Scale Space and PDE methods in image analysis and processing

Arjan Kuijper

Fraunhofer Institute for Computer Graphics Research
Interactive Graphics Systems Group, TU Darmstadt
Fraunhoferstrasse 5, 64283 Darmstadt, Germany

Tel.: +49 (0) 6151 155 - 103
Fax.: +49 (0) 6151 155 - 431
arjan.kuijper@igd.fraunhofer.de
http://www.gris.tu-darmstadt.de/~akuijper
Summary of the previous weeks

- The **Gaussian kernel**... 
  - Is a filter derived from “almost trivial” assumptions 
  - Is the solution of the heat equation 
  - Regularizes non-differential functions 
  - is a special case of the *Gabor* kernels. 
  - is a special case of the *α scale spaces*.
- **Scale** is an essential aspect of observations – the width of the kernel 
  - The scale **cannot** be taken too small or too large 
- **Derivatives** of images are to be taken by convolution of the image with the derivatives of the Gaussian filter. 
  - Gaussian derivatives are polynomials multiplied with the Gaussian and are **not** orthogonal kernels. 
- **Blurring** is stable, deblurring unstable. 
  - The **Wiener filter** is a special case of a regularizing filter 
  - a scale space Taylor expansion with negative time is a general regularized deblurring filter.
Today

- The differential structure of images
  - Differential image structure
  - Isophotes and flow lines

- Coordinate systems and transformations
- First order gauge coordinates
- Gauge coordinate invariants: examples

- Second order structure
- Third order image structure: T-junction detection
- Fourth order image structure: junction detection

- Scale invariance and natural coordinates
- Irreducible invariants

Differential image structure

- The differential structure of (discrete) images is the structure described by the local multi-scale derivatives of the image.

- Using heightlines, local coordinate systems and independence of the choice of coordinates.

- This is differential geometry, a field designed for the structural description of space and the lines, curves, surfaces etc. (a collection known as manifolds) that live there.

- Generate formulas for the detection of particular features, that detect special, semantically circumscribed, local meaningful structures (or properties) in the image, like edges, corners, T-junctions, monkey-saddles, etc.

- Only local!
Differential image structure

- Combinations of derivatives into expressions give nice feature detectors in images.

- Edges:

\[ \sqrt{\left( \frac{\partial L}{\partial x} \right)^2 + \left( \frac{\partial L}{\partial y} \right)^2} \]

- Corners:

\[ \left( \frac{\partial L}{\partial y} \right)^2 \frac{\partial^2 L}{\partial x^2} - 2 \frac{\partial L}{\partial x} \frac{\partial L}{\partial y} \frac{\partial^2 L}{\partial x \partial y} + \left( \frac{\partial L}{\partial x} \right)^2 \frac{\partial^2 L}{\partial y^2} \]

- Why do these work? Can we use any combination of derivatives? Does a reasonably small set of basis descriptors exist?
Isophotes and flow lines

Lines in the image connecting points of equal intensity are called isophotes. They are the heightlines of the intensity landscape when we consider the intensity as 'height'.

Example: Isophotes at different scales
Isophotes and flow lines

- Simple use: The **segmentation** method by thresholding and separating the image in pixels lying within or without the isophote at the threshold luminance.

- Properties:
  - isophotes are **closed curves**. Most isophotes in 2D images are a so-called Jordan curve: a non-self-intersecting planar curve topologically equivalent to a circle;
  - isophotes can intersect themselves. These are the critical isophotes. These always go through a **saddle point**;
  - isophotes do not intersect other isophotes;
  - any planar curve is completely described by its **curvature**, and so are isophotes;
  - isophote shape is **independent** of grayscale transformations, such as changing the contrast or brightness of an image.
Isophotes and flow lines

- A special class of isophotes is formed by those isophotes that go through a *singularity* in the intensity landscape, thus through a minimum, maximum or saddle point.
Isophotes and flow lines

- When the image is slightly changed, isophotes also change. **Critical isophotes** (those through critical points) are not stable:
Isophotes and flow lines

- **Flow lines** are the lines everywhere perpendicular to the isophotes.

- Flow lines are the *integral curves* of the gradient, made up of all the small little gradient vectors in each point integrated to a smooth long curve.

- In 2D, the flow lines and the isophotes together form a *mesh* or *grid* on the intensity surface.

- Just as in principle all isophotes together completely describe the intensity surface, so does the set of all flow lines.

- Flow lines are the *dual* of isophotes, vice versa.

- Just as the isophotes have a *singularity* at minima and maxima in the image, so have flow lines a singularity in direction in such points.
Isophotes and flow lines on the slope of a Gaussian blob. The circles are the isophotes, the flow lines are everywhere perpendicular to them. Inset: The height and intensity map of the Gaussian blob.
Coordinate systems and transformations

- **Local structure** is the local shape of the intensity landscape, like how sloped or curved it is, if there are saddle points, etc.

- The first order derivative gives us the slope, the second order is related to how curved the landscape is, etc.

- Use the Taylor expansion:

  \[ L(\delta x, \delta y) = L(0, 0) + \left( \frac{\partial L}{\partial x} \delta x + \frac{\partial L}{\partial y} \delta y \right) + \frac{1}{2!} \left( \frac{\partial^2 L}{\partial x^2} \delta x^2 + \frac{\partial^2 L}{\partial x \partial y} \delta x \delta y + \frac{\partial^2 L}{\partial y^2} \delta y^2 \right) + \frac{1}{3!} \left( \frac{\partial^3 L}{\partial x^3} \delta x^3 + \frac{3 \partial^3 L}{\partial x^2 \partial y} \delta x^2 \delta y + \frac{3 \partial^3 L}{\partial x \partial y^2} \delta x \delta y^2 + \frac{\partial^3 L}{\partial y^3} \delta y^3 \right) + O(\delta x^4, \delta y^4) \]

- However... The most important constraint for a good local image descriptor comes from the requirement that we want to be **independent** of our choice of coordinates.
Coordinate systems and transformations

- The **frame** of the coordinate system is formed by the unit vectors pointing in the respective dimensions.

- Focus on the change of
  - orientation (rotation of the axes frame),
  - translation (x and/or y shift of the axes frame), and
  - zoom (multiplication of the length of the units along the axes with some factor).

- We call all the possible instantiations of a transformation the **transformation group**.
  - All rotations form the rotational group.
    In 2D the coordinate frame is rotated over an angle $\phi$, the coordinates are multiplied with the matrix

\[
\begin{pmatrix}
\cos[\phi] & \sin[\phi] \\
-\sin[\phi] & \cos[\phi]
\end{pmatrix}
\]
Coordinate systems and transformations

- In general a transformation is described by a set of equations:

\[ x'_1 = f_1(x_1, x_2, \ldots, x_n) \]

\[ \vdots \]

\[ x'_n = f_n(x_1, x_2, \ldots, x_n) \]

- When we transform a space, the volume often changes, and the density of the material inside is distributed over a different volume. To study the change of a small volume we need to consider the matrix of first order partial derivatives, the Jacobian:

\[ J = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial(x')_1}{\partial x_1} & \ldots & \frac{\partial(x')_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial(x')_n}{\partial x_1} & \ldots & \frac{\partial(x')_n}{\partial x_n} \end{bmatrix} \]
Coordinate systems and transformations

- If we consider the change of the infinitesimally small volume, the determinant of the Jacobian is the factor which corrects for the change in volume.
  - When the Jacobian is unity, we call the transformation a special transformation.

- The transformation in matrix notation is expressed as \( \mathbf{x}' = A \mathbf{x} \)
  - A is the transformation matrix.
  - When the coefficients of A are constant, we have a linear transformation, often called an affine transformation.

- A rotation matrix that rotates over zero degrees is the identity matrix or the symmetric tensor or \( \delta \)-operator
  \[
  \begin{pmatrix}
  1 & 0 \\
  0 & 1 \\
  \end{pmatrix}
  \]

- The matrix that rotates over 90 degrees (\( \pi/2 \) radians) is called the anti-symmetric tensor, the \( \varepsilon \)-operator or the Levi-Civita tensor
  \[
  \begin{pmatrix}
  0 & 1 \\
  -1 & 0 \\
  \end{pmatrix}
  \]
Coordinate systems and transformations

- A function is said to be **invariant** under a group of transformations, if the transformation has no effect on the value of the function.

- The *only* geometrical entities that make **physically** sense are invariants.

- The derivatives to x and y are *not* invariant to rotation; However, the combination

\[
\sqrt{\left(\frac{\partial L}{\partial x}\right)^2 + \left(\frac{\partial L}{\partial y}\right)^2}
\]

is invariant: use

\[
\partial_x = (\cos \phi, \sin \phi) \cdot \nabla
\]

\[
\partial_y = (-\sin \phi, \cos \phi) \cdot \nabla
\]
Coordinate systems and transformations

Notice that with invariance we mean invariance for the transformation (e.g. rotation) of the coordinate system, not of the image. The value of the local invariant properties is the same when we rotate the image.
First order gauge coordinates

- Consider *intrinsic geometry*: every point is described in such a way, that if we have the same structure, or local landscape form, no matter the rotation, the description is always the same.

- This can be accomplished by setting up in each point a dedicated coordinate frame which is determined by some special local directions given by the landscape locally itself.

- *In each point separately* the local coordinate frame is fixed in such a way that one frame vector points to the direction of maximal change of the intensity, and the other perpendicular to it (90 degrees clockwise).
First order gauge coordinates

- So we set

\[
\vec{w} = \left( \frac{\partial L}{\partial x}, \frac{\partial L}{\partial y} \right)
\]

\[
\hat{v} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \vec{w} = \left( \frac{\partial L}{\partial y}, -\frac{\partial L}{\partial x} \right)
\]

- We have now fixed locally the direction for our new intrinsic local coordinate frame \((\hat{v}, \vec{w})\).
  This set of local directions is called a gauge, the new frame forms the gauge coordinates.

- Usually we divide the frame vectors by their length

- The frame can be rewritten as a rotation on the gradient vectors.
First order gauge coordinates

Example:
First order gauge coordinates

- We want to take derivatives with respect to the gauge coordinates. As they are fixed to the object, no matter any rotation or translation, we have the following very useful result:

  any derivative expressed in gauge coordinates is an orthogonal invariant.

- Then it is clear that \( \frac{\partial L}{\partial w} \) is the derivative in the gradient direction, and this is just the length of the gradient itself, an invariant.

- Furthermore, \( \frac{\partial L}{\partial v} \equiv 0 \), as there is no change in the luminance as we move tangentially along the isophote, and we have chosen this direction by definition.

\[
\partial_w = \frac{L_x \partial_x + L_y \partial_y}{\sqrt{L_x^2 + L_y^2}} = \frac{1}{\sqrt{L_x^2 + L_y^2}} w \cdot \nabla
\]

\[
\partial_v = \frac{L_y \partial_x - L_x \partial_y}{\sqrt{L_x^2 + L_y^2}} = \frac{1}{\sqrt{L_x^2 + L_y^2}} v \cdot \nabla
\]
First order gauge coordinates

- From the derivatives with respect to the **gauge coordinates**, we always need to go to **Cartesian coordinates** in order to calculate the invariant properties on a computer.

- The transformation to the \((\hat{v}, \hat{w})\) from to the Cartesian \((\hat{x}, \hat{y})\) frame is done by implementing the definition of the directional derivatives.

  Important is that **first** a directional partial derivative (to whatever order) is calculated with respect to a **frozen** gradient direction.

  **Then** the formula is calculated which expresses the gauge derivative into this direction, and **finally** the frozen direction is filled in from the **calculated** gradient.

\[
\partial_w = \frac{L_x \partial_x + L_y \partial_y}{\sqrt{L_x^2 + L_y^2}} = \frac{1}{\sqrt{L_x^2 + L_y^2}} w \cdot \nabla
\]

\[
\partial_v = \frac{L_y \partial_x - L_x \partial_y}{\sqrt{L_x^2 + L_y^2}} = \frac{1}{\sqrt{L_x^2 + L_y^2}} v \cdot \nabla
\]
Gauge coordinates

So:

\[ \partial_w = \frac{L_x \partial_x + L_y \partial_y}{\sqrt{L_x^2 + L_y^2}} = \frac{1}{\sqrt{L_x^2 + L_y^2}} w \cdot \nabla \]

\[ \partial_v = \frac{L_y \partial_x - L_x \partial_y}{\sqrt{L_x^2 + L_y^2}} = \frac{1}{\sqrt{L_x^2 + L_y^2}} v \cdot \nabla \]

Do the trick:

\[ \cos \phi = \frac{L_x}{\sqrt{L_x^2 + L_y^2}} \]

\[ \sin \phi = \frac{L_y}{\sqrt{L_x^2 + L_y^2}} \]

And thus

\[ \partial_w = (\cos \phi, \sin \phi) \cdot \nabla \]

\[ \partial_v = (\sin \phi, -\cos \phi) \cdot \nabla \]
First order gauge coordinates

This gives

\( L_w = \sqrt{L_x^2 + L_y^2} \)

\( L_{ww} = \frac{L_x^2 L_{xx} + 2 L_x L_{xy} L_y + L_y^2 L_{yy}}{L_x^2 + L_y^2} \)

\( L_{vv} = \frac{-2 L_x L_{xy} L_y + L_{xx} L_y^2 + L_x^2 L_{yy}}{L_x^2 + L_y^2} \)

\[ L_{vv} + L_{ww} = \ldots \]

\[ L_{xx} + L_{yy} \]
First order gauge coordinates

- The gauge coordinates are not defined if
  \[ L_x = 0 \quad \text{and} \quad L_y = 0 \]

- In practice however this is not a problem: we have a finite number of such points, typically just a few, and we know from Morse theory that we can get rid of such a singularity by an infinitesimally small local change in the intensity landscape.

- Due to the fixing of the gauge by removing the degree of freedom for rotation, we have an important result: every derivative to \( v \) and \( w \) is an orthogonal invariant.

- This also means that polynomial combinations of these gauge derivative terms are invariant.

- We have found a complete family of differential invariants, that are invariant for rotation and translation of the coordinate frame.
Ridge detection

$L_{vv}$ is a ridge detector, since at ridges the curvature of isophotes is large.
Isophote curvature

- **Isophote curvature** $\kappa$ is defined as the change of the tangent vector $w' = \frac{\partial w}{\partial v} = v$

  in the gradient-gauge coordinate system.

  The definition of an isophote is $L(v,w)=\text{Constant}$, and $w=w(v)$.

- Differentiation gives $w' = 0$

  Two times differentiating gives $\kappa = - \frac{L_{vv}}{L_w}$

- In Cartesian coordinates:

  $\kappa = - \frac{-2 L_x L_{xy} L_y + L_{xx} L_y^2 + L_x^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}}$
Isophote curvature

Example at several scales:

Tolansky's curvature illusion. The three circle segments have the same curvature 1/10:
Edge detection

- To find maxima of the gradient: use $L_{ww}$

- Historically, much attention is paid to the zero crossings of the Laplacian due to the groundbreaking work of Marr and Hildreth.
  - See CV online pages on Sobel and Laplace edge detection and Canny edge detection
  - The zero crossings are however displaced on curved edges.

- From the expression of the Laplacian in gauge coordinates:
  \[
  \Delta L = L_{ww} + L_{vv} = L_{ww} - \kappa L_w
  \]
  - There is a deviation term $\kappa L_w$, proportional to the isophote curvature $\kappa$.
  - Only on a straight edge with local isophote curvature zero the Laplacian is numerically equal to $L_{ww}$. 
Edge detection

- Contours of $L_{ww}$ (left) and $\Delta L=0$ (right) superimposed on an X-thorax image for a scale of 4 pixels.
### Gauge coordinates

- The term \( \gamma = - \frac{L_{ww}}{L_w} \) can be interpreted as a *density* of isophotes.

- The term \( \mu = - \frac{L_{vw}}{L_w} \) is the flow line curvature.

- Note that \( \kappa, \mu, \nu \) have equal dimensionality for the intensity in both nominator and denominator.
  - This leads to the desirable property that these expressions do not change when we e.g. manipulate the contrast or brightness of an image.
  - In general, these terms are said to be *invariant under monotonic intensity transformations*.
Affine invariant corner detection

- Corners are defined as locations with high isophote curvature and high intensity gradient.

- Take the product of isophote curvature $L_w$ and the gradient raised to some (to be determined) power $n$:

\[
\Theta[n] = -\frac{L_{vv}}{L_w} L_w^n = \kappa L_w^n = -L_{vv} L_w^{n-1}
\]

- An affine transformation is a *linear* transformation of the coordinate axes:

\[
\begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{1}{ad - bc} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}
\]

- The equation for the affinely distorted coordinates now becomes:

\[
-\frac{L_{va} v_a L_{wa}^{n-1}}{(b^2 + d^2) L_y^2 + (b c - a d)^2 \left( \frac{L_{xx} L_{xy} L_y - L_{xx} L_y^2 - L_x L_{yy}}{2} \right)} \left( \frac{L_{xx}^2 + 2 (a b + c d) L_{xx} L_y + (b^2 + d^2) L_y^2}{(b c - a d)^2} \right)^{\frac{1}{2} (-3+n)}
\]
Affine invariant corner detection

\[
\frac{1}{(b \ c - a \ d)^2} \left( \frac{(a^2 + c^2) \ L_x^2 + 2 \ (a \ b + c \ d) \ L_x \ L_y + (b^2 + d^2) \ L_y^2}{(b \ c - a \ d)^2} \right)^{\frac{1}{2} (-3+n)} \ (2 \ L_x \ L_{xy} \ L_y - L_{xx} \ L_y^2 - L_x^2 \ L_{yy})
\]

- When \( n=3 \) and for an affine transformation with unity Jacobean \((a \ d - b \ c=1, \) a so-called \textit{special} transformation) we are independent of the parameters \( a, \ b, \ c \) and \( d. \) This is the affine invariance condition.

- The expression

\[
\Theta = \frac{L_{vv}}{L_w} \ L_w^3 = L_{vv} \ L_w^2 = 2 \ L_x \ L_{xy} \ L_y - L_{xx} \ L_y^2 - L_x^2 \ L_{yy}
\]

is an \textit{affine invariant corner detector}. This feature has the nice property that it is not singular at locations where the gradient vanishes, and through its affine invariance it detects corners at all 'opening angles'.
Gauge coordinates

- Example $L_{vv} L_w^2$:

- Example $L_{ww} L_w^2$:
Gauge coordinates

Example $L_{vv}$:

Example $L_{ww}$: