selection

3 Affine-invariant features
- Corner-like features
- Blob-like features

Derivatives in scale-space

$$g_{x^m y^n} = (\partial x^m \partial y^n g) = (\partial x^m \partial y^n G)\, f$$

- Order of filtering and derivation is arbitrary
- **Gaussian derivatives**: derivatives of Gaussian filter

$$\partial x^m \partial y^n G(\mathbf{x}, \xi) \doteq G_{x^m y^n}(\mathbf{x}, \xi)$$

- Use rotational symmetry and separability of filter
- For example, Gaussian gradient vector:

$$\nabla G(\mathbf{x}) = \big(G(y) G_x(x), G(x) G_y(y)\big),$$

$$\text{where} \quad G_x(x) = -\frac{x}{\xi} G(x)$$

Edge detection in scale-space

- Edge detection by **gradient** operator $\nabla G = (G_x, G_y)$

$$|\nabla g| = \sqrt{g_x^2 + g_y^2} \qquad \text{edge magnitude}$$

  - search for locations of large $|\nabla g|$

- Edge detection by **Laplacian-of-Gaussian** (LoG) operator

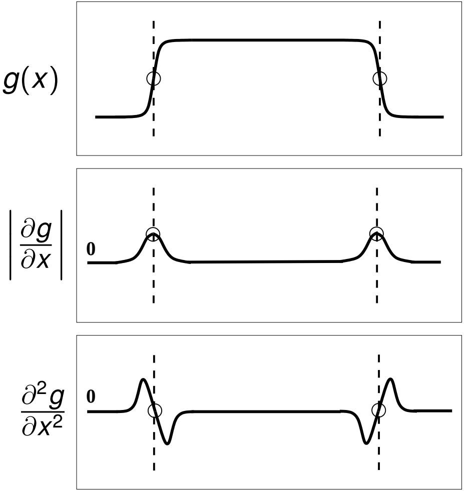
$$\Delta G = G_{xx} + G_{yy}$$

  - search for zero-crossings of Δg

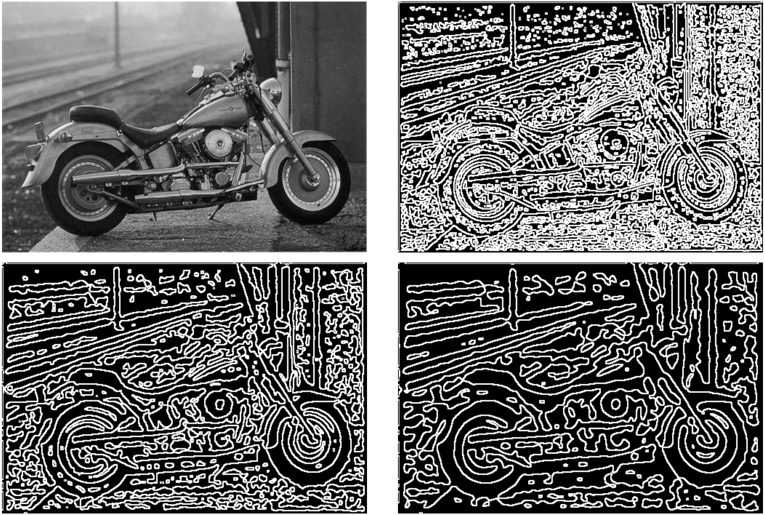
Relations between edges and image derivatives

- Signal
- Edges
  - maxima of abs value of first derivative

  or
  - zero-crossings of second derivative



Zero-crossing edge detection at decreasing detail

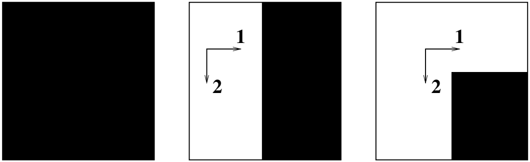


Corner detection in scale-space

- Corner detection with **local structure matrix** (tensor) M

$$M = \begin{bmatrix} g_x g_x & g_x g_y \\ g_x g_y & g_y g_y \end{bmatrix}$$

  - Search for locations where eigenvalues λ_1, λ_2 of matrix M are large



uniform image $\lambda_1 = \lambda_2 = 0$

ideal step edge $\lambda_1 > 0, \lambda_2 = 0$

ideal corner $\lambda_1 \geq \lambda_2 > 0$

Blob detection in scale-space

- Blob detection using $\operatorname{trace} H$ or $\det H$ operators

$$H = \begin{bmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{bmatrix} \qquad \text{Hesse matrix}$$

$$\operatorname{trace} H = g_{xx} + g_{yy} = \Delta g$$

$$\det H = g_{xx} g_{yy} - g_{xy}^2 \qquad \text{Determinant-of-Hessian (DoH)}$$

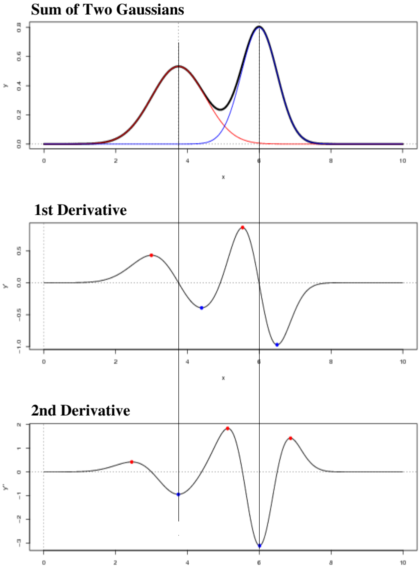
- Eigenvalues of H are proportional to main curvatures of $g(\mathbf{x})$
- Search for local extrema in image:

$$\mathbf{x}_b = \arg\min_{\mathbf{x}} \Delta g \qquad \text{for bright blobs}$$

$$\mathbf{x}_d = \arg\max_{\mathbf{x}} \Delta g \qquad \text{for dark blobs}$$

$$\mathbf{x}_a = \arg\max_{\mathbf{x}} \det H \qquad \text{for all blobs}$$

Relations between blobs and derivatives



- Two close bright blobs of Gaussian shape
  - with some overlap
- Location of blob:
  - maximum of g
  - zero of g_x
  - valley of g_{xx}

Scale-normalised co-ordinates

- What to do if blobs in image are of varying size?
  - Δg and det H are sensitive to blob size
  - $\rightarrow$ operators must be tuned to size

- Adaptive selection of scale parameter also needed when
  - size of objects is unknown
  - distance between object and camera varies

- Scale-normalised co-ordinates

$$\zeta = \frac{1}{\sqrt{\xi}} x = \frac{1}{\sigma} x$$
$$\eta = \frac{1}{\sqrt{\xi}} y = \frac{1}{\sigma} y$$

- Used for automatic scale selection

Automatic scale selection

- Based on Gaussian operators

$$G_{\zeta^m \eta^n}(\mathbf{x}, \xi) = \xi^{\frac{m+n}{2}} G_{x^m y^n}(\mathbf{x}, \xi)$$

- Operators are formed by **normalised derivatives**

$$\partial_\zeta = \sqrt{\xi}\partial_x = \sigma\partial_x$$
$$\partial_\eta = \sqrt{\xi}\partial_y = \sigma\partial_y$$

- In scale-space, search for **extrema** of features expressed by normalised derivatives

Normalised derivatives and image features

- For image features expressed by normalised derivatives:
  - if feature attains local maximum at scale ξ_0
  - then in image resized by factor s maximum will be at scale $s^2\xi_0$

$$\xi_0 \longrightarrow s^2\xi_0, \quad \text{that is}$$
$$\sigma_0 \longrightarrow s\sigma_0$$

  - this property is called **scale-invariance**

- Scale-invariant and -adaptive features can be expressed by normalised derivatives
  - blob, corner, edge
  - extension to affine-invariant features
  - affine-invariant region descriptors

Detection of blobs of varying size

- Use scale-normalised operators

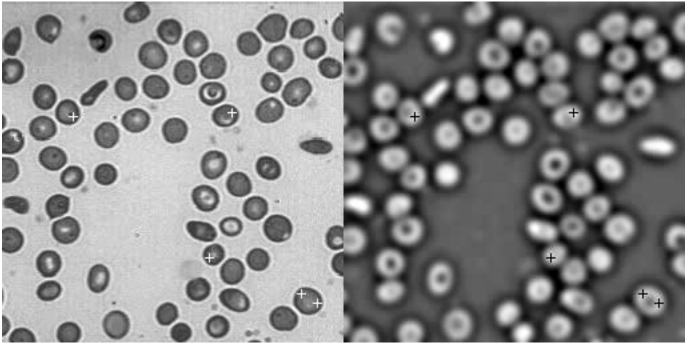
$$\Delta g_{norm} = \xi \left(g_{xx} + g_{yy} \right)$$
$$\det H_{norm} = \xi^2 \left(g_{xx} g_{yy} - g_{xy}^2 \right)$$

- Search for local extrema **in scale-space***

$$\mathbf{x}_d = \arg\min_{\mathbf{x},\xi} \Delta g_{norm} \qquad \text{for dark blobs}$$
$$\mathbf{x}_b = \arg\max_{\mathbf{x},\xi} \Delta g_{norm} \qquad \text{for bright blobs}$$
$$\mathbf{x}_a = \arg\max_{\mathbf{x},\xi} \det H_{norm} \qquad \text{for all blobs}$$

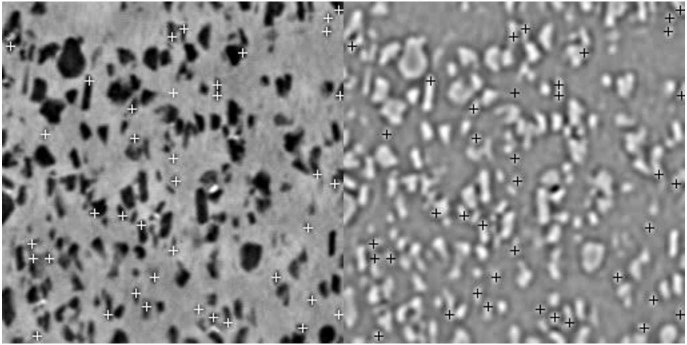
* Note that sign of Δg_{norm} has changed!

Blood cell detection: scale-variant version



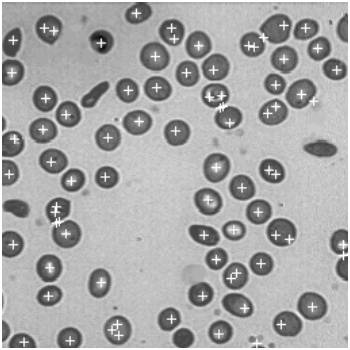
- Left video: all blobs detected by now
- Right video: LoG scale-space with current maxima
- 24 scales used $\rightarrow$ larger blobs detected at larger scales
- Some blobs detected several times
  $\rightarrow$ post-processing needed (non-maxima suppression)

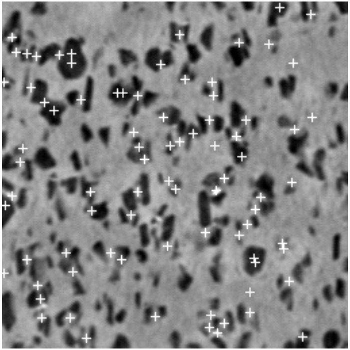
Another example of scale-variant blob detection



- 25 scales examined
- Two or more close blobs can merge at greater scale
- Again, post-processing needed to handle multiple detections

Two results of scale-invariant algorithm





- Only 6 scales examined in narrow range
  - $\rightarrow$ fine scale tuning still necessary
- In discrete case, invariance not perfect
  - $\rightarrow$ scale-variant results are probably slightly better

Outline

1. Scale-space
   - Scale-space and diffusion

2. Image features in scale-space
   - Scale selection

3. Affine-invariant features
   - Corner-like features
   - Blob-like features

Invariant features in 3D reconstruction

- Point correspondence across views: (multiview) stereo, video
- Invariant local feature points
  - detection: points, regions
  - description: region $\longrightarrow$ point (neighbourhood)
  - robust point/region matching
- Invariance
  - scale (near-far)
  - perspective distortion
  - illumination
- Local approximation of perspective distortion
  - small region, locally flat surface patch

  $\rightarrow$ **affine distortion** $\longrightarrow$ affine invariance

Covariant or invariant?

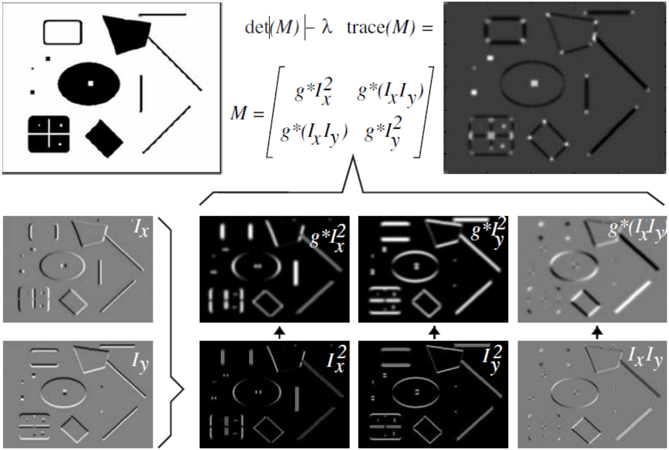
- **Covariant**: changing in the same way
  - function $f(x)$ **covariant** with transformation A: $f(Ax) = Af(x)$

  $\rightarrow$ axis of inertia of 2D shape 'rotates' with shape
- **Invariant**: not changing
  - function $f(x)$ **invariant** to transformation A: $f(Ax) = f(x)$

  $\rightarrow$ area of 2D shape is invariant to shape rotation
- Regions **covariant** with affine distortion

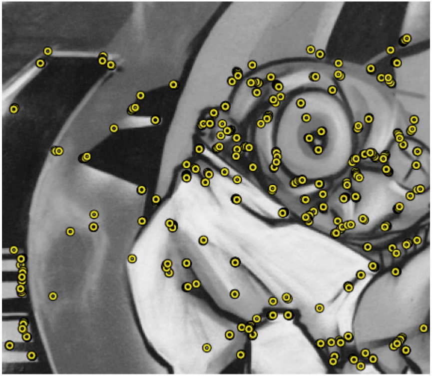
  $\rightarrow$ undergo affine distortion
- Descriptions **invariant** (insensitive) to affine distortion
  - description of affine-normalised region
  - or inherently affine-invariant, e.g, affine-invariant moments
- For simplicity, the two terms are often used in same sense
  - e.g., invariant regions

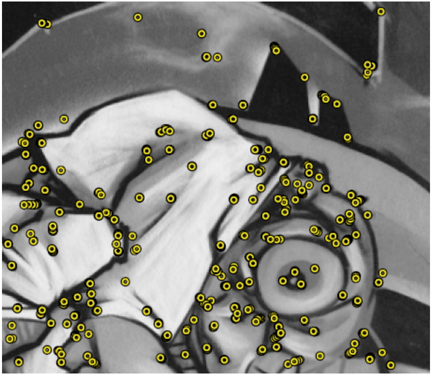
Operation of Harris corner detector



- Here, g is the Gaussian filter (G)
- Source: **Tuytelaars, Mikolajczyk (2007)**

Rotation invariance of Harris operator





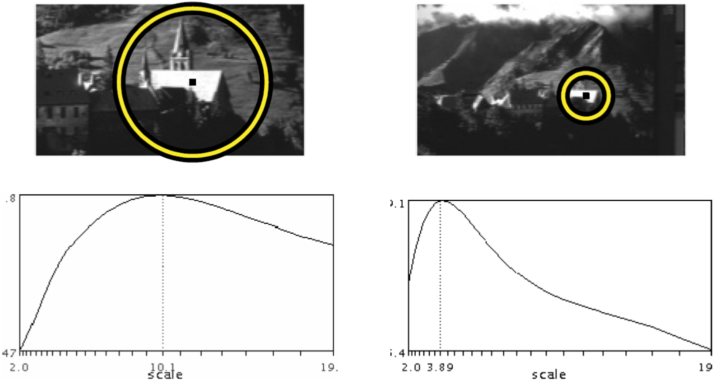
- Under rotation, great majority of features are preserved
  $\rightarrow$ result of Harris operator is stable

Steps of Harris-Laplace operator

1. Corner detection by *multiscale* Harris operator
   $\rightarrow$ corners at varying degree of detail
2. Calculating *characteristic scale* for every corner
   - by rotation-symmetric Laplace operator
   - maximum similarity between local image structure and operator

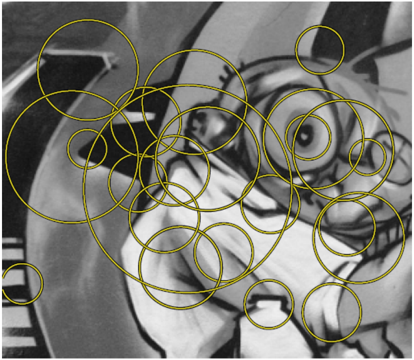
   $\rightarrow$ characteristic scale and size (radius) of region
3. Characteristic scale can be selected in different ways
   - rotation-symmetric Laplace operator gives best result
   - conclusion of experimental studies

Characteristic scales at different zooms



- Change of (abs) Laplacian values in selected points
  - $\rightarrow$ characteristic scales: 10.1 (left) and 3.9 (right)
  - $\rightarrow$ ratio of scales: 2.5 $=$ degree of magnification
  - $\rightarrow$ radius of circle: characteristic scale $\times$ 3

Scale and rotation invariance of Harris-Laplace

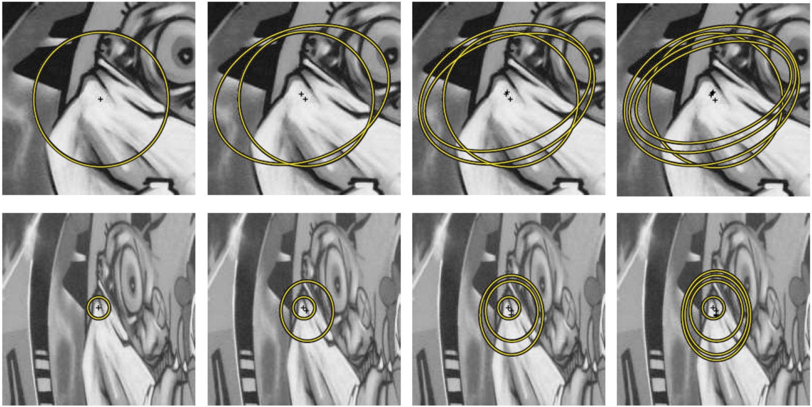


- Circles indicate characteristic scales of features
- For better visibility, some features are not shown

Main steps of Harris-affine operator

1. Detection of initial point and region by Laplace-Harris
   $\rightarrow$ initial scale, feature point, circular region
2. Estimation of affine region using structure matrix M
   $\rightarrow$ feature point with elliptic region
3. Normalisation of elliptic affine region to circular shape
   $\rightarrow$ normalised image
4. Calculation of new position and scale in normalised image
   $\rightarrow$ modified scale and position
5. Calculation of eigenvalues of new matrix M
   - if eigenvalues are different, go to step 2
   - otherwise, output final scale, position and circular region

Iterative detection of affine-invariant features



- First column: points used for initialisation
- Further columns: points and regions after iterations $1, 2, 3$
  - $\rightarrow$ after third iteration, shapes converge to corresponding regions

Affine normalisation of Harris operator 1/2

- Eigenvalues of matrix M define affine region (ellipse)
- Search for transformation when two eigenvalues become equal
  $\rightarrow$ ellipse becomes circle when iterations stop
- This can be achieved by **square root** of M:

$$\mathbf{x}'_L = M_L^{1/2}\mathbf{x}_L \text{ left image}$$
$$\mathbf{x}'_R = M_R^{1/2}\mathbf{x}_R \text{ right image}$$

- Since transformed images are analysed, **inverse** matrices are often used and iterated

$$\mathbf{x}_L = M_L^{-1/2}\mathbf{x}'_L,$$
$$\mathbf{x}_R = M_R^{-1/2}\mathbf{x}'_R$$

Affine normalisation of Harris operator 2/2

- Normalised images are identical up to rotation:

$$\mathbf{x}'_L = R\mathbf{x}'_R$$

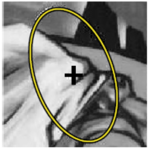
  - rotation of region does not affect ration of eigenvalues

  $\rightarrow$ affine distortion can only be determined up to rotation

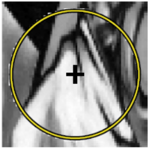
- In normalised images M'_L and M'_R are **rotation matrices**
- Affine normalisation procedure works if
  - $\det M > 0 \longrightarrow M^{-1/2}$ exists
  - signal-noise ratio is sufficiently large

  $\rightarrow$ e.g., for initial points detected by Harris-Laplace

Example of affine normalisation for stereo images



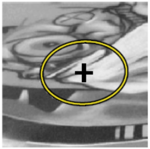
$$\mathbf{x}_L \longrightarrow M_L^{-1/2}\mathbf{x}'_L$$



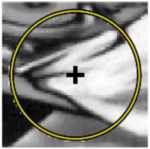
$$\downarrow$$

$$\mathbf{x}'_L \longrightarrow R\mathbf{x}'_R$$

$$\downarrow$$

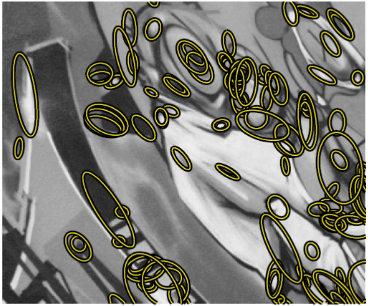
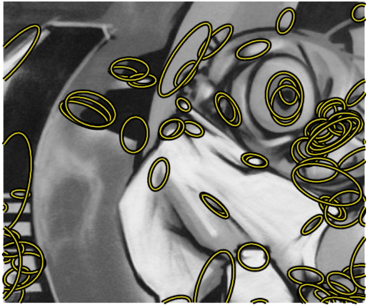


$$\mathbf{x}_R \longrightarrow M_R^{-1/2}\mathbf{x}'_R$$



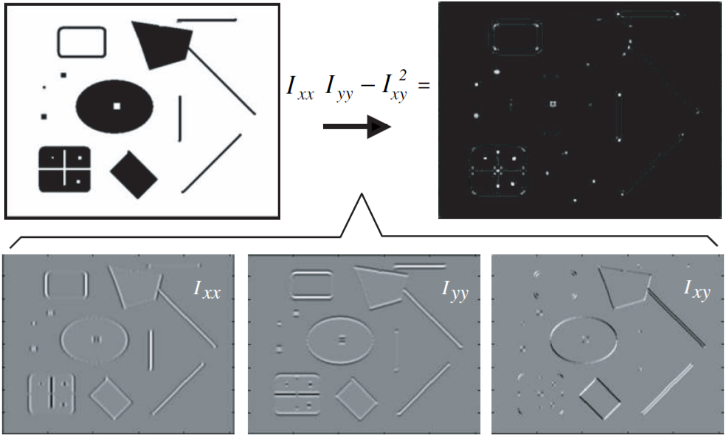
- Image co-ordinates **x** are transformed by matrix $M^{1/2}$
- Normalised images are identical up to rotation (R)

Example and summary of Harris-affine operator

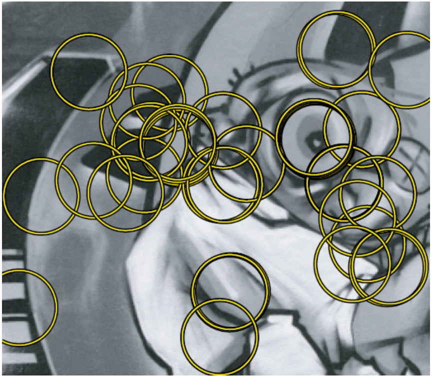
- Regions correspond despite affine distortion
- Versions of Harris operator are often used
  - single or multiscale, scale-invariant, affine-invariant
  - efficient, stable; controllable number of points, can be large
- Image corners are well detected in locally flat surface areas
  - poor performance where surface variation is strong

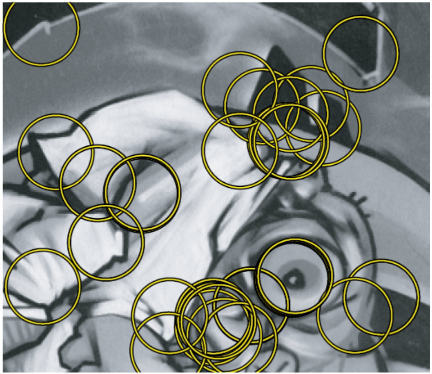
Operation of Determinant-of-Hessian (DoH)



- Components of Laplacian enhance lines, as well
- DoH enhances blobs, corners and ends of lines

Rotation invariance of DoH blob detector



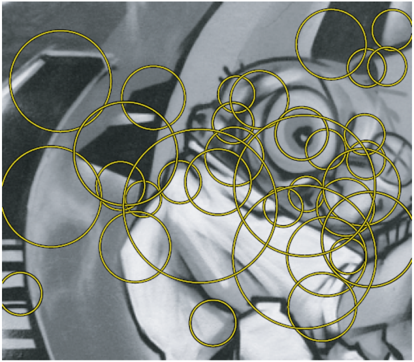


- Under rotation, great majority of features are preserved
  - $\rightarrow$ result of DoH operator is stable

Hesse-Laplace and Hesse-affine operators

- Similar to correspondent corner detectors
  - Hesse-Laplace $\rightarrow$ Harris-Laplace
  - Hesse-affine $\rightarrow$ Harris-affine
- Essential, natural difference:
  - initial points are DoH features rather than Harris corners
- The rest is similar, e.g., iterative affine normalisation:
  - estimation of affine region using structure matrix M
  - affine region normalisation to circular shape
  - calculation of new position and scale in normalised image
  - calculation of eigenvalues of new matrix M

Zoom invariance of Hesse-Laplace



- Circles indicate characteristic scales of features
- For better visibility, some features are not shown

Example and summary of Hesse-affine operator

- Versions of Hesse operator are often used
  - yield many points, can cover image
  - number of points controllable by DoH and Laplace thresholds
- Can detect corners, as well
  - better scale estimation than by Harris
  - → second-order derivatives for all operations
  - → Harris: mixed, first and second order