

# Finding the Minimum-Cost Path Without Cutting Corners

R. Joop van Heekeren, Frank G.A. Faas, and Lucas J. van Vliet

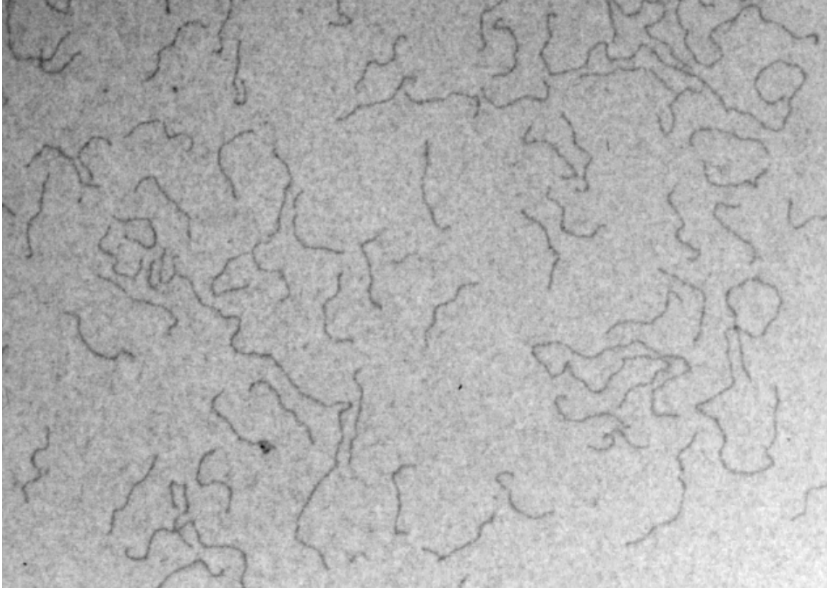
Quantitative Imaging Group, Delft University of Technology, The Netherlands  
L.J.vanVliet@tudelft.nl

**Abstract.** Applying a minimum-cost path algorithm to find the path through the bottom of a curvilinear valley yields a biased path through the inside of a corner. DNA molecules, blood vessels, and neurite tracks are examples of string-like (network) structures, whose minimum-cost path is cutting through corners and is less flexible than the underlying centerline. Hence, the path is too short and its shape too stiff, which hampers quantitative analysis. We developed a method which solves this problem and results in a path whose distance to the true centerline is more than an order of magnitude smaller in areas of high curvature. We first compute an initial path. The principle behind our iterative algorithm is to deform the image space, using the current path in such a way that curved string-like objects are straightened before calculating a new path. A damping term in the deformation is needed to guarantee convergence of the method.

## 1 Introduction

Algorithms for computing the minimum-cost path have played an important role in various fields of science and engineering. These algorithms try to find the path connecting a selected start and end point that minimizes the integrated costs. In optics, light rays travel along a minimum-cost path from source to destination. A wave front of light propagates with a speed that depends inversely proportional on the refractive index of the medium. A space varying velocity map suggests that the path with the shortest arrival time will in general be longer than the Euclidean distance between the start and end points. If you consider the local cost as the inverse of the local speed, then calculating the minimum integrated cost corresponds to calculating the smallest possible arrival time from a start point to all points in the domain.

In many fields of science and engineering we encounter images of string-like structures in which the centerline conveys important information about the underlying objects. Examples are DNA-strands (cf. Fig. 1), blood vessels, or neurite tracks. The tracking results as depicted in [5] display exactly the problems that we are addressing in this paper. The minimum-cost path does not follow the curvilinear valley of the cost function, but is biased towards the inside of corners. In general, the minimum-cost path is cutting corners, and is therefore shorter and stiffer than the underlying centerline of the cost valley. Quantification of the



**Fig. 1.** DNA molecules labeled with uranyl acetate and visualized by transmission electron microscopy. The images are kindly provided by Dr. Dmitry Cherny, PhD, Dr. Sc.

bending energy of DNA plays a key role in understanding cellular processes. To verify the competing models and measure the so-called persistence length, an accurate path through these structures is required. The key to quantifying the length and shape of these objects is to find the centerline of these objects. A minimum-cost path guarantees a connected path that approximates this centerline even if the curvilinear object contains small gaps and is corrupted by noise (cf. Fig. 1). Solving the bias problem of minimum-cost path algorithms will be of utmost importance in many fields of science and engineering.

A typical implementation of such a standard minimum-cost path approach consists of the following steps:

- Define one start point in the image domain. Having an end point is not mandatory, but may assist in defining an early stopping criterion.
- Compute the minimum integrated cost from the starting point to all points in the domain, or until the pre-defined end point has been reached.
- Descend along the opposite gradient direction of the integrated cost image from the end point to the start point. Due to the smoothness of the integrated cost images one can obtain sub-pixel accuracy in the location of the minimum-cost path.

The minimum integrated cost  $T$  is given as the minimum cumulative cost along all possible paths  $\mathbf{P}$  connecting the start point  $S$  with any end point  $E$  in our domain. Or mathematically:

$$T = \min_{\forall \mathbf{P}_{SE}} \int_S^E I(\mathbf{P}_{SE}(l)) dl \quad (1)$$

with  $\mathbf{P}_{SE}(l) = (x(l), y(l))$ . This is equivalent to solving the Eikonal equation [2]

$$|\nabla u(x, y)| = I(x, y) \quad (2)$$

in which  $I(x, y)$  denotes the local cost function and  $u(x, y)$  the local arrival time. For uniform costs, the solution of the Eikonal equation is identical to the result given by the (domain constrained) Euclidean distance transform. For space variant costs we have the gray-weighted distance transform (GDT) [9,7,6,4] and fast marching (FM) algorithms [1,3,8]. Both methods are based on wave front propagation. The GDT constructs a path using a superposition of cost-weighted basis vectors, thereby quantizing the local path direction to the directions of a set of basis vectors in a 3x3 (or 5x5) neighborhood. The FM algorithm models the wave front by a straight front, which does not restrict the propagation direction to a limited set of discrete directions. Both methods produce an image containing the minimum integrated cost from a starting point to all points in the domain.

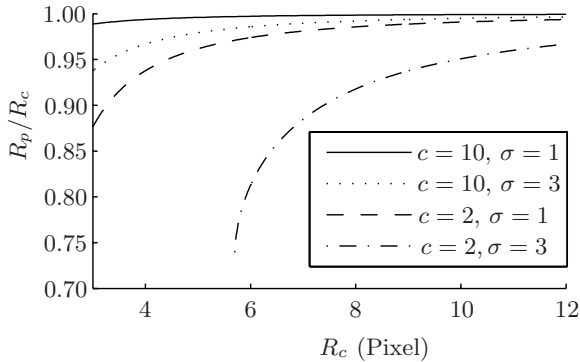
The minimum-cost path can be obtained by a steepest descend (from the end point back to the start point) along the opposite gradient direction of the integrated cost map created by the aforementioned methods. Since the cost function is usually smooth, the integrated cost function is even smoother. This permits sub-pixel accuracy in computing the steps taken during the steepest descend. Due to the finite step size and approximation errors in the aforementioned algorithms, the path will not end exactly at the starting point but in very close (sub-pixel) proximity.

In section 2 we quantify the cutting corner problem for circular arcs with a Gaussian line profile and present our iterative algorithm to solve it. In section 3 we present quantitative results on the displacement error as a function of the number of iterations and qualitative results on TEM images of uranyl acetate labeled DNA. Section 4 presents the conclusions of our work.

## 2 Method

A correct implementation of a minimum-cost path algorithm applied to curved linear structures will always result in a path that is shorter and stiffer (less bending energy) than the centerline of the underlying linear structure. Especially in highly curved areas the minimum-cost path is cutting corners. The minimum-cost path does not follow the path through the minimum of the cost function in curved areas. To illustrate this we consider a circular path with a Gaussian cross-section

$$I(r, \sigma, c) = 1 + c \left( 1 - \exp\left(-\frac{(r - R_c)^2}{2\sigma^2}\right) \right) \quad (3)$$



**Fig. 2.** The relative radius ( $R_p/R_c$ ) of the minimum-cost path for the Gaussian profile as a function of centerline radius  $R_c$

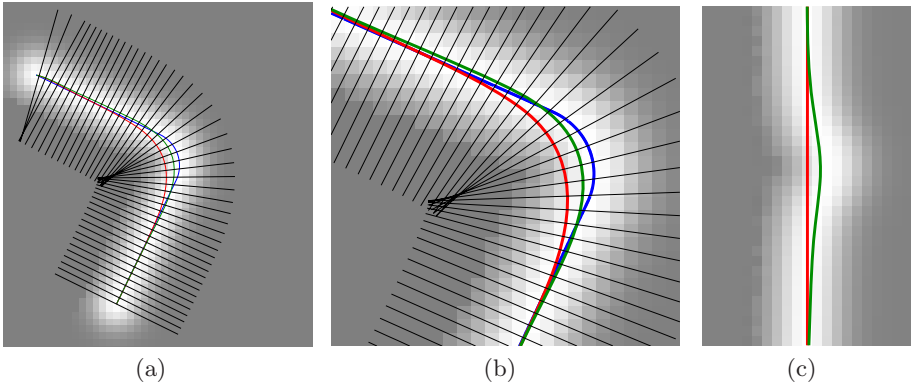
with the cost contrast  $c = \frac{c_b - c_p}{c_p}$ , in which  $c_b$  and  $c_p$  are respectively the cost values of the background and the path, and  $R_c$  the radius of curvature of the centerline. The integrated cost of a circular path with radius  $r$  around such a circle is

$$T(r) = 2\pi r I(r, \sigma, c) = 2\pi r \left( 1 + c \left( 1 - \exp\left(\frac{-(r - R_c)^2}{2\sigma^2}\right) \right) \right) \quad (4)$$

To find the minimum-cost path, we calculated the radius  $R_p$  for which  $T(r)$  is minimized,  $R_p = \operatorname{argmin} T(r)$ . Fig. 2 shows the relative radius  $R_p/R_c$  of the minimum-cost paths for different values of line width  $\sigma$  and contrast  $c$ . The results suggest that increasing the contrast or decreasing the line width (for example by scaling the cost function:  $I(\mathbf{r}) \rightarrow I^\alpha(\mathbf{r}), \alpha > 1$ ) of the cost function will reduce the bias in the minimum-cost path. In practice the bias will be reduced by these measures to some extent, but will never produce the desired smooth centerline path. This is shown by considering the limit ( $c \rightarrow \infty$  or  $\alpha \rightarrow \infty$ ), this will reduce the problem to a binary problem discarding all the gray value information and therefore produce a rough, binary skeleton type path instead of smooth centerline path. This skeleton path will also be hampered and possibly even interrupted due to the presence of noise in the original image. We claim to have developed an algorithm not based on increasing the contrast or decreasing the line width which solves the bias problem and still finds a smooth path, approximating the true centerline, through this class of objects.

## Algorithm

Our method is based on the idea that a standard minimum-cost path algorithm such as FM will only give the correct centerline path for straight string-like objects (assuming the start and end points are located on the centerline). Hence, the principle behind our algorithm is to deform the image space in such a way

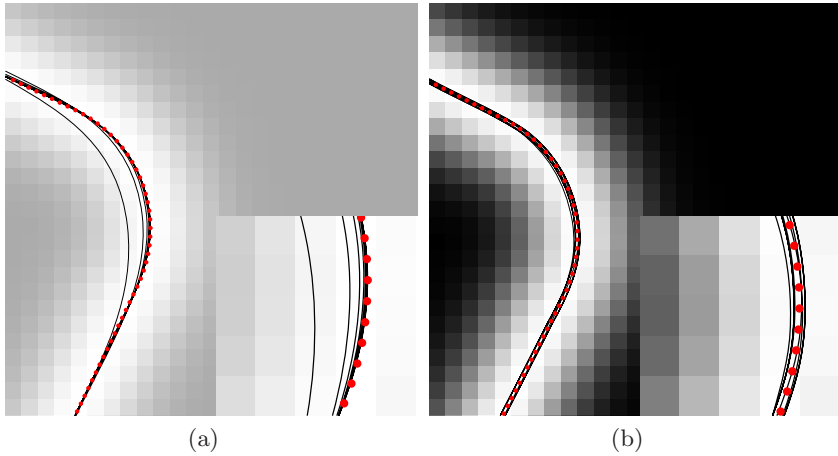


**Fig. 3.** (a) The red line is the path extracted by a classical minimum-cost path algorithm, the green line is the path after the first iteration and the blue line is the true centerline of the object. (b) Zoomed in version of (a). (c) Deformed image obtained by equi-distant sampling perpendicular to the initial (red) path. The green path is the minimum cost path formed in the deformed image.

that a curvilinear object becomes straight. After an initial path through the object is extracted using a standard minimum-cost path algorithm, two cubic splines are defined through the data points found by a steepest descend; one for the  $x$ -values and one for  $y$ -values of the data points, using the distance from the end points along the path as the running variable. Using cubic splines guarantees that the path is continuous up to the second derivative. As shown in Fig. 3(a-b), lines perpendicular to the splines separated by a distance of one pixel are defined. A new image Fig. 3(c) is sampled using cubic interpolation on equi-distant points along these perpendicular lines. A new minimum-cost path is calculated in the deformed space (the green line in Fig. 3(c)). This new path is again represented by two splines. Next, the perpendicular distance between the centerline of the deformed image and the splines is calculated. By defining points on the perpendicular lines in the original image with the same distance from the original path, the new path is transferred back to the original image space. Two new splines are fitted through the coordinates of these points to produce a new path. As shown in Fig. 3(a), this path is already much closer to the desired centerline. Repeating the process described above yields a path that converges to the true centerline of the object.

### 3 Results

We first tested our algorithm on synthetic data, allowing us to measure its performance by comparing the results with a ground truth. Later we used images of DNA-strands made using an electron microscope to examine its real world performance qualitatively.

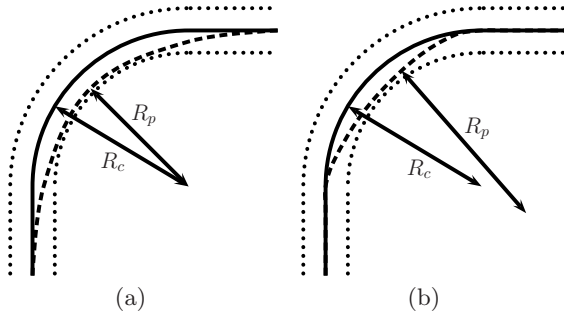


**Fig. 4.** (a) A low contrast ( $c = 1$ ) image depicting the path converging to the true centerline of the object. (b) A High contrast ( $c = 10$ ) image depicting the oscillation effect causing the path to lie alternately on either side of the true centerline of the object. In both images the true centerline is denoted by the red dotted line.

### 3.1 Synthetic Data

As synthetic data we used images of curved Gaussian line profiles, with a ninety degrees change in orientation and a curvature radius  $R_c$  (Fig. 3(a)). The cross section of this profile is defined as in Eq. 3. The algorithm was tested for different centerline radii  $R_c$ , noise levels and contrast ratios  $c$ . As shown in Fig. 4(a) the first iteration already results in a path which is significantly closer to the centerline. We measured the performance by looking at the distance between the centerline of the object and the path found using our algorithm. We computed the root-mean-square (RMS) of the perpendicular distance between the path and the ground truth at ten points separated by a pixel in the middle of the curve.

Initially this RMS error decreases for all the settings. However, after a number of iterations (one to three for the high contrast images and about six for the low contrast images) it starts to increase for certain values of radius  $R_c$  and contrast level  $c$ . This is due to an oscillation effect, which results in the paths lying alternately on either side of the centerline of the object between successive iterations. The effect is depicted in the close up of Fig. 4(b). We suspect it originates from the two fundamentally different ways to cut a corner. In Fig. 5 the two possible ways are shown. On the left side the radius of the path  $R_p$  which cuts the corner is larger than the radius of the centerline  $R_c$  of the object. In contrast to the situation on the right where the radius of the path is smaller. Due to this sharper bend, we overcompensate for the bending and hence change sign of the curvature in the deformed space. In cases with a bend which is less sharp than the true centerline, we only partially compensate and hence do not change the sign of the curvature. Therefore, the oscillating effect is only observed when



**Fig. 5.** (a) The first way to cut a corner. Notice the radius of the path ( $R_p$ ) being larger than the radius of the centerline ( $R_c$ ). (b) The second way to cut a corner. Notice the radius of the path ( $R_p$ ) being smaller than the radius of the centerline ( $R_c$ ).

an intermediate path has a sharper bend than the true centerline of the object and significantly cuts the corner.

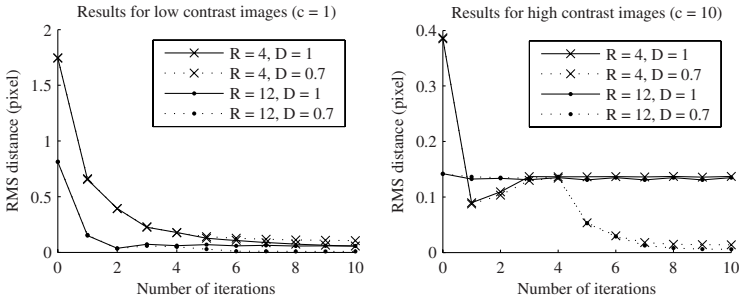
To counteract this effect we introduced a damping term at the transition of the path from the deformed space to the original image space. The damping is being reduced exponentially. After  $N$  iterations the distance between the last and the new path in the  $n^{\text{th}}$  iteration is multiplied by a damping factor  $D^{(n-N)}$  ( $D < 1$ ). This damping assures stable results. Elaborate testing has showed us that  $D = 0.7$  is either the optimal or near optimal over a wide range of values for  $c$  and  $R_c$ . Only very low contrast settings require less damping to allow the path to reach the centerline.

In Fig. 6 the mean of the RMS distance of twenty realizations is plotted as a function of the number of iterations with low ( $c = 1$ ) and high ( $c = 10$ ) contrast settings and no noise added. The plots show that the RMS error decreases dramatically in comparison with standard minimum-cost path algorithms (the  $0^{\text{th}}$  iteration) for all radii and contrast levels. The damping is switched on after the fourth iteration. We observed that the damping decreased the RMS distance on all high contrast images, but on the low contrast images only for the curves with a large radius.

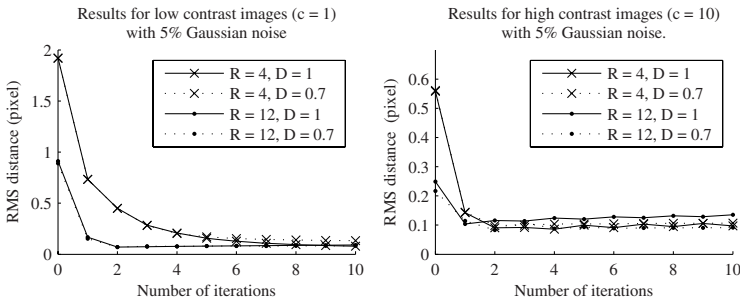
Fig. 7 shows the mean RMS distance for images with 5 percent Gaussian noise added. The RMS distance slightly increases after a number of iterations. This is not due to oscillating behavior but caused by the fact that the path also adapt to the noise pattern. For medium to high SNR's the path corrections in the first iterations are dominated by the signal. The iterative procedure should stop when the noise becomes the dominant factor.

### 3.2 Real Data

The proposed algorithm has been extensively tested on transmission electron microscope images of DNA-strands labeled with uranyl acetate to quantify their shape. Empirically we deduced that twenty-five iterations were sufficient to reach a stable result on all of the images. Because no oscillating behavior was observed,



**Fig. 6.** The mean of the RMS distance for twenty realizations as a function of the number of iterations using low ( $c = 1$ ) and high ( $c = 10$ ) contrast and line width  $\sigma = 2.5$ . For the cases with  $D = 0.7$  the damping is switched on after four iterations.



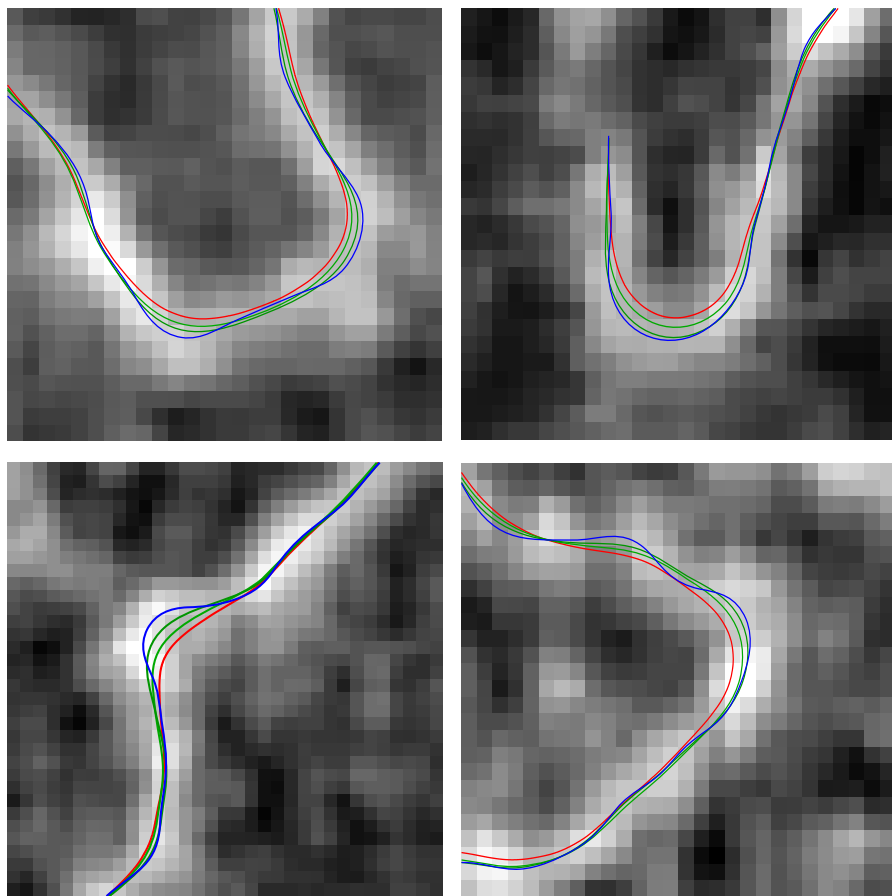
**Fig. 7.** The mean of the RMS distance for twenty realizations as a function of the number of iterations using low ( $c = 1$ ) and high ( $c = 10$ ) contrast and line width  $\sigma = 2.5$ . 5% Gaussian noise is added to the images. For the cases with  $D = 0.7$  the damping is switched on after four iterations.

no exponential damping was used. Fig. 8 shows four typical results from the more than thousand molecules that were processed. The red line is depicting the path found by the fast marching algorithm, the blue indicates the final result after twenty-five iterations, the green lines in between are the results after respectively one and four iterations. Note that the final results describe the centerline of the object much better, especially in regions with high curvature. The blue line follows the local minimum of the cost function without cutting corners. This work permits the computation of the persistence length of DNA with much greater accuracy, especially over small distances. Earlier results always overestimated the persistence length in this regime due to the stiffness of the minimum-cost path.

### 3.3 Computational Speed

The time needed for one iteration is comparable to the time needed to calculate a classical minimal-cost path. Therefore, it is evident that the amount of





**Fig. 8.** Four TEM images of DNA-strands. The red line depicts the path found by a standard minimum-cost path algorithm (FM). The green lines indicate the results of our method after one and four iterations, the blue line after 25 iterations.

computation needed increases approximately linearly with the number of iterations. Note that one often can limit the amount of image space to be evaluated after the first iteration, hence reducing the computation time in subsequent iterations.

## 4 Conclusion

In this paper we present an improvement on minimum-cost path algorithms, which significantly boosts their performance in describing the centerline of string-like objects. The method can be incorporated in any minimum-cost path algorithm. We have demonstrated that our algorithm results in a path which corresponds much better to the centerline of both simulated and real-world

string-like objects. The RMS displacement error decreases more than a factor of ten, especially in highly curved areas. Displacement errors of several pixels can be repaired. The behavior depends on conditions such as contrast, noise level and line width. Under certain conditions, such as high contrast, the method only converges after incorporating a damping term. Ten to twenty-five iterations are needed, using an exponentially reducing damping term after several iterations. The method has been successfully applied to several thousands of DNA molecules in high-resolution images obtained by TEM and AFM. The paths we obtained on the images of DNA-strands follow the valley through the cost function without cutting corners. Hence the length measurement remains unbiased and the curvature is no longer underestimated. This is of utmost importance for measuring the bending energy of DNA-strands on a nanometer scale.

## References

1. Adalsteinsson, D., Sethian, J.A.: A fast level set method for propagating interfaces. *Journal of Computational Physics* 118(2), 269–277 (1995)
2. Born, M., Wolf, E.: *Principles of Optics*, 6th edn. 1977 Pergamon Press, London (1980)
3. Danielsson, P.-E., Lin, Q.: A modified fast marching method. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 1154–1161. Springer, Heidelberg (2003)
4. Fouard, C., Gedda, M.: An objective comparison between gray weighted distance transforms and distance transforms on curved spaces. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 259–270. Springer, Heidelberg (2006)
5. Meijering, E., Jacob, M., Sarria, J.-C.F., Steiner, P., Hirling, H., Unser, M.: Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry* 58A(2), 167–176 (2004)
6. Saha, P.K., Wehrli, F.W., Gomberg, B.R.: Fuzzy distance transform: Theory, algorithms, and applications. *Computer Vision and Image Understanding* 86(3), 171–190 (2002)
7. Toivanen, P.J.: New geodesic distance transforms for gray-scale images. *Pattern Recognition Letters* 17(5), 437–450 (1996)
8. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* 40(9), 1528–1538 (1995)
9. Verbeek, P.W., Verwer, B.J.H.: Shading from shape, the eikonal equation solved by grey-weighted distance transform. *Pattern Recognition Letters* 11, 681–690 (1990)