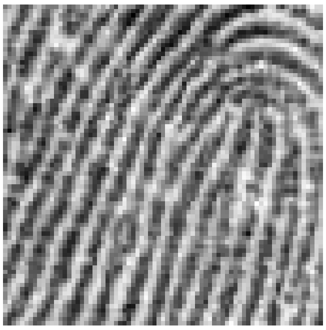
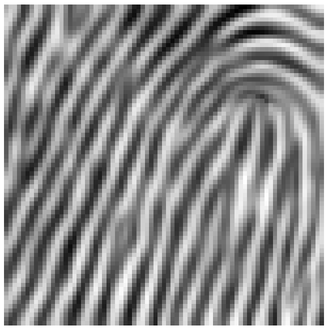
Beyond Linear Scale-Space
Part I:

Geometry Driven Diffusion

Original Image



Coherence Enhancing Diffusion



Rein van den Boomgaard
University of Amsterdam

1 Introduction

2 Linear Scale-Space Theory

Scale-dependent observations. All physical measurements are done with a 'probe' of finite spatial-temporal size. If no a priori preference for a specific scale can be made the obvious choice is to look at all scales: a *scale-space*. Scale dependent observations are thus turned in an operational device (both by human vision and computer vision).

The local structure of images. The collection of all spatial image derivatives up to order N (known as the N-jet) provides a local characterization of the image 'landscape'. Differential geometry is the mathematical tool that leads to *invariant* descriptors of local image structure.

Causality in scale-space. *Causality* in scale-space is determined by the *diffusion equation* describing in what way a small change of observation scale may influence the observed value.

2.1 Scale dependent observations

A linear density probe is best modeled with a Gaussian convolution:

$$\begin{aligned} L(x,s) &= (L_0 * G^s)(x) \\ &= \int_{\mathbb{R}^d} L_0(x-y) G^s(y) dy \end{aligned}$$

The Gaussian probe defines the *physical* notion of a point.

2.2 The local structure of images

Locally an image is characterized with its Taylor expansion. The second order Taylor expansion around the point x is:

$$L(x+y,s) = L(y,s) + x^{\mathsf{T}}(\nabla L)(y,s) + \frac{1}{2}x^{\mathsf{T}}H_L(y,s)x + \cdots$$

where ∇L is the *gradient* of the image function:

$$(\nabla L)(x,s) = \begin{pmatrix} (\partial_x L)(x,s) \\ (\partial_y L)(x,s) \end{pmatrix} = \begin{pmatrix} L_x(x,s) \\ L_y(x,s) \end{pmatrix}$$

and H_L is the Hessian matrix:

$$(H_L)(x,s) = \begin{pmatrix} (\partial_{xx} L)(x,s) & (\partial_{xy} L)(x,s) \\ (\partial_{xy} L)(x,s) & (\partial_{yy} L)(x,s) \end{pmatrix}$$

Notation We will often write:

$$\nabla L = \begin{pmatrix} L_x \\ L_y \end{pmatrix}$$

and

$$H_L = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix}.$$

This is 'overloading' of notational conventions as we construct a 'vector' of images and a 'matrix' of images. This notation should be read as the gradient vector in every location and as the Hessian matrix in every location respectively.

With this short hand notation we can write the 2nd order Taylor expansion of the image L as:

$$L(x+y,s) = L + x^\top \nabla L + \frac{1}{2} x^\top H_L x$$

with the convention that on the right hand side the image (and its derivatives) are evaluated at the location y at scale s.

2-Jet The image derivatives up to order 2 are collectively known as the *2-jet*:

$$\begin{array}{ccccc} & & L & & \\ & L_x & & L_y & \\ L_{xx} & & L_{xy} & & L_{yy} \end{array}$$

The individual components of the 2-Jet have no geometrical meaning, just like the elements of a vector do not have a meaning. It is the collection that does have a geometrical meaning.

In sect:gauges we introduce the *gauge coordinates* that *do* have a geometrical meaning.

2.3 Gaussian Derivatives

In *computational vision* we replace the mathematical derivatives by *fuzzy derivatives* or *Gaussian Derivatives*[1]:

$$\partial L \longrightarrow \partial^s L = \partial(L * G^s) = L * \partial G^s$$

Local image observations are therefore always done at a finite scale to be selected by the observer (programmer).

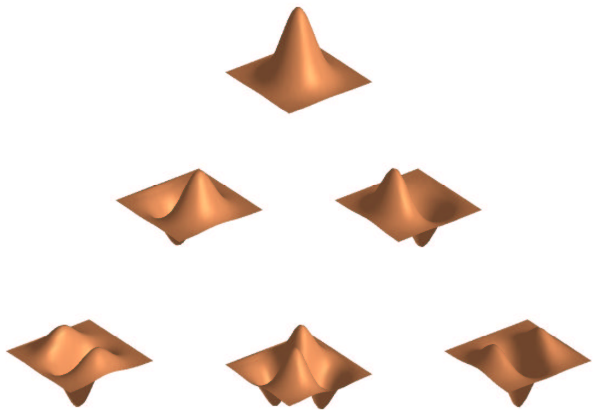


Figure 1: **Gaussian Derivative Kernels.** From top-to-bottom, left-to-right: G^s, G^s_x, G^s_y, G^s_{xx}, G^s_{xy} and G^s_{yy}.

[1] A notable exception is when we construct numerical schemes to solve partial differential image equations; like the non-linear diffusion equations that are the subject of this report

2.4 Gauge coordinates

- The classical Euclidean image coordinate frame is an arbitrary convenient choice without much (local) relevance[2].
- Attaching the coordinate frame to the local image structure makes the coordinate frame independent of arbitrary Euclidean transformations (rotations, scalings and translations).
- The most important *gauge* coordinate systems are:

  Gradient gauge (v, w)**.** One of the coordinate axes (w) is aligned with the image gradient vector.

  Curvature gauge (p, q)**.** The coordinate axes are aligned are aligned with the direction of minimal and maximal directional second order derivative.

[2]On a global scale the classical choice *is* important as the notion of *up/down*, *above/below* has a special relevance in the analysis of most of the images made of the 3D world.

2.5 Gradient Gauge

Within the gradient gauge coordinate system (v, w) several important local geometrical image properties are easily described:

Edge strength L_w. The derivative in the direction of the gradient is always positive and is an indicator of the edge strength.

Edge location $L_{ww} \approx 0$. Where the gradient strength is maximal (in the gradient direction) the edge can be localized.

Isophote curvature $\kappa \propto -L_{vv}$. The isophote curvature is high in the vicinity of sharp corners.

2.6 Edge Detection

The Marr edge detector and the Canny edge detector are perhaps the most famous examples of edge detectors that nicely fit within the local geometrical image model (in the gradient gauge description).

Marr edge detector. The zero crossings in the image Laplacian $\nabla^2 L = L_{xx} + L_{yy} = L_{vv} + L_{ww}$ are taken as edge indicators. This is a straightforward generalization of where in the 1D case we have maximal edge strength.

Canny edge detector. This edge detector differs from the Marr detector in that it looks for the zero crossings in the second order derivative in the direction of the gradient $L_{ww} \approx 0$.

Note that the Canny and Marr detectors are equivalent in cases where the isophote curvature is low (then $L_{vv} + L_{ww} \approx L_{ww}$).

2.7 Curvature Gauge

The curvature gauge is important in those cases where the gradient vector vanishes (and thus where the gradient gauge is ill defined). E.g. the midpoints of line like structures in images.

The eigenvectors of the Hessian matrix H_L then provide a powerful way of defining a gauge system.

3 Causality in Scale-Space

The solution of the diffusion equation:

$$\partial_t L = \nabla^2 L$$

with initial condition $L(x, 0) = L_0$ is given by:

$$L(x,t) = (L_0 * G^{\sqrt{2t}})(x),$$

where G^s is the Gaussian function[3] at scale s:

$$G^s(x) = \frac{1}{(s\sqrt{2\pi})^d} e^{-\frac{\|x\|^2}{2s^2}}.$$

This is the classical linear scale-space 'generated' by the Gaussian convolution.

[3] d is the dimension of the space

Koenderink's Causality Criterium. Luminance values should not be created when increasing the observation scale, i.e. a value $L_0 = L(x,t)$ should be traceable to a location in the infinitesimal neighborhood of x at infinitesimally smaller scale.

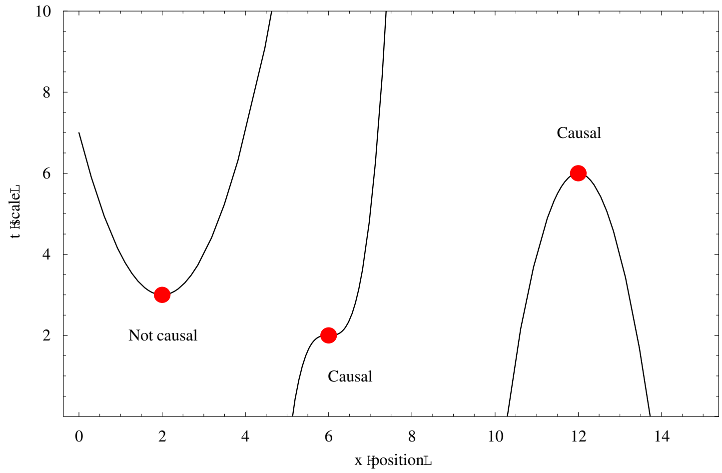


Figure 2: **Causality in scale-space.** Three isophotes is scale-space are drawn. The left isophote corresponds with a non causal trace in scale-space, whereas the other two isophotes are causal.

Critical points. In the neighborhood of a *critical point* (x_0, t_0) (where the scale-space isophote is 'horizontal' (see Fig. **??**)the isophote is a Monge patch that can be encoded with a function $t = T(x)$. For the isophote we thus have:

$$L(x, T(x)) = \text{constant}. \tag{1}$$

For a critical point we have $T'(x_0) = 0$. For such a point to be causal we must have that $T''(x_0) < 0$, i.e. the isophote must lie (partly) *below* the tangent plane.

The diffusion equation. In order to find a relation for T'' we differentiate both sides of Eq. 1 twice with respect to x. This results in:

$$L_{xx} + 2L_{xt}T' + L_{tt}(T')^2 + L_tT'' = 0,$$

where we have omitted the arguments of the functions. In a critical point x_0 we have: $T'(x_0) = 0$. This leads to:

$$L_{xx} + L_tT'' = 0$$

or equivalently:

$$T'' = -\frac{L_{xx}}{L_t}.$$

Causality requires that in a critical point $T'' < 0$, thus

$$L_t = \alpha^2 L_{xx}.$$

The simplest choice is the constant function $\alpha = 1$ leading to the requirement that $L_t = L_{xx}$ *in all critical points in scale-space.* Evidently in case we require $L_t = L_{xx}$ for *all* points in scale-space, the causality requirement is obviously met with. Finally we thus arrive at:

$$L_t = L_{xx}$$

the classical linear diffusion equation.

The diffusion equation. The analysis generalized quite easily to higher dimensional spaces. Then we can show that causality is guaranteed in case:

$$L_t = \nabla^2 L$$

where ∇^2 is the *Laplacian* operator.

Remarks. The 'proof' of the Gaussian linear scale-space as the unique construction of a causal scale-space is misleading. Implicitly we have assumed that the isophotes in scale-space are smooth differentiable surfaces. *Causality does not require them to be differentiable.*

It can be shown that even a morphological operator (erosions and dilations using a parabolic kernel) results in a causal scale-space.

4 Numerical Solutions I

We can smooth (diffuse) an image by solving the PDE:

$$L_t = \nabla^2 L$$

with initial condition $L(x, 0) = L_0$. For infinitesimal dt we can write:

$$L(x, t + dt) = L(x, t) + dt(\nabla^2 L)(x, t).$$

For a sampled image $L^t_{i,j} = L(i\Delta x, j\Delta y, t)$ we have:

$$L^{t+dt}_{i,j} = L^t_{i,j} + dt(\nabla^2 L)^t_{i,j} \tag{2}$$

where now $(\nabla^2 L)^t_{i,j}$ is a discrete approximation of the Laplacian given the sampled image $L^t_{i,j}$.

Stability and Accuracy. Eq. 2 can be used to generate the scale-space starting with the initial image L_0 and successively adding the discrete Laplacian.
The *stepsize* dt should be chosen quite small:

Stability. Too large a stepsize results in complete nonsense (even numerical instability),

Accuracy. Too large a stepsize (but still leading to a stable solution) may lead to inaccurate solutions. I.e. a poor discrete approximation of the image L at scale t.

Example. The simplest numerical discretization of the Laplacian operator is:

$$\begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix}$$

(where we give the convolution kernel of a linear translation invariant operator).

Adding dt times the Laplacian to the image results in the following linear operator:

$$\begin{pmatrix} & dt & \\ dt & 1-4dt & dt \\ & dt & \end{pmatrix}$$

Note that for $dt > 0.25$ the central weight in the above kernel becomes negative. This leads to an unstable solution scheme. It is beyond the scope of this report to show that indeed $dt < 0.25$ leads to stable numerical schemes.

Example (Stability).

Figure 3: **Numerical stability.** On the left the original image, in the middle the linear diffusion equation numerically solved with $dt = 0.23$ and on the right $dt = 0.26$.

So indeed a 'stable' stepsize should be chosen, but what about accuracy...

Example (Accuracy).

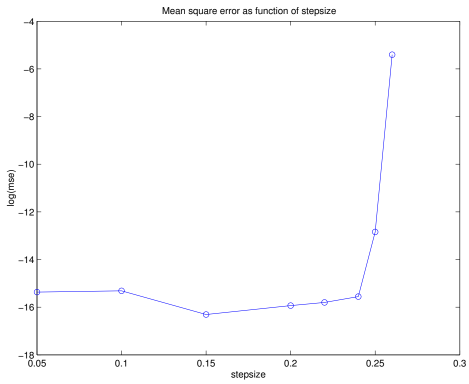


Figure 4: **Numerical accuracy.** The (log) mean square error is plotted as a function of the stepsize. We let the diffusion run until 'time' $t \approx 5$ for different stepsizes. The correct final image can be calculated using a simple Gaussian convolution. This allows us to compare the numerical result with the true result.

5 The Physics of Diffusion

The notion of *diffusion* comes from physics. Diffusion is the physical process where heat (mass) redistributes itself in a medium as a consequence of heat (mass) concentration differences.

As an example consider a totally isolated thin metal plate with an uneven temperature distribution at time $t = 0$. If at time $t = 0$ no more heat is added to the plate (and we assume there is no heat loss to the air) then the heat will redistribute itself until there are no temperature differences.

There are two physical principles involved:

Heat flow. Due to concentration differences heat flows in the opposite direction of the concentration gradient.

Conservation law. If no heat is added from outside the system and no heat can flow to outside of the system the total amount of heat is preserved.

5.1 Heat flow

Heat *flows* in the direction where the heat is locally minimal, i.e. it flows in the opposite direction of the temperature *gradient*:

$$j = -D\nabla L$$

(we take L to be the temperature distribution, as a function of position and time).

In general D is a *tensor*. The flow need not be parallel to the gradient. Material properties (in which the heat is flowing) may direct the flow along preferential directions (think of laminated materials). The tensor D may also depend on the spatial position.

The *diffusion tensor* D can orient the flow along such preferential directions.

5.2 Conservation law

Consider an infinitesimal small surface element dS in the space where the heat is flowing. Let $dS = dAn$ where dA is the area of the surface element and n the surface normal.

The heat flowing through this surface element per unit time interval is:

$$J_{dS} = j \cdot dS = (j \cdot n)dA$$

Now consider a closed surface A around point x (say a sphere), then

6 Scalar Diffusion

The generic diffusion equation:

$$\partial_t L = \nabla \cdot (D \nabla L)$$

is simplified by selecting $D = cI$ where the conductivity c is a scalar *possibly spatially varying*.

The basic idea is:

> select a large conductivity in image areas where we want a lot of smoothing and a small conductivity where we don't want smoothing (e.g. near edges).

7 Numerical Schemes II

8 Tensor Diffusion

8.1 Edge Enhancing Diffusion

8.2 Coherence Enhancing Diffusion

9 Numerical Schemes III

10 Generalizations & Improvements