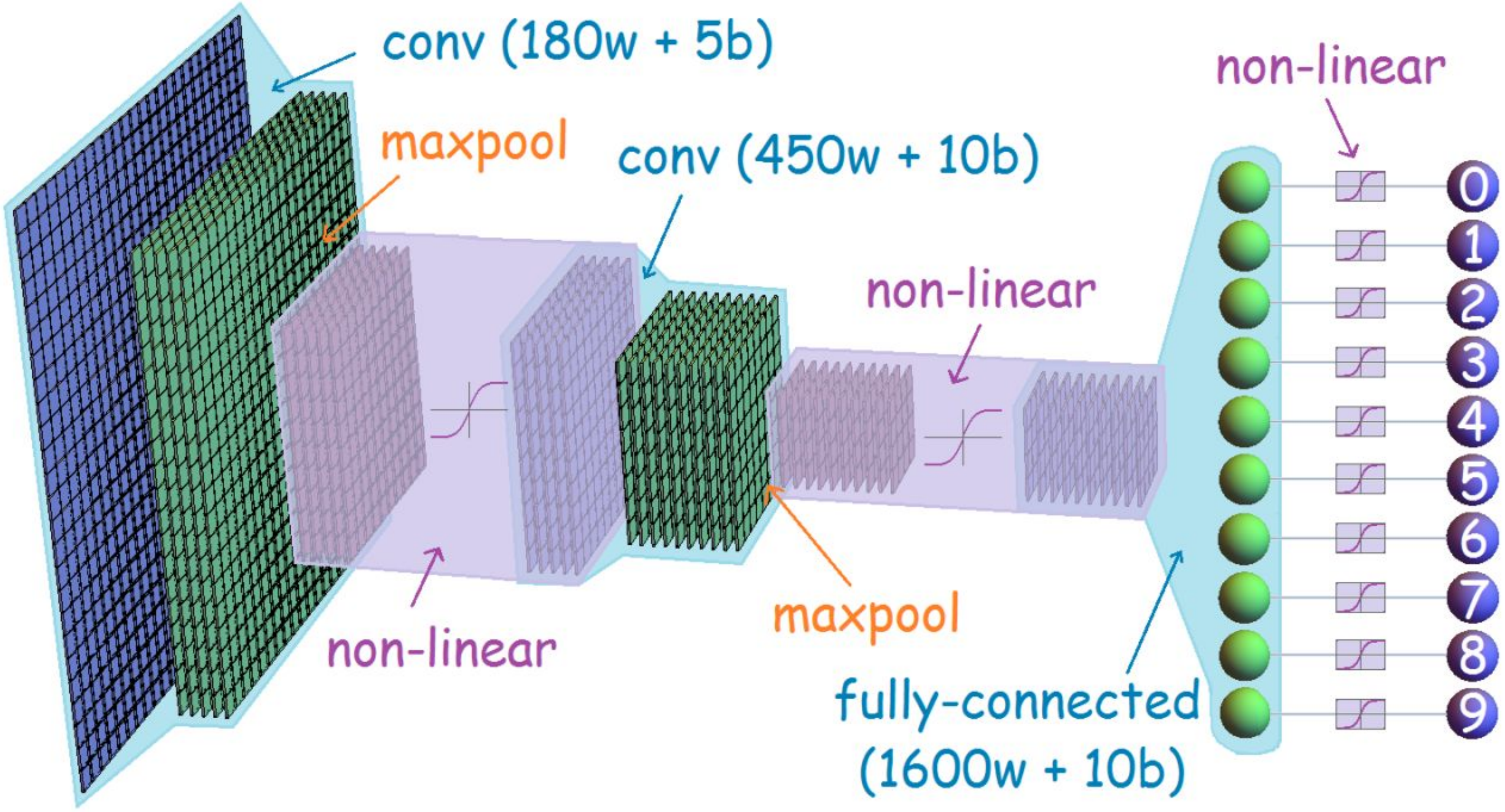
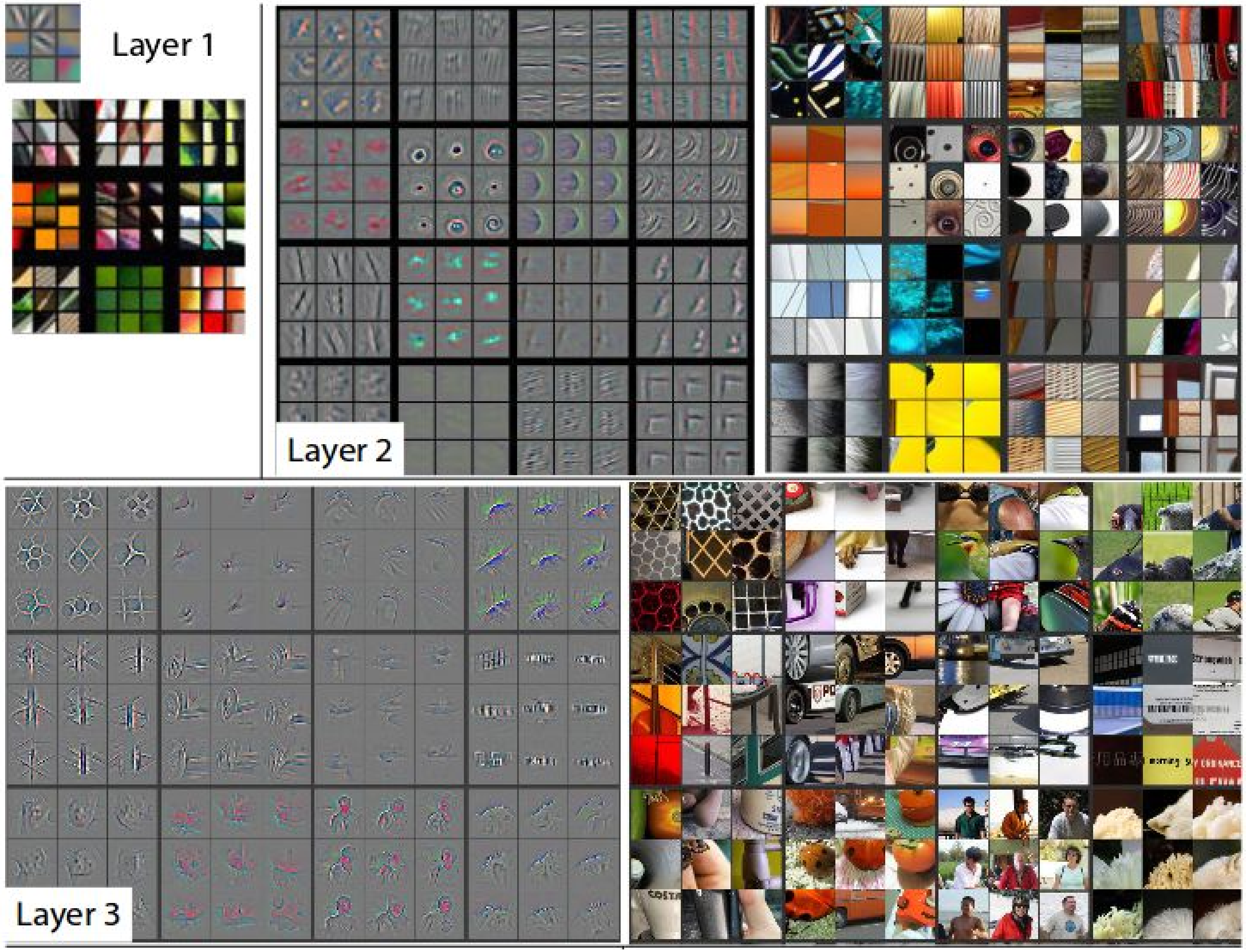
Chapter 6: Local Structure

Leo Dorst & Rein van den Boomgaard

Convolutional Neural Networks





(Zeiler & Fergus 2014] use a deconvnet to visualize features by reconstructing the top 9 patterns that cause the highest activations in a given feature map as well as showing the corresponding image patch.

Correlation

- When doing the *facet model of interpolation* we found the need to multiply an image by a local coefficient schema:

$$\frac{1}{9}\left\{\begin{array}{rcr} -1 & 2 & -1 \\ 2 & \underline{5} & 2 \\ -1 & 2 & -1 \end{array}\right\}$$

$$\frac{1}{6}\left\{\begin{array}{rcr} -1 & 0 & 1 \\ -1 & \underline{0} & 1 \\ -1 & 0 & 1 \end{array}\right\} \qquad \frac{1}{6}\left\{\begin{array}{rcr} 1 & 1 & 1 \\ 0 & \underline{0} & 0 \\ -1 & -1 & -1 \end{array}\right\}$$

$$\frac{1}{6}\left\{\begin{array}{rcr} 1 & -2 & 1 \\ 1 & \underline{-2} & 1 \\ 1 & -2 & 1 \end{array}\right\} \qquad \frac{1}{6}\left\{\begin{array}{rcr} 1 & 1 & 1 \\ -2 & \underline{-2} & -2 \\ 1 & 1 & 1 \end{array}\right\} \qquad \frac{1}{4}\left\{\begin{array}{rcr} -1 & 0 & 1 \\ 0 & \underline{0} & 0 \\ 1 & 0 & -1 \end{array}\right\}$$

Correlation

- Such a point-wise multiplication of two ‘images’ is called a *correlation*.
- Example for an average horizontal derivative, in the facet model:

$$p_2 = \frac{1}{6}\left\{\begin{array}{ccc} -1 & 0 & 1 \\ -1 & \underline{0} & 1 \\ -1 & 0 & 1 \end{array}\right\}$$

- Looks straightforward:

$$G(i,j) = \sum_{k=-1}^{1}\sum_{l=-1}^{1} F(i+k,j+l)\,W(k,l).$$

Correlation and Convolution

- Correlation:

$$G(i,j) = \sum_{k=-1}^{1} \sum_{l=-1}^{1} F(i+k, j+l)\, W(k,l).$$

- Convolution:

$$G(i,j) = \sum_{k=-1}^{1} \sum_{l=-1}^{1} F(i-k, j-l)\, W(k,l).$$

Convolution is correlation with point-mirrored version of coefficients.

$$(f * g)(i,j) = \sum_{k'=-1}^{1} \sum_{\ell'=-1}^{1} f(i+k', j+\ell')\, g(-k', -\ell')$$

$$= \sum_{k=-1}^{1} \sum_{\ell=-1}^{1} f(i-k, j-\ell)\, g(k, \ell)$$

$$\left\{ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & \underline{0} & 1 \\ 0 & 0 & 0 \end{array} \right\}$$

g for translation
1 pixel to the right
by correlation.

$$\left\{ \begin{array}{ccc} 0 & 0 & 0 \\ 1 & \underline{0} & 0 \\ 0 & 0 & 0 \end{array} \right\}$$

g for translation
1 pixel to the right
by convolution.

Algebraic properties of convolution

- $$f * (\alpha g) = \alpha(f * g), \quad \alpha \in \boldsymbol{R}$$
  $$f * (g_1 + g_2) = f * g_1 + f * g_2$$

$$f * (g * h) = (f * g) * h$$

$$f * g = g * f$$

Correlation does not have such nice properties. That is why everybody prefers the convolution formulation. Better get used to it.

Convolution (section 6.2)

- All *translation-invariant linear operators* between images *f* and *g* can be written as a convolution:

$$(f * g)(\mathbf{x}) = \sum_{\mathbf{y} \in E} f(\mathbf{x} - \mathbf{y})\, g(\mathbf{y})$$

- Examples: smoothing, sharpening, translation, n^{th} order derivatives, …
- The basis operation in the very successful CNNs !

Convolution by hand:

- Simple moving average

$$\frac{1}{9}\left\{\begin{array}{ccc} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{array}\right\}$$

- Simple derivative (from facet model)

$$\frac{1}{6}\left\{\begin{array}{ccc} 1 & 0 & -1 \\ 1 & \underline{0} & -1 \\ 1 & 0 & -1 \end{array}\right\}$$

- Translation

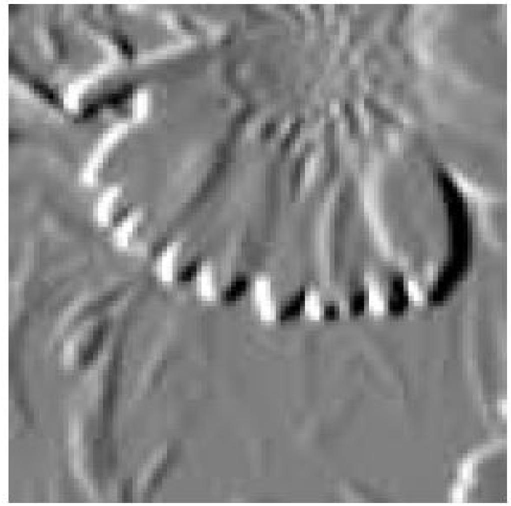
$$\left\{\begin{array}{ccc} 0 & 0 & 0 \\ 0 & \underline{0} & 1 \\ 0 & 0 & 0 \end{array}\right\}$$

Python examples

- https://staff.fnwi.uva.nl/r.vandenboomgaard/IPCV20162017/LectureNotes/IP/LocalOperators/convolutionExamples.html#convolutionexamples

Derivatives at a spatial scale

f_0



f_x

f_y

- (derivative (blur(*f*)) = (blurred derivative) *f*

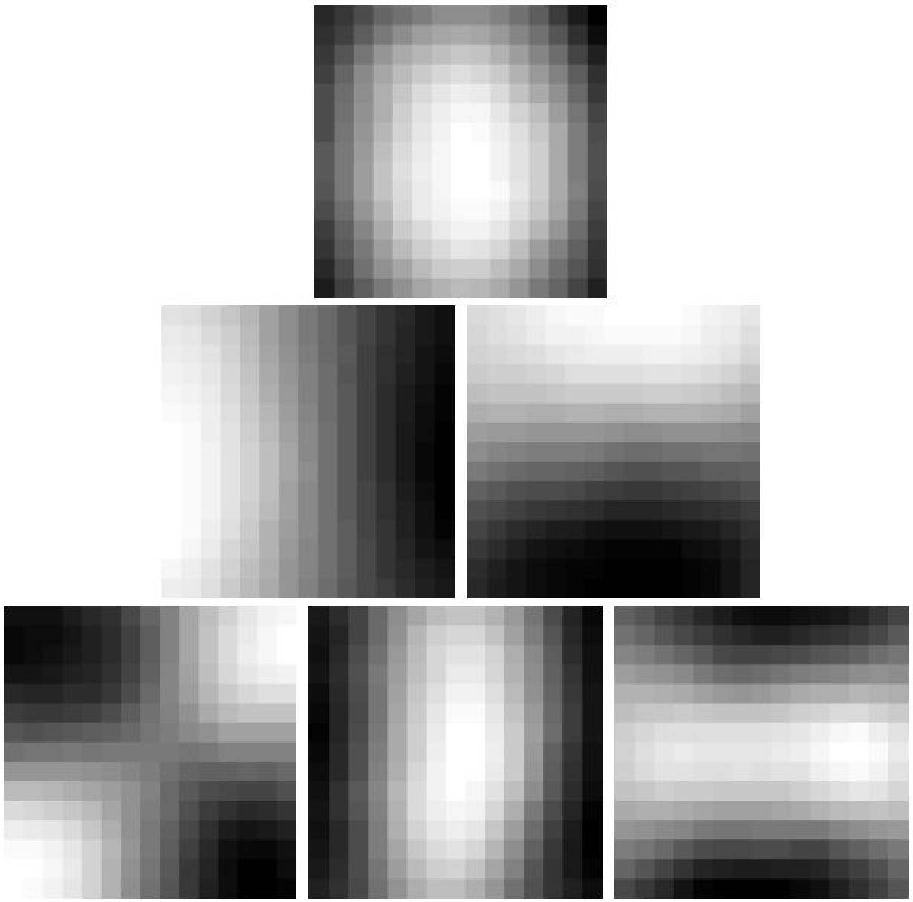
$$D * (B * f) = (D * B) * f = (B * D) * f$$

Local Image Structure

Local image structure can be characterized as

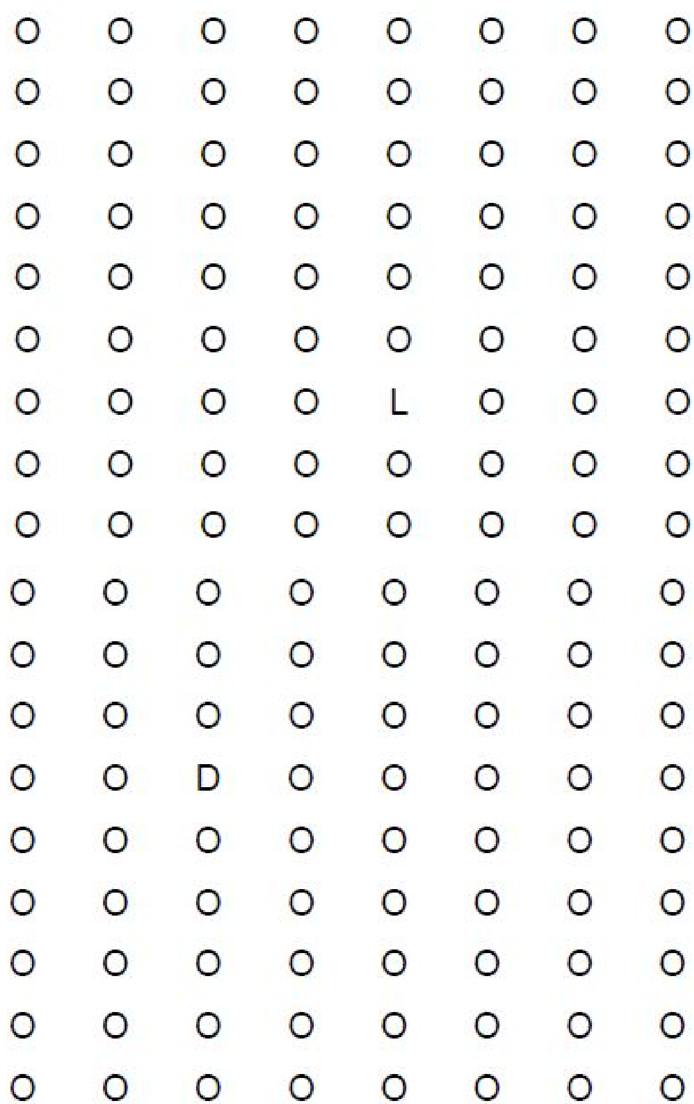
- image patches with **constant** grey value
- image patches showing a straight **edge** (transition from dark to bright regions)
- image patches showing **corners** (strongly curved edges)
- image patches showing dark (bright) **lines** on a bright (dark) background
- **T-juntions**
- **textured** image patches
- and many more. . .

Learning Local Image Structure



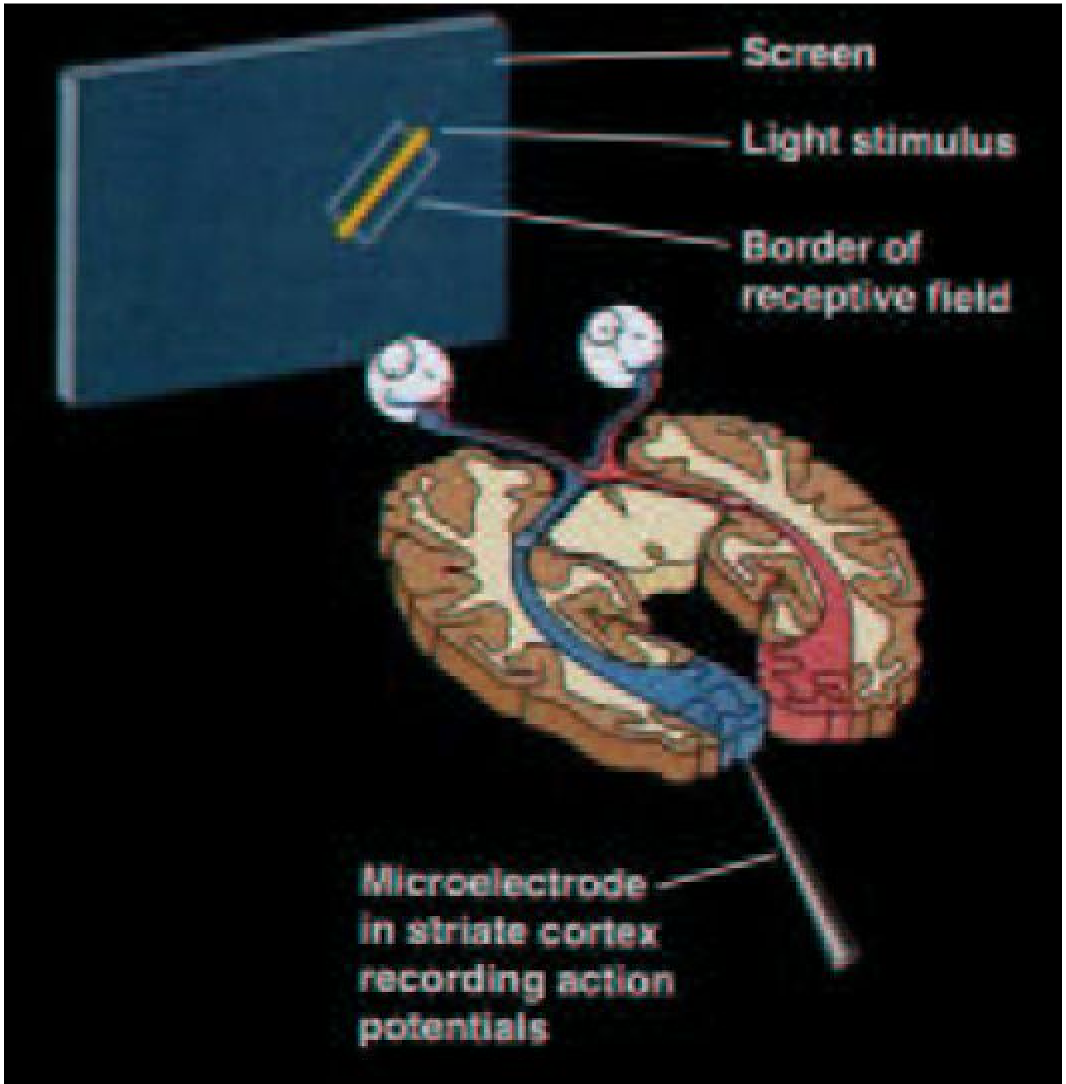
A local image patch (say N pixels) is a vector in N-dimensional 'image space'. A principal component analysis £nds the basis in image space that best suits the observed image patches in the sense that only a few basis vectors in that new basis for image space are really important.

Local Structure Detection in the Brain



The ‘L’ among all the ‘O’s is much easier to spot than the ‘D’, but more surprisingly the time it takes to spot the ‘L’ is independent of the number of ‘O’s surrounding it.

Measuring Local Structure Detectors in the Brain



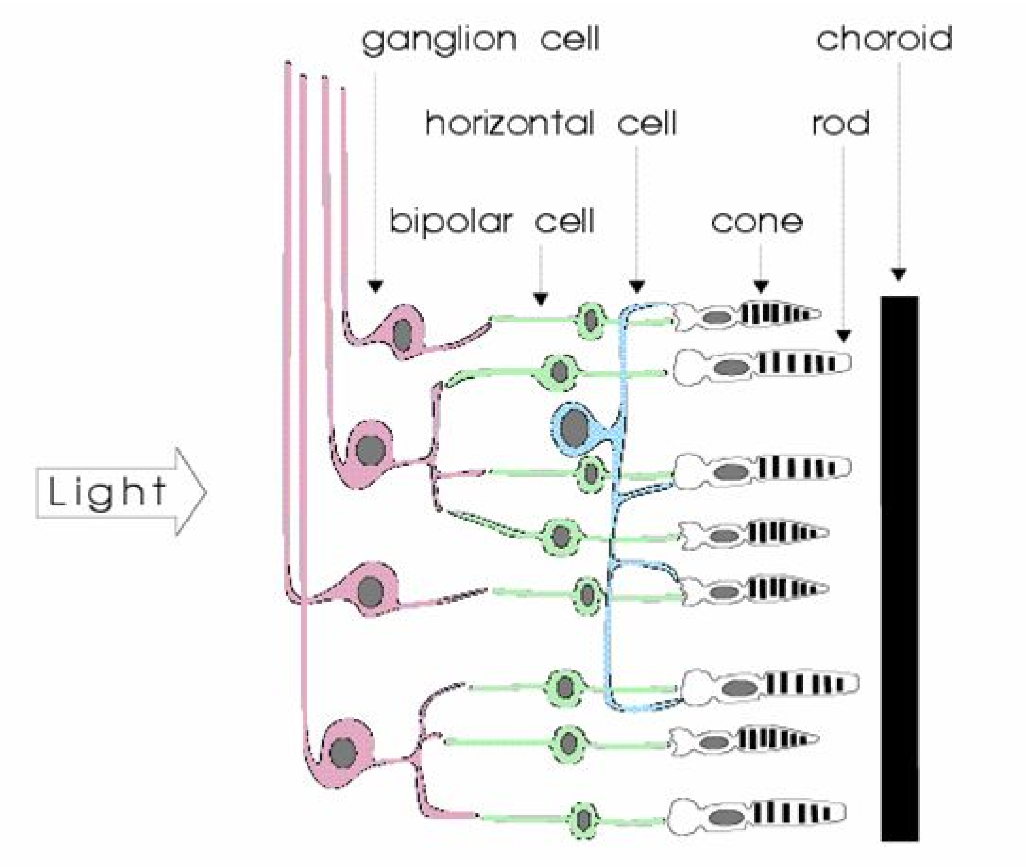
Hubel and Wiesel were the first to measure the response of neurons in the visual cortex to visual input in-vivo [1959].

Their measurements indicate the human visual system is 'hard-wired' to recognize specific details.

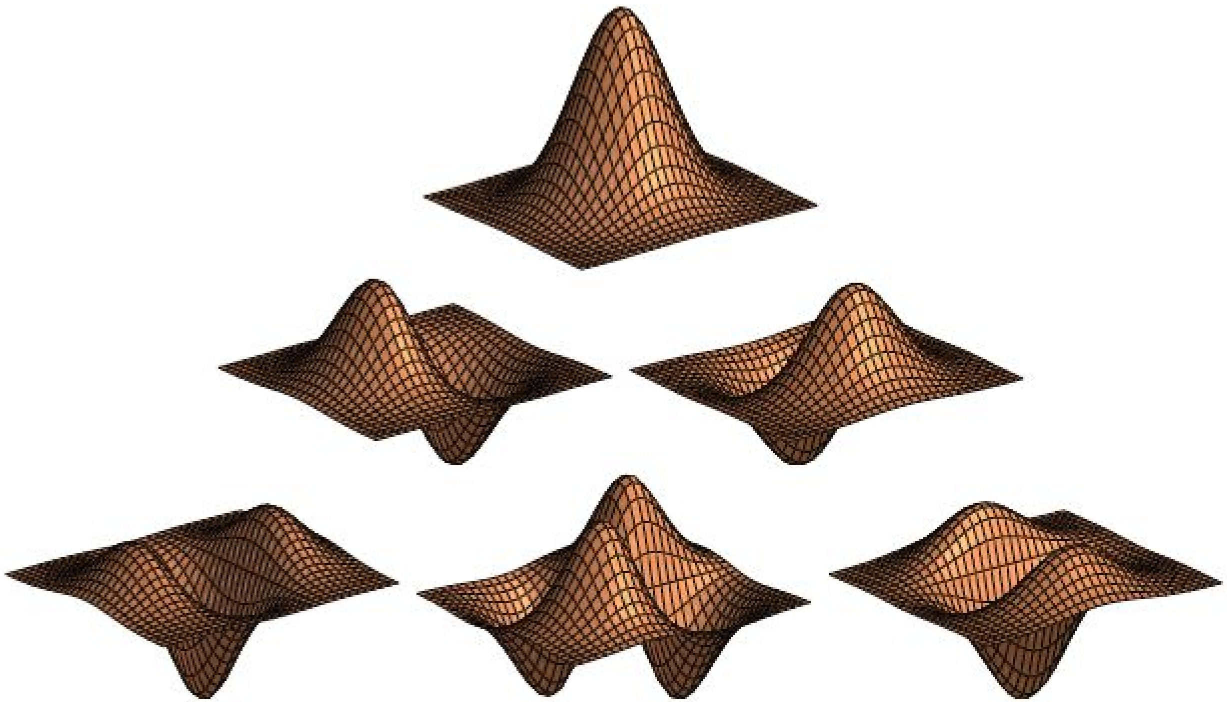
These details show great resemblance with the local details as those learned with a PCA.

The human visual brain thus has adapted itself to the visual stimuli that it is likely to see.

retinal processing



Local Structure Detection from Basic Principles



Prof. J. Koenderink (Utrecht University) showed [1980s] that the local image details as detected by the human brain also resemble the details that follow from a mathematical analysis based on basic (symmetry and causality) principles.

In these lecture series (a simplified version of) the mathematical theory is bluntly stated without formal derivation from basic principles.

Convolution

All *linear translation-invariant* operations on signals/images can be written as a *convolution with a specific kernel:*

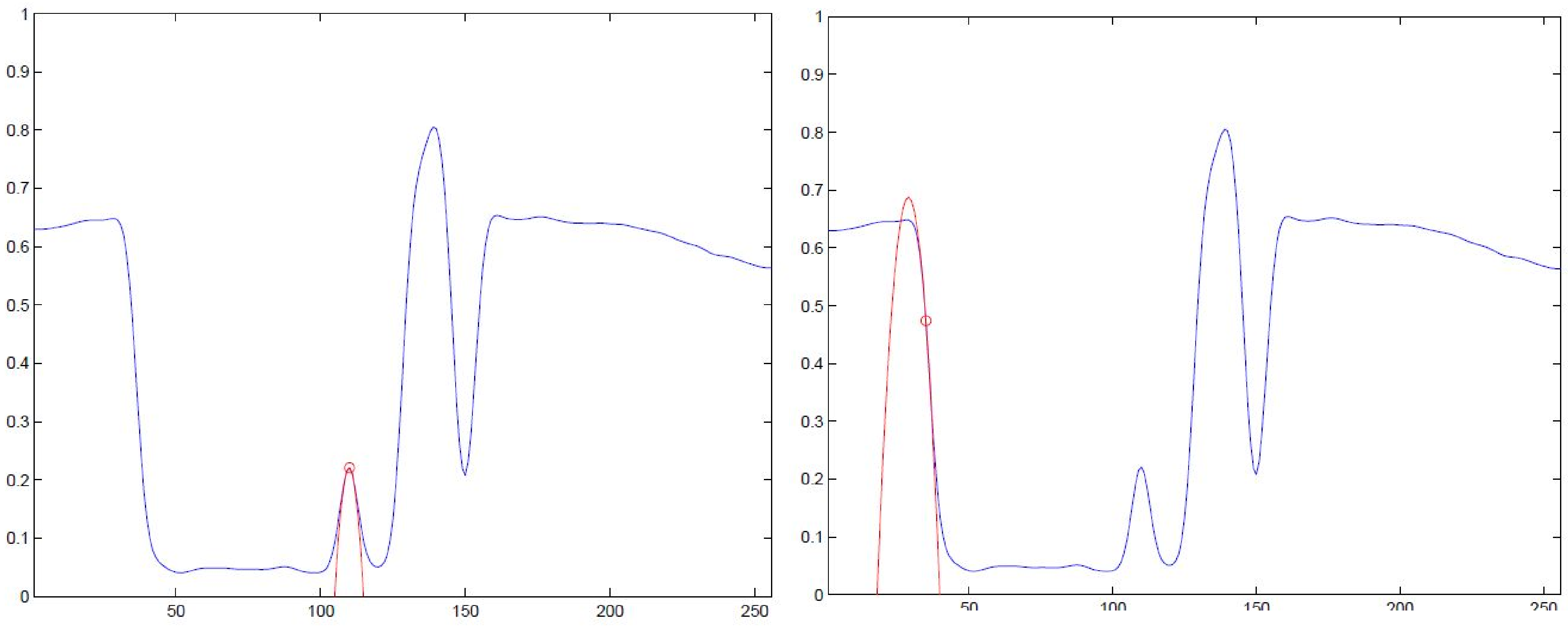
$$(f * g)(i,j) = \sum_k \sum_\ell f(i-k, j-\ell)\, g(k,\ell)$$

Let us look at that in detail (on the board).

Described in section 6.2.2 (and Appendix A).

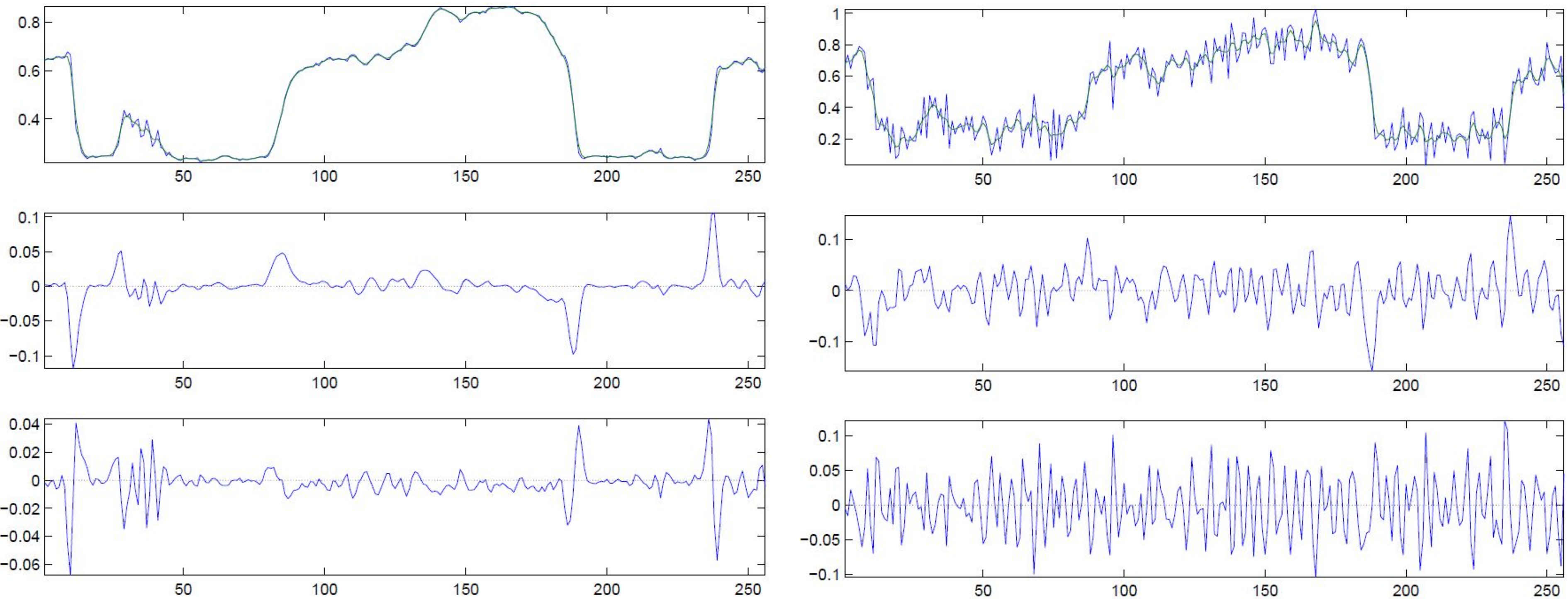
Local Taylor Series in 1D

$$f(a+x) \approx f(a) + xf'(a) + \frac{1}{2}x^2 f''(a)$$



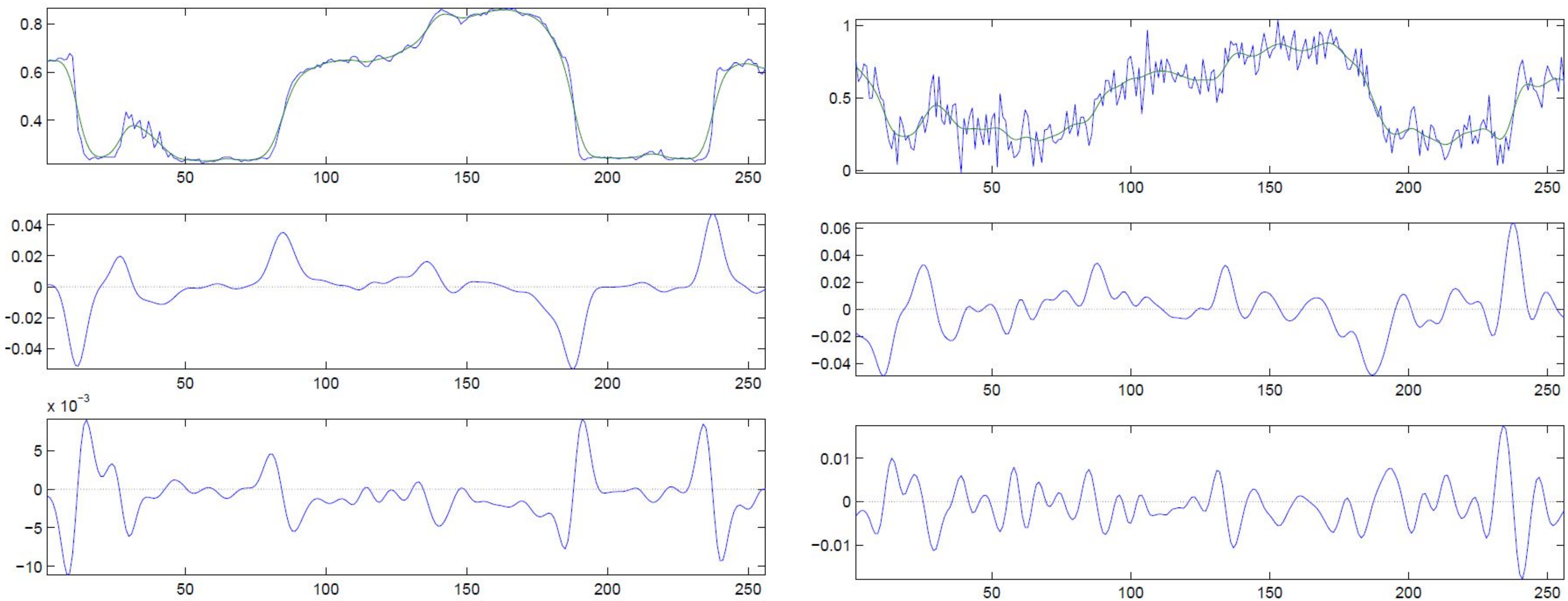
Gaussian Derivatives at Different Scales and Noise

- Scale 1



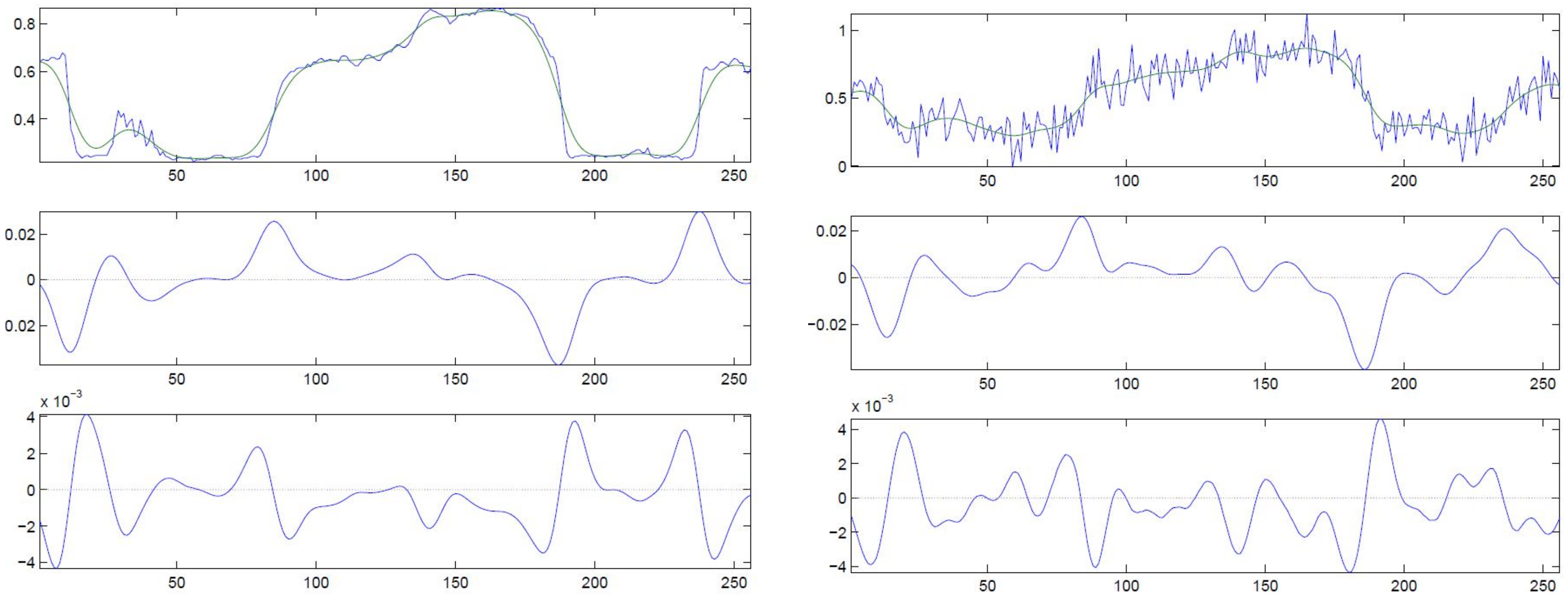
Gaussian Derivatives at Different Scales and Noise

- Scale 3



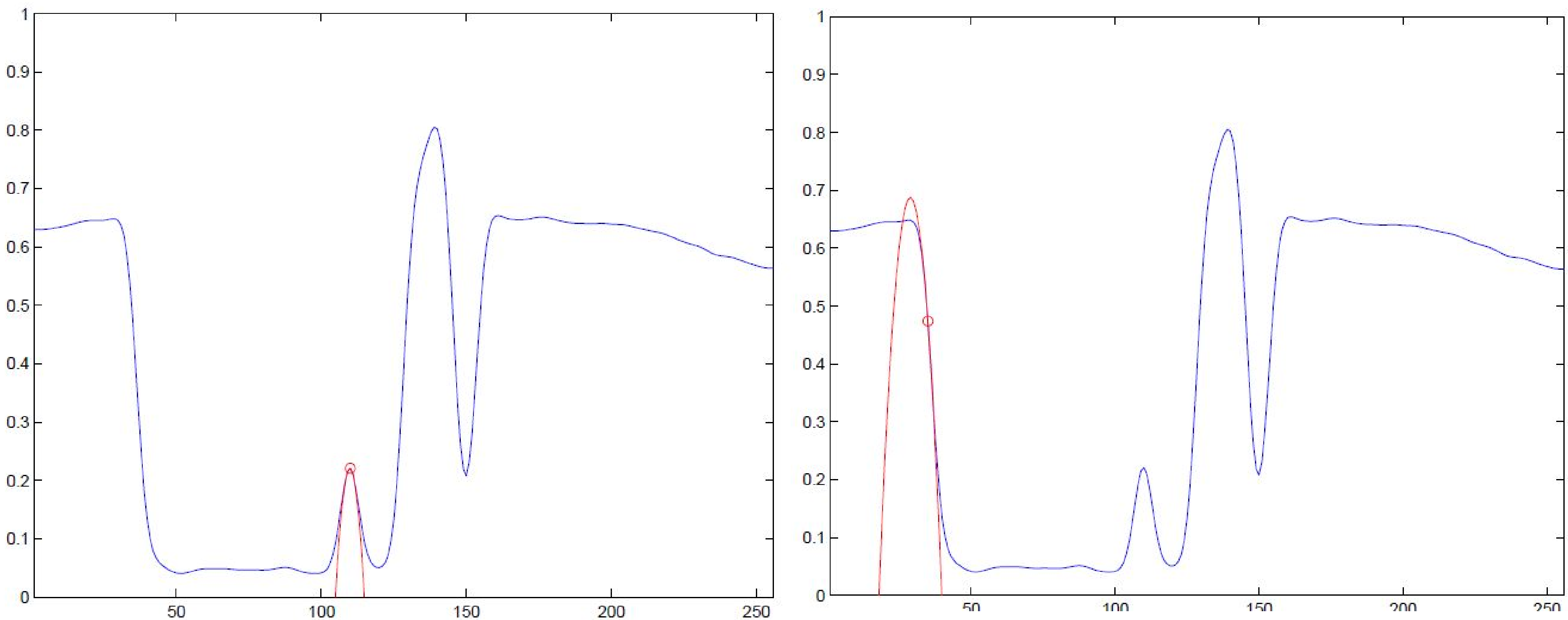
Gaussian Derivatives at Different Scales and Noise

- Scale 5



Local Taylor Series in 1D

$$f(a+x) \approx f(a) + xf'(a) + \frac{1}{2}x^2 f''(a)$$



Local Taylor Series in 2D

$$f(\mathbf{a}_X + x, \mathbf{a}_Y + y) \approx$$
$$\begin{aligned}\approx\ & f(\mathbf{a}_X, \mathbf{a}_Y) \\ & + x f_x(\mathbf{a}_X, \mathbf{a}_Y) + y f_y(\mathbf{a}_X, \mathbf{a}_Y) \\ & + \tfrac{1}{2}x^2 \frac{\partial^2 f}{\partial x^2}(\mathbf{a}_X, \mathbf{a}_Y) + xy \frac{\partial^2 f}{\partial x \partial y}(\mathbf{a}_X, \mathbf{a}_Y) + \tfrac{1}{2}y^2 \frac{\partial^2 f}{\partial y^2}(\mathbf{a}_X, \mathbf{a}_Y).\end{aligned}$$

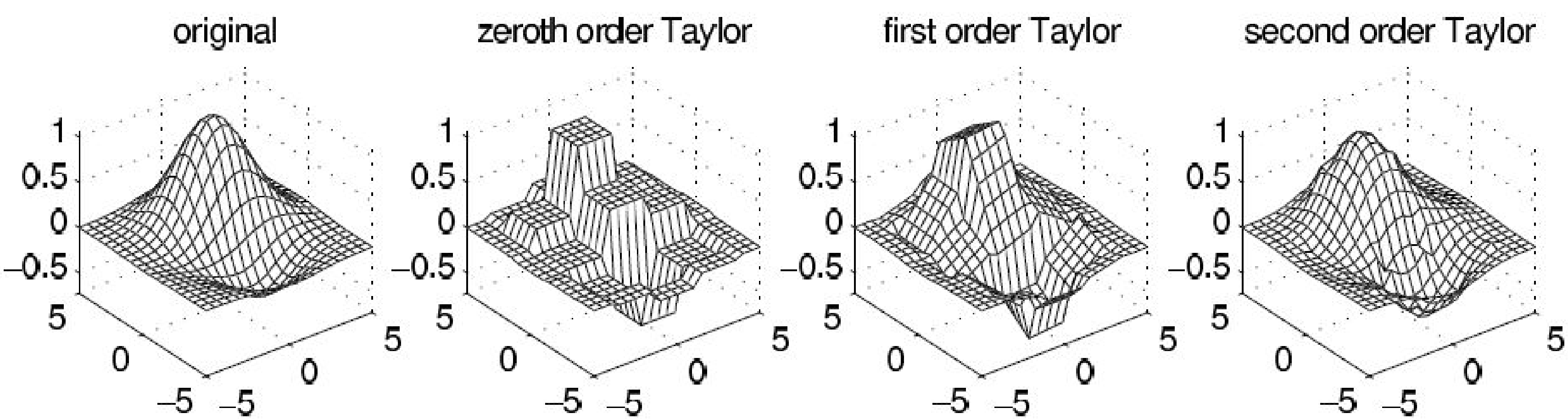


Figure 6.2: **Approximations of an image function** through Taylor approximations in local neighborhoods: zeroth order, first order, second order.

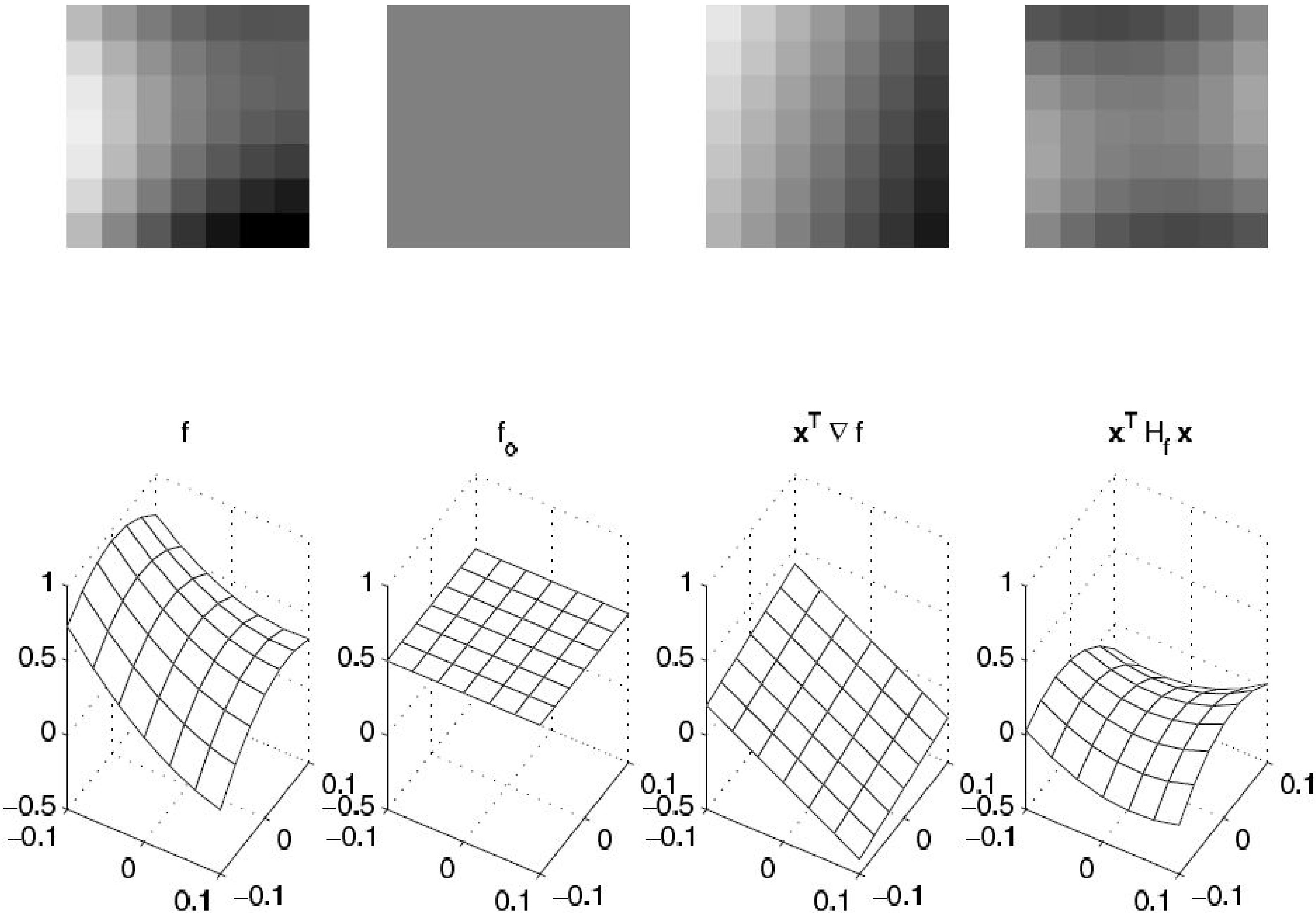


Figure 6.1: **Local Taylor approximations of a neighborhood f**: original f, zeroth order term f_0 (this has constant gray value), first order term $\mathbf{x}^\mathsf{T}\boldsymbol{\nabla} f$, second order term $\mathbf{x}^\mathsf{T} H_f \mathbf{x}$ (those two are depicted with 0 represented by grey).

Local Taylor Series in 2D

$$f(\mathbf{a}+\mathbf{x}) \approx f(\mathbf{a}) + \mathbf{x}^\top \nabla f(\mathbf{a}) + \frac{1}{2}\mathbf{x}^\top H_f(\mathbf{a})\mathbf{x}$$

where

$$\nabla f(\mathbf{a}) = \begin{pmatrix} f_x(\mathbf{a}) \\ f_y(\mathbf{a}) \end{pmatrix}$$

is the *gradient vector* and

$$H_f(\mathbf{a}) = \begin{pmatrix} f_{xx}(\mathbf{a}) & f_{xy}(\mathbf{a}) \\ f_{xy}(\mathbf{a}) & f_{yy}(\mathbf{a}) \end{pmatrix}$$

is the *Hessian matrix*.

Gradient

$$(\nabla f)(\mathbf{a}) = \begin{pmatrix} f_x(\mathbf{a}) \\ f_y(\mathbf{a}) \end{pmatrix}$$

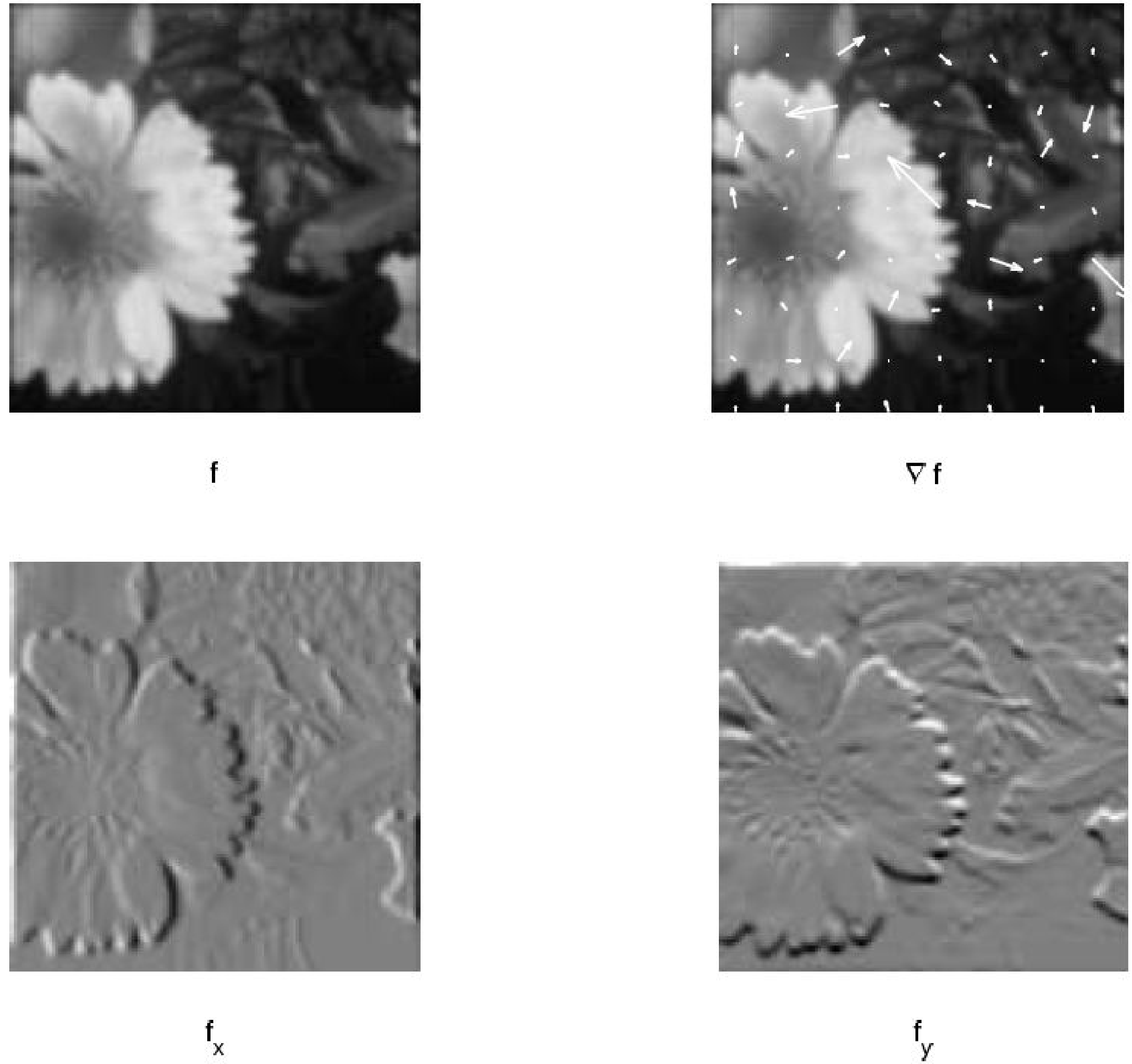
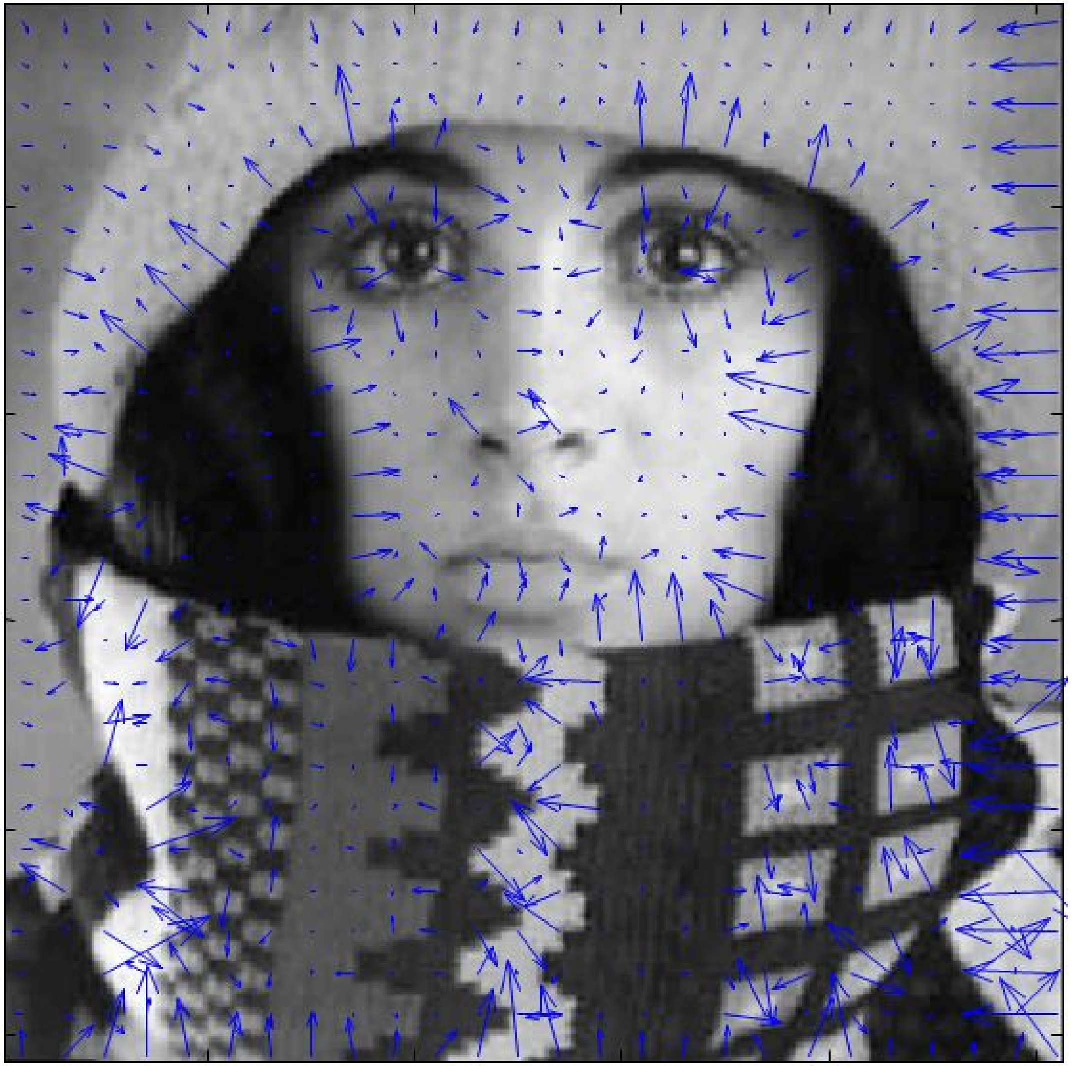
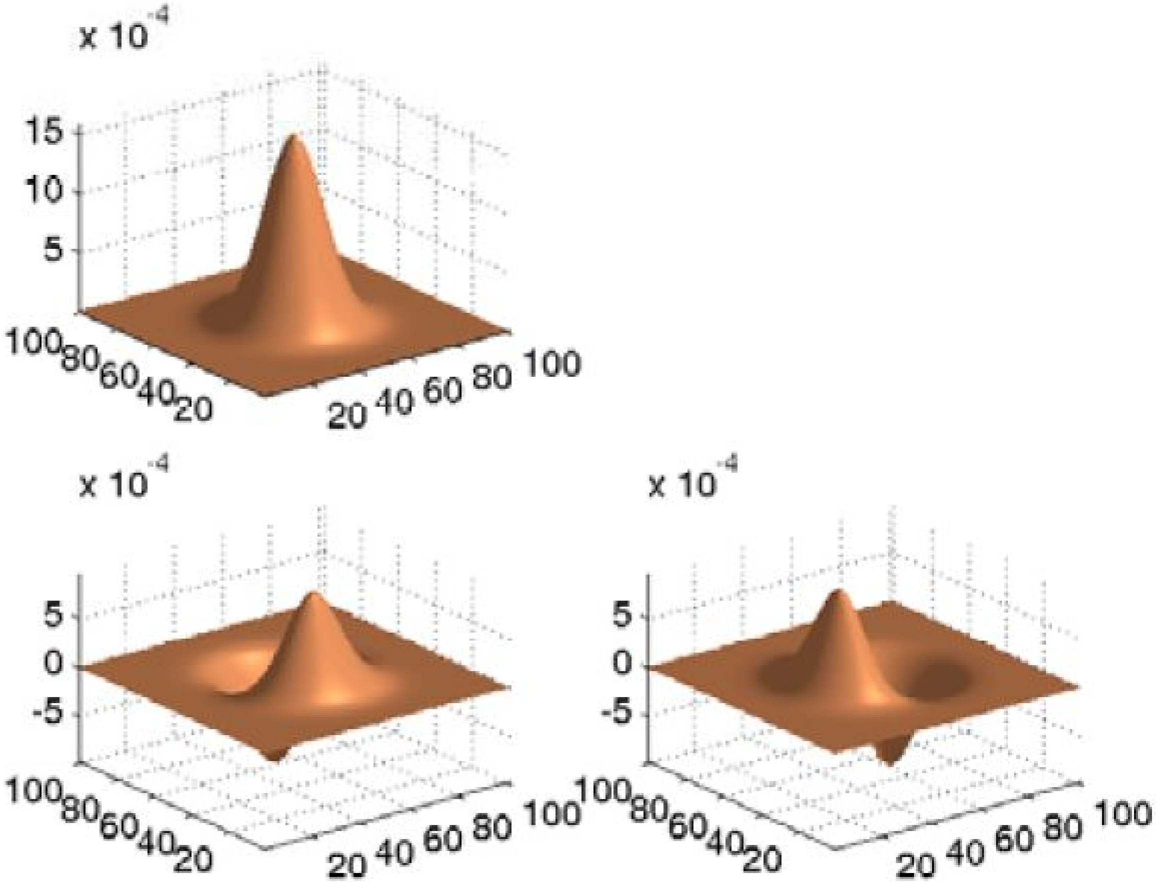


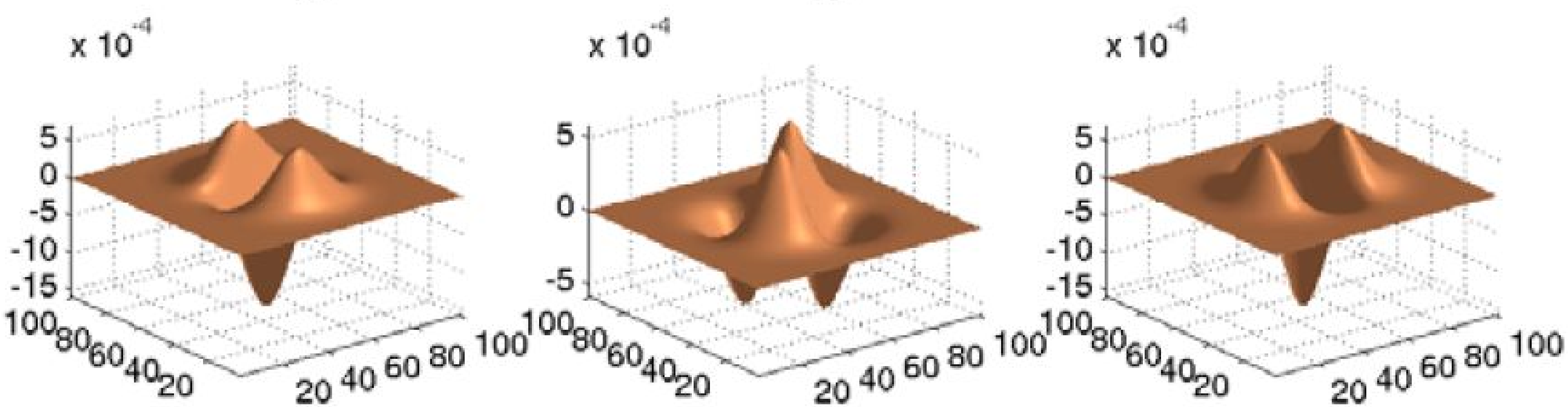
Figure 6.3: **Gradient.** An image f and its gradient: a vector-valued image of vectors ∇f (only a few are indicated). The components of ∇f on the (x, y)-frame are depicted as the gray-valued images f_x and f_y, with gray denoting zero, positive values bright and negative values dark.

Gradient: The Vector of Local Change

Gradient: The Vector of Local Change







We can/will use Gaussian Derivatives to determine the required local derivatives for the Taylor series in a structural and computationally stable manner.

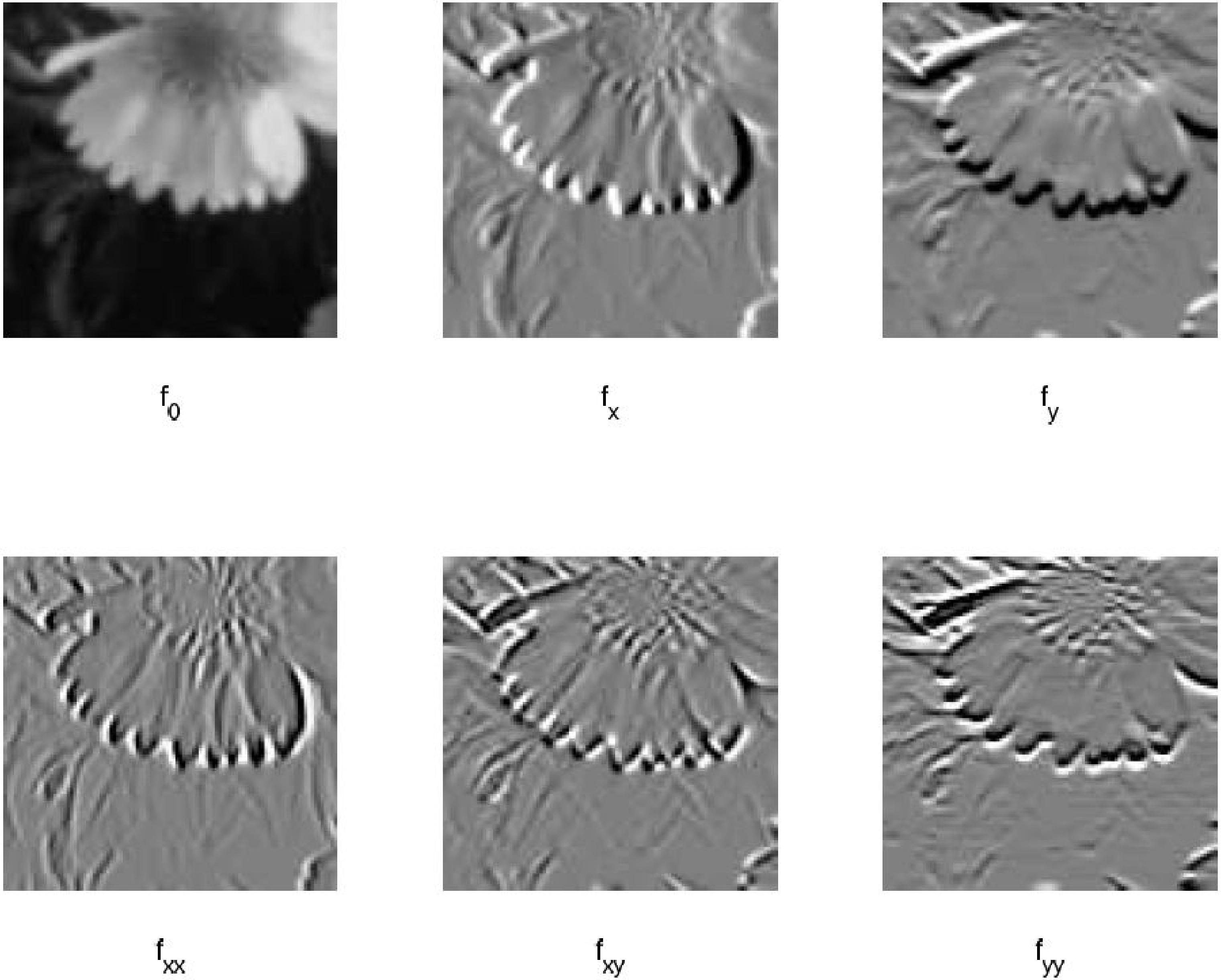


Figure 6.8: **Derivatives.** The derivatives along the (x, y)-directions, at a scale of 1.

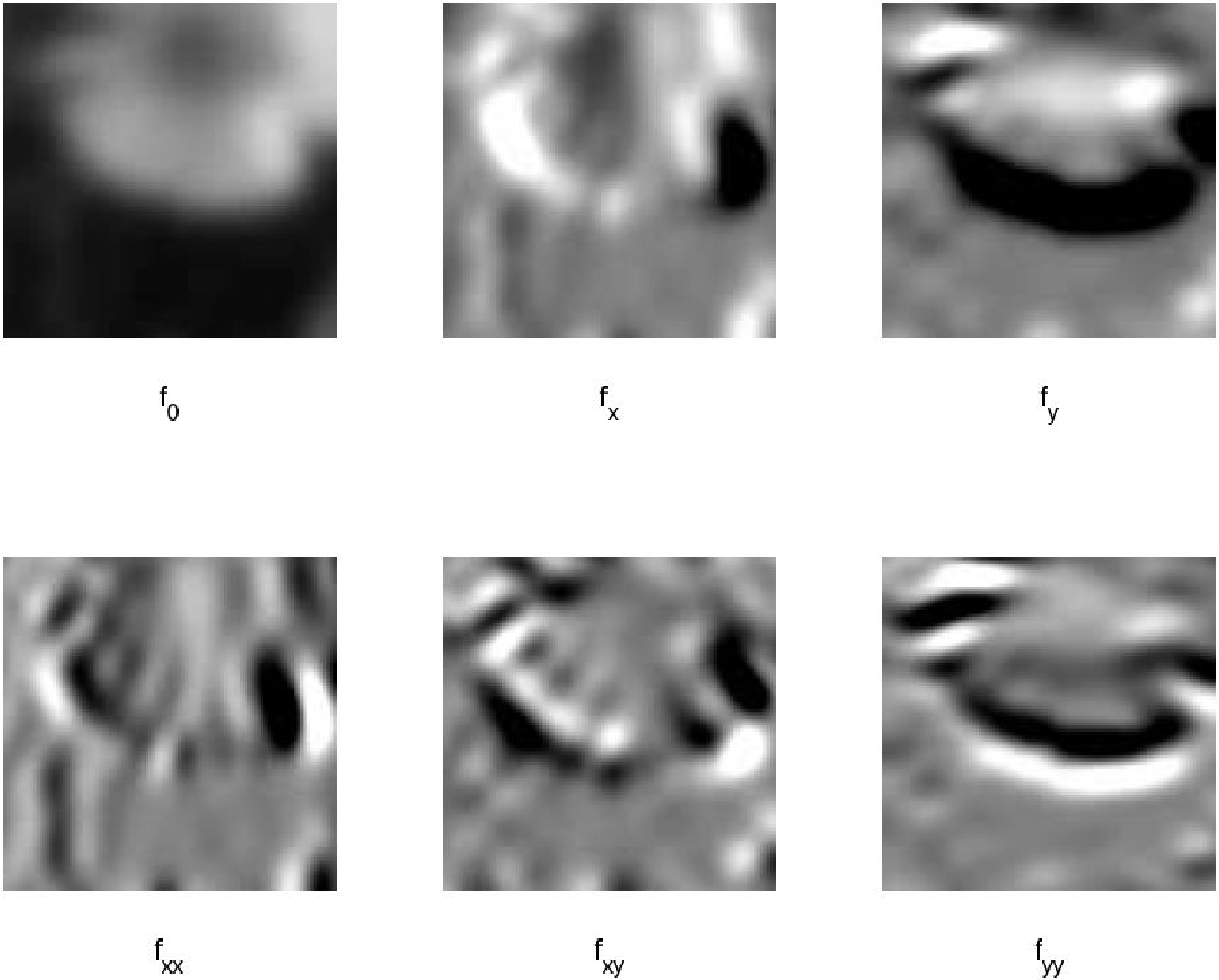


Figure 6.9: **Derivatives.** The derivatives along the (x, y)-directions, at a scale of 5.

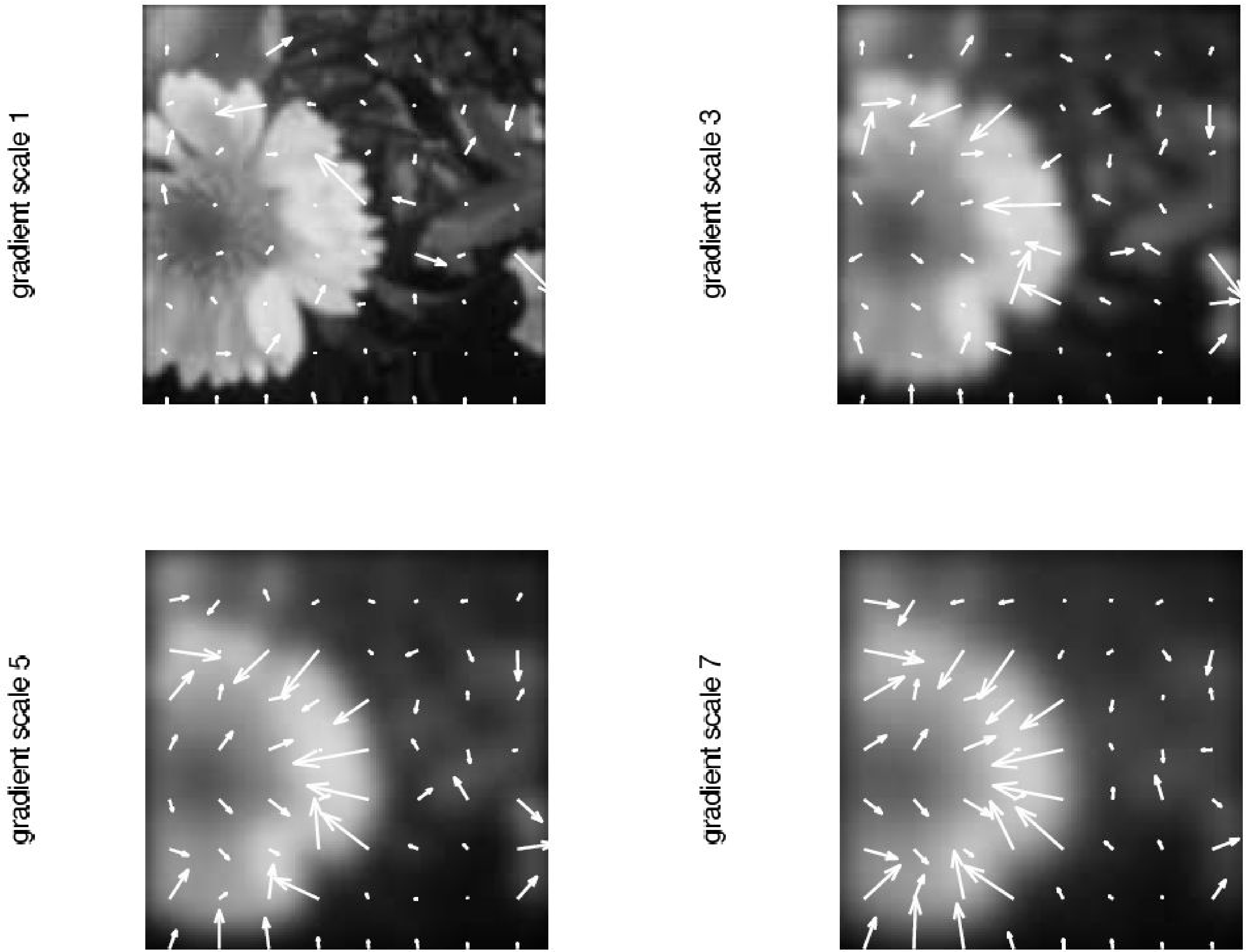
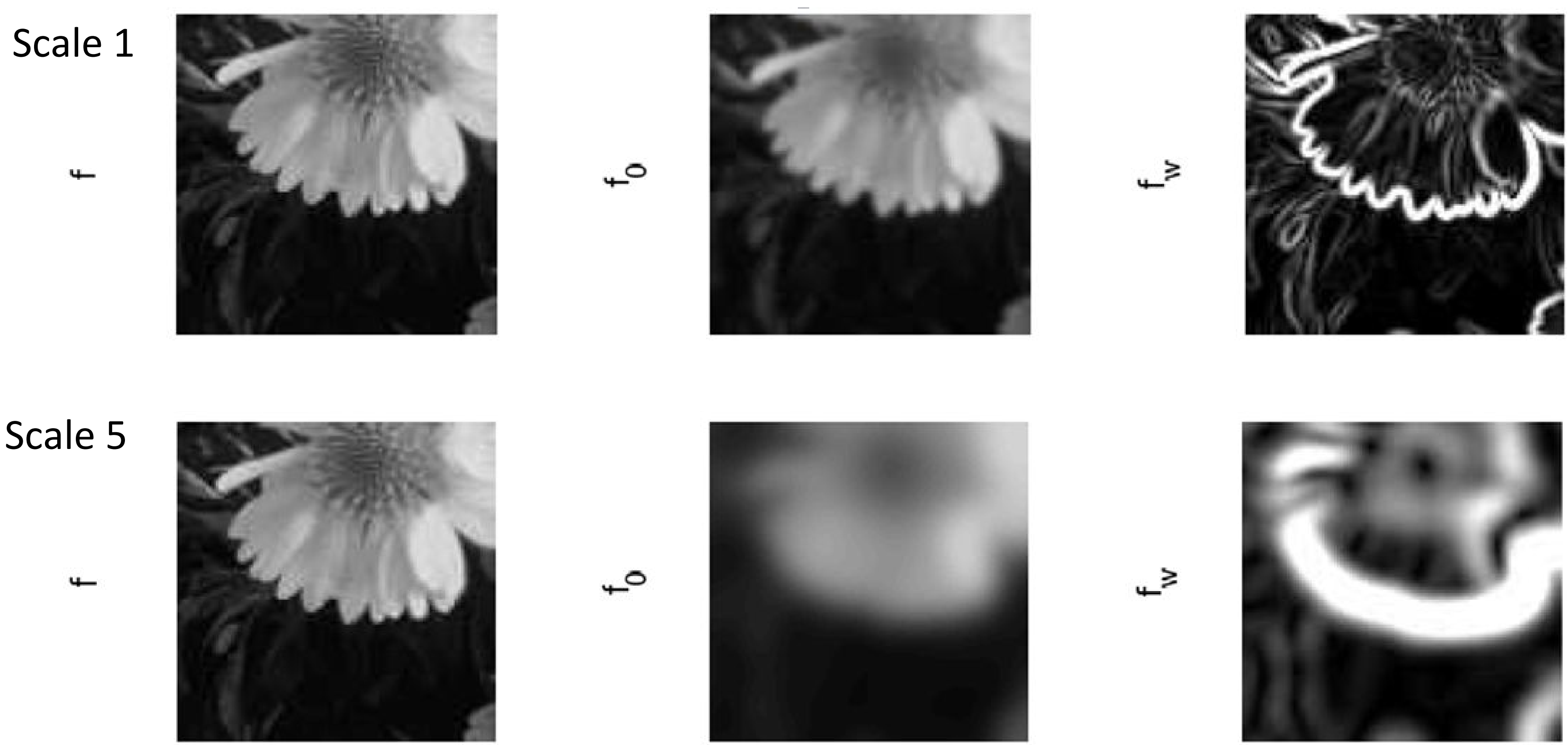
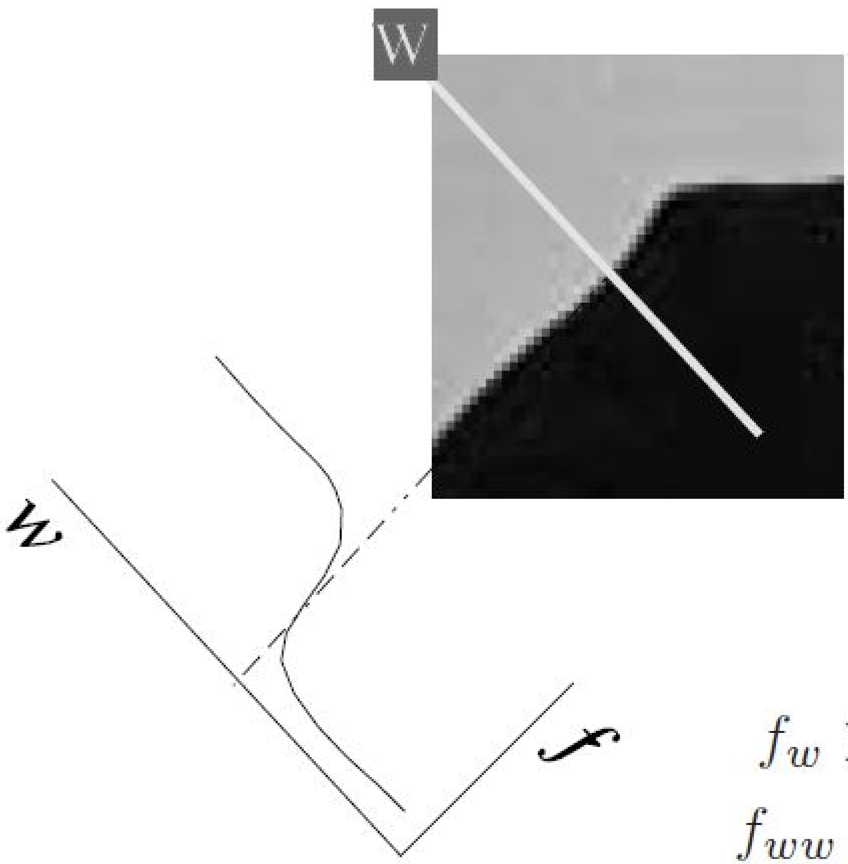


Figure 6.10: **Gradient vector.** The gradient vector ∇f at different resolutions of an image.

Gradient *components* dependent on coordinate system; *magnitude* not!



(Canny) Edge Detection



At the edges, find the zero crossing of the second derivative in the direction of the gradient.

$$f_w \gg 0 \quad : \quad \text{edge detection}$$

$$f_{ww} \approx 0 \quad : \quad \text{edge localization}$$

The Gradient Gauge (1): definition

Locally we can write the Taylor expansion as:

$$f(\mathbf{a}+\mathbf{v}) \approx f(\mathbf{a}) + \mathbf{v}^\top \nabla f(\mathbf{a}) + \tfrac{1}{2}\mathbf{v}^\top H_f(\mathbf{a})\mathbf{v}$$

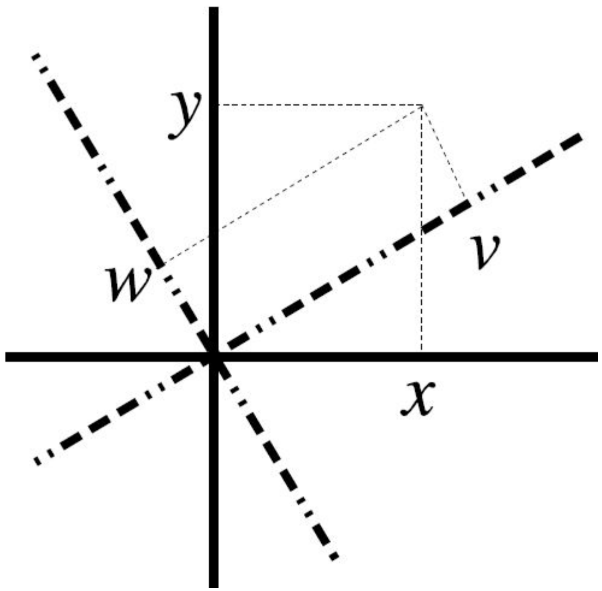
where

$$\mathbf{v} = \begin{pmatrix} v \\ w \end{pmatrix}_{(v,w)}$$

$$\nabla f(\mathbf{a}) = \begin{pmatrix} f_v(\mathbf{a}) \\ f_w(\mathbf{a}) \end{pmatrix}_{(v,w)}$$

$$H_f(\mathbf{a}) = \begin{pmatrix} f_{vv}(\mathbf{a}) & f_{vw}(\mathbf{a}) \\ f_{vw}(\mathbf{a}) & f_{ww}(\mathbf{a}) \end{pmatrix}_{(v,w)}$$

The Gradient Gauge (2): transformation



A vector $(x\ y)^\top$ in the xy-coordinate system can also be expressed in the vw-coordinate system:

$$\begin{pmatrix} v \\ w \end{pmatrix} = \underbrace{\frac{1}{\sqrt{f_x^2 + f_y^2}} \begin{pmatrix} f_y & -f_x \\ f_x & f_y \end{pmatrix}}_{\text{Rotation } R} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\mathbf{v} = R\mathbf{x}$$

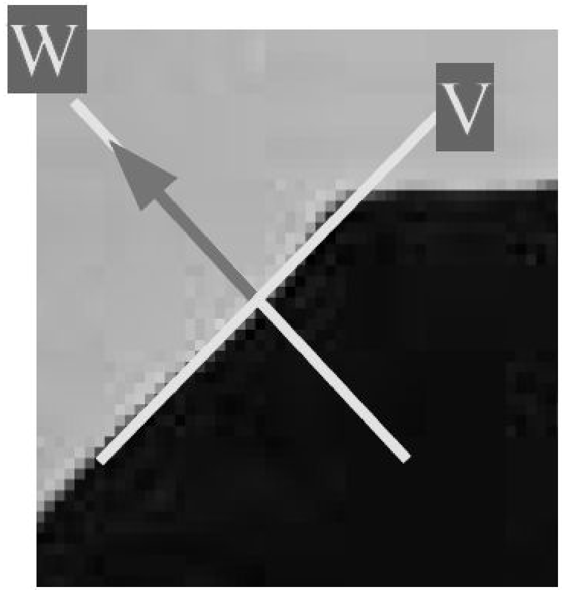
For the gradient vector $(f_x\ f_y)^\top$ we £nd:

$$\nabla f = \begin{pmatrix} f_x \\ f_y \end{pmatrix}_{(x,y)} = \begin{pmatrix} 0 \\ \sqrt{f_x^2 + f_y^2} \end{pmatrix}_{(v,w)}$$

$$\mathbf{x}^\top H_f \mathbf{x} = (R^\top \mathbf{v})^\top H_f (R^\top \mathbf{v}) = \mathbf{v}^\top R H_f R^\top \mathbf{v}$$

The Gradient Gauge (3): Hessian re-expressed

$$\mathbf{e}_w \equiv \frac{1}{\sqrt{f_x^2+f_y^2}}\begin{pmatrix} f_x \\ f_y \end{pmatrix}$$



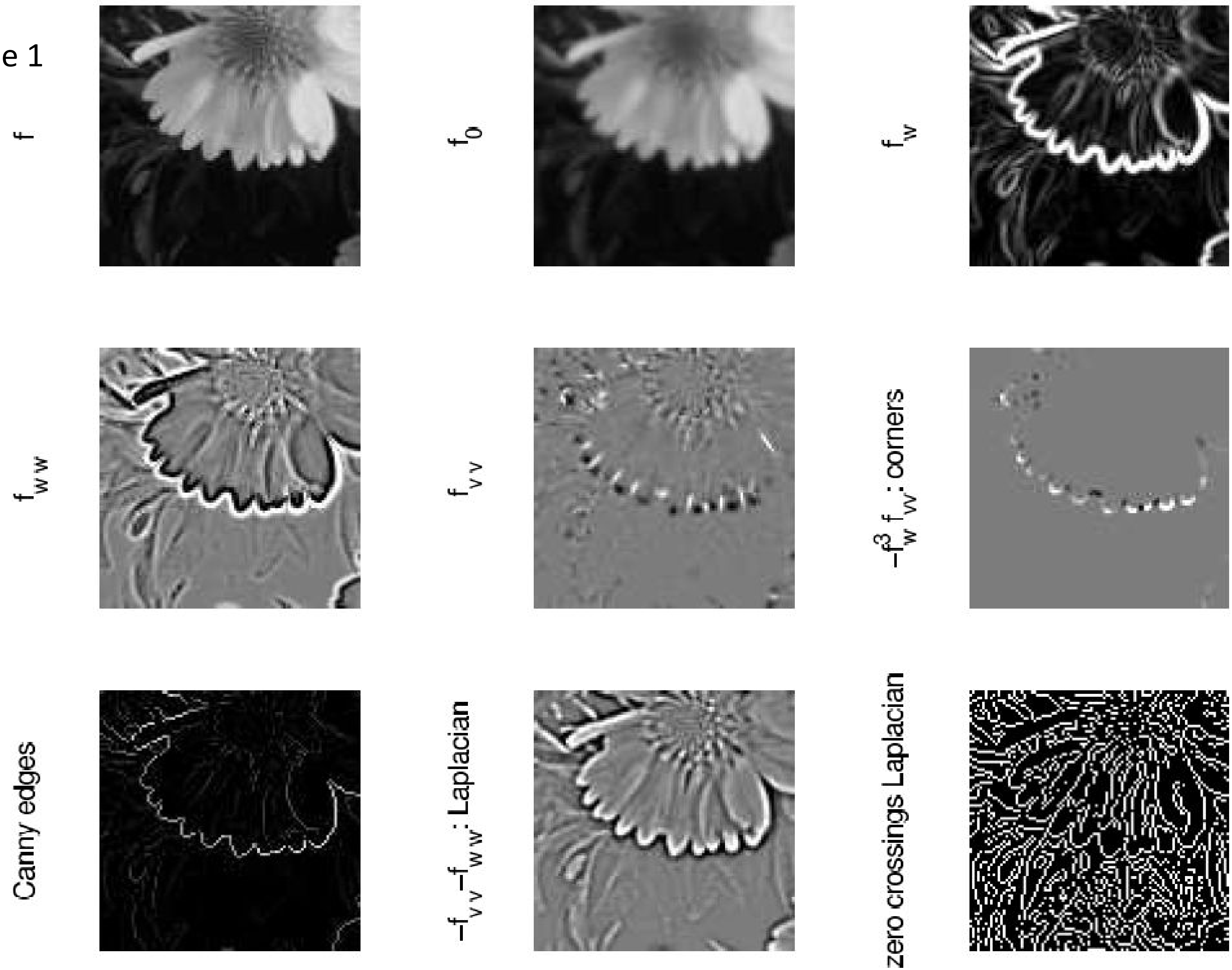
$$\mathbf{e}_v \equiv \frac{1}{\sqrt{f_x^2+f_y^2}}\begin{pmatrix} f_y \\ -f_x \end{pmatrix}$$

$$\begin{pmatrix} f_{vv} & f_{vw} \\ f_{vw} & f_{ww} \end{pmatrix} =$$

$$= \frac{1}{f_x^2+f_y^2}\begin{pmatrix} f_y & -f_x \\ f_x & f_y \end{pmatrix}\begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}\begin{pmatrix} f_y & f_x \\ -f_x & f_y \end{pmatrix}$$

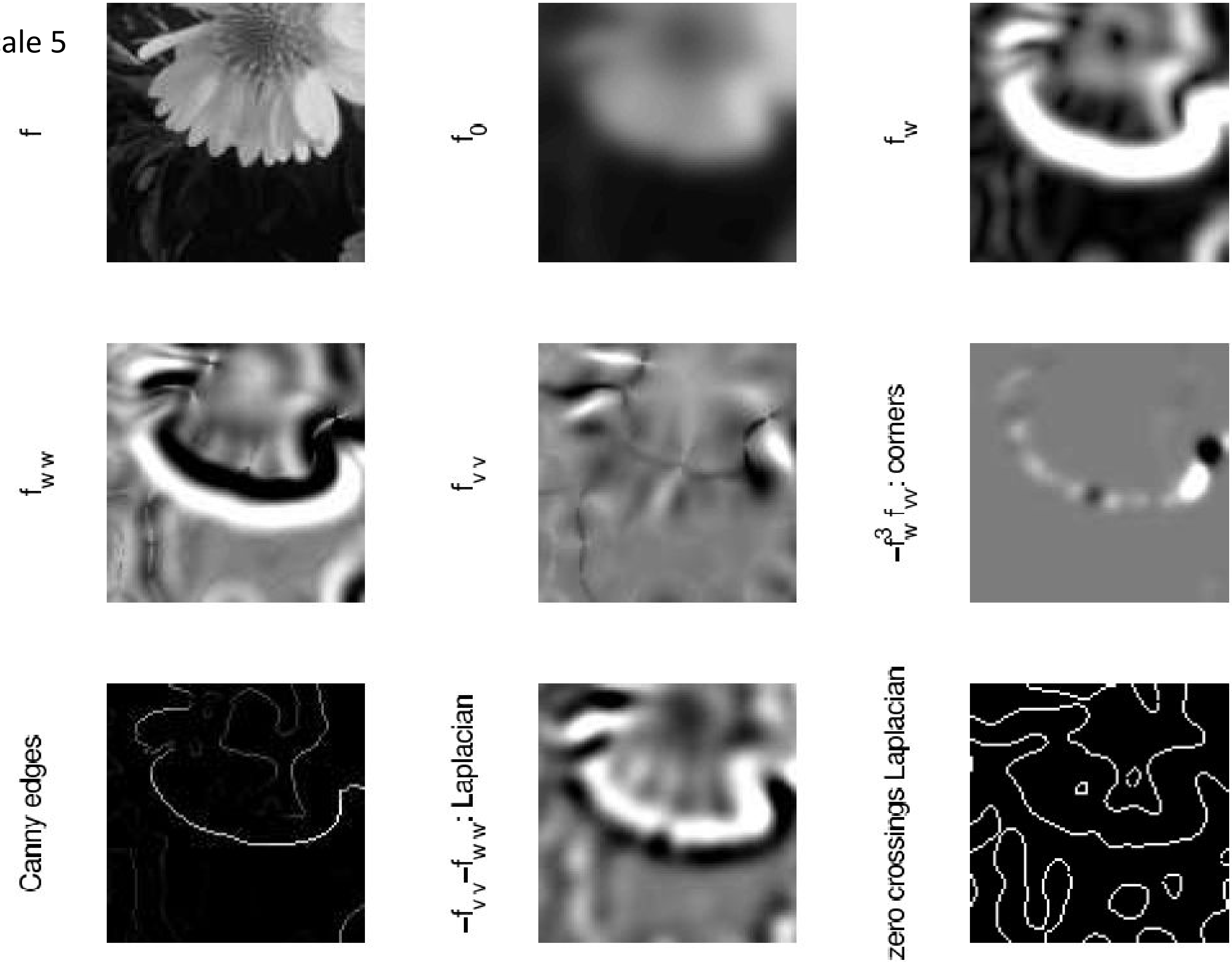
$$= \frac{1}{f_x^2+f_y^2}\begin{pmatrix} f_x^2 f_{yy} - 2 f_x f_y f_{xy} + f_y^2 f_{xx} & (f_y^2 - f_x^2) f_{xy} + f_x f_y (f_{xx} - f_{yy}) \\ (f_y^2 - f_x^2) f_{xy} + f_x f_y (f_{xx} - f_{yy}) & f_x^2 f_{xx} + 2 f_x f_y f_{xy} + f_y^2 f_{yy} \end{pmatrix}$$

Scale 1

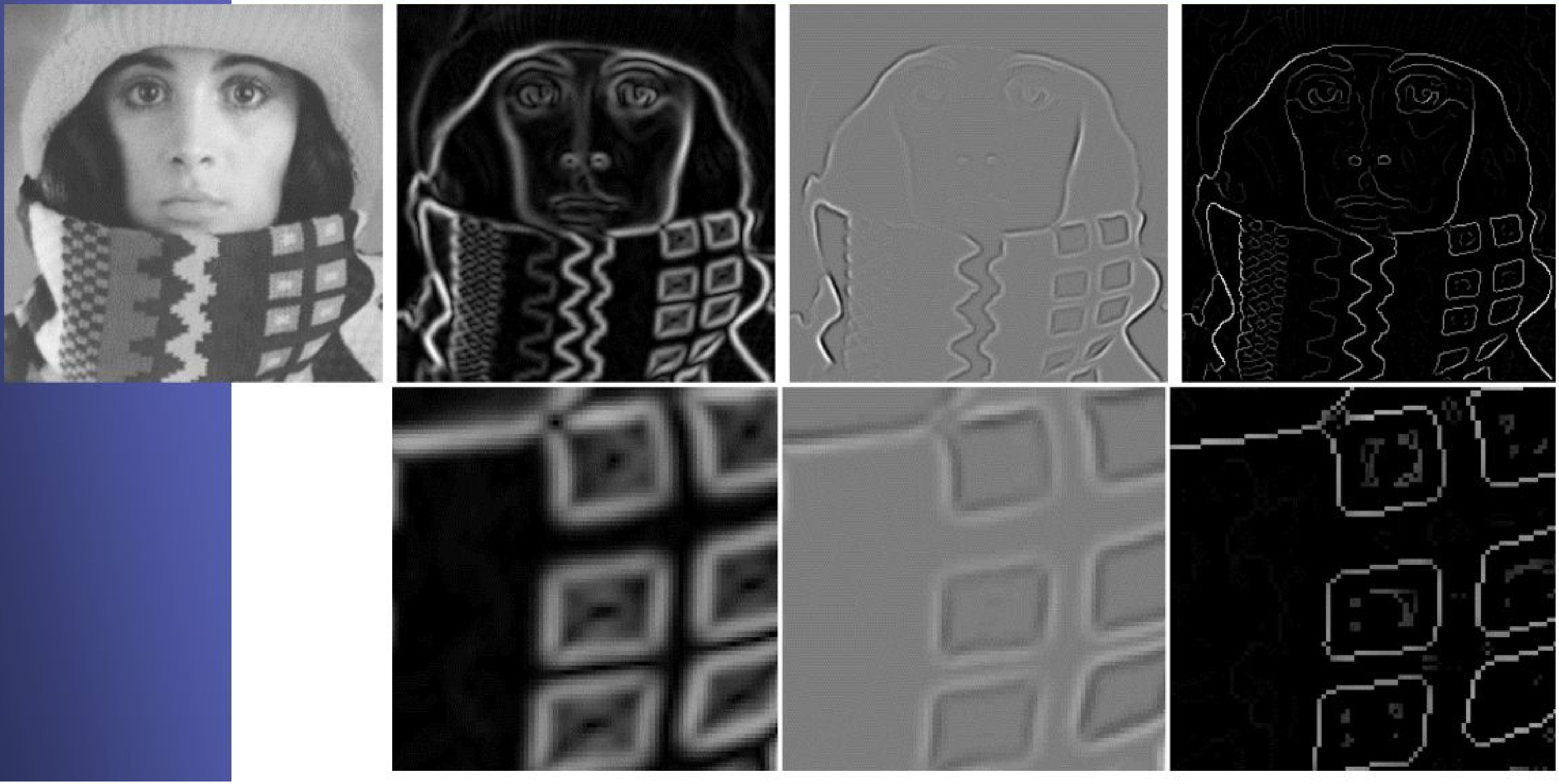


f
f0
fw
fww
fvv
−fw3 fvv: corners
Canny edges
−fvv−fww: Laplacian
zero crossings Laplacian

Scale 5


f
f_0
f_w
f_{ww}
f_{vv}
$-f_w^3 f_{vv}$: corners
Canny edges
$-f_{vv}-f_{ww}$: Laplacian
zero crossings Laplacian

Canny edge detector



Classification of Local Neighborhoods

2nd order neighborhoods with a gradient

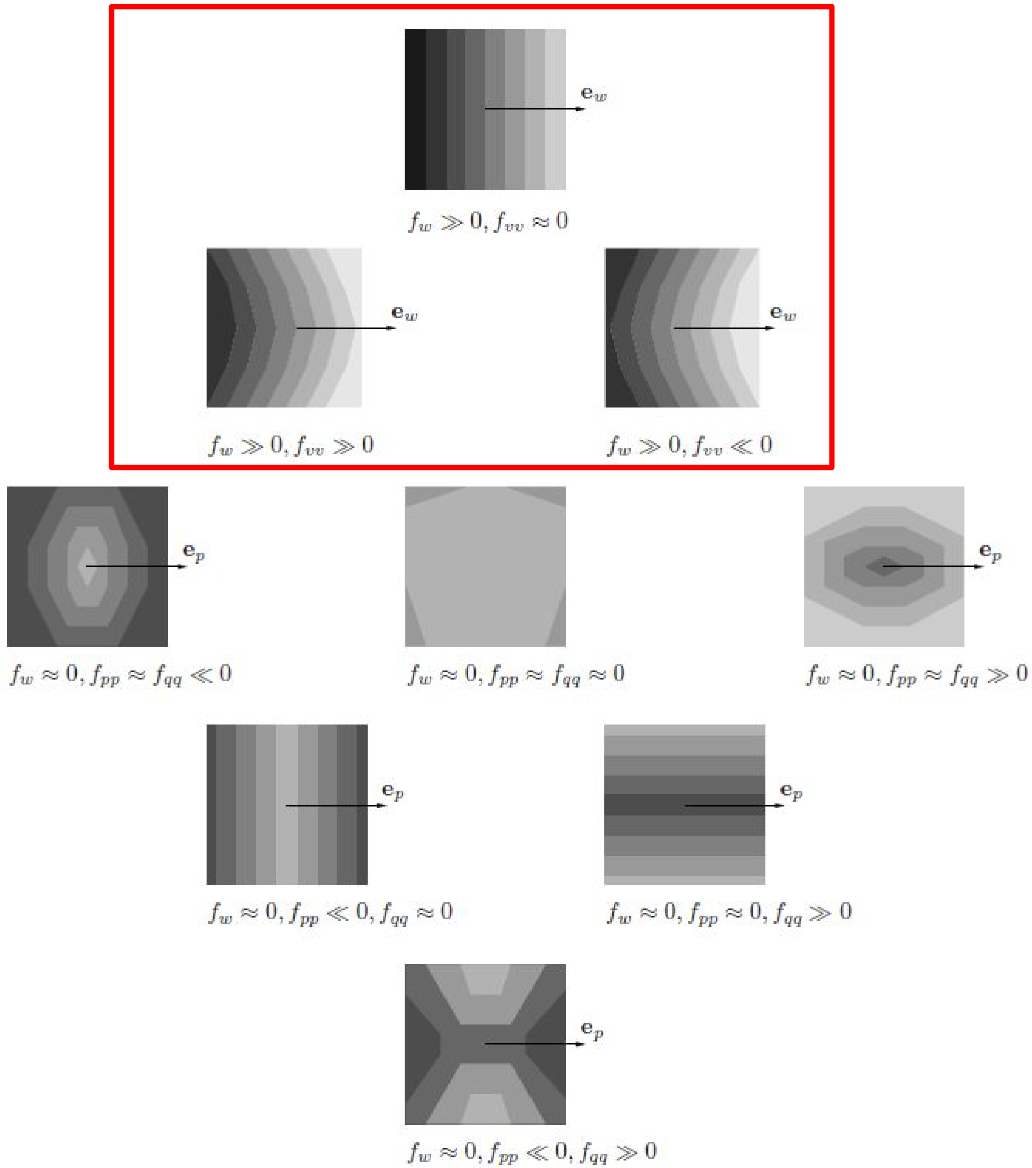


Figure 6.5: **Classification of local neighborhoods.**

Corner detection (1): Isophote Curvature

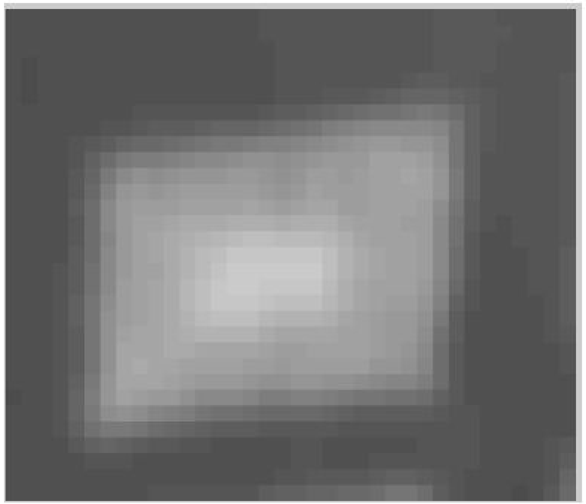
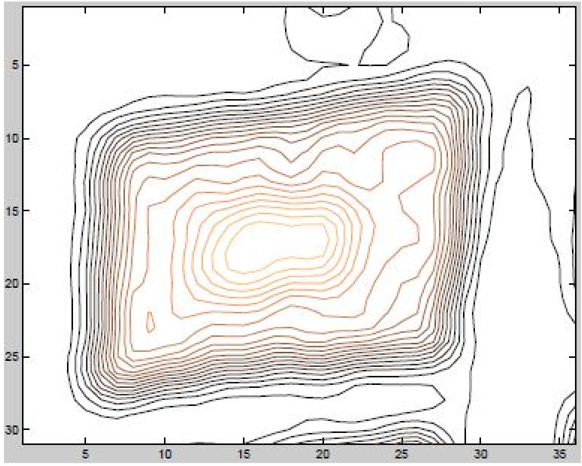
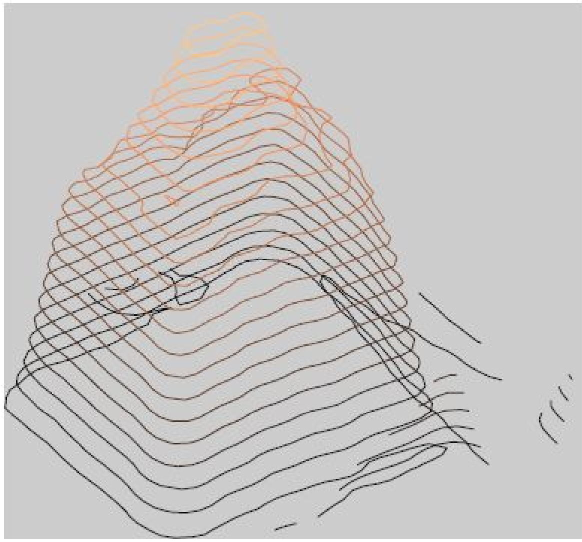


Image (detail)



Isophotes of Image



Isophotes in 3D

Corner detection (2): Isophote Curvature

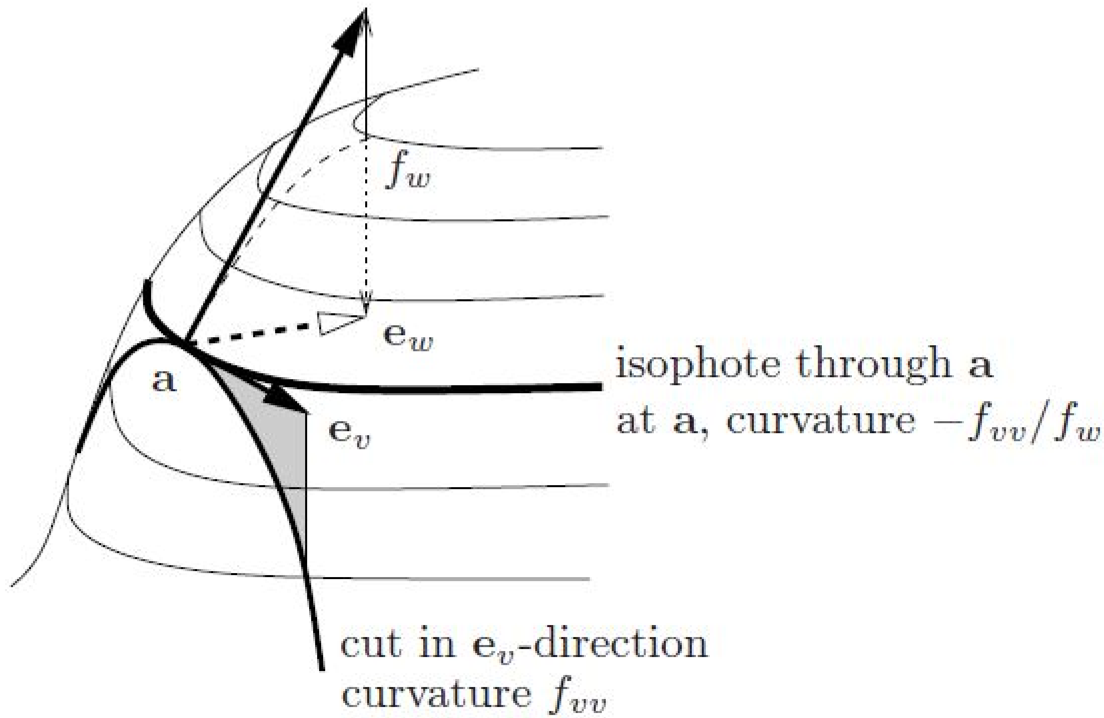


Figure 6.4: The local neighborhood and the isophote through its center point.

Corner detection (3): Isophote Curvature

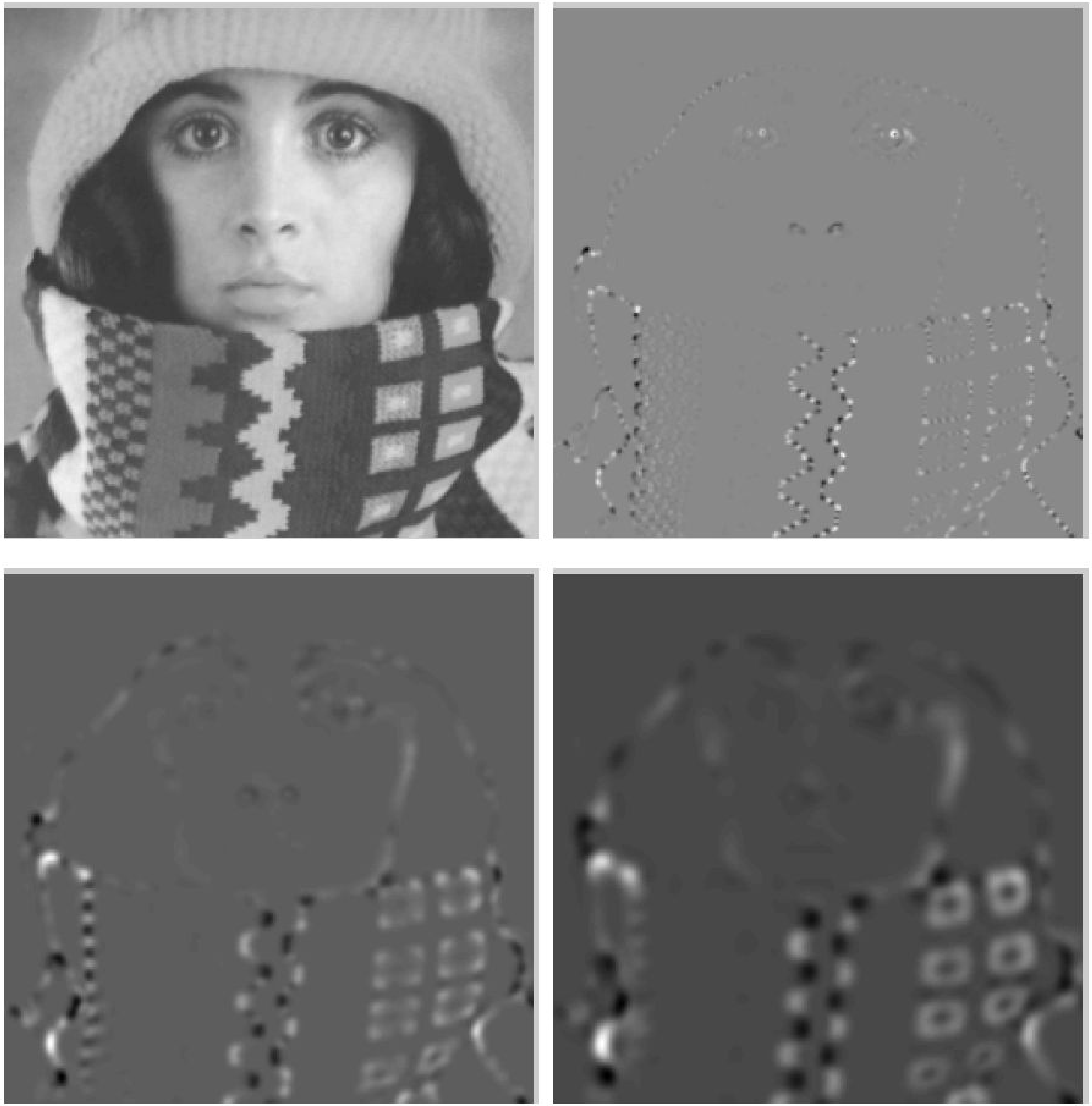
$$
\begin{aligned}
f(v, W(v)) &= \text{constant} \\
\textit{differentiate to } v & \downarrow \\
f_v + f_w W_v &= 0 \\
\textit{differentiate to } v & \downarrow \\
f_{vv} + f_{vw} W_v + f_{wv} W_v + f_{ww} W_v^2 + f_w W_{vv} &= 0 \\
\textit{use } W_v(0) = 0 & \downarrow \\
f_{vv} + f_w W_{vv} &= 0
\end{aligned}
$$

isophote curvature:

$$W_{vv} = -\frac{f_{vv}}{f_w}.$$

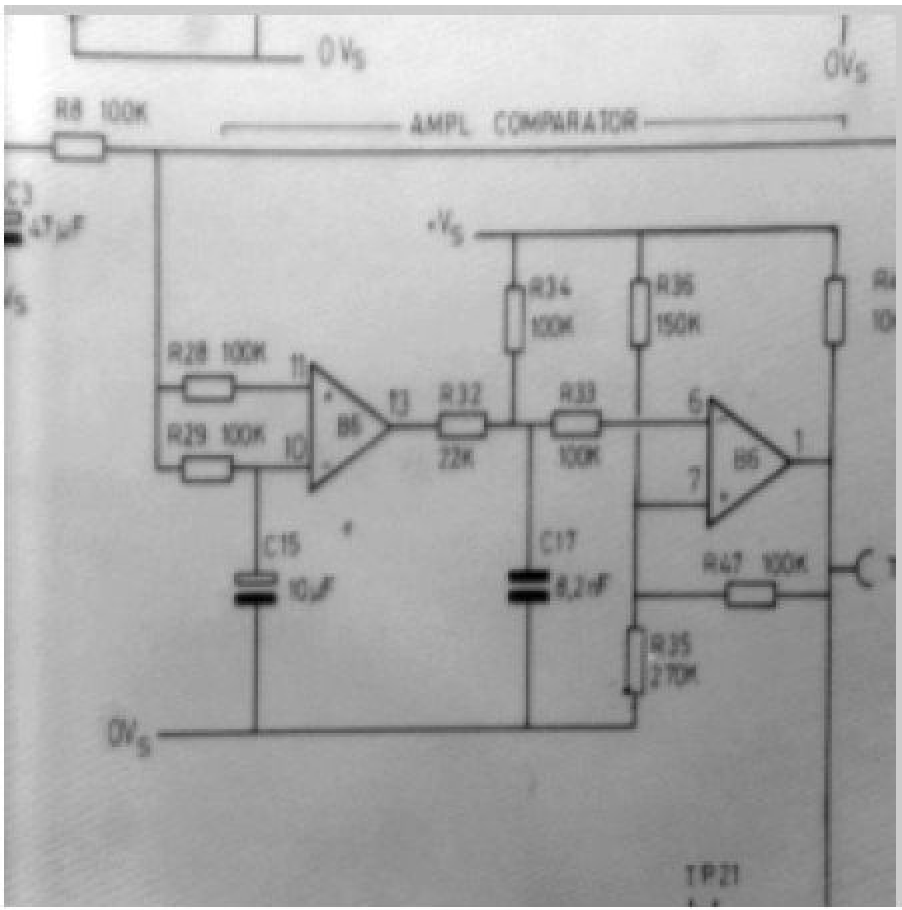
Corner Detection

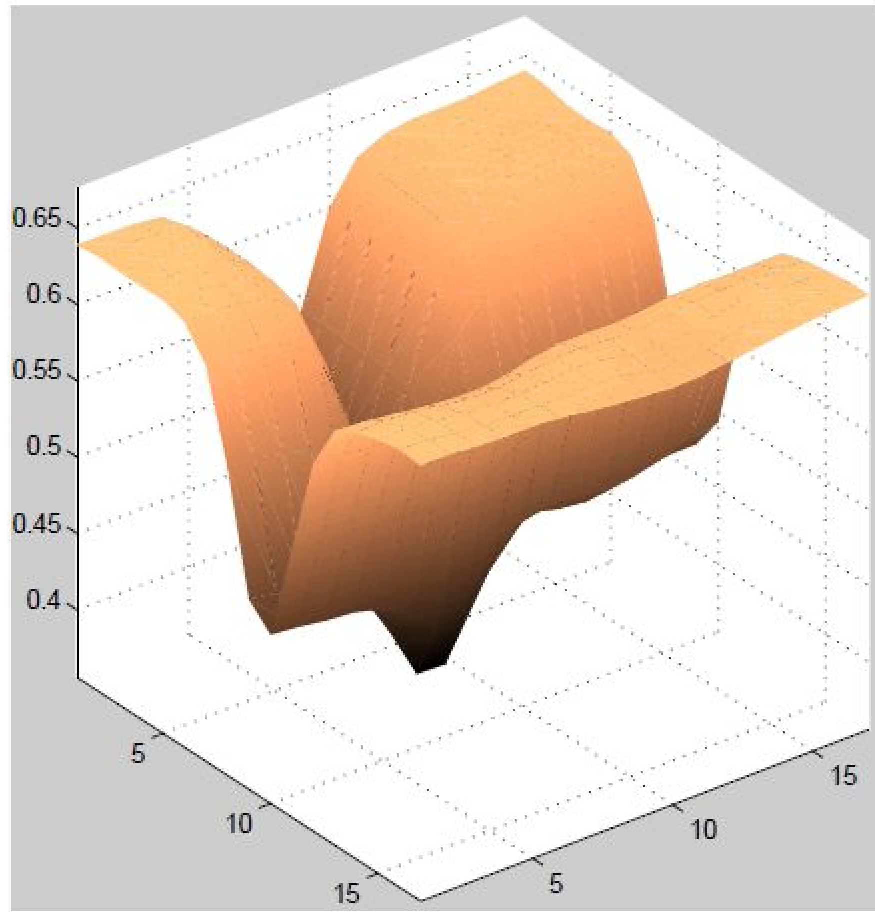
(by isophote curvature)



The Curvature Gauge (1): motivation

There are interesting points where the gradient vanishes





The Curvature Gauge (2): eigenvectors

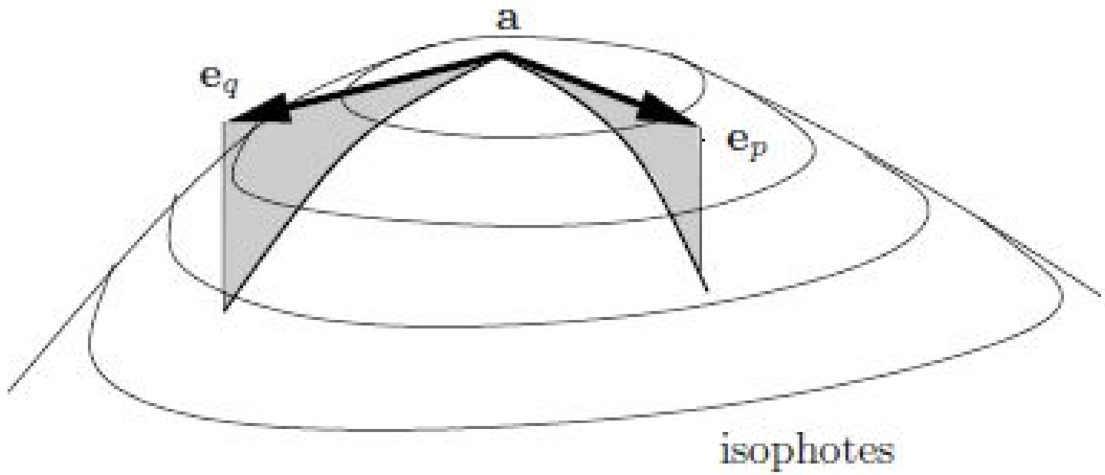
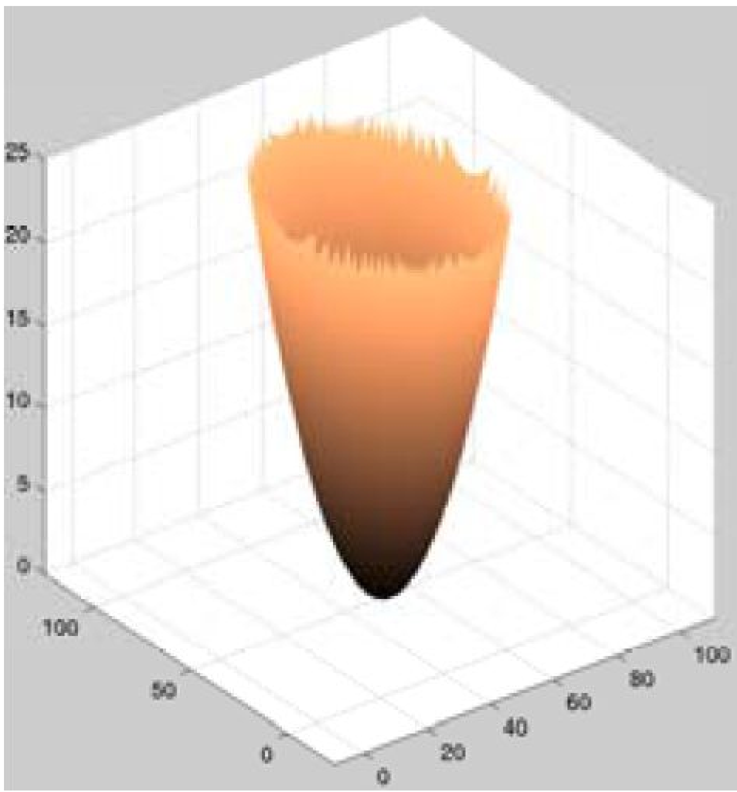


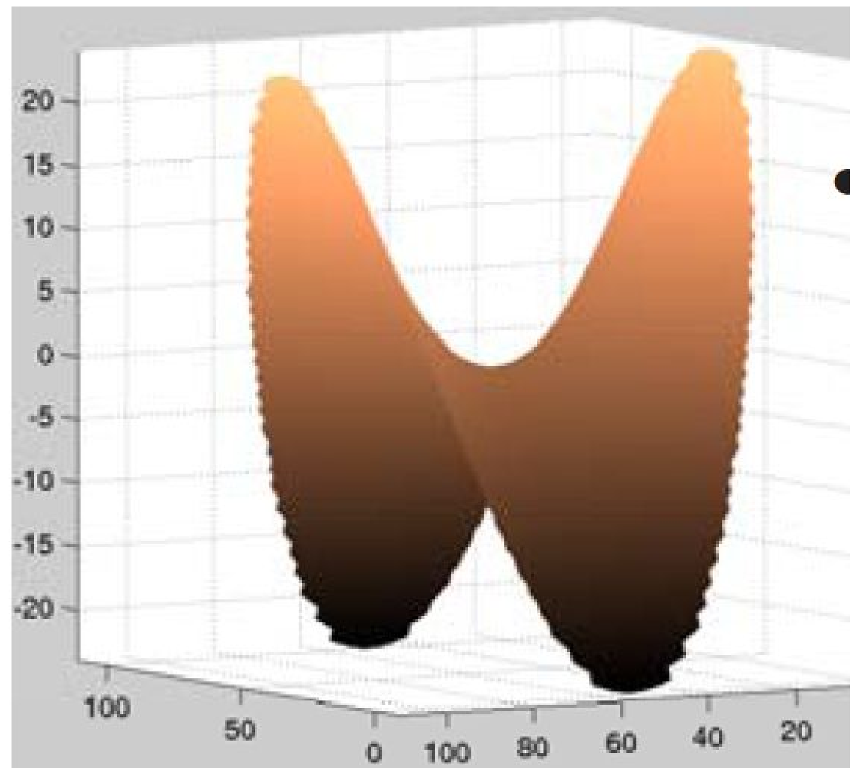
Figure 6.6: The curvature gauge is applied in the absence of a gradient.

$$\mathbf{p} = \begin{pmatrix} f_{xx} - f_{yy} - \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \\ 2f_{xy} \end{pmatrix}$$

$$\mathbf{q} = \begin{pmatrix} f_{xx} - f_{yy} + \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \\ 2f_{xy} \end{pmatrix}$$

Curvature Gauge (3): typical 'hoods





Curvature Gauge (4): eigenvalues

- The second order derivatives in the pq-coordinate frame can be expressed in terms of the derivatives in the xy-coordinate frame:

$$\begin{aligned} f_{pp} &= f_{xx} + f_{yy} - \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \\ f_{qq} &= f_{xx} + f_{yy} + \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \end{aligned}$$

- The values of f_{pp} and f_{qq} determine the type of local image patch (always assuming that $f_w \approx 0$).

  dark blob: $f_{pp} \approx f_{qq} \gg 0$

  bright blob: $f_{pp} \approx f_{qq} \ll 0$

  dark bar: $f_{pp} \approx 0, \quad f_{qq} \gg 0$

  bright bar: $f_{pp} \ll 0, \quad f_{qq} \approx 0$

  locally constant: $f_{pp} \approx f_{qq} \approx 0$

  saddle point: $f_{pp} \ll 0, \quad f_{qq} \gg 0$

Classification of Local Neighborhoods

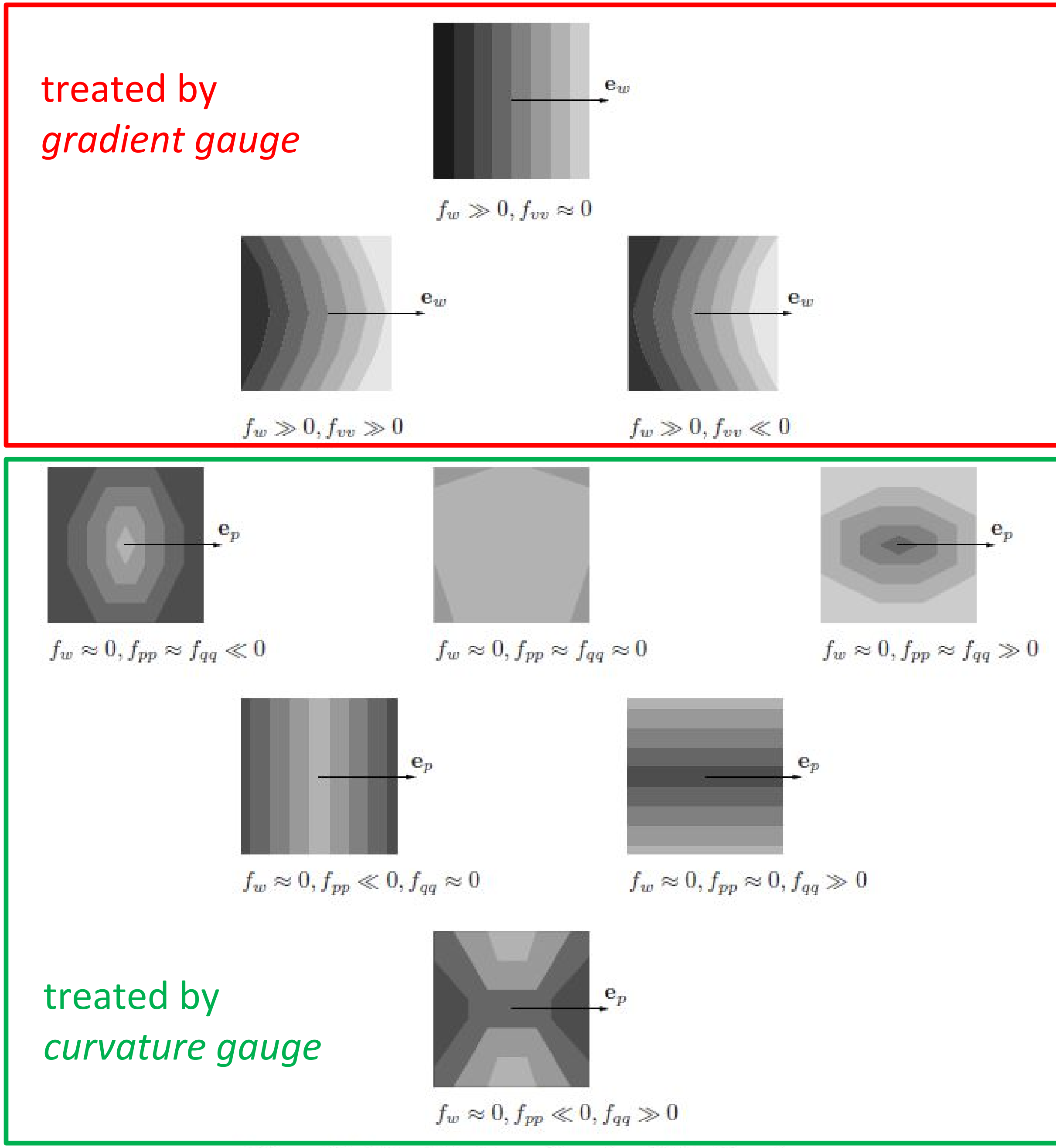


Figure 6.5: **Classification of local neighborhoods.**