Clifford Convolution And Pattern Matching On Vector Fields

Julia Ebling Gerik Scheuermann

University of Kaiserslautern
Department of Computer Science
P.O. Box 3049, D-67653 Kaiserslautern
Germany
E-mail: ebling@informatik.uni-kl.de

Abstract

The goal of this paper is to transfer image processing to vector fields and flow visualization by defining a suitable convolution operation. For this, a multiplication of vectors is necessary. Clifford algebra provides such a multiplication of vectors. So we define a Clifford convolution on vector fields with uniform grids. The Clifford convolution works with multivector filter masks. Scalar and vector masks can be easily converted to multivector fields. So, filter masks from image processing on scalar fields can be applied as well as vector and multivector masks. Furthermore, a method for pattern matching with Clifford convolution on vector fields is described. The method is independent of the direction of the structures. This provides an automatic approach to feature detection. The features can be visualized using any known method like glyphs, isosurfaces or streamlines. The features are defined by filter masks instead of analytical properties and thus the approach is more intuitive.

CR Categories: I.4.3 [Image Processing And Computer Vision]: Filtering—; I.4.6 [Image Processing And Computer Vision]: Edge and Feature Detection—; J.2 [Physical Sciences And Engineering]: Aerospace—Engineering;

Keywords: Flow Visualization, Convolution, Pattern Matching

1 Introduction

Classical methods for vector field visualization are hedgehog, flow-probe, streamlines, stream ribbons, stream tubes, spot noise and line integral convolution [7, 8]. Hedgehog and flow-probe are very complex if drawn everywhere, especially in 3D. Streamlines are a good method for visualization if you know where to start them. Spot noise and line integral convolution are good for visualization in 2D, but result in clutter in 3D.

Visualization methods based on feature extraction are topology graphs and vortex core methods [7, 8, 10, 12]. Topologic methods are very good for feature extraction and visualization in 2D but get complex in 3D. Vortex core methods have one problem as there exists no consistent definition of a vortex. They work on different definitions of vortices and can not be applied to other features. Roth [8] defined a parallel operator that covers a lot of the different vortex core definitions.

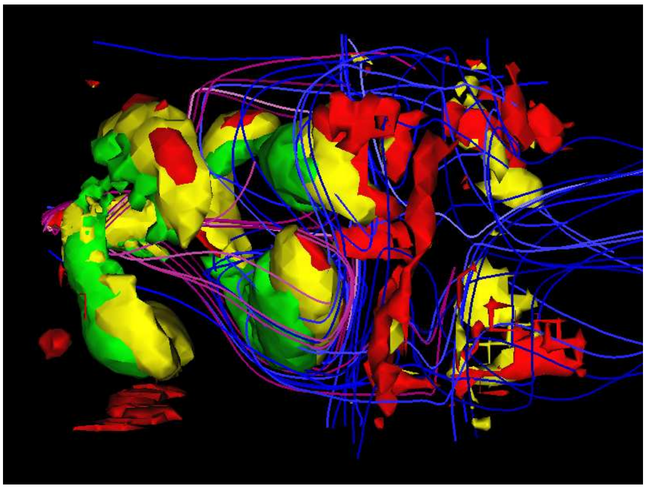

Figure 1: Vortices in a gas furnace chamber

Image processing [5, 6] has a lot of useful tools. Granlund and Knutsson [2] and Heiberg [3] have tried to transfer image processing to vector field visualization. Granlund and Knutsson [2] treat a vector field as several scalar fields. Heiberg [3] defines a scalar convolution on vector fields based on the scalar product of two vectors. He gives an algorithm for pattern matching on vector fields which is discussed in detail in the next section.

One way to convey image processing to vector fields is to define the convolution on vector fields. Therefore, a product of vectors has to be defined as the convolution has to calculate the product of vectors. In this paper, we define a Clifford convolution on vector fields with uniform grids. This is motivated by the need to define a multiplication of vectors. In Clifford algebra, a multiplication of vectors is given. This multiplication supplies us with sinus and cosinus of the angle between two vectors and the plane in which the angle is measured. Thus, Clifford convolution based on this multiplication results in an approximation of sinus and cosinus of the angle between the filter mask and the 0 in the vector field. Furthermore it gives us the plane in which the angle lies, which is very important in 3D. Clifford algebra works with multivectors. A multivector in 3D consists of the sum of a scalar, a 3D vector, a 3D bivector and a trivector. Scalar and vector are as usual. In 3D the unit bivectors can be identified with a planar direction and a limited oriented area and the unit trivector gives the volume spanned by three orthogonal unit vectors building a right hand system. Clifford convolution works on multivector fields. As scalars and vectors can be easily converted to multivectors, this is no problem. It means that scalar filter like gradient or smoothing filter from image processing can

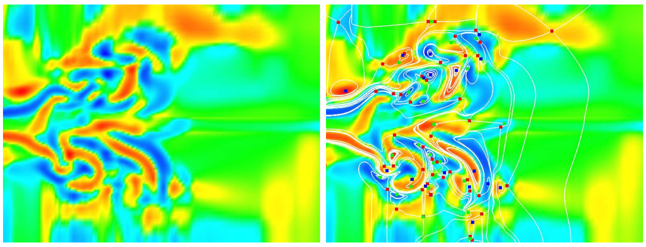

Figure 2: Pattern matching of a 2D vector field with a 8×8 rotation filter mask. On the right, the topology of the original vector field is added.

be applied just like vector filter for pattern matching. The scalar convolution defined by Heiberg [3] is a special case of the Clifford multiplication as shown in section 4.

We use Clifford convolution for pattern matching of the vector field and vector filter masks. Clifford convolution gives approximations of sinus and cosines of the angle between the filter mask and the structure in the vector field and the plane where the angle is measured. Out of these approximations, the direction of the structure can be calculated. The mask is then rotated in the direction of the structure. The scalar convolution of the rotated mask and the vector field is computed as a similarity measure. As the approximations for sinus and cosines depend on the angle between filter mask and the direction of the structure in the field, additional mask directions have to be used. It has been proven that three directions in 2D and six directions in 3D evenly distributed on the circle or the sphere are enough. The algorithm can be accelerated by discretizing the directions and rotating the filter mask only one time in all directions. Rotated structures are recognized with a similarity value of more than 90% in 3D and 95% in 2D using the acceleration with 72 mask directions in 2D and 258 mask directions in 3D.

The similarity values can be visualized using an isosurface algorithm like marching cubes.The points with high similarity values can also be used as starting points for streamlines or related techniques. For our tests, we used data from a turbulent planar jet and the turbulent flow inside a gas furnace chamber. Fig. 1 shows some results from the gas furnace chamber while Fig. 2 gives a first impression on the 2D results.

2 Related Work

The first idea regarding image processing on vector fields is to simply treat a vector field as several scalar fields. Thus, the Fourier transformation can be used. As the scalar fields are not independent, the results depend on the chosen coordinate system. Granlund and Knutson [2] have investigated this approach in 2D. They define lines and edges by a simple neighborhood. This means that the neighborhood can be modeled by a function that varies only in one direction. This function is called simple function. When the points are in a simple neighborhood, they can estimate local orientation, symmetries and curvature. They use these methods to extract texture borders that can be described as a sudden change in a feature vector descriptor field.

Another approach is to define a multiplication of vectors and thus convey the convolution to vector fields. Heiberg [3] defines convolution on vector fields with the scalar product of two vectors:

$$s_n(r) = \int\int\int_{\Omega} U(\xi) * P_n(r-\xi) d\xi$$

where s_n is the filter response, U is the normalized vector field and P_n the filter mask with direction n. This convolution is referred to as scalar convolution in the following. As the scalar product is used, it gives an approximation of the cosines of the angle between the structure in the vector field and the direction of the filter mask.

Furthermore, Heiberg gives an algorithm to compute a similarity measure in 3D independent of the direction of the filters. The vector field is normalized and the filter mask weighted with a rotational symmetric function. The filter mask is rotated in six directions evenly distributed over a hemisphere. The six rotated filters form a filter set. Next, the convolutions of the six filters and the field are computed. Then, with help of a tensor, direction and similarity are calculated out of the squared filter responses and the directions of the six filter masks.

The algorithm can be conveyed to 2D, the necessary formulas can be found in the book of Granlund and Knutson [2]. Because of the square of the filter responses, only the direction and not the orientation of the structure can be computed. The algorithm is robust in terms of noise sensitivity. The disadvantages are that it only works on symmetric filter masks and even not properly on all of them. A structure like the one in Fig. 3 is not recognized at all if rotated in a disadvantageous direction. The algorithm could be adapted to this cases by simply computing the whole algorithm a second time with the filters distributed over the other half of the sphere. Regarding only vortices, this effect doesn't take place. Stretched vortices and those with a curvilinear vortex core line can only be approximated.

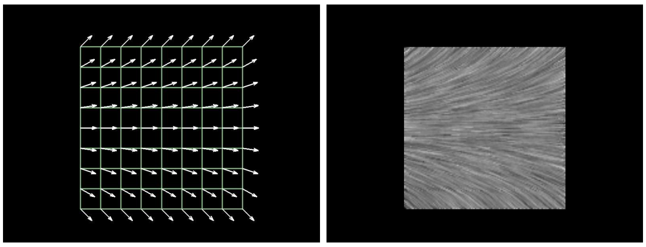

Figure 3: A divergent flow. The algorithm of Heiberg doesn't recognize this structure if it is rotated disadvantageous.

3 Clifford Algebra

For a convolution on vector fields, one has to define a multiplication of vectors. In Clifford algebra [4, 10], a multiplication of vectors is defined. Furthermore, a geometric interpretation of the product of two vectors is given. This product contains sinus and cosines of the angle between the two vectors and the plane where the angle is measured. The rotation of vectors can be easily described and calculated within Clifford algebra, too. In general, Clifford algebra extends the classical description of an Euclidean n-space as a real n-dimensional vector space with scalar product to a real algebra.

3.1 Clifford Algebra in 3D

For the 3-dimensional euclidian vector space, we get a 8-dimensional R-algebra G^3 with the basis $1, e_1, e_2, e_3, e_1e_2, e_2e_3, e_3e_1, e_1e_2e_3$ as a real vector space. The elements of the algebra are called multivectors. The multiplication of multivectors is defined as associative, bilinear and by the equations

$$1e_j = e_j, \qquad j = 1,2,3$$

$$e_je_j = 1, \qquad j = 1,2,3$$

$$e_je_k = -e_ke_j, \qquad j,k = 1,2,3, \; j \neq k$$

Thus, a multiplication of vectors is described, too. The usual vectors $(x, y, z) \in R^3$ are identified with

$$xe_1 + ye_2 + ze_3 \in E^3 \subset G^3.$$

An arbitrary multivector A can be written as

$$A = \alpha + a + i(\beta + b)$$

with $\alpha, \beta \in R,\ a, b \in E^3,\ i = e_1e_2e_3,\ i^2 = -1$.
The reversion is defined as

$$A^+ = \alpha + a - i(\beta + b).$$

The grade projectors $<>_j$: $G^3 \to G^3$ are the maps

$$< A >_0 = \alpha,\ < A >_1 = a,$$

$$< A >_2 = ib,\ < A >_3 = i\beta.$$

For two vectors $a, b \in E^3$ it holds

$$ab = a \wedge b + < a, b > .$$

Furthermore, we have

$$< ab >_0 = < a, b > = \|a\|\,\|b\| \cos \alpha$$

$$\|< ab >_2\| = \|a \wedge b\| = \|a\|\,\|b\| \sin \alpha$$

and $< ab >_2$ corresponds to the normal vector of the plane through a and b as it is the corresponding bivector. Thus, sinus and cosines of the angle between a and b and the angle itself can be calculated. Clifford convolution uses this multiplication to gain an approximation of the direction of the structure in the vector field.

The 2D Clifford algebra is defined analog, see Hestenes [4] and Scheuermann [9].

3.2 Vector derivative

Let $\{g_1, ..., g_n\}$ be a basis of E^n with $g_1 \wedge ... \wedge g_n = \gamma i, \gamma \neq 0$. The reciprocal basis $\{g^1, ..., g^n\}$ is defined as

$$g^k = \frac{(-1)^{k-1}}{\gamma}(g_1 \wedge ... \wedge g_{k-1} \wedge g_{k+1} \wedge ... \wedge g_n i^+).$$

A map $A : E^3 \to G^3$ is called multivector field. Let $b \in E^3$. If the limit

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

exists, it is called directional derivative of A in direction b. It holds:

1. $A_b(r)$ is linear in b, that means $A_{\beta_1 b_1 + \beta_2 b_2} = \beta_1 A_{b_1} + \beta_2 A_{b_2}$
2. $(AB)_b(r) = A_b(r)B(r) + A(r)B_b(r)$

Let $A : E^3 \to G^3$ be a multivector field. The vector derivative of A,

$$\partial A : E^3 \to G^3$$

is defined as

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

where g_1, g_2, g_3 is a basis of E^3 and g^1, g^2, g^3 is the reciprocal basis.

The vector derivative ∂ can be discretized using many different approaches. Let the neighborhood of a vector $v_{i,j}$ in 2D be as follows:

$v_{i-1,j-1}$	$v_{i,j-1}$	$v_{i+1,j-1}$
$v_{i-1,j}$	$v_{i,j}$	$v_{i+1,j}$
$v_{i-1,j+1}$	$v_{i,j+1}$	$v_{i+1,j+1}$

Actually, the values at the nodes of the grid are taken but the above visualization is more demonstrative.

1. The central difference is

   $$\partial v \approx e_1 \frac{v_{i+1,j} - v_{i-1,j}}{2} + e_2 \frac{v_{i,j+1} - v_{i,j-1}}{2}.$$

   This can be interpreted as a mask:

	↓	
→		←
	↑	

2. The forward difference is

   $$\partial v \approx e_1 \frac{v_{i+1,j} - v_{i,j}}{2} + e_2 \frac{v_{i,j+1} - v_{i,j}}{2}.$$

   As a mask:

	↓	
	↗	←

3. The backward difference is

   $$\partial v \approx e_1 \frac{v_{i,j} - v_{i-1,j}}{2} + e_2 \frac{v_{i,j} - v_{i,j-1}}{2}.$$

   As a mask:

→	↙	
	↑	

4. Another possible approach is the following mask which can be interpreted as convergence at the point$(i, j)^T$:

↘	↓	↙
→		←
↗	↑	↖

In 3D, the vector derivative can be discretized in the same way. Central difference and the last mask make the most sense for discretizing the vector derivative. This is motivated by the following definitions of divergence and rotation at point $v_{i,j}$:

1. Def: The gradient $\nabla : G^3 \to R$ is defined by

   $$\nabla(\Phi) = (\frac{\partial}{\partial e_1}\Phi, \frac{\partial}{\partial e_2}\Phi, \frac{\partial}{\partial e_3}\Phi)$$

2. Def: The divergence at point v is defined as

   $$div\, v = \frac{\partial}{\partial e_1}v + \frac{\partial}{\partial e_2}v + \frac{\partial}{\partial e_3}v = < \nabla, v > = \frac{1}{2}(\partial v + v\partial)$$

3. Def: The rotation at point v is defined as

   $$rot\, v = [\nabla, v] = \frac{1}{2}(\partial v - v\partial)$$

Thus, it holds $\partial v = div\, v + irot\, v$ and $v\partial = div\, v - irot\, v$. Furthermore, using the discretized vector derivative as a filter mask and the Clifford convolution defined in section 4, it holds:

$$div\, v = < c_n(v) >_0$$

$$rot\, v = \frac{1}{i} < c_n(v) >_2 .$$

3.2.1 Reflection and rotation

Let $u, v \in R^3$ be unit vectors and $x \in R^3$. We look at the map $U(x) = -uxu$. x can be decomposed in a part $x_\perp$ orthogonal to u and a part $x_{||}$ parallel to u. Thus, it holds

$$U(x) = -uxu = -u(x_{||} + x_\perp)u = -(x_{||} - x_\perp) = x_\perp - x_{||}$$

This means that x is reflected at u. Now, we look at $V(x) = -vU(x)v = vuxuv$. This map can be written as $V(x) = T^+ xT$ with

$$T = uv = < u, v > + u \wedge v = e^{\frac{1}{2}A}$$

$$T^+ = vu = < u, v > - u \wedge v = e^{-\frac{1}{2}A}$$

Decomposing x in a part $x_\perp$ orthogonal to T and a part $x_{||}$ parallel to T, it holds

$$T(x) = T^+ xT = T^+(x_{||} + x_\perp)T = x_\perp + x_{||}e^A.$$

This describes a rotation about $|A|$ in the plane described by A as shown in Fig. 4. With $A = ia, a \in R$, it holds $T = \alpha + i\beta = e^{\frac{1}{2}ia}$, $\alpha = \cos(\frac{1}{2}a)$ and $\beta = \sin(\frac{1}{2}a)$. Every rotation can be described by two reflections. uv and vu are multivectors with scalar and bivector unequal zero. Reflection and rotation in 2D are analog.

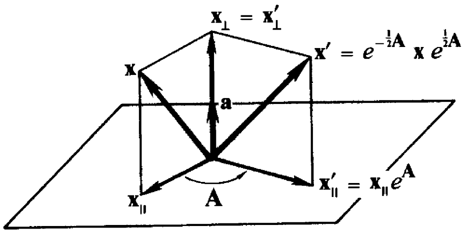
$\mathbf{x}_\perp = \mathbf{x}'_\perp$
$\mathbf{x}' = e^{-\frac{1}{2}\mathbf{A}}\,\mathbf{x}\,e^{\frac{1}{2}\mathbf{A}}$
x
a
$\mathbf{x}_{||}$
A
$\mathbf{x}'_{||} = \mathbf{x}_{||}e^{\mathbf{A}}$

Figure 4: Rotation of a vector using Clifford algebra

4 Clifford convolution

The multiplication of two vectors, as described in section 3, results in sinus and cosines of the angle between the vectors and the plane where the angle is measured. Then, one of the vectors can be calculated out of the product of the two vectors and the second vector. This multiplication can also be regarded as a convolution of a point in the vector field with a 1 × 1 filter mask. Thus, the direction of a structure in the field can be computed out of this convolution and the direction of the filter mask. The Clifford convolution presented here generalizes this concept to arbitrary filter masks. The Clifford multiplication is defined between multivectors. This presents no problem as vectors can be easily converted into multivectors. As scalars can be converted, too, the convolution works also between a vector field and a filter mask with scalar values. That means that gradient and smoothing filters from image processing on scalar fields can be applied, too. Fig. 5 presents some typical vector masks.

4.1 Convolution in scalar fields

In image processing, a filter is a map from one image to another. For a continuous signal $g : R^d \to C$, the convolution with the filter $h : R^d \to C$ is defined by

$$(g * h)(x) = \int_{R^d} h(y)g(x-y)dy$$

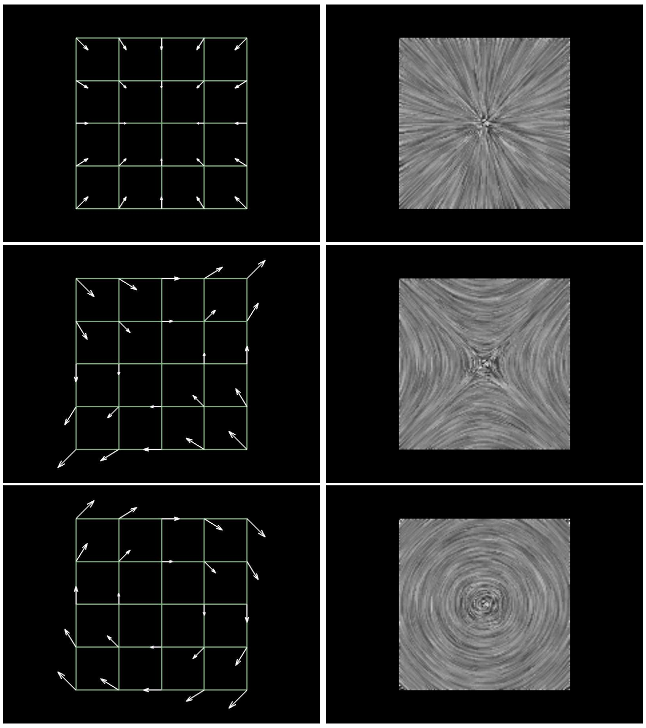

Figure 5: Some filter masks in 2D. On the left, they are visualized using hedgehogs. On the right, line integral convolution on the masks is shown. From top to bottom: convergence, saddle point and rotation.

Every linear and shift invariant filter (LSI filter) can be described as a convolution with a filter mask. A lot of filters for smoothing images and for edge detection are LSI filter. Thus, the convolution is an important operation in image processing [6]. An example of LSI filter are binomial filters used for smoothing. Now, the convolution has to be transfered to vector fields.

4.2 Convolution in vector fields

In a first step, the vector field is normalized. As streamlines are everywhere tangent to the vector field, one can regard pattern matching of a normalized vector field as pattern matching of streamlines.

Heiberg [3] defined a convolution on vector fields with the scalar product of two vectors. Let U be the normalized vector field and P_n the normalized filter mask with direction n.

$$s_n(r) = \int\int\int_\Omega < P_n(\xi), U(r-\xi) > d\xi$$

In the following, this convolution will be referred to as scalar convolution. The sensibility of this filter is proportional to the cosines of the angle between the directions of the filter and the structure within the vector field. This is because the scalar product of two normalized vectors has the cosines of the angle between the two vectors as result. Thus, the filter results in the largest values when the directions of the filter mask and the structure in the field are the same. If the angle between filter mask and structure is $\frac{\pi}{2}$, the filter result is zero. If the angle is π, the filter result has the largest negative value.

Now let U be a normalized multivector field and P_n a normalized multivector filter mask with direction n. The Clifford convolution is defined as

$$c_n(r) = \int\int\int_\Omega P_n(\xi)U(r-\xi)d\xi.$$

For discrete vector fields, the convolutions have to be discretized. In 3D, they are:

$$s_n(i,j,k) = \sum_{s=-r}^{r}\sum_{t=-r}^{r}\sum_{u=-r}^{r} < P_n(s,t,u), U(i-s, j-t, k-u) >$$

and

$$c_n(i,j,k) = \sum_{s=-r}^{r}\sum_{t=-r}^{r}\sum_{u=-r}^{r} P_n(s,t,u)U(i-s, j-t, k-u)$$

with $i, j, k, s, t, u \in Z$. r is the dimension of the grid of the filter mask and the (i, j, k) are grid nodes. A filter mask with different sizes in the three dimensions can be described by a filter mask with the same size in all three directions by simply filling up with zeros. So, this discretized convolutions are general enough. The convolutions in 2D are analog.

As it holds $s_n(r) =< c_n(r) >_0$, the scalar convolution is a special case of the Clifford convolution.

The convolution has to be computed at every point of the grid. But at the border of the vector field, the convolution needs values outside the vector field. So, similar to image processing, there is the problem of boundary values. The solutions for this problem are the same as in image processing with all their advantages and disadvantages [6]. The values can be chosen as follows:

1. Zero. Thus, artificial edges at the border are created.

2. Extrapolated. At the simplest case, one can take the values at the boundary. Another possible extrapolation is to mirror the values at the border. All extrapolations lay too much stress on the bordervalues.

3. Cyclic convolution. The image is assumed to be periodic and the values are set accordingly. This is very much dependent on the chosen display window as most images and vector fields are not periodic.

4. Window function. The values are gradually reduced to zero near the boundary. Some values at the border are lost but otherwise this is the preferred approach.

5 Pattern matching in vector fields

In flow visualization, it is important to find vortices as they use a lot of energy. Sometimes, many vortices are desired if for example two gases shall be mixed or if lift of an airplane is to be supported. Sometimes, vortices are not wanted as they slow the flow and put a lot of stress on the surrounding material. Other interesting features are shock waves, separation lines and attachment lines. Regions with divergence and convergence in the flow are of interest, too. Features like these can be described by small filter masks. Thus, they can be found with pattern matching based on Clifford convolution as described here.

The similarity measure should be independent of the direction of the structure within the vector field and the mask. Otherwise, one has to rotate the filter mask many times and compute the similarities for all the rotated masks. In a last step, it would be necessary to compute the maximum similarity and take the corresponding direction as the direction of the structure. The Clifford convolution gives the direction of the structure in the field directly.

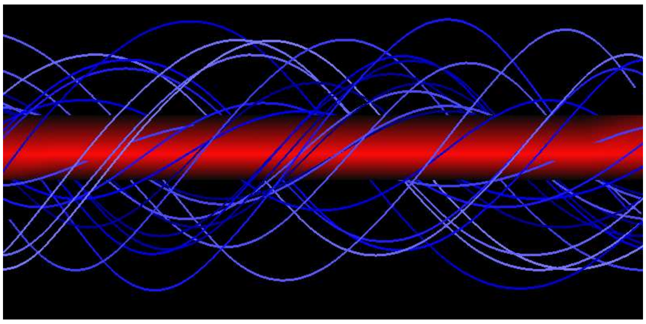

Figure 6: Vortex core found with pattern matching

The algorithm is in many aspects similar to the algorithm of Heiberg [3]. It computes the similarity of a filter mask and structures in the vector field. The similarity is independent of the directions of the structures. The directions of the structures are computed, too. First, vector field and mask are normalized. Then the mask is rotated to generate a filter set. After this, the Clifford convolutions of each of the filter masks and the vector field are computed. This has been analog to Heiberg who uses scalar convolution. Then, the direction of the structure in the vector field is calculated. Heiberg computes similarity and direction with the help of a tensor out of the filter responses. Here, the filter mask is rotated in the direction of the structure and the similarity is computed by another scalar convolution.

5.1 Principal idea

The Clifford multiplication gives sinus and cosines of the angle between two vectors. Furthermore, it gives the plane where the angle is measured. Thus, Clifford convolution gives an approximation of sinus and cosines of the angle between mask and structure and the plane where the angle lies. With normalized field and mask it holds

1. 2D:

   (a) $< c_n(r) >_0 \approx \gamma \cos \alpha_r$

   (b) $< c_n(r) >_2 \approx \gamma \sin \alpha_r$

2. 2D:

   (a) $< c_n(r) >_0 \approx \gamma \cos \alpha_r$

   (b) $\| < c_n(r) >_2 \| \approx \gamma \sin \alpha_r$

   (c) $< c_n(r) >_2$ is the normal vector of the plane of the angle α_r

a_r is the angle between filter mask and structure at point r and $\gamma = \sum_{i,j,k} P(r) \le \sum_{i,j,k} 1$. So one can simply compute the direction of the structure with sinus, cosines and plane of the angle. The mask is rotated in this direction and one scalar convolution for the similarity is computed. When filter mask and structure are equal, the similarity is $l(r) = 1$.

5.2 Filter directions

Unfortunately, praxis is worse than theory. The approximation of the angle between the directions of mask and structure gets more imprecise when the angle is bigger. So additional masks with different directions have to be used. The distribution of the directions is:

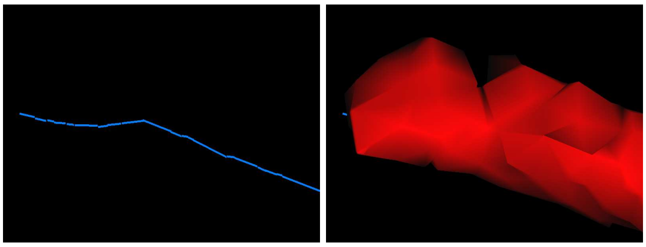

Figure 7: Part of the gas furnace chamber. On the left, the algorithm of Sujudi-Haimes. On the right, pattern matching with a $3 \times 3 \times 3$ rotation mask.

1. 2D: $a = 0.5$ and $b = \frac{\sqrt{3}}{2}$
   (a) $n_1 = (1, 0)^T$
   (b) $n_2 = (-a, b)^T$, that is n_1 rotated $\frac{2\pi}{3}$ counterclockwise.
   (c) $n_2 = (-a, -b)^T$, that is n_1 rotated $\frac{4\pi}{3}$ counterclockwise.
2. 3D:
   (a) $n_1 = (1, 0, 0)^T$, $n_2 = (-1, 0, 0)^T$,
   (b) $n_3 = (0, 1, 0)^T$, $n_4 = (0, -1, 0)^T$,
   (c) $n_5 = (0, 0, 1)^T$, $n_6 = (0, 0, -1)^T$,

The algorithm also works with other directions and other numbers of directions. As usually, one can trade precision for computational speed. The masks are rotated in the desired direction using Clifford algebra and linear interpolation.

Out of the approximations n'_i of the single masks, the direction n' of the structure has to be computed. In 2D, the mask with the smallest angle to the structure is sufficient as criterion. In 3D, the direction is computed analog to the computation of a center of gravity. The directions calculated out of the filter outputs with $< c_{n_i}(r) >_0 \geq 0$ are weighted with the scalars $< c_{n_i}(r) >_0 / \gamma_i$ and summed. The resulting vector is normalized and gives the direction n' of the structure. The algorithm is insensitive to noise as shown in section 7. Outline of the algorithm:

1. Normalize vector field and filter mask
2. Rotate mask to get the filter set and compute γ_i
3. For each grid node of the field:
   (a) Compute Clifford convolution with all masks
   (b) Compute direction n' of the structure
   (c) Rotate mask in direction n'
   (d) Compute scalar convolution of rotated mask and field for the similarity value

5.3 Acceleration

The rotation of the mask at every node of the grid is computational expensive. Therefore, the directions of the filter mask for the final scalar convolution are discretized and all rotated masks are computed only once. The mask with the direction closest to the direction of the structure is taken for the scalar convolution at this point. The direction of the mask and $\gamma_i = \sum_{i,j,k} P(r)$ are computed and saved with the rotated filter mask. In 2D, the mask can be chosen by the angle between the direction and the vector $(1, 0, 0)^T$. In 3D, it is not easy to distribute the directions evenly over the sphere. A subdivision algorithm on the sphere starting with an octaeder gives an approximation. Each triangle of the octaeder is divided in 4 new triangles and the vertices are normalized. With three subdivision steps, one gets 258 directions.

step	# points	# triangles
0	6	8
1	18	32
2	66	128
3	258	512

There are 45 directions in each octant. For the computation of the nearest mask, the right octant has to be identified. Then the scalar product of the direction of the structure and all directions in the octant are computed. The mask with the direction resulting in the smallest scalar product is chosen. As there are only a few points in each octant, a more complex search pattern is not necessary.

6 Filter Design

As mentioned before, the filter masks can be scalar, vector and multivector masks. Actually, there are only multivector filter masks but scalar and vector filter masks can be easily converted to multivector masks. Therefore, the terms scalar and vector filter mask are used to point up which kind of multivector filter mask is used. All masks used in image processing [5, 6] can be applied to vector fields using the Clifford convolution.

6.1 Smoothing and scale spaces

Box and binomial filter for smoothing can be applied to vector fields. The binomial filter is better suited for smoothing than the box filter as the box filter can permute local minima and maxima. Fig. 8 illustrates 2D and 3D binomial filters.

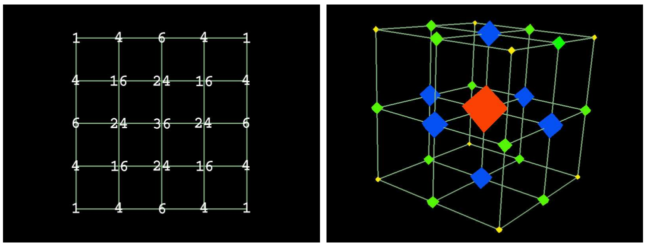

Figure 8: A 5×5 and a $3 \times 3 \times 3$ binomial mask. Only the scalar values are shown as all other values are zero. The weights of the $3 \times 3 \times 3$ mask are as follows: yellow:1, green:2, blue:4 and red:8. The weights of both masks have to be divided by the sum of all values of the mask.

Repeated smoothing of the vector field and subsampling leads to multiresolution pyramids as in image processing. Thereby, a scale space can be constructed. Thus, the behavior of the field can be analyzed at different scales. The scale space can be combined with the pattern matching method. This has the advantage that only small masks have to be used as the computation of large masks is computationally expensive. Furthermore, the scales of features can be figured out and the features divided into features of different scales. Fig. 10 shows the results of applying a rotation mask to a scale space of a turbulent flow data set described in section 7.

6.2 Vector masks for pattern matching

Vector filter masks are used for pattern matching. They can represent features such as vortices, saddlepoints, convergence, divergence and many more. One mighty filter is the small convergence filter in 2D as it computes divergence and rotation of the field as defined in section 3. The divergence is given in the scalar and the rotation in the bivector part of the result. In 3D, it is more complex as the bivector consists of a 3D vector but nonetheless useful. Some 3D vector filter masks are given in Fig. 9.

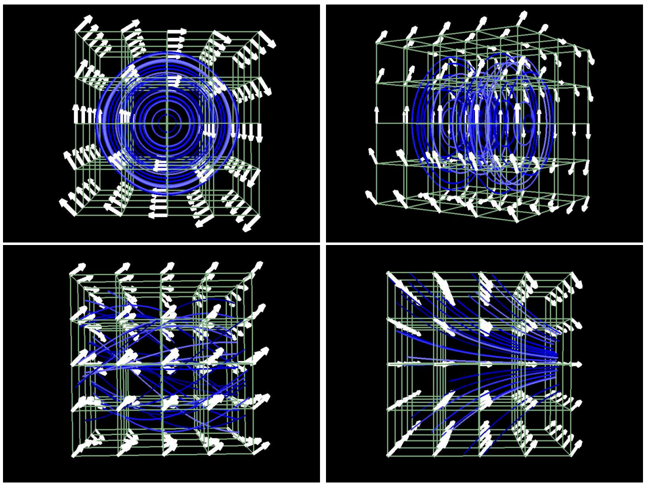

Figure 9: Some filter masks in 3D. The rotation bottom left is approximately what one would describe as a swirl. But it also responds to homogeneous flows. Thus the rotation top left and right is more suited as a similarity mask for rotations as it doesn't respond to homogeneous flows. Bottom right is a divergent flow mask.

7 Results

The main goal of this paper is to convey image processing to vector fields on uniform grids. All LSI filter can be described by scalar filter masks. Scalar filter masks can be easily converted to multivector filter masks. Thus, they can be used with the Clifford convolution defined in this paper and applied to vector fields. As the LSI filter are very important in image processing, most of the methods from image processing can now be applied to vector fields on uniform grids. With smoothing filters like the binomial filter, scale spaces can be build. This is very helpful for pattern matching.

Furthermore, we want to have an algorithm for pattern matching on vector fields that is robust and insensitive to noise. It should also work on all kinds of filter masks. It has to be at least as precise as the algorithm of Heiberg [3]. The computed directions should be accurate and it should recognize a rotated copy of itself with a similarity value of 90%-100%.

For these reasons, out first tests consisted in rotating the given mask by all possible angles (steps different from the rotations used to speed the algorithm). The second class of tests added white noise to the field to check for robustness.

The algorithm for pattern matching computes a similarity measure and the direction of the structure. With the described acceleration, it recognizes rotated structures with a similarity value of about 100% in 2D and 96%-100% in 3D. The filter masks can be arbitrary. Curved structures and not symmetric ones present no problem. The algorithm works local and is insensitive to noise. Rotating every vector in a field randomly up to 18° reduces the similarity values about 3%. Rotating every vector in a field randomly up to 32° results in similarity values of the same structures of still more then 90%. Compared to the adjusted algorithm of Heiberg, the algorithm presented here is only half as fast because of the computation of the Clifford convolution. It is more precise and it works on more filter masks. The similarity values of the field can be visualized using a marching cubes algorithm. Otherwise they can be used to determine starting points for streamlines or the like.

After these rather academic tests, we have chosen two test data sets from real applications. The first data set is a turbulent swirling jet entering a fluid at rest. The simulation considers a cylinder and assumes rotational symmetry, so that a planar cut along the axis of the cylinder can be used as a domain. The domain is discretized by a 124×101 rectilinear grid with smaller rectangles towards the axis of the cylinder. Since a lot of small and large scale vortices are present in the flow, a discrete numerical simulation (DNS) using a higher order finite difference scheme is used to solve the incompressible Navier-Stokes equations. The results of the Clifford convolution are shown in Fig. 2 including an overlay with the topological structure. A scale space example is shown in Fig. 10.

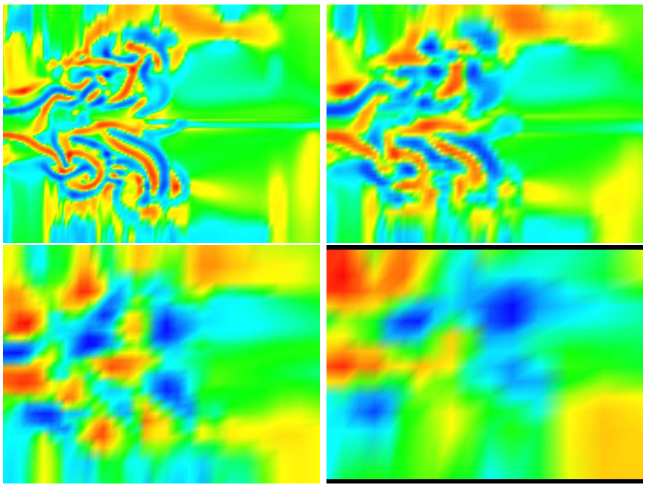

Figure 10: A scale space build by smoothing with a 5×5 binomial filter mask. Every second value was taken in the subsampling. On the resulting fields, pattern matching with a 5×5 rotation mask was applied. On the top left the original image filtered with the mask is shown.

Our second application is a gas furnace chamber as it is used for heating a house. In the top left picture in Fig. 11, the swirling gas enters the chamber in the center of the left face while the air enters from 9 openings on the top and 9 openings on the bottom, so that the combustion takes place in the center area of the chamber. The products of the combustion leave the chamber on the right. The simulation solves compressible Navier-Stokes equations using a turbulent model applied on a irregular grid consisting of 174341 tetrahedra with 32440 vertices. The flow is highly turbulent and exhibits a lot of different scale vortices. Since our method works only on rectilinear grids so far, we used a local resampling around each vertex. Since the cells differ substantially in size, we took as distance in the local resampling the length of the shortest edge at the vertex. The results of a $3 \times 3 \times 3$, a $5 \times 5 \times 5$ and a $8 \times 8 \times 8$ rotation mask are shown in Fig. 11. Obviously, the result of the $3 \times 3 \times 3$ mask (given in red) gives a larger region of potential vortices than the $5 \times 5 \times 5$ mask (yellow) and the $8 \times 8 \times 8$ mask (green). As usual, larger masks mean more computation, smaller similarity and, if even larger than here, less precise location of the feature.

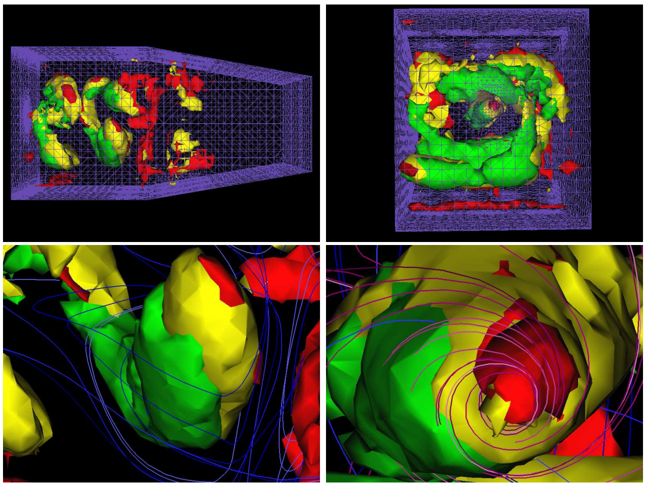

Figure 11: Parts of a gas furnace chamber. Pattern matching of the vector field of the chamber and a $3\times3\times3$ (red), a $5\times5\times5$ (yellow) and a $8\times8\times8$ (green) rotation mask was computed. The similarities are visualized using a marching tetrahedra with an isovalue of 0.5. On the right, the main inflow of the gas is shown. At the bottom, some streamlines are drawn additionally.

8 Conclusion and future work

We have presented a way to convey image processing to vector fields on uniform grids. Tools from image processing such as smoothing filter and scale spaces have been successfully applied to vector fields. Thus, a broad new approach to vector field visualization is given.

Furthermore, we have presented a robust algorithm for pattern matching on these vector fields. We have shown that the algorithm for pattern matching of Heiberg [3] has some disadvantages and is limited in the choice of the filter masks. Our pattern matching algorithm is not limited to any masks and naturally includes scalar masks like binomial filter. Furthermore, it is insensitive to noise and at least as precise as the algorithm of Heiberg.

Future work will include the search for a really convincing adaption to irregular grids since our method described for the gas furnace chamber should be improved. Another important issue will be the design of additional filters that are more directed towards the features in question.

Acknowledgments

First of all, we like to thank the members of the FAnToM development team at University of Kaiserslautern, especially Tom Bobach, Christoph Garth, David Gruys, Kai Hergenröther, Max Langbein and Xavier Tricoche, for their help with programming and production of the pictures. We further acknowledge the fruitful discussions in the whole computer graphics group at Kaiserslautern. We thank Prof. Kollmann, MAE department, University of California at Davis, for producing the swirling jet data set. Further thanks go to Markus Rüttger, DLR Göttingen, for providing the gas furnace chamber data set.

References

[1] Dirk Bauer and Ronald Peikert, Vortex Tracing In Scale-Space, in Data Visualisation 2002, D. Ebert, P. Brunet, I. Navazo (eds.), Association for Computational Machinery, New York, USA, 2002, pp. 233-240

[2] G. Granlund and H. Knutsson, Signal Processing For Computer Vision. Link¨oping: Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[3] Einar Brandt Heiberg, Automated Feature Detection In Multidimensional Images. Link¨oping Studies in Science and Technology, Thesis No. 909, Department of Biomedical Engineering & Medicine and Care, Link¨oping University, Sweden, 2001.

[4] David Hestenes, New Foundations For Classical Mechanics, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993.

[5] Anil K. Jain, Fundamentals Of Digital Image Processing, Prentice Hall, Englewoods Cliffs, NJ, USA, 1989.

[6] Bernd J¨ahne, Digitale Bildverarbeitung, Springer Verlag, Berlin, Germany, 2002.

[7] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, H. Doleisch, Feature Extraction And Visualization Of Flow Fields, in Eurographics 2002 State of the Art Reports, D. Fellner, R. Scopigno (eds.), The Eurographics Association, Saarbr¨ucken, Germany, 2002, pp. 69-100.

[8] Martin Roth, Automatic Extraction Of Vortex Core Lines And Other Line Type Features For Scientific Visualization, Diss. ETH No. 13673, Hartung-Gorre Verlag Konstanz, 2000.

[9] Gerik Scheuermann, Hans Hagen, Heinz Kr¨uger, Martin Menzel, Alyn Rockwood, Visualization of Higher Order Singularities in Vector Fields, In Proceedings of IEEE Visualization, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 67-74.

[10] Gerik Scheuermann, Topological Vector Field Visualization With Clifford Algebra, Dissertation, Fachbereich Informatik, University of Kaiserslautern, Germany, 1999.

[11] Will Schroeder, Ken Martin, Bill Lorensen, The Visualisation Toolkit 2nd Edition, Prentice Hall, New Jersey, USA, 1998.

[12] Xavier Tricoche, Gerik Scheuermann, Hans Hagen, A Topology Simplification Method For 2D Vector Fields, in Proceedings of IEEE Visualisation, D. Ebert, M. Gross, B. Hamann (eds.), IEEE Computer Society Press, Los Alamitos CA, 2000, pp 359-366.

Appendix

Clifford multiplication

Since Clifford algebra might be considered complicated, here is a hint on the implementation of the Clifford product. Let

$$A = a_0 + a_1e_1 + a_2e_2 + a_3e_1e_2$$

and

$$B = b_0 + b_1e_1 + b_2e_2 + b_3e_1e_2$$

be multivectors in 2D. Then a_0, b_0 are scalars, $a_1e_1, a_2e_2, b_1e_1, b_2e_2$ are vectors and $a_3e_1e_2, b_3e_1e_2$ are bivectors. The Product AB can be computed as follows:

$$\begin{aligned} AB &= a_0B + a_1e_1B + a_2e_2B + a_3e_1e_2B \\ &= a_0b_0 + a_0b_1e_1 + a_0b_2e_2 + a_0b_3e_1e_2 \\ &\quad +a_1e_1b_0 + a_1e_1b_1e_1 + a_1e_1b_2e_2 + a_1e_1b_3e_1e_2 \\ &\quad +a_2e_2b_0 + a_2e_2b_1e_1 + a_2e_2b_2e_2 + a_2e_2b_3e_1e_2 \\ &\quad +a_3e_1e_2b_0 + a_3e_1e_2b_1e_1 + a_3e_1e_2b_2e_2 + a_3e_1e_2b_3e_1e_2 \\ &= a_0b_0 + a_1b_1 + a_2b_2 + a_3b_3 \\ &\quad +(a_0b_1 + a_1b_0 - a_2b_3 + a_3b_2)e_1 \\ &\quad +(a_0b_2 + a_2b_0 + a_1b_3 - a_3b_1)e_2 \\ &\quad +(a_0b_3 + a_3b_0 + a_1b_2 - a_2b_1)e_1e_2 \end{aligned}$$

as it holds

$$e_1e_1 = e_2e_2 = 1$$

and

$$e_2e_1 = -e_1e_2$$

as defined in section 3. The multiplication in 3D can be computed in the same way.