Structure Tensor

Computation,
Visualization,
and Application

Vedrana Andersen Dahl

June 8, 2019

1 Computation

In the context of volumetric (3D) image analysis, a structure tensor is a 3-by-3 matrix which summarizes orientation in a certain neighborhood around a certain point.

For example, consider volumetric data showing a bundle of roughly parallel fibers. If we extract two cubes from this volume, mutually displaced along the predominant fiber orientation, the two cubes will have very similar intensities. For two cubes displaced along other orientations the cube intensities would be more different. For this reason, measuring the change of intensities between slightly displaced cubes may be used for determining predominant orientation of imaged structures.

It turns out that, given an initial point and a size of the cube, squared change of intensities may be expressed as $\mathbf{u}^T\mathbf{S}\mathbf{u}$, where $\mathbf{u}$ is the direction of the displacement and $\mathbf{S}$ is 3-by-3 symmetric positive semi-definite matrix – a structure tensor. Finding predominant orientation now amounts to finding $\mathbf{u}$ which minimizes $\mathbf{u}^T\mathbf{S}\mathbf{u}$.

For a more formal derivation, consider a volumetric data V defined on the domain $\Omega \subset \mathbb{R}^3$, where $V(\mathbf{p})$ denotes voxel intensity at the point $\mathbf{p} = [p_x \quad p_y \quad p_z]^T$.

Consider an arbitrary but fixed neighborhood N around a point $\mathbf{p}$, such that $N(\mathbf{p}) \subset \Omega$. Computing

$$D = \sum_{\mathbf{p}' \in N(\mathbf{p})} \left(V(\mathbf{p}' + \mathbf{u}) - V(\mathbf{p}') \right)^2 .$$

gives us a measure of how intensities change when we displace this neighborhood along $\mathbf{u}$, and we will be looking for $\mathbf{u}$ which yields smallest D.

Note that $\mathbf{u} = \mathbf{0}$ trivially minimizes D, and to eliminate the length of displacement from the minimization we will be looking for a solution where $\mathbf{u}$ is a unit vector.

Assuming a small displacement we may use first order Taylor expansion

$$V(\mathbf{p}' + \mathbf{u}) = V(\mathbf{p}') + \begin{bmatrix} V_x(\mathbf{p}') & V_y(\mathbf{p}') & V_z(\mathbf{p}') \end{bmatrix} \mathbf{u}$$

where we use notation $V_x = \frac{\partial V}{\partial x}$, and correspondingly for partial derivatives in y and z direction. We now have

$$D = \sum_{\mathbf{p}' \in N(\mathbf{p})} \left(\begin{bmatrix} V_x(\mathbf{p}') & V_y(\mathbf{p}') & V_z(\mathbf{p}') \end{bmatrix} \mathbf{u} \right)^2$$

Finally, exploiting commutativity of the inner product leads to

$$D = \mathbf{u}^T \sum_{\mathbf{p}' \in N(\mathbf{p})} \begin{bmatrix} V_x(\mathbf{p}') \\ V_y(\mathbf{p}') \\ V_z(\mathbf{p}') \end{bmatrix} \begin{bmatrix} V_x(\mathbf{p}') & V_y(\mathbf{p}') & V_z(\mathbf{p}') \end{bmatrix} \mathbf{u} .$$

So, as earlier claimed, we arrived to expression $D = \mathbf{u}^T \mathbf{S} \mathbf{u}$, where $\mathbf{S}$ is a 3-by-3 matrix – a structure tensor computed in a point $\mathbf{p}$ and using a neighborhood N. That $\mathbf{S}$ is symmetric and positive semi-definite follows directly from the construction of $\mathbf{S}$ (note that $D \geq 0$).

If this requires demystifying, it helps to understand that D is a quadratic function of coordinates of $\mathbf{u}$. The expression $D = \mathbf{u}^T \mathbf{S} \mathbf{u}$ is a compact notation for

$$D = s_{xx} x^2 + s_{yy} y^2 + s_{zz} z^2 + s_{xy} xy + s_{xz} xz + s_{yz} yz ,$$

where $\mathbf{u} = [x \quad y \quad z]^T$ and

$$\mathbf{S} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{xy} & s_{yy} & s_{yz} \\ s_{xz} & s_{yz} & s_{zz} \end{bmatrix} .$$

Due to the construction of $\mathbf{S}$, that values of D along any line through origin are quadratic with zero in origin.

While it is difficult to illustrate D in 3D, its 2D counterpart is elliptic paraboloid, see Fig. 1. Finding unit vector $\mathbf{u}$ that minimizes D in 2D case corresponds to considering values of paraboloid above the unit circle, and finding the direction where values of D are smallest.

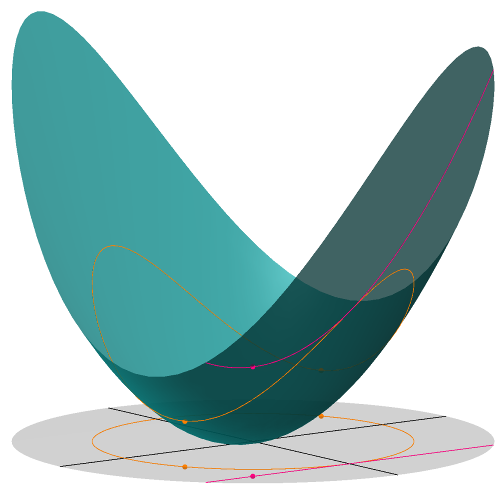

Figure 1: The illustration of minimization problem in 2D, with paraboloid surface defined by values of D for any displacement (x, y). In orange-red we show the values of D above the unit circle, and orange-red dot indicates minimal value which corresponds to predominant orientation. In magenta-red we show the values of D above the line where $y = 1$, and magenta-red dot indicates minimal value which corresponds to optical flow.

Using a compact notation for gradient $\nabla V = \begin{bmatrix} V_x & V_y & V_z \end{bmatrix}^T$, structure tensor is

$$\mathbf{S} = \sum \nabla V \, (\nabla V)^T .$$

Her we imply that structure tensor is computed for every voxel of the volume, i.e. structure tensor is a matrix-valued function over Ω. The summation is conducted over a neighborhood of each voxel, and result will be the same (up to the multiplicative factor) if summation is replaced by an averaging filter.

Two Gaussian filters are usually involved in computing structure tensor, see [1] for detailed description of 2D case. The one Gaussian filter has to do with averaging orientation information in the neighborhood. This can be achieved using a convolution with a Gaussian K_ρ, where parameter ρ, called *integration scale*, reflects the size of the neighborhood. Now we have

$$\mathbf{S} = K_\rho * \left(\nabla V \, (\nabla V)^T \right) .$$

The second Gaussian has to do with computing partial derivatives in gradient ∇V. To make differentiation less sensitive to noise we may convolve the volume with a Gaussian prior to computing derivatives. More efficiently, utilizing derivative theorem of convolution, partial derivatives can be computed by convolving with derivatives of Gaussian. We denote such gradient $\nabla_\sigma V$. The parameter σ is called *noise* *scale*. Expression with both Gaussians is

$$\mathbf{S} = K_\rho * \left(\nabla V_\sigma \, (\nabla V_\sigma)^T\right).$$

Computing structure tensor for each voxel of a volume V involves three steps.

1. Convolve V with derivatives of Gaussian with standard deviation σ to obtain V_x, V_y and V_z. For efficiency, use separability of Gaussian kernel.
2. Using element-wise multiplication compute six volumes V_x^2, V_y^2, V_z^2, $V_x V_y$, $V_x V_z$ and $V_y V_z$.
3. Convolve each of the six volumes with the Gaussian kernel with standard deviation ρ. For efficiency, use separability of Gaussian kernel. The resulting volumes contain per-voxel elements s_{xx}, s_{yy}, s_{zz}, s_{xy}, s_{xz} and s_{yz} of the structure tensor

$$\mathbf{S} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{xy} & s_{yy} & s_{yz} \\ s_{xz} & s_{yz} & s_{zz} \end{bmatrix}.$$

Predominant orientation is found by minimizing Rayleigh coefficient $\mathbf{u}^T \mathbf{S}\, \mathbf{u}$ trough eigendecomposition of $\mathbf{S}$. Being symmetric and positive semi-definite $\mathbf{S}$ yields three positive eigenvalues $\lambda_1 \le \lambda_2 \le \lambda_3$ and mutually orthogonal eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. The eigenvector $\mathbf{v}_1$ corresponding to the smallest eigenvalue is an orientation leading to the smallest variation in intensities, which indicates a predominant orientation in the volume. Note that $\mathbf{v}_1$ is an orientation, and we usually represent it using an unit vector, but this is still not an unique representation since there are two opposite unit vectors sharing the orientation with $\mathbf{v}_1$.

Eigendecomposition of a 3-by-3 real symmetric matrix can be computed efficiently using an analytic approach by Smith [2] which uses an affine transformation and a trigonometric solution of a third order polynomial.

If there is no strong orientation in the volume, all eigenvalues will be similar, and the dominant direction will be influenced by small local variations or the noise in the data. For this reason it is customary to analyze the ratio between eigenvalues to determine the degree of anisotropy in the neighborhood, and how (locally) line-like or plane-like the imaged structure is, see illustration 2. Inspired by diffusion tensor processing [3], we define values, so-called shape measures,

$$c_l = \frac{\lambda_2 - \lambda_1}{\lambda_3}\,, \quad c_p = \frac{\lambda_3 - \lambda_2}{\lambda_3}\,, \quad c_s = \frac{\lambda_1}{\lambda_3}$$

where c_l gives a measure of linearity, while c_p and c_s measure planarity and sphericity. Shape values are positive and sum to 1.

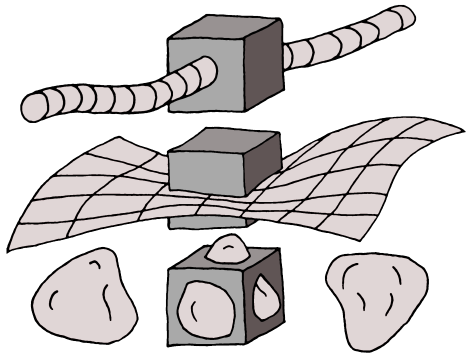

Figure 2: Neighborhoods and structures corresponding to linear, planar and spherical shape. On a linear structure *(top)* cubical neighborhood can move along the predominant direction leading to small change in intensities, while other two orthogonal directions lead to significantly larger and approximately equal change. On a planar structure *(middle)* two directions lead to small and approximately equal change in intensities, while third direction leads to significantly larger change. For a case with no predominant direction *(bottom)* the three orthogonal directions lead to roughly equal changes in intensities.

Normalizing structure tensor by division with tr(**S**) might be desirable. This is because the size of eigenvalues corresponds to the magnitude of the local variation in volume intensities, and might be irrelevant for the analysis. A normalized structure tensor has all eigenvalues smaller than 1. Normalization will not change shape measure and dominant orientation.

Diffusion tensors, as for example measured in magnetic resonance imaging, are closely related to structure tensors. The important difference between structure tensors and diffusion tensors is that diffusion tensors represent the average diffusion rate along a direction. Consequently, the values represented by diffusion tensors will be *large* for the predominant orientation. Strong linearity in diffusion tensor is evident as one eigenvalue being large compared to other two values.

Shape measures for diffusion tensor with *sorted* eigenvalues l_i are

$$m_l = \frac{l_3 - l_2}{l_3}\,, \quad m_p = \frac{l_2 - l_1}{l_3}\,, \quad m_s = \frac{l_1}{l_3}$$

Ellipsoids can be used to visualize diffusion tensors, see Fig. 3. In such a visualization, axis of the ellipsoid are aligned with eigenvector directions and the length of the axis is given by corresponding eigenvalues. The same visualization used directly on structure tensor would be contra intuitive: strong directionality would correspond to a very flat ellipse lying orthogonal to the dominant orientation. Instead, we want to transform structure tensor **S** into a tensor **D** which would lead to an illustrative

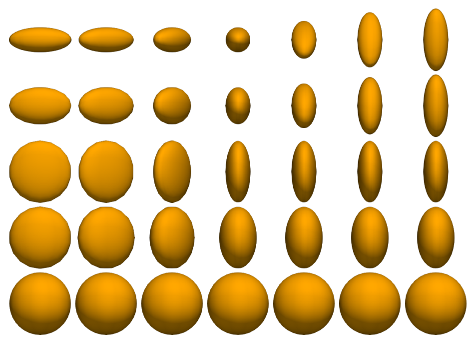

Figure 3: Ellipsoids corresponding to tensors of different anisotropy and direction.

visualization using ellipsoids. Transformation should be such that $\mathbf{S}$ and $\mathbf{D}$ share eigenvectors, and that eigenvalues of $\mathbf{D}$ are computed from eigenvalues of $\mathbf{S}$.

Inspired by Weickert's model for 2D (see [1] Sec. 5.2.1) we define a transformation

$$\mathbf{D} = \exp\left(\frac{-\mathbf{S}}{s}\right)$$

or equivalently, utilizing the properties of matrix exponential,

$$\mathbf{D} = \mathbf{V}\exp\left(\frac{-\mathbf{\Lambda}}{\sigma}\right)\mathbf{V}^{-1},$$

where matrices $\mathbf{V}$ and $\mathbf{\Lambda}$ are given by eigendecomposition $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$. Parameter s controls the anisotropy of $\mathbf{D}$. Large values of s lead to high anisotropy. TODO: can a good s be computed by requiring that shape measures are unchanged? Seems that values around 0.4 are good.

This expression is continent since the matrix exponential of diagonal matrix can be obtained by computing exponentials of elements on the diagonal.

Optical flow, typically used for determining apparent motion of objects in video frames, is closely related to orientation estimation. A sequence of video frames is also represented as 3D structure, but the third (temporal) dimension is treated differently when analyzing object motion. Since time always moves forward, the motion of objects between frames is $[x \quad y \quad 1]^T$. Similar situation occurs when analyzing volumetric 3D data where imaged objects have a predominant direction, e.g. along z axis, and we are concerned with finding a local deviation from this predefined direction. See Fig. 4.

To compute optical flow, we need to minimize $\mathbf{u}^T\mathbf{S}\mathbf{u}$,constrained by $\mathbf{u} = [x \quad y \quad 1]^T$. Note the difference from the problem of finding dominant orientation – here, the displacement in z direction is set to 1. For optical flow, choosing $x = 0$ and $y = 0$ is not trivially solving the equation so we need no additional constrain on the length of the displacement, and the optimal displacement may be of arbitrary length. Furthermore, because of fixed z the solution (x, y) is oriented – the opposite direction $(-x, -y)$ would be a solution for $z = -1$. It is again useful to consider Fig. 1.

For minimization, recall that $\mathbf{u}^T\mathbf{S}\mathbf{u}$ is quadratic in respect to x and y, so the point where partial derivatives vanish can be found by solving a linear system. The system to solve is

$$\begin{bmatrix} s_{xx} & s_{xy} \\ s_{xy} & s_{yy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} s_{xz} \\ s_{yz} \end{bmatrix} . \tag{1}$$

So elements of the structure tensor $\mathbf{S}$ may be used for computing optical flow in each voxel of the volume.

We have now covered optical flow and its connection to structure tensor. However, an explanation of how optical flow is computed usually starts with considering one point $\mathbf{p}$, and a problem of finding x and y such that

$$V(\mathbf{p} + [x \quad y \quad 1]^T) - V(\mathbf{p}) = 0\,.$$

As earlier, we can use first order Taylor expansion yielding

$$V_x(\mathbf{p})x + V_y(\mathbf{p})y + V_z(\mathbf{p}) = 0\,,$$

which is an equation with two unknowns, and solution can not be dertermined.

Lucas-Kanade solution for optical flow involves considering a neighborhood, such that every pixel $\mathbf{p}_i \in N$ contributes with one equation. This gives an overdetermined system

$$\begin{bmatrix} \vdots & \vdots \\ V_x(\mathbf{p}_i) & V_y(\mathbf{p}_i) \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} \vdots \\ V_z(\mathbf{p}_i) \\ \vdots \end{bmatrix}$$

Finding linear least squares solution to an overdetermined system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is equivalent to solving a system $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$, and for our case this is exactly (1)

In summary, structure tensor is a 3-by-3 matrix that can be computed in each volume voxel. The computation of structure tensor requires two parameters: noise scale σ and integration scale ρ. Being symmetric, structure tensor can be represented with 6 scalar values. The most important information extracted from structure tensor is a dominant orientation. Dominant orientation is a unit vector with equivalence relation $-\mathbf{v} \equiv \mathbf{v}$. Shape measures, also extracted from structure tensor, may also be of interest. Shape measures are three scalar, c_l, c_p and c_s, which sum to 1.

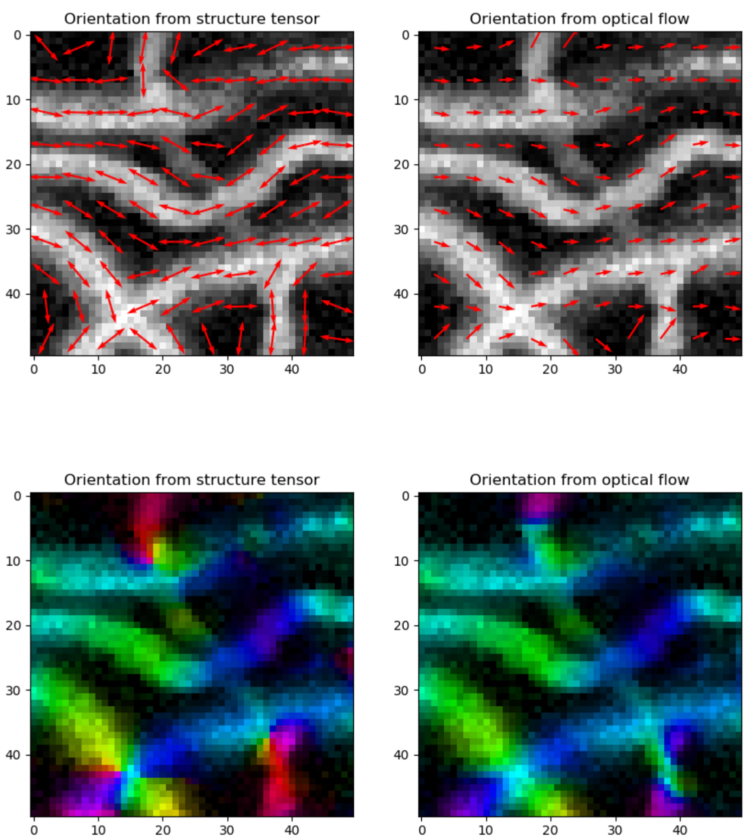

Figure 4: Difference between predominant orientation and optical flow shown on 2D example.

2 Visualization

Having extracted structure tensor and performed its eigendecomposition, following values are available for every volume voxel.

- Voxel intensity V, a scalar value in a certain range.
- Shape measures c_l, c_p and c_s, three scalar values summing to 1.
- Dominant orientation $\mathbf{v_1}$, an orientation vector (unit vector with equivalence relation $-\mathbf{v} \equiv \mathbf{v}$).
- Other information, such as $\mathbf{v_2}, \mathbf{v_3}$, and the values λ_1, λ_2 and λ_3 is also available, but usually of no special interest.

Visualizing this information often requires some care.

Shape measures, being three scalar values can be conveniently represented using three RGB color channels. Voxels with large c_l will appear red, large c_p will be green, and large c_s blue. This is the approach used in Fig. 5.

Predominant orientation, is a 3D unit vector with equivalence relation $-\mathbf{v} \equiv \mathbf{v}$. A common way of visualizing orientation is to use absolute values of vector coordinates, i.e. $|v_x|$, $|v_y|$, and $|v_z|$, as three RGB color channels. This satisfies the required property that$-\mathbf{v}$ and $\mathbf{v}$ map to the same color. In this mapping, here denoted *RGB color mapping*, orientations roughly aligned with x direction will be red, those aligned with y green, and z blue. RGB mapping is favorable for interpretation of the orientations when the imaged object is aligned with the coordinate system (for example, object elongated in z direction). Another example of is medical image analysis which uses anatomical planes (sagittal, coronal and transverse).

Undesirable property of the RBG color scheme is that different orientations map to the same color, for example four orientations corresponding to main diagonals in unit cube all map to gray. In Fig. 5 RGB color mapping gives a useful information on orientation of the structures aligned with coordinate axses. However, a rotation of the object results in a situation with two different (orthogonal) orientations being visualized in similar colors.

A similar situation occurs when analyzing composite material as shown in Fig. 6. In this case, orientation distribution is largely planar, so a fan-like color scheme has been devised for visualizing orientations. In more general cases, icosahedron-based scheme might be favorable.

TODO: Explain fan-line color scheme

$$(r, g, b) = (1 - z^2)\text{hsv2rgb}\left(\arctan\left(y/x\right), 1, 1\right) + 0.5z^2$$

where hsv2rgb is a piecewise linear function describing conversion from HSV color space to RGB color space.

Shape measure and predominant orientation are computed for every voxel in the volume – regardless of whether the voxel is within the object or material which we investigate. When using volume rendering to visualize the extracted measures, if the value is shown in every voxel, the valuable information might be occluded. For this reason it might be beneficiary to combine visualization of extracted measures with the intensity values, which carry information on voxels containing, or not containing, material. A visually pleasing result is obtained if voxels containing no material are shown transparent. Furthermore, predominant orientation is only relevant for voxels exhibiting high linearity, so shape measure may be combined with visualization of predominant orientation,

TODO: Refer to the book [4] for visualization.

3 Applications

Some uses of orientation information obtained from the structure tensor are:

- Volumetric visualization of the shape measures and/or the predominant orientation. See Sec. 3.1.
- Producing distributions fo orientations by collecting orientation vectors in the volume. These distributions live on a half sphere, which can complicate the visualization, comparison, and fitting of distributions.
- Orientation information extracted form volumetric data may be used for modeling, or compared against results from simulation.
- Using local orientation to segment the volume into regions of constant orientation,
- Using local orientation to tune smoothness constraint in MRF segmentation.
- Using local orientation to guide fibre tracking.
- Orientation-aware smoothing.

.

TODO: Expand application.

3.1 Visualizing cotton fibres

Visualizing dominant orientation using may be used for complex fiber systems. In Fig. 7 we colorize tomographic data of plain-weave cotton fabric. The yarns of the fabric are single (one-ply), where cotton fibres are held together by a small amount of twist. Orientation of the cotton fibres is therefore given by the weave pattern and the twist of the yarn. Since fibres mostly lye in the plane of the fabric, fan-based colors are well suited for visualizing fibre orientation.

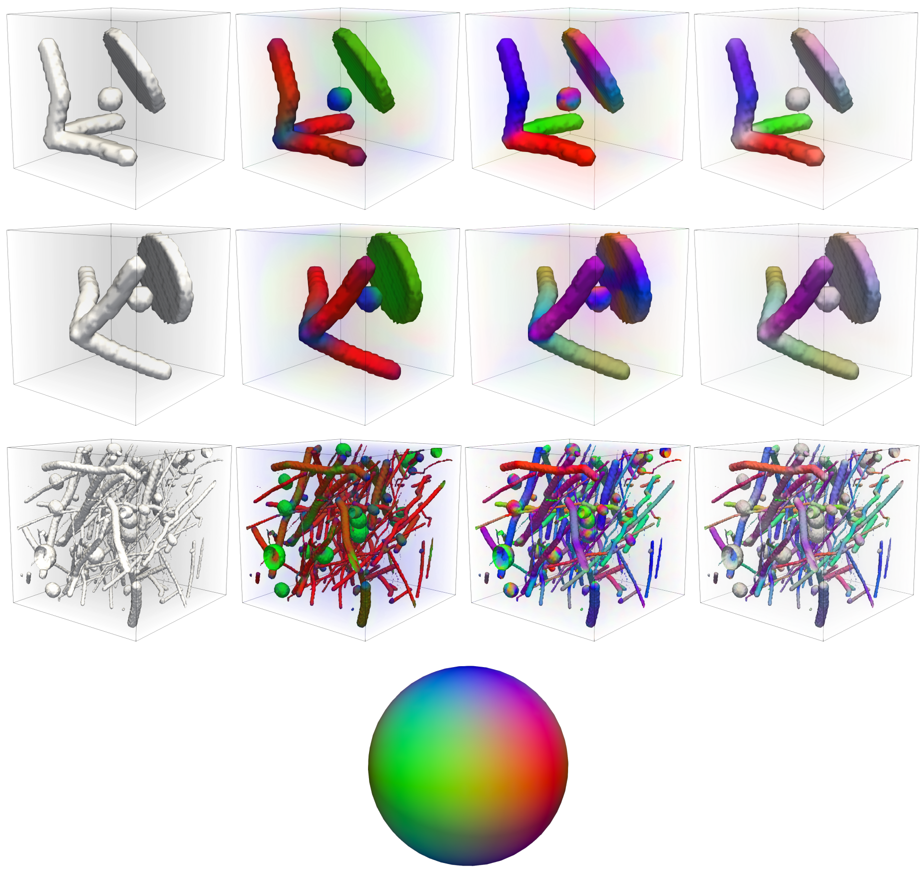

Figure 5: Orientation analysis on a synthetic objects. Top row: a configuration containing linear structures aligned with the coordinate axis, a sphere and a disc. Second row: the same configuration as above, but rotated. Third row: linear structures with random orientations and spherical structures. Forth row: RGB color mapping used for visualizing orientations. Left column: Input data. Second column: shape measures shown using colors (red - linearity, green - planarity, blue - sphericity). Third colum: dominant orientation shown using RGB color mapping. Fourth column: dominant orientation weighted by linearity.

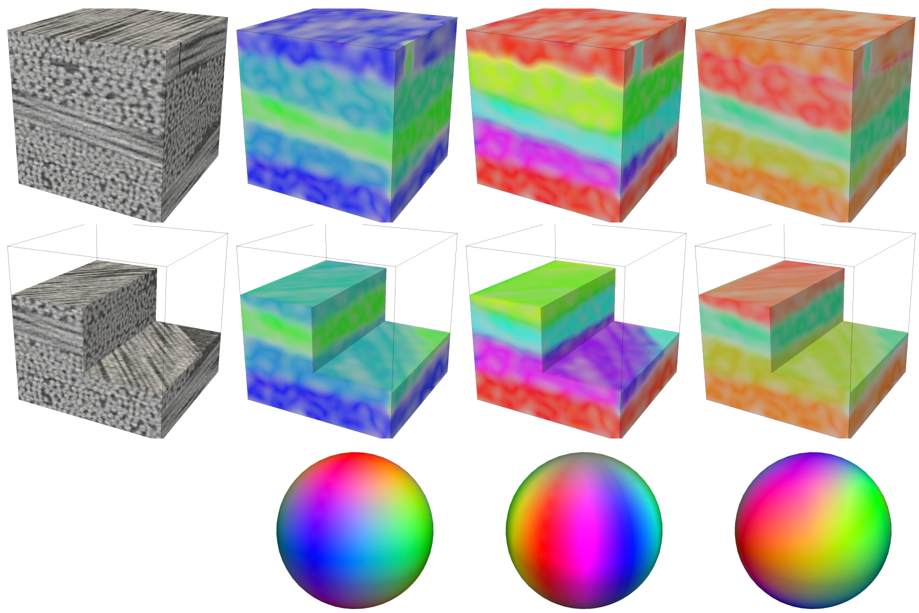

Figure 6: Orientation analysis of composite material visualized using three different color mappings. Top row: a volume with layered fibre structure. Middle row shows cut trough the volume to reveal the orientations of the different layers. Bottom row shows different color mapping for visualizing orientations. First column shows input data. Second column uses RGB color mapping. Third column uses fan-based colors. Forth column uses icosahedron-based colors.

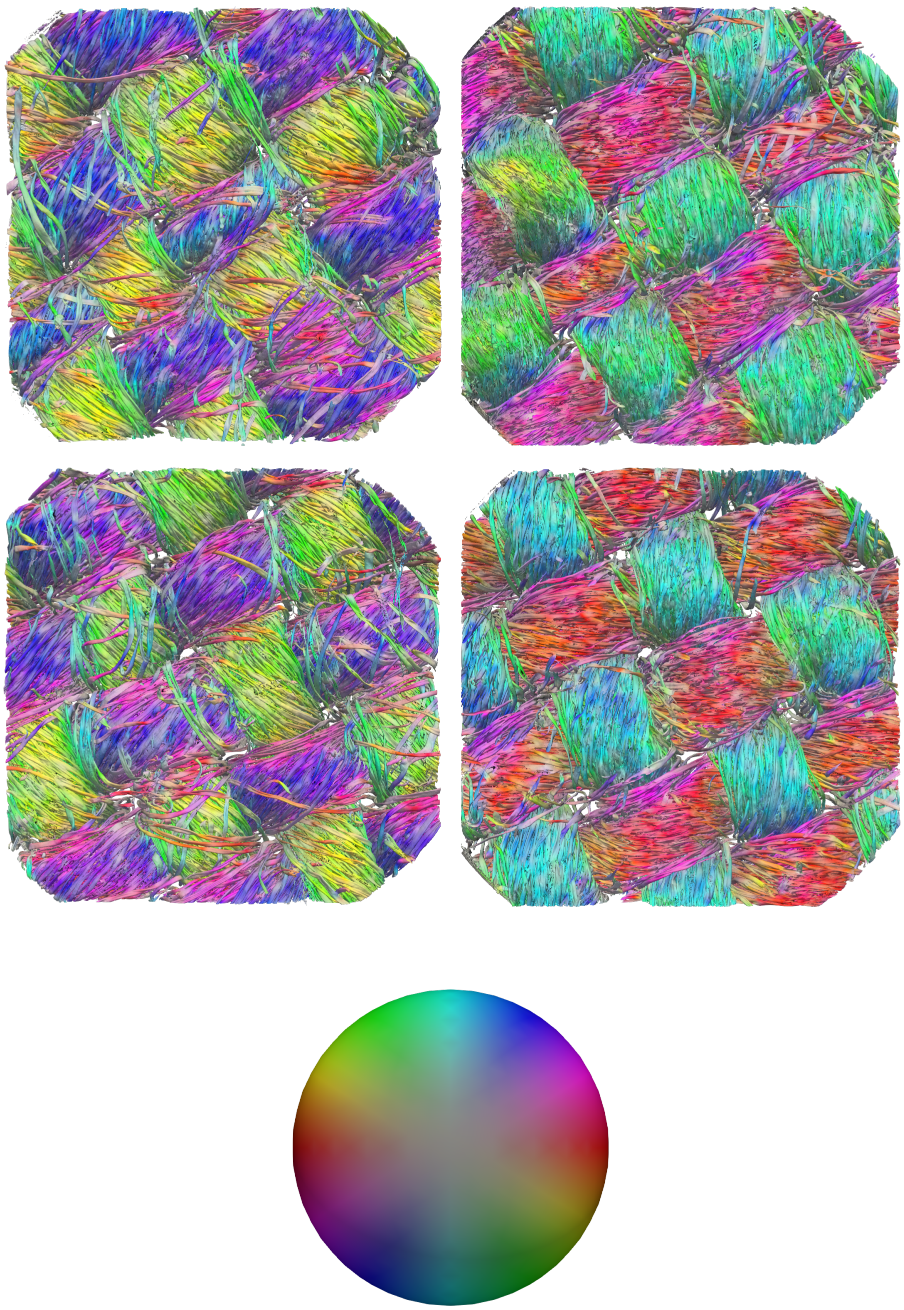

Figure 7: Using colors to visualize dominant orientation of the cotton fibres in the woven fabric. Color reveals the direction of the twist holding the cotton fibres together in a yarn.

References

[1] J. Weickert, *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998, vol. 1. [Online]. Available: https://www.mia.uni-saarland.de/weickert/Papers/book.pdf

[2] O. K. Smith, “Eigenvalues of a symmetric 3× 3 matrix,” *Communications of the ACM*, vol. 4, no. 4, p. 168, 1961. [Online]. Available: https://dl.acm.org/citation.cfm?id=366316

[3] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis, “Processing and visualization for diffusion tensor MRI,” *Medical image analysis*, vol. 6, no. 2, pp. 93–108, 2002.

[4] J. Weickert and H. Hagen, *Visualization and processing of tensor fields*. Springer Science & Business Media, 2005.