# 2. Foundations of scale-space

*"There are many paths to the top of the mountain,*
*but the view is always the same"* -Chinese proverb.

## 2.1 Constraints for an uncommitted front-end

To compute any type of representation from the image data, information must be extracted using certain *operators* interacting with the data. Basic questions then are: Which operators to apply? Where to apply them? How should they look like? How large should they be?

Suppose such an operator is the derivative operator. This is a difference operator, comparing two neighboring values at a distance close to each other. In mathematics this distance can indeed become infinitesimally small by taking the limit of the separation distance to zero, but in physics this reduces to the sampling distance as the smallest distance possible. Therefore we may foresee serious problems when we deal with such notions as mathematical differentiation on discrete data (especially for high order), and sub-pixel accuracy.

From this moment on we consider the aperture function as an operator: we will search for *constraints* to pin down the exact specification of this operator. We will find an important result: for an unconstrained front-end there is a *unique* solution for the operator. This is the Gaussian kernel $g(x;\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{x^2}{2\sigma^2}}$, with $\sigma$ the *width* of the kernel. It is the same bell-shaped kernel we know from probability theory as the probability density function of the *normal distribution*, where $\sigma$ is the *standard deviation* of the distribution.

Interestingly, there have been many derivations of the front-end kernel, all leading to the unique Gaussian kernel.

This approach was pioneered by Iijima (figure 2.2) in Japan in the sixties [Iijima1962], but was unnoticed for decades because the work was in Japanese and therefore inaccessible for Western researchers.

Independently Koenderink in the Netherlands developed in the early eighties a rather complete multi-scale theory [Koenderink1984a], including the derivation of the Gaussian kernel and the linear diffusion equation.

```
<< FrontEndVision`FEV`;
```

$$\sigma = 1;\ \mathtt{Plot}\left[\frac{1}{\sqrt{2\,\pi\,\sigma^2}}\ \mathtt{E}^{-\frac{x^2}{2\,\sigma^2}},\ \{x,\ -4,\ 4\},\ \mathtt{ImageSize} \to 200\right];$$

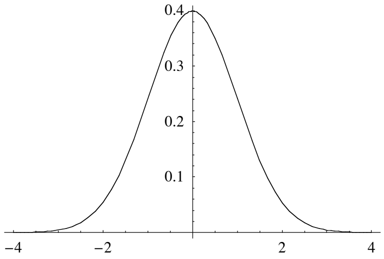


Figure 2.1 The Gaussian kernel with unit standard deviation in 1D.

Koenderink was the first to point out the important relation to the receptive field families in the visual system, as we will discuss in forthcoming chapters. Koenderink's work turned out to be monumental for the development of scale-space theory. Lindeberg pioneered the field with a tutorial book [Lindeberg1994a]. The papers by Weickert, Ishikawa and Imija (who together discovered this Japanese connection) present a very nice review on these early developments [Weickert1997a, Weickert1999a].

```
Show[Import["Iijima.gif"], ImageSize -> 150];
```

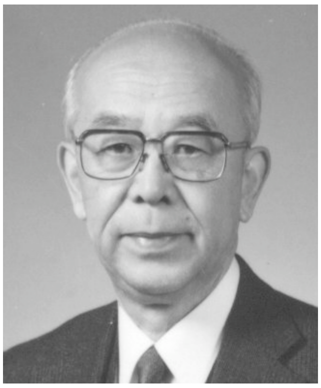


Fig. 2.2 Prof. Taizo Iijima, emeritus prof. of Tokyo Technical University, Japan, was the first to publish the axiomatic derivation of 'the fundamental equation of figure'.

We will select and discuss two fundamentally different example approaches to come to the Gaussian kernel in this book:

1. An axiomatic approach based on dimensional analysis and the notion of having 'no preferences' (section 2.2);
2. An approach based on the maximization of local entropy in the observation (section 2.5);

## 2.2 Axioms of a visual front-end

The line of reasoning presented here is due to Florack et al. [Florack1992a]. The requirements can be stated as *axioms*, or postulates for an uncommitted visual front-end. In essence it is the mathematical formulation for being uncommitted: "we know nothing", or "we have no preference whatsoever".

- **linearity**: we do not allow any nonlinearities at this stage, because they involve knowledge of some kind. So: no knowledge, no model, no memory;

- **spatial shift invariance**: no preferred location. Any location should be measured in the same fashion, with the same aperture function;

- **isotropy**: no preferred orientation. Structures with a particular orientation, like vertical trees or a horizontal horizon, should have no preference, any orientation is just as likely. This necessitates an aperture function with a circular integration area.

- **scale invariance**: no preferred size, or scale of the aperture. Any size of structure, object, texture etc. to be measured is at this stage just as likely. We have no reason to look only through the finest of apertures. The visual world consists of structures at any size, and they should be measured at any size.

In order to use these constraints in a theory that sets up the reasoning to come to the aperture profile formula, we need to introduce the concept of dimensional analysis.

### 2.2.1 Dimensional analysis

Every physical unit has a *physical dimension.*

It is this that mostly discriminates physics from mathematics. It was Baron Jean-Baptiste Fourier who already in 1822 established the concept of dimensional analysis [Fourier1955]. This is indeed the same mathematician so famous for his Fourier transformation.

```
Show[Import["Fourier.jpg"], ImageSize → 140];
```

Figure 2.3 Jean-Baptiste Fourier, 1792-1842.

Fourier described the concept of dimension analysis in his memorable work entitled "Théorie analytique de la chaleur" [Fourier1955] as follows: "*It should be noted that each physical quantity, known or unknown, possesses a dimension proper to itself and that the terms in an equation cannot be compared one with another unless they possess the same dimensional exponent*".

When a physicist inspects a new formula he invariably checks first whether the dimensions are correct. It is for example impossible to add meters to meters/second. One of the most fundamental laws in physics is that the physical laws should be rock solid, independent of a chosen description, anywhere and anytime. This law is the *law of scale invariance*, which indicates that we have full freedom of reparametrization:

[Law of Scale Invariance] **Physical laws must be independent of the choice of fundamental parameters.**

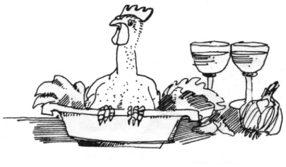
'Scale invariance' here refers to the notion of scale with respect to dimensional units (remember the microns, kilometers or milliseconds as the aperture size of the measurement instrument).

In essence the law of scale invariance states that the left and right part of the equation of a physical equation should have the same dimensional units, and they should describe the same process, whether expressed in Cartesian or polar coordinates.

Core in dimensional analysis is that when the dimensions in a complex physical system are considered, only a limited number of dimensionless combinations can be made: the basis or null-space of the system. It is an elegant and powerful tool to find out basic relations between physical entities, or even to *solve* a problem. It is often a method of first choice, when no other information is available. It is often quite remarkable how much one can deduct by just using this technique. We will use dimensional analysis to establish the expression defining the basic linear isotropic scale-space kernel. First some examples which illustrate the idea.

### 2.2.2 The cooking of a turkey

```
Show[Import["Turkey.gif"], ImageSize -> 150];
```



This example is taken from the delightful paper by Geoffrey West [West1988]. When cooking a turkey, or a goose, there is the problem of knowing how long to cook the bird in the oven, given the considerable variation that can exist in its weight and size.

Many (inferior) cookbooks specify simply something like '20 minutes per pound', implying a linear relation. There are superior cookbooks, however, such as the 'Better Homes and

Gardens Cookbook' that recognize the nonlinear nature of this relationship. In figure 1.4 we have adapted the graph from this cookbook, showing a log-log plot of the cooking time as a function of the weight of the turkey. The slope of the linear relation is about 0.6. It turns out that we can predict this relation just by dimensional analysis.

```
data = {{5, 3}, {7, 3.8}, {10, 4.5}, {14, 5}, {20, 6}, {24, 7}};
LogLogListPlot[data, PlotJoined -> False,
  Ticks -> {{5, 10, 15, 20}, {3, 5, 7}}, Frame -> True,
  FrameLabel -> {"Cooking time (hrs)", "Weight (lb)"},
  PlotStyle -> PointSize[0.02], ImageSize -> 220];
```

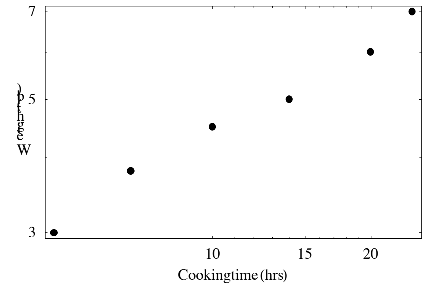


Figure 2.4. Turkey cooking time as a (nonlinear) function of turkey weight. The slope of the log-log plot is about 0.6. (Based on a table in *Better Homes and Gardens Cookbook*, Des Moines: Meridith Corp., 1962, p. 272).

Let us list the physical quantities that are involved:
- the temperature distribution inside the turkey $T$, in degrees Kelvin,
- the oven temperature $T_0$ (both measured relative to the outside air temperature), in degrees Kelvin,
- the bird density $\rho$ in $\mathrm{kg}/m^3$,
- the diffusion coefficient of the turkey $\kappa$ from Fourier's diffusion equation for $T$: $\frac{\partial T}{\partial t} = \kappa \Delta T$ where $\Delta$ is the Laplacian operator $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$, in $m^2/\mathrm{sec}$,
- the weight of the turkey $W$, in kg,
- and the time $t$ in seconds.

In general, for the dimensional quantities in this problem, there will be a relationship of the form $T = f(T_0, W, t, \rho, \kappa)$. We can make a matrix of the units and their dimensions:

```
m = {{0, 0, 0, 0, -3, 2},
   {0, 0, 0, 1, 0, -1}, {0, 0, 1, 0, 1, 0}, {1, 1, 0, 0, 0, 0}};
TableForm[m, TableHeadings -> {{"length", "time", "mass", "degree"},
   {"T0", "T", "W", "t", "ρ", "κ"}}]
```

```
          T0     T      W      t      ρ      κ
length    0      0      0      0      -3     2
time      0      0      0      1      0      -1
mass      0      0      1      0      1      0
degree    1      1      0      0      0      0
```

A matrix can be described with a basis spanned by basis vectors, whose linear combinations satisfy the matrix equation `m.x == 0`. The command `NullSpace` gives us the list of basis vectors:

```
NullSpace[m]

{{0, 0, -2, 3, 2, 3}, {-1, 1, 0, 0, 0, 0}}
```

The (famous) Pi-theorem in dimensional analysis by Buckingham (see http://www.treasure-troves.com/physics/BuckinghamsPiTheorem.html) states that one can make as many independent dimensionless combinations of the physical variables in the system under study as the number of basis vectors of the nullspace of the dimension matrix `m`. These are determined by the nullspace.

So, for the turkey problem we can only construct two independent dimensionless quantities (just fill in the exponents given by the basis vectors): $\frac{\rho^2\,t^3\,\kappa^3}{W^2}$ and $\frac{T}{T_0}$.

So, from the nullspace vector `{-1,1,0,0,0,0}` we found $\frac{T}{T_0}$ and from `{0,0,-2,3,2,3}` we found $\frac{\rho^2\,t^3\,\kappa^3}{W^2}$. Because both these quantities are dimensionless one must be expressible in the other, giving the relationship: $\frac{T}{T_0} = f\left(\frac{\rho^2\,t^3\,\kappa^3}{W^2}\right)$. Note that since the lefthand side is dimensionless, the arbitrary function $f$ must be a dimensionless function of a dimensionless variable. This equation does not depend on the choice of units, since dimensionless units remain invariant to changes in scale: the scale invariance.

The graph in the cookbook can now be understood: when geometrically similar birds are considered, cooked to the same temperature distribution at the same oven temperature, there will be the following scaling law: $\frac{\rho^2\,t^3\,\kappa^3}{W^2} = \text{constant}$. If the birds have the same physical characteristics, which means the same $\rho$ and $\kappa$, we find that $t^3 = \frac{W^2}{\rho^2\,\kappa^3}$, so the cooking time $t$ is proportional to $W^{2/3}$ which nicely explains the slope.

### 2.2.3 Reynold's number

From [Olver1993] we take the example of the Reynold's number. We study the motion of an object in some fluid.

As physical parameters we have the resistance $D$ of the object (in $\frac{\text{kg}}{m\,s^2}$), the fluid density $\rho$ (in $\frac{\text{kg}}{m^3}$), the velocity relative to the fluid $v$ (in $\frac{m}{s}$), the object diameter $d$ (in $m$) and the fluid viscosity $\mu$ (in $\frac{\text{kg}}{m\,s}$). The dimension matrix becomes then:

```
m = {{-3, 1, 1, -1, -1}, {0, -1, 0, -1, -2}, {1, 0, 0, 1, 1}};
TableForm[m,
 TableHeadings -> {{"meter", "second", "kg"}, {"ρ", "v", "d", "μ", "D"}}]

          ρ       v       d       μ       D
meter     -3      1       1       -1      -1
second    0       -1      0       -1      -2
kg        1       0       0       1       1
```

We calculate the nullspace:

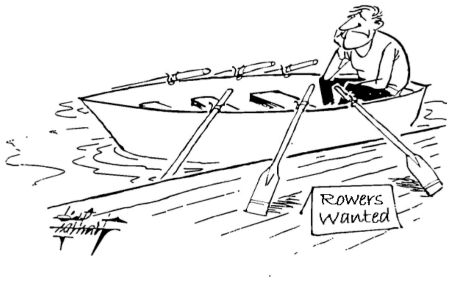
```
NullSpace[m]

{{-1, -2, 0, 0, 1}, {-1, -1, -1, 1, 0}}
```

From the nullspace we easily find the famous Reynolds number: $R = \frac{\rho v d}{\mu}$. The other dimensionless entity $\frac{D}{\rho v^2}$ is the friction factor.

### 2.2.4 Rowing: more oarsmen, higher speed?

```
Show[Import["Rowerswanted.gif"], ImageSize → 210];
```



Another illuminating example is the problem of the number of oarsmen in a competition rowing boat: Do 8 oarsmen need less time to row a certain distance, say 2000 meter, then a single skiffer, despite the fact that the water displacement is so much bigger? Let's study the physics again: We first find the relation for the drag force $F$ on a ship with length $l$ moving with velocity $v$ through a viscous fluid with viscosity $\mu$ and density $\rho$.

The final term to take into account in this physical setup is the gravity $g$. Again we can make a dimensional matrix for the six variables involved:

```
m = {{1, 1, 1, -1, -3, 1}, {-2, 0, -1, -1, 0, -2}, {1, 0, 0, 1, 1, 0}};
TableForm[m, TableHeadings ->
  {{"meter", "second", "kg"}, {"F", "l", "v", "μ", "ρ", "g"}}]
```

```
           F     l     v     μ     ρ     g
meter      1     1     1     -1    -3    1
second     -2    0     -1    -1    0     -2
kg         1     0     0     1     1     0
```

Figure 2.5 Dimensional matrix for the physics of drag of an object through water. $F$ is the drag force, $l$ resp $v$ are the length resp. the velocity of the ship, $\mu$ is the viscosity of the water.

and study the nullspace:

```
NullSpace[m]
```

```
{{0, 1, -2, 0, 0, 1}, {-1, 2, 2, 0, 1, 0}, {-1, 1, 1, 1, 0, 0}}
```

```
rowdata1 = {{1, 6.88}, {2, 6.80}, {4, 6.60}, {8, 5.80}};
rowdata2 = {{1, 7.01}, {2, 6.85}, {4, 6.50}, {8, 5.85}};
rowdata3 = {{1, 7.04}, {2, 6.85}, {4, 6.40}, {8, 5.95}};
rowdata4 = {{1, 7.10}, {2, 6.95}, {4, 6.50}, {8, 5.90}};
```

```
MultipleListPlot[rowdata1, rowdata2,
  rowdata3, rowdata4, Ticks -> {{1, 2, 4, 8}, Automatic},
  AxesLabel -> {"# of\noarsmen", "Time for\n2000 m (min)"},
  PlotJoined -> True, PlotRange -> {Automatic, {5, 8}}];
```

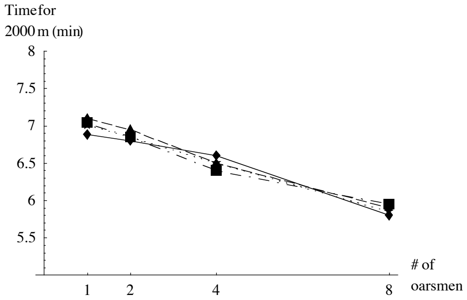


Figure 2.5. The results of best US regatta rowing (2000 m) of Summer 2000 for different numbers of oarsmen. The slope of the graph is about -1/9. Source: http://rowingresults.com/

The dimensionless units are: $\frac{v^2}{l g}$ (Froude's number), $\frac{F}{\rho v^2 l^2}$ (the pressure coefficient) and $\frac{l v \mu}{F}$ (the Poisseuille coefficient). So we have $\frac{F}{\rho v^2 l^2} = f\left(\frac{v^2}{l g}, \frac{l v \mu}{F}\right)$ or $F \approx \rho v^2 l^2 f$ where $f$ is a dimensionless number. The power $E$ produced by the $n$ oarsmen together to overcome the drag force $F$ is given by $F v$. Thus $E = F v = \rho v^3 l^2 f \approx n$ because $E$ is directly proportional to $n$.

The weight $W$ of a ship is proportional to the volume of displaced water (Archimedes law), so $W \approx l^3$. This implies $\frac{F}{W} \approx \frac{1}{l}$ which means that larger ships have advantages, because, for similar bodies, the ratio $\frac{F}{W}$ decreases as the size of the ship increases. We know $\rho = 1$ for water and $W \approx l^3$ (Archimedes again) and $W \approx n$ in good approximation, we find $v^3 \approx n^{1/3}$ so $v \approx n^{1/9}$. So eight oarsmen indeed go faster, though little, than less oarsmen.

There are several nice references to the technique of dimensional analysis [West1988, Pankhurst1964a, Olver1993], often with quite amusing examples, some of which were presented in this section.

| | | |
|---|---|---|
| Archimedes ' Number | Froude Number | Monin - Obukhov Length |
| Bingham Number | Grashof Number | Nusselt Number |
| Biot Number | Internal Froude Number | Péclet Number |
| Boussinesq Number | Mach Number | Prandtl Number |
| Critical Rayleigh Number | Magnetic Reynolds Number | Rayleigh Number |
| Ekman Number | Mason Number | Richardson Number |
| Fresnel Number | Moment of Inertia Ratio | Timescale Number |

Figure 2.6. A list of famous dimensional numbers. From Eric Weisstein's World of Physics. URL: http://scienceworld.wolfram.com/physics/topics/UnitsandDimensionalAnalysis.html.

Many scaling laws exist. In biology scaling laws have become a powerful technique to find surprising relations (see for a delightful easy-to-read overview the book by McMahon and Bonner [McMahon1983] and the classical book by Thompson [Thompson1942]).

## 2.3 Axiomatic derivation of the Gaussian kernel

The dimensional analysis discussed in the previous section will now be used to derive the Gaussian kernel as the unique solution for the aperture function of the uncommitted visual front-end.

We do the reasoning in the Fourier domain, as this turns out to be easier and leads to smaller equations. We give the theory for 2D. We will see that expansion to other dimensionalities is straightforward. We use scripted symbols for variables in the Fourier domain. We consider 'looking through an aperture'. The matrix $m$ and the nullspace become:

```
m = {{1, -1, -2, -2}, {0, 0, 1, 1}};
TableForm[m,
 TableHeadings -> {{"meter", "candela"}, {"σ", "ω", "L0", "L"}}]
```

```
              σ      ω      L0     L
meter         1      -1     -2     -2
candela       0      0      1      1
```

Figure 2.8 Dimensional matrix for the physics of observation through an aperture.

```
NullSpace[m]
```

```
{{0, 0, -1, 1}, {1, 1, 0, 0}}
```

were $\sigma$ is the size of the aperture, $\omega$ the spatial coordinate (frequency in the Fourier domain), $\mathcal{L}_0$ the luminance of the outside world (in candela per square meter: $\mathrm{cd}/m^2$), and $\mathcal{L}$ the luminance as processed in our system. The two dimensionless units $\frac{\mathcal{L}}{\mathcal{L}_0}$ and $\sigma\omega$ can be expressed into each other: $\frac{\mathcal{L}}{\mathcal{L}_0} = \mathcal{G}(\sigma\omega)$, where $\mathcal{G}$ is the kernel (filter, aperture) function in the Fourier domain to be found (the Gaussian kernel we are after). We now plug in our constraints, one by one.
No preference for location, together with the prerequisite for linearity, leads to the recognition of the process as a *convolution*. The aperture function is shifted over the whole image domain, with no preference for location: any location is measured (or filtered, observed) with the same aperture function (kernel, template, filter, receptive field: all the same thing). This is written for the spatial domain as:

$$L(x, y) = L_0(x, y) \otimes G(x, y) \equiv \int_{-\infty}^{\infty} L_0(u, v)\, G(x-u, y-v)\, du\, dv$$

In the Fourier domain, a convolution of functions translates to a regular product between the Fourier transforms of the functions: $\mathcal{L}(\omega_x, \omega_y) = \mathcal{L}_0(\omega_x, \omega_y)\ .\ \mathcal{G}(\omega_x, \omega_y)$

The axiom of isotropy translates into the fact that we now only have to consider the *length* $\|\vec{\varpi}\|$ of our spatial frequency vector $\vec{\varpi} = \{\omega_x, \omega_y\}$: $\omega = \|\vec{\varpi}\| = \sqrt{\omega_x{}^2 + \omega_y{}^2}$ . This is a scalar.

The axiom of scale-invariance is the core of the reasoning: when we observe (or blur) an observed image again, we get an image which is blurred with the *same* but wider kernel:

$\mathcal{G}(\omega\,\sigma_1)\,\mathcal{G}(\omega\,\sigma_2) = \mathcal{G}(\omega\,\sigma_1 + \omega\,\sigma_2)$. Only the exponential function is a general solution of this equation: $\mathcal{G}(\omega\,\sigma) = \exp\left((\alpha\,\omega\,\sigma)^p\right)$ where $\alpha$ and $p$ are some arbitrary constants.
We must raise the argument here to some power $p$ because we are dealing with the *dimensionless* parameter $\omega\,\sigma$. In general, we don't know $\alpha$ or $p$, so we apply the following constraint: isotropy.
The dimensions are independent, thus separable: $\|\vec{\omega}\,\sigma\| = (\omega_1\,\sigma)\,\vec{e}_1 + (\omega_1\,\sigma)\,\vec{e}_2 + \ldots$ where the $\vec{e}_i$ are the basis unit coordinate vectors. Recall that the vector $\vec{\omega}$ ($\vec{\omega} = \omega_x\,\vec{e}_x + \omega_y\,\vec{e}_y + \omega_x\,\vec{e}_z$) in the Fourier domain is the set of spatial frequencies in the spatial dimensions. The magnitude of $\|\vec{\omega}\,\sigma\|$ is calculated by means of Pythagoras from the projections along $\vec{e}_i$, so we add the squares: $p = 2$. We further demand the solution to be real, so $\alpha^2$ is real. We notice that when we open the aperture fully, we blur everything out, so $\lim_{\sigma\omega\downarrow 0}\mathcal{G}(\omega\sigma) \to 0$ . This means that $\alpha^2$ must be negative. We choose $\alpha = -\frac{1}{2}$ . As we will see, this (arbitrary) choice gives us a concise notation of the diffusion equation. So we get: $\mathcal{G}(\vec{\omega}, \sigma) = \exp(-\frac{1}{2}\,\sigma^2\,\omega^2)$. We go to the spatial domain with the inverse Fourier transform:

$$\mathtt{Clear[\sigma];}$$

$$\mathtt{g[x\_,\ \sigma\_]\ =\ Simplify\Big[InverseFourierTransform\Big[Exp\Big[-\frac{\sigma^2\ \omega^2}{2}\Big],\ \omega,\ x\Big],\ \sigma > 0\Big]}$$

$$\frac{e^{-\frac{x^2}{2\,\sigma^2}}}{\sigma}$$

The last notion to use it that we want a *normalized* kernel. The integrated area under the curve must be unity:

$$\mathtt{Simplify\Big[Integrate\Big[\frac{e^{-\frac{x^2}{2\,\sigma^2}}}{\sigma},\ \{x,\ -\infty,\ \infty\}\Big],\ \sigma > 0\Big]}$$

$$\sqrt{2\,\pi}$$

We divide by this factor, so we finally find for the kernel: $G(\vec{x}, \sigma) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{\vec{x}.\vec{x}}{2\sigma^2}\right)$. This is the *Gaussian kernel*, which is the Green's function of the linear, isotropic *diffusion equation* $\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} = L_{\mathrm{xx}} + L_{\mathrm{yy}} = \frac{\partial L}{\partial t}$, where $t = 2\,\sigma^2$ is the variance.

Note that the 'derivative to scale' in the diffusion equation (as it is typically called) is the derivative to $2\sigma^2$, which also immediately follows from a consideration of the dimensionality of the equation. The variance $t$ has the dimensional unit of $m^2$. The original image is the *boundary condition* of the diffusion: it starts 'diffusing' from there.Green's functions are named in honor of the English mathematician and physicist George Green (1793-1841).

So from the prerequisites 'we know nothing', the axioms from which we started, we have found the Gaussian kernel as the *unique* kernel fulfilling these constraints. This is an important result, one of the cornerstones in scale-space theory. There have been more ways in which the kernel could be derived as the unique kernel. Weickert [Weickert1997a] gives a systematic and thorough overview of the historical schemes that have been published to derive the Gaussian.

## 2.4 Scale-space from causality

Koenderink presented in his famous paper "The structure of images" [Koenderink1984a] the elegant and concise derivation of the linear diffusion equation as the generating partial differential equation for the construction of a scale-space.

The arguments were taken from the physics of *causality*: when we increase the scale and blur the image further, we have the situation that the final blurred image is completely *caused by* the image we started from.

The previous level of scale is the cause of events at the next level. We first discuss the situation in 1D.

```
Clear[f]; f[x_] := Sin[x] + Sin[3 x]; gr = Plot[f[x], {x, -3, 3}, Epilog ->
(Arrow[{x, f[x]}, {x, f[x] + Sign[f''[x] ] .5}] /. Solve[f'[x] == 0, x]),
  AxesLabel -> {"x", "intensity"}, ImageSize -> 200];
```

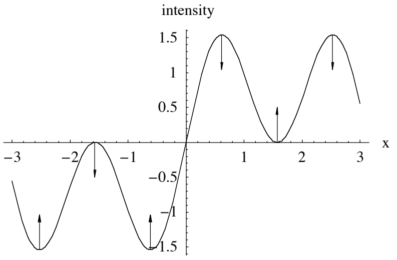


Figure 2.9 Under causal blurring signals can only change in the direction of less structure. Generation of new structure is impossible, so the signal must always be closed to above (seen from both sides of the signal). The arrows indicate the direction the intensity moves under blurring.

A higher level of scale contains always less structure. It is physically not possible that new structure is being generated. This is one of the most essential properties of a scale-space. We will encounter this property again when we consider nonlinear scale-spaces in chapter 19.

The direction of the arrows in figure 2.9 are determined by the fact if the extremum is a maximum or a minimum. In a maximum the intensity is bound to decrease, in a minimum the intensity is bound to increase. The second order derivative determines the *curvature* of the signal, and the sign determines whether the function is locally convex (in a maximum) or concave (in a minimum). We have the following conditions:

maximum: $\frac{\partial^2 u}{\partial x^2} < 0$, $\frac{\partial u}{\partial t} < 0$, intensity always decreasing;
minimum: $\frac{\partial^2 u}{\partial x^2} > 0$, $\frac{\partial u}{\partial t} > 0$, intensity always increasing.

These conditions can be summarized by $\frac{\partial^2 u}{\partial x^2} \frac{\partial u}{\partial t} > 0$.

The most important property to include next is the requirement of linearity: the second order derivative to space $\frac{\partial^2 u}{\partial x^2}$ is linearly related to the first order derivative to scale $\frac{\partial u}{\partial t}$, so: $\frac{\partial^2 u}{\partial x^2} = \alpha \frac{\partial u}{\partial t}$. We may resample any scale axis in such a way that $\alpha = 1$ so we get $\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}$. This is the 1D *linear isotropic diffusion equation*, an important result. The Green's function of the linear diffusion equation is the Gaussian kernel $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$, which means that any function upon which the diffusion is applied, is convolved with this Gaussian kernel.

We can check (with $t = \frac{1}{2}\sigma^2$, the double **==** means equation, test of equality, not assignment):

$$\mathtt{Clear[x,\ t];}\ \partial_{x,x}\left(\frac{1}{\sqrt{4\,\pi\,t}}\,\mathrm{E}^{-\frac{x^2}{4\,t}}\right) == \partial_t\left(\frac{1}{\sqrt{4\,\pi\,t}}\,\mathrm{E}^{-\frac{x^2}{4\,t}}\right)\ \mathtt{//\ Simplify}$$

```
True
```

Also any spatial derivative of the Gaussian kernel is a solution. We test this for the first order derivative:

$$\mathtt{Clear[x,\ t];}\ \partial_{x,x}\left(\partial_x\left(\frac{1}{\sqrt{4\,\pi\,t}}\,\mathrm{E}^{-\frac{x^2}{4\,t}}\right)\right) == \partial_t\left(\partial_x\left(\frac{1}{\sqrt{4\,\pi\,t}}\,\mathrm{E}^{-\frac{x^2}{4\,t}}\right)\right)$$

```
True
```

▲ Task 2.1 Show that this holds true for any order of derivative, including mixed derivatives for 2- or higher dimensional Gaussians.

In 2D and higher dimensions the reasoning is the same. Again we demand the function to be closed to the top. No new structure can emerge.

The requirement for the sign of the second order derivative is now replaced by the requirement on the sign of the rotation invariant *Laplacian*, $\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}$.

The reasoning leads to $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$, the 2D linear isotropic diffusion equation, or $\Delta u = \frac{\partial u}{\partial t}$ in any dimension (the Laplacian *operator* $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is often indicated as $\Delta$).

In the following chapters we will study the Gaussian kernel and the Gaussian derivatives in detail. First we present in the next section an alternative and particularly attractive alternative approach to derive the scale-space kernel starting from the maximization of *entropy*.

## 2.5 Scale-space from entropy maximization

An alternative way to derive the Gaussian kernel as the scale-space kernel of an uncommitted observation is based on the notion that the 'uncommittedness' is expressed in a statistical way using the entropy of the observed signal. The reasoning is due to Mads Nielsen, IT-University Copenhagen [Nielsen1995, Nielsen1997a]:

First of all, we want to do a measurement. We have a *device* which has some integration area over which the measurement is done. As we have seen before, the area (or length or volume) of this detector should have a finite width. It cannot be brought down to zero size, because then nothing would be measured anymore.

The measurement should be done *at all locations* in the same way, with either a series of identical detectors, or the same detector measuring at all places. In mathematical language this is stating that the measurement should be *invariant* for translation.

We also want the measurement to be *linear* in the signal to be measured, for example the intensity. This means that when we measure a signal twice as strong, also the output of the measurement should be doubled, and when we measure two signals, the measurement of the sum of the signals should be equal to the sum of the individual measurements. In mathematical language again this is called invariance for translation along the intensity axis.

These requirements lead automatically to the formulation that the observation must be a *convolution*: $h(x) = \int_{-\infty}^{\infty} L(\alpha)\, g(x-\alpha)\, d\alpha$.

$L(x)$ is the observed variable, in this example the luminance, $g(x)$ is our aperture, $h(x)$ the result of our measurement.

The aperture function $g(x)$ should be a *unity* filter. Such a filter is called a *normalized* filter. Normalization means that the integral over its weighting profile should be unity: $\int_{-\infty}^{\infty} g(x)\, dx = 1$. The filter should not multiply the data with something other than 1.

The *mean* of the filter $g(x)$ should be at the location where we measure (say at $x_0$), so the expected value (or *first moment*) should be $x_0$: $\int_{-\infty}^{\infty} x\, g(x)\, dx = x_0$. Because we may take *any* point for $x_0$, we may take for our further calculations as well the point $x_0 = 0$, which makes life somewhat easier.

The *size* of the aperture is a very essential element. We want to be free in choice of this size, so at least we want to find a *family* of filters where this size is a free parameter. We can then monitor the world at all these sizes by 'looking through' the complete set of kernels simultaneously. We call this 'size' $\sigma$. It has the dimension of length, and is the yardstick of our measurement. We call it the inner scale. Every physical measurement has an inner scale. It can be $\mu$m, milliseconds, light-years, anything, but for every dimension we need a yardstick. Here $\sigma$ is our yardstick. We can express distances in a measurement in "number of $\sigma$'s that we stepped around".

If we weight distances quadratically with our kernel we separate the dimensions: two *orthogonal* vectors fulfill $(a+b)^2 = a^2 + b^2$. Distances (or lengths) add up quadratically by Pythagoras' law. We call the weighted metric $\sigma^2$: $\int_{-\infty}^{\infty} x^2\, g(x)\, dx = \sigma^2$.

The last equation we add to the set that will lead to the final formula of the kernel, comes from the incorporation of the request to be as uncommitted as possible. We want no filter that has a preference for something, such as vertical structures, or squares or circles. Actually, we want, in statistical terms, the 'orderlessness' or disorder of the measurement as large as possible, there should be no ordering, ranking, structuring or whatsoever. Physically, this is expressed through the *entropy*, a measure for disorder. The entropy of very regular data is low, we just want maximal entropy. The formula for entropy of our filter is: $H = \int_{-\infty}^{\infty} g(x) \ln g(x)\, dx$ where `ln(x)` is the natural logarithm.

We look for the $g(x)$ for which the entropy is maximal, *given the constraints* that we derived before:

$$\int_{-\infty}^{\infty} g(x)\, dx = 1,\quad \int_{-\infty}^{\infty} x\, g(x)\, dx = 0 \text{ and } \int_{-\infty}^{\infty} x^2\, g(x)\, dx = \sigma^2.$$

When we want to find a maximum under a set of given constraints, we apply a standard mathematical technique named the *method of Euler-Lagrange equations* (see for an intuitive explanation of this method Petrou and Bosdogianni [Petrou1999a, page 258]).

This is a technique from the calculus of variations. We first make the Euler-Lagrange equation, or *Lagrangian*, by adding to the entropy term the constraints above, each multiplied with an unknown constant $\lambda$, which we are going to determine. The Lagrangian $E$ becomes:

$$E = \int_{-\infty}^{\infty} g(x) \ln g(x)\, dx + \lambda_1 \int_{-\infty}^{\infty} g(x)\, dx + \lambda_2 \int_{-\infty}^{\infty} x\, g(x)\, dx + \lambda_3 \int_{-\infty}^{\infty} x^2\, g(x)\, dx$$

The condition to be minimal for a certain $g(x)$ is given by the vanishing of the first variation (corresponding to the first derivative, but in this case with respect to a function) to $g(x)$: $\frac{\partial E}{\partial g} = 0$. This gives us: $-1 + \lambda_1 + x\,\lambda_2 + x^2\,\lambda_3 - \ln g(x) = 0$ from which we can easily solve $g(x)$: $g(x) = e^{-1+\lambda_1+x\,\lambda_2+x^2\,\lambda_3}$. So, $g(x)$ is beginning to get some shape: it is an exponential function with constant, linear and quadratic terms of $x$ in the exponent. Let us solve for the $\lambda$'s:

```
g[x_] := E^(-1+λ1+x λ2+x^2 λ3);
```

From the equation we see that at least $\lambda_3$ must be negative, otherwise the function explodes, which is physically unrealistic. We then need the explicit expressions for our constraints, so we make the following set of constraint equations, simplified with the condition of $\lambda_3 < 0$:

```
eqn1 = Simplify[∫_{-∞}^{∞} g[x] ⅆx == 1, λ3 < 0]
```

$$\frac{e^{-1+\lambda 1-\frac{\lambda 2^{2}}{4\,\lambda 3}}\,\sqrt{\pi}}{\sqrt{-\lambda 3}} == 1$$

```
eqn2 = Simplify[∫_{-∞}^{∞} x g[x] ⅆx == 0, λ3 < 0]
```

$$\frac{e^{-1+\lambda 1-\frac{\lambda 2^{2}}{4\,\lambda 3}}\,\sqrt{\pi}\,\lambda 2}{2\,(-\lambda 3)^{3/2}} == 0$$

```
eqn3 = Simplify[∫_{-∞}^{∞} x^2 g[x] ⅆx == σ^2, λ3 < 0]
```

$$\frac{e^{-1+\lambda 1-\frac{\lambda 2^{2}}{4\,\lambda 3}}\,\sqrt{\pi}\,(\lambda 2^{2}-2\,\lambda 3)}{4\,(-\lambda 3)^{5/2}} == \sigma^{2}$$

Now we can solve for all three $\lambda$'s:

```
solution = Solve[{eqn1, eqn2, eqn3}, {λ1, λ2, λ3}]
```

$$\left\{\left\{\lambda 1\to\frac{1}{4}\,\mathrm{Log}\left[\frac{e^{4}}{4\,\pi^{2}\,\sigma^{4}}\right],\ \lambda 2\to 0,\ \lambda 3\to-\frac{1}{2\,\sigma^{2}}\right\}\right\}$$

```
g[x_, σ_] = Simplify[E^(-1+λ1+x λ2+x^2 λ3) /. Flatten[solution], σ > 0]
```

$$\frac{e^{-\frac{x^{2}}{2\,\sigma^{2}}}}{\sqrt{2\,\pi}\,\sigma}$$

which is the Gaussian function. A beautiful result. Again, we have found the Gaussian as the *unique* solution to the set of constraints, which in principle are a formal statement of the *uncommittment* of the observation.

## 2.6 Derivatives of sampled, observed data

All *partial derivatives* of the Gaussian kernel are solutions too of the diffusion equation.

So the first important result is that we have found the Gaussian kernel and all of its partial derivatives as the *unique* set of kernels for a front-end visual system that satisfies the constraints: no preference for location, scale and orientation, and linearity. We have found a one-parameter *family* of kernels, where the scale $\sigma$ is the free parameter.

Here are the plots of some members of the Gaussian derivative family:

```
g := 1/(2 π σ^2) Exp[-(x^2 + y^2)/(2 σ^2)]; σ = 1;

Block[{$DisplayFunction = Identity},
  graphs = Plot3D[Evaluate[#], {x, -3.5, 3.5}, {y, -3.5, 3.5}] & /@
    {g, ∂x g, ∂x ∂y g, ∂x,x g + ∂y,y g}];
```

```
Show[GraphicsArray[graphs], ImageSize -> 400];
```

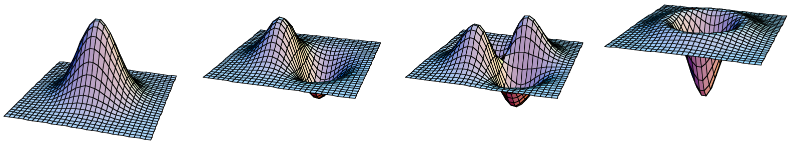


Figure 2.10 Upper left: the Gaussian kernel G(x,y;$\sigma$) as the zeroth order point operator; uppe right: $\frac{\partial G}{\partial x}$ ; lower left: $\frac{\partial^2 G}{\partial x\, \partial y}$ ; lower right: the Laplacian $\frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$ of the Gaussian kernel.

Because of their importance, we will discuss properties of the Gaussian kernel and its derivatives in detail in the next chapters. In chapters 6 and 7 we will see how sensitivity profiles of cells in the retina closely resemble the Laplacian of the Gaussian, and in the primary visual cortex they closely resemble Gaussian derivatives, as was first noticed by Young [Young1985, Young1986, Young1986b, Young1987a] and Koenderink [Koenderink1984a].

The derivative of the observed data $L_0(x, y) \otimes G(x, y; \sigma)$ (the convolution is the observation) is given by $\frac{\partial}{\partial x} \{L_0(x, y) \otimes G(x, y; \sigma)\}$, which can be rewritten as $L_0(x, y) \otimes \frac{\partial}{\partial x} G(x, y; \sigma)$. Note that we cannot apply the chainrule of differentiation here: the operator between $L_0(x, y)$ and $G(x, y; \sigma)$ is a convolution, not a product. The *commutation* (exchange) of the convolution operator and the differential operator is possible because of their linearity. It is best appreciated when we consider the equation $\frac{\partial}{\partial x} \{L_0(x, y) \otimes G(x, y; \sigma)\}$ in the Fourier domain. We need the two rules:

- The Fourier transform of the derivative of a function is equal to $-i\,\omega$ times the Fourier transform of the function, where $i \equiv \sqrt{-1}$ , and
- convolution in the spatial domain is a product in the Fourier domain:

```
Clear[f]; FourierTransform[f[x], x, ω]

FourierTransform[f[x], x, ω]

FourierTransform[D[f[x], x], x, ω]

-ⅈ ω FourierTransform[f[x], x, ω]
```

So we get ($\hat{L}$ denotes the Fourier transform of $L$): $\frac{\partial}{\partial x} \{L_0(x, y) \otimes G(x, y; \sigma)\} \xrightarrow{\mathcal{F}} -i\,\omega \left\{\hat{L}\,.\,\hat{G}\right\} = \hat{L}\,.\left\{-i\,\omega\,\hat{G}\right\} \xrightarrow{\mathcal{F}^{-1}} L_0(x, y) \otimes \frac{\partial}{\partial x} G(x, y; \sigma)$

The commutation of the convolution and the derivative operators , which is easily shown in the Fourier domain. From this we can see the following important results:

- Differentiation and observation are done in a single step: convolution with a Gaussian derivative kernel.

- Differentiation is now done by *integration*, namely by the convolution integral.

This is a key result in scale-space theory. We can now apply differentiation (even to high order) to *sampled data* like images.

We just convolve the image with the appropriate Gaussian derivative kernel. But where do we need the derivatives, and where do we need higher order derivatives?

An important area of application is the exploitation of *geometric* information from images. The most basic example is the first order derivative, which gives us *edges*.

Edges are defined as a sudden change of intensity $L$ when we walk over the image and this is exactly what a derivative captures: $\frac{\partial L}{\partial x}$.

Derivatives abound in the detection of *differential features* (features expressed as some (polynomial) expression in image derivatives). They also show up with the detection of motion, of stereo disparity to find the depth, the detection of structure in color images, segmentation, image enhancement and denoising, and many other application areas as we will see in the rest of the book.

Some more implications of the theory so far:

- The Gaussian kernel is the *physical* analogue of a *mathematical* point, the Gaussian derivative kernels are the physical analogons of the mathematical differential operators. Equivalence is reached for the limit when the scale of the Gaussian goes to zero:

- $\lim_{\sigma \to 0} G(x; \sigma) = \delta(x)$, where $\delta(x)$ is the Dirac delta function, and $\lim_{\sigma \to 0} \left\{ f(x) \otimes \frac{\partial G(x;\sigma)}{\partial x} \right\} = \lim_{\sigma \to 0} \int_{-\infty}^{\infty} f(\alpha)\ \partial_x G(x - \alpha; \sigma)\, d\alpha =$ $\int_{-\infty}^{\infty} f(\alpha)\ \delta(x - \alpha)\, d\alpha = \partial_x f(x)$.

```
∫_{-∞}^{∞} f[α] D[DiracDelta[α - x], x] ⅆα

f′[x]
```

- There is an intrinsic and unavoidable relation between differentiation and blurring. By its definition, *any* differentiation on discrete (observed) data blurs the data somewhat, with the amount of the scale of the differential operator. There is no way out, this increase of the inner scale is a physical necessity. We can only try to minimize the effect by choosing small scales for the differentiation operator. However, this minimal scale is subject to constraints (as is the maximal scale). In chapter 7 we develop the fundamental relation between the scale of the operator, its differential order and the required amount of accuracy.

The *Mathematica* function **gD[im,nx,ny,σ]** implements a convolution with a Gaussian derivative on the image **im**, with order of differentiation **nx** with respect to $x$ resp. **ny** with respect to $y$. Figure 2.12 shows the derivative to $x$ and $y$ of a simple test image of a square:

```
im = Table[If[80 < x < 170 && 80 < y < 170, 1, 0], {y, 1, 256}, {x, 1, 256}];
Block[{$DisplayFunction = Identity},
  imx = gD[im, 1, 0, 1]; imy = gD[im, 0, 1, 1]; grad = √(imx^2 + imy^2) ;
  p1 = ListDensityPlot /@ {im, imx, imy, grad}];
Show[GraphicsArray[p1], ImageSize → 400];
```

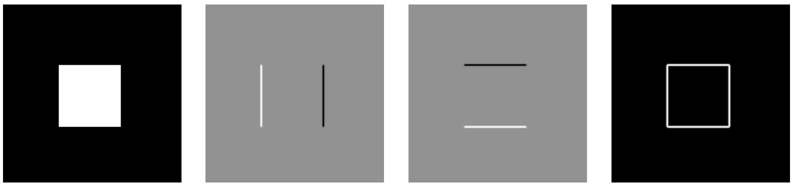


Figure 2.11 The first order derivative of an image gives edges. Left: original test image $L(x, y)$, resolution $256^2$. Second: the derivative with respect to $x$: $\frac{\partial L}{\partial x}$ at scale $\sigma = 1$ pixel. Note the positive and negative edges. Third: the derivative with respect to $y$: $\frac{\partial L}{\partial y}$ at scale $\sigma = 1$ pixel. Right: the gradient $\sqrt{(\frac{\partial L}{\partial x})^2 + (\frac{\partial L}{\partial x})^2}$ at a scale of $\sigma = 1$ pixel which gives all edges.

- The Gaussian kernel is the unique kernel that generates no *spurious resolution*. It is the blown-up physical *point operator*, the Gaussian derivatives are the blown-up physical *multi-scale derivative operators*.

```
Show[Import["blown-up ddx.jpg"], ImageSize -> 300];
```

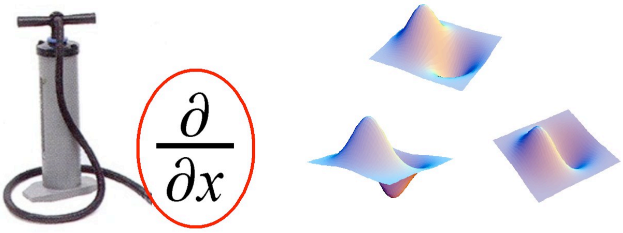


Figure 2.12 Convolution with a Gaussian derivative is the blown-up version of convolution with the Delta Dirac function. Taking the limit of the scale to zero ('letting the air out") leads to the 'regular' mathematical formulation.

- Because convolving is an integration, the Gaussian kernel has by definition a strong *regularizing* effect. It was shown by Schwartz [Schwartz1951] that differentiation of *distributions* of data ('wild' data, such as discontinuous or sampled data) has to be accomplished by convolution with a smooth testfunction. This smooth testfunction is our Gaussian kernel here. So, we recognize that the process of observation *is* the regularizer. So there is no need to smooth the data first. Actually, one should never change the input data, but only make modifications to the process of observation where one has access: the filter through which the measurement is done. The visual system does the same: it employs filters at many sizes and shapes, as we will see in the chapter on human vision.

- Recently some interesting papers have shown the complete equivalence of Gaussian scale space regularization with a number of other methods for regularization such as splines, thin plate splines, graduated convexity etc. [Scherzer2000a, Nielsen1996b, Nielsen1997b]. In chapter 10 we will discuss the aspects of differentiation of discrete data (it is 'ill-posed') and the property of regularization in detail.

- The set of Gaussian derivative kernels (including the zeroth order derivative: the Gaussian kernel itself) forms a *complete* set of derivatives. This set is sometimes referred to as the *N-jet*.

Now the basic toolkit is there to do differential geometry, tensor analysis, invariant theory, topology and apply many more mathematical tools on our discrete data. This will be the topic of much of the rest of this book.

## 2.7 Scale-space stack

A scale-space is a stack of 2D images, where scale is the third dimension. One can make a scale-space of any measurement, so one can measure an intensity scale-space, a gradient magnitude scale-space, a Laplacian scale-space etc.

```
im = Import["mr64.gif"][[1, 1]];
Block[{$DisplayFunction = Identity, xres, yres, max},
 {yres, xres} = Dimensions[im]; max = Max[im];
 gr = Graphics3D[ListPlot3D[Table[0, {yres}, {xres}],
    Map[GrayLevel, im / max, {2}], Mesh → False, Boxed → False]];
gb = Table[blur = gD[im, 0, 0, i]; Graphics3D[ListPlot3D[
     Table[i 10, {yres}, {xres}], Map[GrayLevel, blur / max, {2}],
     Mesh → False, Boxed → False]], {i, 1, 6}]];
Show[{gr, gb}, BoxRatios → {1, 1, 1}, ViewPoint -> {1.190, -3.209, 1.234},
  DisplayFunction -> $DisplayFunction, Boxed -> True, ImageSize -> 240];
```

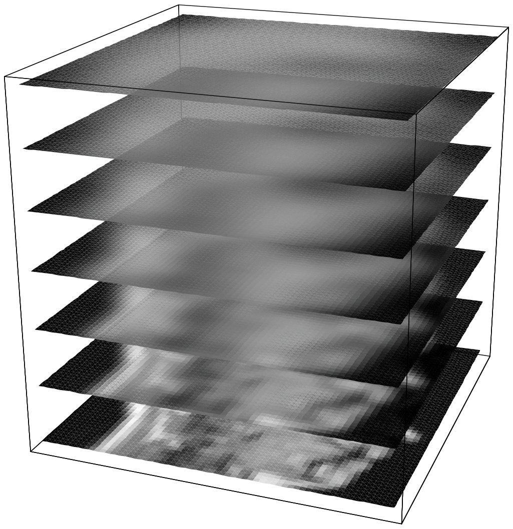


Figure 2.13 A scale-space of a 2D MRI sagittal slice, dimensions $64^2$, for a range of scales $\sigma$ =1,2,3,4,5,and 6 pixels.

We found a family of kernels, with the scale $\sigma$ as a free parameter. When we don't know what scale to apply in an uncommitted measurement, we just take them all. It is like sampling at spatial locations: we put CCD elements all over our receptor's sensitive area. We will see that the visual system does just that: it has *groups* of rods and cones in the retina (termed receptive fields) of a wide range of circular diameters, effectively sampling at many different scales.

We will see in the chapters on the 'deep structure' of images (i.e. the structure along the scale axis), that in the scale-space the *hierarchical, topological* structure of images is embedded. See chapters 13-15.

One can make scale-spaces of any dimension. A scale-space stack of 3D images, such as 3D datasets from medical tomographic scanners, is a 4D space (x,y,z;$\sigma$) and is termed a *hyperstack* [Vincken 1990].

And here are two scale-spaces of a real image, a scale-space of the intensity (no derivatives, only blurred) and a scale-space of the Laplacian (the Laplacian is the sum of the second order derivatives of the image, $\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}$ ).

```
im = Import["mr128.gif"][[1, 1]];
DisplayTogetherArray[
  {Table[ListDensityPlot[gD[im, 0, 0, E^τ]], {τ, 0, 2.1, .3}],
   Table[ListDensityPlot[gD[im, 2, 0, E^τ] + gD[im, 0, 2, E^τ]],
    {τ, 0, 2.1, .3}]}, ImageSize -> 390];
```

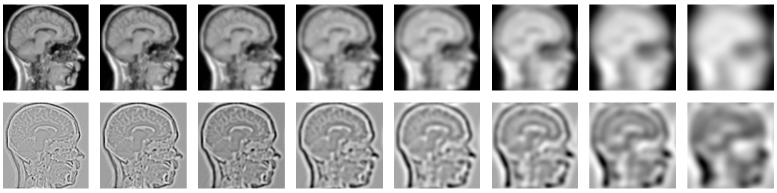


Figure 2.14 A scale-space is a stack of images at a range of scales. Top row: Gaussian blu scale-space of a sagittal Magnetic Resonance image, resolution $128^2$, exponential scale range from $\sigma = e^0$ to $\sigma = e^{2.1}$. Bottom row: Laplacian scale-space of the same image, same scale range.
The function `gD[im,nx,ny,σ]` will be explained later (chapter 4 and 5). It convolves the image with a Gaussian derivative.

## 2.8 Sampling the scale-axis

From the example from the trip through scale in the "Powers of 10" series we made steps of a *factor* 10 each time we took a new picture. This is an exponential stepping through scale, and we know this as experimental fact. We step in 'orders of magnitude'. The scale parameter $\sigma$ gives a logical length parameter for the level of resolution.

If we consider how to parametrize scale $\sigma$ with a dimensionless parameter $\tau$, then we realize that *scale-invariance* (or *self-similarity*) must imply that $d\sigma / d\tau$ must be proportional to $\sigma$.

In other words, the change that we see when we step along the scale axis, is proportional to the level of resolution at hand. Without loss of generality we may take $\frac{d\sigma}{d\tau} = \sigma$ with $\sigma|_{\tau=0} = \epsilon$. We call the dimensionless parameter $\tau$ the *natural scale parameter*: $\sigma = \in e^{\tau}$

where $\tau$ can be any number, even negative. Note that the artificial singularity due to the problematic value of $\sigma = 0$ is now no longer present.

There is a difference between 'zooming' and 'blurring':

zooming is the reparametrization of the spatial axis, $\tilde{x} \mapsto a\,x$, so we get a larger or smaller image by just setting them further apart of farther away. There is no information gained or lost. Blurring is doing an observation with a larger aperture: the image is blurred. Now information *is* lost, and this is exactly what is a requirement for a scale-space: reduction of information. Because we have a larger $\sigma$ over the same image domain, we can effectively perform a *sampling rate reduction* [Vincken1990].

How much information is lost when we increase scale? Florack [Florack1994b] introduced the following reasoning:

The number of (equidistant) samples on a given domain, given a fixed amount of overlap between neighboring apertures, on scale level $\sigma$ relative to the number of samples at another scale level $\sigma_0$ is given by $\frac{N(\sigma)}{N(\sigma_0)} = (\frac{\sigma_0}{\sigma})^D$, where $D$ is the dimension.
Or, in terms of the natural scale parameter $\tau$ with $\sigma = \epsilon\, e^\tau$:

$$N(\sigma) = N(\sigma_0)\,(\frac{\epsilon e^{\tau_0}}{\epsilon e^{\tau}})^D = N(\sigma_0)\,e^{D(\tau_0 - \tau)}$$

which is the solution of the differential equation $\frac{dN}{d\tau} + D\,N = 0$. At the highest scale, we have just a single wide aperture left and we achieved total blurring; the image domain has become a single point. Notice that the sampling rate reduction depends on the dimension $D$. When we consider natural, generic images, we expect the information in the images to exist on all scales. We could think of a 'density of local generic features' such as intensity maxima, minima, saddle points, corners etc. as relatively homogeneously distributed over the images over all scales when we consider enough images. This 'feature density' $N_F\,(\tau)$ might then be related to the number of samples $N(\tau)$, so $\frac{dN_F}{d\tau} + D\,N_F = 0$. In chapter 20 we will see that the number of extrema and saddle points in a scale-space of generic images indeed decreases with a slope of $\frac{d \ln N_F}{d\tau} \approx -2$ for 2D images and a slope of -1 for 1D signals.

The factor $\in$ in the equation for natural scale appears for dimensional reasons: it is the scale for $\tau = 0$, and is a property of our imaging device; it is the pixelsize, CCD element size, the sampling width etc.: the *inner scale* of the measurement.

```
Block[{$DisplayFunction = Identity},
  p1 = Graphics[
    Table[Circle[{x, y}, .6], {x, 1, 10}, {y, 1, 10}], AspectRatio -> 1];
p2 = Graphics[Table[Circle[{x, y}, 1.2], {x, 1, 10, 2}, {y, 1, 10, 2}],
    AspectRatio -> 1];
p3 = Graphics3D[Table[{EdgeForm[], TranslateShape[Sphere[.6, 10, 10],
       {x, y, z}]}, {x, 1, 6}, {y, 1, 6}, {z, 1, 6}], Boxed -> False];
p4 = Graphics3D[Table[{EdgeForm[], TranslateShape[
       Sphere[1.2, 10, 10], {x, y, z}]}, {x, 1, 6, 2},
     {y, 1, 6, 2}, {z, 1, 6, 2}], Boxed -> False]];
Show[GraphicsArray[{p1, p2, p3, p4}], ImageSize -> 400];
```

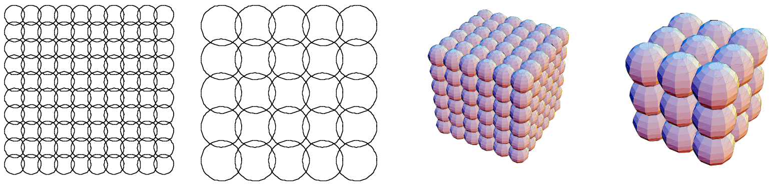


Figure 2.15 The number of samples on a 2D domain, given a fixed amount of overlap decreases with $(\frac{\sigma_0}{\sigma})^2$ (left two figures), on a 3D domain with $(\frac{\sigma_0}{\sigma})^3$ (right two figures). So the number of samples decreases as a function of scale with a slope of $-D$, where $D$ is the dimension (see text). The sampling rate reduction is dependent on the dimensionality of the measurement.

For positive $\tau$ we go to larger scale, for negative $\tau$ we go to smaller scale. In the expression for the natural scale the singularity at $\sigma = 0$ is effectively removed.

The exponential stepping over the scale axis is also evident in the *Hausdorff dimension*, the number of boxes counted in a quadtree of a binary image (see also [Pedersen2000] and chapter 15, section 15.1.4).

Of course, there is no information within the inner scale, so here problems are to be expected when we try to extract information at sub-pixel scale. Only by taking into account a *context* of voxels through a proper model, we can go to the subpixel domain.

This is an important notion: any observation at a single point is an independent measurement, and we can do a lot of measurements there.

In the next few chapters we will derive many features related to the measurement of derivatives at our pixel. It turns out that we can make lots of specific polynomial combinations, like edge strength, 'cornerness' etc. but they all describe information in that point. It is a 'keyhole observation'. The important 'perceptual grouping' of neighboring points into meaningful sets is accomplished by specifying constraints, like *models*. In this book we first derive many local (differential) features.

In the second part we go a little further in the cascade of visual processing steps, and investigate local neighborhood relations through comparison of local properties like orientation, strength of derivative measurements etc. We also explore the *deep structure* of images (a term first coined by Koenderink), by which we mean the relations over scale. In the deep structure we may expect the hierarchical, structuring, more topological information: what is 'embedded in' what, what is 'surrounded by' what, what is 'part of' what etc. This takes us to a next level of description in images, which is currently receiving a lot of attention.

Fractals are famous examples of self similar functions. This self-similar fractal shows a tree in three dimensions [Cabrera, www.mathsource.com]. Parameters: $\alpha$ = branch angle; $\epsilon$ = scale factor; m = number of branches from previous branch; n = deepness.

## 2.9 Summary of this chapter

Scale-space theory was discovered independently by Iijima in Japan in the early sixties, and by Koenderink in Europe in the early seventies.

Because we have specific physical constraints for the early vision front-end kernel, we are able to set up a 'first principle' framework from which the exact sensitivity function of the measurement aperture can be derived. There exist many such derivations for an uncommitted kernel, all leading to the same unique result: the Gaussian kernel. We discussed two approaches: the first started with the assumptions of linearity, isotropy, homogeneity and scale-invariance.

With the help of the Pi-theorem from dimensional analysis one is able to derive the Gaussian by plugging in the constraints one by one.

The second derivation started from causality: it is impossible that maxima increase and minima decrease with increasing scale, every blurred version is the causal consequence of the image it was blurred from. This means that the extrema must be closed from above. This leads to s constraint on the sign of the second derivative, from which the diffusion equation emerges.

The third derivation started from the minimization of the entropy at the very first measurement. Through the use of Lagrange multipliers, where the constraints are used one by one, one can again derive the Gaussian kernel as the unique kernel for the front-end.

A crucial result is that differentiation of discrete data is done by the convolution with the derivative of the observation kernel, in other words: by an integration. Differentiation is now possible on discrete data by means of convolution with a finite kernel. In chapter 14 we discuss this important mathematical notion, which is known as *regularization*.

This means that differentiation can never be done without blurring the data somewhat. We find as a *complete* family of front-end kernels the family of all partial derivatives of the Gaussian kernel. The zeroth order derivative is just the Gaussian blurkernel itself.

Scale is parametrized in an exponential fashion (we consider 'orders of magnitude' when scaling). The exponent in this parametrization is called the natural scale parameter.

```
Rotz[t_] = {{Cos[t], Sin[t], 0}, {-Sin[t], Cos[t], 0}, {0, 0, 1}};
Roty[t_]  = {{Cos[t], 0, -Sin[t]}, {0, 1, 0}, {Sin[t], 0, Cos[t]}};
Rot3D[ψ_, θ_] = Roty[θ].Rotz[ψ]; SphericalCoordinates[{x_, y_, z_}] =
 { Sqrt[x^2 + y^2 + z^2], ArcTan[z, Sqrt[x^2 + z^2]] , ArcTan[x, y]};
NextBranches[α_, ϵ_, m_][ Branch[r1_List, r0_List, th_] ] :=
    Module[{r, ψ, θ}, {r, θ, ψ} = SphericalCoordinates[r1 - r0];
        {Branch[ϵ * (r1 - r0) + r1, r1, ϵ * th], Sequence @@ Table[
Branch[r1 + ϵ * r {Sin[α] Cos[φ], Sin[α] Sin[φ], Cos[α]}.Rot3D[ψ, θ],
                                 2 Pi
        r1, ϵ * th], {φ, 0, 2 Pi, ────}]} // N];
                                  m
NextBranches[α_, ϵ_, m_][w_List] := Map[NextBranches[α, ϵ, m], w];
Tree2D[α_, ϵ_, m_, r_List, th_, n_] :=
NestList[NextBranches[α, ϵ, m], Branch[r, {0, 0, 0}, 1] , n] /.
  Branch[r1_, r0_, t_] :>
   {RGBColor[0, 0.6 (1 - t) + 0.4, 0], Thickness[th * t], Line[{r1, r0}]}
Show[Graphics3D[Tree2D[α, ϵ, m, r, th0, n] /.
    {α -> Pi / 8, ϵ -> 0.6, m -> 5, r -> {0.01, 0, 1}, n -> 4, th0 -> 0.03}],
  PlotRange -> {{-1, 1}, {-1, 1}, {0, 2.5}},
  ViewPoint -> {3.369, -0.040, 0.312}, ImageSize -> 200];
```

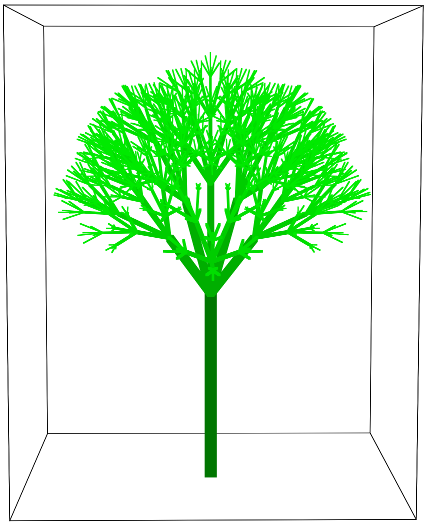


Figure 2.16 Fractals are famous examples of self similar functions. This self-similar fractal shows a tree in three dimensions [Cabrera, www.mathsource.com]. Parameters: $\alpha$ = branch angle; $\epsilon$ = scale factor; m = number of branches from previous branch; n = deepness. Source: Renan Cabrera, www.mathsource.com.