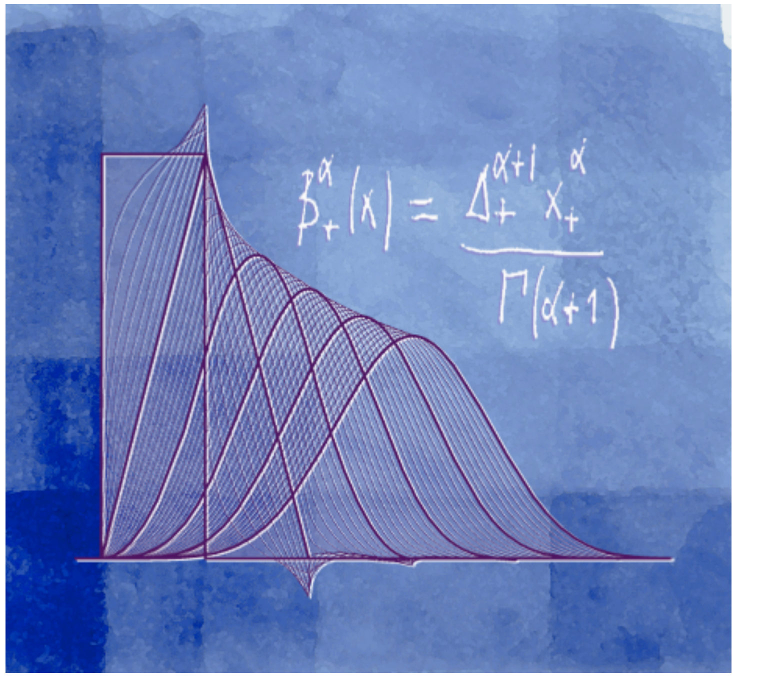
Image Processing

Chapter 6

Directional Image Analysis and Processing

Prof. Michael Unser, LIB



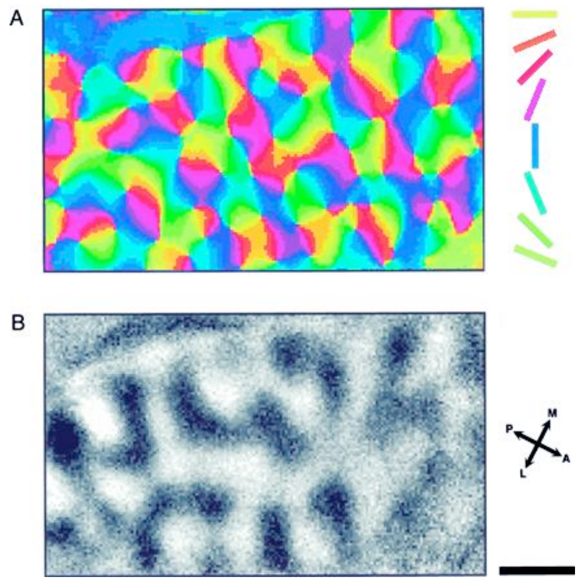
February 2024

CONTENT

- **6.1 Mathematical foundations**
  - Rotation in the Fourier transform
  - Radon transform
  - Rotation of polynomials
  - Directional derivatives
- **6.2 Local directional analysis**
  - Structure tensor ← IP-LAB-4
- **6.3 Steerable filters**
  - Derivative-based filters
- **Appendix**
  - Edge and ridge detectors ← Demo

Directionality in image processing

- Importance of directional cues
  - Edges, ridges, patterns, texture
  - Visual perception is orientation-sensitive
  - Organization of primary visual cortex: orientation-selective detectors (Hubel-Wiesel)



- Invariant processing and feature detection
  - Invariant operators: Gradient, Laplacian, ...
- Computational challenges
  - Selectivity to orientation
  - Steerability (orientation can be arbitrary)
  - Separable filters are not orientation-sensitive

6.1 Mathematical foundations

- Rotation property of the Fourier transform
- Central-slice theorem
- Steerability of polynomials
- Directional derivatives

Rotation in the Fourier domain

Continuous-domain Fourier transform: $\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} f(\boldsymbol{x}) e^{-j\langle \boldsymbol{\omega}, \boldsymbol{x}\rangle} \mathrm{d}x_1 \cdots \mathrm{d}x_d$

- Rotation property of the Fourier transform

$$f(\boldsymbol{R}\boldsymbol{x}) \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad \hat{f}(\boldsymbol{R}\boldsymbol{\omega})$$

$\boldsymbol{R}$: Orthonormal $d \times d$ matrix such that $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$

Example (2×2 rotation matrix): $\boldsymbol{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Proof (by change of variable): $\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x} \quad \rightarrow \quad \mathrm{d}y_1 \cdots \mathrm{d}y_d = \underbrace{\det(\boldsymbol{R})}_{=1} \mathrm{d}x_1 \cdots \mathrm{d}x_d$

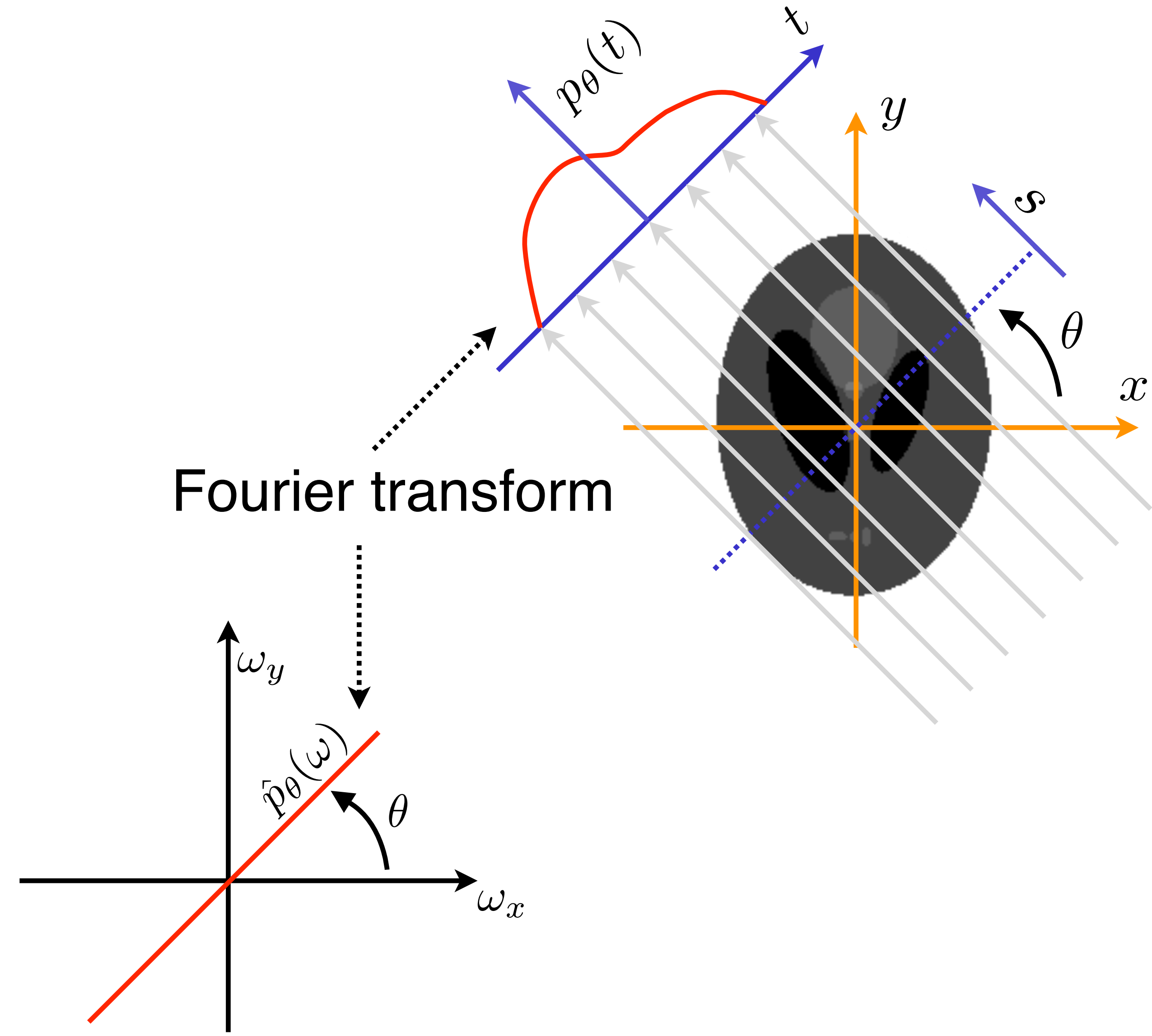
$$\begin{aligned} \int_{\mathbb{R}^d} f(\boldsymbol{R}\boldsymbol{x}) e^{-j\langle \boldsymbol{\omega}, \boldsymbol{x}\rangle} \, \mathrm{d}x_1 \cdots \mathrm{d}x_d &= \int_{\mathbb{R}^d} f(\boldsymbol{y}) e^{-j\langle \boldsymbol{\omega}, \boldsymbol{R}^T \boldsymbol{y}\rangle} \, \mathrm{d}y_1 \cdots \mathrm{d}y_d \\ &= \int_{\mathbb{R}^d} f(\boldsymbol{y}) e^{-j\langle \boldsymbol{R}\boldsymbol{\omega}, \boldsymbol{y}\rangle} \, \mathrm{d}y_1 \cdots \mathrm{d}y_d = \hat{f}(\boldsymbol{R}\boldsymbol{\omega}) \end{aligned}$$

Central-slice theorem

Radon transform

$$p_\theta(t) = R\{f(x,y)\}(t,\theta) = \int_{-\infty}^{+\infty} f(\underbrace{\boldsymbol{R}_\theta \boldsymbol{t}}_{\boldsymbol{x}=(x,y)})\,\mathrm{d}s$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \boldsymbol{R}_\theta \boldsymbol{t} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\boldsymbol{R}_\theta} \begin{pmatrix} t \\ s \end{pmatrix}$$



Central-slice theorem

$$\hat{p}_\theta(\omega) = \hat{f}(\omega\cos\theta,\, \omega\sin\theta) = \hat{f}_{\mathrm{pol}}(\omega,\theta)$$

Proof: for $\theta = 0$

$$\hat{f}(\omega,0) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j\omega x}\,\mathrm{d}x\mathrm{d}y = \int_{-\infty}^{+\infty} \underbrace{\left(\int_{-\infty}^{+\infty} f(x,y)\,\mathrm{d}y\right)}_{p_0(x)} e^{-j\omega x}\,\mathrm{d}x = \hat{p}_0(\omega)$$

then use rotation property. . .

Steerability of polynomials

Property. The rotated version of a multidimensional polynomial of degree p is a polynomial of degree p. This implies that polynomials are "steerable".

Key observations for establishing rotation property

- A multidimensional polynomial of total degree p is a linear combination of monomials of degree $n \leq p$: $(x_1^{k_1} x_2^{k_2} \cdots x_d^{k_d})$ with $\sum_{i=1}^{d} k_i = n$
- Applying an affine transformation to a monomial of (total) degree k yields a linear combination of monomials of degree $k = \sum_{i=1}^{d} l_i$ witk $l_i \geq 0$

$$\left(\sum_{i=1}^{d} a_i x_i\right)^k = \sum_{l_1, l_2, \dots, l_d} \binom{k}{l_1, l_2, \dots, l_d} \left(a_1^{l_1} a_2^{l_2} \cdots a_d^{l_d}\right) x_1^{l_1} x_2^{l_2} \cdots x_d^{l_d}$$

- $(a\, x_i^{n_1}) \cdot (b\, x_i^{n_2}) = (ab)\, x_i^{n_1+n_2}$

$\Rightarrow$ Overall result of rotation = linear combination of monomials

Reminder: multinomial expansion

- Multinomial coefficients:

$$\binom{n}{k_1, k_2, \ldots, k_d} = \frac{n!}{k_1!\, k_2! \cdots k_d!}$$

with convention that $0! = 1$

- Generalized version of binomial expansion

$$(y_1 + y_2 + \cdots + y_d)^n = \sum_{k_1, k_2, \ldots, k_d} \binom{n}{k_1, k_2, \ldots, k_d} y_1^{k_1} y_2^{k_2} \cdots y_d^{k_d}$$

with summation over all nonnegative multi-integers such that $\sum_{i=1}^{d} k_i = n$

Explicit steering of a 2D polynomial (optional)

Rotated polynomial of degree M: $p_\theta(y_1, y_2) = \sum_{m=0}^{M} \sum_{n=0}^{m} a_{m,n} y_1^{m-n} y_2^n$

Steering transformation: $y_1 = \cos\theta\, x_1 - \sin\theta\, x_2, \quad y_2 = \sin\theta\, x_1 + \cos\theta\, x_2$

$$y_1^{m-n} = \sum_{k=0}^{m-n} \binom{m-n}{k} (\cos\theta)^k (-1)^{m-n-k} (\sin\theta)^{m-n-k} x_1^k x_2^{m-n-k}$$

$$y_2^n = \sum_{l=0}^{n} \binom{n}{l} (\sin\theta)^l (\cos\theta)^{n-l} x_1^l x_2^{n-l}$$

$$p(x_1, x_2) = \sum_{m=0}^{M} \sum_{n=0}^{m} a_{m,n} \sum_{k=0}^{m-n} \sum_{l=0}^{n} \binom{m-n}{k} \binom{n}{l} (\cos\theta)^{k+n-l} (-1)^{m-n-k} (\sin\theta)^{m-n-k+l} x_1^{k+l} x_2^{m-(k+l)}$$

Change of variables: $i - j = k + l; \quad j = m - (k + l)$ and collecting factors

$$\Rightarrow \quad p_\theta(y_1, y_2) = p(x_1, x_2) = \sum_{i=0}^{M} \sum_{j=0}^{i} b_{i,j}\; x_1^{i-j} x_2^j$$

with $b_{i,j} =$ polynomial$(\cos\theta, \sin\theta)$

Gradient and directional derivatives

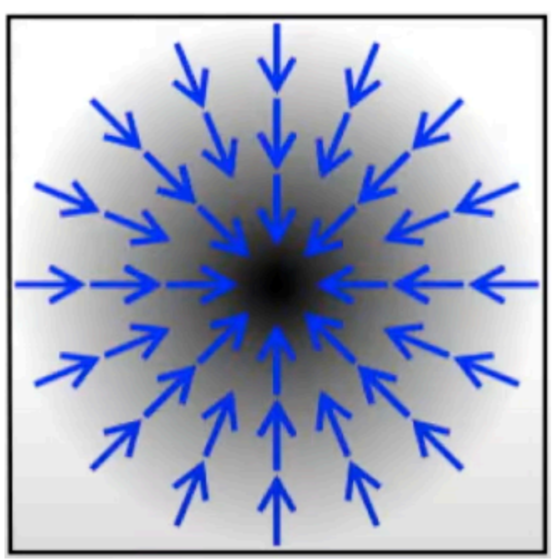
Direction specification: $\boldsymbol{u} = (u_1, \dots, u_d) \in \mathbb{R}^d$ with $\|\boldsymbol{u}\| = 1$ (unit vector)

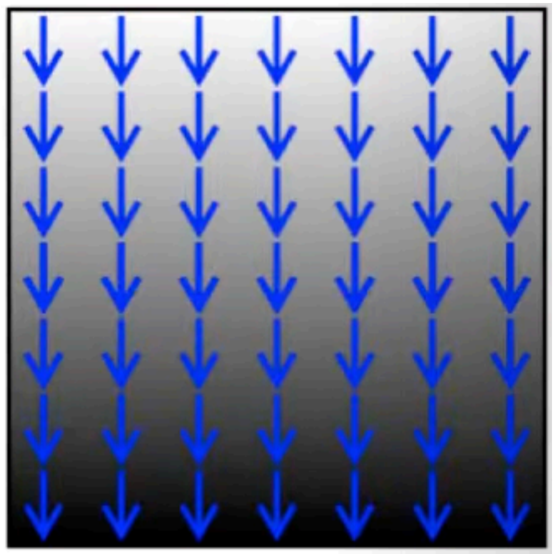
- First-order directional derivatives

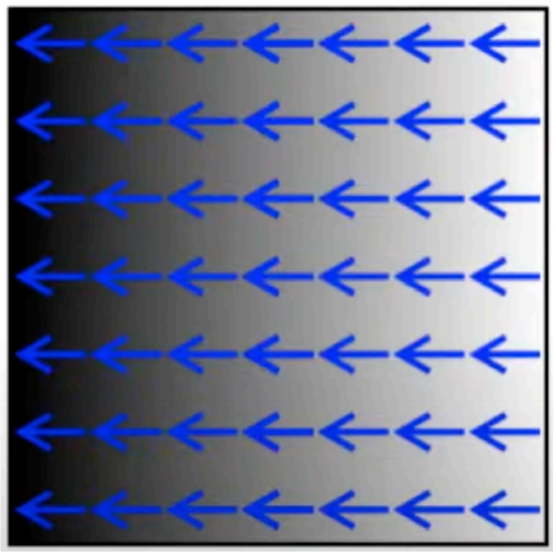
$$\mathrm{D}_{\boldsymbol{u}} f(\boldsymbol{x}) = \lim_{h \to 0} \frac{f(\boldsymbol{x}) - f(\boldsymbol{x} - h\boldsymbol{u})}{h}$$

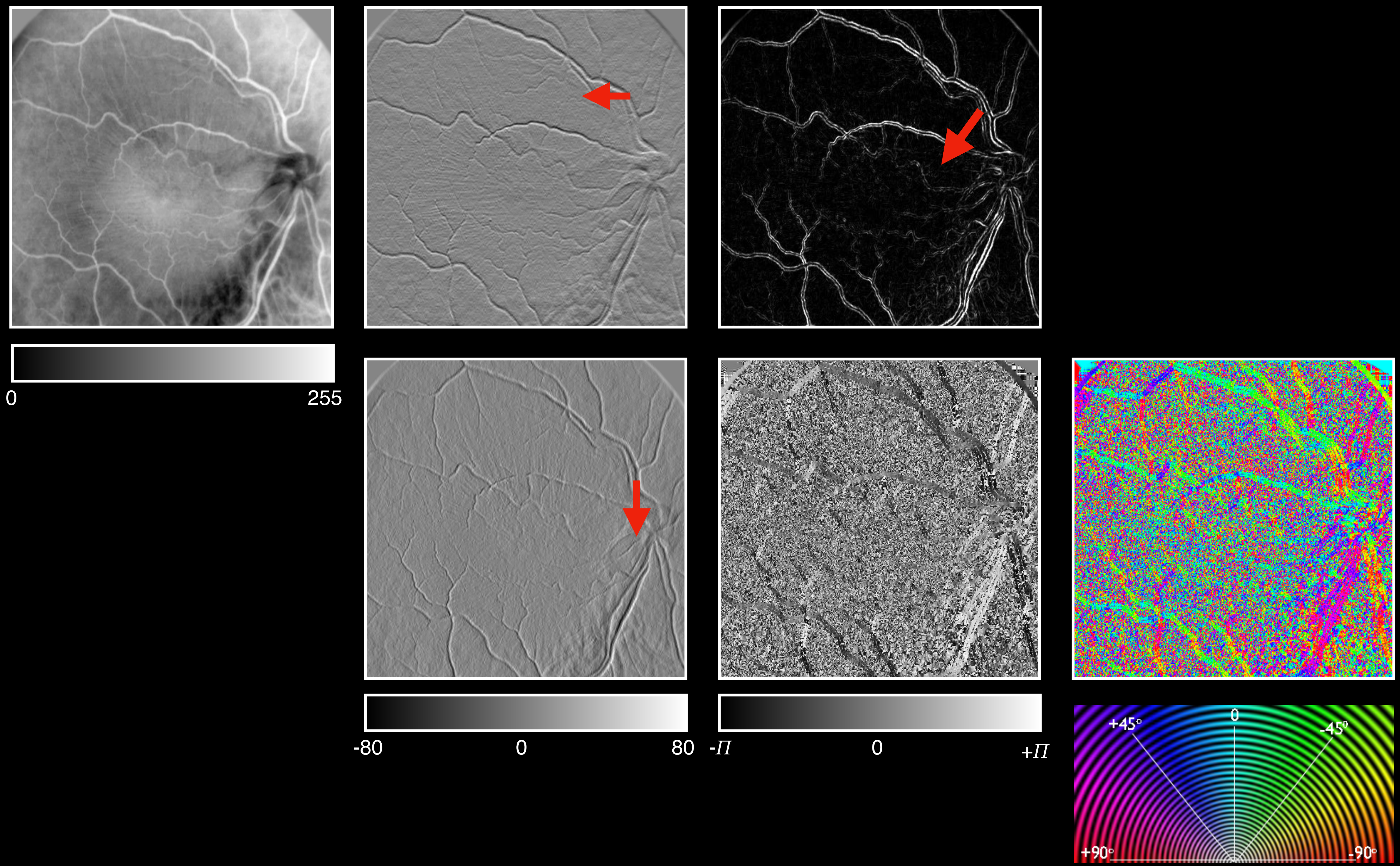
$$= \langle \boldsymbol{u}, \boldsymbol{\nabla} f(\boldsymbol{x}) \rangle \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad j\langle \boldsymbol{u}, \boldsymbol{\omega} \rangle \hat{f}(\boldsymbol{\omega})$$

$$= \sum_{i=1}^{d} u_i \frac{\partial f(\boldsymbol{x})}{\partial x_i} \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad \left(j \sum_{i=1}^{d} u_i \omega_i \right) \hat{f}(\boldsymbol{\omega})$$








0
255
-80
0
80
-Π
0
+Π
+45°
0
-45°
+90°
-90°

Higher-order directional derivatives

Direction specification: $\boldsymbol{u} = (u_1, \dots, u_d) \in \mathbb{R}^d$ with $\|\boldsymbol{u}\| = 1$ (unit vector)

- Directional derivatives of order n

$$\mathrm{D}^n_{\boldsymbol{u}} f(\boldsymbol{x}) = \underbrace{\mathrm{D}_{\boldsymbol{u}}\mathrm{D}_{\boldsymbol{u}} \cdots \mathrm{D}_{\boldsymbol{u}}}_{n \text{ times}} f(\boldsymbol{x}) \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad \left(j \sum_{i=1}^{d} u_i \omega_i\right)^n \hat{f}(\boldsymbol{\omega})$$

$$= \sum_{k_1, k_2, \dots, k_d} \binom{n}{k_1, k_2, \dots, k_d} u_1^{k_1} u_2^{k_2} \cdots u_d^{k_d} \frac{\partial^n f(\boldsymbol{x})}{\partial x_1^{k_1} \partial x_2^{k_2} \cdots \partial x_d^{k_d}}$$

$\Rightarrow$ linear combination of partial derivatives of order n

- 2D example: $n = 2$
  - Unit vector: $\boldsymbol{u}_\theta = (\cos\theta, \sin\theta)$
  - $\mathrm{D}^2_{\boldsymbol{u}_\theta} f(x_1, x_2) = \cos^2\theta \frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} + 2\cos\theta \sin\theta \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} + \sin^2\theta \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2}$

6.2 Directional image analysis

- Structure tensor
- Implementation
- Examples of 2D directional analysis

Structure tensor

- Structure tensor at location $\boldsymbol{x}_0$

$$\mathbf{J}(\boldsymbol{x}_0) = \int_{\mathbb{R}^d} w(\boldsymbol{x} - \boldsymbol{x}_0) \Big(\boldsymbol{\nabla} f(\boldsymbol{x}) \boldsymbol{\nabla}^T f(\boldsymbol{x}) \Big) \, \mathrm{d}x_1 \cdots \mathrm{d}x_d$$

  - $w(\boldsymbol{x})$: nonnegative symmetric observation window (e.g., Gaussian)
  - $\mathbf{J}$: $d \times d$ symmetric matrix
  - Eigenvectors and eigenvalues: $\mathbf{J}\mathbf{u}_i = \lambda_i \cdot \mathbf{u}_i$, with $\lambda_1 \geq \cdots \geq \lambda_d$

- Interpretation (for window centered at $\boldsymbol{x}_0 = \mathbf{0}$)

  - Weighted differential inner-product matrix: $\mathbf{J} = \langle \boldsymbol{\nabla} f, \boldsymbol{\nabla} f \rangle_w$

    $[\mathbf{J}]_{i,j} = \langle \frac{\partial f}{\partial x_i}, \frac{\partial f}{\partial x_j} \rangle_w$ with $\langle f_1, f_2 \rangle_w = \int_{\mathbb{R}^d} w(\boldsymbol{x}) f_1(\boldsymbol{x}) f_2(\boldsymbol{x}) \, \mathrm{d}x_1 \cdots \mathrm{d}x_d$

  - Energy of the derivative in the direction $\mathbf{u}$

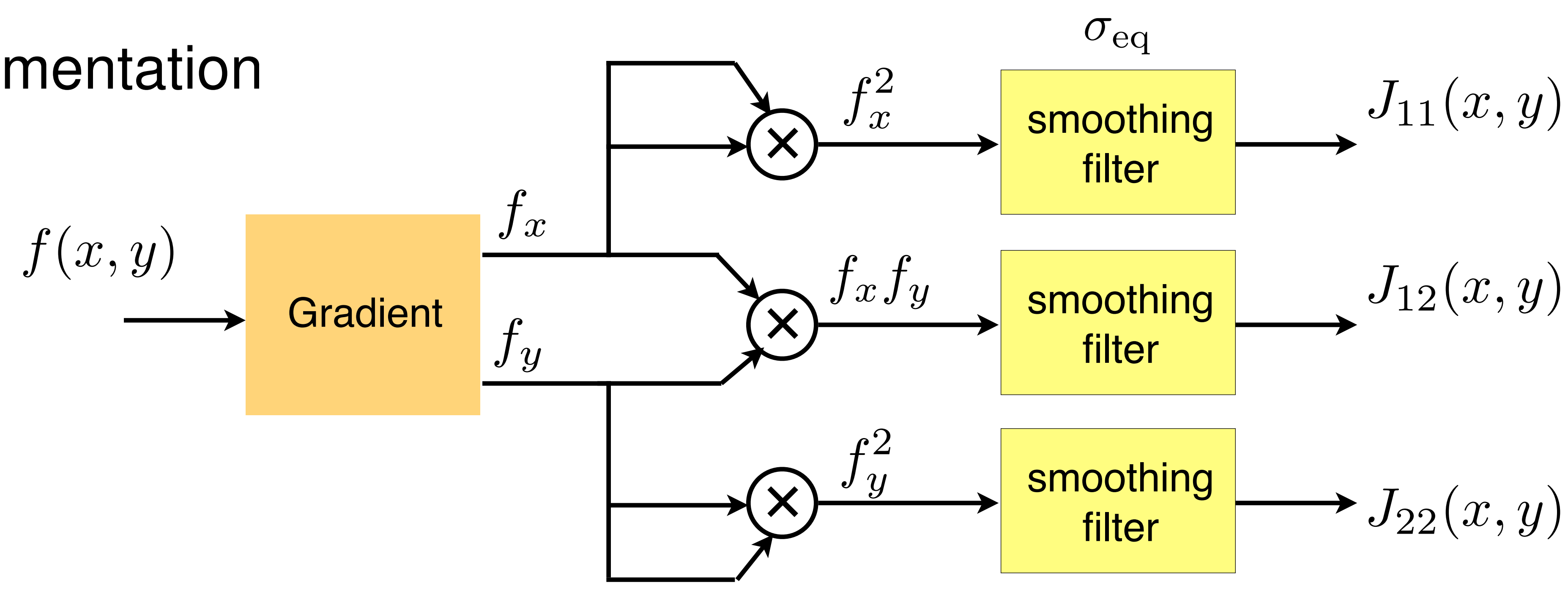
    $$\|\mathrm{D}_{\mathbf{u}} f\|_w^2 = \langle \mathbf{u}^T \boldsymbol{\nabla} f, \mathbf{u}^T \boldsymbol{\nabla} f \rangle_w = \mathbf{u}^T \langle \boldsymbol{\nabla} f, \boldsymbol{\nabla} f \rangle_w \mathbf{u} = \mathbf{u}^T \mathbf{J} \, \mathbf{u}$$

  - Dominant orientation of neighborhood: $\mathbf{u}_1 = \arg \max_{\mathbf{u}, \|\mathbf{u}\|=1} \|\mathrm{D}_{\mathbf{u}} f\|_w^2$

    Eigenvalues: $\lambda_i = \mathbf{u}_i^T \mathbf{J} \, \mathbf{u}_i$

Structure tensor in 2D

- Implementation

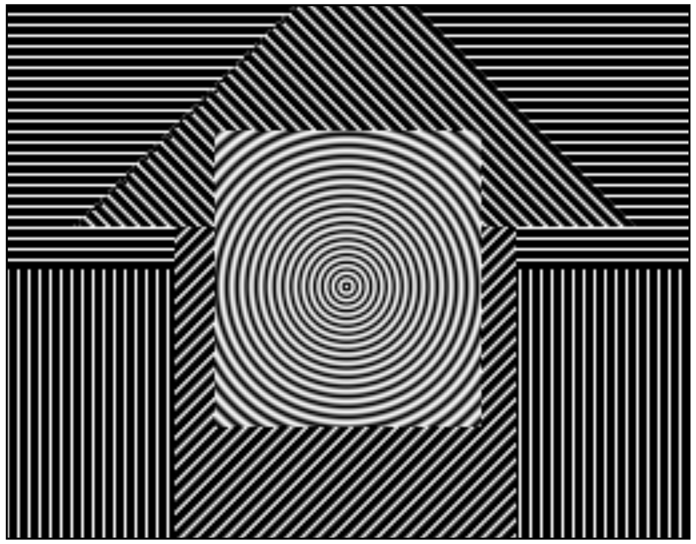


- Local features

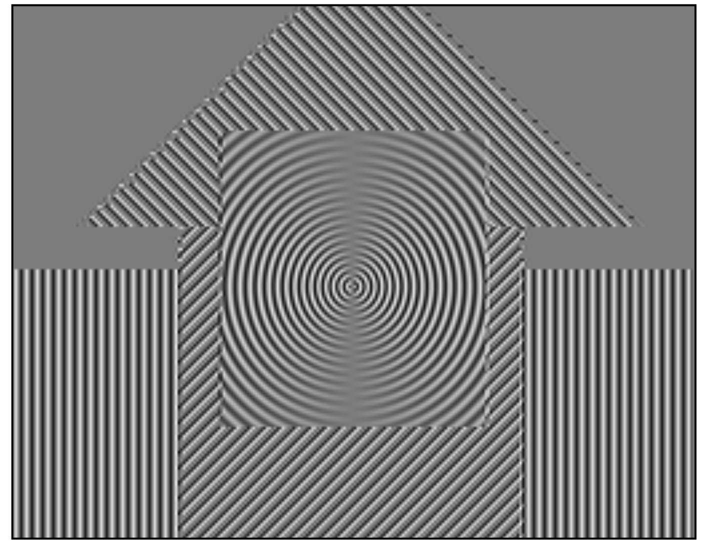
$$\mathbf{J}(x,y) = \begin{bmatrix} J_{11}(x,y) & J_{12}(x,y) \\ J_{12}(x,y) & J_{22}(x,y) \end{bmatrix}$$

  - Gradient energy: $E = \mathrm{trace}(\mathbf{J}) = J_{11} + J_{22}$
  - Orientation: $\mathbf{u}_1 = (\cos\theta, \sin\theta)$ with $\theta = \frac{1}{2}\arctan\left(\frac{2J_{12}}{J_{22} - J_{11}}\right)$
  - Coherency: $0 \leq C = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{\sqrt{(J_{22} - J_{11})^2 + 4J_{12}^2}}{J_{22} + J_{11}} \leq 1$
  - Harris corner index: $H = \det(\mathbf{J}) - \kappa\, \mathrm{trace}(\mathbf{J})^2$ with $\kappa \in [0.04, 0.15]$

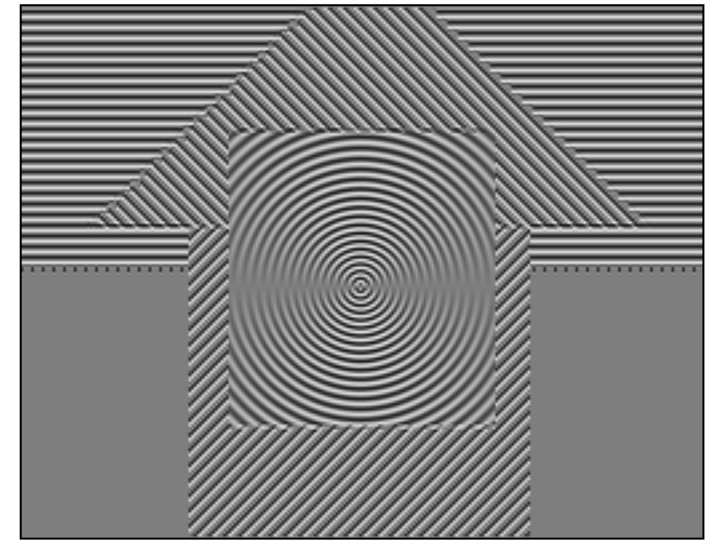
Examples of directional analysis



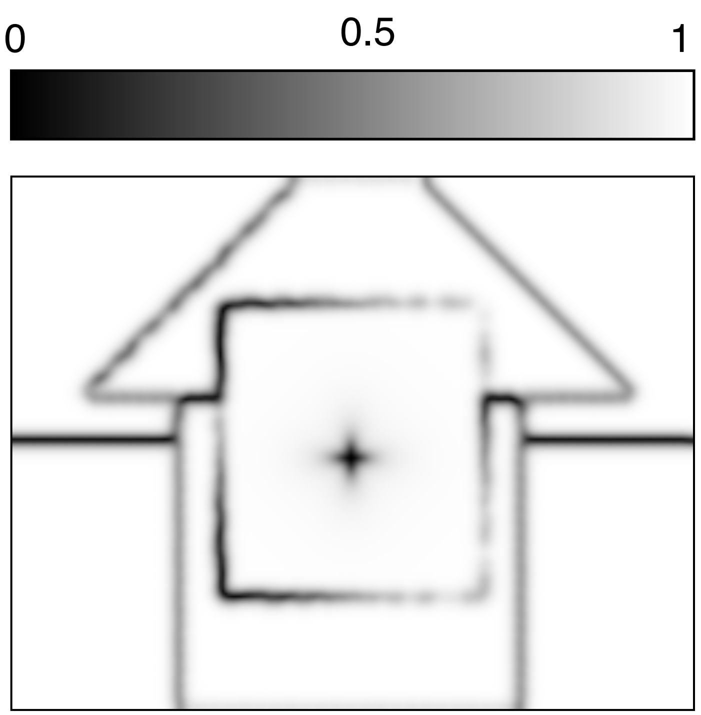
Input



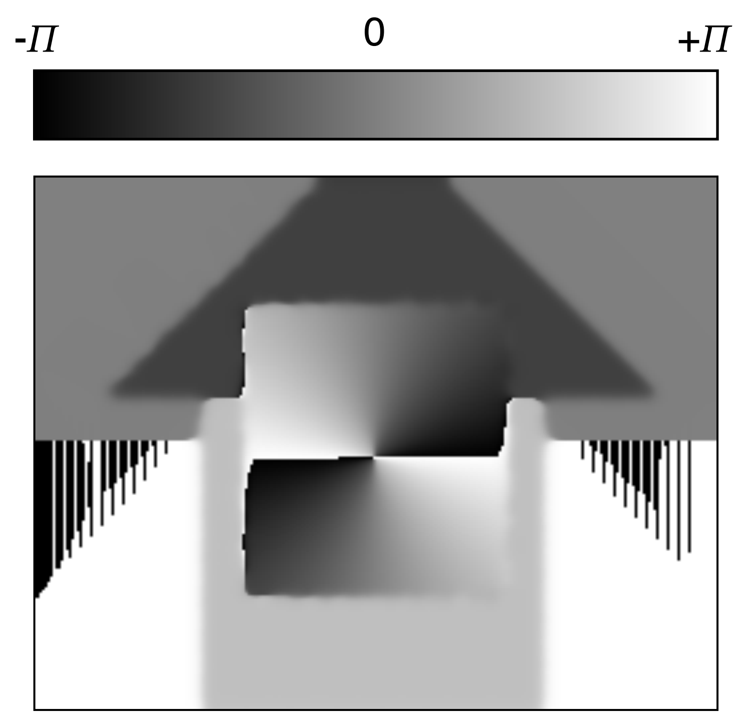
Dx



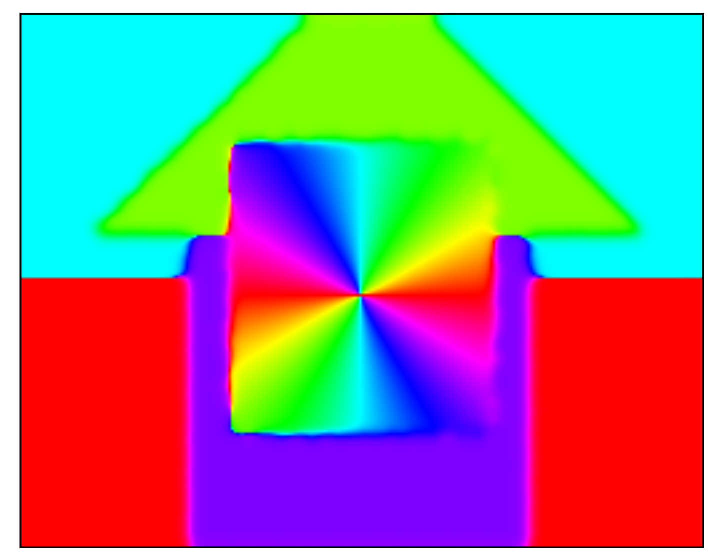
Dy



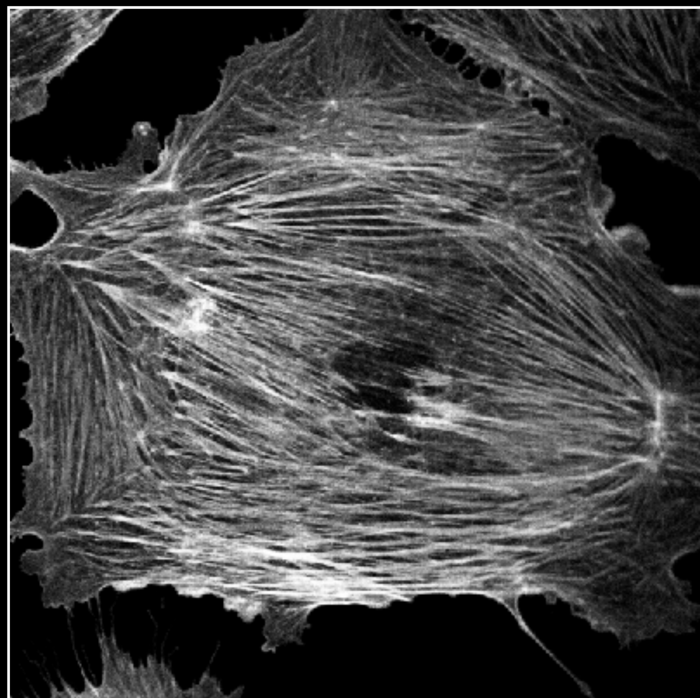
Coherency



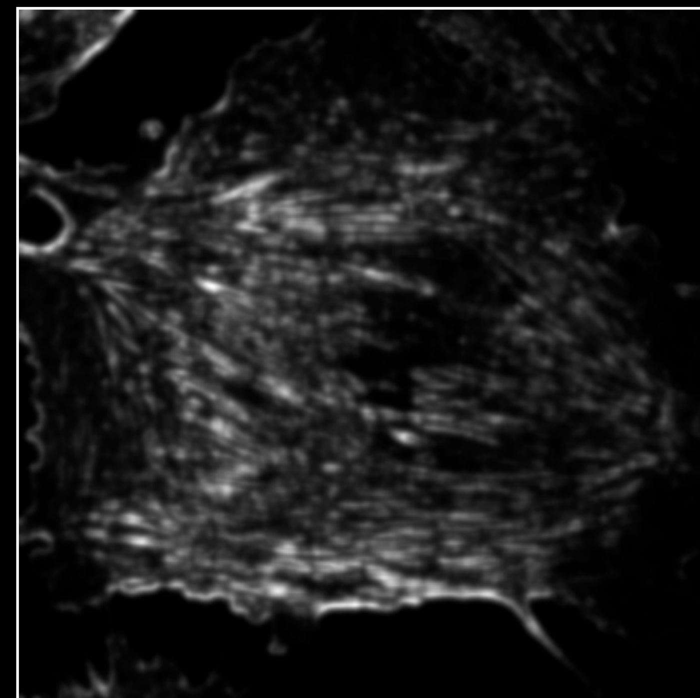
Orientation



Orientation

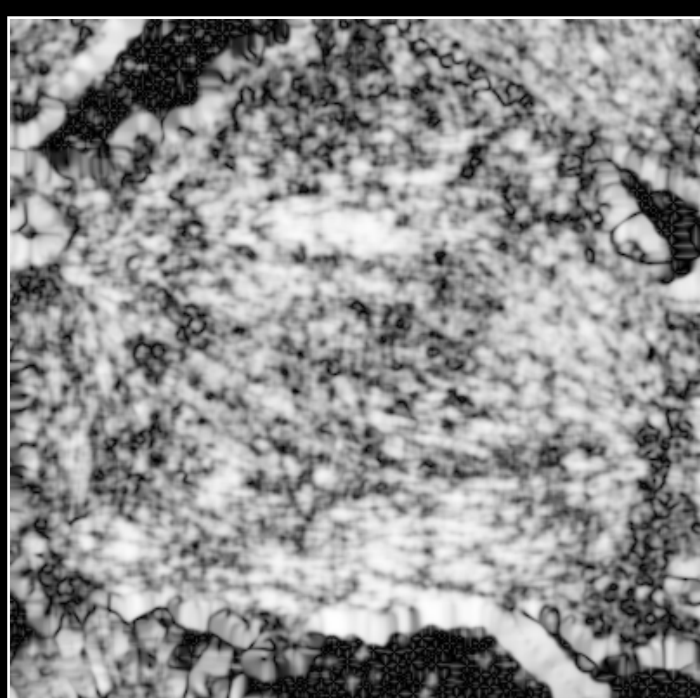


Input

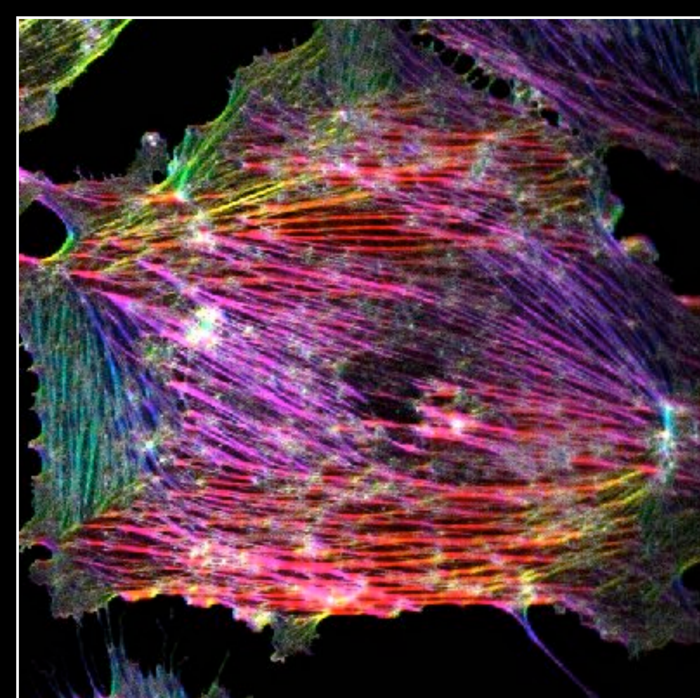


Energy

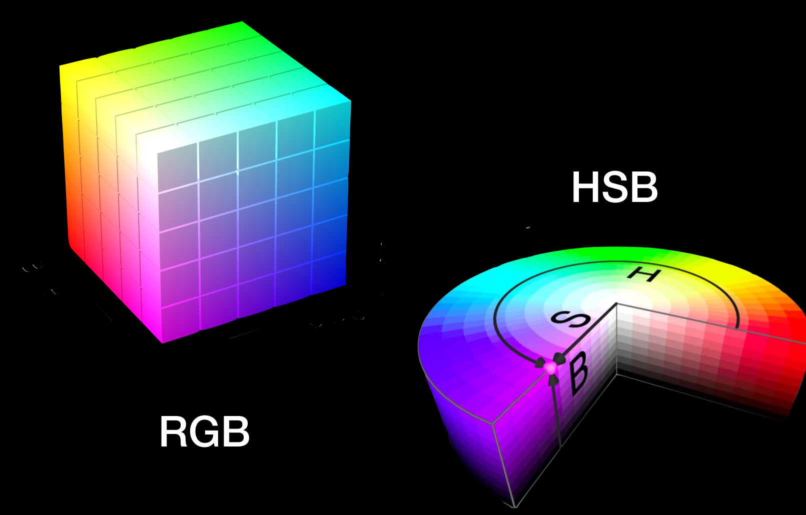
Orientation



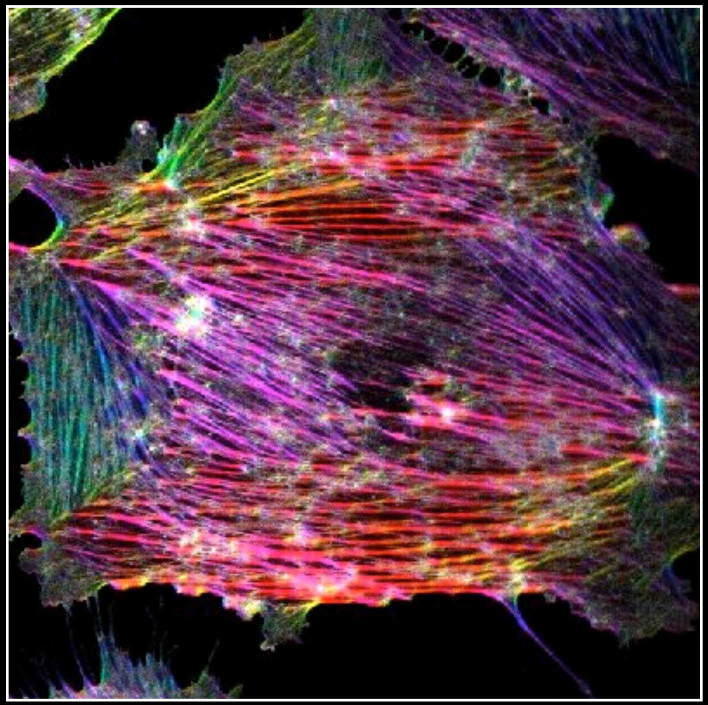
Coherency



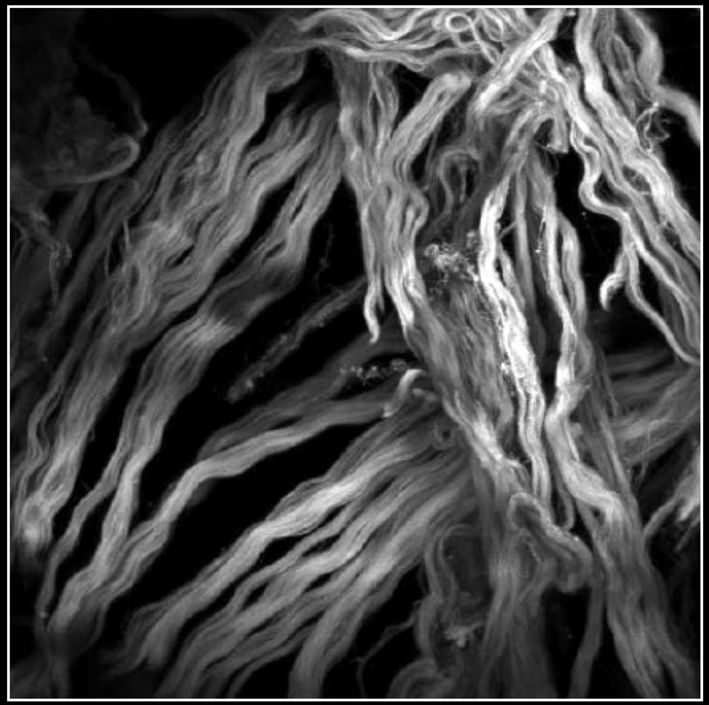
Color HSB representation



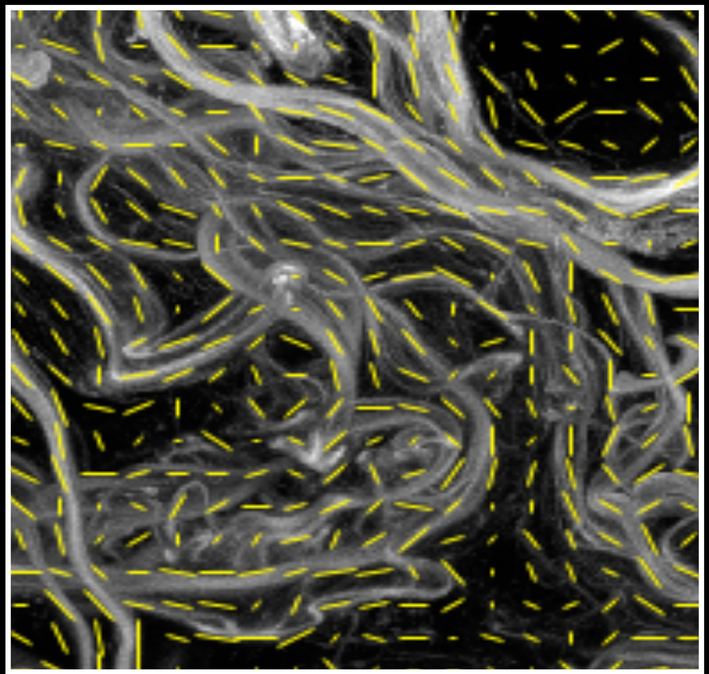
Hue: orientation
Saturation: coherency
Brightness: input



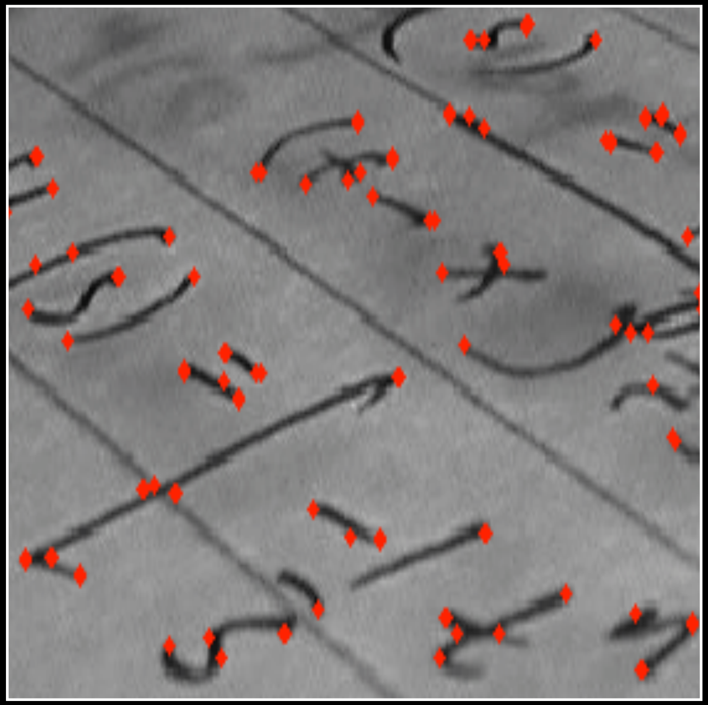
Directional Analysis



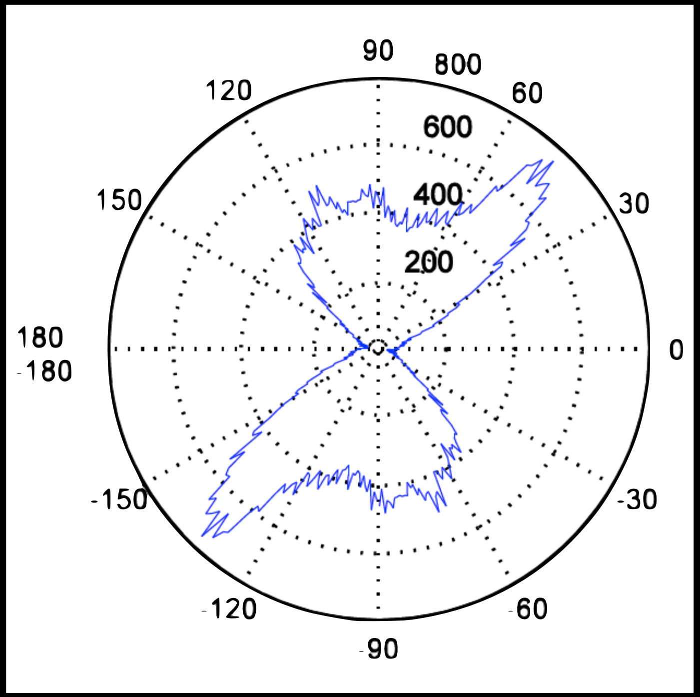
Distribution of orientations



Vector Field



Keypoints detector (Harris Corner)



Open-source software
Plugins of ImageJ
OrientationJ

Z. Püspöki et al. "Transforms and Operators for Directional Bioimage Analysis: A Survey," Advances in Anatomy, Embryology and Cell Biology, Springer International Publishing, 2016.

6.3 Steerable filters

- Directional pattern matching
- Steerable filters
- Derivative-based filters
- Appendix: specific implementations

Directional pattern matching

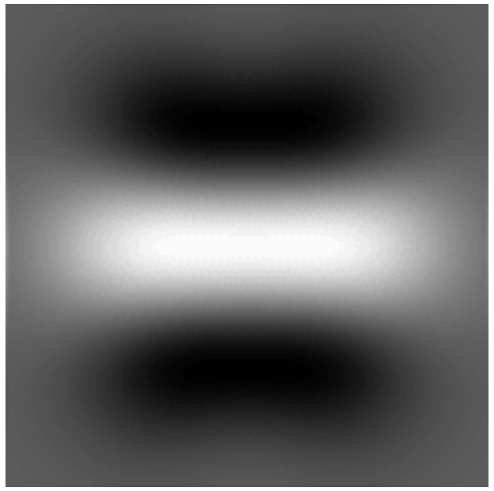
Task: detection/enhancement of a given type of directional pattern

Example: edge, line, ridge, filament, corner, etc.

- Local measurement model

$$f(\boldsymbol{x}) = I \cdot f_0\big(\boldsymbol{R}_\theta(\boldsymbol{x} - \boldsymbol{x}_0)\big) + n(\boldsymbol{x})$$

  - $f_0(\boldsymbol{x})$: feature template (elongated blob)
  - $\boldsymbol{x}_0$: spatial location (unknown)
  - $\boldsymbol{R}_\theta$: 2×2 rotation matrix by θ (unknown)
  - I: intensity (unknown)
  - $n(\boldsymbol{x})$: additive Gaussian noise



- Maximum-likelihood estimator (rotating matched filter)

$$\tilde{\theta}(\boldsymbol{x}_0) = \arg\max_\theta \{(f * h(\boldsymbol{R}_\theta \cdot))\,(\boldsymbol{x}_0)\}$$ where $h(\boldsymbol{x}) = f_0(-\boldsymbol{x})$

$$\tilde{I}(\boldsymbol{x}_0) = \big(f * h(\boldsymbol{R}_{\tilde{\theta}} \cdot)\big)\,(\boldsymbol{x}_0)$$

Problem: computationally very expensive!

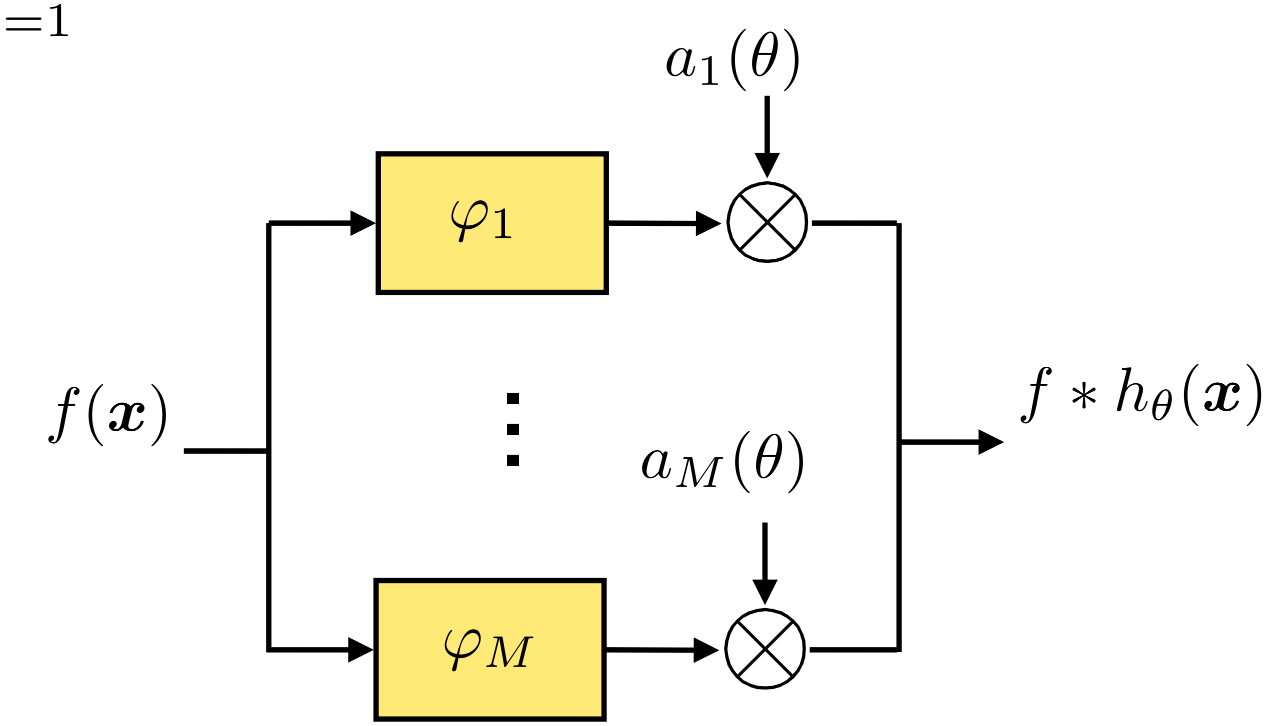
Steerable filters

(Freeman & Adelson, 1991)

Definition. A 2D filter $h(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^2$ is steerable of order M iff. there exist some basis filters $\varphi_m(\boldsymbol{x})$ and coefficients $a_m(\theta)$ such that

$$\forall \theta \in [-\pi, \pi], \quad h_\theta(\boldsymbol{x}) := h(\boldsymbol{R}_\theta \boldsymbol{x}) = \sum_{m=1}^{M} a_m(\theta)\, \varphi_m(\boldsymbol{x})$$

- Fast filterbank implementation



- Fourier-domain equivalence

  $h(\boldsymbol{x})$ steerable $\Leftrightarrow$ $\hat{h}(\boldsymbol{\omega})$ steerable

Steerable, derivative-based filters

Isotropic lowpass function (e.g., Gaussian): $\varphi(x_1, x_2)$

Subspace of steerable derivative-based templates:

$$h(x_1, x_2) = \sum_{m=0}^{M} \sum_{n=0}^{m} b_{m,n} \underbrace{\frac{\partial^m}{\partial x_1^{m-n} \partial x_2^n} \varphi(x_1, x_2)}_{\varphi_{m,n}(x_1,x_2)}$$

Basis functions

Expansion coefficients

Proposition. $h(x_1, x_2)$ is steerable in the sense that

$$(f * h(\boldsymbol{R}_\theta \cdot))(\boldsymbol{x}) = \sum_{m=0}^{M} \sum_{n=0}^{m} a_{m,n}(\theta) f_{m,n}(\boldsymbol{x})$$

where $f_{m,n}(\boldsymbol{x}) = \left(f * \frac{\partial^m}{\partial x_1^{m-n} \partial x_2^n} \varphi \right)(\boldsymbol{x})$ and $a_{m,n}(\theta) = \mathrm{poly}(\cos\theta, \sin\theta)$.

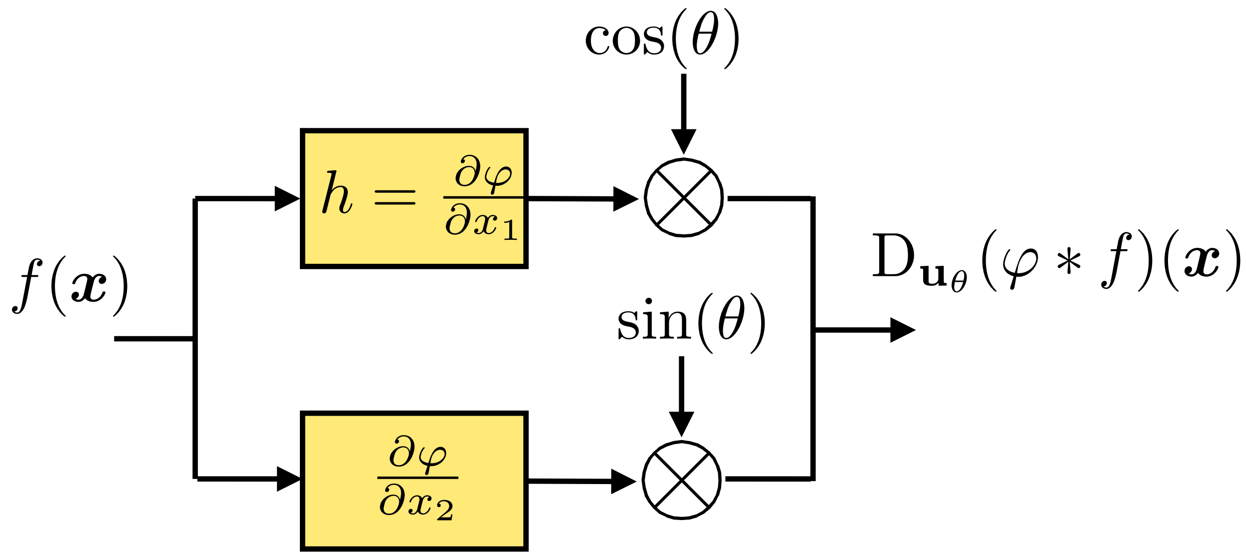
Justification: $\hat{h}(\boldsymbol{\omega}) = \hat{\varphi}(\|\boldsymbol{\omega}\|) \cdot \underbrace{\sum_{m=0}^{M} \sum_{n=0}^{m} b_{m,n} \, (j\omega_1)^{m-n} (j\omega_2)^n}_{\text{steerable polynomial}}$

Examples: Basic edge and ridge detectors

- Gradient-based edge detector

$$h(\boldsymbol{x}) = \varphi_1(\boldsymbol{x}) = \frac{\partial \varphi(\boldsymbol{x})}{\partial x_1}$$

$$\varphi_2(\boldsymbol{x}) = \frac{\partial \varphi(\boldsymbol{x})}{\partial x_2}$$



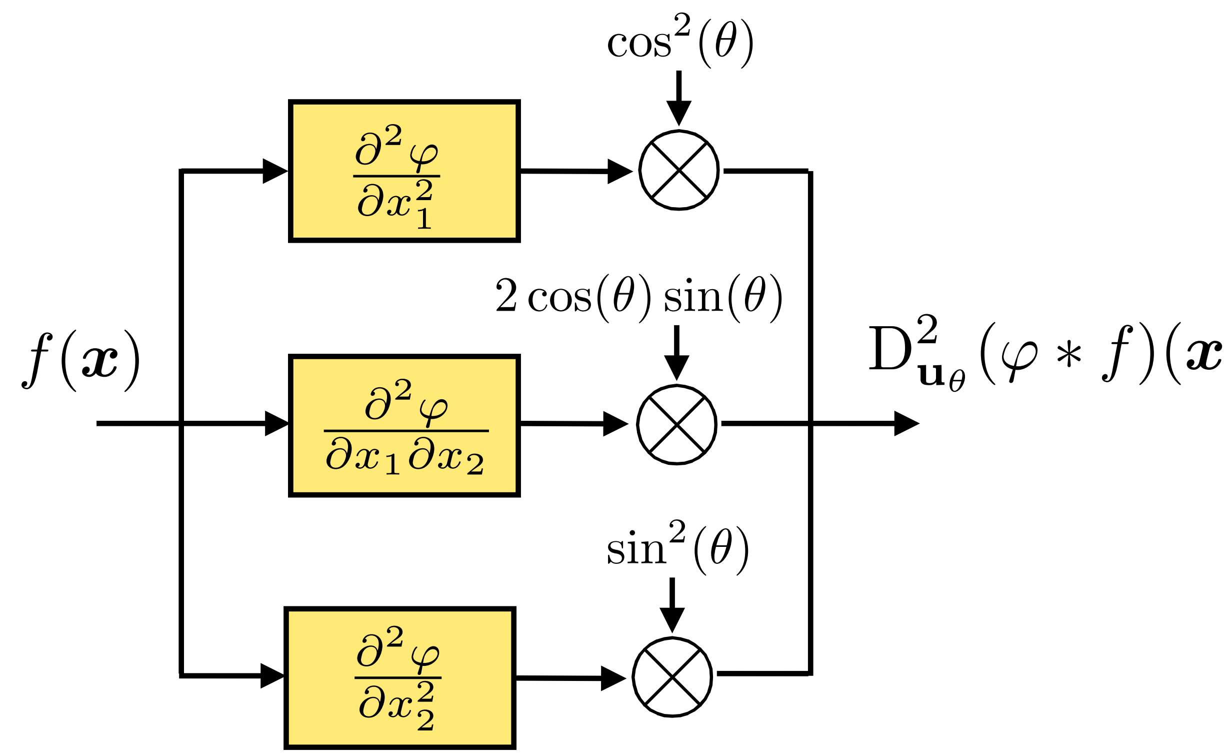
$$h(\mathbf{R}_\theta \boldsymbol{x}) = \mathrm{D}_{\mathbf{u}_\theta} \varphi(\boldsymbol{x})$$

$$= \langle \mathbf{u}_\theta, \boldsymbol{\nabla} \varphi \rangle = \cos\theta\, \varphi_1(\boldsymbol{x}) + \sin\theta\, \varphi_2(\boldsymbol{x})$$

- Hessian-based ridge detector

$$h(\boldsymbol{x}) = \varphi_{20}(\boldsymbol{x}) = \frac{\partial^2 \varphi(\boldsymbol{x})}{\partial x_1^2}$$

$$\varphi_{ij}(\boldsymbol{x}) = \frac{\partial^2 \varphi(\boldsymbol{x})}{\partial x_i \partial x_j}$$



$$h(\mathbf{R}_\theta \boldsymbol{x}) = \mathrm{D}^2_{\mathbf{u}_\theta} \varphi(\boldsymbol{x})$$

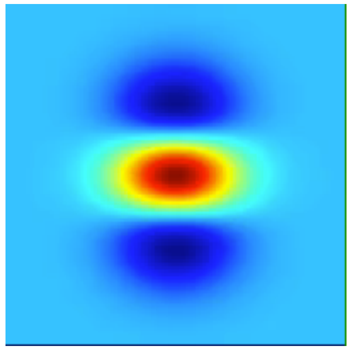
$$= (\cos\theta)^2\, \varphi_{20}(\boldsymbol{x}) + 2\cos\theta \sin\theta\, \varphi_{11}(\boldsymbol{x}) + (\sin\theta)^2\, \varphi_{02}(\boldsymbol{x})$$

Optimized steerable detectors

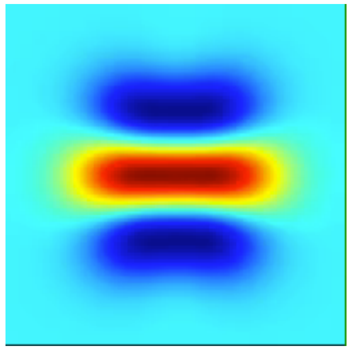
- Canny-like criterion of optimality
  - Reasonable approximation of ideal detector: $f_0(x, y) = \delta(y)$
  - Maximum signal-to-noise ratio
  - Good spatial localization
  - Reduced oscillation

Constrained optimization of $\alpha_{k,i}$ using Lagrange multipliers

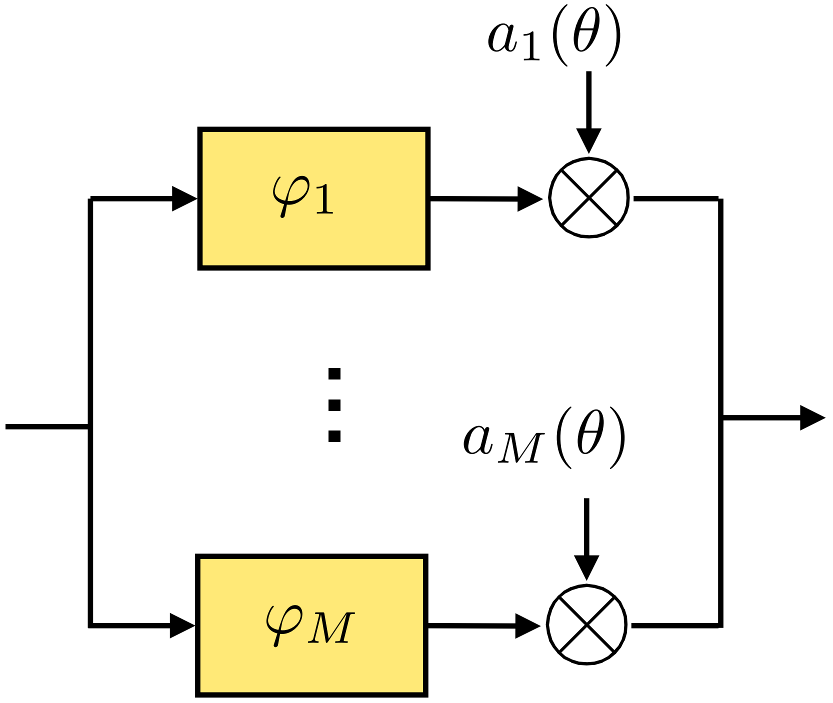
Ridge detectors



2nd order



4th order



$$\theta^*(\boldsymbol{x}) = \arg\max_{\theta} \{(h_\theta * f)(\boldsymbol{x})\}$$

thresholded magnitude

(Jacob *et al.*, IEEE-PAMI, 2004)

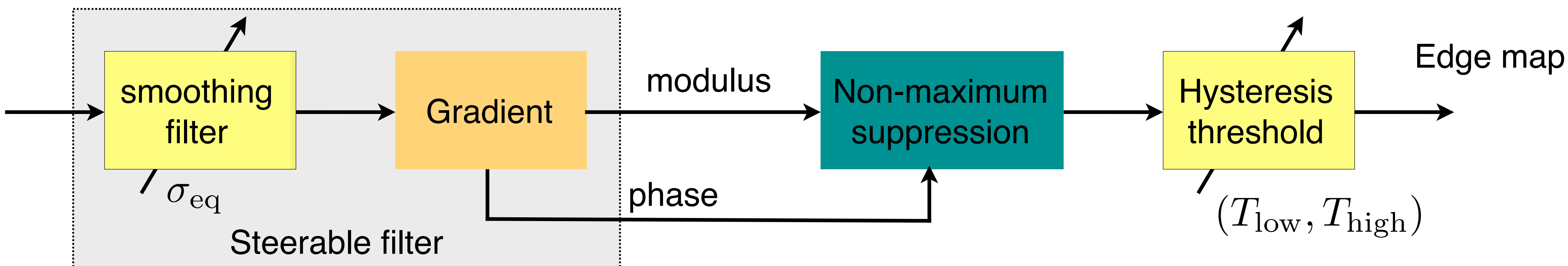
Appendix: specific implementations

- Edge detector
- Ridge detector

Edge detector: implementation

- State-of-the-art edge detector

  Edge point = local maximum of first directional derivative

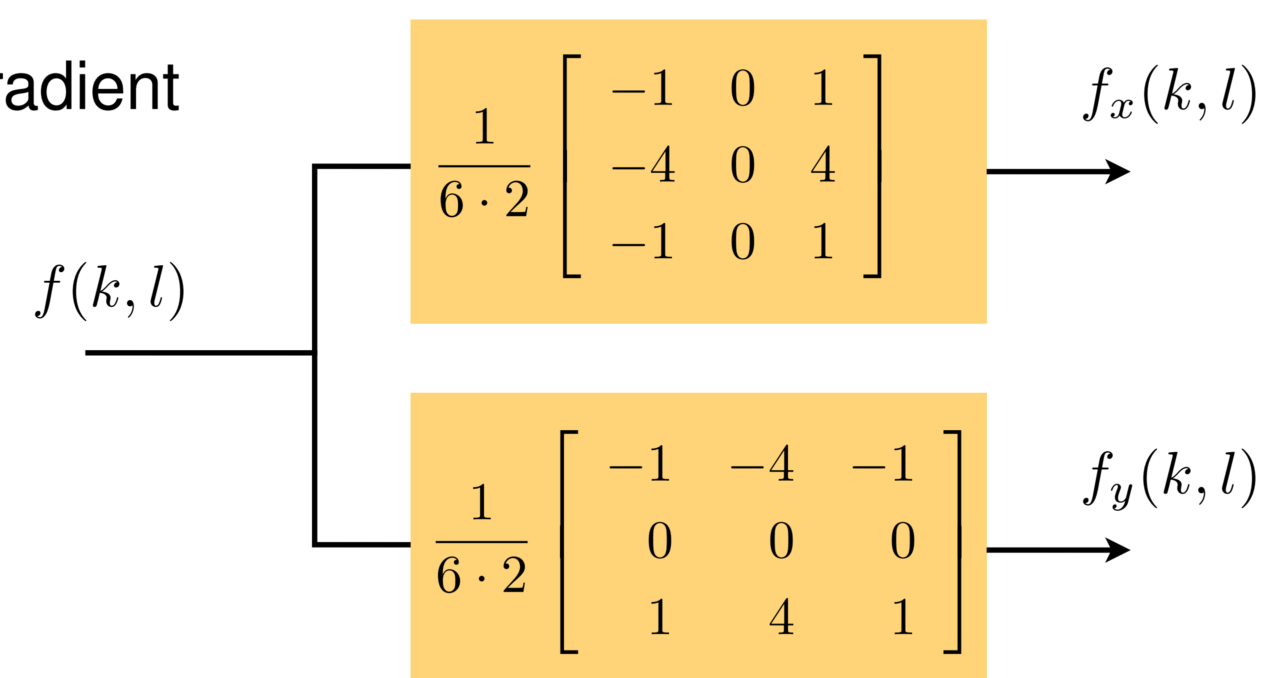


- Smoothing

  Gaussian filter: isotropic + separable

  Implementation: cascade of simple recursive filters

- Gradient



$$f_x(k,l) = \frac{1}{6\cdot 2}\begin{bmatrix} -1 & 0 & 1 \\ -4 & 0 & 4 \\ -1 & 0 & 1 \end{bmatrix} * f(k,l)$$

$$f_y(k,l) = \frac{1}{6\cdot 2}\begin{bmatrix} -1 & -4 & -1 \\ 0 & 0 & 0 \\ 1 & 4 & 1 \end{bmatrix} * f(k,l)$$

$$\boldsymbol{\nabla} f = (f_x, f_y)$$

$$\|\boldsymbol{\nabla} f\| = \sqrt{f_x^2 + f_y^2}$$

Edge detector (Cont'd)

- Non-maximum suppression

Current point: $\boldsymbol{x}_0 = (k, l)$

$\boldsymbol{u} = \dfrac{\boldsymbol{\nabla} f(\boldsymbol{x}_0)}{\|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|}$: unit vector in the direction of the gradient

If $\|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\| \geq \|\boldsymbol{\nabla} f(\boldsymbol{x}_0 \pm \boldsymbol{u})\|$ then $g(\boldsymbol{x}_0) = \|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|$

else $g(\boldsymbol{x}_0) = 0$

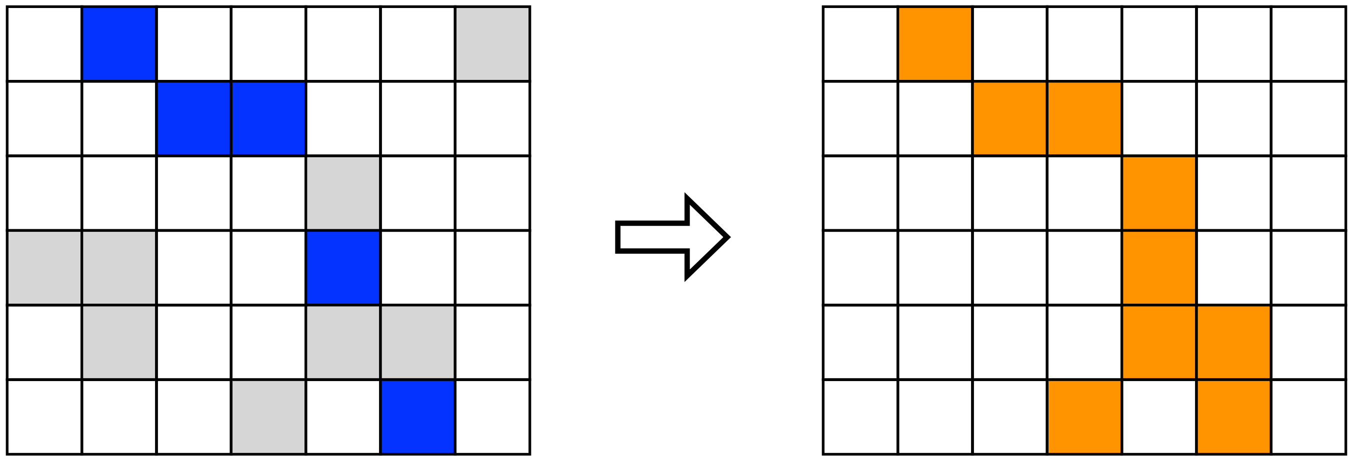
- Hysteresis threshold

Set of points: $\boldsymbol{k} = (k, l) \in \mathbb{Z}^2$

Two auxiliary edge maps:

$E_{\mathrm{low}} = \{\boldsymbol{k} \mid T_{\mathrm{low}} \leq g[\boldsymbol{k}] < T_{\mathrm{high}}\}$

$E_{\mathrm{high}} = \{\boldsymbol{k} \mid T_{\mathrm{high}} \leq g[\boldsymbol{k}]\}$



Final edge map:

$E = \{\boldsymbol{k} \in E_{\mathrm{low}} \cup E_{\mathrm{high}} \mid$ there exits a path that connects $\boldsymbol{k}$ to $E_{\mathrm{high}}\}$

Ridge or line detector

- Signature of a ridge point

Strong (negative) response to "steered" ridge detector & weak response in orthogonal direction

Smoothed Hessian matrix at location $\boldsymbol{x}$: $\mathbf{H} = \begin{bmatrix} (\varphi_{xx} * f)(\boldsymbol{x}) & (\varphi_{xy} * f)(\boldsymbol{x}) \\ (\varphi_{yx} * f)(\boldsymbol{x}) & (\varphi_{yy} * f)(\boldsymbol{x}) \end{bmatrix}$

Steering in the direction $\boldsymbol{u} = (\cos\theta, \sin\theta)$ (unit vector): $\mathrm{D}^2_{\mathbf{u}}\varphi * f = \boldsymbol{u}^T\mathbf{H}\,\boldsymbol{u}$

Extrema values:

$$\lambda_{\max} = \max_{\|\boldsymbol{u}\|=1} \{\boldsymbol{u}^T\mathbf{H}\,\boldsymbol{u}\}$$ (maximum eigenvalue of $\mathbf{H}$)

$$\lambda_{\min} = \min_{\|\boldsymbol{u}\|=1} \{\boldsymbol{u}^T\mathbf{H}\,\boldsymbol{u}\}$$ (minimum eigenvalue of $\mathbf{H}$)

$\boldsymbol{u}$ is an eigenvector of $\mathbf{H}$ with eigenvalue λ

$\Updownarrow$

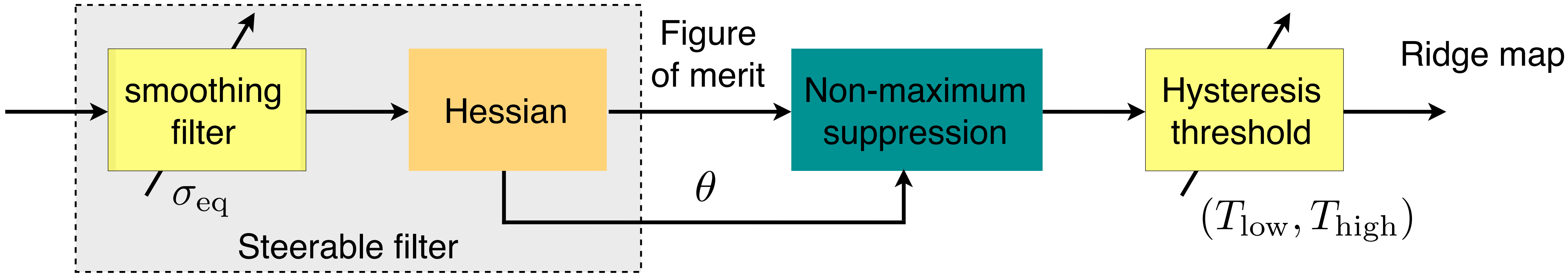
$\mathbf{H}\boldsymbol{u} = \lambda\boldsymbol{u} \quad \Rightarrow \quad \det(\mathbf{H} - \lambda\mathbf{I}) = 0$

- Positive ridge detection
  - Criterion: $\lambda_{\min} \ll 0$ and $\lambda_{\max} \approx 0$
  - Figure of merit: $g(\boldsymbol{x}) = \sqrt{|\lambda_{\min}|} \cdot \sqrt{|\lambda_{\max} - \lambda_{\min}|} \approx |\lambda_{\min}|$ (on ridge)
  - Direction perpendicular to the ridge: $\boldsymbol{u}_{\min}$ (eigenvector of $\mathbf{H}$ corresponding to $\lambda_{\min}$)

Ridge detector: implementation

- Canny-inspired ridge detector

Positive ridge point : local minimum of second derivative



Eigendecomposition of Hessian matrix $\Longrightarrow$

$$g(\boldsymbol{x}) = \sqrt{|\lambda_{\min}|} \cdot \sqrt{|\lambda_{\max} - \lambda_{\min}|}$$

$$\boldsymbol{u}_{\min} = (\cos\theta, \sin\theta)$$

- Hessian masks

$$\frac{\partial^2}{\partial x^2} : \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ 4 & -8 & 4 \\ 1 & -2 & 1 \end{bmatrix}$$

$$\frac{\partial^2}{\partial x \partial y} : \frac{1}{2 \cdot 2} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial^2}{\partial y^2} : \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 \\ -2 & -8 & -2 \\ 1 & 4 & 1 \end{bmatrix}$$

References

- A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press, 1987.
- W. Forstner, “A Feature Based Correspondence Algorithm for Image Processing,” *Int. Archives of Photogrammetry and Remote Sensing*, vol.26, pp.150-166, 1986.
- W.T. Freeman and E.H. Adelson, “The Design and Use of Steerable Filters,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, Sept. 1991.
- J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.
- M. Jacob, M. Unser, “Design of Steerable Filters for Feature Detection Using Canny-Like Criteria,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1007-1019, August 2004.