Geometry-driven diffusion:
*nonlinear scale-space* – *adaptive scale-space*
The advantage of selecting a larger scale is

• the improved reduction of noise,
• the appearance of more prominent structure,

but the price to pay for this is reduced localization accuracy.

Linear, isotropic diffusion cannot preserve the position of the differential invariant features over scale.

A solution is to make the diffusion, i.e. the amount of blurring, locally adaptive to the structure of the image.
1. **Nonlinear partial differential equations** (PDE's), i.e. nonlinear diffusion equations which evolve the luminance function as some function of a flow;

2. **Curve evolution of the isophotes** (curves in 2D, surfaces in 3D) in the image;

3. **Variational methods that minimize some energy functional** on the image.

It takes geometric reasoning to come up with the right nonlinearity for the task, to include knowledge.

We call this field evolutionary computing of image structure, or the application of evolutionary operations.
A conductivity coefficient \( c \) is introduced in the diffusion equation:

\[
\frac{\partial L}{\partial s} = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}
\]

\[
\frac{\partial L}{\partial s} = \nabla . c \nabla L
\]

\[
c = c(L, \frac{\partial L}{\partial x}, \frac{\partial^2 L}{\partial x^2}, \ldots)
\]

It is a divergence of a flow. We also call \( c \nabla L \) the flux function. With \( c = 1 \) we have normal linear, isotropic diffusion: the divergence of the gradient flow is the Laplacian.
- **linear diffusion**, equivalent to **isotropic diffusion**;
- **geometry-driven diffusion**, the most general naming;
- **variable conductance diffusion**, the 'Perona and Malik' type of gradient magnitude controlled diffusion;
- **inhomogeneous diffusion**: the diffusion is different for different locations in the image; this is the most general naming;
- **anisotropic diffusion**: the diffusion is different for different *directions*;
- **tensor driven diffusion**: the diffusion coefficient is a tensor, not a scalar.

- **coherence enhancing diffusion**: the direction of the diffusion is governed by the direction of local image structure, for example the eigenvectors of the *structure matrix* or *structure tensor* (the outer product of the gradient with itself, explained in chapter 6), or the local ridgeness.
The Perona & Malik equation (1991):

\[ \frac{\partial L}{\partial s} = \nabla \cdot c(\|\nabla L\|) \nabla L \]

\[ c_1 = e^{-\frac{\|\nabla L\|^2}{k^2}} \]

\[ c_2 = \frac{1}{\left(1 + \frac{\|\nabla L\|^2}{k^2}\right)} \]

\[ 1 - \frac{(\nabla L)^2}{k^2} + \frac{(\nabla L)^4}{2k^4} + O((\nabla L)^5) \]

The conductivity terms are equivalent to first order.
The function $k$ determines the weight of the gradient squared, i.e. how much is blurred at the edges.

For the limit of $k$ to infinity, we get linear diffusion again.
Working out the differentiations, we get a strongly nonlinear diffusion equation:

\[
\frac{\partial L}{\partial s} = e^{-\frac{L_x^2 + L_y^2}{k^2}} \left( (k^2 - 2 L_x^2) L_{xx} - 4 L_x L_{xy} L_y + (k^2 - 2 L_y^2) L_{yy} \right) \]

The solution is not known analytically, so we have to rely on numerical methods, such as the forward Euler method:

\[
\delta L = \delta s \left( \nabla \cdot c \nabla L \right)
\]

This process is an evolutionary computation.
Test on a small test image:

Note the preserved steepness of the edges with the strongly reduced noise.
GDD is particularly useful for ultrasound edge-preserving speckle removal
Locally adaptive elongation of the diffusion kernel: 

Coherence Enhancing Diffusion

\[
\frac{1}{2\pi\sigma^2} e^{-\mathbf{x} \cdot \mathbf{D} \cdot \mathbf{x}}
\]
Coherence enhancing diffusion

J. Weickert, 2001
Interestingly, the S/N ratio increases during the evolution.

Signal = mean difference
Noise = variance
SNR = S/N
Interestingly, the S/N ratio has a maximum during the evolution. Blurring too long may lead to complete blurring (always some leak).

Signal = mean difference
Noise = variance

SNR = S/N
The Perona & Malik equation is ill-posed.

\[ L_s = \frac{\partial}{\partial x} \left( c \left( \frac{\partial L}{\partial x} \right) \frac{\partial L}{\partial x} \right) = c' L_x + c L_{xx} \]

Suppose that the flow \( c L_x \) is decreasing with respect to \( L_x \) at some point \( x_0 \). Then \( \frac{\partial}{\partial L_x} (c L_x) = c + c' L_x = -a \) with \( a > 0 \). Now \( c' = \frac{\partial c}{\partial L_x} \).

So \( L_s + a L_{xx} = 0 \), from which we get \( L_s = -a L_{xx} \).

Locally we have an inverse heat equation which is well known to be ill-posed. This heat equation locally blurs or deblurs, dependent on the condition of \( c \).
The function $c_1 L_x$ decreases for $L_x > \frac{1}{2} \sqrt{2} k$ and $c_2 L_x$ decreases for $L_x > k$. $k$ determines the turnover point.
Atherosclerotic plaque classification – P&M, $k = 50$, $\delta s = 5$
\[ \frac{\partial L}{\partial t} = \nabla \cdot e \frac{\| \nabla L \|^2}{k^2} \nabla L = \]

\[ \frac{1}{k^2} e \frac{L_x^2 + L_y^2 + L_z^2}{k^2} \left( k^2 (L_{xx} + L_{yy} + L_{zz}) - 2(L_x^2 L_{xx} + L_y^2 L_{yy} + L_z^2 L_{zz}) \right) \]

\[ -4(L_x L_y L_{xy} + L_x L_z L_{xz} + L_y L_z L_{yz}) \]
Stability of the numerical implementation

The maximal step size is limited due to the Neumann stability criterion (1/4). The maximal step size when using Gaussian derivatives is substantially larger:

\[ R = \frac{\Delta t}{\Delta x^2} \leq 2 \varepsilon s = \varepsilon \sigma^2 \]
Test image and its blurred version.

Note the narrow range of the allowable time step maximum.

The critical timestep $\Delta s = e \sigma^2 = 2.1718 \times 0.8^2 = 1.74$

timestep = 1.82857  timestep = 1.77778  timestep = 1.72973  timestep = 1.68421  timestep = 1.64103
Euclidean shortening flow:

\[
\frac{\partial L}{\partial s} = \nabla \cdot \left( \frac{\nabla L}{|\nabla L|} \right)
\]

Working out the derivatives, this is \(L_{vv}\) in gauge coordinates, i.e. the ridge detector.

\[
\frac{L_{yy} L_x^2 - 2 L_{xy} L_y L_x + L_{xx} L_y^2}{L_x^2 + L_y^2}
\]

So:

\[
\frac{\partial L}{\partial s} = L_{vv}
\]

Because the Laplacian is

\[
\Delta L = L_{xx} + L_{yy} = L_{vv} + L_{ww}
\]

we get

\[
\frac{\partial L}{\partial s} = \Delta L - L_{ww}
\]

We see that we have corrected the normal diffusion with a factor proportional to the second order derivative in the gradient direction (in gauge coordinates: \(\nabla\)). This subtractive term cancels the diffusion in the direction of the gradient.
This diffusion scheme is called Euclidean shortening flow due to the shortening of the isophotes, when considered as a curve-evolution scheme.

Advantage: there is no parameter k.
Disadvantage: rounding of corners.
<table>
<thead>
<tr>
<th>Name of flow</th>
<th>Luminance evolution</th>
<th>Curve evolution</th>
<th>Timestep N.N.</th>
<th>Timestep Gauss der.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>( \frac{\partial L}{\partial t} = \Delta L )</td>
<td>( \frac{\partial C}{\partial t} = - \frac{\Delta L}{</td>
<td>\nabla L</td>
<td>} )</td>
</tr>
<tr>
<td>Variable conductance</td>
<td>( \frac{\partial L}{\partial t} = \nabla \cdot (c \nabla L) )</td>
<td>( \frac{\partial C}{\partial t} = - \frac{\nabla \cdot (c \nabla L)}{</td>
<td>\nabla L</td>
<td>} )</td>
</tr>
<tr>
<td>Normal or constant motion</td>
<td>( \frac{\partial L}{\partial t} = cL_w )</td>
<td>( \frac{\partial C}{\partial t} = c \hat{N} )</td>
<td>( \frac{\Delta x}{c} )</td>
<td>-</td>
</tr>
<tr>
<td>Euclidean shortening</td>
<td>( \frac{\partial L}{\partial t} = L_{vv} )</td>
<td>( \frac{\partial C}{\partial t} = \kappa \hat{N} )</td>
<td>( \frac{(\Delta x)^2}{2} )</td>
<td>2 es</td>
</tr>
<tr>
<td>Affine shortening</td>
<td>( \frac{\partial L}{\partial t} = L_{vv} \frac{1}{3} L_w \frac{2}{3} )</td>
<td>( \frac{\partial C}{\partial t} = \frac{1}{3} \kappa \hat{N} )</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Affine shortening modified</td>
<td>( \frac{\partial L}{\partial t} = \left[ L_{vv} \frac{1}{3} L_w \frac{2}{3} \right] \sigma_1 )</td>
<td>( \frac{\partial C}{\partial t} = \kappa \frac{1}{3} \left( \frac{L_w}{k} \right)^{-\frac{2}{3}} \kappa \hat{N} )</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Entropy</td>
<td>( \frac{\partial L}{\partial t} = \beta_0 L_w + \beta_1 L_{vv} )</td>
<td>( \frac{\partial C}{\partial t} = (\beta_0 + \beta_1 \kappa) \kappa \hat{N} )</td>
<td>( -, \frac{(\Delta x)^2}{2} )</td>
<td>- , 2 es</td>
</tr>
</tbody>
</table>