Clifford Convolution and Fourier Transform on Vector Fields

Julia Ebling, Gerik Scheuermann
University of Leipzig

Flow Feature Detection Using Image Processing?

- Pattern matching intuitive
- Convolution robust in terms of noise
- Analysis of filter behaviour

Convolution and Fourier Transform on Vector Fields

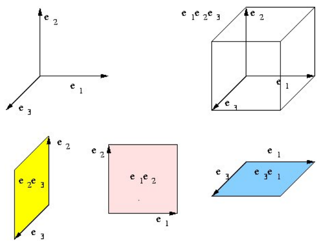
- Convolution of each coordinate separately [Granlund, Knutsson 1995]
- Scalar product in convolution [Heiberg 2001]
- Clifford Convolution [Ebling, Scheuermann 2003]
- Clifford Fourier Transform [Ebling, Scheuermann 2004]

Overview

- Clifford Algebra
- Clifford Convolution
- Clifford Fourier transform
- Gabor Filter
- Future work

Clifford Algebra

For Euclidean 3D-space, we use a 8-dimensional real algebra G^3 with the vector basis $\{1, e_1, e_2, e_3, e_1e_2, e_3e_1, e_2e_3, e_1e_2e_3\}$.



Clifford Algebra

Multiplication is bilinear and associative with the rules:

$$\begin{aligned} 1\, e_j &= e_j & j &= 1,2,3 \\ e_j e_j &= 1 & j &= 1,2,3 \\ e_j e_k &= -e_k e_j & j,k &= 1,2,3,\ j \neq k \end{aligned}$$

Multiplication is not commutative!

Clifford Algebra

Vectors are described as

$$v = x e_1 + y e_2 + z e_3 \in E^3 \subset G^3$$

and an arbitrary multivector can be described as

$$A = \alpha + a + i(b + \beta) \in G^3 \quad \alpha, \beta \in \mathbb{R}, a, b \in E^3$$

with $i = e_1e_2e_3$.

Clifford Algebra

In Clifford algebra, the multiplication of vectors describes the complete geometric relation between two vectors:

$$ab = a \cdot b + a \wedge b$$

Here, $a \cdot b$ is the scalar (inner) product between two vectors and $a \wedge b$ is the outer product:

$$\begin{aligned} a \cdot b &= |a||b| \cos(a;b) \\ |a \wedge b| &= |a||b| \sin(a;b) \end{aligned}$$

Clifford Algebra

Let $A : E^3 \rightarrow G^3$ be a multivector field. As directional derivative, we define

$$A_b(r) = \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} [A(r+\epsilon b) - A(r)], \quad \epsilon \in \mathbb{R}$$

and as total derivative

$$\partial A(r) = \sum_{k=1}^{3} e_k A_{e_k}(r)$$

As integral, we define

$$\int_{E^3} A\,dx = \lim_{i \rightarrow \infty} \sum_i A(x_i) \Delta(x_i)$$

Clifford Algebra

Curl and divergence of a vector valued function f are defined as:

$$curl(f) = \nabla \wedge f = \frac{(\partial f - f \partial)}{2}$$

$$divergence(f) = \langle \nabla, f \rangle = \frac{(\partial f + f \partial)}{2}$$

Clifford Convolution

With the integral, the Clifford convolution is defined as

$$(F * V)(x) = \int_{E^3} F(y) V(x-y)\,dy$$

$$(F * V)(j,k,l) = \sum_{s,t,u=-r}^{r} F(s,t,u) V(j-s, k-t, l-u)$$

A comparison with [Heiberg 2001] shows that we get for a vector filter his scalar convolution plus a bivector part describing the relative position in space.

Interpretation

Instead of the convolution, we may look at the spatial coherence:

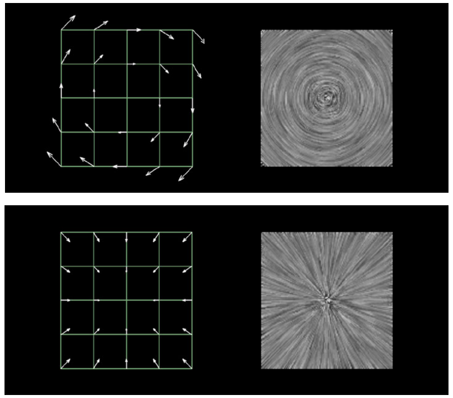
$$(F \times P)(x) = \int_{E^3} F(y) P(x+y)\,dy$$

$$(F \times P)(j,k,l) = \sum_{s,t,u=-r}^{r} F(s,t,u) P(j+s, k+t, l+u)$$

At each position x, we compute the coherence between the mask centered at its origin and the vector field! By the Clifford product, we get the relative geometric position between mask and vector field.

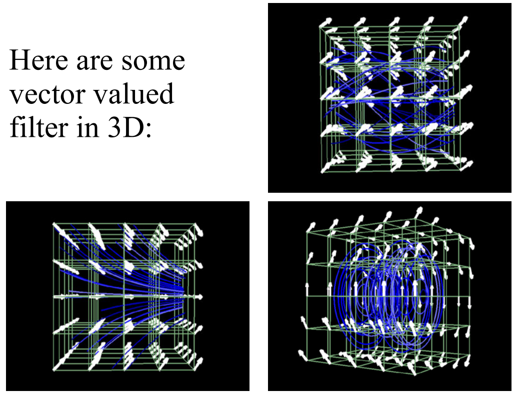
Vector Filter

As vector valued masks for pattern matching we can use typical pattern such as rotation or convergence.



Vector Filter

Here are some vector valued filter in 3D:



Pattern matching

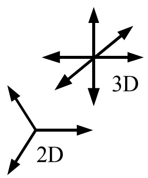
Clifford Convolution:

- Approximation of rotation between local structure in field and mask
- Rotate mask to align field and mask
- Compute scalar convolution for similarity

→ **Rotation invariant pattern matching**

Approximation not good enough:

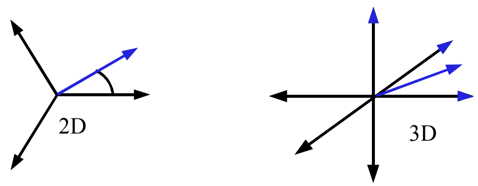
- Use 3 (2D) /6 (3D) mask directions to compute the local direction



Pattern matching

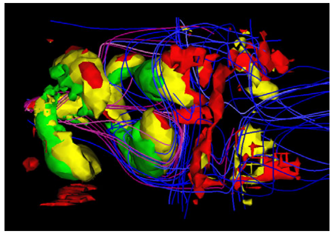
Computation of local direction:

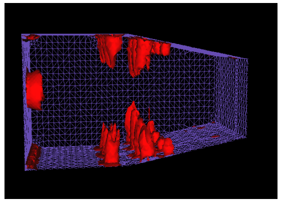
- 2D: approximation with smallest angle
- 3D: weighted averaging of approximations with positive scalar

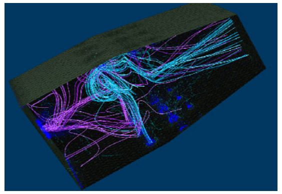


Gas furnace chamber

Gas furnace chamber. Pattern matching with different mask sizes: 3◊3◊3 (red), 5◊5◊5 (yellow), 8◊8◊8 (green)

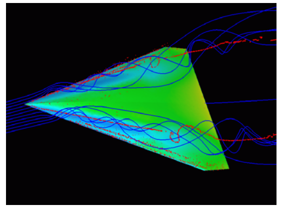


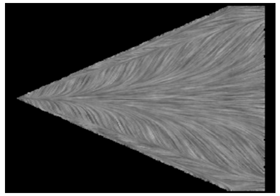


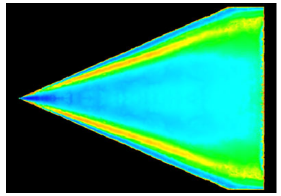


Flattened surface of delta wing

Delta wing at 0.2 mach velocity,
angle of attack: 25∞







Clifford Fourier transform

- $\{1, I_3 = e_1 e_2 e_3\}$ isomorph to complex numbers
- I_3 commutes with every multivector
- Use I_3 instead of *i* in Fourier Kernel

Clifford Fourier Transform in 3D:

$$F\{f\}(u) = \int f(x)\, e^{-2\pi I_3 \langle x,u\rangle}\, dx$$

$$F\{f\}(u) = \int f(x)\, e^{2\pi I_3 \langle x,u\rangle}\, dx$$

Clifford Fourier Transform in 3D:

- Shift Theorem: $F\{(x-x')\}(u)=F\{f\}(u)e^{-2\pi I_3 \langle x', u\rangle}$
- Convolution: $F\{h*f\}(u)=F\{h\}(u)F\{f\}(u)$
- Derivation: $F\{\nabla f\}(u)=2\pi I_3 u F\{f\}(u)$

  $F\{\Delta f\}(u)=-4\pi^2 u^2 F\{f\}(u)$

Clifford Fourier Transform in 3D:

- Basicly 4 complex Fourier transforms of:

1 - $e_1e_2e_3$
e_1 - e_2e_3
e_2 - e_3e_1
e_3 - e_1e_2,

Clifford Fourier Transform in 2D:

- $\{1, I_2=e_1e_2\}$ isomorph to complex numbers
- I_2 commutes with every spinor
  I_2 anticommutes with vector
- Use I_2 instead of *i* in Fourier Kernel
- Theorems a bit more complicated

Clifford Fourier Transform in 2D:

- Basicly 2 complex Fourier transforms of:

1 - e_1e_2
e_1 - e_2

as $a_1e_1 + a_2e_2 = e_1(a_1 1 + a_2e_1e_2)$
can be understood as a complex number

Clifford Fourier Transform:

- Parsevals theorem
- Sampling theorem
- Discretization
- Fast transform

Turbulent swirling jet entering fluid at rest

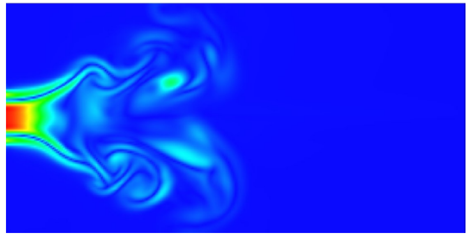
Resolution: 256*128

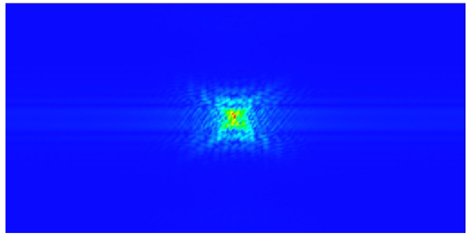
Color coding of the absolute values of the (multi) vectors

Top:
original vector field

Bottom:
fast CFT, 2D vectors are converted to vectors

(3D vectors convert to vector + bivector)

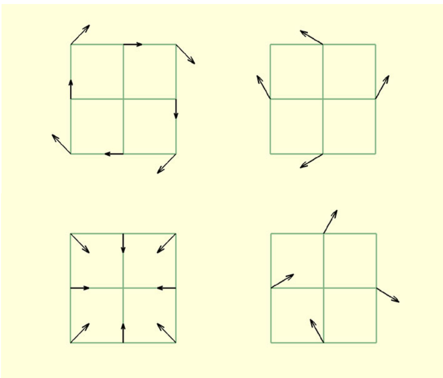




Vector Filter in Frequency Domain

Rotation, Divergence, Saddle Points:

Differ only in phase, not in amplitude!



Gabor Filter

- Short time or windowed Fourier transform
- Optimally localized in both spatial and Fourier domain
- Gabor expansion, wavelets, filter banks

Gabor Filter

- scalar Gabor filter in spatial domain:

$$h(x)=g(x)*e^{-2\pi i\langle x,U\rangle}$$

→ multivector valued Gabor filter in spatial domain:

$$h(x)=g(x)*e^{-2\pi I_k\langle x,U\rangle}$$

Conclusion: Pro/Contra

- Clifford Convolution:
  + unifying notation for scalar / vector fields
  + vector fields: similarity and geometric position
- Pattern matching:
  + robust in terms of noise
  + rotation invariant
  + applicable to irregular grids
  - slow

Conclusion: Pro/Contra

- Fourier transform:
  + convolution thorem, ...
  + mathematical basis for analysis of filter
  + acceleration of convolution via FFTs
  - irregular grids
- Gabor Filter:
  + inherent multiscale approach
  - direct approach gave no big advantages for matching

Future Work

- Irregular Grids
- Analysis of interpolation, smoothing, sampling, derivation and the induced errors

→Filter Design

- Scale space consideration / hierarchical features

Acknowledgments

- FAnToM software development team
  (Tom Bobach, Christoph Garth, David Gruys, Kai Hergenrˆ ther, Nikolai Ivlev, Max Langbein, Martin ÷ hler, Michael Schlemmer, Xavier Tricoche, Thomas Wischgoll)
- Computergraphics Group at TU Kaiserslautern and University of Leipzig
- Data Sets:
  Wolfgang Kollmann, MAE Department, UC Davis
  Markus R¸ tten, DLR Gˆ ttingen

Fourier Transform of Second-Order Tensor Fields

- Convolution using matrix multiplication

$$(f*v)(x)=\int_{E^3} f(y)\,v(x-y)\,dy$$

- Fourier transform

$$F\{f\}(u)=\int f(x)\,\mathrm{e}^{-2\pi i I\langle x,u\rangle}\,dx$$

→ Convolution theorem:

$$F\{h*f\}(u)=F\{h\}(u)\,F\{f\}(u)$$

Fourier Transform of Arbitrary Order Tensor Fields

- Convolution using tensor product

$$(f*v)(x)=\int_{E^3} f(y)\,v(x-y)\,dy$$

- Fourier transform

$$F\{f\}(u)=\int f(x)\,\mathrm{e}^{-2\pi i\langle x,u\rangle}\,dx$$

→ Convolution theorem:

$$F\{h*f\}(u)=F\{h\}(u)\,F\{f\}(u)$$