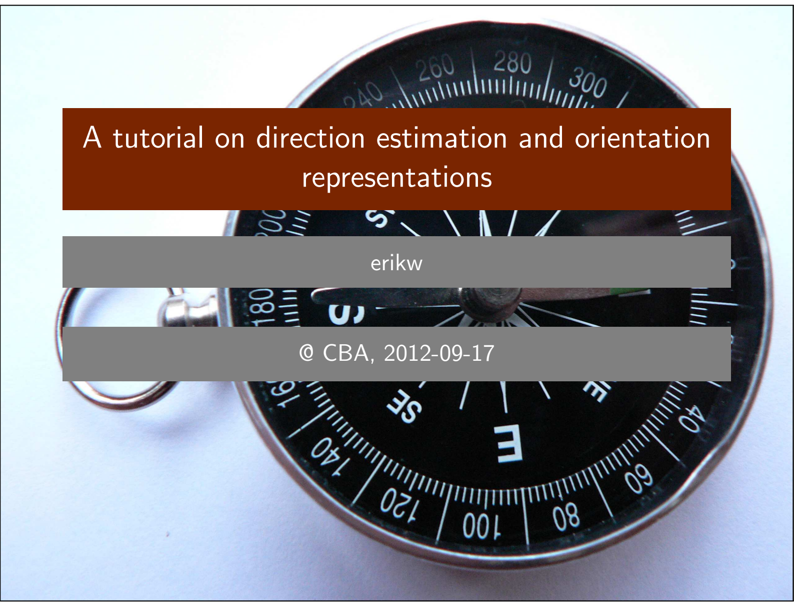


A tutorial on direction estimation and orientation representations

erikw

Introduction

Problem 1

Assign orientation to each pixel using local information

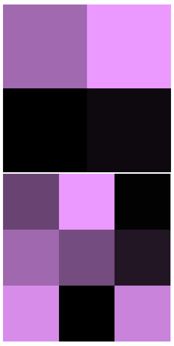
- Strategies
- Uncertainity principles

Problem 2

Orientation for larger regions, circular data etc...

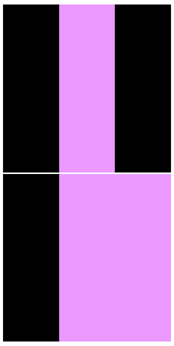
- Averaging
- Representation
- Orientation vs direction

Introduction
Type 1 problems

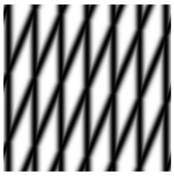


Local estimation

- Any small patch
- Lines
- Edges

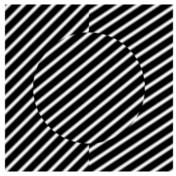


Introduction
Type 2 problems



Semi local/global estimation

- Averaging
- Multiple orientations
- Symmetric representations



Local estimation of orientation

Taylor series
Jacobian/Gradient

One variable

$$f(x) = \frac{f'(0)}{1!}x + \frac{f''(x)}{2!}x^2 + \ldots$$

Several variables (dimensions)

Let $f : R^N \rightarrow R$, then

$$f(\mathbf{x}) = f(\mathbf{0}) + Df(\mathbf{0})^T\mathbf{x} + \ldots$$

$$Df(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{0}), \frac{\partial f}{\partial x_2}(\mathbf{0}), \ldots, \frac{\partial f}{\partial x_N}(\mathbf{0})\right)$$

and $Df : R^N \rightarrow R^N$

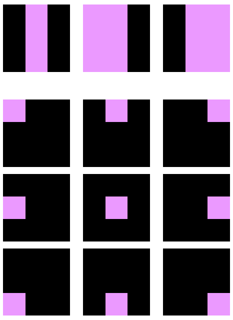
Finite differences
Directions from discretized derivatives I

To discretise the gradient, the smallest stencil is: $[1, -1]$ for each partial derivative.

In matlab

```
dx=convn(I, [1,-1], 'same');
dy=convn(I, [1,-1]', 'same');
```

- Very local, depends only on three pixels.
- Half pixel offset!
- Rotationally invariant? (I.e. do we get the same result if we rotate the image first, then calculate the gradient, and then rotate back?) See fig (1,2) and (1,3)!



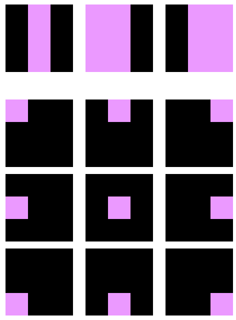
Finite differences
Directions from discretized derivatives I

Second smallest filter: $[1, 0, -1]/2$

In matlab

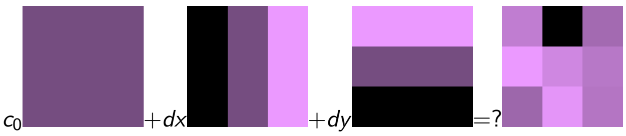
```
dx=convn(I,[1,0,-1]/2,'same');
dy=convn(I,[1,0,-1]'/2,'same');
```

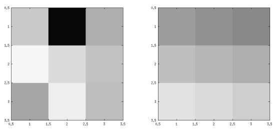
- Invariant to [...,1,0,1,0,1,...]
- Still a little too discrete?
- Symmetric
- Local



Gradient as a Least Squares Problem

$$\begin{pmatrix} 1/9 & -1 & 1 \\ 1/9 & -1 & 0 \\ 1/9 & -1 & -1 \\ 1/9 & 0 & 1 \\ 1/9 & 0 & 0 \\ 1/9 & 0 & -1 \\ 1/9 & 1 & 1 \\ 1/9 & 1 & 0 \\ 1/9 & 1 & -1 \end{pmatrix} \begin{pmatrix} c_0 \\ dx \\ dy \end{pmatrix} =? \begin{pmatrix} M(1,1) \\ M(2,1) \\ M(3,1) \\ M(1,2) \\ M(2,2) \\ M(3,2) \\ M(1,3) \\ M(2,3) \\ M(3,3) \end{pmatrix}$$





When the solution x to $Ax = b$ can't be found by matrix inverse (i.e. too low rank), we can find

$$\arg\min_x ||Ax - b||^2$$

i.e.

$$x = (A^T A)^{-1} A^T y$$

Least Squares and Projections

Say we have some data points $\{y_i\} := y(x_i), i = 1, ..., N$ and a basis function $\{b_i\}$. Now we want to find the c that minimises

$$E(c) = ||cb(x) - y(x)||. \tag{1}$$

In the least squares approach, we expand Eq. 1 as

$$E(c) = \sum_{i=1}^{N} \left[c^2 b_i^2 - 2cb_i y_i + y^2\right]. \tag{2}$$

Least Squares and Projections

Derivation with respect to c:

$$\frac{d}{dc} E(c) = \sum \left[cb_i^2 - b_i y_i\right] = 0, \tag{3}$$

gives

$$c = \frac{\sum b_i y_i}{\sum b_i^2}. \tag{4}$$

With the projection approach, the projection of y to b is expressed

$$\mathrm{Proj}_b y = b \frac{(b, y)}{||b||^2} \tag{5}$$

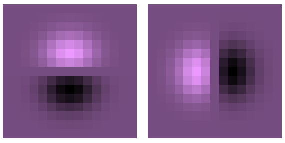
so we identify

$$c = \frac{(b, y)}{||b||^2} = \frac{by^T}{bb^T} = \frac{\sum b_i y_i}{\sum b_i^2}. \tag{6}$$

Gaussian Derivatives
Scale Space Theory

Equivalences
- Convolutions
- Projection on linear bases
- Least squares solutions

- Rotational invariance $\rightarrow$ round support
- Non ringing $\rightarrow$ smooth radial profile
- $\rightarrow$ Gaussian derivatives!



See Koenderink and Lindeberg!

Gaussian derivatives in Matlab
Generalises to ND

```
function y=gpartial(x, d, sigma)
w = 10*ceil(sigma); w = w+mod(w+1,2); % Filter length
g =fspecial('gaussian', [w,1], sigma); % 1D Gaussian
x=(-(w-1)/2:(w-1)/2)';
k0=1/sqrt(2*pi*sigma^2); k1=1/(2*sigma^2);
dg=-2*k0*k1.*x.*exp(-k1*x.^2); % d/dx Gaussian
if d==1
  y=convn(x, reshape(dg, [w,1]), 'same');
  y=convn(y, reshape( g, [1,w]), 'same');
end
if d==2
  y=convn(x, reshape( g, [w,1]), 'same');
  y=convn(y, reshape(dg, [1,w]), 'same');
end
```

A1: Sub pixel location of extremal points

Another example of Taylor expansion in image analysis. Sub pixel location of local extreme points is acheived by second order Taylor expansion using the 3^N closest points by:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial x^2}\mathbf{x}$$

$$\hat{\mathbf{x}} = -\left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1}\frac{\partial D}{\partial \mathbf{x}}$$

A2: Location of edges

A one dimensional signal $P(x)$. We define a unit step edge located at $x = 0$ by

$$\theta(x) = \begin{cases} 1, & x \geq 0 \\ 0, & < 0 \end{cases} \tag{7}$$

Def 1: An edge can be located where the first derivative of the signal has an extremal value (zero crossing of second derivative)

$$E_D = \{x \;:\; \frac{d^2}{dx^2}P(x) = 0\}. \tag{8}$$

Def 2: the edge can be located where the signal obtains a specific value or level c, i.e. the set

$$E_l = \{x \;:\; P(x) = c\}. \tag{9}$$

A2: Location of edges
Definitions

$$G_\sigma(x) := \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-x^2}{2\sigma^2}} := ae^{-x^2 b}, \tag{10}$$

$$a = \frac{1}{\sqrt{2\pi\sigma^2}}, \quad b = \frac{1}{2\sigma^2}$$

$$\mathrm{erf}_\sigma(x) = \int_{-\infty}^{x} G_\sigma(\xi)d\xi, \tag{11}$$

A2: Location of edges
Differential Definition, Def 1

Step edges: $P(x) = G_\sigma * \theta(x)$, E_D is the set of points that satisfy

$$0 = \frac{d^2}{dx^2}G_\sigma * \theta(x) = \frac{d^2}{dx^2}\mathrm{erf}_\sigma(x) = G'_\sigma(x), \tag{12}$$

gives $E_D = \{0\}$

Lines: $P = \delta(x) \approx \frac{1}{\epsilon}(\theta(x) - \theta(x+\epsilon))$ for a small ϵ. E_D is the set of points that satisfies,

$$0 = \frac{d^2}{dx^2}\delta * G_\sigma(x) = e^{-bx^2}\left(4x^2b^2a - 2ab\right), \tag{13}$$

which are

$$x = \pm\sqrt{\frac{2ab}{4ab^2}} = \pm\sqrt{2}\sigma. \tag{14}$$

Two detections, none at $x = 0$.

A2: Location of edges

Unit ridges $P(x) = \theta(x) - \theta(x - w)$, where $w > 0$ is the width. E_D contains the points that satisfy

$$0 = G'_\sigma(x) - G'_\sigma(x - w) = -2abxe^{-bx^2} + 2ab(x-w)e^{-b(x^2-2wx+w^2)}, \quad (15)$$

or simplified

$$0 = 2abe^{-bx^2}\left\{(x-w)e^{-b(w^2-2wx)} - x\right\}. \quad (16)$$

which further can be reduced to

$$0 = x - (x - w)e^{-b(w^2-2wx)}. \quad (17)$$

Neither $x = 0$ or $x = w$ are solutions.

A2: Location of edges
Iso-level definition, def 2

For an ideal step edge,

$$E_I = \{x \ : \ G_\sigma * \theta(x) = \mathrm{erf}_\sigma(x) = c\}, \quad (18)$$

and since $E_i = \{0\}$ is required, $c = 1/2$.
For lines, no, one or two edges will be detected since the condition is that

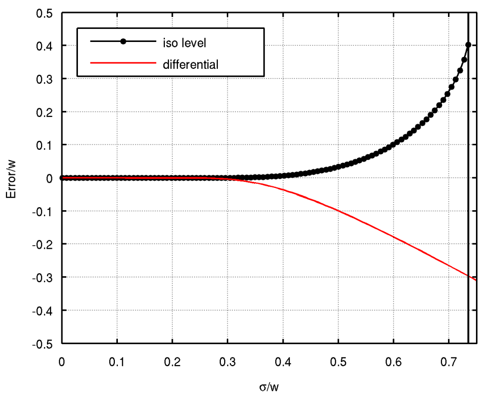
$$E_I = \{x \ : \ G_\sigma(x) = 1/2\}. \quad (19)$$

For finite ridges,

$$E_I = \{x : \mathrm{erf}_\sigma(x) - \mathrm{erf}_\sigma(x - w) = 1/2\}. \quad (20)$$

none (or one) or two edges are detected.

Location error for ridges



A2: Location of edges

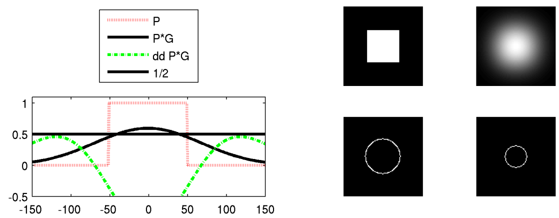


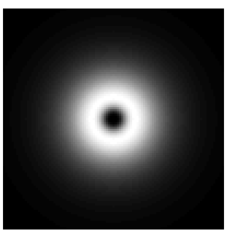
Figure: Left: A unit ridge, smoothed, its second derivative (scaled) and the 1/2 line. Right: NW: A ridge. NE: Smoothed with $\sigma/w = 0.7$. SW: Canny edge detection. SE: Pixels with intensity above 0.5.

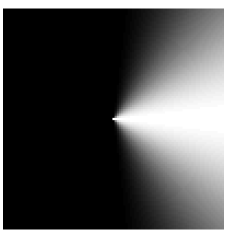
GOP / Quadrature Filters
Phase invariance

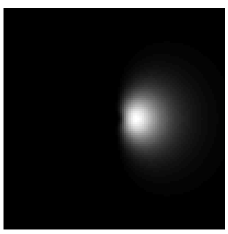
$$F(u, v) = F(r, \theta) = R(r)T(\theta). \quad (21)$$

$$R(\rho) = e^{\frac{-4}{B^2 \ln 2}\ln^2(\rho/\rho_0)} \quad (22)$$

$$T_{\mathbf{d}}(u) = \begin{cases} \langle u, \mathbf{d}\rangle^2, & \langle u, \mathbf{d}\rangle > 0, \\ 0, & \langle u, \mathbf{d}\rangle \le 0. \end{cases} \quad (23)$$

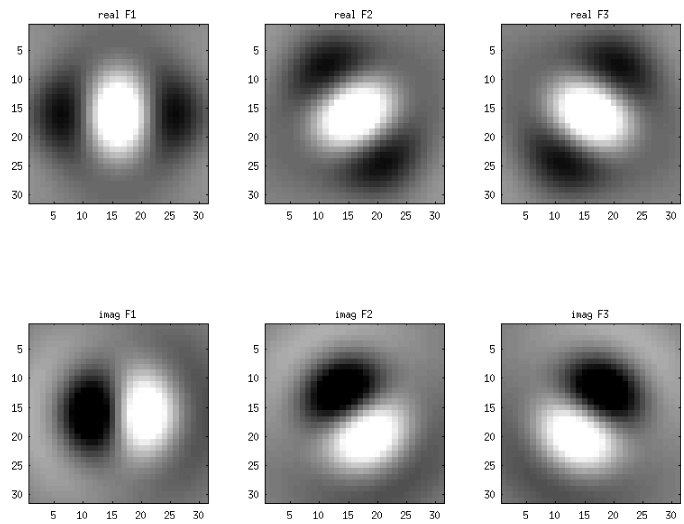

radial part


angular part


combined

→ can also be achieved by averaging techniques.

Quadrature filter in the spatial domain



Intermediate summary:

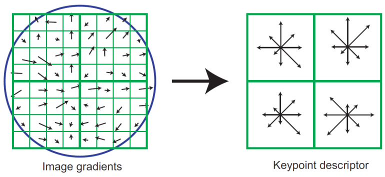
- Gaussian derivatives to calculate gradients!
- Gradients vanish for some structures.
- Higher order constructions are needed needed. One such technique is the Hessian.
- Phase invariant filters are good.
- Sub pixel location of edges is not trivial (see Van Vleet)

Representing directions and orientations

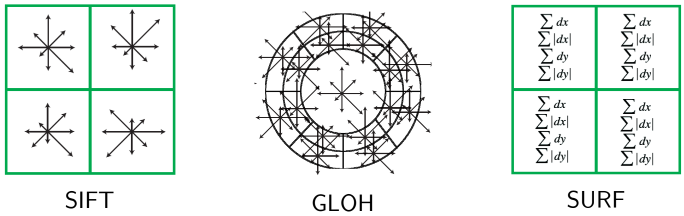
Histograms

Example: SIFT (2004)

- Angular Histogram, $[0, 2\pi]$ is divided into eight bins
- Gradient directions $\theta = \text{atan2}(dx, dy)$.
- 16 spatial bins, 8 angular bins (128)
- Gaussian Weights
- 128-dimensional



Approaches from the SIFT family



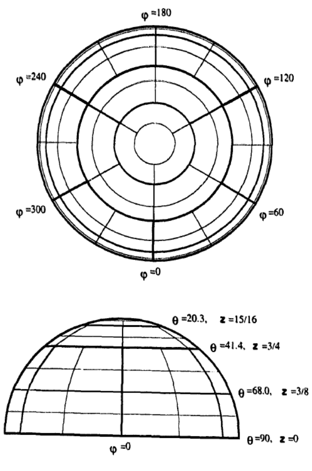
Properties of Histograms

Histograms:

- Discretizations
- Number of binns
- Rotations do no commute
- Discontinuous at $2\pi = 0$
- Quantitative
- Tesselation in $R^{>2}$

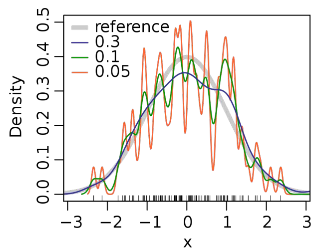
Another representation?

1. Commuting rotations
2. Discontinuity-free
3. Perfect retrieval



G. Borgefors

Kernel Density Estimators (KDE)



E. Parzen *On Estimation of A Probability Density Function and Mode* Ann. Math. Statist. 33(3) 1962

"Given a sequence of independent identically distributed random variables $X_1, X_2, \ldots, X_n, \ldots$ with common probability density function $f(x)$, how can one estimate $f(x)$?"

- Extensions to manifolds
- A standard approach, > 5000 citations.

Derivation, pt. I

- The KDE is a linear sum of weighting functions

$$K(x) = \sum_{i=1}^{N} W_N(x - x_i)$$

- Circular means that $x \in [-\pi, \pi)$ and $\lim_{x \to \pi} K(x) = K(-\pi)$

Derivation, pt. II

- Express K as a Fourier series (parameter: M)

$$K = \sum_{k=0}^{\infty} c_k e^{ik\theta} = \underbrace{\sum_{k=0}^{M-1} c_k e^{ik\theta}}_{K_F} + \sum_{k=M}^{\infty} c_k e^{ik\theta}$$

- The coefficients are

$$c_k = \langle K, e^{-ik\theta} \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} K e^{-ik\theta} d\theta$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{i=1}^{N} \{W(\theta - \theta_i)\} \, e^{-ik\theta} d\theta$$

Derivation, pt. III

- Since the formulas are linear, the contribution from each sample to the coefficients of the fourier series can be split. Let i enumberate the samples such that

$$c_k = \sum_{i=1}^{N} c_{ki}$$

- Then the coefficients are given by

$$c_{ki} = \frac{1}{2\pi} \int_{-\infty}^{\infty} W(\theta - \theta_i) e^{-ik\theta} d\theta$$

Derivation, pt. IV

- A Gaussian W yields a simple expressoin. Let

$$W(x) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{x^2}{2\sigma_w^2}}$$

- A few calculations later, we get the contribution to the series from each sample:

$$c_{ki} = 2\pi e^{-ik\theta_i} e^{-\frac{k^2\sigma_w^2}{2}}$$

Summary

Parameters: M, σ_w
Representation: $\{c_i, i = 0, ..., M-1\}$

Relation to the structure tensor

Set weighting function to $\cos^2(x)$, then for one observation, the kde is,

$$f(\theta) = \cos^2(\theta - \theta_0) = (\cos\theta \cos\theta_0 + \sin\theta \sin\theta_0)^2$$

The structure tensor constructed from the same angle $s = (\cos\theta_0, \sin\theta_0)$ is

$$S = ss^T = \begin{pmatrix} \cos^2\theta_0 & \sin\theta_0 \cos\theta_0 \\ \sin\theta_0 \cos\theta_0 & \sin^2\theta_0 \end{pmatrix}.$$

- So, for an arbitrary angle, $v = (\cos\theta, \sin\theta)$, $v^T S v = f(\theta)$.
- Induction and linearity gives the full story
- Conclusion: The structure tensor admits an interpretation as a special kde.

The gradient structure tensor

```
function st = gst(I, dsigma, tsigma)
% Calculate the image gradient
g = zeros([size(V), 3]);
for kk=1:3
    g(:,:,:,kk)=gpartial(V, kk, dsigma);
end
% gradient to structure tensor
st=zeros([size(g,1),size(g,2),size(g,3), 6]);
st(:,:,:,1)=g(:,:,:,1).*g(:,:,:,1);
st(:,:,:,2)=g(:,:,:,1).*g(:,:,:,2);
st(:,:,:,3)=g(:,:,:,1).*g(:,:,:,3);
st(:,:,:,4)=g(:,:,:,2).*g(:,:,:,2);
st(:,:,:,5)=g(:,:,:,2).*g(:,:,:,3);
st(:,:,:,6)=g(:,:,:,3).*g(:,:,:,3);
% Average per coefficient
for kk=1:6
        st(:,:,:,kk)=gsmooth(st(:,:,:,kk), tsigma);
end
```

The outer product $(\nabla I)^T \nabla I$

The gradient of I is

$$\nabla I = (\frac{\partial}{dx_1}, \frac{\partial}{dx_2}, \frac{\partial}{dx_2}),$$

so the structure of the outer product

$$E := (\nabla I)^T \nabla I \approx \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix}. \qquad (24)$$

- E is Self-Adjoint since it is real and symmetric.
- The *Spectral Theorem* for real vector spaces then states that the eigenvectors to E, v_i form an orthonomal (ON) basis.
- A shorter proof that the eigenvectors corresponding to distinct eigenvalues are ON. Assume that $Ed = \delta$ and $Ee = \epsilon$ then

$$(\delta - \epsilon)\langle d, e\rangle = \langle Td, e\rangle - \langle d, T^* e\rangle = \langle Td, e\rangle - \langle Td, e\rangle = 0$$

and since $\delta - \epsilon \neq 0$, it hold that $\langle d, e\rangle = 0$.

Using S_x as a quadratic form

Denote the eigenvalues to S_x as λ_i and the eigenvectors v_i. Then the structure tensor maps vectors as

$$\begin{aligned} \langle Ew, w\rangle &= \langle \lambda_1 \operatorname{Proj}_{v_1} w + \lambda_2 \operatorname{Proj}_{v_2} w + \lambda_3 \operatorname{Proj}_{v_3} w, w\rangle \\ &= \lambda_1 \langle\langle w, v_1\rangle v_1, w\rangle + \lambda_2 ... \\ &= \lambda_1 \langle w, v_1\rangle\langle w, v_1\rangle + \lambda_2 ... \\ &= \lambda_1 \cos^2\theta_1 + \lambda_2 \cos^2\theta_2 + \lambda_3 \cos^2\theta_3 \end{aligned}$$

Where the angles θ_i is the angle between w and each eigenvector, v_i.

The 2x2 eigenvalue problem

Eigenvalues

The eigenvalue problem det $Ax = \lambda x$ has the characteristic polynomial $(a-\lambda)(c-\lambda) - b^2 = 0$ when $A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ and the solutions $\lambda = \frac{a+c}{2} \pm \sqrt{b^2 - ac + (\frac{a+c}{2})^2}$, equivalent to $\lambda = \mathrm{Tr}/2 \pm \sqrt{(\mathrm{Tr}/2)^2 - \mathrm{D}}$, where $\mathrm{Tr} = \mathrm{Trace}\, A$ and $\mathrm{D} = \mathrm{Det}\, A$.

Eigenvectors

If we set $x_1 = 1$, we get $x_2 = -b/(c-\lambda)$. When $b \approx 0$, A is diagonal and $\mathbf{x} = (1,0)^T$ when $\lambda \approx a$ and $(0,1)^T$ when $\lambda \approx c$.

The symmetric eigenvalue problem

Introduction

1. The 3x3 eigenvalue problem i.e. to find $x \in R^3 - (0,0,0)$ and $\lambda \in R$ which satisfies $Ax = \lambda x$ for $A = A^T \in R^{3x3}$.
2. Multiple approaches possible.
3. Cardano's solution to the characteristic equation $(det(Ax - \lambda I) = 0$ is not suited for numerical computations. (Demmel)
4. Jacobis method is the fastest?

A plane rotation matrix

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

has the properties $R^{-1}(\theta) = R(-\theta)$. A 2×2 real and symmetric matrix

$$M = \begin{pmatrix} \alpha & \gamma \\ \gamma & \beta \end{pmatrix}$$

can be diagonalised with such rotation matrix so that

$$R^{-1} M R = D. \qquad (25)$$

After the rotation, D and M are similar, i.e. have the same eigenvalues.

θ that makes D diagonal is not explicitly needed:

$$\epsilon = \frac{\alpha - \beta}{2\gamma},$$

$$t = \frac{|\epsilon|}{|\epsilon| + \sqrt{1+\epsilon^2}},$$

$$c := \cos\theta = (1+t^2)^{-1/2} \quad s := \sin\theta = ct.$$

And,

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix}\begin{pmatrix} \alpha & \gamma \\ \gamma & \beta \end{pmatrix}\begin{pmatrix} c & s \\ -s & c \end{pmatrix} = \begin{pmatrix} \alpha - \gamma t & 0 \\ 0 & \beta + \gamma t \end{pmatrix}.$$

With Jacobi rotations, two-dimensional subspaces are rotated. There are three of them:

$$R_{12} = \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}, R_{13} = \begin{pmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{pmatrix}, R_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{pmatrix}.$$

To use those matrices iteratively to diagonalise A is the core of the Jacobi method.

1. Input $A_0 := A$. Initialise $E_0 := \text{diag}(1, 1, 1)$ which will contain the eigevectors and set the tolerances value $tol = 10^{-14}$.
2. Find the largest off diagonal element of $A_n(i, j)$,

$$(i, j) = \arg\max |A_n(i, j)|, i < j.$$

3. Find c and s using

$$\alpha = A(ii), \beta = A(j, j), \gamma = A(i, j)$$

4. Rotate A, $A_n := R_{ij} A_{n-1} R_{ij}^T$
5. Rotate E, $E_n := R_{ij}^T E_{n-1}$
6. If $\max |A_n(ij)| <$ tol end, else repeat from step 2.

- Matrix multiplications are explicitly written out (generality vs speed)
- Quadratic convergence
- Well suited for parallelisation
- 30% faster than DIPLib (single core)
- Get code from me

Direction vs Orientation I

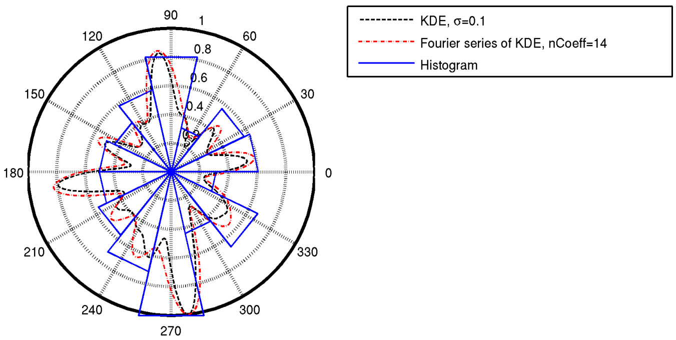
A vector in a metric space represents a direction. In $\mathbb{R}^N$, $N-1$ scalars are required (example). A direction points out how to get from point A to point B in R^N An orientation tells you to point your nose at B and have your feet down. There is a strong relationship between orientations and rotations. The natural setting for a discussion on orientations is group theory (see my thesis!) Bild: Jordglob

Direction vs Orientation II

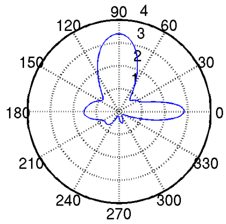
The dimensionality of orientation

Of necessity, rotation matrices are ON. All eigenvalues have length 1. The minimal number of elements that are needed to describe this is $1 + 2 + ... + (N-1) = N(N-1)$ (odd dimensions)

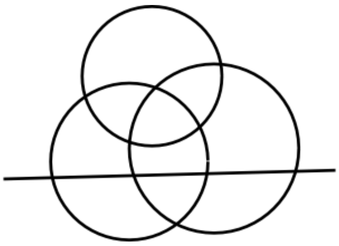
Example I, KDE vs histogram



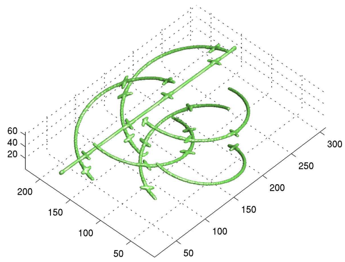
Example II, structure description



Example III, rotation space

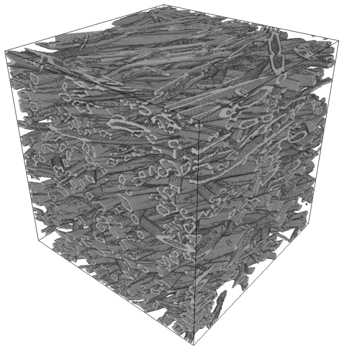


Input, 2D image

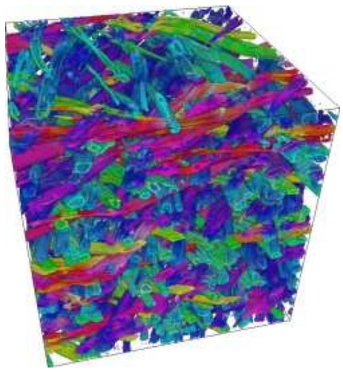


Outut, 3D. Isosurface shown

Example IV, Structure Tensor



CT image of wood fibre/plastic composite



Pseudo colored by orientation

(Maria Axelsson)

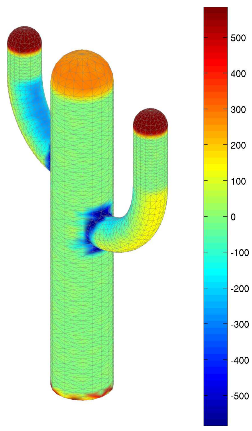
Example V, Structure Tensor

CT image of wood fibre/plastic composite

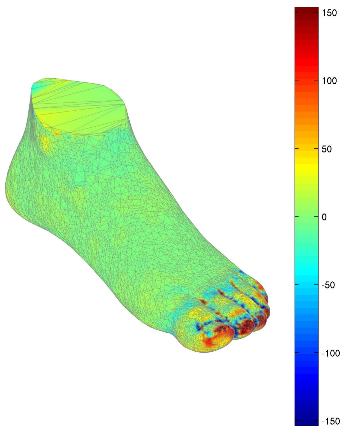
Pseudo colored by orientation

Example VI, curvature

On meshes



Gaussian Curvature $k_1 k_2$



Gaussian Curvature $k_1 k_2$

Summary

- Not to choose is also a choice!
- There are a few different techniques for local direction estimation.
- For larger regions, orientation can be estimated as well.
- I'd like to see more KDEs!
- There is much more to this subject!

Selected References

- Michael Van Ginkel, *Image Analysis using Orientation Space Based on Steerable Filters*, PhD Theis, 2002
- Gösta Granlund, *In Search for a General Picture Processing Operator*, Computer Graphics and Image Processing, 8, 1978
- Heinrich W. Guggenheimer, *Differential Geometry*, Dover, 1977