19. Steerable kernels

19.1 Introduction

Clearly, orientation of structure is a multi-scale concept. On a small scale, the local orientation of structural elements, such as edges and ridges, may be different from the orientations of the same elements at a larger scale. Figure 19.1 illustrates this.

```
<< FrontEndVision`FEV`
im = Import["fabric.gif"][[1, 1]];
DisplayTogetherArray[Prepend[
   (ListDensityPlot[gauge2DN[im0, 2, 0, #] /. im0 → im]) & /@ {1.5, 5},
   ListDensityPlot[im]], ImageSize -> 350];
```

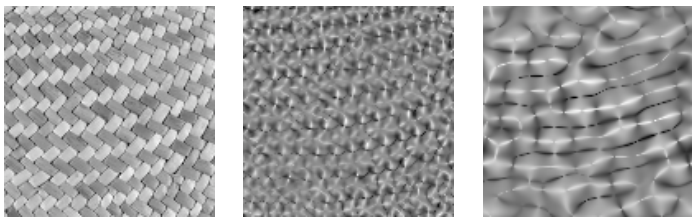

Figure 19.1 The multi-scale nature of local orientation. Left: original, basket texture. Middle: at a small scale ($\sigma = 1.5$ pixels), orientation is governed by the fibers as is shown by the 'ridgeness' L_{vv}. Right: at a larger scale ($\sigma = 5$ pixels), an other orientation can be seen by calculating the L_{vv} ridgeness (bright areas, more or less horizontal in the picture).

Orientation plays an important role as parameter in establishing similarity relations between neighboring points. As such, it is an essential ingredient of methods for *perceptual grouping*. E.g. the grouping of edge pixels in a group that defines them as belonging to the same contour, could be done using similarity in orientation of the edges, i.e. of their respective gradient vectors. In chapter 12 we have seen that the visual system is particularly well equipped to take measurements of differential properties at a continuum of orientations:

the typical spokewheel structure of orientation columns that make up each hypercolumn (the wetware for the analysis of a single binocular visual field 'pixel') in the visual primary cortex, where the receptive fields were found at all orientations. In this chapter, we will study in detail the orientation properties of Gaussian derivative filters, and a concept named 'steerability'. We come to a proper definition of a directional derivative for all orders of differentiation, and how this can best be computed in a Cartesian framework. We then study as an application of orientation analysis the detection of stellate tumors in mammography (example taken from [Karssemeier1995a, Karssemeier1996a]), which is based on a global analysis of orientation. This is an example of computer-aided diagnosis (CAD).

Other examples of context dependent perception of curvature are some of the well known obscured parallel lines illusions, shown in figure 19.2.

```
square[x_, y_, s_] :=
  Line[{{x, y}, {x + s, y}, {x + s, y + s}, {x, y + s}, {x, y}}];
DisplayTogetherArray[
  {Show[Graphics[{Thickness[.01], Line[{{#, 0}, {0, #}}], If[EvenQ[#],
         Table[Line[{{# - n, n - 1}, {# - n, n + 1}}], {n, .1, 10, .5}],
         Table[Line[{{# - n - 1, n}, {# - n + 1, n}}], {n, .1, 10, .5}]]} & /@
      Range[1, 20, 3]], PlotRange -> {{0, 10}, {0, 10}}],
Show[Graphics[{Table[Line[{{-Cos[ϕ], -Sin[ϕ]}, {Cos[ϕ], Sin[ϕ]}}],
       {ϕ, 0, π, π / 16}], square[-.25, .25, .5], square[-.25, -.75, .5],
      Circle[{-.5, 0}, .25], Circle[{.5, 0}, .25]}],
    PlotRange -> {{-1, 1}, {-1, 1}}, AspectRatio -> Automatic]},
  ImageSize -> 400];
```

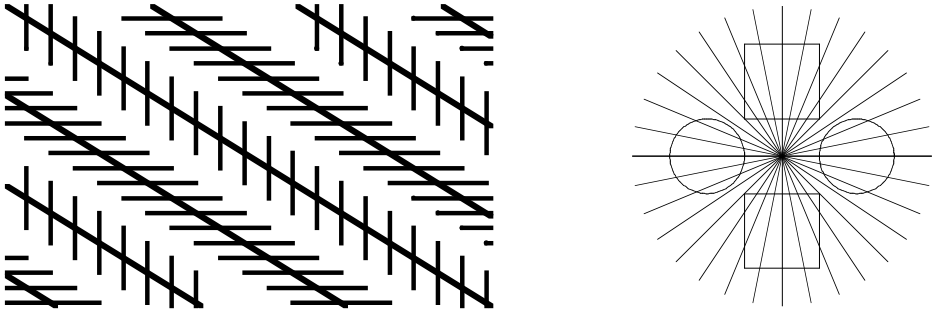

Figure 19.2 Obscured parallel lines illusions. Left: We perceive the lines as not parallel, in reality they are parallel. Right: the squares seem parallelograms, the circles seem to have an egg shape. The context around the contours of the figures determines the perceived orientation.

19.2 Multi-scale orientation

We define the *orientation* of a vector relative to a coordinate frame as the set of angles between the vector and with the frame vectors of the coordinate frame. Because we deal with orthogonal coordinates in this book, we need a single angle for a 2D vector, and two angles for a 3D vector.

Formal definition of angulation (2D), tilt and spin (3D). Figures.

The *direction* of a vector is the absolute value of the orientation. E.g. the direction of the x-axis is horizontal, and a identical direction is created when we rotate the x-axis over π radians (180 degrees).

In chapter 6,section 5, we studied the orientation of first order differential structure: the gradient vector field, and in chapter 6, section 7.3 we encountered the orientations of the principal curvatures, as the orientations of the Eigenvectors of the Hessian second order matrix.

19.3 Orientation analysis with Gaussian derivatives

In order to build the machinery for the extraction of orientation information, we need to fully understand the behavior of the Gaussian derivative operator at different orientations, and how to control this behavior. We will limit the analysis to 2D. The zeroth order Gaussian is isotropic by definition, and has no orientation. The action of this operator on an image is rotational invariant. All non-zero partial derivatives are equipped with a direction.

The first order Gaussian derivative kernel $\frac{\partial G}{\partial x}$ is shown in figure 19.3, and we define the orientation of this kernel to be zero, i.e. the angle ϕ of its axis along which the differentiation is done is zero radians with the x-axis. The orientation ϕ of the Gaussian derivative kernel to y, $\frac{\partial G}{\partial y}$, is $\pi/2$, pointing upwards along the positive y-axis. So increments in ϕ are positive in a counterclockwise fashion.

The first order Gaussian derivative kernel in another orientation can readily be made from its basic substituents: it is well known that a kernel with orientation ϕ can be constructed from $\mathrm{Cos}(\phi)\ \frac{\partial G}{\partial x} + \mathrm{Sin}(\phi)\ \frac{\partial G}{\partial y}$.

Figure 19.3 illustrates this on the convolution of the kernel on a Dirac-delta function, i.e. a single spike, which gives us the figure of the kernel, as we have seen in chapter 2:

```
im = Table[0, {128}, {128}]; im[[64, 64]] = 100;
ϕ = π / 6; Block[{$DisplayFunction = Identity},
 imx = gD[im, 1, 0, 15]; imy = gD[im, 0, 1, 15];
 imϕ = Cos[ϕ] gD[im, 1, 0, 15] + Sin[ϕ] gD[im, 0, 1, 15];
 p1 = ListDensityPlot[#] & /@ {imx, imy, imϕ}];
Show[GraphicsArray[p1]];
```

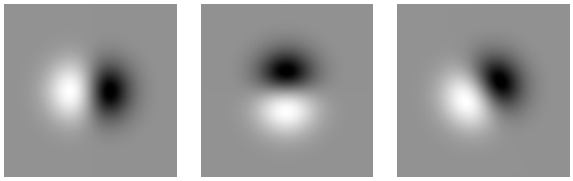

Figure 19.3 A first order Gaussian derivative at any orientation can be constructed with the partial derivatives $\frac{\partial G}{\partial x}$ (left) and $\frac{\partial G}{\partial y}$ (middle) as a basis. Right: $\mathrm{Cos}(\phi)\ \frac{\partial G}{\partial x} + \mathrm{Sin}(\phi)\ \frac{\partial G}{\partial y}$, for $\phi = \pi/6$.

A class of filters where a filter of any orientation can be constructed from a linear combination of other functions is called a *steerable filter* [Freeman1991a]. The rotational components form a *basis*. A basis will contain more elements when we go to higher order. The question is now: what are the basis functions? How many basis functions do we need for the construction of a rotated Gaussian derivative of a particular order? We will discuss two important classes of basis functions (see figure 19.6):

a. basis functions that are rotated copies of the Gaussian derivative itself;
b. basis functions taken from the set of all partial derivatives in the Cartesian framework;

In particular, we will derive later in the text general formulas for e.g. the rotation of $\frac{\partial^3 G(x,y)}{\partial x^3}$ over 30 degrees clockwise ($-\pi/6$ radians). In the two bases the results are the same ($\frac{\partial^3 G(x,y)}{\partial x^3}\Big|_{45^o}$ denotes the 45^o rotated version of the third derivative kernel):

$$\frac{\partial^3 G(x,y)}{\partial x^3}\Big|_{-30^o} = \frac{\sqrt{3}}{4}\,\frac{\partial^3 G(x,y)}{\partial x^3}\Big|_{0^o} + \frac{-3+\sqrt{3}}{4\sqrt{2}}\,\frac{\partial^3 G(x,y)}{\partial x^3}\Big|_{45^o} + \frac{1}{4}\,\frac{\partial^3 G(x,y)}{\partial x^3}\Big|_{90^o} - \frac{3+\sqrt{3}}{4\sqrt{2}}\,\frac{\partial^3 G(x,y)}{\partial x^3}\Big|_{135^o}$$

$$\frac{\partial^3 G(x,y)}{\partial x^3}\Big|_{-30^o} = -\frac{1}{8}\,\frac{\partial^3 G(x,y)}{\partial y^3} + \frac{3\sqrt{3}}{8}\,\frac{\partial^3 G(x,y)}{\partial x\,\partial y^2} + \frac{9}{8}\,\frac{\partial^3 G(x,y)}{\partial x^2\,\partial y} - \frac{3\sqrt{3}}{8}\,\frac{\partial^3 G(x,y)}{\partial x^3}$$

The first class will be derived in of basis functions may have interesting similarities in the control of directional derivatives in the visual system because of its self-similarity, symmetry and a reduced set of receptive field sensitivity profiles, the second is more apt for computer implementation. The derivation for the first class will be given in section 19.4, the second class will be derived in section 19.5. The two basis sets are different, and an illustrating example of multi-scale steerable filters.

19.4 Steering with self-similar functions

We need to know what is the necessary condition for a function to be steerable. We look at the first order Gaussian derivative as a first example. We express the kernel in polar coordinates $\{x, y\} = \{r\,\mathrm{Cos}(\phi),\ r\,\mathrm{Sin}(\phi)\}$ where r is the distance to the origin, and ϕ the angle with the real x-axis. With the function **TrigToExp** we express the result in complex exponentials $e^{i\phi}$:

```
Clear[ϕ]; {TrigToExp[Cos[ϕ] + I Sin[ϕ]], Exp[I ϕ] // ExpToTrig}
```

```
{ⅇ^(ⅈ ϕ), Cos[ϕ] + ⅈ Sin[ϕ]}
```

We transform our Gaussian kernel and some of its partial derivatives, and display the result, after division by the Gaussian kernel, in **MatrixForm** for notational clarity:

```
g := 1/(2 π σ^2) E^(-(x^2+y^2)/(2 σ^2)); ϕ =.;
t = {{g, ∂x g}, {∂y g, ∂x,y g}} /. {x → r Cos[ϕ], y → r Sin[ϕ]} // TrigToExp // Simplify;
t // MatrixForm
```

$$\begin{pmatrix} \frac{e^{-\frac{r^2}{2\sigma^2}}}{2\pi\sigma^2} & -\frac{e^{-\frac{r^2}{2\sigma^2}-i\phi}\,(1+e^{2i\phi})\,r}{4\pi\sigma^2\,\sigma^2} \\ \frac{i\,e^{-\frac{r^2}{2\sigma^2}-i\phi}\,(-1+e^{2i\phi})\,r}{4\pi\sigma^2\,\sigma^2} & -\frac{i\,e^{-\frac{r^2}{2\sigma^2}-2i\phi}\,(-1+e^{4i\phi})\,r^2}{8\pi\sigma^2\,\sigma^4} \end{pmatrix}$$

To see the structure more clearly, we divide the Gaussian itself out and collect the complex exponentials:

```
Collect[t/t[[1, 1]], E^(I ϕ)] // MatrixForm
```

$$\begin{pmatrix} 1 & -\frac{e^{-i\phi} r}{2\sigma^2} - \frac{e^{i\phi} r}{2\sigma^2} \\ -\frac{i e^{-i\phi} r}{2\sigma^2} + \frac{i e^{i\phi} r}{2\sigma^2} & \frac{i e^{-2i\phi} r^2}{4\sigma^4} - \frac{i e^{2i\phi} r^2}{4\sigma^4} \end{pmatrix}$$

The resulting function is separable in a radial part $a(r)$ and an angular part $e^{i n \phi}$ where n is equal to the differential order: $f(r, \phi) = \sum_{j=1}^{M} a_n(r)\, e^{i n \phi}$. A function with these properties *steers* when it can be written as rotated versions of itself.

The so-called *steerability constraint* is $f^{\theta}(x, y) = \sum_{j=1}^{M} k_j(\theta)\, f^{\theta_j}(x, y)$, where M is the number of filters, $f^{\theta}(x, y)$ is the rotated kernel, and the $f^{\theta_j}(x, y)$ are the rotated basis kernels. The weighting factors $k_j(\theta)$ are to be determined, as follows. When we fill in the polar representation of the kernel in the steerability constraint, we get

$$a_n(r)\, e^{i n \phi} = \sum_{j=1}^{M} k_j(\theta)\, a_n(r)\, e^{i n \phi}. \qquad (1)$$

We can divide by $a_n(r)$, and when $a_n(r) = 0$ for some n we just remove this constraint from the set of equations. The equations are equal for $-n$ and n, so we have to consider only angular frequencies in the range $0 < n < M$. Because n is equal to the order of differentiation, we have here as first result that the number of necessary basis filters for steerability in $n + 1$.

To steer the first order Gaussian derivative, we needed two basis functions, for steering a second order derivative kernel we need three basis functions. Equation (1) is a set of equations from which we can solve the weighting constants $k_j(\theta)$. Written more explicitly, they look like

$$\begin{pmatrix} 1 \\ e^{i\theta} \\ . \\ e^{i n \theta} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ e^{i\theta_1} & e^{i\theta_2} & \dots & e^{i\theta_M} \\ . & . & & . \\ e^{i n \theta_1} & e^{i n \theta_2} & \dots & e^{i n \theta_M} \end{pmatrix} \begin{pmatrix} k_1(\theta) \\ k_2(\theta) \\ . \\ k_M(\theta) \end{pmatrix}.$$

The 1's in the top row are for $e^{i 0 \theta}$. Let us solve this set of equations.

We need to choose the orientations θ_j of the basis functions such that the columns in the matrix above are linearly independent. For symmetry reasons we choose them equally spaced over the space of directions, i.e. over the range $0 - \pi$. So, for the first order Gaussian derivative the basis orientations are 0 and $\pi/2$, for the second order they are 0, $\pi/3$ and $2\pi/3$, etc. Starting from an arbitrary angle ϕ we get $\{\phi, \phi + \pi/2\}$ and $\{\phi, \phi + \pi/3, \phi + 2\pi/3\}$ etc.

```
n = 2; θn = Table[ϕ + i π/(n + 1), {i, 0, n}]
```

$$\left\{\phi, \frac{\pi}{3} + \phi, \frac{2\pi}{3} + \phi\right\}$$

```
Exp[I 2 θn]
```

$$\left\{e^{2\,i\,\phi},\ e^{2\,i\,\left(\frac{\pi}{3}+\phi\right)},\ e^{2\,i\,\left(\frac{2\,\pi}{3}+\phi\right)}\right\}$$

Because the complex exponentials are complex, we have to solve them separately for both the real and the imaginary part.

```
n = 2; kj = Array[k, n + 1];
{ComplexExpand[Re[Exp[I n θ]] == Re[kj.Exp[I n θn]]],
 ComplexExpand[Im[Exp[I n θ]] == Im[kj.Exp[I n θn]]]}
```

$$\left\{\text{Cos}[2\,\theta] == \text{Cos}[2\,\phi]\,k[1] + \text{Cos}\left[2\left(\frac{\pi}{3}+\phi\right)\right]k[2] + \text{Cos}\left[2\left(\frac{2\,\pi}{3}+\phi\right)\right]k[3],\right.$$
$$\left.\text{Sin}[2\,\theta] == k[1]\,\text{Sin}[2\,\phi] + k[2]\,\text{Sin}\left[2\left(\frac{\pi}{3}+\phi\right)\right] + k[3]\,\text{Sin}\left[2\left(\frac{2\,\pi}{3}+\phi\right)\right]\right\}$$

For the even orders of differentiation we need to solve the equations specified by the top row and the even rows (if n is even, we start in row 0), for the odd orders we need the odd rows (if n is odd we start in row 1). The equation set is easily solved with *Mathematica:*

```
angularweights[n_] := Module[{k, i, ni}, Clear[θ, ϕ, k];
  θn = Table[ϕ + i π/(n + 1), {i, 0, n}]; kj = Array[k, n + 1];
sol = Solve[Flatten[
     Table[{ComplexExpand[Re[Exp[I ni θ]] == Re[kj.Exp[I ni θn]]],
       ComplexExpand[Im[Exp[I ni θ]] == Im[kj.Exp[I ni θn]]]},
      {ni, If[EvenQ[n], 0, 1], n, 2}]
], kj]; Flatten[kj /. sol] // Simplify // TrigReduce]
```

For the first order we find for the angular weights $k_j\,(\theta)$:

```
angularweights[1]
```

```
{Cos[θ - ϕ], Sin[θ - ϕ]}
```

Indeed, a correct result, found from these equations:

```
n = 1; Clear[θ, ϕ, k]; θn = Table[ϕ + i π/(n + 1), {i, 0, n}]; kj = Array[k, n + 1];
Table[{
  ComplexExpand[Re[Exp[I ni θ]] == Re[kj.Exp[I ni θn]]],
  ComplexExpand[Im[Exp[I ni θ]] == Im[kj.Exp[I ni θn]]]},
 {ni, If[EvenQ[n], 0, 1], n, 2}]
```

```
{{Cos[θ] == Cos[ϕ] k[1] - k[2] Sin[ϕ], Sin[θ] == Cos[ϕ] k[2] + k[1] Sin[ϕ]}}
```

For the second and third order Gaussian derivatives things get more complicated:

```
angularweights[2]
```

$$\left\{\frac{1}{3}\,(1+2\,\text{Cos}[2\,\theta-2\,\phi]),\ \frac{1}{3}\,(1-\text{Cos}[2\,\theta-2\,\phi]+\sqrt{3}\,\text{Sin}[2\,\theta-2\,\phi]),\right.$$
$$\frac{1}{9}\,((-1)^{1/6}\sqrt{3}-(-1)^{5/6}\sqrt{3}-(-1)^{1/6}\sqrt{3}\,\text{Cos}[2\,\theta-2\,\phi]+$$
$$(-1)^{5/6}\sqrt{3}\,\text{Cos}[2\,\theta-2\,\phi]-2\sqrt{3}\,\text{Sin}[2\,\theta-2\,\phi]-$$
$$\left.(-1)^{1/3}\sqrt{3}\,\text{Sin}[2\,\theta-2\,\phi]+(-1)^{2/3}\sqrt{3}\,\text{Sin}[2\,\theta-2\,\phi])\right\}$$

```
angularweights[3]
```

$$\left\{\frac{1}{2}\,(\text{Cos}[3\,\theta-3\,\phi]+\text{Cos}[\theta-\phi]),\right.$$
$$\frac{1}{4}\,(-\sqrt{2}\,\text{Cos}[3\,\theta-3\,\phi]+\sqrt{2}\,\text{Cos}[\theta-\phi]+\sqrt{2}\,\text{Sin}[3\,\theta-3\,\phi]+\sqrt{2}\,\text{Sin}[\theta-\phi]),$$
$$\frac{1}{2}\,(-\text{Sin}[3\,\theta-3\,\phi]+\text{Sin}[\theta-\phi]),$$
$$\left.\frac{1}{4}\,(\sqrt{2}\,\text{Cos}[3\,\theta-3\,\phi]-\sqrt{2}\,\text{Cos}[\theta-\phi]+\sqrt{2}\,\text{Sin}[3\,\theta-3\,\phi]+\sqrt{2}\,\text{Sin}[\theta-\phi])\right\}$$

We have found the general result for steerability of Gaussian derivative kernels.
A Gaussian derivative kernel can be steered, i.e. made in any orientation, by a linearly weighted sum of rotated versions of itself, the basis functions. There are $n+1$ functions required, equally spaced over an angle range of $0-\pi$.

For $n=1$ the basis includes $\theta = 0\,°$ and $90\,°$;
For $n=2$ the basis includes $\theta = 0\,°$, $60\,°$ and $120\,°$;
For $n=3$ the basis includes $\theta = 0\,°$, $45\,°$, $90\,°$ and $135\,°$;
For $n=3$ the basis includes $\theta = 0\,°$, $36\,°$, $72\,°$, $108\,°$ and $144\,°$; etc.
In other words: for each value of n we have a different set of basis functions.

```
ang2 = angularweights[2]; ang3 = angularweights[3];
ϕ = 0; Block[{$DisplayFunction = Identity},
 p1 = PolarPlot[#, {θ, 0, 2 π},
      PlotRange → {{-1, 1}, {-1, 1}}, Frame → True] & /@ ang2;
 p2 = PolarPlot[#, {θ, 0, 2 π}, PlotRange → {{-1, 1}, {-1, 1}},
      Frame → True] & /@ ang3];
Show[GraphicsArray[{p1, p2}], ImageSize → 500];
```

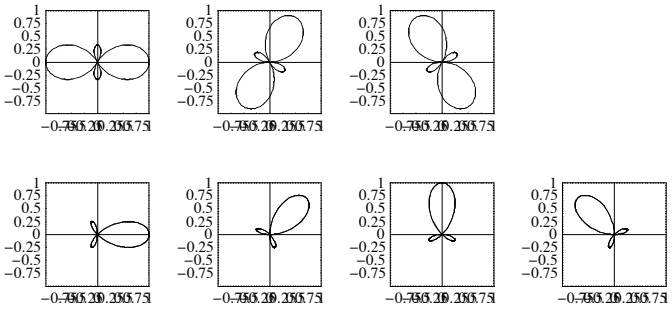

Figure 19.4 Top row: polar plot of the three coefficients to construct the rotated second order Gaussian derivative kernel. Bottom row: Polar plots of the 4 coefficients for the third order rotated derivative operator. A polar plot $f(\theta)$ is the radial plot of the radius f versus the angle θ.

The weights are found from a set of linear equations originating from the steeribility constraint. So the third order derivative $G_{xxx}|_{\phi}$ under an arbitrary angle of ϕ with the x-axis can be constructed from four basis kernels, each 45 degrees rotated, i.e.

$$\begin{aligned} G_{xxx}|_{\phi} = & \tfrac{1}{2}\,(\mathrm{Cos}[3\,\phi] + \mathrm{Cos}[\phi])\,G_{xxx}\,\big|_{0} \\ & + \tfrac{1}{4}\left(-\sqrt{2}\,\mathrm{Cos}[3\,\phi] + \sqrt{2}\,\mathrm{Cos}[\phi] + \sqrt{2}\,\mathrm{Sin}[3\,\phi] + \sqrt{2}\,\mathrm{Sin}[\phi]\right) G_{xxx}\,\big|_{\frac{\pi}{4}} \\ & + \tfrac{1}{2}\,(-\mathrm{Sin}[3\,\phi] + \mathrm{Sin}[\phi])\,G_{xxx}\,\big|_{\frac{\pi}{2}} \\ & + \tfrac{1}{4}\left(\sqrt{2}\,\mathrm{Cos}[3\,\phi] - \sqrt{2}\,\mathrm{Cos}[\phi] + \sqrt{2}\,\mathrm{Sin}[3\,\phi] + \sqrt{2}\,\mathrm{Sin}[\phi]\right) G_{xxx}\,\big|_{\frac{3\pi}{4}} \end{aligned}$$

When we fill in $\phi = -\pi/6$ we get the formula from the beginning of this chapter. It is instructive to look at the polar plots of these coefficients. This is the plot of the function as a function of the angle, with the amplitude of the function as distance to the origin. We quickly see then how the angular weights are distributed over the orientations, and we recognize the orientations of the basis functions, see figure 19.4.

19.5 Steering with Cartesian partial derivatives

When we just rotate our coordinates, we get a particular convenient representation for computer implementations. We use the same strategy as developed for the gauge coordinates in chapter 6. Instead of a locally adaptive set of directions for our frame to the orientation of the gradient vectorfield, we now choose a fixed rotation of our frame vectors.

We use the same formulas, and notice that the orientation of the {Lx, Ly} unit vector frame is given by {Sin(ϕ), Cos(ϕ)} where ϕ is our required angle of rotation:

```
Unprotect[gDϕ];
gDϕ[im_, nv_, nw_, σ_, ϕ_] :=
 Module[{Lx, Ly, v, w, im0}, v = {-Ly, Lx}; w = {Lx, Ly};
  Simplify[Nest[(v.{∂x #, ∂y #} &), Nest[(w.{∂x #, ∂y #} &), L[x, y], nw],
     nv] /. {Lx → Sin[ϕ], Ly → -Cos[ϕ]}]]
```

We denote **gDϕ[im_,nv_,nw_,σ_,ϕ_]** our rotated Gaussian derivative operator, and define this function in the appropriate way by repeated action of the differential operators (with **Nest**) and *thereafter* replacing the fixed direction {Lx, Ly} with the particular choice of {Sin(ϕ), $-$Cos(ϕ)}. A final step is the replacement of any appearance of a derivative into our regular (and now familiar) multi-scale Gaussian derivative operator **gD.**

Some examples: Here is the third derivative G_{xxx} rotated over $-\pi/6$ (30 degrees clockwise), the example of the beginning of this chapter, both in explicit formula and plotted at a scale of $\sigma = 15$ pixels (128^2 image):

```
im =.; gDϕ[im, 3, 0, 15, -π / 6] // shortnotation
```

$$\frac{1}{8}\left(3\sqrt{3}\,L_{xxx} - 9\,L_{xxy} + 3\sqrt{3}\,L_{xyy} - L_{yyy}\right)$$

And a 4th order example:

```
gDϕ[im, 3, 1, 15, π / 5] // shortnotation
```

$$\frac{1}{32}\left(\left(\sqrt{50-10\sqrt{5}}+2\sqrt{10-2\sqrt{5}}\right)L_{xxxx}+8\,L_{xxxy}-6\sqrt{10-2\sqrt{5}}\,L_{xxyy}-8\sqrt{5}\,L_{xyyy}-\sqrt{50-10\sqrt{5}}\,L_{yyyy}\right)$$

Task 19.1 Show that L_{xxxy}, when rotated over $\pi/2$, gives L_{xyyy}. Explain this.

The numerical version for discrete images is **gDϕN**:

```
Unprotect[gDϕN];
gDϕN[im_, nv_, nw_, σ_, ϕ_] := gDϕ[im, nv, nw, σ, ϕ] /.
   Derivative[n_, m_][L][x, y] → gD[im0, n, m, σ] /. im0 -> im
im = Table[0, {128}, {128}]; im[[64, 64]] = 100;
DisplayTogetherArray[ListDensityPlot /@
   {gD[im, 3, 0, 15], gDϕN[im, 3, 0, 15, -π / 6]}, ImageSize -> 380];
```

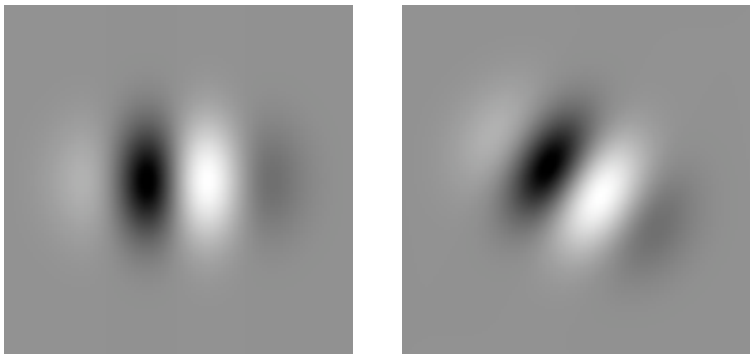

Figure 19.5 Left: the third order Gaussian derivative kernel to x. Right: the same kernel rotated 30 degrees clockwise, calculated from the expression in partial Cartesian derivatives above.

This is the proper multi-scale directional derivative. *Any* angle can now readily be constructed, no more need for 'only in 8 directions'.

To summarize this section we show the figures of the two different bases discussed. The first basis is a good starting point for models of oriented simple cells in the primary visual cortex. In chapter 9 we have seen that the orientation columns in the mammalian visual cortex contain the simple cells in a strikingly regular arrangement of ordered orientation. All orientations are present to analyze the local visual field in the hypercolumns, the arrangement in a pinwheel fashion. This representation is the topic of the last section of this chapter. The second basis, made up of Cartesian separable functions, is the basis of choice for computer implementations as these functions are just our familiar Gaussian derivative convolution kernels (**gD**). As we now have all tools for making the oriented kernels, we show both multi-scale steering bases for Gaussian derivative kernels below for the second order derivative.

```
im = Table[0, {128}, {128}]; im[[64, 64]] = 100;
Block[{$DisplayFunction = Identity},
n = 3; θn = Append[Table[i π/(n + 1), {i, 0, n}], -π / 6];
p1 = ListDensityPlot[gDϕN[im, 2, 0, 15, #]] & /@ θn;
p2 = Apply[ListDensityPlot[gD[im, #1, #2, 15]] &,
   {{3, 0}, {1, 2}, {2, 1}, {0, 3}}, 2];
t1 = Graphics[Text["\!\(\*
StyleBox[\"⇒\",\nFontFamily->\"Courier New
      \",\nFontSize->24,\nFontWeight->\"Plain\"]\)", {0, 0}]];
tt1 = Insert[p1, t1, 5]]; tt2 = {p2, t1, p1[[5]]} // Flatten;
Show[GraphicsArray[{tt1, tt2}], ImageSize -> 350];
```

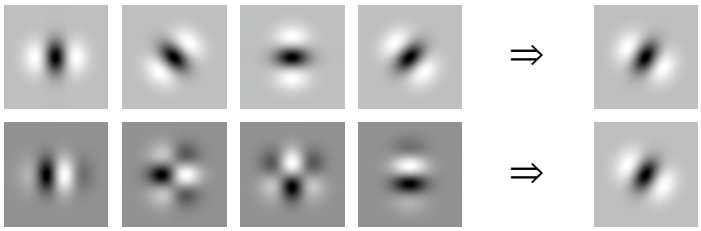

Figure 19.6 Two different sets of basis functions can be used for the construction of a steered Gaussian derivative kernel. Upper row: the basis set is formed from rotated versions of the kernel itself; bottom row: the basis set is formed from Cartesian x, y-separable Gaussian derivative kernels. The weights are calculated in the section above. Kernel as in figure 19.5.

19.6 Detection of stellate tumors

Computer-aided diagnosis (CAD) is the branch of computer vision in medical imaging concerned with the assistance of the computer in the diagnostic process. This is a rapidly growing field. There are two major reasons for its growth: a increasing array of methodologies outperform the human diagnosis in specificity, and the sheer volume of medical diagnostic data necessitates support. The system is always used as a 'second opinion' system, the final responsibility of the diagnosis is always laid on the medal specialist.

Recently, a number of commercial products have acquired FDA approval to put the system on the market and into clinical use. See e.g. the web pages of R2 Technology (www.r2tech.com), Deus Technologies (www.deustech.com) and Fujifilm (www.fujifilm.com). It is expected that more products soon will follow.

CAD up till now is mainly applied in fields where large scale screening of patients is performed. Two classical areas are screening for microcalcifications and stellate (stella = Latin: star; stellate = star-shaped) mammographic tumors (also called *spiculated* lesions), and screening for X-thorax deviations (tuberculosis screening, lung cancer etc.). We discuss the detection of stellate tumors in mammography. The following procedure has first been presented by Karssemeijer [Karssemeier1995a, Karssemeier1996a]. This is now the basis of one of the methods employed by R2 Technology.

A mammogram is a 2D X-ray photograph of the female breast, taken at high resolution (typically 2000^2 or higher) and with a low X-ray tube voltage (typically 28-35 kiloVolt), in

order to enhance the contrast between the soft tissue structures. The structure of the tissue is highly tubular: many channels are present, all converging in a tree-like structure to the nipple, so when a tumor expands, it is likely its outgrowth follows the channels. This gives often rise to a particular stellate pattern seen around the shadow of the lesion.

The *geometric reasoning:* when we investigate a pixel for the presence of a stellate structure, we actually have to investigate its immediate surrounding for the presence of lines oriented towards the pixel. Because we study a large group of surrounding pixels, we can approach a good statistical mean. The local orientation is detected by convolution of the second order Gaussian derivative, a 'classical bar-detector'. The oriented kernel is a function of L_{xx}, L_{xy} and L_{yy}:

```
Clear[ϕ, σ]; gDϕ[im, 2, 0, σ, ϕ] // shortnotation
```

```
Cos[ϕ]^2 L_xx + Sin[2 ϕ] L_xy + Sin[ϕ]^2 L_yy
```

For each pixel in the neigborhood of the pixel for which we inspect a stellated surround, we calculated the three 'basis' derivatives L_{xx}, L_{xy} and L_{yy}. We create a kernel for the same area where each pixel indicates under which angle this pixel is located with respect to the central pixel. In this kernel we calculate again a triplet, now of the trigonometric coefficients of the second order oriented Gaussian derivative.

We use the speed of **ListCorrelate** to multiply all triplets in the kernel with the triplets of basis derivatives in the same area in the mammogram. The resulting area is somewhat smaller than the original image. It is not useful to take information into account from a periodic or mirrored boundary. The input image is **mammo**, at scale **σ** of the differential operator and **size** is the size of the search region:

```
stellatedetection[mammo_, σ_, size_] :=
  Module[{derivs, kernel}, derivs = Transpose[{-gD[mammo, 2, 0, σ],
      gD[mammo, 1, 1, σ], gD[mammo, 0, 2, σ]}, {3, 1, 2}];
kernel = Table[With[{ϕ = ArcTan[x, y]}, {Cos[ϕ]^2, Sin[2 ϕ], Sin[ϕ]^2}],
     {x, -size - 0.5, size + 0.5}, {y, -size - 0.5, size + 0.5}];
Flatten /@ ListCorrelate[kernel, derivs]];
```

We start with an artificial mammogram of 200x200 pixels, containing two stellate patterns with diameters of 30 and 20 pixels resp. (see figure 19.8, left), and uniformly distributed noise:

```
insert[mammo_, stellate_, x_, y_] :=
  Module[{ydim, xdim, x1, y1}, {ydim, xdim} = Dimensions[stellate];
   locsm = Flatten[Table[{y1, x1},
      {y1, y, y + ydim - 1}, {x1, x, x + xdim - 1}], 1];
   locss = Flatten[Table[{y1, x1}, {y1, 1, ydim}, {x1, 1, xdim}], 1];
   ReplacePart[mammo, stellate, locsm, locss]];

stellate[diam_] := Table[
   If[x == y || x == Round[diam / 2] || y == Round[diam / 2] || x == diam - y, 1, 0],
   {x, diam}, {y, diam}];
mammo = Table[0, {y, 200}, {x, 200}];
```

```
noise = Table[Random[], {200}, {200}];
mammo = insert[mammo, stellate[30], 50, 70];
mammo = insert[mammo, stellate[20], 120, 140] + noise;
```

We extract the n largest maxima with the following function, which was first defined in chapter 13:

```
nMaxima[im_, n_] := Module[{p, d = Depth[im] - 1},
   p = Times @@ Table[(Sign[im - Map[RotateLeft, im, {i}]] + 1)
        (Sign[im - Map[RotateRight, im, {i}]] + 1), {i, 0, d - 1}] / 4^d;
   maxs = Take[Reverse[Union[{10 Extract[im, #], Reverse[#]} & /@
        Position[p, 1]]], n];
   Apply[{10 #1 / maxs[[1, 1]], #2} &, maxs, {1}]];
```

The resulting detection is indicated with circles. The radii of the circles indicate the likelihood of being the center of a stellate region. The radii are normalized to the highest likelihood with a radius of 10 pixels.

```
{ydim, xdim} = Dimensions[mammo]; size = 20;
n = 2;
smammo = Take[mammo, {1 + size, ydim - size - 1}, {1 + size, xdim - size - 1}];
DisplayTogetherArray[{ListDensityPlot[smammo],
   ListDensityPlot[p1 = stellatedetection[mammo, 4, 20], Epilog →
     {Red, Circle @@@ Reverse /@ nMaxima[p1, n]}]}, ImageSize → 450];
```

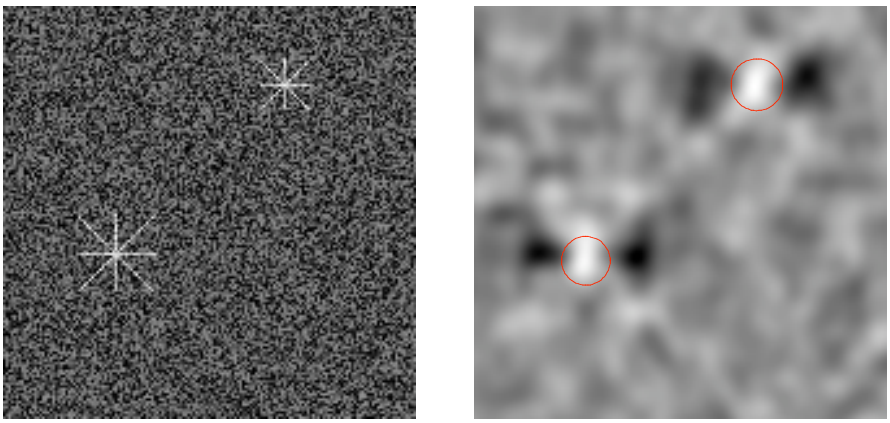

Figure 19.7 Computer-aided diagnosis in mammography: detection of two artificial stellate tumors in a noisy mammogram. Left: the input artificial mammogram (size 200x200 pixels) with 2 stellate regions. Right: the detected cumulative response of an oriented second order Gaussian derivative kernel, integrated over an area of 20x20 pixels.

▲ Task 19.2 Experiment with different values for the following parameters:
- the scale σ of the differential operator;
- the signal to noise ratio of the input image;
- the area of the search around each pixel;
- the distance between two stellate lesions;

- the size of the stellate lesion;
- the weight over the search area as a function of distance to the central pixel.

Now for a digital mammogram:

```
mammo = Import["mammogram04.jpg"][[1, 1]];
{ydim, xdim} = Dimensions[mammo];
detected = stellatedetection[mammo, σ = 3, size = 50]; n = 3;
smammo = Take[mammo, {size + 10, -size - 11}, {size + 10, -size - 11}];
sdetected = Take[detected, {10, -10}, {10, -10}];
circles = {Red, Circle @@@ Reverse /@ nMaxima[sdetected, n]};
DisplayTogetherArray[ListDensityPlot[#, Epilog → circles] & /@
   {smammo, sdetected}, ImageSize → 480];
```

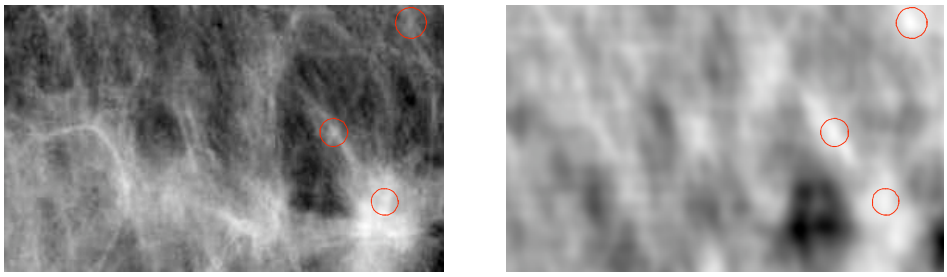

Figure 19.8 Left: Selection from a digital mammogram (size 403x291 pixels). Right: the detected cumulative response of an oriented second order Gaussian derivative kernel at $\sigma = 3$ pixels, integrated over an area of 50x50 pixels. The number of maxima to be reported is 3.

▲ Task 19.3 Experiment with other line detection kernels, such as a strongly elliptical oriented Gaussian kernel of zeroth order.

▲ Task 19.4 Find images with stellated tumors on the internet. See e.g. marathon.csee.usf.edu/Mammography/Database.html.

Note that this section only introduces the notion of using orientation sensitive responses in a *geometric reasoning* scheme, which might be part of a computer-aided diagnosis procedure. The method described above should never be used in any diagnostic judgement.

19.7 Classical papers and student tasks

The paper by Freeman and Adelson [Freeman1991a] formed the basis of this section. The references therein give a good overview of the literature on steerable filters. Other instructive (and historical) papers are by Jan Koenderink in his classical paper on receptive field models [Koenderink1988b], by Pietro Perona [Perona1991a, Perona1992, Perona1995], by Wolfgang Beil [Beil1994], by Per-Erik Danielson [Danielson1990] and Eduard Simoncelli and coworkers [Simoncelli1995, Simoncelli1996a, Farid1997a]. The construction of rotated partial derivatives can also be set up with Lie group theory, finding the Lie 'infinitesimal generators'. The papers by Michaelis and Sommer [Michaelis1995a, Michaelis1995b] and by Teo and Hel-Or [Teo1998] give a fine introduction, many examples and references of this powerful technique. One of the early theoretical studies of orientation tuning in the context of the front-end visual system is by Daughman [Daughman1983, Daughman1985]. See for an invertible orientation 'bundle' the paper by Kalitzin [Kalitzin1997a, Kalitzin1998a]. Multi-scale orientation analysis, based on models inspired by the 'spokewheel' structure as observed in the columns in the visual primary cortex, is a promising terrain for perceptual grouping research.

Task 19.5 Find the expressions for the weighting functions for a mixed Gaussian derivative kernel, e.g. $\frac{\partial^3 G}{\partial x^2 \partial y}$, for both bases.

Task 19.6 Pentland [Pentland1990] has suggested that shape-from-shading analysis can be performed by a linear filtering operation in many situations, e.g. when the reflectance function is approximately linear. Freeman and Adelson [Freeman1991a] give suggestions how to implement this with steerable functions. Make a *Mathematica* scale-space implementation for linear shape from shading.

Task 19.7 Make a *Mathematica* implementation, based on the results developed in this chapter, for 3D steerable filters. See again Freeman and Adelson [Freeman1991a] for theoretical support.

Task 19.8 Trabecular bone (the sponge-like interior of most of our bones) has an intricate multi-scale orientation structure. Extract from 2D and 3D datasets the local orientation structure at multiple scales, and come up with sensible definitions for the local structure of the bone. For inspiration, see [TerHaarRomeny1996f, Niessen1997b, Lopéz200a].

19.8 Summary of this chapter

The local vectorfield specified by the gradient and its clockwise rotated perpendicular vector form the first order orientation structure. This vectorfield is also specified by the Eigenvectors of the local *structure matrix*.
The local vectorfield specified by the unit vectors of the principal curvature vectors in each point form the second order orientation structure. This vectorfield is also specified by the Eigenvectors of the local *Hessian matrix*.
Gaussian derivative kernels are steerable kernels. They can be constructed in any direction (as directional derivative operators) in two ways: as a polynomial expressed in rotated versions of the Gaussian derivative kernel itself, or as a polynomial combination of Cartesian partial derivatives.
The geometric reasoning for the detection of structures can now be expanded to the inclusion of responses to oriented structures, such as lines. An example is given for the detection of stellate tumors in mammography.