Vision, Modeling, and Visualization (2011)
Peter Eisert, Konrad Polthier, and Joachim Hornegger (Eds.)

Particle-Based Anisotropic Sampling for Two-Dimensional Tensor Field Visualization

A. Kratz, N. Kettlitz and I. Hotz

Zuse Institute Berlin (ZIB), Germany

Abstract

We present a particle-based approach to generate unstructured distributions of elliptical samples. Size and shape of the samples are determined by a local metric that is derived from a two-dimensional tensor field. In contrast to previous methods, we propose the use of an anisotropic Delaunay triangulation of particle positions. It guarantees exact neighbor computations and provides a good means for an explicit and automatic control of prominent holes and overlaps, which otherwise would result in unpleasant visualizations. We use the final distribution to compute a generalized Voronoi diagram, which represents a novel and flexible visualization technique for two-dimensional tensor fields. Via texturing of Voronoi regions, many possibilities arise to design the final image.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Since the 1980s, aperiodic point distributions with varying density and blue-noise properties are a central research topic in computer graphics [LKF*08]. Methods range from simple dart throwing techniques [Coo86] to real-time tile-based methods [KCODL06, LD08]; applications range from sampling theory to mesh generation and illustrative rendering. In general, these methods focus on isotropic samples and are hard to extend to work with anisotropic samples. The latter is particularly true for the very efficient tile-based approaches.

In this work, we present a particle-based approach to generate an unstructured distribution of elliptical samples whose size and shape is determined by a local metric derived from a given tensor field. A force function exerts attractive and repelling forces depending on the particle distances. The system has converged when all elliptical samples centered at the particle positions are closely packed without intersecting each other. We use an anisotropic Delaunay triangulation of particle positions to guarantee meaningful and effective neighbor detections and to avoid long-distance particle interactions. It further serves as basis for an explicit control of holes and overlaps in the distribution.

On the basis of previous work in the context of meshing [SYI00], anisotropic sampling [FHHJ08] and glyph placement [KW06, HSH07], we have developed a stable and efficient method especially designed to handle ellipses that exhibit varying size. We show that two-dimensional tensor field visualization can profit in multiple ways from anisotropic sample distributions. The most obvious example is the placement of tensor glyphs. Previous work [KW06, HSH07, FHHJ08] has impressively demonstrated that glyph depictions become more informative and visually more pleasing if they are not just placed at the underlying grid positions. We further show that anisotropic sample distributions can be used as input for visualization techniques that are similar to line integral convolution (LIC). Finally, we present a novel tensor visualization method that is based on generalized Voronoi diagrams. Being computed on the stable particle configuration, the diagram mimics a continuous view of the tensor field having a painterly character. Furthermore, Voronoi regions can be texturized and, thus, provide a variety of possibilities to design the final visualization.

2. Related Work

We focus on relevant previous work that is related to the creation of anisotropic samplings. In general, these approaches start with an initial sampling distribution that is iteratively refined. We distinguish geometry- and particle-based approaches. For more information about the generation of

DOI: 10.2312/PE/VMV/VMV11/145-152

isotropic point distributions, we refer the reader to existing reviews [LKF*08, LD08, LLC*10].

Geometry-based approaches oftentimes rely on Lloyd relaxation [Llo82, DFG99]. In the context of tensor field visualization, [FHHJ08] presented an approach that combines ideas from sampling theory [DFG99] and mesh generation [LS03] for the creation of anisotropic Voronoi cells. As this approach only depends on attractive forces, it provides a stable solution. A limitation is the lack of an explicit control of overlaps and a low convergence behavior for bad initial sample distributions. A strength of geometry-based approaches, in general, is that they are mostly parameter-free.

Particle-based approaches rely on the definition of a force function that controls the movement of particles: Repelling forces avoid overlaps and attracting forces close holes between the samples. Closely related to our work are methods from geometry processing for the generation of anisotropic triangulations [BH96, SYI00]. In contrast to our approach, which has the goal to visualize a *given* tensor field, these methods explicitly *define* a tensor field that specifies an anisotropic spacing between mesh vertices. The goal is to create an efficient, adaptive mesh. In the work of [SYI00], such a mesh is represented by an anisotropic Delaunay triangulation. Therefore, the authors generalized the Delaunay criterion by incorporating anisotropy in the *circumcircle test*. For diffusion tensors, [KW06] adapted ideas from [SYI00]. In their work, the positive-definite tensor field is mapped to a potential energy field that determines inter-particle forces. Hlawitschka et al. [HSH07] extended the approach of [KW06] focusing on improved initial sampling and interactivity through the use of an isotropic Delaunay triangulation for fast neighbor-queries. A strength of particle-based approaches is that they are easier to extend to work on implicit surfaces. A limitation is that they are sensitive to parameter choices so that these need to be handled carefully.

3. Assumptions and Requirements

Input of our algorithm is a two-dimensional tensorfield comprised of second-order symmetric tensors. To reconstruct tensors at arbitrary particle positions $p = (x,y) \in \mathbf{R}^2$ over a domain $D \subset \mathbf{R}^2$ with boundary ∂D, we use bilinear component-wise interpolation. The basis of the computation of unstructured sampling positions is the definition of a metric that reflects the tensor field [HFH*04]. The local metric $m(x,y)$ defines the size and shape of an elliptical sample centered at the particle position $p = (x,y)$. It is given by a two-by-two symmetric positive-definite matrix

$$m(x,y) = \begin{pmatrix} m_{11}(x,y) & m_{12}(x,y) \\ m_{12}(x,y) & m_{22}(x,y) \end{pmatrix}. \quad (1)$$

In the following, we will simply use the term *sample* instead of *elliptical sample*.

We have designed our particle system such that it can handle samples of varying size across D. However, to obtain a reasonable visualization of the tensor field it is important that the samples can be considered as representatives of the area they cover. This is only the case if the variation of the tensor field, compared to the size of the sample, is relatively low. For previous work in the context of diffusion tensors [KW06, HSH07], this requirement was less important as the focus was on normalized glyphs of relatively uniform size that mainly differ in anisotropy.

4. Isotropic vs. Anisotropic Space

In this paper, we map an arbitrary type of tensor to a *metric tensor* (Equation (1)) that describes distances between particle positions $p \in D$. It can be considered as distortion of a circle into an ellipse. Hence, we distinguish between the *undistorted* Euclidean space and the *distorted* metric space.

A special case arises when the metric tensor field does not vary across D, i.e., $m = m(x,y) = const$. For this case, we can transform the distorted space into the undistorted Euclidean space via linear transformation using the inverse metric m^{-1}. Common methods for isotropic point distributions could be used in this space.

In this work, we require our technique to handle spatially varying anisotropies (Section 3). Therefore, a global transformation of the complete space is not possible. For some computations, however, it is advantageous to apply a local transform (assuming that the field is locally constant) into the undistorted Euclidean space via multiplication with $m(q)^{-1}$. To approximate m, we use the metric at the barycenter q of particles p_i in a neighborhood n

$$m(q) = m(\frac{1}{n}\sum_{i=1}^{n} p_i). \quad (2)$$

To determine, e.g., the metric m at the barycenter $q \in D$ of a triangle, we use its $n = 3$ corners p_1, p_2, p_3 (Figures 3 and 4).

5. Particle-Based Anisotropic Sampling

Starting point of our algorithm is an initial sample distribution (Section 5.1) that is iteratively refined (Section 5.3) until all forces are balanced or a maximum number of iterations is reached. An anisotropic Delaunay triangulation (Section 5.4) enables fast and correct neighbor-queries and serves as a basis for an automatic population control (Section 5.5).

5.1. Initial Sample Distribution

We use generalized relaxation dart throwing to create an initial sample distribution that has the following properties: (1) The distances between particles are well-balanced so that significant holes and overlaps are minimized. (2) The samples cover the domain densely. (3) It is unstructured so that periodic arrangements are reduced.

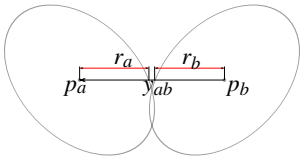


Figure 1: *Approximated distance computation between the neighboring samples p_a and p_b.*

According to [SYI00], we sample the given domain D in order of dimension: First, samples are placed at the corners of D, which are kept fixed throughout the whole simulation. Second, samples on the edges of D are distributed via subdivision. To avoid a structured placement of edge samples, they are repositioned using a one-dimensional particle movement (Section 5.3). Now, these positions are kept fixed throughout the whole simulation, which leads to a closed boundary treatment that avoids particles leaving the domain. Furthermore, we do not need to define an external force [KW06, HSH07]. To distribute samples inside the domain, we use dart throwing [Coo86], i.e., samples are successively *thrown* into D. They are rejected if they would intersect with any previously accepted ellipse. To generalize this algorithm to work with anisotropic samples, we use the same distance measure (Equation (4)) as for the force function. The original dart throwing algorithm terminates if the rejection-rate exceeds some threshold. In this work, we apply relaxation dart throwing [ME92]. Starting with larger samples, their size is successively reduced if the rejection-rate becomes too large, i.e., if the ratio of the number of accepted against the number of rejected samples falls below a user-specified threshold value ε (to create the images depicted in this paper, we used $\varepsilon = 0.01$). To determine ellipse sizes, we use the relative Poisson disk radius r given as $r = \rho r_{max}$, with $\rho \in [0,1]$ [LD05]. In our case r_{max} is implicitly given by the ellipses. For relaxation dart throwing we start with a value of $\rho = 0.9$ which is reduced by 0.05 whenever the rejection rate becomes too large. We stop reducing sample sizes when $\rho = 0.75$. As said in [LKF*08], ρ needs to be large but not too large to avoid regular configurations ($\rho = 1$ corresponds to a hexagonal arrangement). An advantage of this algorithm is that it terminates when a desired number of samples has been accepted, which is one of our requirements. In order to estimate this number N, we compute an average value of the determinant det_{avg} of m, which gives us a hint of the average size of elliptical samples. Given the area A of the domain D, an appropriate number N of samples then is $N = A/det_{avg}$.

5.2. Distance Measure

The force function is a function of particle distances exerting attracting and repelling forces. Previous work approximate geodesic distances [FHHJ08, LWSF10] via distance computation in the distorted space. These measures, however, are not symmetric, which would require an additional conflict check [LWSF10]. According to [SYI00], we define the optimal distance d_{opt} between sample positions p_a and p_b to be the sum of the two lengths r_a and r_b to the ellipses' surface in the direction of the connecting line $y_{ab} = p_a - p_b$ (Figure 1)

$$d_{opt} = \underbrace{\frac{\|m(p_a)\cdot y_{ab}\|}{\|y_{ab}\|}}_{r_a} + \underbrace{\frac{\|m(p_b)\cdot y_{ab}\|}{\|y_{ab}\|}}_{r_b}. \tag{3}$$

The force function (Section 5.3) then is defined over the relation d of the inter-particle distance y_{ab} and d_{opt}

$$d = \frac{\|y_{ab}\|}{d_{opt}}. \tag{4}$$

Thus, the optimal distance with respect to the force function is always 1 independent from the local metric. Figure 1 shows that this approximation does not avoid all kinds of intersections. However, we find that this measure is a good compromise with respect to the generation of holes and overlaps. Preventing, e.g., the two samples shown in Figure 1 from intersection would result in a bigger hole at the top.

5.3. Force Function

The force function f, being defined over the distance d (Equation (4)), controls the particle movement. The goal is to compute stable particle distances that are characterized by elliptical samples that are closely packed and non-intersecting. Therefore, we define a piecewise cubic spline $f(d)$ with the following properties: (1) It has a zero crossing at $d = 1$ so that particles with optimal distance do not move. (2) If $]0 < d < 1[$, repelling forces are exerted pushing particles apart. (3) If $]1 < d < (1+\gamma)[$, attracting forces are exerted pulling particles together. The parameter γ controls the range of attracting forces. As we only consider particle interactions of neighboring samples, excluding long-distance interactions, our method is not very sensitive with respect to γ. It can be set on a fixed value of $\gamma = 0.5$. The force F_a, which acts on particle a, is determined from the sum of the forces from the neighboring particles $F_a = \sum_{b,b\neq a} f_{ab}$. The equations of motion [KW06, HSH07]

$$\frac{d^2p}{dt^2} = F_a + C_{drag}\frac{d_p}{d_t} \tag{5}$$

are solved numerically via Euler integration using a stepsize of $\Delta t = 0.2$. Since the force scales with sample size this leads to small steps for small particles and larger steps for large samples. The drag parameter C_{drag} counteracts the particle motion to avoid oscillations and to guarantee numerical stability [KW06].

5.4. Anisotropic Delaunay Triangulation

To guarantee a stable solution for samples of varying size, interactions between particles that are no immediate neigh-

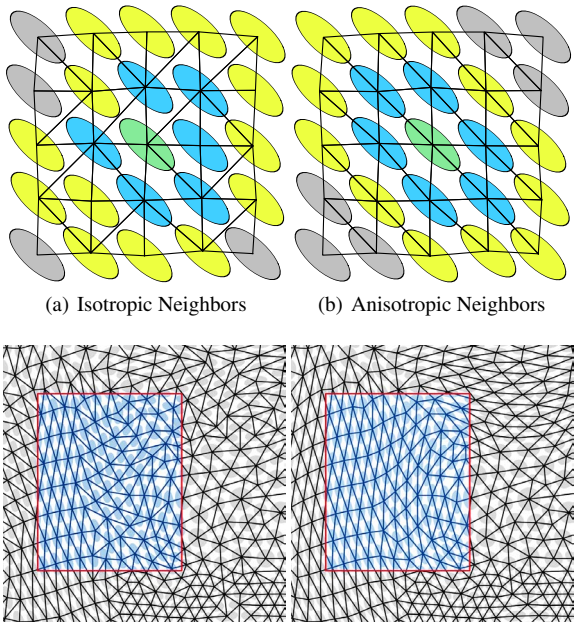
bors have to be avoided. Space-subdivision into bins of constant size [KW06], for example, does not fit this requirement. Mainly for optimization purposes, [HSH07] use an isotropic Delaunay triangulation of particle positions to overcome the problem of choosing good bin sizes. Thus, fast neighbor-searches are guaranteed by traversing the outgoing edges of a particle. However, simple isotropic Delaunay triangulations do not guarantee correct neighbor detections (Figure 2). Using the two-edge distance as proposed in [HSH07] further would result in clutter for elliptical samples that strongly vary in size.

For this reason, we compute an anisotropic Delaunay triangulation of particle positions during simulation. For a constant field, it would be possible to apply a global transform into the isotropic space (Section 4) and use the common *circumcircle test*, which says that a triangle is valid if no points lie in the circle that encloses the triangle. The generalized Delaunay criterion, which we use in this work, applies a local transform *before* the test is performed (Figure 3). For the transformation of the tensors at the $n = 4$ particle positions, we use the metric tensor at the barycenter of these four points (Equation (2)).

5.5. Automatic Insert/Delete Operations

The estimation of the number of particles for the initial placement (Section 5.1) can be challenging. Especially considering our requirement of a stable particle system that can handle samples of varying size, N is only a rough estimate. Therefore, we include automatic delete and insert operations that adjust the number of samples during simulation. This further enables an explicit control of holes and overlaps. According to [SG95], we identify over- and underpopulated regions via analyzing inter-particle forces. As soon as F (Section 5.3) exceeds (falls below) a given threshold, particles are removed (inserted). Contrary to geometry-based approaches, which only depend on attractive forces, particle-based methods are more sensitive to holes. Thus, overpopulated regions appear only rarely.

To automatically insert samples in underpopulated regions, we exploit the generalized Delaunay triangulation (Section 5.4). Therefore, for each triangle, the fill-rate, being the ratio of the covered area and the triangle's area, is computed. Assuming a hexagonal packing as gold standard, the optimal fill-rate is given as $\delta_{fill} = \frac{\pi}{2\sqrt{3}} \approx 90.7\% \approx 0.9$. To compute the fill-rate $\delta_{fill^\triangle}$ of a triangle covered by elliptical samples, we need to determine the sum of all three triangle-sample intersections and compare it to the triangular area (Figure 4). For anisotropic samples, the intersection would be hard to compute. Therefore, we transform the samples at the triangle's corners into the isotropic space using the inverse tensor at the triangle's barycenter q (Equation (2)). If $\delta_{fill^\triangle}$ is significantly below δ_{fill}, a new sample is inserted at q. In this work, we inserted a new sample whenever $\delta_{fill^\triangle} \leq 0.6$.



(a) Isotropic Neighbors (b) Anisotropic Neighbors

(c) Isotropic Delaunay Mesh (d) Anisotropic Delaunay Mesh

Figure 2: *Top row: Neighbor-computation of the current sample (green) for isotropic (a) and anisotropic (b) Delaunay triangulations of a constant field. In this example, the isotropic Delaunay triangulation and the usage of a one-edge distance (blue samples) can lead to wrong neighbor detections and undesired particle interactions (a). Using a two-edge distance (yellow samples) does not provide a valid solution, either, and can lead to cluttering if the samples vary in size (a). For anisotropic Delaunay triangulations, a one-edge distance is sufficient (b). Bottom row. Figures (c,d) demonstrate the superior quality of the anisotropic Delaunay triangulation, which aligns naturally with the major eigenvector field. The differences are particularly pronounced in the highlighted areas (c,d).*

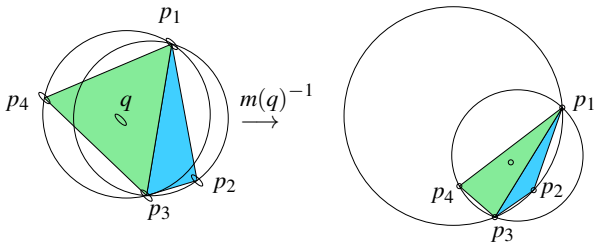


Figure 3: *The generalized Delaunay criterion applies a circumcircle test after transformation of the triangle corners into the undistorted Euclidean space. The shown triangles do not fulfill this criterion. Valid combinations are p_2, p_3, p_4 and p_1, p_2, p_4.*

6. Generalized Voronoi Textures

Based on the stable particle positions, we compute a generalized Voronoi diagram. Let $\{p_i \in D, i \in I\}$ be the set of stable particle positions, the *Voronoi sites*, where I is the corresponding index set. A Voronoi region $Vor(p_i)$ of a point p_i then is defined as the set of all points $P \subset D$ that are at least as close to p_i than to any other point $p_j \in D$ with $j \in I$

$$Vor(p_i) = \{P \in D | d(p_i, P) \leq d(p_j, P),\ with\ i \neq j\}. \quad (6)$$

For the isotropic case, a Voronoi diagram can be simply computed as dual graph of the Delaunay triangulation. For the anisotropic case, this property is not trivially generalizable. As mentioned in [FHHJ08], the most obvious way is to use a geodesic distance measure, which, however, is computationally expensive. Anisotropic Voronoi diagrams can further require the handling of *orphans* [LS03, FHHJ08], i.e., the part of a Voronoi region that does not contain its barycenter. This can happen for anisotropic Voronoi diagrams which might consist of not connected components. In our case, such orphans appear if the size of neighboring samples vary significantly, which happens only rarely. As we use the Voronoi diagram for visualization purposes only, an approximation of Voronoi regions is sufficient and the rare appearance of orphans is acceptable.

We perform the computation of the Voronoi regions in a fragment shader to enable interactive rendering updates and flexible texture mapping. The computation is initiated by rasterizing the anisotropic Delaunay triangulation. Shader input are the six vertex coordinates p_i and indices of the current triangle and its three adjacent ones. For each rasterized fragment p_f, we now determine its Voronoi cell by computing the minimum distance to its neighboring vertices using the following distance measure [FHHJ08, BWY08]

$$d(p_i, p_f) = \sqrt{(p_f - p_i)^\top m(p_i) \cdot (p_f - p_i)}. \quad (7)$$

Finally, the fragment is colored according to the cell index of its nearest Voronoi site. Furthermore, Voronoi regions can be texturized (Figure 5(f), (g)) which enables a high flexibility to design the visualization. To determine the texture coordinates $(s,t) \in [0,1]$ for the current region, we transform the fragment coordinates $p_f \in D$ into the local coordinate system of the current Voronoi region and rotate the local coordinates via linear transformation with the inverse eigenvector matrix E of the tensor at the region's Voronoi site.

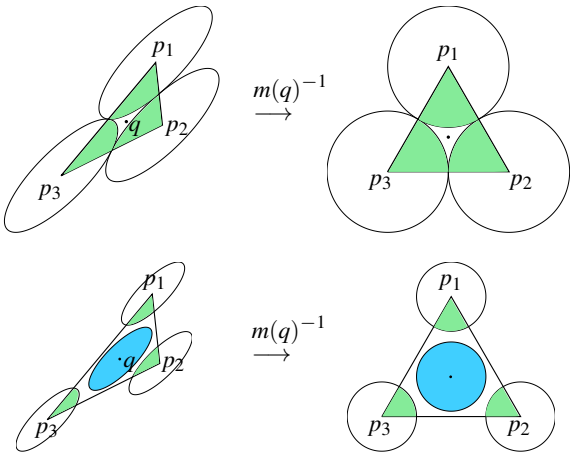


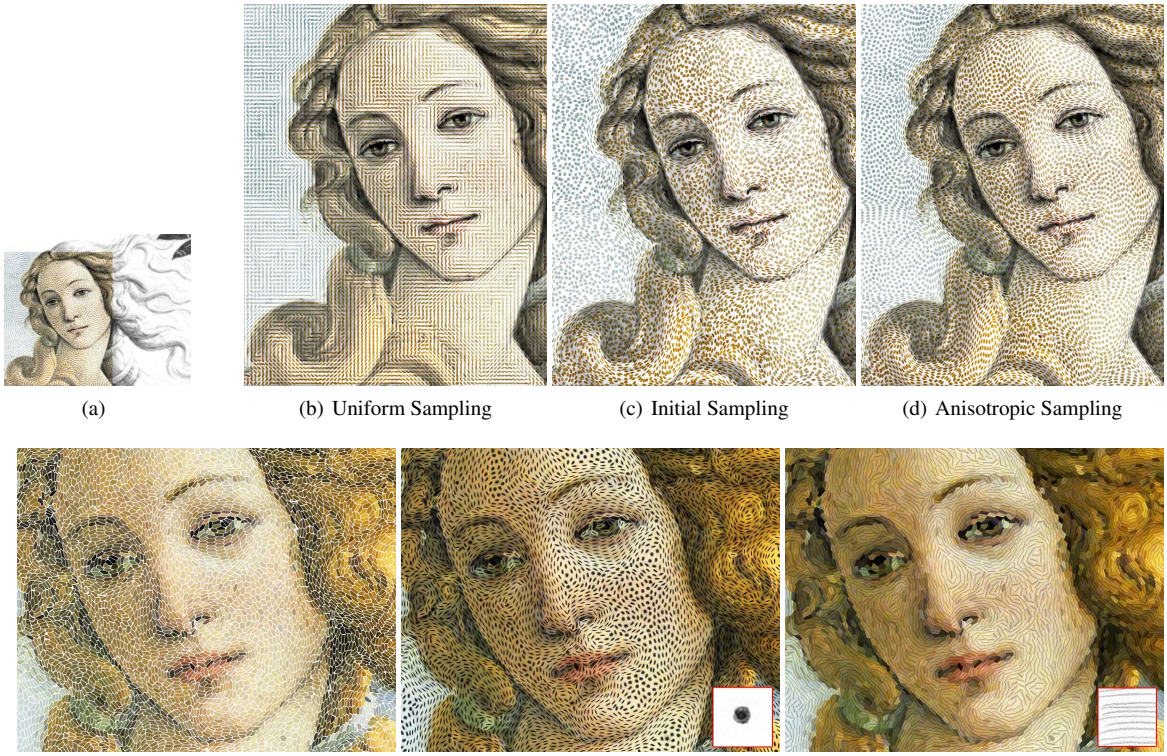
Figure 4: *The fill-rate of a triangle, computed as ratio of the covered area (green) and the triangle area in isotropic space (right), determines if a new sample needs to be inserted. Top row: Triangle that exhibits an optimal fill-rate. Bottom row: A new sample (blue) is inserted due to the triangle's poor fill-rate.*

N	Initial Seeding	100 Iterations	1000 Iterations
1000	7s	0.3s	3s
10000	90s	4s	40s

Table 1: *Timing statistics in seconds measured on an 2.6 GHz Intel Xeon Quad Core. The performance of our technique mainly depends on the desired sampling resolution N and the initial seeding. Particle movement is very fast.*

7. Results and Applications

We have presented a particle-based approach to generate unstructured distributions of non-intersecting elliptical samples. The time needed for generation mainly depends on the desired sampling resolution and the number of iterations needed until a stable placement has been reached (Table 1). For our datasets an update of the Delaunay triangulation every eighth iteration is sufficient to guarantee precise neighbor detections and high-quality results. The frequency of updates is independent from the number of samples. Instead it depends on the variation of sample size, i.e., more frequent updates are required if the sample size across D varies strongly. The most time-consuming process it the initial sampling. In our examples, we found a stable configuration after 100 iterations. For comparison, Kindlmann and Westin [KW06] computed 2000 iterations to create the 2D results in their work, which took about 8 minutes. Usable results appeared after 400 iterations. Unfortunately, they do not say anything about the number of glyphs. Hlawitschka et al. [HSH07] mention that their packing becomes stable after *several hundred steps*. For $N = 1000(10000)$ glyphs and 100 iterations they need $2s(55s)$. As all of these measures were carried out on different PCs and in different years, the times should be considered only as a rough comparison value. In fact, more interesting are the number of iterations. In the following, we show that the distribution is well suited for several applications in two-dimensional tensor field visualization: It can be used directly for the placement of tensor glyphs (Section 7.1), as input noise texture for the creation of *fabric textures* (Section 7.3), and to create *Voronoi textures* - a novel tensor visualization technique (Section 7.2). To cre-



(a) (b) Uniform Sampling (c) Initial Sampling (d) Anisotropic Sampling

(e) Voronoi Regions (f) Texturized Voronoi Regions (g) Texturized Voronoi Regions

Figure 5: *In these images, size and shape of elliptical samples are determined from a metric based on the image gradient and its orthogonal vector field. To compute the sampling distribution for these images,* 1000 *iterations were needed. Top row: The images show the samples placed at the grid positions (b), before particle movement (c) and the result of our technique (d). Bottom row: The images show visualizations that have been created on the basis of the Voronoi diagram. Drawing the borders of the Voronoi regions results in mosaic-like images (e). For the texturized images (f,g), we have used two different textures as input (depicted in the right-hand corner).*

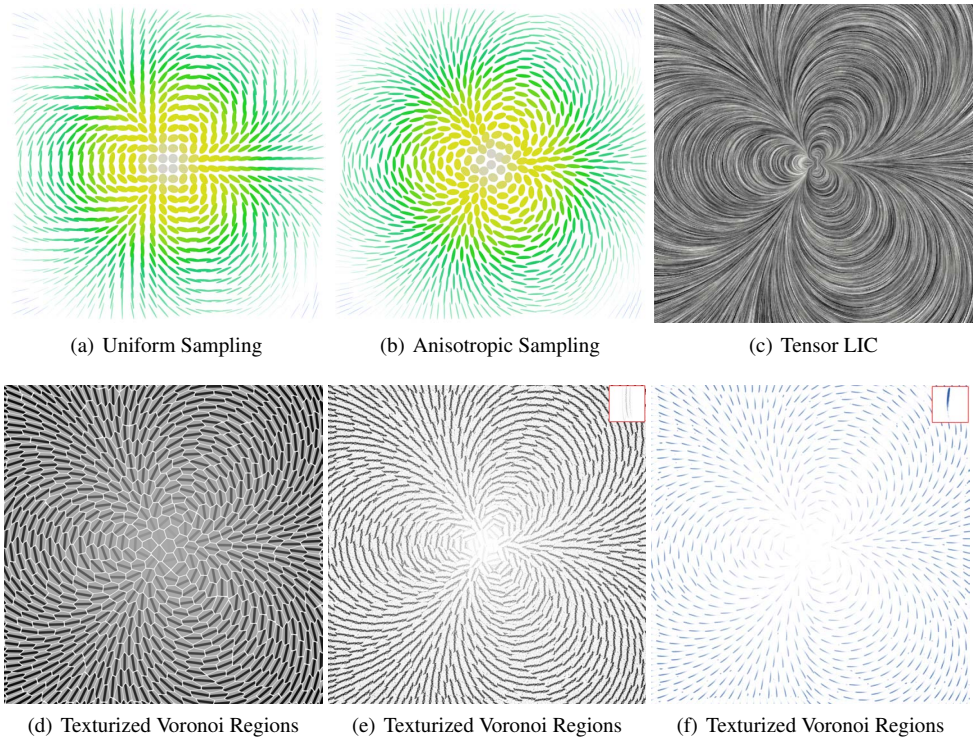
ate the images in this paper, we have used an anti-symmetric mapping to create the metric tensor field [HFH*04].

7.1. Glyph Positioning

For the placement of tensor glyphs a good distribution should have the following properties: (1) Dense, to provide all relevant details and the evolution of tensors across the field. (2) Unstructured, as the human visual system is very sensitive to the recognition of patterns. (3) Evenly, minimizing prominent holes and overlaps. Figures 5 (d) and 6 (b) demonstrate the superior quality of glyph-based depictions when the glyphs are centered at the particle positions of the stable configuration. The images become more informative and visually pleasant. Figure 5(c) further shows the good quality of our initial sample distribution.

7.2. Voronoi Textures

Tensor glyphs can encode the full tensor information at discrete positions. We have shown that glyph depictions provide a more continuous view when the glyphs are positioned w.r.t. an anisotropic sample distribution (Section 7.1). However, still only local information is provided. We, therefore, present a novel tensor visualization technique that is based on generalized Voronoi diagrams. Voronoi regions are distorted with respect to the local metric encoding its full information. Thus, a more continuous visualization of the tensor field is achieved, which has a painterly character (Figure 5(e)). The distribution that is used to compute the Voronoi diagram should have the same properties as for glyph positioning. Holes and overlaps, however, have a lower impact on the final visualization. Main design parameters are the metric and textures that are mapped into Voronoi regions. Through the use of textures, we have a variety of possibilities to design the final image. A hatch pattern, for example, creates images that have a painterly character (Figures 5(f), (g)). Using a single line as input texture (Figures 6(e), (f)) produces images that are similar to tensor LIC (Figure 6(c)).



(a) Uniform Sampling (b) Anisotropic Sampling (c) Tensor LIC

(d) Texturized Voronoi Regions (e) Texturized Voronoi Regions (f) Texturized Voronoi Regions

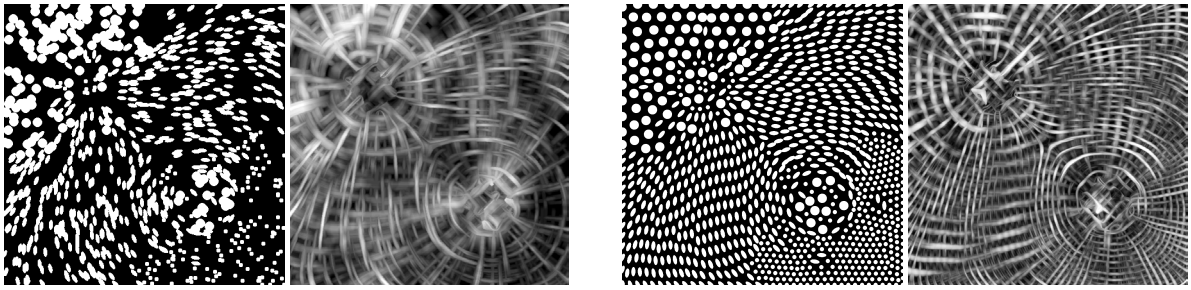
Figure 6: *The images show various depictions of an analytic tensor field that has one degenerate point in the center. To compute the sampling distribution for these images,* 100 *iterations were needed. Top row: If samples are placed at grid positions, the grid's pattern becomes visible (a) which disturbs the perception of the field's continuous behavior (c). Being placed at the particle positions computed with our technique, the glyph depiction becomes more informative and visually pleasing (b). Bottom row: The images show texturized Voronoi regions using three different line textures (d,e,f). The color is more transparent in isotropic regions where directions are not defined.*

7.3. Fabric Textures

We have shown that the use of line textures in combination with Voronoi diagrams enables, for example, the depiction of a distinguished direction. A two-dimensional tensor field, however, has two directions. Hotz et al. [HFH*04] have presented a global texture-based method that is especially designed to represent the central features of stress tensors, namely compression and expansion. Therefore, a LIC-like texture for *every* eigenvector field is created and then both resulting textures are blended, which results in images that resemble a fabric: thinner fibers indicate compression and thicker fibers indicate expansion. For this method, sparse noise textures are preferred (Figures 7(b), (d)). However, especially sparse noise textures can suffer from holes which are visible in the final image (Figures 7(a), (b)). A good distribution that is used as input noise texture should have the following properties: (1) Even, to ensure a uniform brightness across the field. Otherwise, brighter regions would be emphasized and, thus, might be perceived as more important. (2) Unstructured, to avoid patterns to become visible in the final visualization. Figures 7 (a-d) compare the results using a random noise texture as input (Figures 7(a,b)) and one computed with our technique (Figures 7(c,d)). It is hard to say, which image is visually more pleasing. However, using a noise distribution computed with our technique results in a *cleaner* visualization where contrasts are more balanced. Furthermore, it has less the appearance of being three-dimensional, which can lead to misinterpretations.

8. Discussion and Conclusion

Our main design goal was the handling of elliptical samples of varying size, which are more sensitive to clutter due to wrong neighbor detections. Therefore, we propose the use of an anisotropic Delaunay triangulation, which guarantees correct neighbors. We have presented an automatic insert operation that detects over- and underpopulated regions during simulation. Due to this automatic control, holes in the initial distribution can be filled during simulation. However, the quality of the initial distribution influences the algorithm's convergence time. In general, 100 iterations are sufficient to create a stable particle configuration. We never needed more than 1000 iterations.



(a) Input (random noise) (b) LIC texture (random noise) (c) Input (our method) (d) LIC texture (our method)

Figure 7: *These images show a slice from a stress tensor dataset from a finite-element simulation that describes the elastic behavior of a block on which a pushing and a pulling force have been applied. To compute the sampling distribution (c) used as input to create a fabric texture (d),* 100 *iterations were needed. Holes/clutter in the random noise input (a) result in darker/brighter regions in the final image (b), which gives the impression of a three-dimensional field. In (d) thinner fibers in compressive regions can be distinguished clearly from thicker fibers in tensile regions.*

We have shown three tensor visualization techniques that can benefit from such a distribution. The most common application is the placement of tensor glyphs. Moreover, it can be used as input texture for the creation of fabric textures. Finally, we have presented a novel tensor visualization technique that is based on an approximation of generalized Voronoi diagrams. Combined with texture mapping it can offer many possibilities to design the visualization.

A generalization of the approach to work with three-dimensional tensor fields might be possible but we think that this would not be very useful to create meaningful visualizations. Instead, we plan an extension of the particle-based sampling to work on implicit surfaces extracted from three-dimensional fields. Furthermore, as the creation of the initial sample distribution is the most time-consuming process, we will optimize it by using the Delaunay triangulation already during relaxation dart throwing.

References

[BH96] BOSSEN F. J., HECKBERT P. S.: A pliant method for anisotropic mesh generation. In *5th Intl. Meshing Roundtable* (1996), pp. 63–74. 2

[BWY08] BOISSONNAT J.-D., WORMSER C., YVINEC M.: Locally uniform anisotropic meshing. In *Proceedings of the twenty-fourth annual symposium on Computational geometry* (New York, NY, USA, 2008), SCG '08, ACM, pp. 270–277. 5

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *ACM Transactions on Graphics* (1986), 51–72. 1, 3

[DFG99] DU Q., FABER V., GUNZBURGER M.: Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review* (1999), 637–676. 2

[FHHJ08] FENG L., HOTZ I., HAMANN B., JOY K.: Anisotropic noise samples. *IEEE Transactions on Visualization and Computer Graphics* (2008), 342–354. 1, 2, 3, 5

[HFH*04] HOTZ I., FENG L., HAGEN H., HAMANN B., JOY K., JEREMIC B.: Physically based methods for tensor field visualization. In *VIS '04: Proceedings of the conference on Visualization '04* (2004), IEEE Computer Society, pp. 123–130. 2, 6, 7

[HSH07] HLAWITSCHKA M., SCHEUERMANN G., HAMANN B.: Interactive glyph placement for tensor fields. In *ISVC (1)* (2007), pp. 331–340. 1, 2, 3, 4, 5

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive wang tiles for real-time blue noise. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)* (2006), 509–518. 1

[KW06] KINDLMANN G., WESTIN C.-F.: Diffusion tensor visualization with glyph packing. *IEEE Transactions on Visualization and Computer Graphics 12* (2006), 1329–1336. 1, 2, 3, 4, 5

[LD05] LAGAE A., DUTRÉ P.: A procedural object distribution function. *ACM Trans. Graph. 24* (October 2005), 1442–1461. 3

[LD08] LAGAE A., DUTRÉ P.: A comparison of methods for generating poisson disk distributions. In *Computer Graphics Forum, 27* (2008), pp. 114–129. 1, 2

[LKF*08] LAGAE A., KAPLAN C. S., FU C.-W., OSTROMOUKHOV V., KOPF J., DEUSSEN O.: Tile-based methods for interactive applications. SIGGRAPH 2008 Class, SIGGRAPH 2008, Los Angeles, USA, August 2008. 1, 2, 3

[LLC*10] LAGAE A., LEFEBVRE S., COOK R., DEROSE T., DRETTAKIS G., EBERT D. S., LEWIS J. P., PERLIN K., ZWICKER M.: State of the art in procedural noise functions. In *EG 2010 - State of the Art Reports* (2010). 2

[Llo82] LLOYD S. P.: Least squares quantization in pcm. *IEEE Transactions on Information Theory* (1982), 129–137. 2

[LS03] LABELLE F., SHEWCHUK J. R.: Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *Proceedings of the nineteenth annual symposium on Computational geometry* (2003), pp. 191–200. 2, 5

[LWSF10] LI H., WEI L.-Y., SANDER P. V., FU C.-W.: Anisotropic blue noise sampling. In *ACM SIGGRAPH Asia 2010 Papers* (2010), pp. 167:1–167:12. 3

[ME92] MCCOOL M., EUGENE F.: Hierarchical poisson disk sampling distributions. In *Proceedings of the conference on Graphics interface '92* (1992), pp. 94–105. 3

[SG95] SHIMADA K., GOSSARD D. C.: Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing. *Annual Autumn Meeting of IPSJ* (1995). 4

[SYI00] SHIMADA K., YAMADA A., ITOH T.: Anisotropic triangulation of parametric surfaces via close packing of ellipsoids. *Int. J. Comput. Geometry Appl.* (2000), 417–440. 1, 2, 3