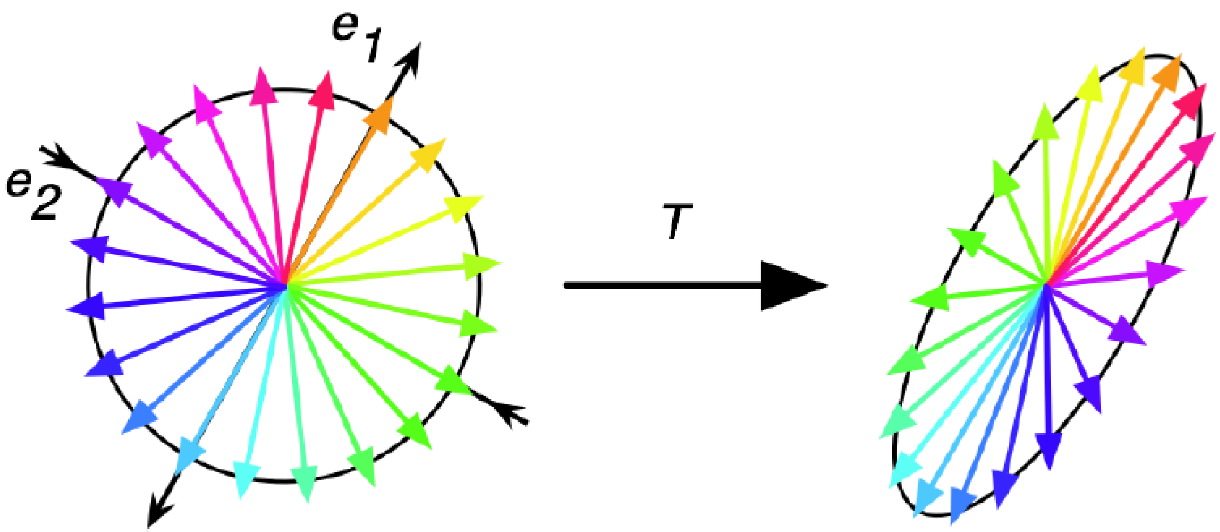
Tensor Field Visualization I

Recall: One Simple Example of Tensor



Tensor describes certain ***<u>higher-order</u>*** property of the space that both scalar and vector-valued cannot

Definition

- A **second-order tensor** *T* is defined as a bilinear function from two copies of a vector space *V* into the space of real numbers

$$T\colon V \times V \to R$$

- Or: a second-order tensor *T* as linear operator that maps any vector $v \in V$ onto another vector $w \in V$

$$T\colon V \to V$$

Definition

- A **second-order tensor** *T* is defined as a bilinear function from two copies of a vector space *V* into the space of real numbers

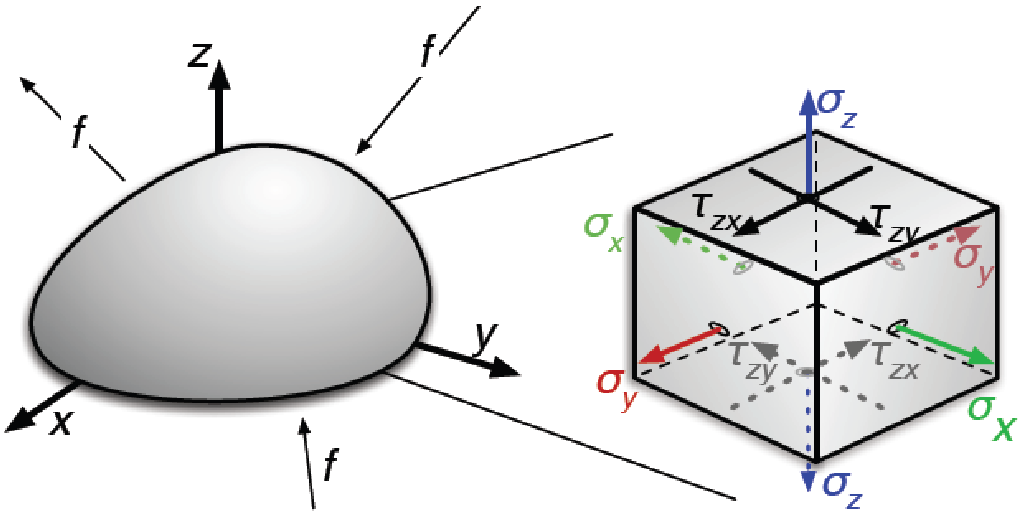
$$T\!: V \times V \rightarrow R$$

- Or: a second-order tensor *T* as linear operator that maps any vector $v \in V$ onto another vector $w \in V$

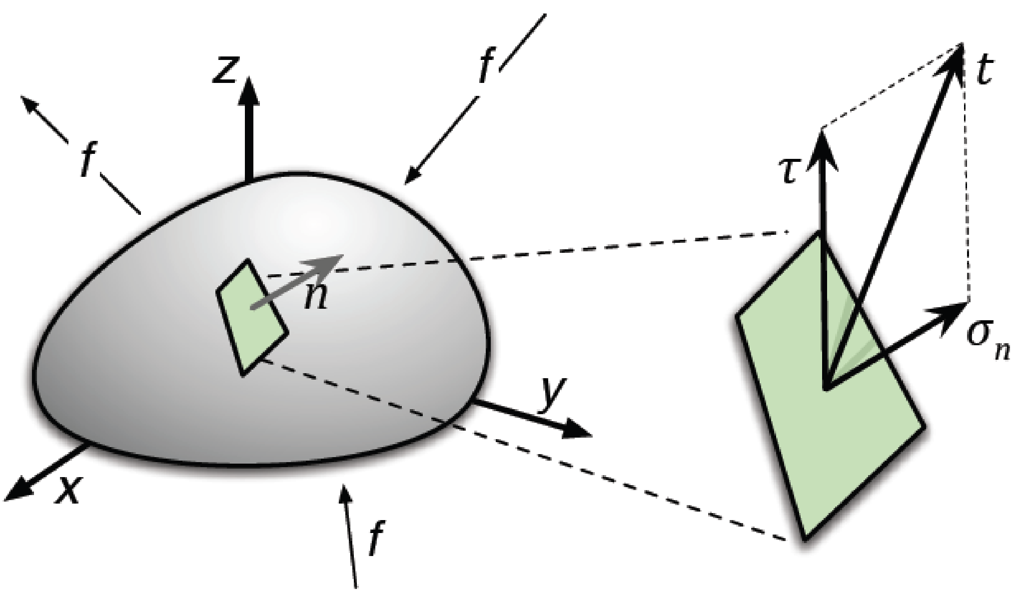
$$T\!: V \rightarrow V$$

- <u>*The definition of a tensor as a linear operator (the second definition) is prevalent in physics*</u>.

Tensors in Mechanical Engineering



(a)



Stress tensors describe internal forces or stresses that act within deformable bodies as reaction to external forces

(a) External forces f are applied to a deformable body. Reacting forces are described by a three-dimensional stress tensor that is composed of three normal stresses *s* and three shear stresses τ.

(b) Given a surface normal n of some cutting plane, the stress tensor maps *n* to the traction vector *t*, which describes the internal forces that act on this plane (normal and shear stresses).

Tensor Properties

- **Symmetric Tensors**. A tensor S is called symmetric if it is invariant under permutations of its arguments
$$S(v,w) = S(w,v) \quad \forall v,w \in V$$

- **Antisymmetric Tensors**. A tensor A is called antisymmetric or skew-symmetric if the sign flips when two adjacent arguments are exchanged
$$A(v,w) = -A(w,v) \quad \forall v,w \in V$$

- **Traceless Tensors**. Tensors T with zero trace, i.e. $tr(T) \ = \sum_{i=0}^{n-1} T_{ii}$, are called traceless.

Tensor Properties

- **Positive (Semi-) Definite Tensors**. A tensor T is called positive (semi-) definite if for all v

$$T(v, v) > (\geq) 0$$

Their eigenvalues and their determinant are greater than zero.

- **Negative (Semi-) Definite Tensors**. A tensor T is called negative (semi-) definite if for all v

$$T(\;v\;\;,)v\quad (<\leq) 0$$

their eigenvalues are smaller than (smaller than or equal to) zero

- **Indefinite Tensors**. Each tensor that is neither positive definite nor negative definite is indefinite.

Tensor Decomposition

- Symmetric/Antisymmetric
- Stretch/Rotation
- Isotropic/Anisotropic
- Shape/Orientation
- Asymmetric tensor decomposition
- Others (e.g., eigen-modes)

What are the Characteristics of Tensors?

or what can be visualized…

- **Scalar** related
  - Components (or individual entries)
  - Determinant
  - Trace
  - Eigen-values
- **Vector** related
  - Eigen-vector fields

The Data: Second-order Tensor Fields

- In visualization, usually not only a single tensor but a whole tensor field is of interest.
- It can be considered as a function which assigns a tensor at any given position in space.

From now on, we consider only second-order tensor which can be represented in the form of matrices.

Visualizing tensor fields is challenging

- Hard to achieve intuitive visualization
- Highly application-dependent

Context

- Multi-variate nature makes it challenging
- Difficulty in preserving tensor characteristics during interpolation
- Perception issue: clutter and occlusion

Technical

Techniques for tensor field visualization

Direct methods

Geometric-based methods

Texture-based methods

Feature-based methods

Techniques for tensor field visualization

Direct methods

Geometric-based methods

Texture-based methods

Feature-based methods

Scalar related
- Components (or individual entries)
- Determinant
- Trace
- <u>Eigen-values</u>

Vector related
- <u>Eigen-vector fields</u>

Techniques for tensor field visualization

Direct methods

Geometric-based methods

Texture-based methods

Feature-based methods

Scalar related

- Components (or individual entries)
- Determinant
- Trace
- <u>Eigen-values</u>

Vector related

- <u>Eigen-vector fields</u>

Techniques for tensor field visualization

Direct methods

Geometric-based methods

Texture-based methods

Feature-based methods

Scalar related

- Components (or individual entries)
- Determinant
- Trace
- <u>Eigen-values</u>

Vector related

- <u>Eigen-vector fields</u>

Techniques for tensor field visualization

Direct methods

Geometric-based methods

Texture-based methods

Feature-based methods

Scalar related

- Components (or individual entries)
- Determinant
- Trace
- <u>Eigen-values</u>

Vector related

- <u>Eigen-vector fields</u>

DIRECT METHODS:
PSEUDO-COLORS AND GLYPHS

Pseudo-Colors

- Any derived scalar properties of the tensor can be mapped to color plots (or DVR in 3D)

Pseudo-Colors

- Any derived scalar properties of the tensor can be mapped to color plots (or DVR in 3D)
- Assume a tensor T is defined at each vertex
  - Components (or entries) T_{ij}
  - Tensor magnitude

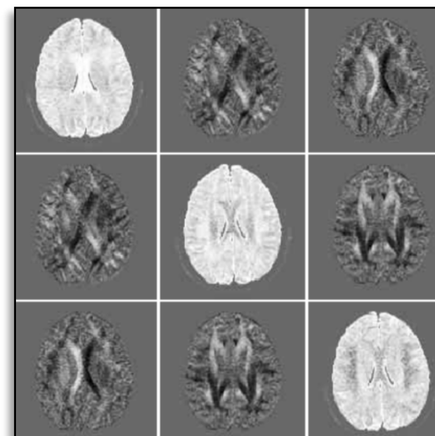
$$\|T\|_F = \sqrt{\frac{1}{2}\sum T_{ij}^2}$$

  - Trace, $tr(T) = \sum T_{ii}$. If T is the Jacobian of a flow field, this tells how much divergence it has.

Pseudo-Colors

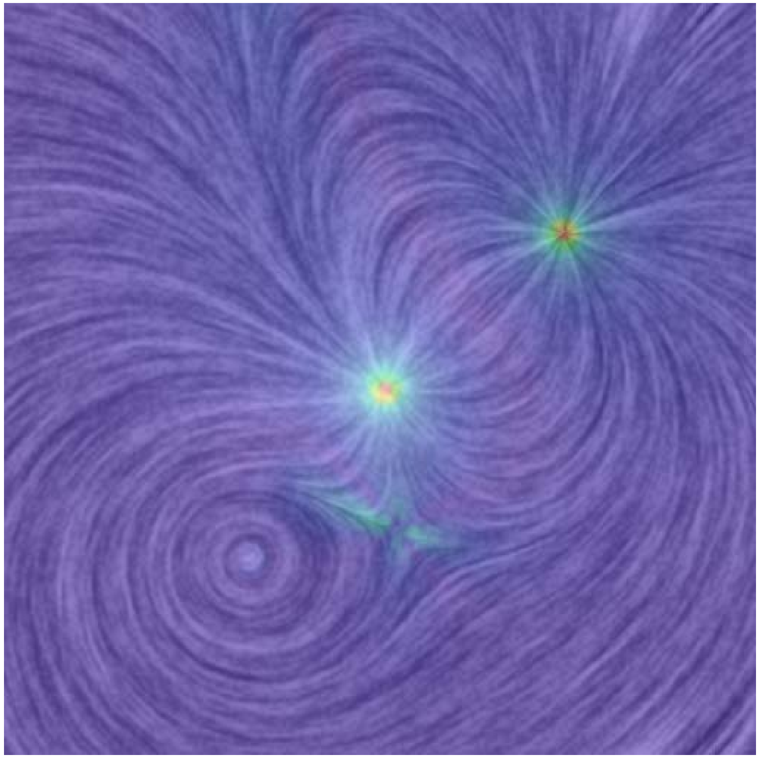
- Any derived scalar properties of the tensor can be mapped to color plots (or DVR in 3D)
- Assume a tensor T is defined at each vertex
  - Components (or entries) T_{ij}
  - Tensor magnitude

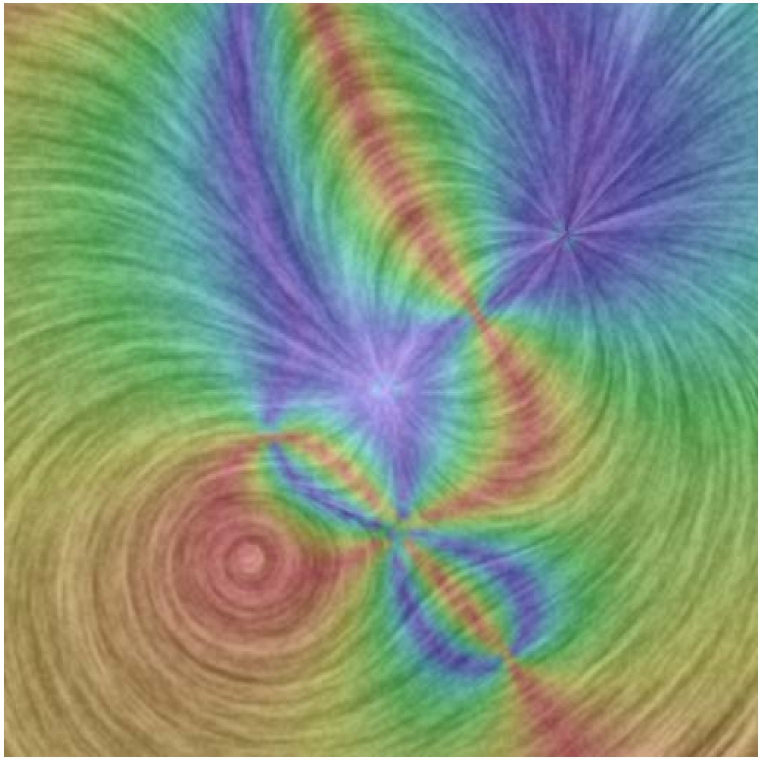
$$\|T\|_F = \sqrt{\frac{1}{2}\sum T_{ij}^2}$$



  - Trace, $tr(T) = \sum T_{ii}$. If T is the Jacobian of a flow field, this tells how much divergence it has.

Pseudo-Colors





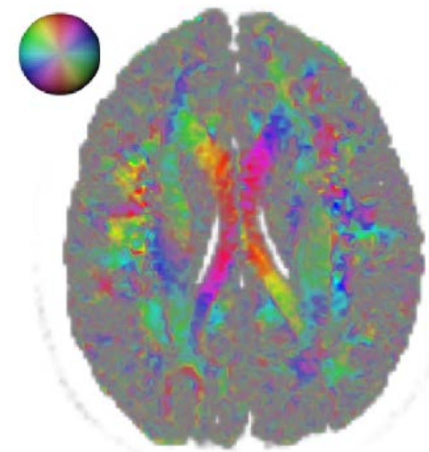
Divergence and curl of a vector field

Pseudo-Colors

- Scalar properties of tensor (continued)
  - Determinant
  - Eigen-values
    - $Te = \lambda e$
    - Can be used to compute the determinant for diagonalizable tensor

Pseudo-Colors

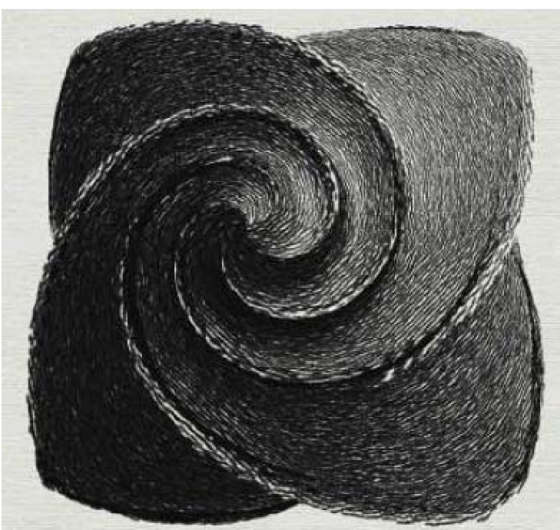
- Scalar properties of tensor (continued)
  - Determinant
  - Eigen-values
    - $Te = \lambda e$
    - Can be used to compute the determinant for diagonalizable tensor
    - More importantly, it can be used to study the **anisotropy** of the *symmetric tensor*, e.g., diffusion tensor used in medical applications

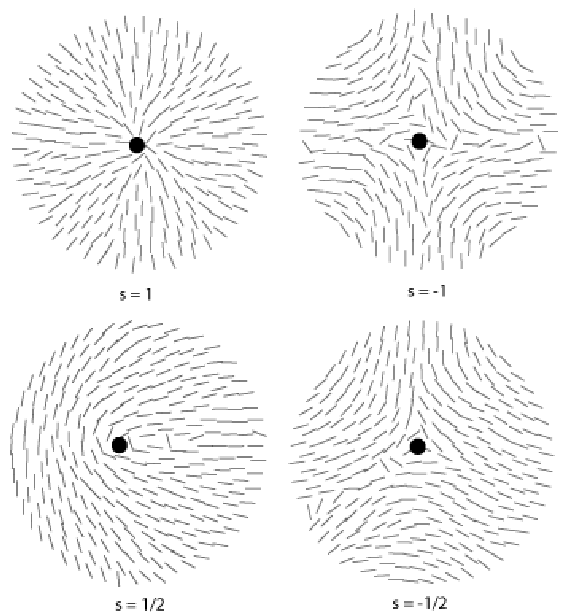


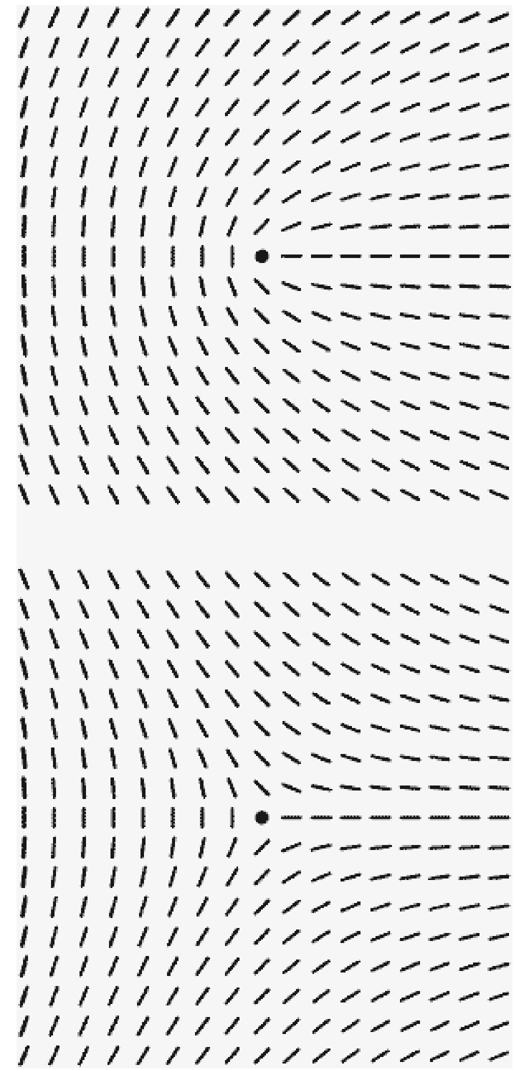
Anisotropy direction mapped to hue + strength mapped to saturation

Line plots

Line field visualization for vector valued properties (e.g., eigen-vectors) for **s.p.d.** tensor fields







GLYPH-BASED METHODS

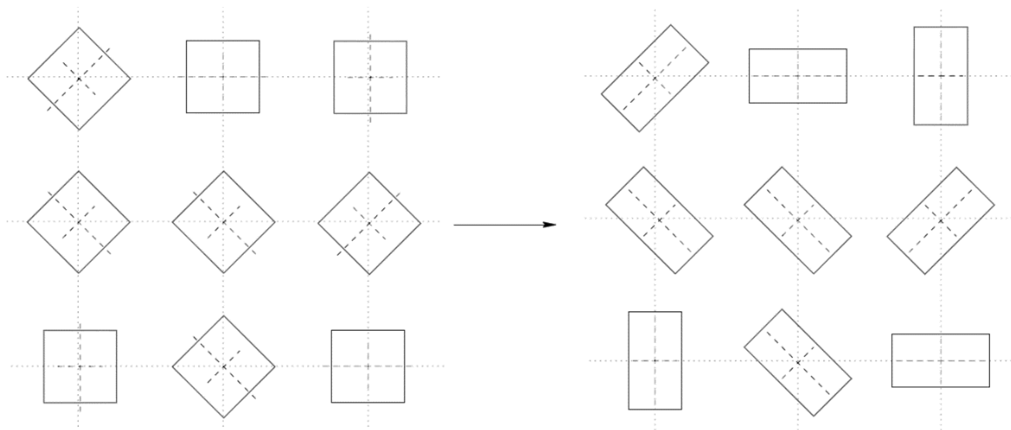
GLYPH DESIGN

A **glyph** is the visual representation of a piece of data where the attributes of **a graphical entity (geometry)** are dictated by one or more attributes of a data record. [From Wikipedia]

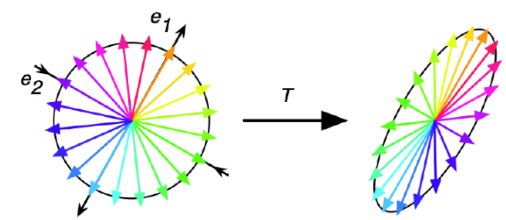
Glyphs for Tensors

- 2D/3D shapes: better visualization of the **local property** of tensor, such as **anisotropy**

2D



Orientation **Shape -> anisotropy**

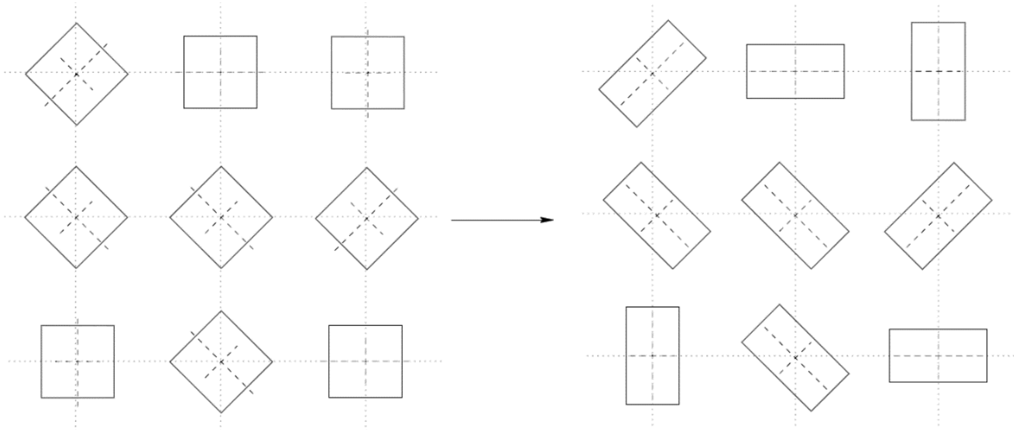


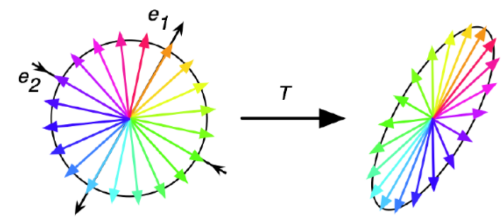
The glyphs for visualizing the anisotropy of a **symmetric** tensor

Glyphs for Tensors

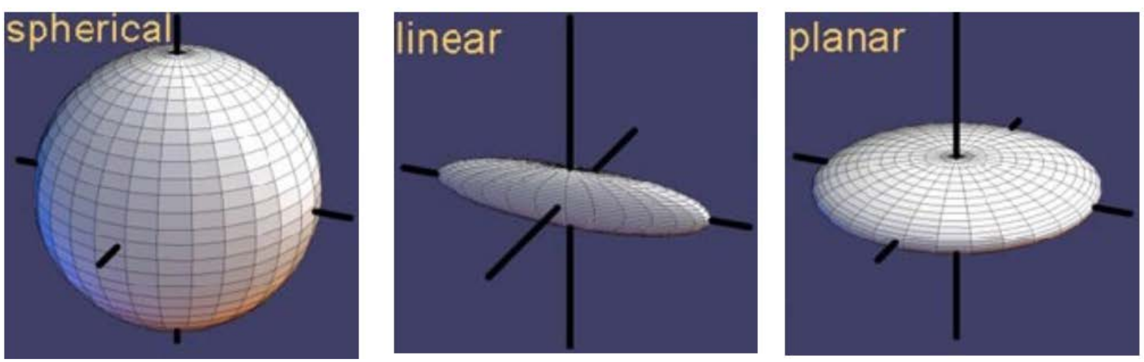
- 2D/3D shapes: better visualization of the **local property** of tensor, such as **anisotropy**

2D



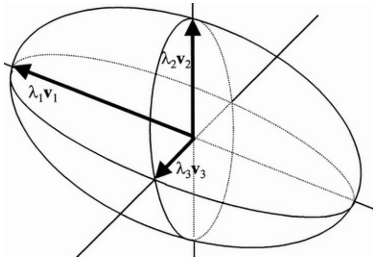


3D



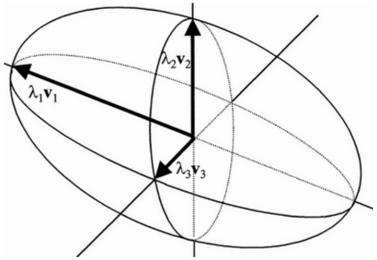
The glyphs for visualizing the anisotropy of a **symmetric** tensor

Glyphs for Tensors



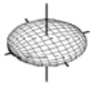
Consider **symmetric tensors** at this moment. They have <u>real</u> eigenvalues and <u>orthogonal</u> eigenvectors. Therefore, they can be intuitively represented as ellipsoids.

Glyphs for Tensors



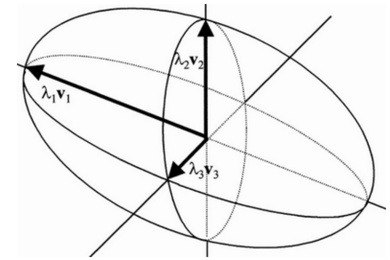
Consider **symmetric tensors** at this moment. They have real eigenvalues and orthogonal eigenvectors. Therefore, they can be intuitively represented as ellipsoids.

Three types of anisotropy:

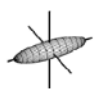
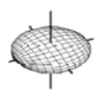
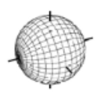
- linear anisotropy 
- planar anisotropy 
- isotropy (spherical) 

Glyphs for Tensors

Consider **symmetric tensors** at this moment. They have <u>real</u> eigenvalues and <u>orthogonal</u> eigenvectors. Therefore, they can be intuitively represented as ellipsoids.



Three types of anisotropy:

- linear anisotropy 
- planar anisotropy 
- isotropy (spherical) 

Anisotropy measure

[Westin et al., 97]

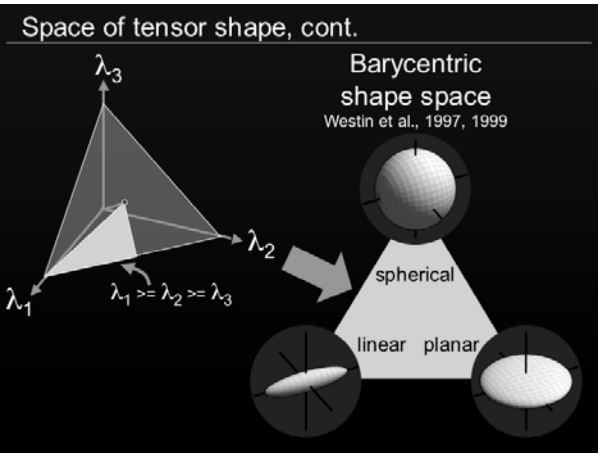
$$c_l = {(\lambda_1 - \lambda_2)}\Big/{(\lambda_1 + \lambda_2 + \lambda_3)}$$

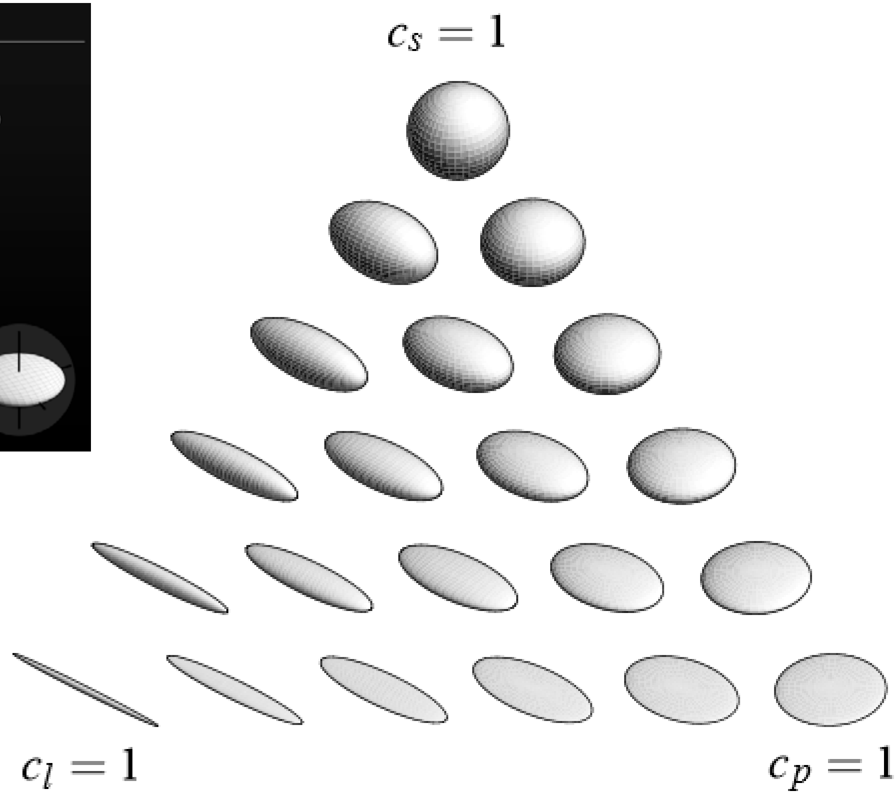
$$c_p = {2(\lambda_2 - \lambda_3)}\Big/{(\lambda_1 + \lambda_2 + \lambda_3)}$$

$$c_s = {3\lambda_3}\Big/{(\lambda_1 + \lambda_2 + \lambda_3)}$$

$$\boldsymbol{\lambda_1 \geq \lambda_2 \geq \lambda_3}$$

Ellipsoidal glyphs provide nice symmetry and smoothness

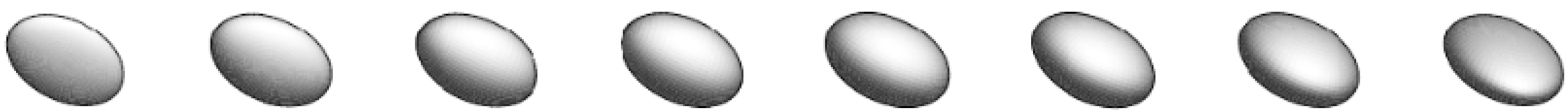




Problem of **ellipsoidal** glyphs:

- Shape is poorly recognized in projected view

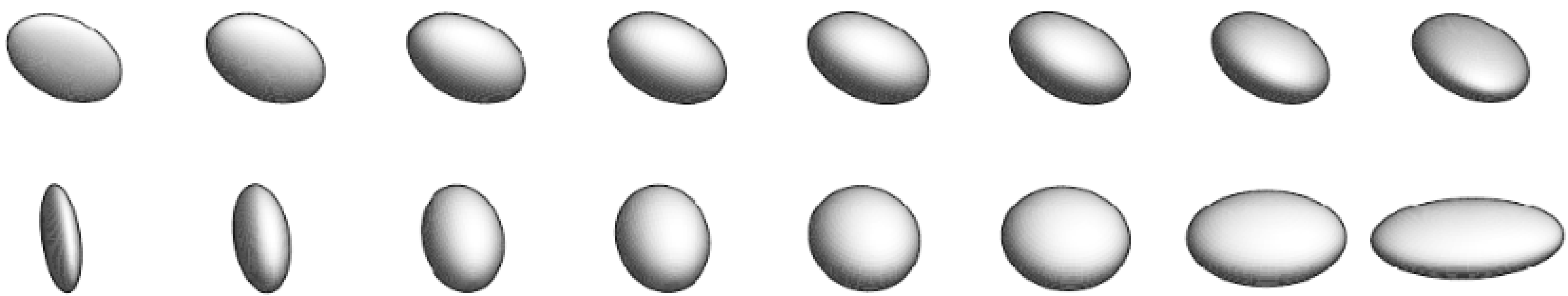
8 **different** ellipsoids



Problem of **ellipsoidal** glyphs:

- Shape is poorly recognized in projected view

8 **different** ellipsoids but in two different views (two rows)



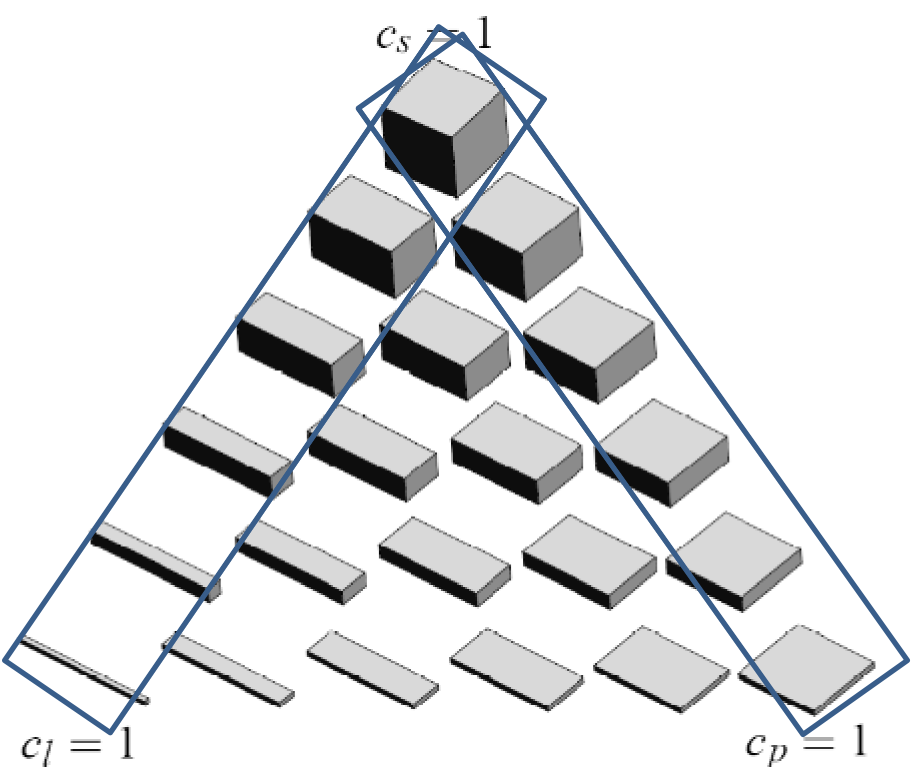
Two Types of Glyphs for the Barycentric Space

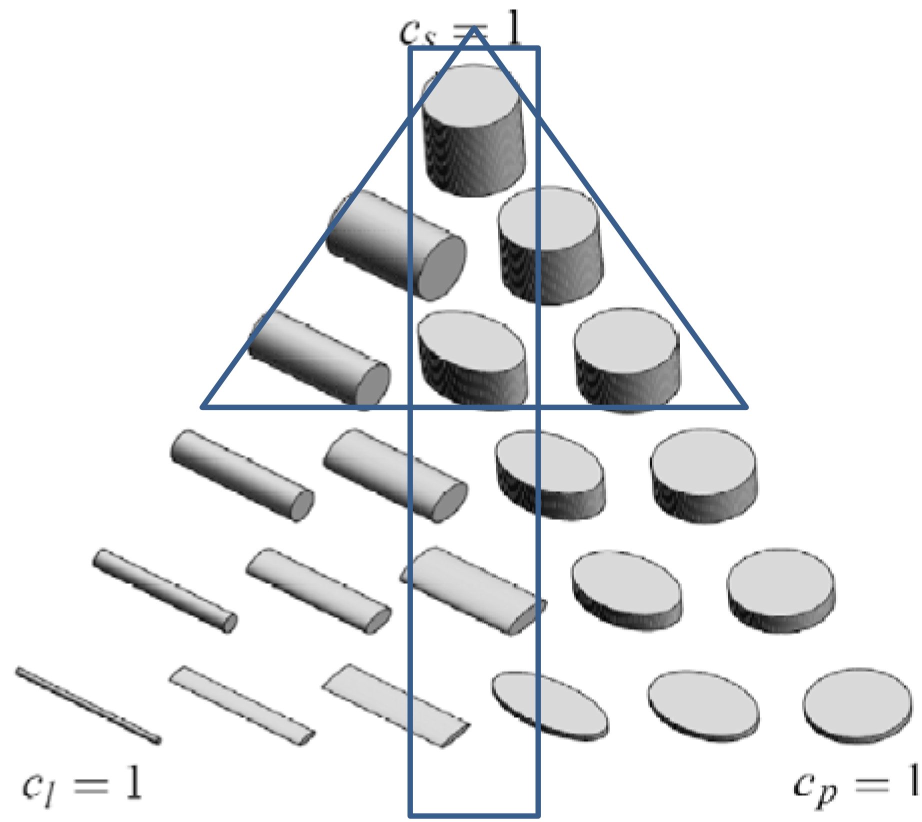
Problem of **cuboid** glyphs

- Missing symmetry
- Doesn't give the sense of smooth

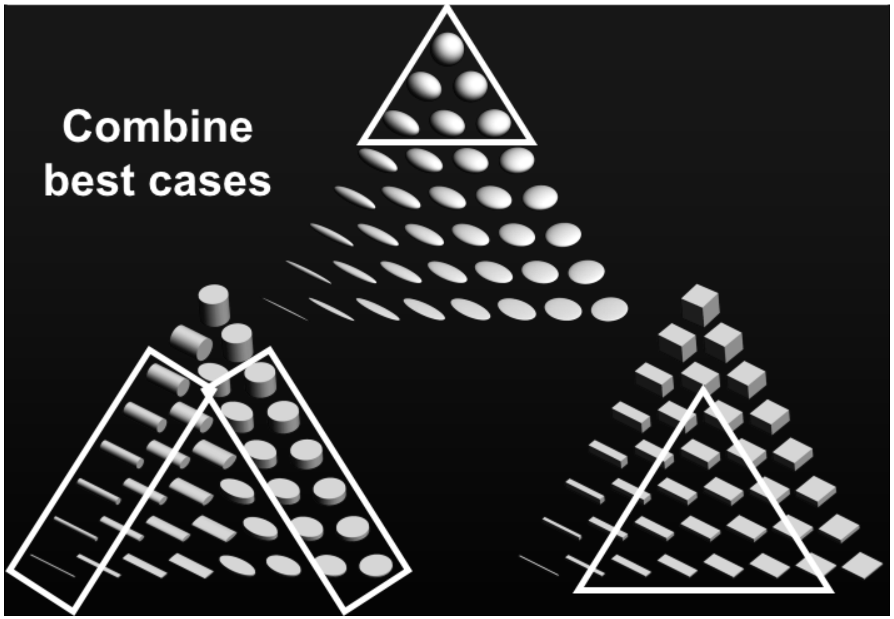
Problem of **cylinder** glyphs

- Seam at $c_l = c_p$
- Losing symmetric close to $c_s = 1$





Combining advantages: **superquadrics**
Superquadrics with Z as primary axis

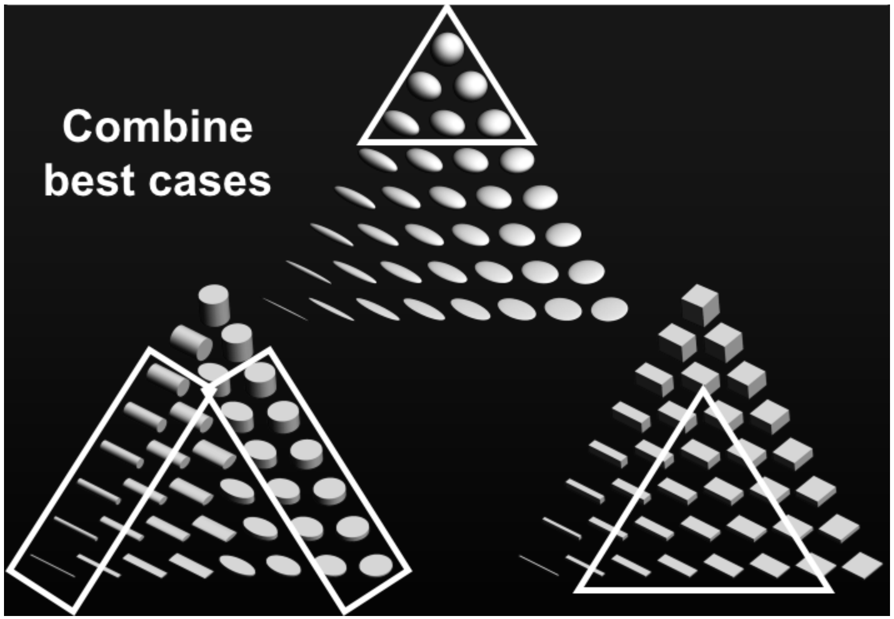


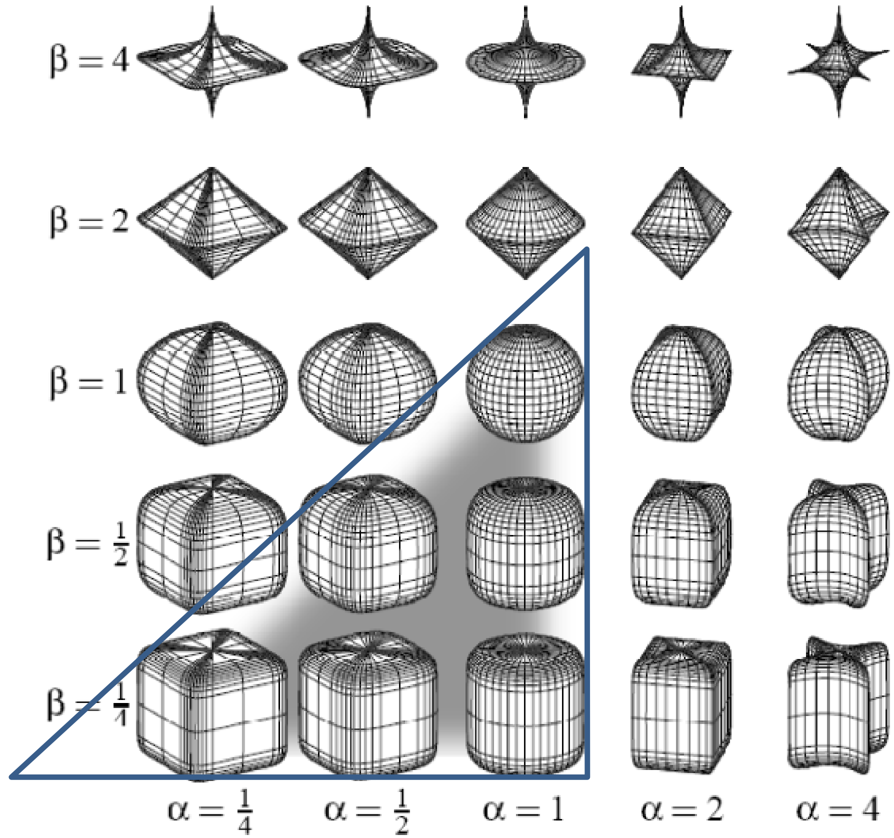
Cylinders for the linear and planar cases, spheres for the spherical case, and cuboids for intermediate cases with smooth blending in between.

The general strategy is that <u>the edge on the glyph surface signifies the anisotropy</u>. When two eigenvalues are equal, the indeterminacy of the eigen-vector can be conveyed with a circular glyph cross-section.

Combining advantages: **superquadrics**

Superquadrics with Z as primary axis





$$q_z(\theta, \emptyset) = \begin{bmatrix} \cos^{\alpha}\theta \sin^{\beta}\emptyset \\ \sin^{\alpha}\theta \sin^{\beta}\emptyset \\ \cos^{\beta}\emptyset \end{bmatrix}$$

$$0 \leq \theta \leq 2\pi, 0 \leq \emptyset \leq 2\pi$$

Similarly, one can define $q_x(\theta, \emptyset)$

Barr, 1981

Superquadrics for some pairs (α, β)

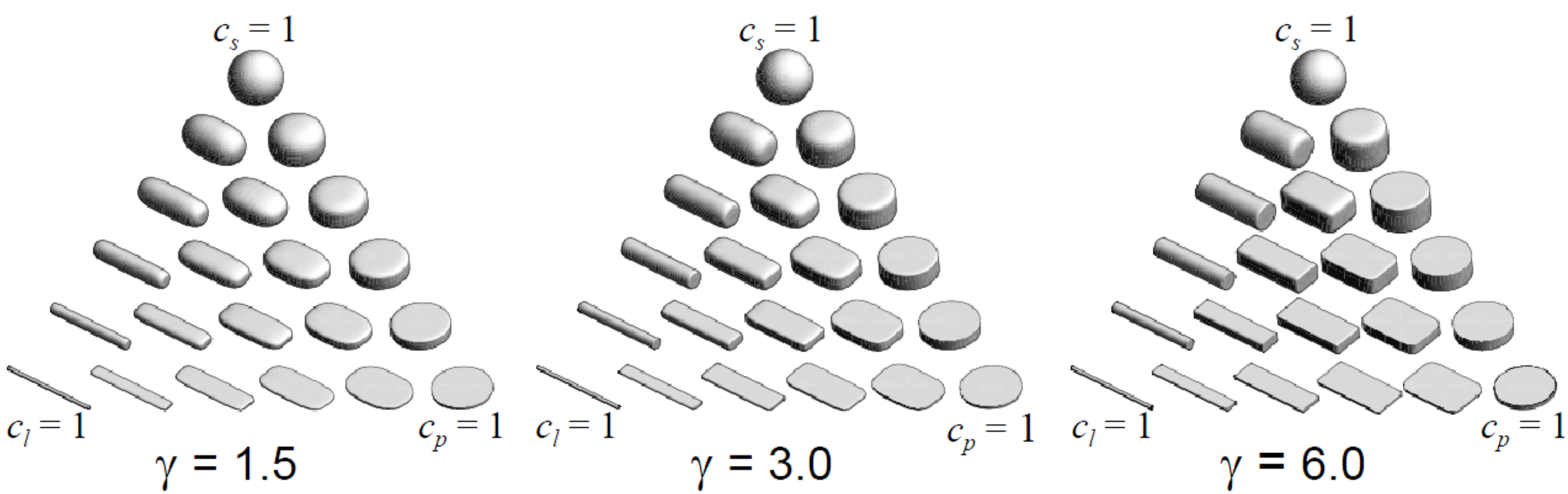
Shaded: sub-range used for glyphs

Superquadric glyphs (Kindlmann): **Given $\boldsymbol{c_l, c_p, c_s}$**

- Compute a base superquadric using an **<u>edge sharpness value $\boldsymbol{\gamma}$</u>**:

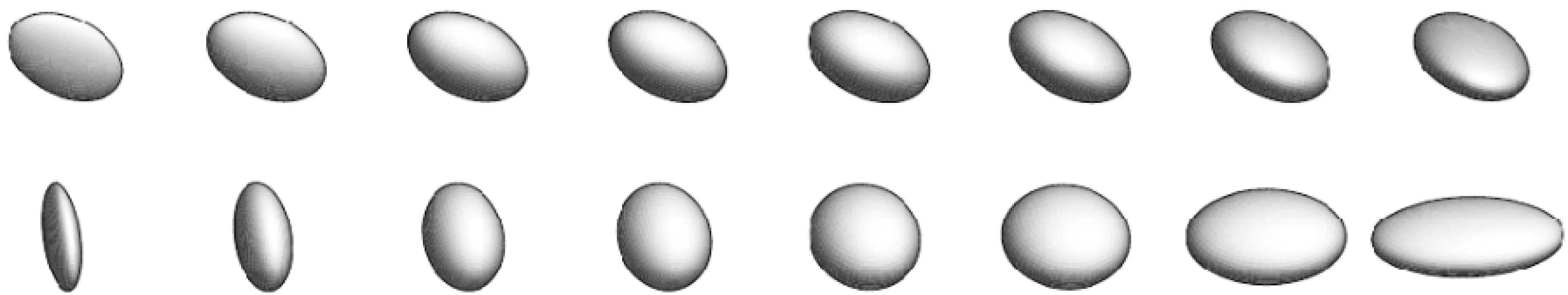
$$q(\theta,\emptyset) = \begin{cases} if\, c_l < c_p\colon\ q_z(\theta,\emptyset)\ with\ \alpha = \left(1-c_p\right)^{\gamma}\ and\ \beta = (1-c_l)^{\gamma} \\ if\, c_l \geq c_p\colon\ q_x(\theta,\emptyset)\ with\ \alpha = (1-c_l)^{\gamma}\ and\ \beta = \left(1-c_p\right)^{\gamma} \end{cases}$$

- Scale with c_l, c_p, c_s along x, y, z and rotate into eigenvector frame

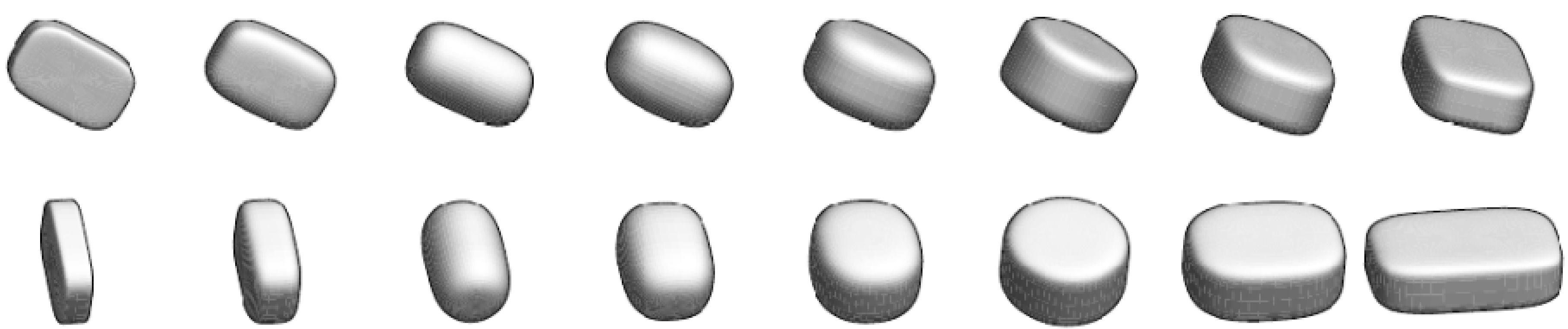


Comparison of shape perception (previous example)

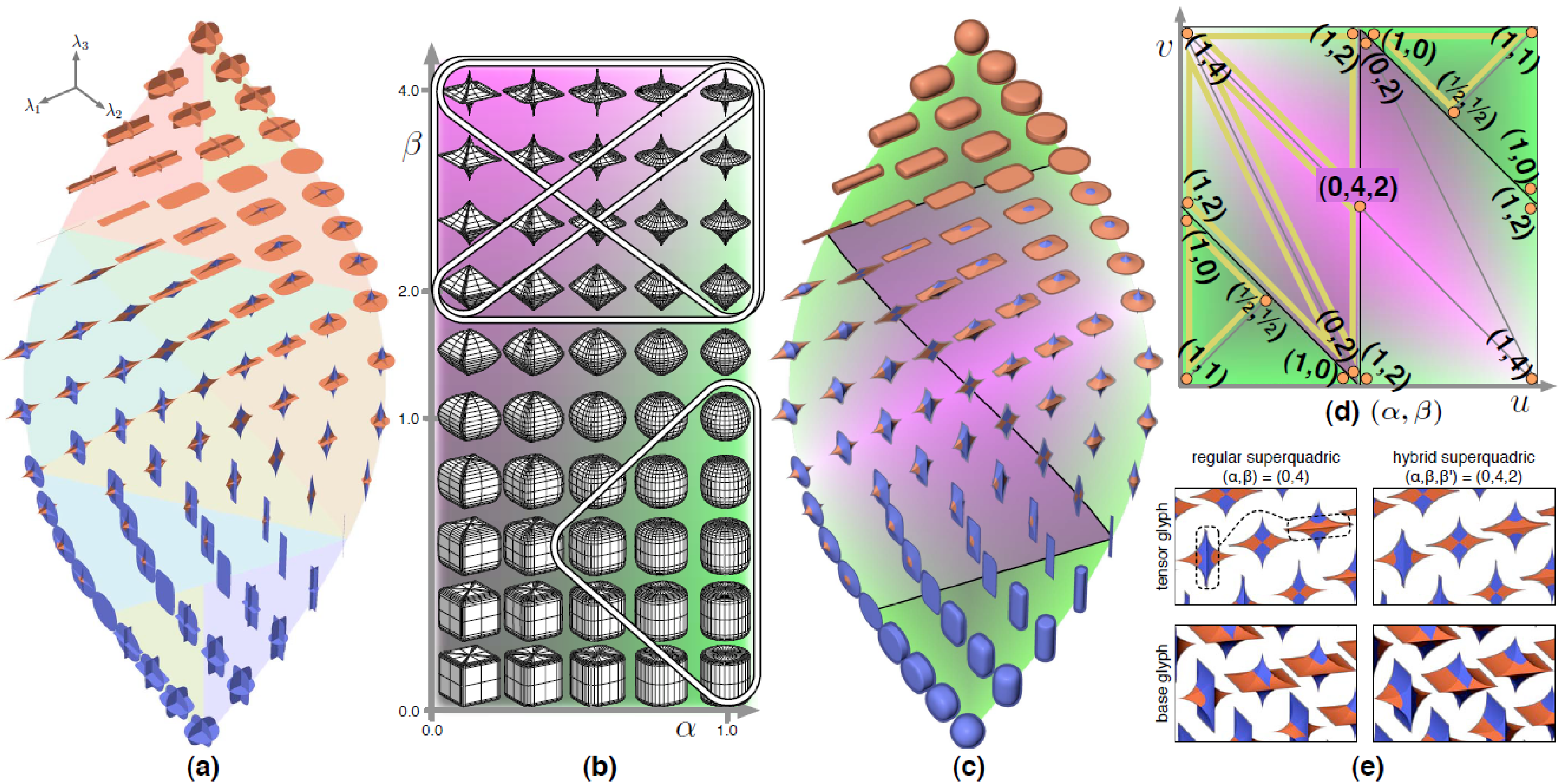
- With ellipsoid glyphs



- With superquadrics glyphs

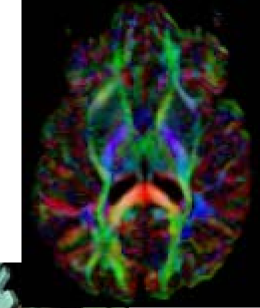


Superquadric Glyphs for Symmetric Second-Order Tensors

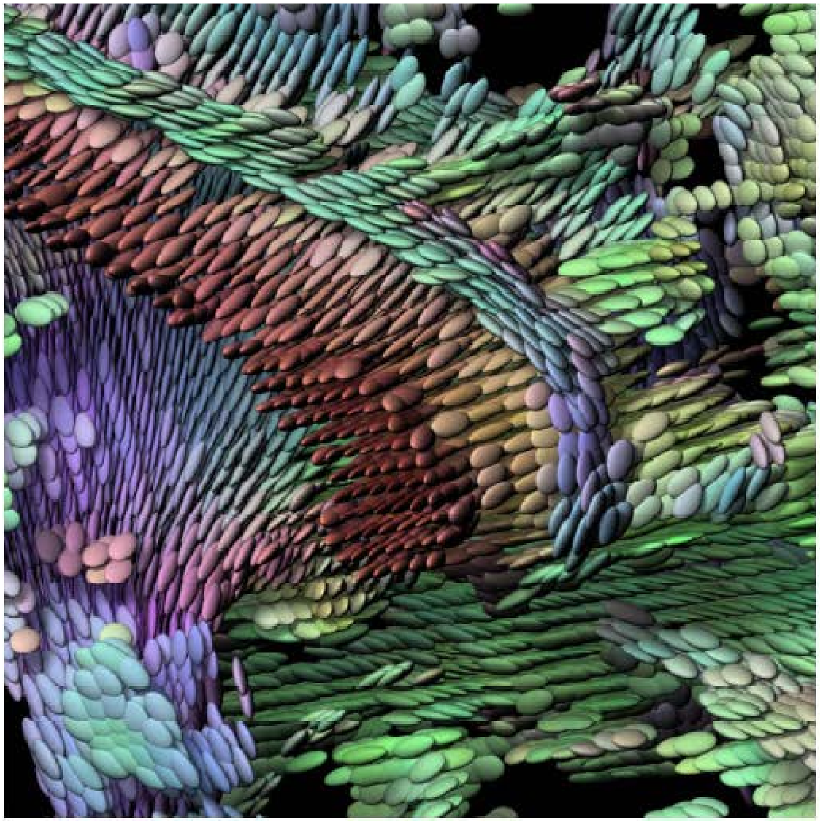


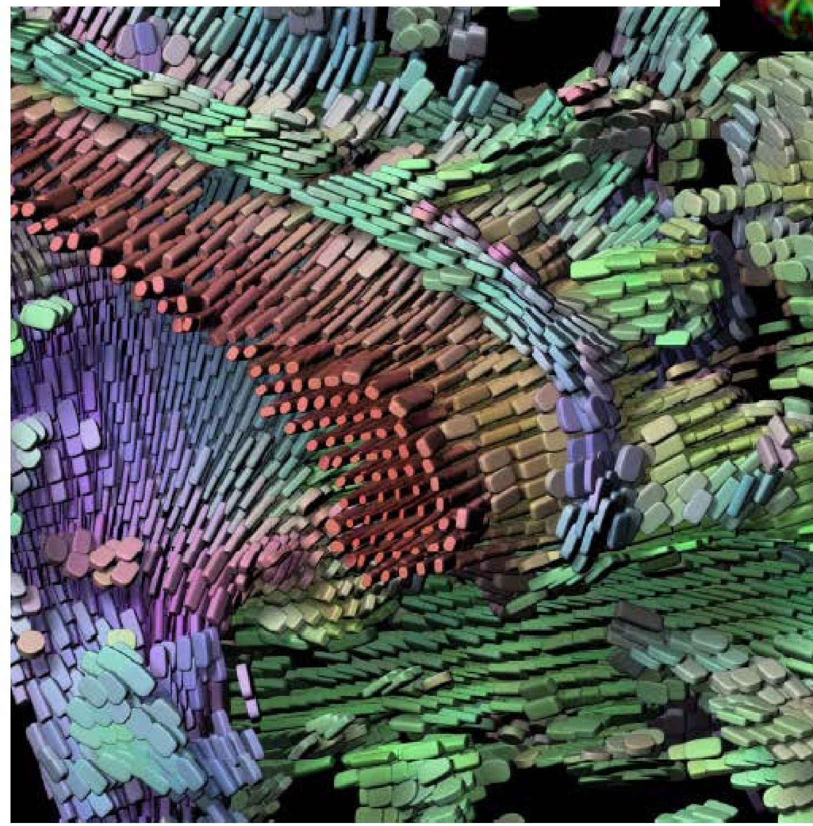
Extended to **general second-order symmetric** tensors that can be **indefinite (hyperbolic)**

[Schultz and Kindlmann, Vis10]



Comparison: Ellipsoids vs. superquadrics (Kindlmann)





Color map $\begin{pmatrix} R \\ G \\ B \end{pmatrix} = c_l \begin{bmatrix} |e_x^{\ 1}| \\ |e_y^{\ 1}| \\ |e_z^{\ 1}| \end{bmatrix} + (1 - c_l) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

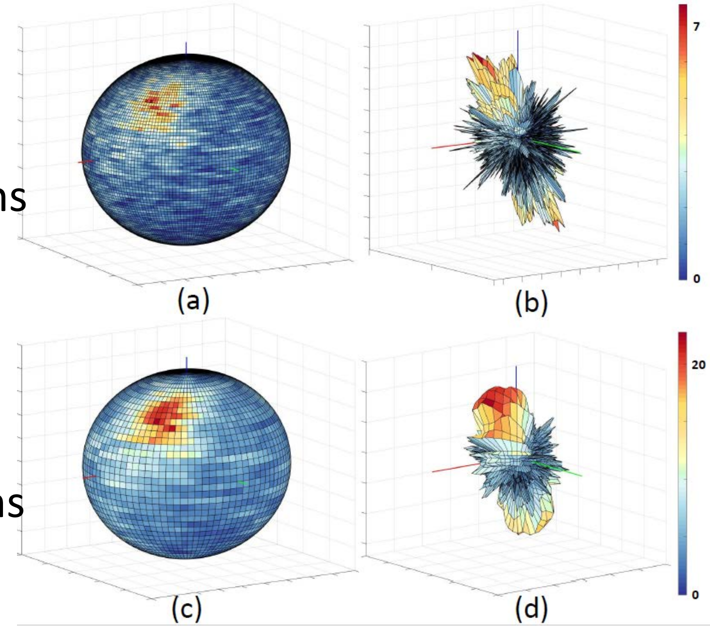
This is half of the brain, looking at the posterior part of the corpus callosum, which is the main bridge between the two hemispheres. And with the superquadrics, you can see that on the surface of the corpus callosum, the glyphs have more of a planar component, but on the inside, they're basically very linear.

Spherical Harmonic Glyphs

Spherical Harmonic Glyphs

1° bins

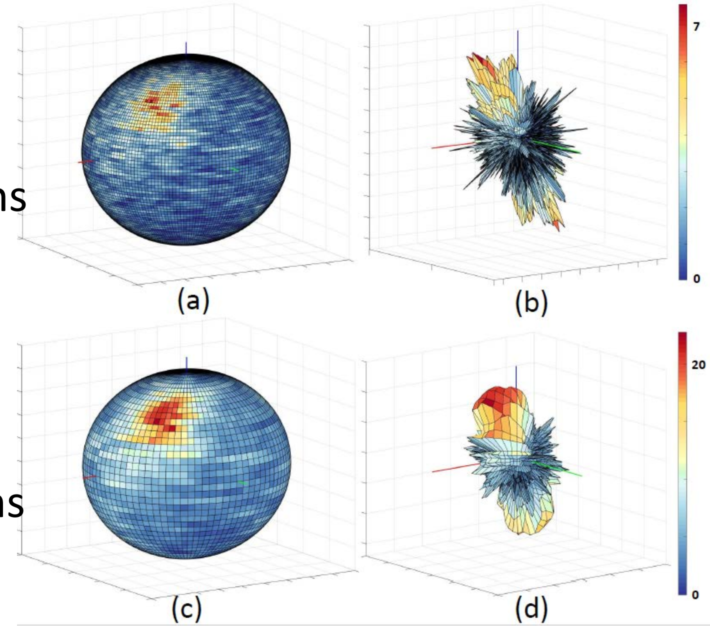
2° bins



Spherical Harmonic Glyphs

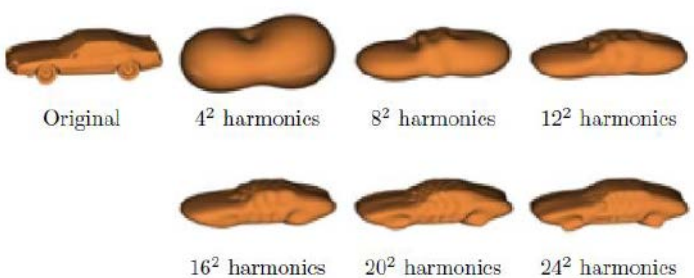
1° bins

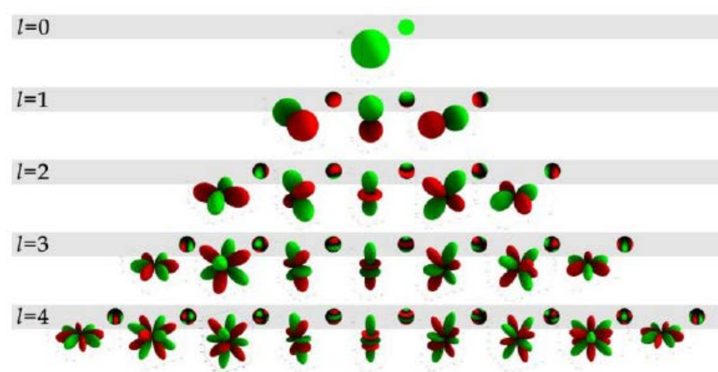
2° bins



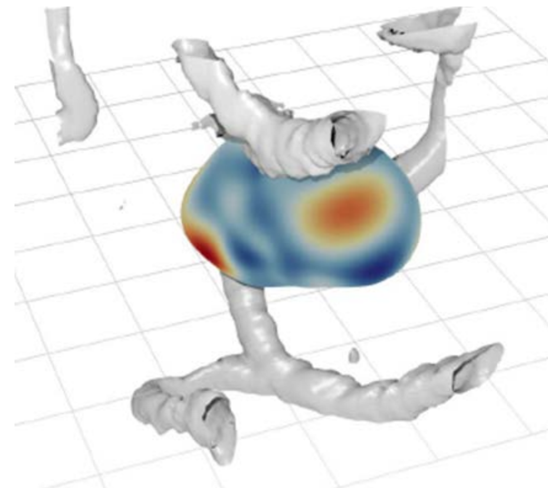
Spherical Harmonic

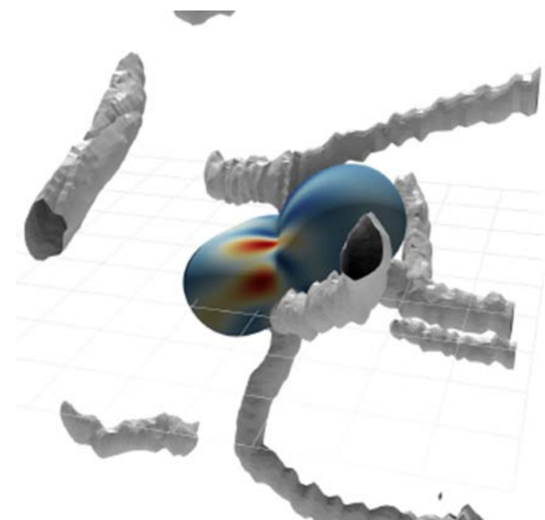
Fourier series in spherical coordinates





Spherical Harmonic Glyphs





The distribution of all fibers in the region can be modeled as a **set of spherical harmonic functions and their coefficients**

We perform a least-squares fitting of every segment in a region to a function defined in terms of a set of spherical harmonic: a cost K

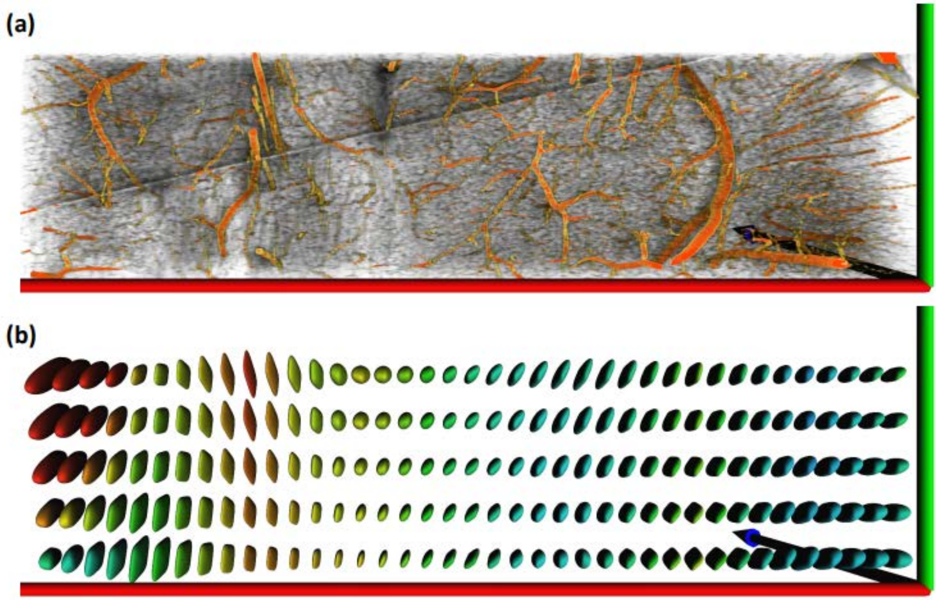
$$K(\vec{c}) = \sum_{n-1}^{N} \left[\sum_{i=0}^{B-1} c_i y_i(\theta_N, \phi_N) - s_n \right]^2$$

We solve by setting a derivative with respect to **c** to 0 and minimizing to get c_i

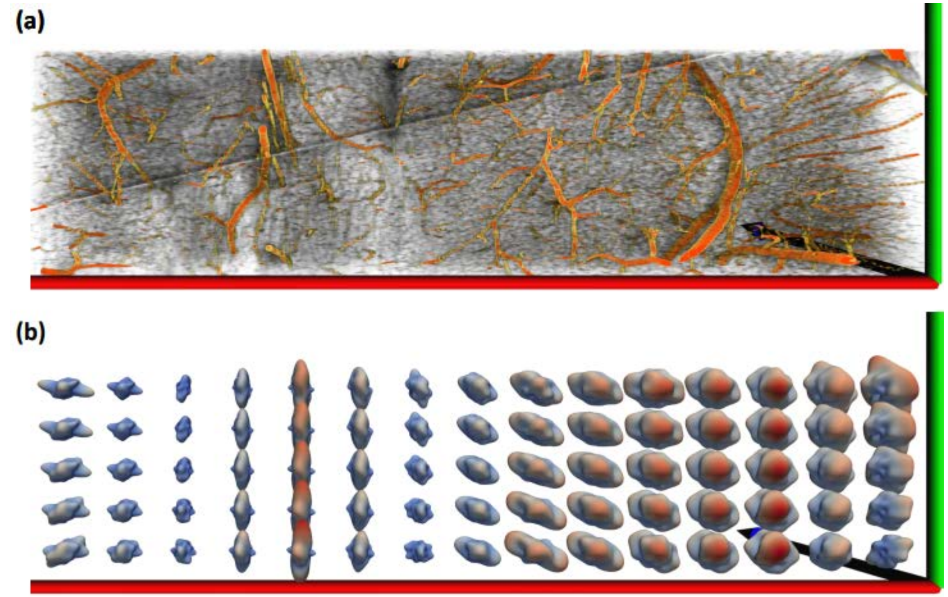
$$y_i(\theta, \phi) = y_{l(l+1)+m}(\theta, \phi) = y_l^m(\theta, \phi)$$

B = L(L+2) + 1, the maximum number of harmonic coefficients

Spherical Harmonic Glyphs



superquadric glyphs



spherical harmonic glyphs

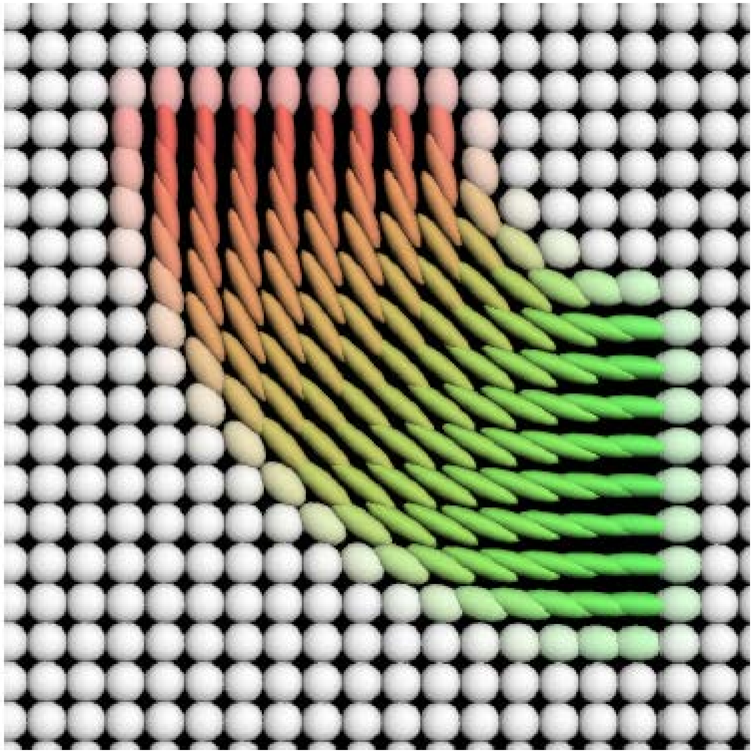
The glyph shape demonstrates the larger trends present in the data. See the upper left of the domain.

The glyph shape and colormap indicate two different features: the glyph shape indicates vessel volume/direction, while the glyph color indicates vessel radius/direction

SH glyphs are better than superquadrics at showing anisotropy and connectivity

GLYPH PACKING

Glyph Packing

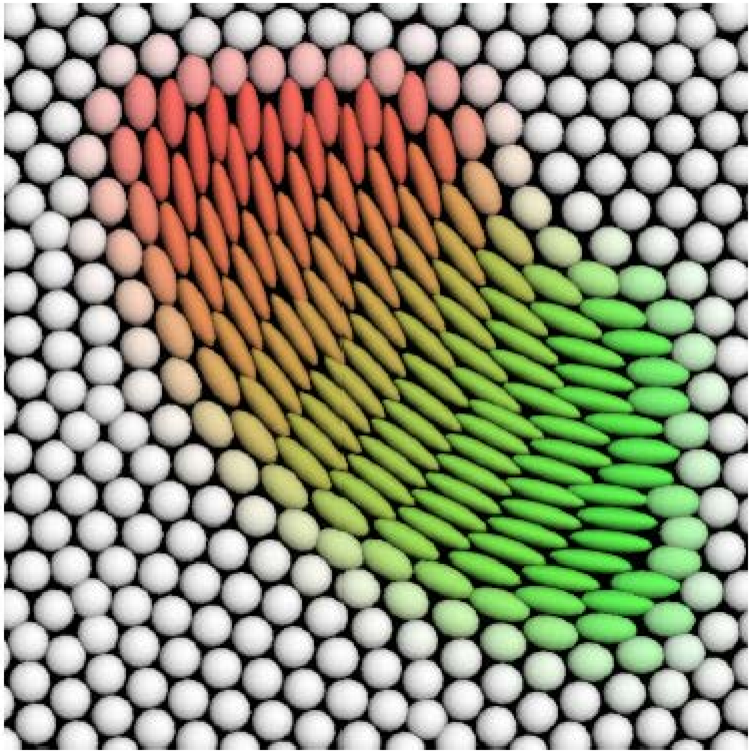


Glyphs are placed at regular grids

[Kindlmann and Westin, Vis06]

Glyph Packing

Glyphs are placed at regular grids



Glyphs are packed in better locations

[Kindlmann and Westin, Vis06]

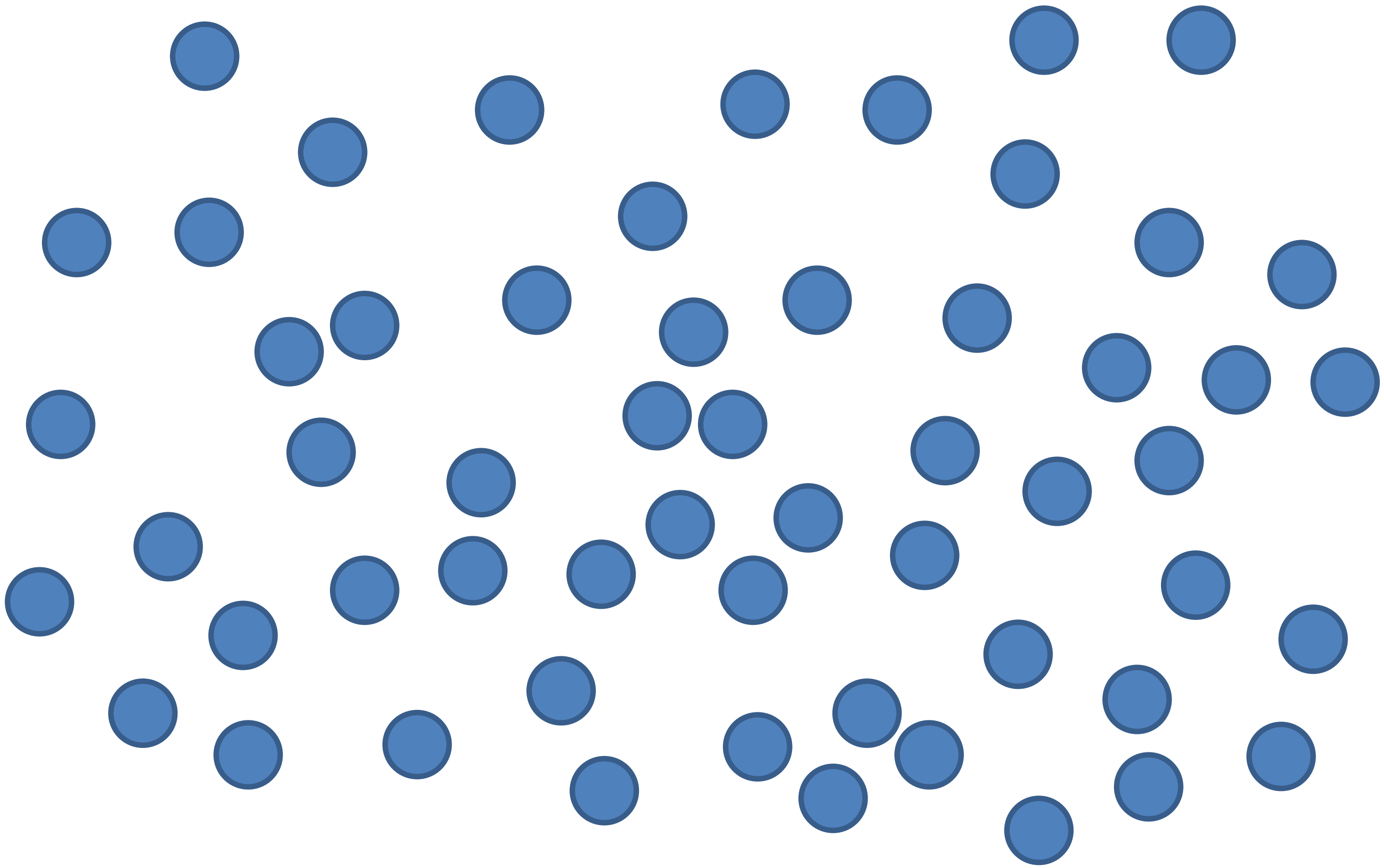
Requirements (Ideal Situation)

- No obvious patterns induced by the underlying spatial discretization
- No gaps between glyphs
- No overlapping

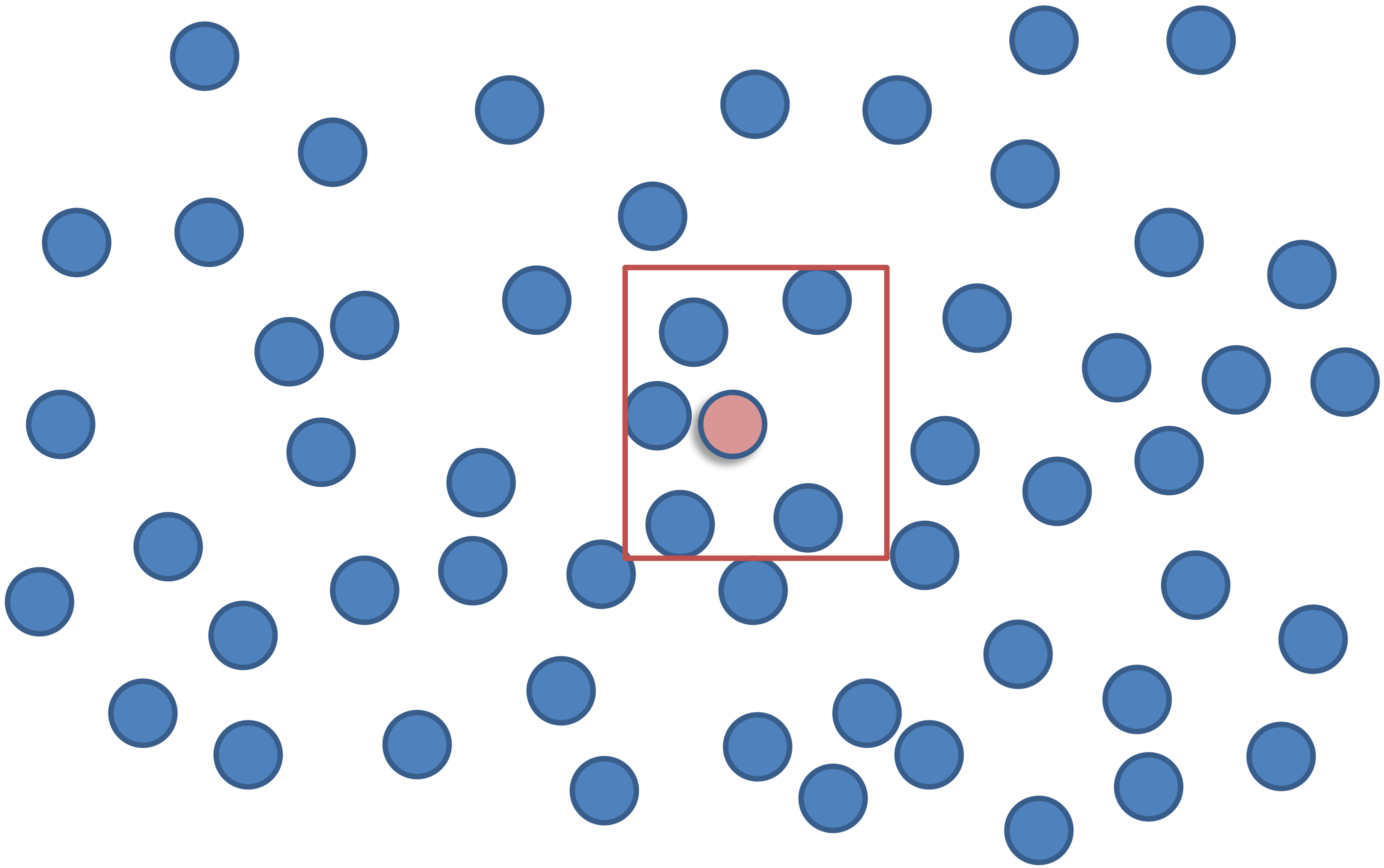
Energy-based Particle Systems

- Basic pipeline
  - Seeding based on some statistical property
  - Force repelling *
    - Each particle tries to push away its neighboring particles
    - This process should eventually converge to a stable configuration.
  - Rendering glyphs

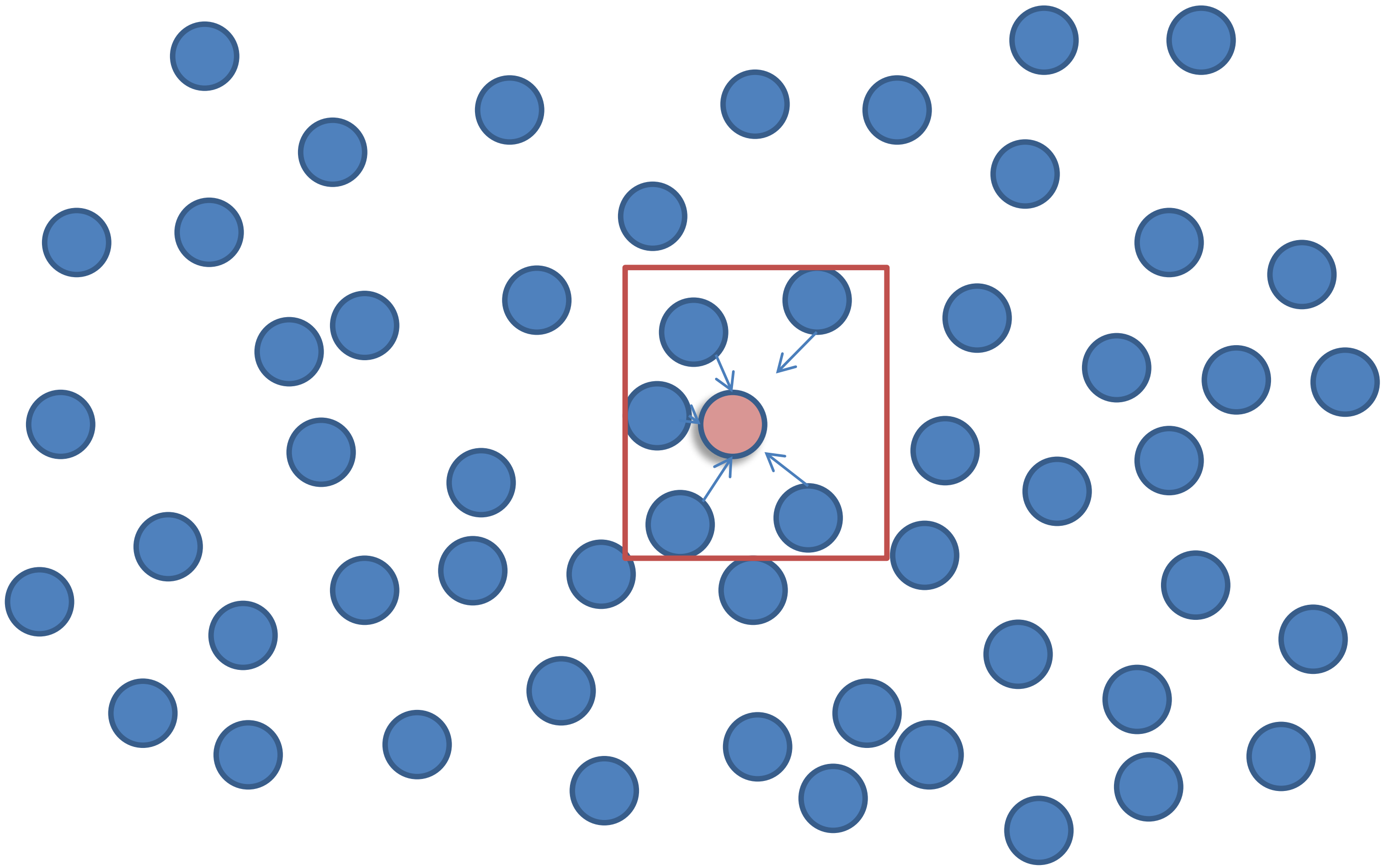
Energy-based Particle Systems



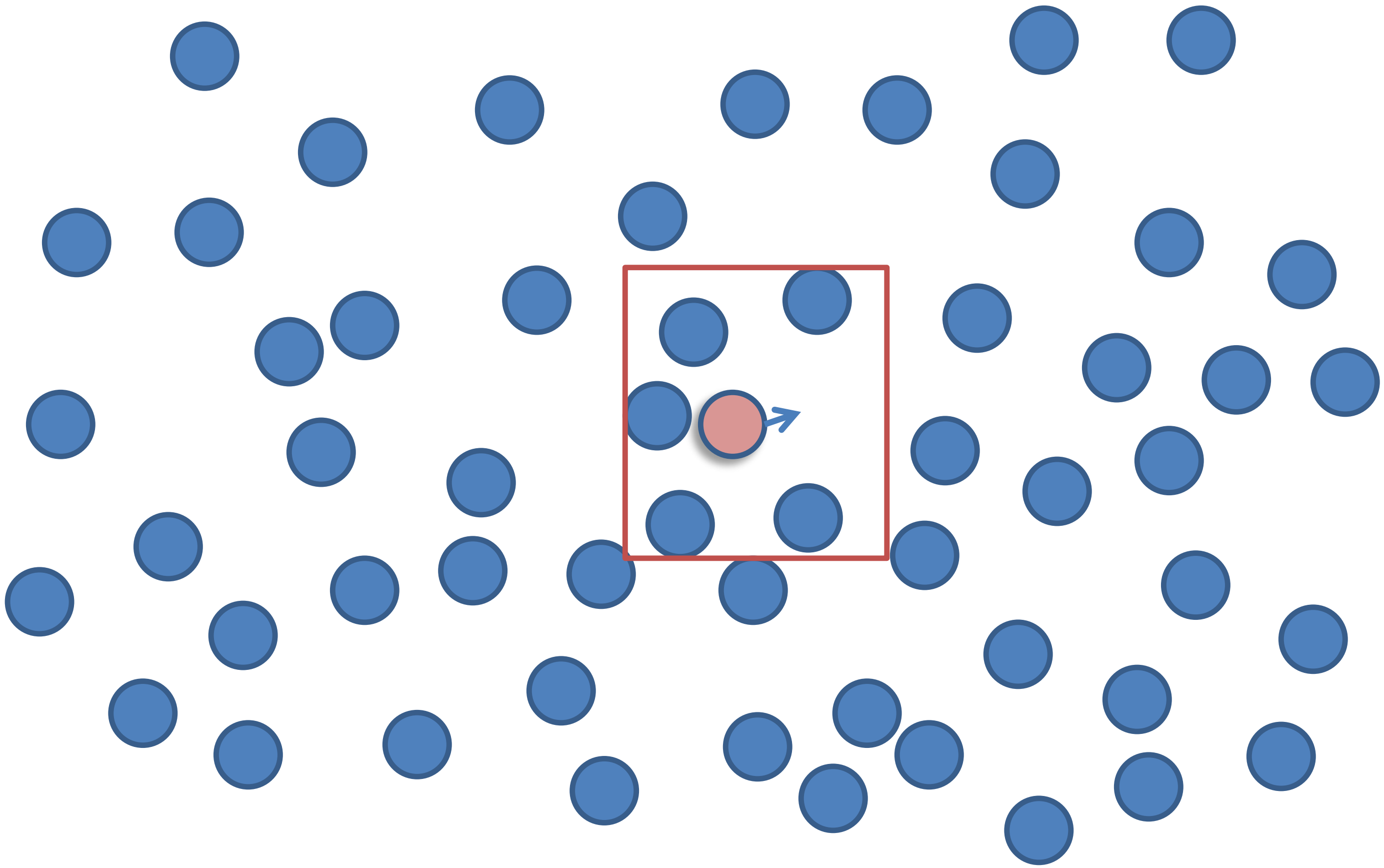
Energy-based Particle Systems



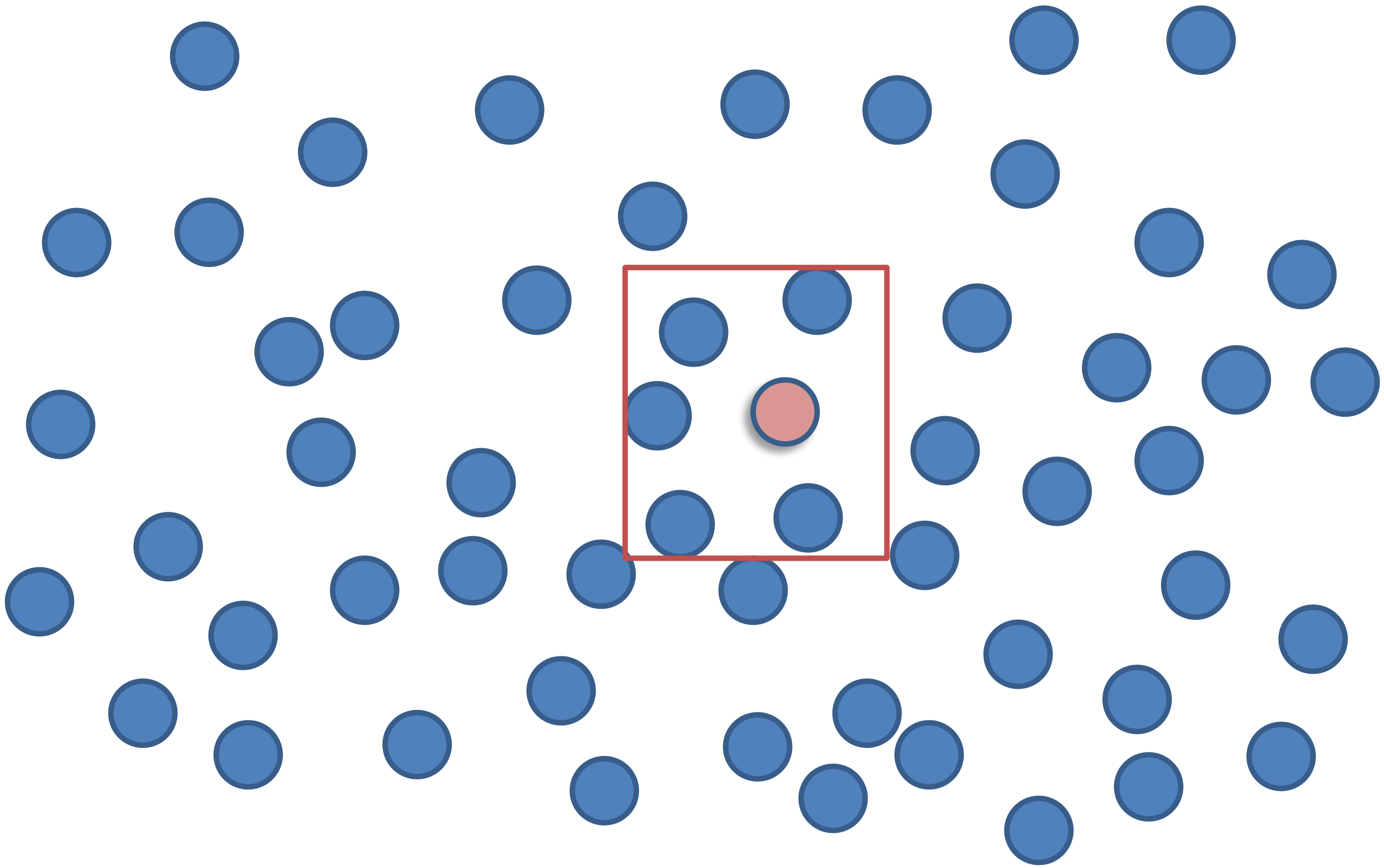
Energy-based Particle Systems



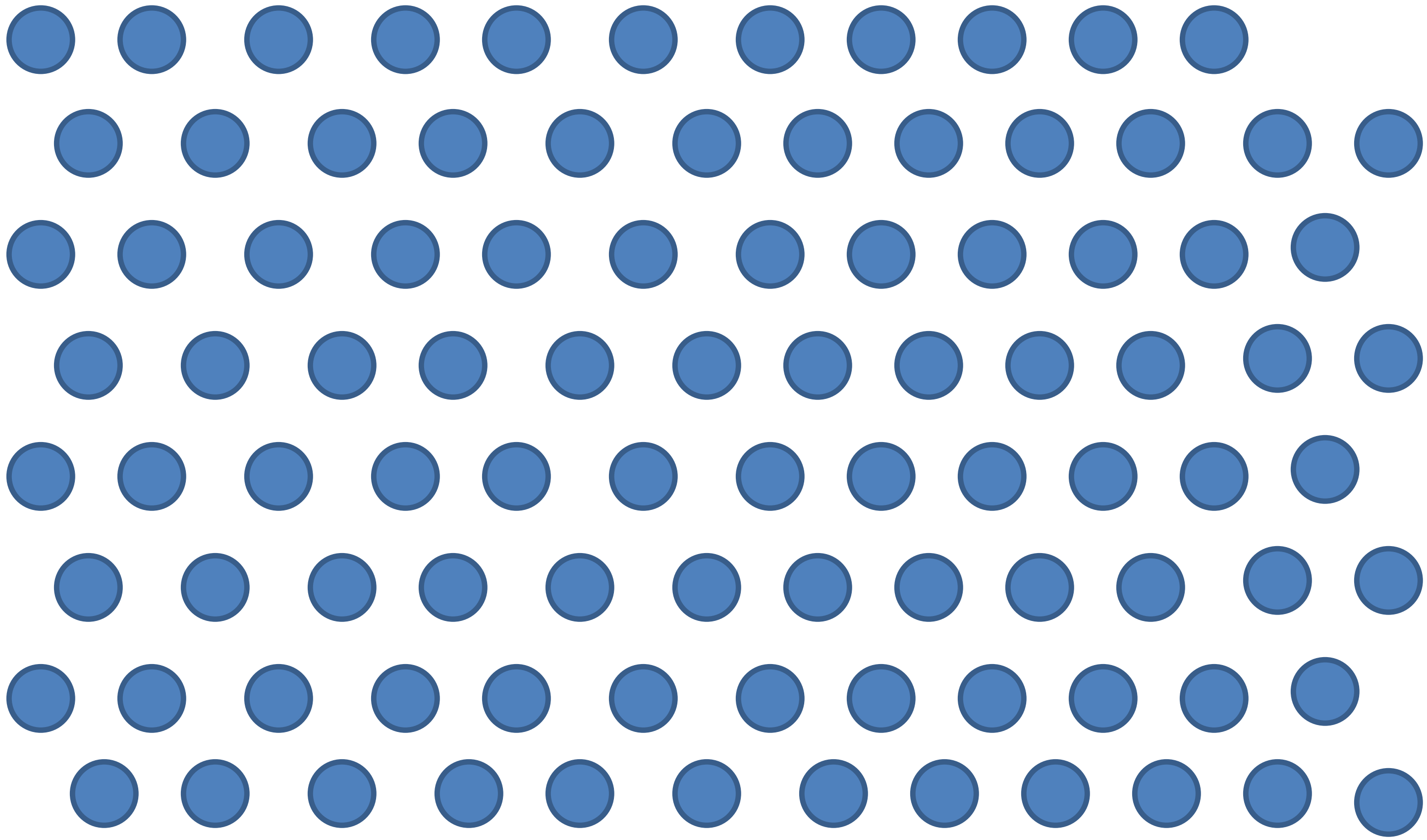
Energy-based Particle Systems



Energy-based Particle Systems



Energy-based Particle Systems



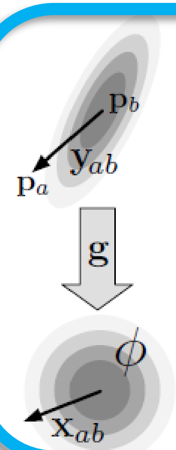
Overall Pipeline

Initialize particle positions
Do the following until convergence
For each particle
    Compute the accumulated force added by other particles
    Determine a direction and velocity
    Move this particle to the new location based on this vector

Computation of the Energy

3.2 Tensor-based Potential Energy

The behavior of particles in our system is determined by the forces acting upon them. The most important of these are the forces between particles, which gives rise to the glyph packing. The inter-particle forces are created by a potential energy field around each particle, shaped by the local tensor value. The energy E_{ab} at position $\mathbf{p}_a$ due to the potential energy field around a particle at $\mathbf{p}_b$ is the composition of functions $\mathbf{g}$, $|\cdot|$, and ϕ:



$$E_{ab} = \phi(|\mathbf{g}(\mathbf{y}_{ab})|) = \phi(r_{ab}) \quad (1)$$

$$\mathbf{y}_{ab} = \mathbf{p}_a - \mathbf{p}_b \quad (2)$$

$$\mathbf{x}_{ab} = \mathbf{g}(\mathbf{y}_{ab}) = \frac{\mathbf{D}_{ab}^{-1}\mathbf{y}_{ab}}{2\alpha}; \quad (3)$$

$$\mathbf{D}_{ab} = \mathbf{D}\left(\frac{\mathbf{p}_a + \mathbf{p}_b}{2}\right) \quad (4)$$

$$r_{ab} = |\mathbf{x}_{ab}| \quad (5)$$

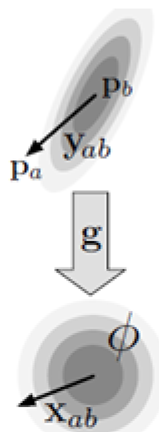
α is the glyph scaling factor

Note that by construction, $\mathbf{g}$ inverts the transform that creates ellipsoidal glyphs from spheres. Conceptually, $\mathbf{g}$ maps vectors $\mathbf{y}_{ab}$ in the field of anisotropic tensors to vectors $\mathbf{x}_{ab}$ in an isotropic space, in which particles have a rotationally symmetric potential energy profile ϕ. O'Donnell *et al.* also use the tensor inverse as a metric for computing geodesics under diffusion tensor warping [30]. One way to characterize the local tensor value $\mathbf{D}_{ab}$ is to sample the tensor field at the midpoint between the two particles, which ensures symmetry $E_{ab} = E_{ba}$. In Equation 3, the factor of two in the denominator of $\mathbf{g}$ allows mutually tangent glyphs to map to mutually tangent spheres of radius $1/2$, with unit distance between centers (relevant for the later definition of ϕ).

[Kindlmann and Westin, Vis06]

Computation of the Forces

α is the glyph scaling factor



$$
\begin{aligned}
E_{ab} &= \phi(|\mathbf{g}(\mathbf{y}_{ab})|) = \phi(r_{ab}) \\
\mathbf{y}_{ab} &= \mathbf{p}_a - \mathbf{p}_b \\
\mathbf{x}_{ab} &= \mathbf{g}(\mathbf{y}_{ab}) = \frac{\mathbf{D}_{ab}^{-1}\mathbf{y}_{ab}}{2\alpha}; \\
& \quad \mathbf{D}_{ab} = \mathbf{D}\left(\frac{\mathbf{p}_a + \mathbf{p}_b}{2}\right) \\
r_{ab} &= |\mathbf{x}_{ab}|
\end{aligned}
$$

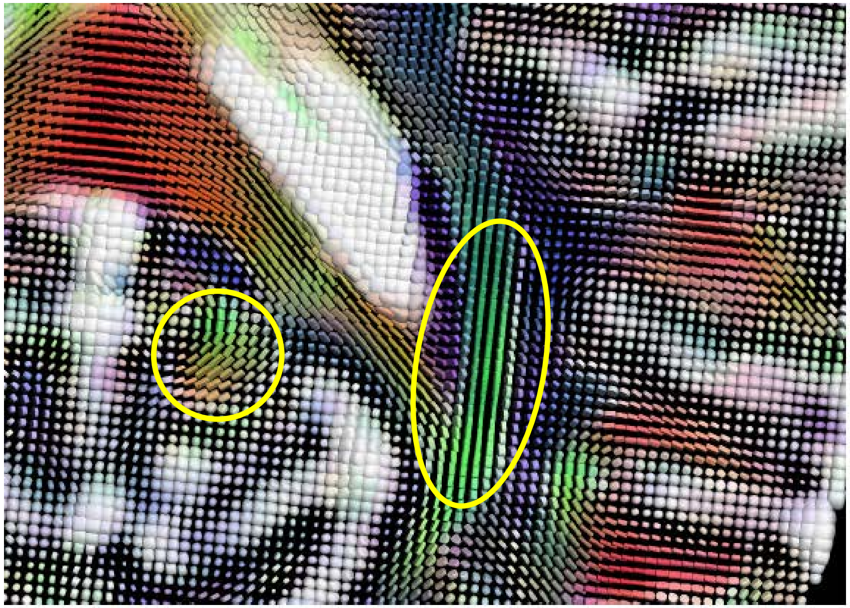
With the assumption of local tensor constancy, the force $\mathbf{f}_{ab}$ on a particle at $\mathbf{p}_a$ from a particle at $\mathbf{p}_b$ is (written as a column vector):

$$\boxed{\mathbf{f}_{ab} = -\left(\frac{dE_{ab}}{d\mathbf{y}_{ab}}\right)^T = -\frac{\phi'(|\mathbf{x}_{ab}|)}{2\alpha|\mathbf{x}_{ab}|}\,\mathbf{D}_{ab}^{-1}\,\mathbf{x}_{ab}} \tag{8}$$

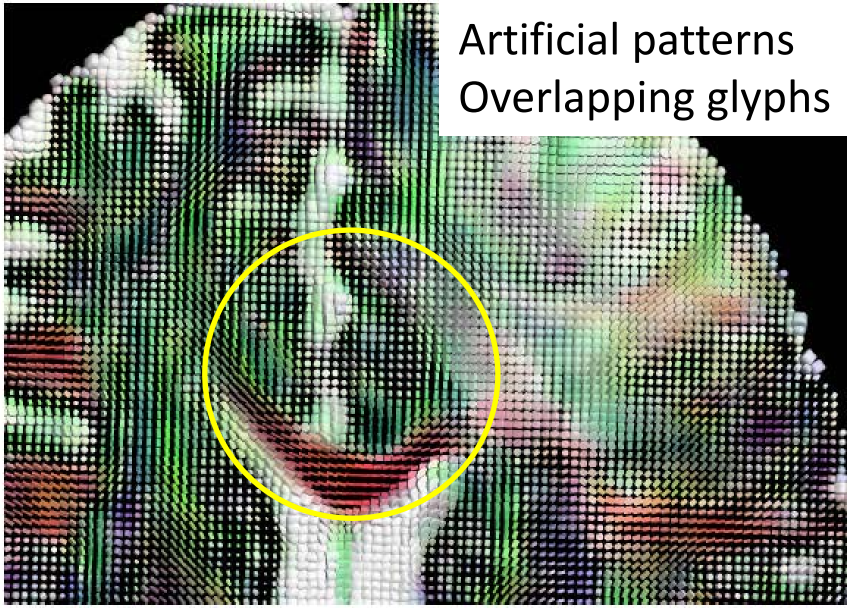
where $(\mathbf{D}^{-1})^T = \mathbf{D}^{-1}$ because diffusion tensors are symmetric. Note that the force between two particles is not in general aligned with the vector between them, which is an unusual property of our particle system. This is analogous to how an object's surface normals are transformed by the inverse-transpose of the object transform, as is commonly used for instancing in ray-tracing [38].

[Kindlmann and Westin, Vis06]

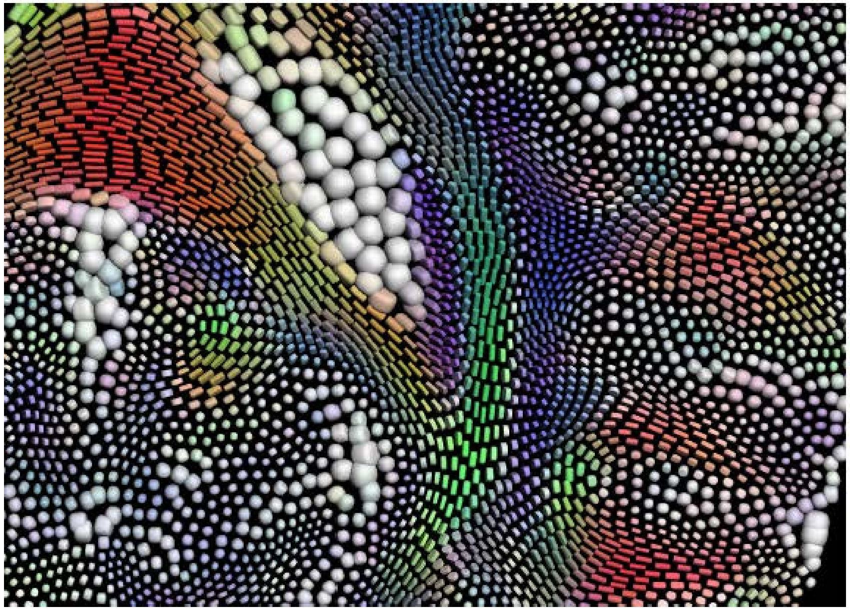
Artificial patterns
Overlapping glyphs

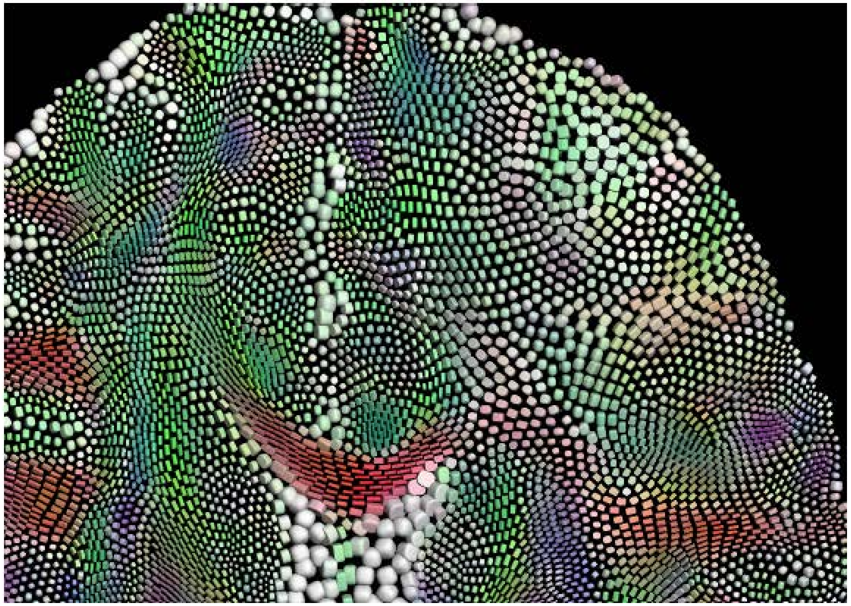


(a) Regular Grid



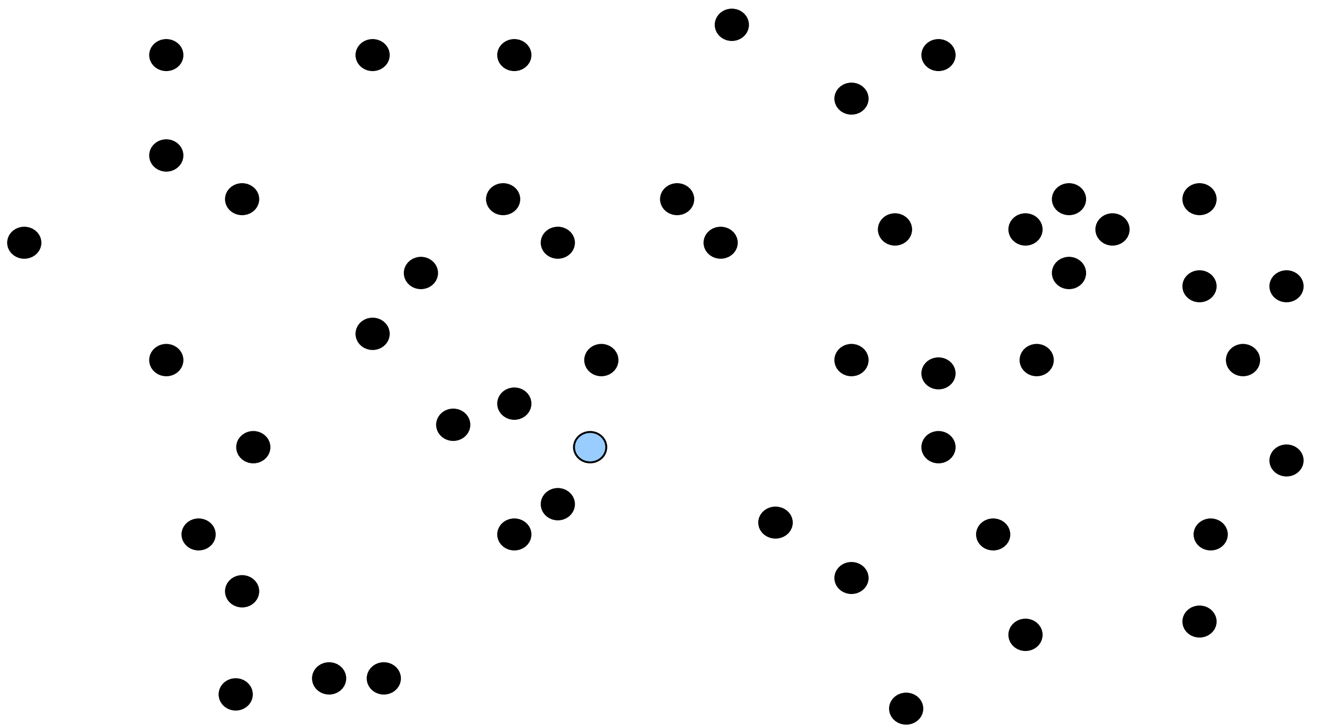
(a) Regular Grid





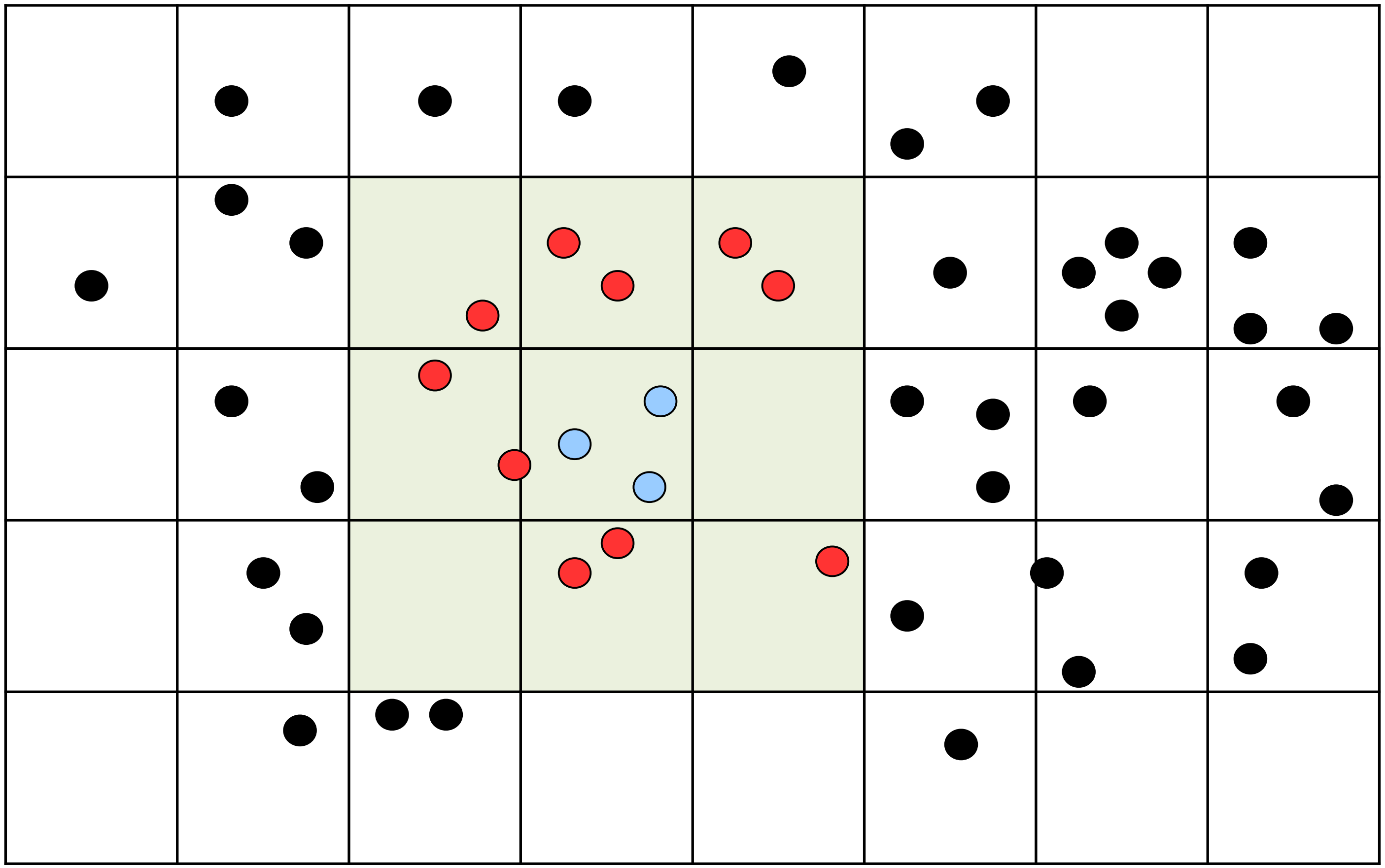
[Kindlmann and Westin, Vis06]

Improvement- Parallel Computation



Original method considers all the particles in the domain

Improvement- Parallel Computation



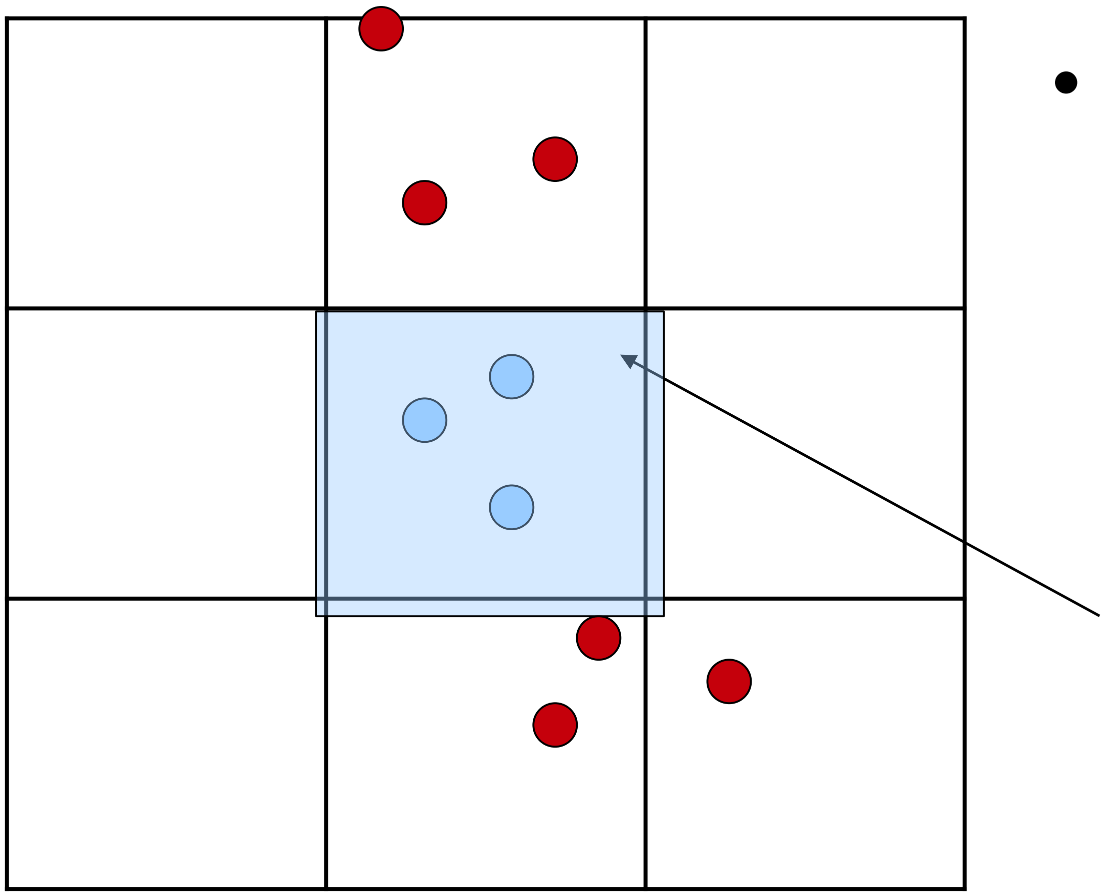
[Kim et al. GPGPU5 2012]

Multithreaded (cont)

- Given the current bin B_i
  - Gather every particle in B_i plus the immediate surrounding bins
    - This is a **neighborhood**
  - For every particle p_i in the bin B_i
    - For every particle p_j in the neighborhood
      - If distance from p_i to $p_j < 1.0$
        - sum the velocity and energy
    - Advect p_i

[Kim et al. GPGPU5 2012]

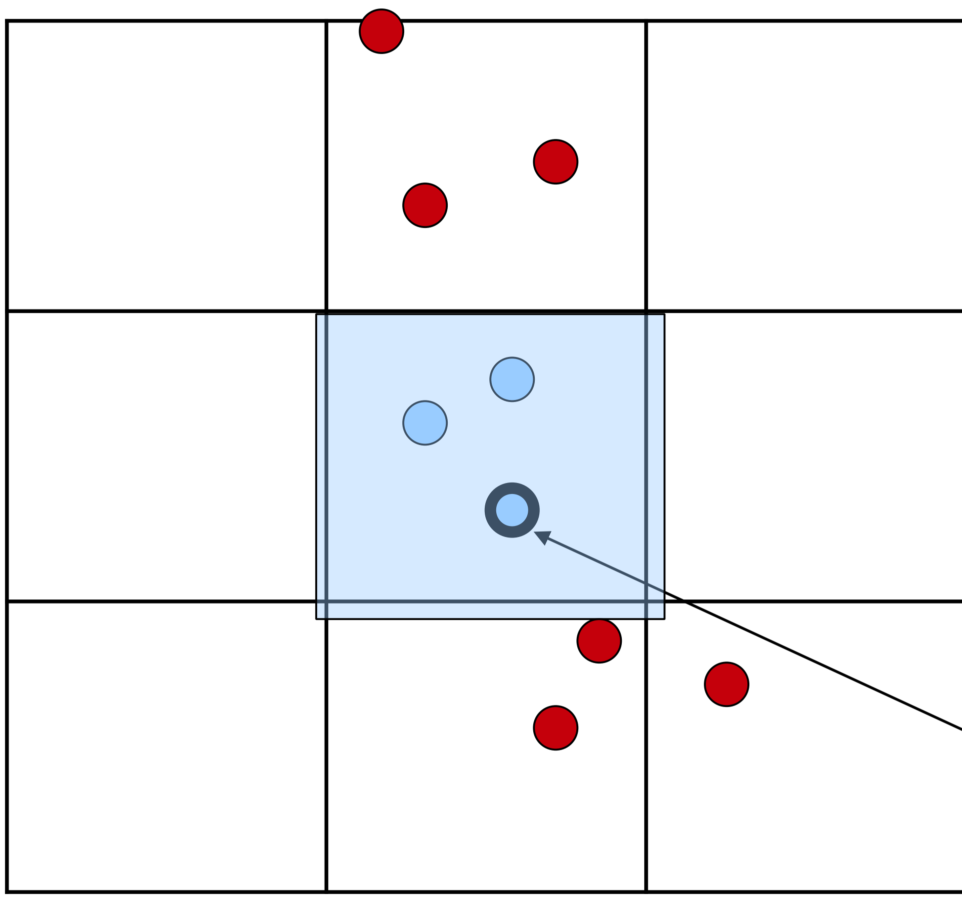
Multithreaded (cont.)



- Process each particle in the current bin

[Kim et al. GPGPU5 2012]

Multithreaded (cont.)

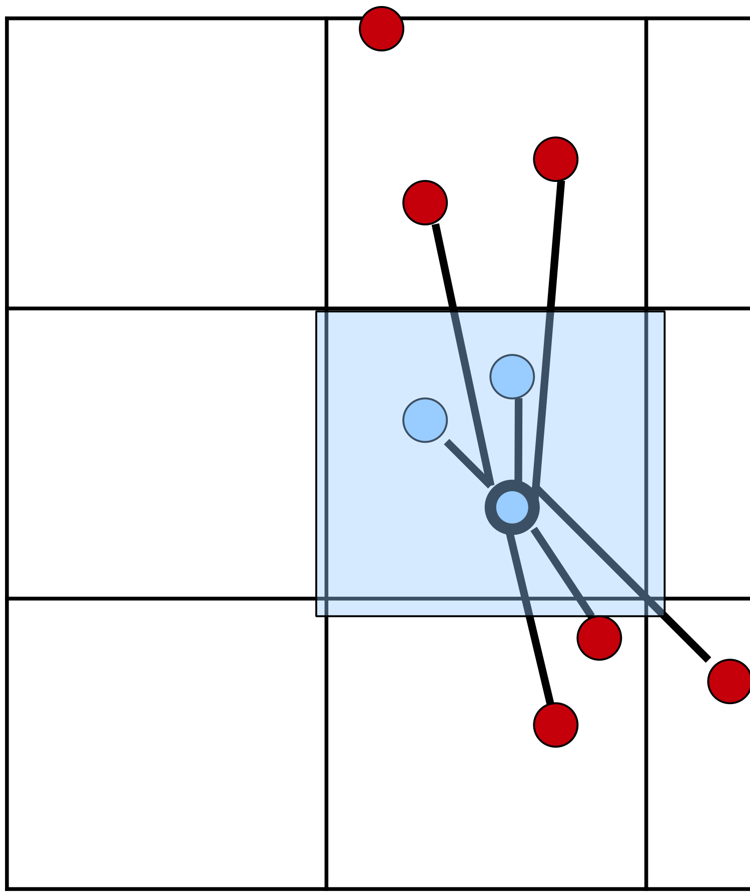


- Process each particle in the current bin

Current Particle

[Kim et al. GPGPU5 2012]

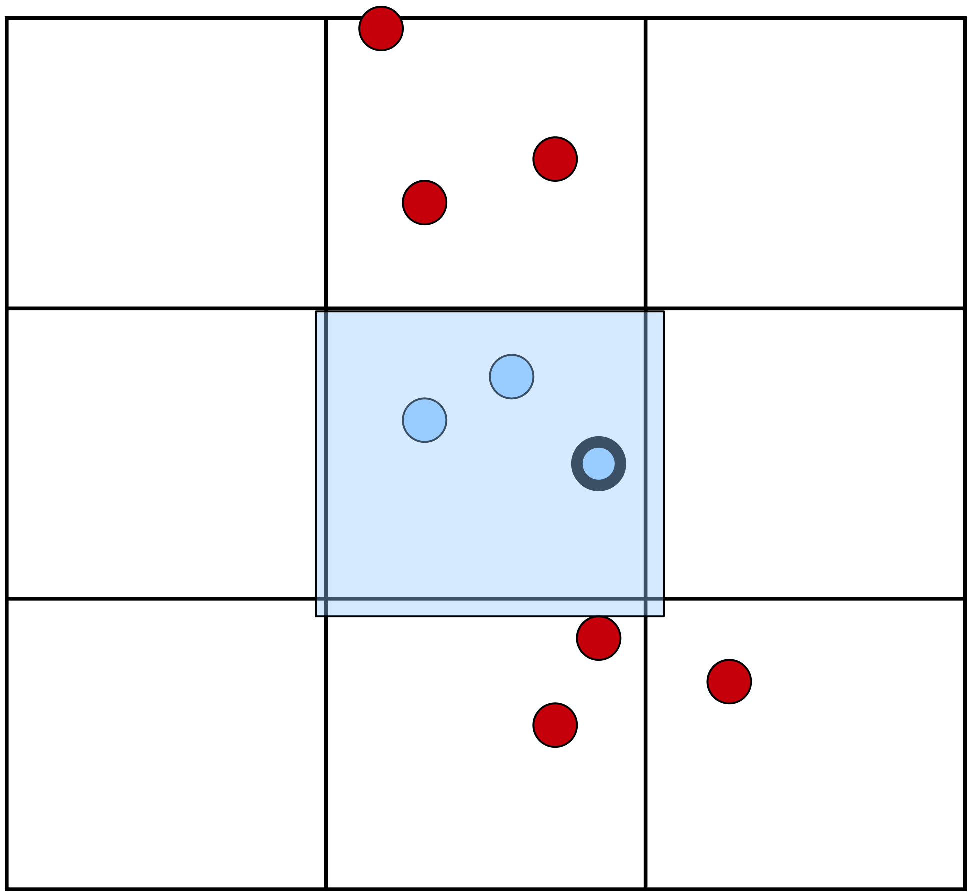
Multithreaded (cont.)



- sum Energy and Force

[Kim et al. GPGPU5 2012]

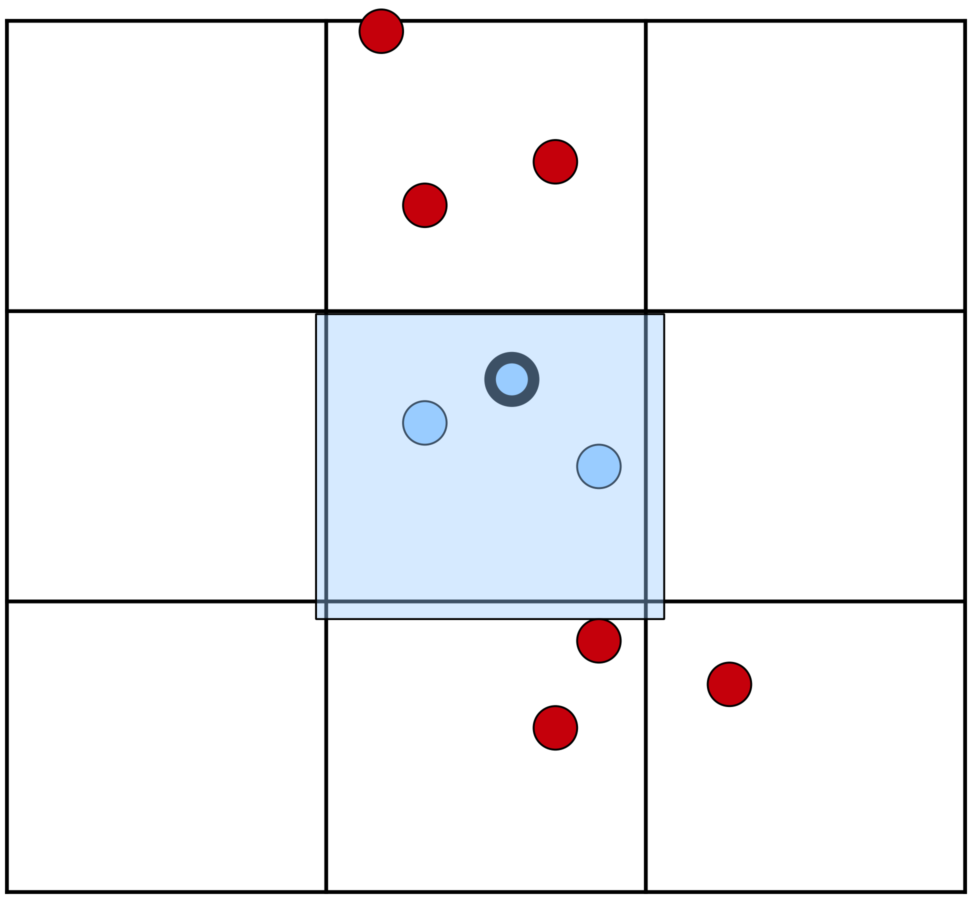
Multithreaded (cont.)



- Move current particle

[Kim et al. GPGPU5 2012]

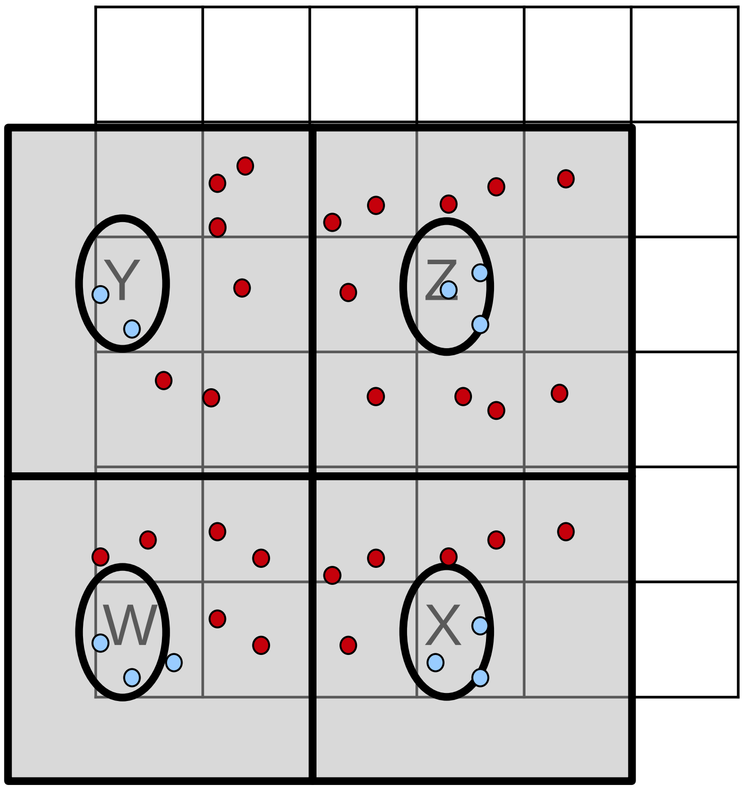
Multithreaded (cont.)



- Process the next particle in the current bin.

[Kim et al. GPGPU5 2012]

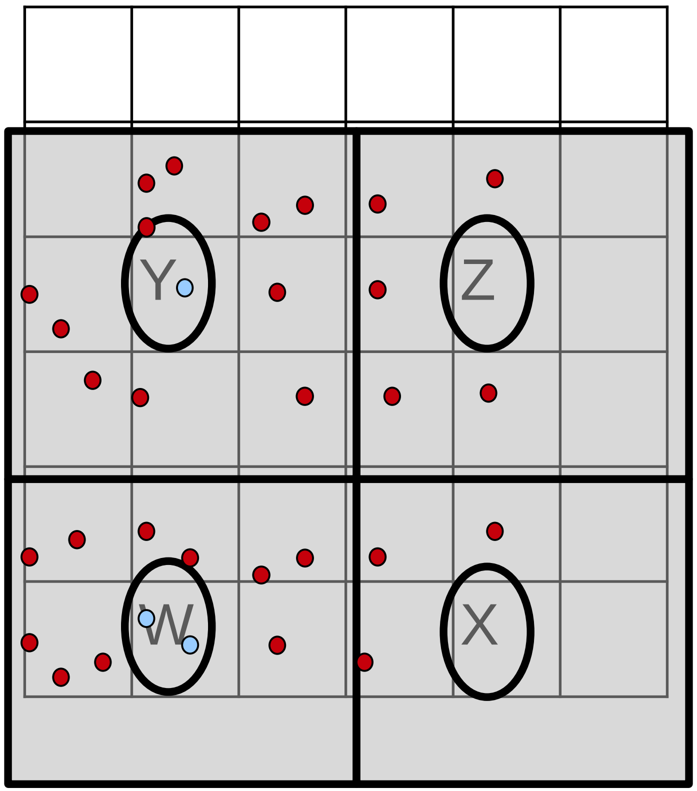
Multithreaded (cont)



- While there are bins to be processed
  - For every particle p in the current bin
    - For every other particle in the neighborhood
      - calculate force and energy
    - Move the particle in the direction F

[Kim et al. GPGPU5 2012]

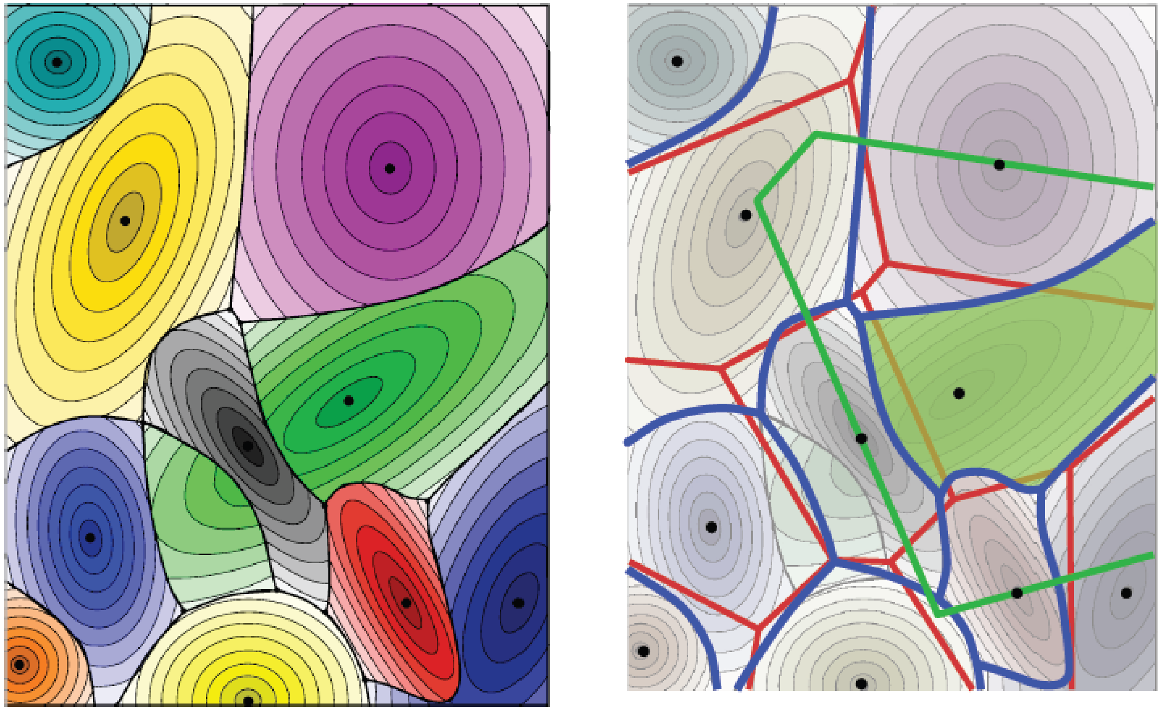
Multithreaded (cont)



- While there are bins to be processed
  - For every particle p in the current bin
    - For every other particle in the neighborhood
      - calculate force and energy
    - Move the particle in the direction F

[Kim et al. GPGPU5 2012]

Anisotropy Sampling



(a) (b)

Fig. 5. (a) Voronoi cells resulting from the approximate distance function defined by the metric in the sample points. The green Voronoi cell is en example for a non-connected cell, having one orphan. Such orphans are undesirable for the relaxation process. (b) Comparison of the various Voronoi cells. The red lines show the boundaries of the standard Euclidean Voronoi cells. The gray lines show the boundaries of the anisotropic Voronoi cells with orphans as defined by Equation 8. Most of the lines are covered by the blue lines, which represent the localized anisotropic Voronoi cells as defined in Equation 9. The green line indicates the area to which the green Voronoi cell is restricted. These examples are based on Labelle and Shewchuk [7].

[Feng et al. TVCG2008]

Anisotropy Sampling

Fig. 7. Anisotropic Voronoi cells resulting from the simplified distance measure. The grid used for relaxation has a resolution of 512×512.

[Feng et al. TVCG2008]

Anisotropy Sampling

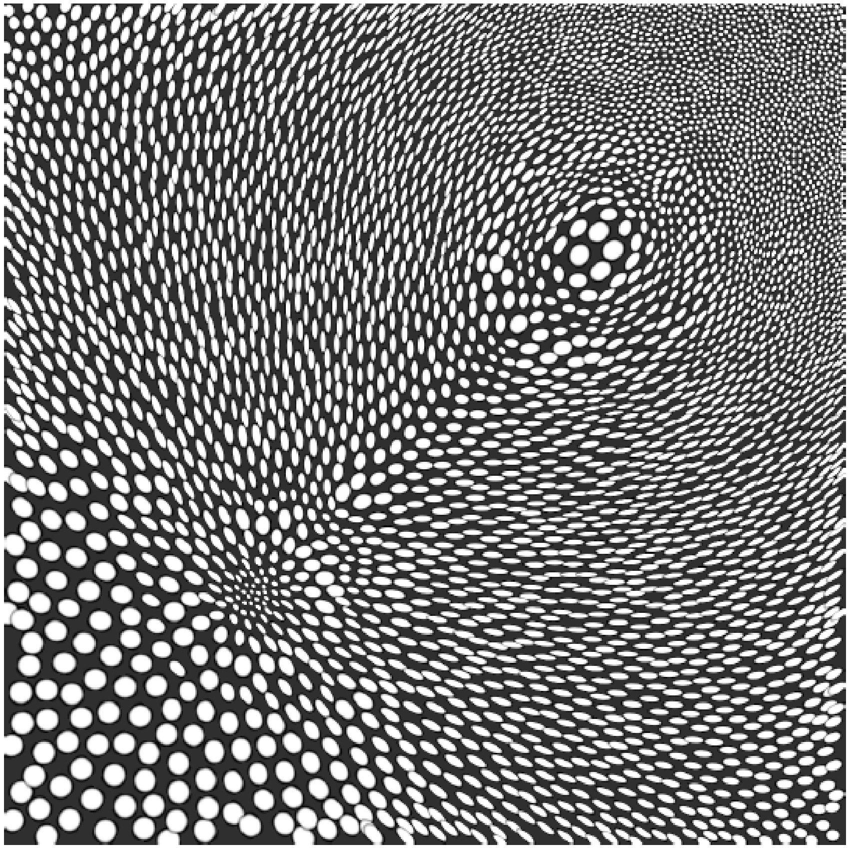


Fig. 15. Slice of a numerical simulation of a solid block with two forces acting on the block, one pushing and one pulling force. This image shows the tensor data as ellipses. The ellipses provide an idea about the directions of contraction or expansion inside the material.

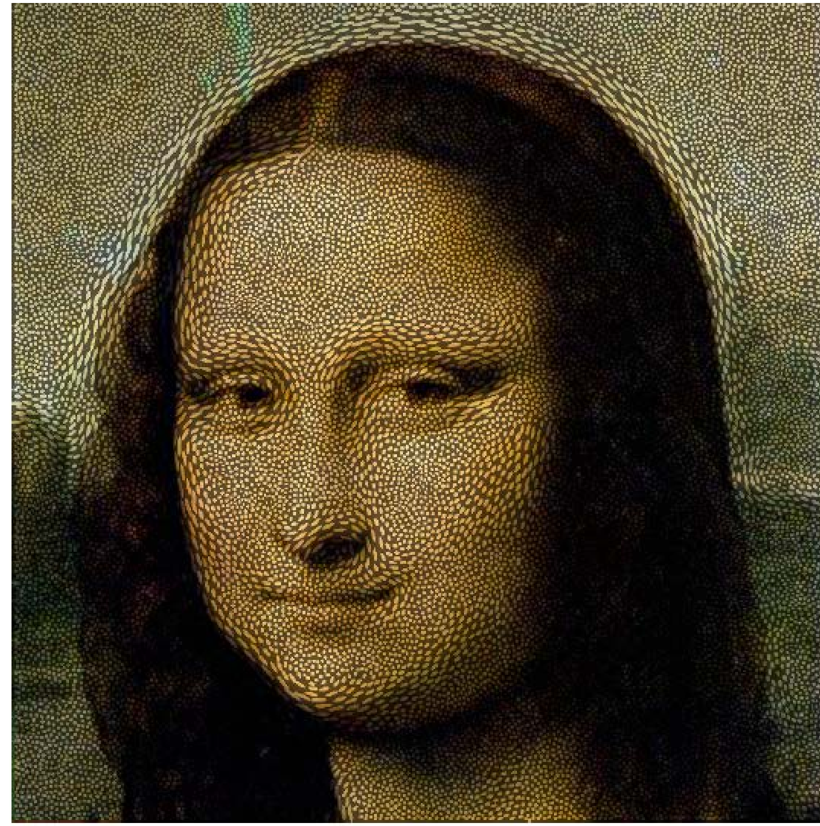
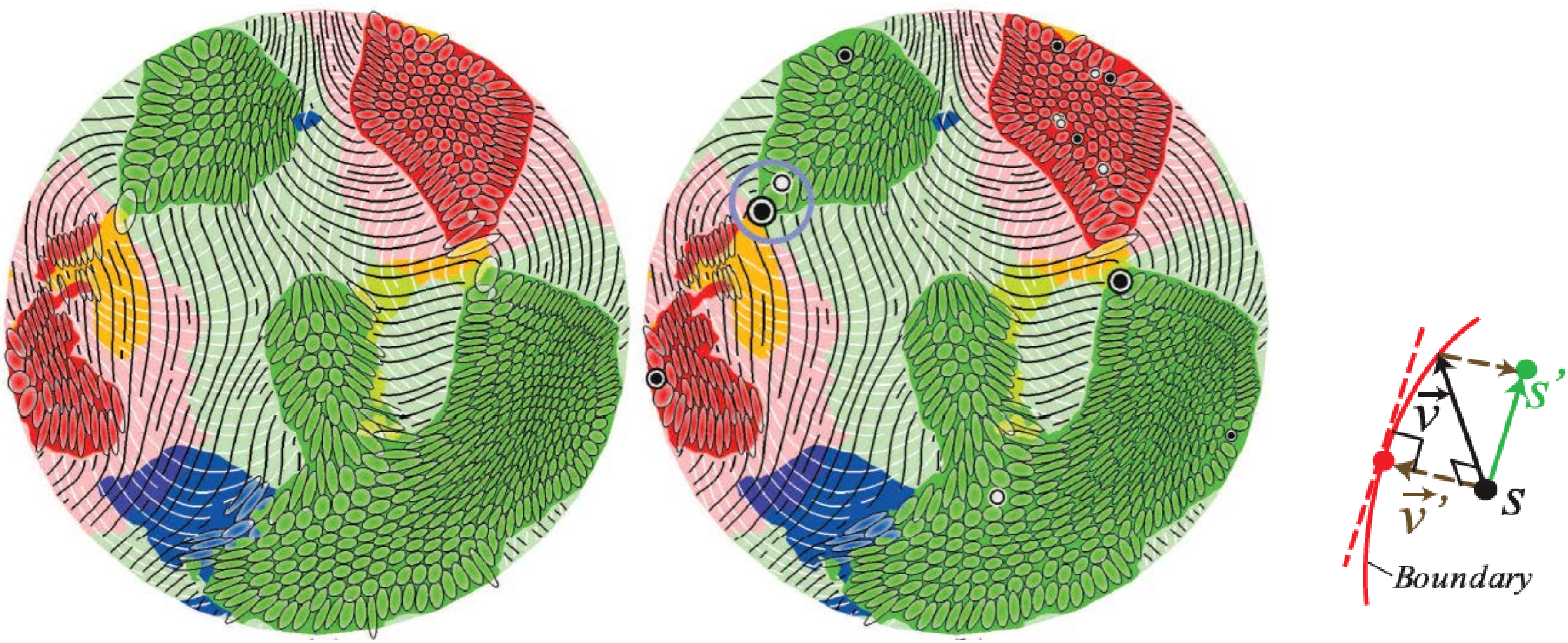


Fig. 19. Mosaic-like image generated by our technique. The metric used for ellipse generation results from the gradient field of the blurred original image. The ellipses automatically align with edges with high gradients and thus emphasize image structure.

[Feng et al. TVCG2008]

Glyph Packing in Bounded Regions



[Chen et al. Vis11]

Additional Readings

- G. Kindlmann. "Superquadric Tensor Glyphs". In Proceedings IEEE TVCG/EG Symposium on Visualization 2004, pages 147-154, May 2004.
- G. Kindlmann and C-F Westin. "Diffusion Tensor Visualization with Glyph Packing." IEEE Trans. on Visualization and Computer Graphics, 12(5):1329-1336, October 2006.
- T. Schultz, G. L. Kindlmann. "Superquadric Glyphs for Symmetric Second-Order Tensors". IEEE Trans. on Visualization and Computer Graphics, Nov/Dec 2010, 16(6):1595-1604.
- Mark Kim, Guoning Chen, and Charles D. Hansen. Dynamic Particle System for Mesh Extraction on the GPU, In Proceeding of 5th Workshop on General Purpose Processing of Graphics Processing Units (GPGPU5), London, March, 2012.
- N Seltzer and G Kindlmann. Glyphs for Asymmetric Second-Order 2D Tensors. Comp. Graph. Forum, 35(3): 141-150, 2016.
- Tim Gerrits, Christian Rössl, and Holger Theisel. Glyphs for General Second-Order 2D and 3D Tensors, IEEE Visualization 2016.

Acknowledgment

- Thanks for materials from
  – Dr. Gordon Kindlmann