

# A Novel Algorithm for Computing Riemannian Geodesic Distance in Rectangular 2D Grids

Ola Nilsson<sup>1</sup>, Martin Reimers<sup>2</sup>, Ken Museth<sup>1</sup>, and Anders Brun<sup>3,\*</sup>

<sup>1</sup> Department of Science and Technology, Linköping University, Sweden

<sup>2</sup> Department of Informatics, University of Oslo, Norway

<sup>3</sup> Centre for Image Analysis,  
Swedish University of Agricultural Sciences, Sweden

anders@cb.uu.se

**Abstract.** We present a novel way to efficiently compute Riemannian geodesic distance over a two-dimensional domain. It is based on a previously presented method for computation of geodesic distances on surface meshes. Our method is adapted for rectangular grids, equipped with a variable anisotropic metric tensor. Processing and visualization of such tensor fields is common in certain applications, for instance structure tensor fields in image analysis and diffusion tensor fields in medical imaging.

The included benchmark study shows that our method provides significantly better results in anisotropic regions and is faster than current state-of-the-art solvers. Additionally, our method is straightforward to code; the test implementation is less than 150 lines of C++ code.

## 1 Introduction

The computation of distances in manifolds is important in both academic and industrial applications, e.g. computational geometry [1], seismology, optics, computer vision [2], computer graphics and image analysis [3,4]. Another cross disciplinary example is the versatile level set method [5] that propagates interfaces embedded in scalar distance fields. Often the computations are performed on rectangular grids. One way to define distance is the *eikonal equation*

$$\|\nabla d(\mathbf{x})\|_2 = 1, \quad \mathbf{x} \in \Omega, \quad (1)$$

which is a nonlinear Hamilton-Jacobi PDE with boundary condition  $d|_{\partial\Omega} = 0$ . It defines  $d(\mathbf{x})$  as the Euclidian distance from  $\mathbf{x}$  to an implicit source,  $\partial\Omega$ . Substituting the right-hand side in equation (1) with a positive scalar “speed function” that depends on position,  $F(\mathbf{x})$ , gives a nonuniform eikonal equation that is often used in applications. However, this is not the most general case. The nonuniform and anisotropic case is described by adapting the norm  $\|\cdot\|$  to an arbitrary and spatially varying metric tensor field  $g_{ij}(\mathbf{x})$ . A metric tensor is a positive definite quadratic form that defines the scalar product between tangent vectors in a point  $\langle \mathbf{v}, \mathbf{u} \rangle_g = \mathbf{v}^T g_{ij} \mathbf{u}$ . This, in turn, interprets the length

---

\* Corresponding author.

of a tangent vector in a point as  $\|\mathbf{x}\|_g = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_g}$ . Thus equation (1) can be generalized to take the metric into account

$$\|\nabla d(\mathbf{x})\|_g = 1, \quad \mathbf{x} \in \Omega. \quad (2)$$

## 2 Previous Work

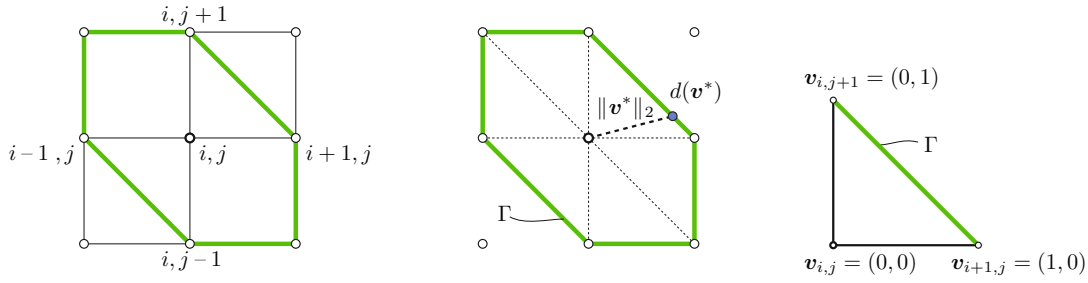
Considerable attention has been devoted to the computation of distances in an anisotropic setting. Early work include shading from shape [6,7] and the salience distance transforms [8]. In optics the problem is typically solved using a shooting type of algorithm, like in [9] where a method based on solving ODEs and tracking individual rays of light is used. This type of Lagrangian approach is not efficient when it comes to computing a a continuous approximation over the rays' embedding domain, which is the case for a distance map.

It is well known that a distance function is the so called *viscosity solution*, see e.g. [10,1]. Much of the previous work on anisotropic distances is based on the discretization in [11] and involves a Godunov-type approximation of the gradient. The solution (per grid point) is selected from a set containing solutions to 8 quadratic equations in 2-D. In the original paper, an iterative sweeping update scheme was used to find the global solution in  $O(kn)$  time where  $n$  is the number of grid points in the discretization and  $k$  the number of sweeps. This algorithm updates all grid points the same number of times, which is sub-optimal. A different iterative update scheme was proposed in [12] based on the same discretization. It keeps an unsorted list holding the expanding solution-front and updates nodes on a need-to basis. Even though the run time is not bounded, the method is fast in practice.

If the distance value at each node only depends on neighboring nodes with smaller values, it is possible to construct an update scheme that finds all distances in the domain in one single pass. This causality property is the basis for the well known *fast marching method* (FMM) [13], see also [14], which approximates isotropic distances on a regular grid in  $n$  steps with  $O(n \log n)$  time complexity. In [1], this is extended to parametric three dimensional manifolds. Two recent one-pass methods are proposed in [15] and [16]; the former is uses a reaction-diffusion setting and the latter is based on control theory. In [17] a modified version of the FMM scheme was proposed that is supposed to be more accurate in certain cases. Finally, [18] finds the exact distance over a triangle mesh in  $O(n^2 \log n)$  time.

## 3 Basic Scheme for 2-D Grids

The problem of computing distances from a source over a regular grid is conceptually simple. It can be broken down to successive applications of grid point updates. If the distance computed is valid locally, and a monotonic update is enforced, then it can be proved that distances will converge globally[19].



**Fig. 1.** Left: A generalized setup around grid point  $i, j$  with local coordinates  $(0, 0)$ . Middle: Assuming an implicit triangulation (dotted) we define a boundary  $\Gamma$  (green) for our domain. Local optimality ensures that the distance at origin is minimal,  $d(\mathbf{0}) = \min \|\mathbf{v}^*\|_2 + d(\mathbf{v}^*)$  for  $\mathbf{v}^* \in \Gamma$ . Right: One of the four triangular simplices resulting from splitting the 4-connectivity neighborhood. Unit grid spacing,  $h = 1$ , is assumed.

A local update can be constructed by placing the update point in a local origin and minimizing the following nonlinear functional

$$d(\mathbf{0}) = \min_{\mathbf{v}^* \in \Gamma} \|\mathbf{v}^*\|_2 + d(\mathbf{v}^*), \tag{3}$$

for any domain with a closed boundary, requiring that the source is outside of the domain. For a grid with an implicit triangulation a possible setup is shown in figure 1. Equation (3) states local optimality and was discretized in [19,20] for triangle meshes. The solution at  $\mathbf{0}$  finds a point  $\mathbf{v}^*$  on the boundary for which the sum  $\|\mathbf{v}^*\|_2 + d(\mathbf{v}^*)$  is minimal.

The update step of this type of method can be summarized as follows, given a grid point  $(i, j)$  and its neighborhood (see figure 1):

1. Split the domain into 6 right angled triangular simplices
2. Solve equation (3) for each of the triangles and set  $d_{i,j}$  to the minimum

The outer loop then repeats this for all grid points (in some order) until convergence. As the 6 different configurations and their solutions are similar, we only depict one simplex illustrated in figure 1 (right).

The update value is found in two steps. First, interpolate the distance values at  $\mathbf{v}_{i+1,j}$  and  $\mathbf{v}_{i,j+1}$  along  $\Gamma$  and denote the interpolant  $\tilde{d}(\mathbf{v}^*)$ . Then compute the minimum of the sum of the edge interpolant and the distance to the edge:

$$d_{i,j} = d(\mathbf{v}_{i,j}) = \min_{\mathbf{v}^* \in \Gamma} \|\mathbf{v}^*\|_2 + \tilde{d}(\mathbf{v}^*). \tag{4}$$

As seen from equation (3), the minimization of (4) would yield the exact solution if the interpolation was exact. Finding and differentiating exact interpolants for complex wave fronts is nontrivial and potentially unstable, thus we constrain our search to interpolants of order one and two.

**Linear Sources.** An early discretization was proposed in 1985 by Gonzalez and Rofman [20]. They deployed a linear interpolant in equation (4), which is

straightforward to minimize. However, we can equally well find the minimizer geometrically, without differentiation. We use the following definition:

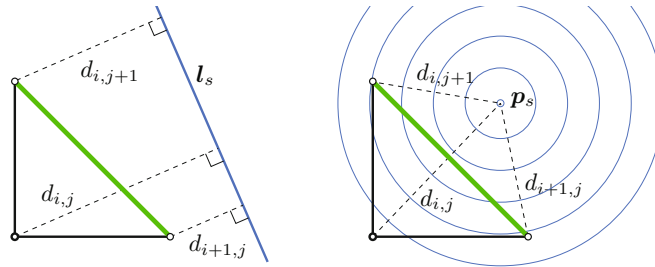
**Definition 1.** *A minimizer to equation (3) is also the closest distance from the local origin to the wavefront source.*

This is illustrated in figure 2. The minimizer (source-origin distance) is easy to compute for a linear source. Following the figure, first find the source line  $\mathbf{l}_s$  using the distances  $d_{i+1,j}$  and  $d_{i,j+1}$ , then compute the closest distance between  $\mathbf{l}_s$  and origin. A similar (linear) approach is taken in the popular fast marching method (FMM) [13], which was derived from a finite difference discretization.

**Point Sources.** A linear scheme can not accurately handle curved wave fronts. This is especially evident close to a point source where curvature is high and linearization is a poor approximation, see figure 2 (right). Both [17] and [19] recognized the need for a more accurate scheme for point sources, and independently came up with similar solutions.

From the geometry of the problem it is straightforward to find a closed form solution. Placing the source point  $\mathbf{p}_s$  in figure 2 (right) reduces to a two-circle intersection problem [19], and the minimizer is found as  $\|\mathbf{p}_s\|_2$ .

A point source approximation will, of course, not accurately find minimizers if the wavefronts are linear functions. Thus care needs to be taken when choosing source type. For this conference paper, we will focus mainly on point sources since they are most commonly found in applications, e.g. when geodesic distance is sought between two points in a manifold.



**Fig. 2.** (Left) The minimizer using linear interpolation can be found as the closest distance from origin to the source line  $\mathbf{l}_s$ , for a linear wave front. (Right) For a point source the minimizer is the distance between origin and the source,  $\mathbf{p}_s$

**Accuracy and Algorithms.** We have seen that it is possible to find the update as a distance in the plane. For a 2-D grid it is possible to place the source *exactly*. Thus the update values, and hence the solutions, computed above are exact for their respective type of source. For other types of sources this type of scheme introduces a first order approximation [20,19].

For a flat metric with a linear interpolant and a regular grid, a one-pass algorithm for the global solution of (4) can be found using the fact that larger

values always depend solely on smaller values. However, when using the quadratic interpolant for point sources, this is no longer true. Additionally, obtuse triangle configurations can (and are likely to) occur when the metric is not flat, as we will see in the next section. A robust and practical algorithm is described in [19] that efficiently addresses these problems. Let  $S$  be the source that is initialized to distance 0. Then add the index (or indices) of  $S$  to the sorted candidate set  $C$ . Push and pop adds and extracts indices of  $C$ , and  $\min C$  refers to the indices associated with the smallest distance.

```

while  $C \neq \emptyset$  do
     $(i, j) \leftarrow \text{pop}(\min C)$ 
    for  $(k, l) \in \text{neighbor}(i, j)$  do
         $d'_{k,l} \leftarrow \text{update}(k, l)$ 
        if  $d'_{k,l} < d_{k,l}$  then
             $d_{k,l} \leftarrow d'_{k,l}$ 
             $C \leftarrow \text{push}(k, l)$ 
    
```

From an implementation point of view, this scheme has a complexity similar to the popular FMM. Due to space constraints we refer to [19] for more details.

### 4 Anisotropic Speed Functions

The update of distance values in section 2 was derived in the Euclidean plane. However, the geometric construction indicates that we may incorporate a metric into equation (4) by adapting the edge lengths of the triangles. The generalized metric dependent minimization reads

$$d_{i,j} = \min_{\mathbf{v}^* \in I} \|\mathbf{v}^*\|_g + \tilde{d}(\mathbf{v}^*), \tag{5}$$

where  $\|\cdot\|_g$  is measuring the geodesic length of  $\mathbf{v}^*$  under  $g$ . Geodesic distance between two points,  $\mathbf{a}$  and  $\mathbf{b}$ , on a manifold is defined by a minimization over all curves  $\gamma$  (in the manifold) joining the two points

$$d_g(\mathbf{a}, \mathbf{b}) = \inf_{\gamma(0)=\mathbf{a}, \gamma(1)=\mathbf{b}} L(\gamma), \text{ where } L(\gamma) = \int_{t=0}^{t=1} \sqrt{\gamma'(t)^T g_{ij}(\gamma_x(t), \gamma_y(t)) \gamma'(t)} dt.$$

If two points are close, like two vertices in a triangle, the space-variant metric can be approximated by a constant metric. Thus,

$$d_g(\mathbf{a}, \mathbf{b}) \approx \sqrt{(\mathbf{b} - \mathbf{a})^T (g_{ij}(\mathbf{a}) + g_{ij}(\mathbf{a}))/2 (\mathbf{b} - \mathbf{a})}, \tag{6}$$

where  $\mathbf{a}$  and  $\mathbf{b}$  are the vertices of a triangle edge <sup>1</sup>. We could now insert a second order interpolant into equation (5), together with the metric distance equation (6) to find minimizers. Directly doing so, however, involves finding roots of a

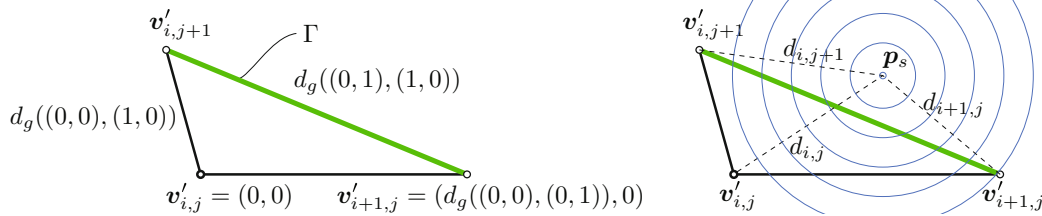
---

<sup>1</sup> A similar scheme was described for isotropic speed functions in [19]. This reference also mentions the possibility of using anisotropic speed functions.

fifth order polynomial; this is feasible but impractical. Since the solution is still only a linear approximation, we propose a simpler and more intuitive strategy.

Using the linear approximation of metric distance is approximately equivalent to linearly “flatten” the simplex under  $\|\cdot\|_g$ , as shown in figure 3. From this follows that minimizing equation 5 is approximately the same as minimizing equation 4 for a transformed simplex.

A consistent positioning of vertices, in Euclidean space, using the metric is shown in figure 3 (left). Since the transformed simplex is flat, the Euclidian distance function can be used to find distances from the origin to  $\mathbf{p}_s$ , and thus approximately solve equation (5). In the rare cases when the edge lengths fails to satisfy the triangle inequality, we have used Dijkstra update is deployed instead.



**Fig. 3.** Left: The simplex is (linearly) flattened using the metric  $g$ . For a consistent positioning we first place  $\mathbf{v}'_{i,j}$  in origin, then  $\mathbf{v}'_{i+1,j}$  along the positive x-axis. The position of  $\mathbf{v}'_{i,j+1}$  is then found using inter-vertex distances. Right: The minimizers to equation (5) can be found in a flattened space as the Euclidian distance from origin to  $\mathbf{p}_s$ , respectively. This is treated analogously for line sources.

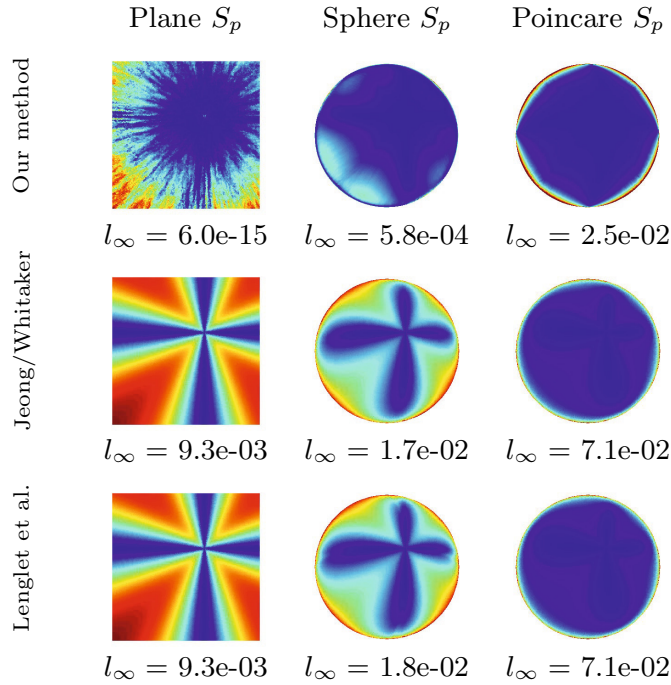
## 5 Results and Benchmarks

To test the accuracy of geodesic distance computation, it is desirable to have a set of representative “test manifolds” for which distances are known analytically. In a 2D-manifold, the local geometry is completely characterized by the Gauss curvature,  $K$ . From a geometric point of view, up to a scaling of the metric, there are three principal cases to test:  $K > 0$ ,  $K = 0$  and  $K < 0$ , corresponding to local isometry with the sphere, the plane and the hyperbolic plane. For this reason, these manifolds have been chosen as test manifolds. In addition, we also included the cone, which has zero curvature in all points except at the apex. We compare our method against two state-of-the-art anisotropic eikonal solvers. One from Lenglet *et al.* [16] and one by Jeong and Whitaker [12].

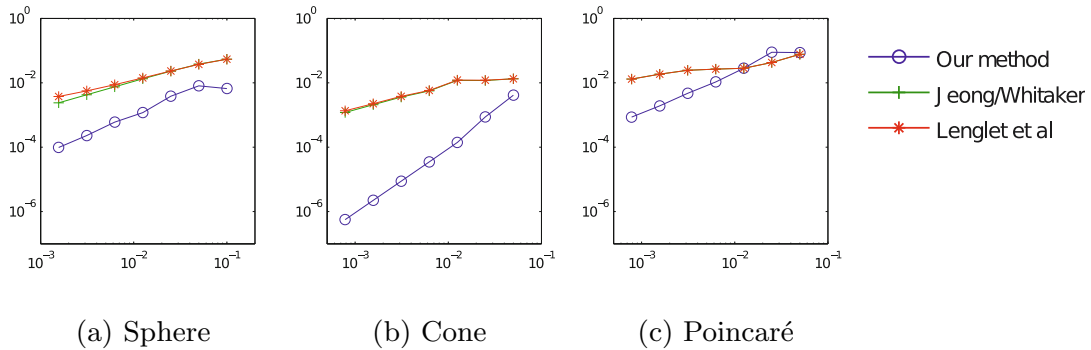
The test grid consists of  $400 \times 400$  grid points with spacing  $h = .005$ . For the metrics (see appendix A) that only live in the unit disk, we have restrained computations outside a radius of 0.98 to avoid degenerate tensors at the border. We place the sources exactly on grid points and report the mean of the absolute error, that is  $l_1/n$ , the max error  $l_\infty$ , and the time for the computations. The findings are listed in table 1 and figure 4. We have also performed a numerical convergence study that is reported in figure 5. Finally, we investigated the smoothness of the derivatives of the distance function for our and competing works, see figure 6.

**Table 1.** Comparison of accuracy and speed against competing work.  $S_p$  means that point sources are used and  $S_l$  corresponds to line sources. Also see figure 4.

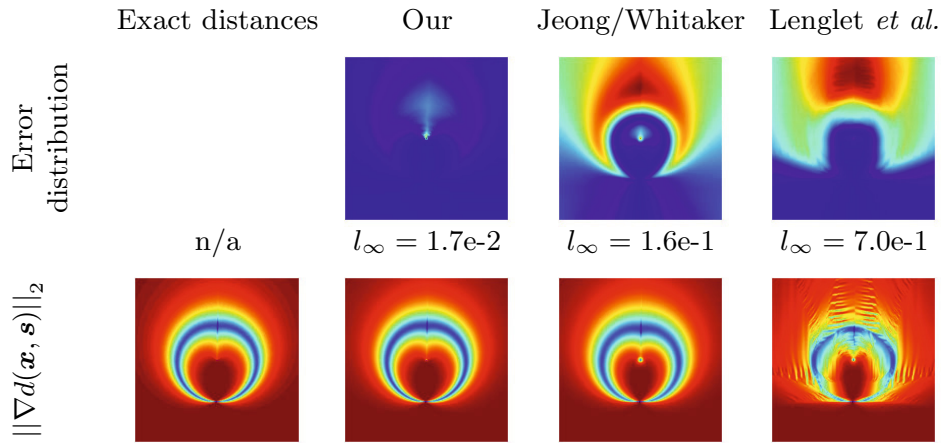
Metric	Our method			Jeong and Whitaker			Lenglet et al.		
	$l_1/n$	$l_\infty$	Time	$l_1/n$	$l_\infty$	Time	$l_1/n$	$l_\infty$	Time
Plane, $S_p$	7.2e-16	6.2e-15	.34s	5.1e-03	9.3e-03	1.4s	5.1e-03	9.3e-03	1.9s
Sphere $S_p$	6.3e-05	5.8e-04	.82s	6.2e-03	1.7e-02	1.9s	6.3e-03	1.8e-02	3.4s
Poincaré $S_p$	1.3e-03	2.5e-02	.54s	1.0e-02	7.1e-02	1.6s	1.0e-02	7.1e-02	2s



**Fig. 4.** Error distribution plots. Color shows absolute error (blue to red) and is linearly scaled to fall in  $[0, l_\infty]$ . For more statistics, see table 1 and figure 5.



**Fig. 5.** Convergence under regular refinement using different metrics and the point source interpolant ( $S_p$ ). Plots show  $l_1/n$  as a function of step size  $h$ . In the Poincaré example (right), the metric tensor is isotropic and [16,12] behave identically.



**Fig. 6.** Visualizing distance map smoothness under a cone metric (with an extreme slope ( $s = 5$ ), compared to figure 5, to emphasize errors). Top row shows color coding of the absolute error (from blue to red) of the distance map. Smoother transition in color indicate smoother maps. Bottom row shows derivative smoothness by color coding of the norm of the gradient of the respective map.

**Anisotropic Partitioning and Sampling.** One interesting application using anisotropic distances is voronoi partitioning and anisotropic sampling [4,3]. One experiment using our method can be seen in figure 7 (left and middle). Applications for this sampling include packing of tensor glyphs, for visualization of diffusion tensor MRI, and adaptive sampling of images.



**Fig. 7.** Left: A picture of a fast moving car. A low-passed structure tensor field defines a metric for the image, which then was partitioned into 300 geodesic Voronoi cells using manifold Lloyd relaxation[3]. Middle: Each cell is colored with its respective mean color. Anisotropic cells align with lines and features in the image. Right top: Delineation of cells using isropic dilation fails. Right bottom: Delineation using anisotropic distances from the structure tensor field for dilation, successfully delineates the cell.

**Distance to Points and Objects.** In many image analysis applications, an important procedure is to compute distances from an object or point. These methods are called distance transforms and they compute all distances from a set of pixels in an image to all other pixels in the image [21,13]. Figure 7 (right) shows one example of an application that benefits from our method,



namely anisotropic curve closing for cell nuclei segmentation [22]. In short, our method enables accurate anisotropic versions of morphological operations such as dilation, erosion, skeletonization and watershed segmentation.

## 6 Conclusion

We have presented a novel metric dependent solver that converges to geodesic distances under grid-refinement. It is simple to implement, less than 150 lines of C++ code, and performs in a predictable manner. It appears to have better convergence properties than two current state-of-the-art algorithms.

Our main motivation for publishing these results is to present a solver that we have found to be easy to implement and adapt. It is a solver that is different from many current approaches, in particular FMM.

## A Test Metrics and Analytic Distances

**A spherical shell:**  $K = 1, \{x, y : x^2 + y^2 \leq 0.98^2\}$ ,

$$g_{ij}(x, y) = \frac{1}{1-x^2-y^2} \begin{bmatrix} 1-y^2 & xy \\ yx & 1-x^2 \end{bmatrix}, \text{ and}$$

$$d_S(\mathbf{a}, \mathbf{b}) = \cos^{-1}(\mathbf{a}^T \mathbf{b} + \sqrt{(1-\mathbf{a}^T \mathbf{a})(1-\mathbf{b}^T \mathbf{b})}).$$

**A flat square:**  $K = 0, (x, y) \in [-1, 1] \times [-1, 1], g_{ij} = \delta_{ij}, d(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_2$ .

**The Poincaré disk model:**  $K = -1, \{x, y : x^2 + y^2 \leq 0.98^2\}$ ,

$$g_{ij}(x, y) = \frac{\delta_{ij}}{(1-x^2-y^2)^2}, \quad d_P(\mathbf{a}, \mathbf{b}) = \frac{1}{2} \cosh^{-1} \left( 1 + \frac{2\|\mathbf{a}-\mathbf{b}\|_2^2}{(1-\|\mathbf{a}\|_2^2)(1-\|\mathbf{b}\|_2^2)} \right).$$

**The cone metric:**  $K = 0$ , a cone  $z = s\sqrt{x^2 + y^2}$  with slope  $s, \{(x, y) \neq (0, 0)\}$ .

$$g_{ij}(x, y) = \delta_{ij} + \frac{s^2}{x^2+y^2} \begin{bmatrix} x^2 & xy \\ yx & y^2 \end{bmatrix}$$

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{[\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 - 2\|\mathbf{a}\|_2\|\mathbf{b}\|_2 \cos(\frac{\angle(\mathbf{a}, \mathbf{b})}{\sqrt{1+s^2}})](1+s^2)}$$

## References

1. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Weighted distance maps computation on parametric three-dimensional manifolds. *Journal of Computational Physics* 225, 771–784 (2007)
2. Bruss, A.R.: The eikonal equation: Some results applicable to computer vision. In: Horn, B.K.P., Brooks, M.J. (eds.) *Shape from Shading*, pp. 69–87. MIT Press, Cambridge (1989)
3. Feng, L., Hotz, I., Hamann, B., Joy, K.: Anisotropic noise samples. *IEEE Transactions on Visualization and Computer Graphics* 14, 342–354 (2008)
4. Du, Q., Wang, D.: Anisotropic centroidal voronoi tessellations and their applications. *SIAM J. Sci. Comput.* 26, 737–761 (2005)

5. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 79, 12–49 (1988)
6. Verbeek, P.W., Verwer, B.J.: Shading from shape, the eikonal equation solved by grey-weighted distance transform. *Pattern Recogn. Lett.* 11, 681–690 (1990)
7. Rouy, E., Tourin, A.: A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis* 29, 867–884 (1992)
8. Rosin, P.L., West, G.A.W.: Saliency distance transforms. *Graph. Models Image Process.* 57, 483–521 (1995)
9. Parazzoli, C.G., Koltzenbah, B.E.C., Greeger, R.B., Lam, T.A., Tanielian, M.H.: Eikonal equation for a general anisotropic or chiral medium: application to a negative-graded index-of-refraction lens with an anisotropic material. *J. Opt. Soc. Am. B* 23, 439–450 (2006)
10. Bardi, M., Capuzzo-Dolcetta, I.: *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhauser (1997)
11. Tsai, Y.H.R., Cheng, L.T., Osher, S., Zhao, H.K.: Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM Journal on Numerical Analysis* 41, 673–694 (2004)
12. Jeong, W.-K., Fletcher, P.T., Tao, R., Whitaker, R.T.: Interactive visualization of volumetric white matter connectivity in dt-mri using a parallel-hardware hamilton-jacobi solver. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE Visualization 2007)*, 1480–1487 (2007)
13. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.* 93, 1591–1595 (1996)
14. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* 40, 1528–1538 (1995)
15. Konukoglu, E., Sermesant, M., Clatz, O., Peyrat, J.-M., Delingette, H., Ayache, N.: A Recursive Anisotropic Fast Marching Approach to Reaction Diffusion Equation: Application to Tumor Growth Modeling. In: Karssemeijer, N., Lelieveldt, B. (eds.) *IPMI 2007*. LNCS, vol. 4584, pp. 687–699. Springer, Heidelberg (2007)
16. Lenglet, C., Prados, E., Pons, J.P., Deriche, R., Faugeras, O.: Brain connectivity mapping using Riemannian geometry, control theory and PDEs. *SIAM Journal on Imaging Sciences* (2008)
17. Novotni, M., Klein, R.: Computing geodesic distances on triangular meshes. In: *Proceedings of The 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2002* (2002)
18. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. In: *SIGGRAPH 2005: ACM SIGGRAPH 2005 Papers*, pp. 553–560. ACM Press, New York (2005)
19. Reimers, M.: *Topics in Mesh based Modelling*. PhD thesis, Univ. of Oslo (2004)
20. Gonzalez, R., Rofman, E.: On deterministic control problems: An approximation procedure for the optimal cost i. the stationary problem. *SIAM Journal on Control and Optimization* 23, 242–266 (1985)
21. Sonka, M., Hlavac, V., Boyle, R.: *Image Processing: Analysis and Machine Vision*. Thomson-Engineering (1998)
22. Malm, P., Brun, A.: Closing curves with Riemannian dilation: Application to segmentation in automated cervical cancer screening. In: *Proc. of 5th International Symposium on Visual Computing, Las Vegas, Nevada, USA* (2009)