Sub–pixel Distance Maps and Weighted Distance Transforms

Ron Kimmel
Electrical Engineering Department

Nahum Kiryati
Electrical Engineering Department

Alfred M. Bruckstein
Computer Science Department
Technion, Haifa 32000, Israel

ABSTRACT

A new framework for computing the Euclidean distance and weighted distance from the boundary of a given digitized shape is presented. The distance is calculated with sub–pixel accuracy. The algorithm is based on an equal distance contour evolution process. The moving contour is embedded as a level set in a time varying function of higher dimension. This representation of the evolving contour makes possible the use of an accurate and stable numerical scheme, due to Osher and Sethian [22]. The relation between the classical shape from shading problem and the weighted distance transform is presented, as well as an algorithm that calculates the *geodesic* distance transform on surfaces.

Keywords: distance transforms, weighted distance, curve evolution, differential geometry, continuous scale morphology.

1 Introduction

Distance maps on a pixel array picture are usually defined as discrete functions reflecting the minimal Euclidean distance of each pixel from the boundary pixels, see [6, 17, 33]. Algorithms that calculate such maps usually operate on segmented objects, *i.e.,* objects that have been transformed into a binary picture. In the process of mapping the shape into "White" interior and "Black" exterior pixels, information that may be crucial, if we desire to establish an accurate distance map, is usually lost.

A distance map may be found with the help of a propagating equal distance contour, or wave front, which starts from the boundary and propagates inwards (into the shape) with a unit velocity. The moving wave front assigns the distance to each pixel as it passes through it. See for example [2, 5], where the equal distance contour is referred to as the front of the flame in the *prairie–fire* model.

In [22] Sethian and Osher have shown that by considering the evolving curve as a level set of a higher dimensional function that propagates according to a well–defined rule, topological changes and numerical problems are inherently solved. We propose to construct an accurate and efficient distance map algorithm based on this idea.

The problem of finding the distance becomes more interesting when considering a different cost or weight to each point in the domain. This leads to the idea of weighted distance transforms. We shall show that the weighted distance transform may be used to solve the classical shape from shading problem where one desires to reconstruct a three dimensional object from its gray level image. Considering other metrics than the Euclidean metric, introduces the continuous scale morphology, [3]. In [27] it was shown that morphological operations at any given scale may be performed via curve evolution, the 'structuring element' of the morphological operation being determined by the metric.

Taking off from planes to surfaces, we search for distance transform of a given area on a given surface, [12]. Calculating this *geodesic* distance map is performed by an equal distance curve propagating on the given three dimensional surface.

The solutions to all the above problems may be achieved by referring to the Osher–Sethian algorithm for propagating the equal distance contour, or by solving a Hamilton Jacobi type of equation that minimizes the difference between a given function gradient and the gradient of the desired solution. Two good examples of the last approach are the viscosity solutions to the shape from shading problem [26, 18], and [31] where such a formulation is used to find the distance map as part of a numerical algorithm that simulates incompressible two–phase flow, see also [14]. We shall study the relation between the two approaches.

This paper presents a new framework for calculating all the above defined distance maps. In Section 2, the propagation rule of a planar curve evolving with constant velocity along its normal is introduced, as well as the level set formulation for that simple propagation rule. Section 3 introduces a cost function over the planar domain and studies the relation of the weighted distance transform and the shape from shading problem. In Section 4, the continuous scale morphology via level sets is used to achieve the distance maps for any given metric. In Section 5, propagation of equal distance contours on surfaces is used to calculate the *geodesic* distance map. Two possible applications are given in Section 6, followed by some examples, and concluding remarks in Section 7.

In the next section we start by considering the evolution equation of a planar curve propagating with constant velocity along its normal direction. This simple planar case is reformulated to the evolution of an implicit representation of the curve. The new implicit formulation (the heart of the Osher–Sethian algorithm, called the *Eulerian* formulation) enables us to implement an efficient numerical scheme that approximates the evolution equation on a pixel grid. By tracking the propagating curve on the grid, the Euclidean distance transform is achieved.

2 Equal distance contour – propagation rule

Consider the propagating planar curve $C(p,t) : [0,S] \times [0,T] \to \mathbb{R}^2$, evolving according to the propagation rule

$$C_t(p,t) = \vec{n}(p),$$

where p parameterizes the curve, t is the time (which represents the distance in the simple constant velocity case,) and $\vec{n}(p)$ is the planar unit normal to $C(p,t)$ at p.

The evolution process generates the propagating family of simple and closed planar curves $X(t)$ which represent the traces of $C(p,t)$,

$$X(t) = \{(x,y) \in C(p,t) | (x,y) \in \mathbb{R}^2\}.$$

Embedding these planar curves as zero level sets of a time varying three dimensional function $\phi(x,y,t) : \mathbb{R}^2 \times [0,T) \to \mathbb{R}$, means that we require the following [22]:

a. $\phi(x,y,0) = 0$ is an implicit representation of $X(0)$.

b. $\phi(x,y,0)$ is smooth, positive in the interior and negative in the exterior of $X(0)$.

c. $\phi(x,y,t) = 0$ provides $X(t)$.

Applying the chain rule on the last requirement yields

$$\phi_t + \nabla\phi \cdot X_t = 0.$$

The planar normal to each level set is given by $\vec{n} = \frac{\nabla\phi}{\|\nabla\phi\|}$. Using the constant (unit) propagation velocity constraint $C_t(p,t) \cdot \vec{n} = 1$, we can write

$$X_t \cdot \frac{\nabla\phi}{\|\nabla\phi\|} = 1.$$

Therefore, an evolution rule given by

$$\phi_t = -\|\nabla\phi\|,$$

meets the desired requirements.

Without any prior segmentation or edge detection procedures, let $\phi(x,y,0) \equiv E(x,y) - T$, where $E(x,y)$ is the gray level picture, and T is the threshold separating the object from the background, see [15] (T may also be adaptive, see *e.g.* [34]). We establish the following distance transform algorithm

1 Initialize $\phi(x,y,0) = E(x,y) - T$, see also [15].

2 Propagate ϕ according to $\phi_t = -\|\nabla\phi\|$, using a stable and accurate numerical algorithm (as provided by Osher and Sethian in [22]).

3 Using a rectangular grid approximation (the pixel–grid), at each grid point: Check whether the zero level set has passed through the grid point. Assign the proper distance to the point if it does, using a simple linear interpolation.

This is a conceptually simple algorithm that calculates the distance map with sub–pixel accuracy. It can also solve the so-called *shape offsetting* problem in CAD, see [15]. The accuracy is governed by the pixel grid resolution and the distance (time) step of the iterative numerical scheme. When not considering any possible redundancy, the calculation effort is of order $O(\frac{D}{\Delta t}m \cdot n)$, where D is the maximal distance from the boundary and $m \cdot n$ in the number of grid points used in the numerical scheme. A parallel implementation reduces the calculation complexity to $O(\frac{D}{\Delta t})$.

3 Weighted distance transforms

In this section we argue that solutions of the classical *Shape from Shading* problem in which the viewer and the light source have the same location and the surface reflects the light with a "Lambertian" shading rule, see *e.g.* [4], also solve a problem of distance transforms with a given traversability map. We first introduce the weighted distance transform problem, and then present the approaches to its solution.

3.1 The path of minimal cost

Consider a problem where we search for a planar path from a source to a destination, so that the accumulation of a penalty function along that path is minimal. This minimization problem may be defined as follows. Let $s, d \in \mathbb{R}^2$ be the given source and destination points, and $f(x,y) : \mathbb{R}^2 \to \mathbb{R}^+$ be a given cost function. Then, find the planar curve l^o connecting s to d so that

$$|l^o| \equiv \int_s^d f(x,y)dl^o = \min_{l \in L}\{\int_s^d f(l(p))dp, \ \left|\frac{dl}{dp}\right| = 1\}, \tag{1}$$

where L is the set of all planar curves $l \in \mathbb{R}^2$ connecting s to d, with arclength parameterization p.

The relation between the weighted distance transform and the shading from shape or contours from shading was dealt with in [32, 25]. The problem of finding the weighted distance may be shown to be equivalent to the well known *Shape from Shading* problem in Computer Vision, in the simple case where the light source and viewer have the same direction. In the recent years various solutions to this problem were proposed [1, 7, 8, 9, 10, 23, 24, 26]. In [4, 11, 13] a solution to the shape from shading problem via level sets is described. Motivated by this work, we propose to similarly solve the weighted distance transform problem and also to find the path of minimal cost from s to d.

As a first step propagate an equal cost contour (equal height contours according to [4]) that obeys the planar curve evolution rule

$$\frac{\partial C}{\partial t} = \frac{1}{f(x,y)}\vec{n}.$$

Here $\vec{n}$ is the unit normal to the curve, and the initial curve is given as a very small contour around the source point $C(0) = C_0$. Formulating this evolution in the so called Eulerian representation, see [30], yields the level set evolution version

$$\begin{cases} \phi_t = \frac{1}{f(x,y)}\|\nabla\phi\| \\ \\ \phi(0) = \phi_0, \end{cases}$$

see [13] for the numerical implementation. In the case of shape from shading, $E(x,y)$ – the gray level (shading) image – equals (under reasonable assumptions) to $\cos(\alpha)$, where α is the angle between the surface normal and the light source direction. When the viewing and the light source direction are the same, see Figure 1, the evolution of an equal height contour is given by

$$\frac{\partial C}{\partial t} = \cot(\alpha)\vec{n},$$

or

$$\frac{\partial C}{\partial t} = \frac{E}{\sqrt{1-E^2}}\vec{n},$$

see [4, 13] for further details. Observe that defining the cost function as function of $E(x,y)$ to be

$$f(x,y) \;=\; \tan(\alpha) \;=\; \frac{\sqrt{1-E^2(x,y)}}{E(x,y)},$$

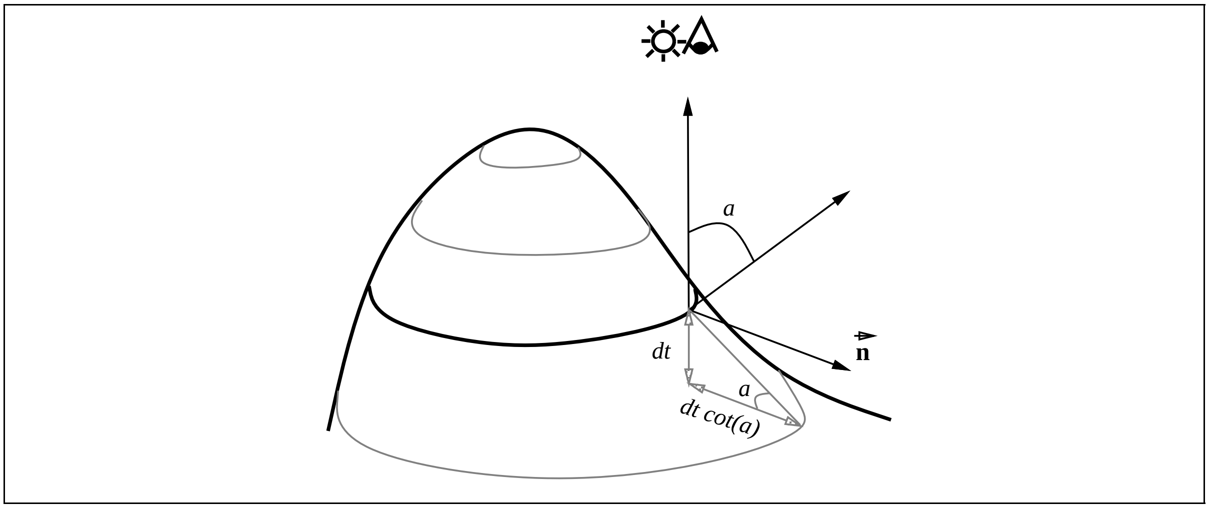

Figure 1: Descending dt in height, the equal height contour should evolve by $dt \cdot \cot(\alpha)$ along its normal direction

the shape from shading problem coincides with the the problem of finding the weighted distance transform.

As it propagates, the equal cost contour assigns the proper time to each grid point it touches. The time assigned to every grid point therefore corresponds to the minimal cost it takes to reach that point (minimal height in the shape from shading case). Define $u_s(x, y)$ and $u_d(x, y)$ to be the weighted distance maps from the source and destination points, respectively. We can proceed and search for the minimal level of the function $u_s(x, y) + u_d(x, y)$, where $u_s(x, y)$ is the accumulated cost at point (x, y) when starting from s. This may be formally represented by

$$l^o = \{(x, y) | u_s(x, y) + u_d(x, y) = \inf_{(x,y)} \{u_s(x, y) + u_d(x, y)\}\}.$$

This way of obtaining the paths of minimal cost may serve as a simple and direct approach for solving many energy or cost minimization problems.

Considering the problem in which the cost function is a traversability map, and the initial source is an area rather than a point, we are led to the "weighted distance transform" problem. In this problem a cost is assigned to each point in the continuous domain (the penalty function), and traveling from the source area to any point in the domain should accumulate the least possible cost.

We have shown in [13] a solution to the shape from shading problem, which also applies in this case. Another approach to this problem was recently presented by Rouy and Tourin in [26]. They propose to consider the "Hamiltonian" of this minimization problem

$$H(\nabla u(x, y)) = |\nabla u(x, y)| - f(x, y),$$

and present a viscosity solution to it which is implemented either via a numerical scheme based on Osher and Rudin [20], or via a simple, but longer procedure that evolves the discretization of the following PDE:

$$u_t = H(\nabla u(x, y)).$$

In the shape from shading problem $H(\nabla u(x, y)) = |\nabla u(x, y)| - \tan(\alpha)$, where u is the reconstructed surface and $\tan(\alpha)$ is the cost function dependent on the gray levels as before. This formulation explains the relation between the two approaches. In equal height contour evolution, the local propagation velocity of the curve is proportional to $\cot(\alpha)$, *i.e.,* when the gradient of the reconstructed shape is high, 'jumping' from one equal height contour to the next involves small steps. In the minimization problem, we try to modify u, so that the magnitude of its gradient will agree with $\tan(\alpha)$, as obtained from the shading image. Observe that the ϕ function introduced in the Eulerian approach is an arbitrary initialized auxiliary function used for an implicit representation of the propagating curve C, while u is the reconstructed surface itself and may also be referred to as the "weighted distance function", "weighted distance map" or "weighted distance transform".

Note that this procedure might also be used to solve the simple distance transform, where $H(\nabla u(x, y)) = |\nabla u(x, y)| - 1$. It is easy to see that the above PDE-s' steady–state is for $|\nabla u(x, y)| = 1$ ($u_t = 0$), which agrees with the distance function whose gradient is 1 almost everywhere. In [31] the distance is computed inward and outward simultaneously by modifying the above PDE to

$$u_t = S(u_0)(1 - |\nabla u|),$$

where $S(\cdot)$ is a smoothed version of a sign function, *e.g.*

$$S(u_0) = \frac{u_0}{\sqrt{u_0^2 + \epsilon^2}}.$$

This way the zero level set, that specifies the shape boundary in our case, is kept fixed while the rest converges to the proper distance from the boundary, negative distance in the interior and positive in the exterior of the shape.

4 Other metrics: Continuous scale morphology

Let us consider other metrics than the simple Euclidean metric. Let $r(\theta)$ be the set of points of unit distance from the origin $(0, 0)$, that is determined by a given metric. In Figure 2.2, an L_1 metric determines the set of points of a 'unit' distance from the origin as a diamond shape. The distance function imposed by a given metric may be determined by evolving the given shape boundary with a *structuring element* that is determined by the set $r(\theta)$ that characterizes the given metric.

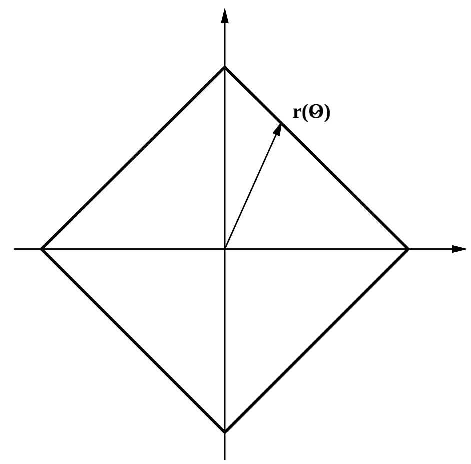

Figure 2: A structuring element is defined by the 'unit sphere' of the given metric. In this example the diamond is the unit sphere of the the L_1 *(city block)* metric.

It may be shown, see [27], that the above operation is the result of *dilating* the given shape by $r(\theta)$. In order to find the set of points of distance T from the boundary of the original shape, we can dilate the shape with a structuring element $r(\theta)T$. The same morphological operation may be achieved as a Minkovsky addition by successively applying an infinitesimal structuring element $\epsilon r(\theta)$, the number of successive dilation operations being given by T/ϵ. In the continuous case the structuring element must be taken to have an infinitesimal value. Observe a given point on the boundary, applying the infinitesimal dilation operation is the same as splitting the point into many points and moving or "propagating" each point to the boundary determined by the structuring element. The same procedure is carried out for each point along the curve. Our search is after the envelope (hull) of all the evolved points. This is a version of the *Huygens principle* form optics [29]. A known result from the theory of curve evolution states that the image (also known as trace, or geometric shape) of the evolving curve may be determined by only considering the normal component of the velocity at each point [28]. Using this fact we may determine only the maximal projection of the splitting point on the normal, and thereby get the evolution rule that agrees with the continuous scale morphology.

Formally, given the boundary of a shape $C_0(s)$, the result of dilating the shape by $r(\theta)T$ is given by propagating the boundary according to:

$$C_t = \sup_{\theta \in [0,2\pi]} \{r(\theta) \cdot \vec{n}\}\vec{n},$$

starting with $C(0) = C_0$ and stopping the propagation at $t = T$, getting $C(T)$ as the desired result. We have shown in [27] (see also [3]), that the above curve propagation rule in the level set formulation is given by

$$\phi_t = \sup_{\theta \in [0,2\pi]} \{r(\theta) \cdot \nabla \phi\}.$$

In this framework we are after the distance imposed by the given metric. Using the above curve evolution we may assign each point a distance equal to t, which is the time the zero level set passed through that point. It was shown that the metric defines the structuring element that is used to dilate the boundary. The dilation operation may be performed in a continuous scale using the above curve evolution rule. The curve evolution is reformulated in the level set (Eulerian) formulation. And the distance is obtained by assigning the time of arrival of the zero level set to each point.

5 Surfaces: Geodesic distance map

Up to this point planar curves were propagated to calculate distances from given planar shapes. The above techniques still apply when considering more than two dimensions, that is, by propagating zero set $\mathbb{R}^n$ manifolds of functions in $\mathbb{R}^{n+1}$. A different kind of problem is calculating distance maps on surfaces in $\mathbb{R}^3$, (or manifolds in $\mathbb{R}^n$, in the general case).

Consider the simple case of a given boundary of a shape defined on a given surface $\mathcal{S} \in \mathbb{R}^3$. The boundary of the shape may be presented as a three dimensional curve $\alpha : [a,b] \in \mathbb{R} \to \mathbb{R}^3$, where $\alpha \in \mathcal{S}$. It was shown in [12] that the evolution rule of an equal distance contour that propagates on a surface is given by

$$\alpha_t = \vec{t} \times \vec{\mathcal{N}},$$

where $\vec{t}$ is the tangent vector of the curve $\vec{t} = \alpha'/|\alpha'|$, and $\vec{\mathcal{N}}$ is the normal of the surface $\mathcal{S}$. Propagating a $3D$ curve is hard to implement. We, therefore, project the evolution to the plane, and consider only the tangential component to the planar curve. Define the projection operation as π, *i.e* $\pi \circ (x,y,z) \equiv (x,y)$. The evolution rule of the planar curve, C, that is the projection of the evolution of the surface curve α, is given by

$$C_t = < \pi \circ (\vec{t} \times \vec{\mathcal{N}}), \vec{n} > \vec{n},$$

where $C = \pi \circ \alpha$, and $\vec{n}$ is the normal of the planar curve C.

A level set formulation of the above planar evolution rule is given by

$$\phi_t = \frac{1}{\sqrt{1+p^2+q^2}} \sqrt{(1+q^2)\phi_x^2 + (1+p^2)\phi_y^2 - 2pq\phi_x\phi_y},$$

where $p = dz/dx$, and $q = dz/dy$ are the derivatives of the surface $\mathcal{S}$, and $\mathcal{S}$ is assumed to be given as a function $(z(x,y),x,y)$.

The distance map on the surface may then be calculated as before, by propagating the planar curve (in its level set formulation) and assigning the distance to each point as the curve passes through that point. All the evolution rules presented here are implemented by numerical schemes that are consistent with the continuous propagation rule, and are motivated by numerical schemes that where developed in solving Hamilton Jacobi type of equations, [22, 21, 16]. Observe that all the planar evolutions described here are of that type. The consistency condition, that is satisfied when constructing the numerical scheme, guarantees that the solution converges to the true one as the grid is refined and the time step in the numerical scheme is kept with the right proportion to the grid size. This is known not to be the case in general graph search algorithms (*e.g.* Dijkstra) that suffer from *digitization bias* due to the *metrication error* when implemented on a grid, see [19].

6 Applications and results

In this section two examples of possible applications of using the sub–pixel distance maps based on the level set approach are presented. The first is the search for the Voronoi diagram between given points on surfaces (or weighted planar domains). The second deals with three point Steiner problems with cost function defined on the planar domain (the same procedure still apply when considering three points on a surface). Some examples of the distance and weighted distance transforms then demonstrate the capability of the algorithms devised.

6.1 Voronoi diagram on surfaces

It is possible to formulate a procedure that finds the Voronoi diagram on surfaces using the level set technique. Observe that the line that separates two points on the surface is given as the zero level set of the distance map from the first point, $\mathcal{M}_1$, subtracted from the distance map form the second point, $\mathcal{M}_2$. The projection of that curve is given by $l = \{(x,y)|\mathcal{M}_1(x,y) - \mathcal{M}_2(x,y) = 0\}$. When more than two points are involved things become more complicated, and similar approaches as those that are used in the planar case should be followed. A possible procedure that finds the projection of the diagram is the following

$$\text{Voronoi Projection} = \{(x,y)|\forall i \forall j, i \neq j : \mathcal{M}_i - \mathcal{M}_j = 0 \bigcap [\bigcap_{k\neq i,j} \mathcal{M}_i < \mathcal{M}_k]\},$$

where $i, j, k \in \{1, 2, ..., N\}$ are indexes of the given points on the surface.

6.2 Solving the three point Steiner problem

The *Steiner* problem is defined as follows: given three points (or areas), find the graph of minimal cost connecting the three points. In this problem we still try to solve the problem as defined in Equation (1), but now L is defined to be the set of all planar curves connecting the three points that may have junctions (brunches). The cost is accumulated by integrating along all brunches. With uniform penalty, the solution is quite simple and looks like a "**Y**", connecting the three points, the junction in the "middle" usually forming three 120^o angles. Given three points and a cost function on the domain, it is easy to show that the situation is similar, but now there might be more than one solution.

In order to solve the problem we first calculate the weighted distance map from each of the given points, then find the infimum of the sum of the three maps. There might be more than one point achieving this infimum if there is more than one solution. These points will be junction points of the solution. The last stage involves computing the weighted distance map from the junction, and determining the path of minimal cost from the junction to each of the original regions using the technique described in the previous sections (that is, finding the minimal level set of the sum of the weighted distance map from the junction and the distance map from the point). Combining the paths to the junctions results in the desired solution. The "cost" of the graph is the infimum of the sum of the three weighted distance maps calculated on the first step.

This is only a simple example of a wide variety of possibilities for using calculated weighted distance maps.

6.3 Examples

The first example shows the sub–pixel distance map of a given gray level picture. The low resolution (64×64) gray level picture Figure 3 is used to initialize the ϕ function, *i.e.* $\phi(x, y, 0) = E(x, y) - T$, where the threshold $T = 128$, and each gray level pixel is represented by an 8-bit number taking the values 0 (white) to 255 (black). Figure 3.**b** shows $\phi(x, y, 0) = (E(x, y) - T)/256$. The outward distance map is shown in Figures 3.**c** as equal distance contours on a gray level indicating the distance, and 3.**d** as distance array. The same procedure applied to the negative picture to achieve the inwards distance map, is shown in Figure 4.

In the last example , Figure 5, the weighted distance transform is used to find the path of minimal cost between two points on a plane with two different traversability factors (two regions with different cost). The same result is obtained by using well known Snell's law in optics, and of course this is not surprising in view of the fact that what we are doing is solving efficiently an eikonal equation.

7 Concluding remarks

We have presented a new approach to find sub–pixel accuracy distance transforms, by efficiently propagating equal distance contours. The proposed numerical approximation uses an implicit representation of the evolving contour. The contour is propa-

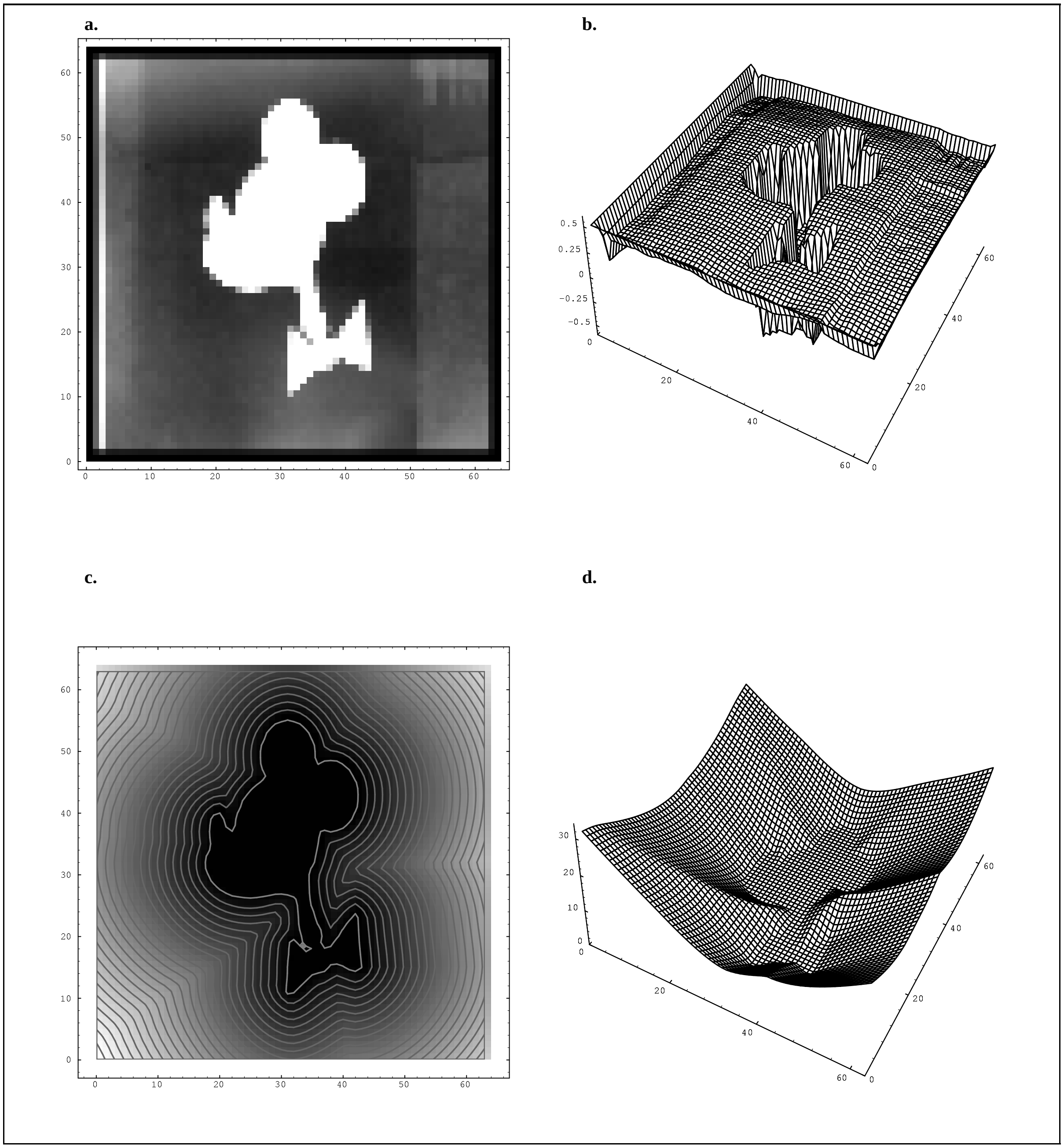

Figure 3: Sub–pixel distance map of a low resolution (64×64) gray level picture. **a.** The picture is given as 8-bit gray level numbers taking the values 0 (white) to 255 (black). **b.** $\phi(x, y, 0) = (E(x, y) - 128)/256$. **c.** The outward distance map as equal distance contours on gray levels indicating the distance. **d.** The distance map given as elevation array.

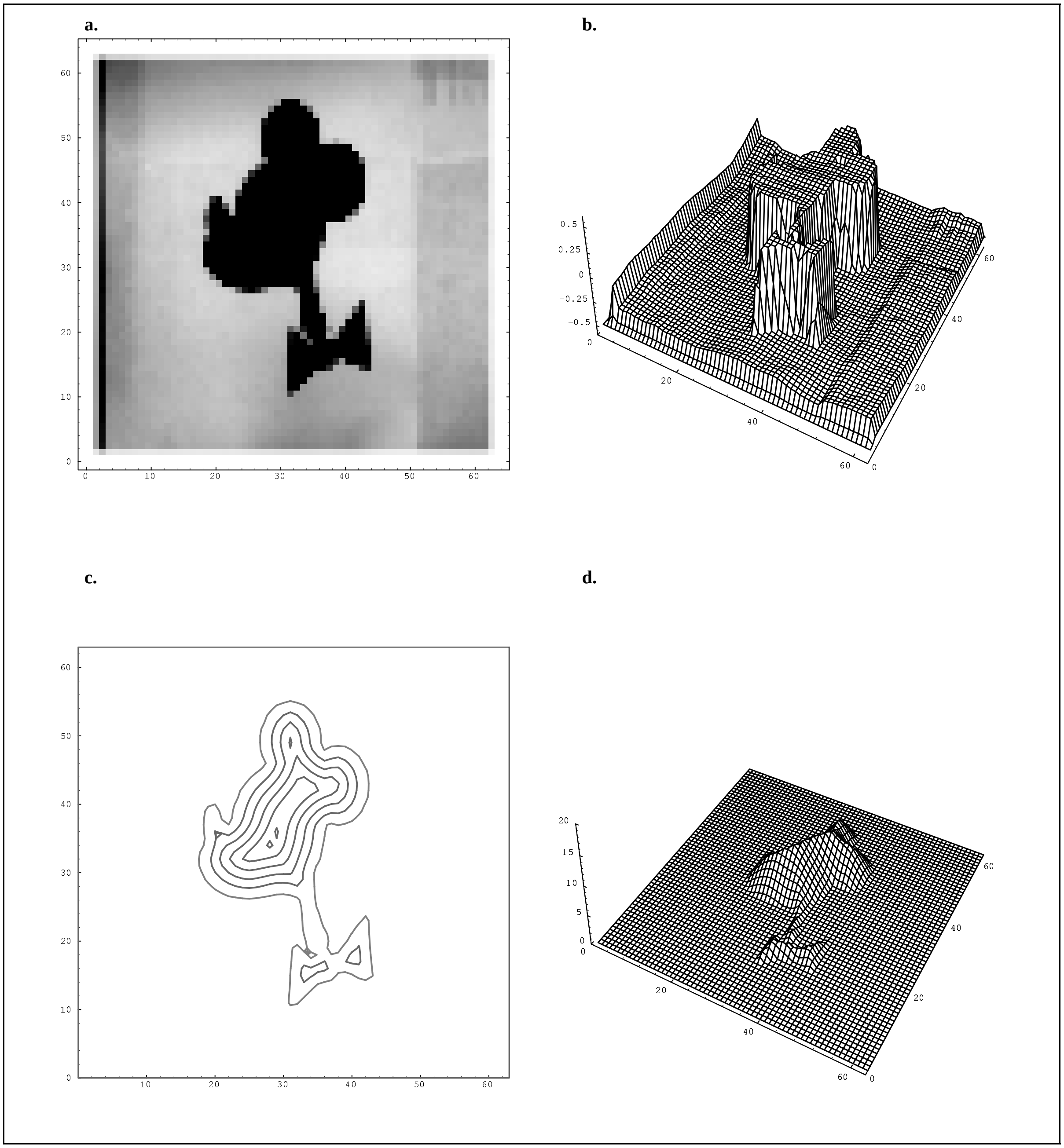

Figure 4: Sub–pixel distance map of a low resolution (64×64) gray level picture, the inverse of the previous one (**a**).

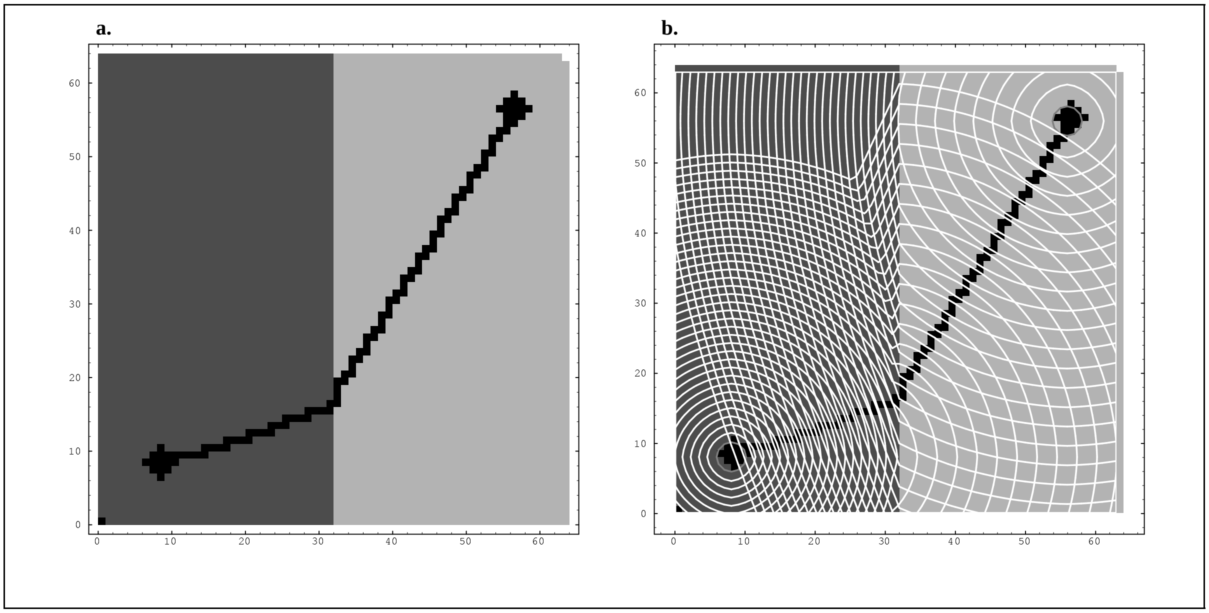

Figure 5: The weighted distance transform is used to find the path of minimal cost, between two points on two opposite sides of a low resolution picture (64×64). The picture displays the two different traversability factors as two gray levels.

gated as a level set of bivariate function. This way numerical problems and topological changes, encountered in traditional direct wave propagation schemes are naturally overcome. A natural initialization to the algorithm was suggested, according to which the gray level image of the shape provides the initial surface. This surface serves as the implicit representation of the initial contour.

A way to calculate the sub–pixel weighted distance transform was described. We have presented the relation of the weighted distance transform and the shape from shading problem, and used shape from shading solutions to calculate the path of minimal cost between two and then three points.

Continuous scale morphology via curve evolution was used to calculate distance maps for any given metric. Calculating the geodesic distance on surfaces was carried out by propagating a planar curve that is the projection of the three dimensional equal distance contour propagating on the surface.

Two examples of using the distance transforms were presented, the solution of the three points Steiner problem (on surfaces or on weighted domains), and the Voronoi diagram on surfaces.

8 Acknowledgments

We thank Dr. Guillermo Sapiro, Mr. Doron Shaked and Mr. Arnon Amir for many helpful discussions on curves, surfaces and distance maps. We also thank Dr.

Stefan Farestam, for noting the possibility of finding the Voronoi diagram of points on a given surface, and Prof. James Sethian and Prof. Stanley Osher for several interesting discussions on level sets and distance maps, during the summer of 1993. This research was supported in part by the Ollendorf Center of the Department of Electrical Engineering.

References

[1] M Bichsel and A P Pentland. A simple algorithm for shape from shading. In *Proceedings IEEE CVPR*, pages 459–465, Champaign, Illinois, May 1992.

[2] H Blum. Biological shape and visual science (part I). *J. theor. Biol.*, 38:205–287, 1973.

[3] R W Brockett and P Maragos. Evolution equations for continuous–scale morphology. In *Proceedings IEEE international Conference on Acoustics, Speech, and Signal Processing*, pages 1–4, San Francisco, California, March 1992.

[4] A M Bruckstein. On shape from shading. *Comput. Vision Graphics Image Process.*, 44:139–154, 1988.

[5] L Calabi and W E Hartnett. Shape recognition, prairie fires, convex deficiencies and skeletons. *Am. Math. Mon.*, 75:335–342, 1968.

[6] P Danielson. Euclidean distance mapping,. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[7] P Dupuis and J Oliensis. Direct method for reconstructing shape from shading. In *Proceedings IEEE CVPR*, pages 453–458, Champaign, Illinois, May 1992.

[8] B K P Horn. Height and gradient from shading. *Int. J. of Comput. Vision*, 5:37–75, 1990.

[9] B K P Horn and M J Brooks, editors. *Shape from Shading.* MIT Press, Cambridge MA, 1989.

[10] K Ikeuchi and B K P Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1-3):141–184, 1981. Also in Horn and Brooks, 1989.

[11] R Kimmel. Shape from shading via level sets. M.Sc. thesis (in Hebrew), Technion - Israel Institute of Technology, June 1992.

[12] R Kimmel, A Amir, and A M Bruckstein. Finding shortest paths on graph surfaces. Center for Intelligent Systems Report CIS #9301, Technion–Israel Institute of Technology, Israel, January 1993.

[13] R Kimmel and A M Bruckstein. Shape from shading via level sets. Center for Intelligent Systems Report CIS #9209, Technion–Israel Institute of Technology, Israel, June 1992.

[14] R Kimmel and A M Bruckstein. Distance maps and weighted distance transforms. In *Proceedings SPIE - Geometric Methods in Computer Vision II*, volume 2031, pages 259–268, San Diego, California, July 1993.

[15] R Kimmel and A M Bruckstein. Shape offsets via level sets. *CAD*, 25(5):154–162, March 1993.

[16] R J LeVeque. *Numerical Methods for Conservation Laws.* Lectures in Mathematics. Birkhauser Verlag, Basel, 1992.

[17] F Leymarie and M D Levine. Fast raster scan distance transform on the discrete rectangular lattice. *CVGIP: Image Understanding*, 55(1):84–94, 1992.

[18] P L Lions, E Rouy, and A Tourin. Shape-from-shading, viscosity solutions and edges. *Numerische Mathematik*, 64:323–353, 1993.

[19] J S B Mitchell, D Payton, and D Keirsey. Planning and reasoning for autonomous vehicle control. *International Journal of Intelligent Systems*, 2:129–198, 1987.

[20] S Osher and L Rudin. Rapid convergence of approximate solutions to shape-from-shading problem. Technical report, Cognitech Inc., 1993. Unpublished.

[21] S Osher and C W Shu. High–order essentially nonoscillatory schemes for Hamilton–Jacobi equations. *SIAM J. Numer. Analy.*, 28(4):907–922, August 1991.

[22] S J Osher and J A Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. of Comp. Phys.*, 79:12–49, 1988.

[23] A P Pentland. Local shading analysis. *IEEE Trans. on PAMI*, 6(2):170–187, 1984. Also in Horn and Brooks, 1989.

[24] A P Pentland. Linear shape from shading. *Int. J. of Comput. Vision*, (4):153–162, 1990.

[25] Y Pnueli and A M Bruckstein. DigiDurer - a digital engraving system. *to appear in The Visual Computer*, 1993.

[26] E Rouy and A Tourin. A viscosity solutions approach to shape-from-shading. *SIAM. J. Numer. Analy.*, 29(3):867–884, June 1992.

[27] G Sapiro, R Kimmel, D Shaked, B Kimia, and A M Bruckstein. Implementing continuous–scale morphology via curve evolution. *Pattern Recognition*, 26(9):1363–1372, 1993.

[28] G Sapiro and A Tannenbaum. Affine invariant scale–space. *Int. J. Comp. Vision.*, 11(1):25–44, 1993.

[29] J A Sethian. Curvature and the evolution of fronts. *Commun. in Math. Phys.*, (101):487–499, 1985.

[30] J A Sethian. A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed. *J. of Diff. Geom.*, (33):131–161, 1989.

[31] M Sussman, P Smereka, and S J Osher. A level set approach for computing solutions to incompressible two-phase flow. Department of Math, UCLA CAM Report 93-18, UCLA, Los Angeles, CA 90024-1555, June 1993.

[32] P W Verbeek and B J H Verwer. Shading from shape, the eikonal equation solved by gray-weighted distance transform. *Pattern Recognition Letters*, 11:681–690, 1990.

[33] L Vincent. Exact euclidean distance function by chain propagation. In *Proceedings IEEE CVPR*, pages 520–525, May 1991.

[34] S D Yanowitz and A M Bruckstein. A new method for image segmentation. *Computer Vision, Graphics, and Image Processing*, 46:82–95, 1989.