the calculation efforts low. We shall refer to such a saddle as the 'saddle of inflection type'.

Starting from singular points as sources of propagating equal distance contours, the merging points of the propagating contours should be saddles of the inflection type. A rule for the merging of the contours should be devised. In the sequel a rule that leads to a global shape from shading algorithm is presented.

5.4 The Global Shape from Shading Algorithm

As mentioned before, the main problems are to classify the singular points, propagate from the minimum points and use the inflection type saddle points as merging points. The solution to this 'three phase' problem seems impossible when one tries to break it into sequential parts, as was traditionally done [14]. Considering the global problem in hand we must use a global approach and the smoothness conditions (the surface being a Morse function) as the key to the solution. The algorithm has in two main steps.

Step 1 For *each* singular point, $\mathbf{m}_i$ ($i \in \{1, .., N\}$, where N is the number of singular points), calculate the 'weighted distance transform'

$$D_i(\mathbf{x}) = \inf_{\mathcal{L}(s) \in L_i} \left\{ \int_0^{|\mathcal{L}|} f(\mathcal{L}(s))ds \;:\; |\frac{\partial \mathcal{L}}{\partial s}| = 1,\ \mathcal{L}(0) = \mathbf{x},\ \mathcal{L}(|\mathcal{L}|) = \mathbf{m}_i \right\},$$

in **all** of the image domain $\Omega \in \mathbb{R}^2$. This transform is the correct solution in the case of a single singular point known to be a minimum (the maximum is the dual case).

The first singular point: $\mathbf{m}_k (\neq \mathbf{m}_i)$, that is identified as a saddle of the inflection type (Figure 5.3 lower row) according to the minimal distance in D_i, is labeled $P_i = \mathbf{m}_k$, its corresponding distance being H_i:

$$H_i = \{D_i(\mathbf{m}_k) \mid \ \mathbf{m}_k \text{ is an inflection type saddle in } D_i \ \},$$

Step 2 An iterative search for the 'proper solution' that satisfies the smoothness demands is performed. A 'merge' of two distance transforms is performed if the following conditions are satisfied:

- The two distance maps have the same singular point as the minimal distance inflection type saddle ($P_i \equiv P_j$).
- The equal height contours, that correspond to the height defined by the inflection type saddle ($D_i(\mathbf{x}) = H_i$ and $D_j(\mathbf{x}) = H_j$), osculate only at the singular point, and do not cut or touch each other elsewhere, *i.e.*

$$\{\mathbf{x} \;:\; D_i^{-1}(H_i) = D_j^{-1}(H_j)\} \equiv P_i (= P_j).$$

When those two conditions are satisfied, a merge or 'logical combination' of the two distance maps is performed as follows:

$$D_r(\mathbf{x}) = \min\{D_i(\mathbf{x}), D_j(\mathbf{x}) + H_j - H_i\}.$$

The map D_r and its minimal distance inflection type saddle if encountered P_r (of height H_r) are added to the array of distance maps. The merging steps are repeated until a map, for which there are no inflection type saddle points, is obtained. This map will be a 'legal' solution to the reconstruction problem. However, it is possible to continue the merging process until there are no more connections (inflection type saddle points) between any of the entries in the array. All the legal maps corresponding to the data will be found this way.

The second step of the algorithm finds all possible solutions to the given shape from shading problem, solutions that obey the Morse conditions at singular points.

The algorithm was implemented on a grid of pixels. The weighted distance map from each of the singular points was calculated using the equal height contour evolution via level sets implemented on the grid. In [66], the numerical properties of the level set algorithm, like accuracy, stability, noise effects, efficiency and its performance on real images were explored. The singular points were determined by considering the grid point of highest intensity (the brightest point) within a connected area of pixels, these *White* areas being isolated by thresholding the shading image, as shown in Chapter 4.

Given a weighted distance transform of a singular point, the first saddle point of the inflection type should be located. Consider a small simple closed curve around each of the singular points. The number of *zero crossings* of the reconstructed surface along that curve, where the *zero* is the height of that singular point, classify that point. For a saddle of the first type - four zero crossings will occur, for a maximum point there are no zero crossings, whereas for a saddle of the inflection type there are two zero crossings, see Figure 5.4. On the grid, the zero crossings were searched along a cyclic chain code representation of a square boundary around each singular point.

Having the weighted distance and the first saddle of the inflection type for each of the singular points the merging process begins. The procedure *Merge(k, l, N_T)* checks for possible merging of the the map k and the map l. If it is possible to merge the two maps, a new map is created and its inflection type saddle is located. The index variable N_T is incremented by one and assigned as a label to the new map. In case the two maps are merged and a inflection type saddle is not detected, the merge of the two maps is a legal solution.

Let N be the number of singular points, the following simple procedure reconstructs the surface:

$N_T = N$

for $k = 2$ **to** N

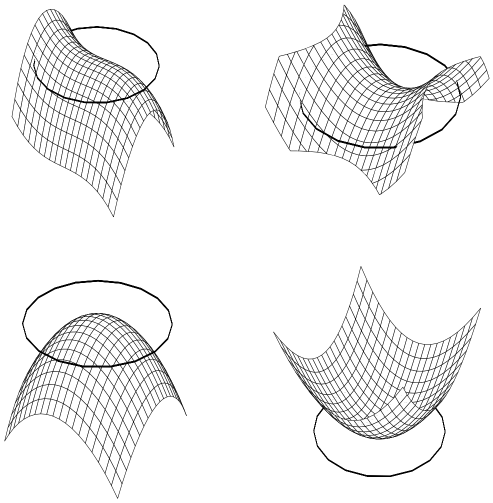

Figure 5.4: The number of crossings of the reconstructed surface with a planar circle, hovering at the singular point's height, classify that point. For a saddle of the inflection type that generates an inflection point there are two crossings (upper left). A first type saddle yields four crossings (upper right). Maximum and minimum points do not intersect the curve at all.

for $l = 1$ **to** $k - 1$

Merge(k, l, N_T)

for $k = N + 1$ **to** N_T

for $l = 1$ **to** N_T

if (k **and** l **were not** *Merge***d before)** *Merge*(k, l, N_T)

Note that the *end-of-loop* variable, N_T, in the second k–loop may change due to merging.

In order to better understand the merging algorithm consider the following simple example. Let the original object be composed of a maxima surrounded by four minima, see Figure 5.5.

After solving the local shape from shading problem for each of the 9 singular (maxima minima and saddle) points, the merging process went as follows, see Figure 5.6: D_4 and D_8 merged into D_{10} via the point 7. D_7 and D_9 merged into D_{11} via the point 8, D_1 and D_3 merged into D_{12} via the point 2, and D_{11} and D_{12} merged into D_{13} via the point 4 resulting in the first legal solution, where there are no points of the second saddle type.

It is possible that merging that do not lead to any solution are performed (*e.g.* 4 and 8 resulting 10), however, the final result of the algorithm is always a legal solution that satisfies the topology restrictions and the irradiance equation. We can let the merging process go on after the first legal solution is obtained and thereby detect all possible solutions (in cases there are more than one).

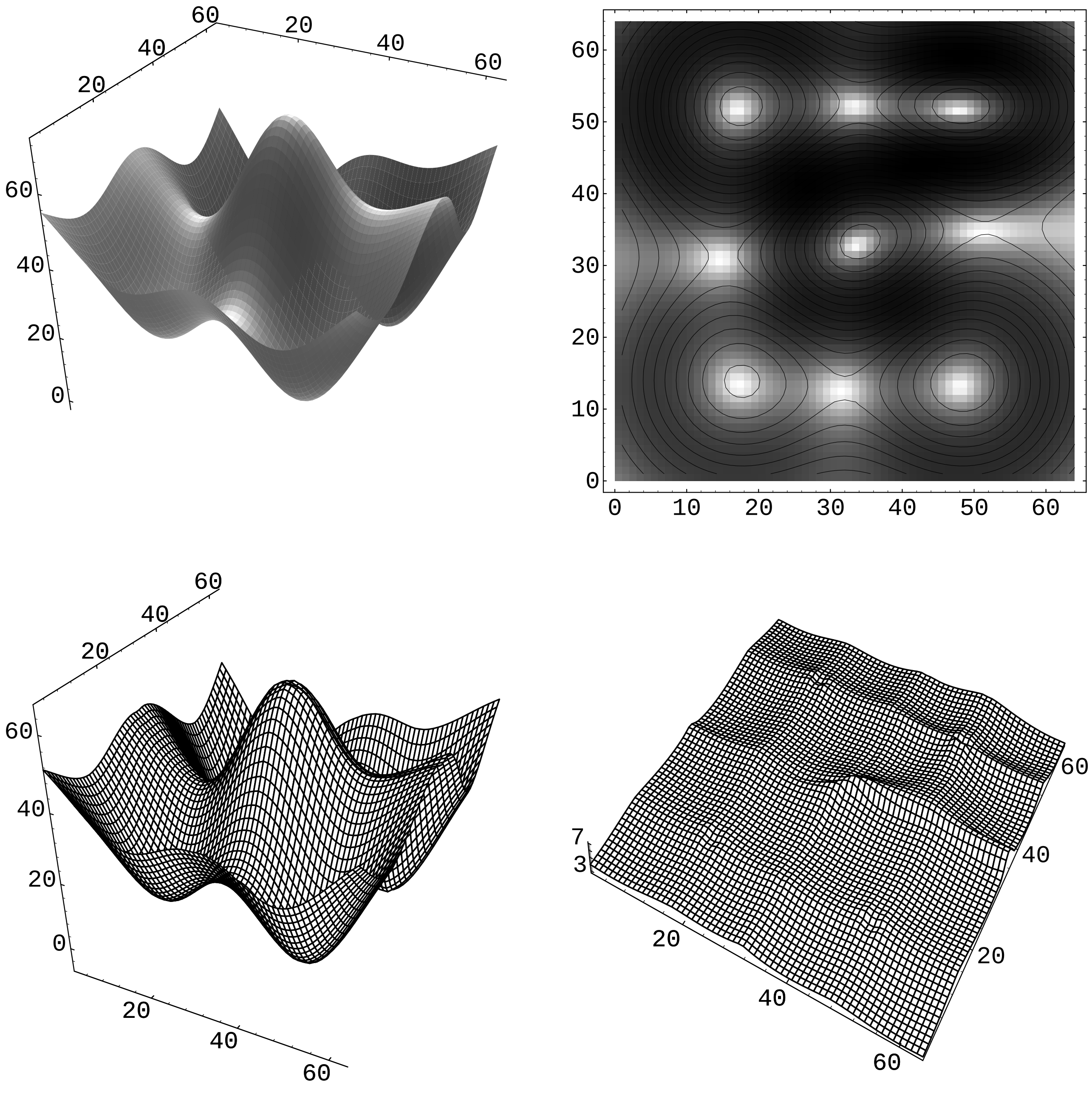

Figure 5.5: **Upper row:** The original surface and its shading image. The equal height contours of the original object are drawn on the shading image. **Lower row:** The result of the algorithm and the error surface obtained by subtracting the solution from the original object.

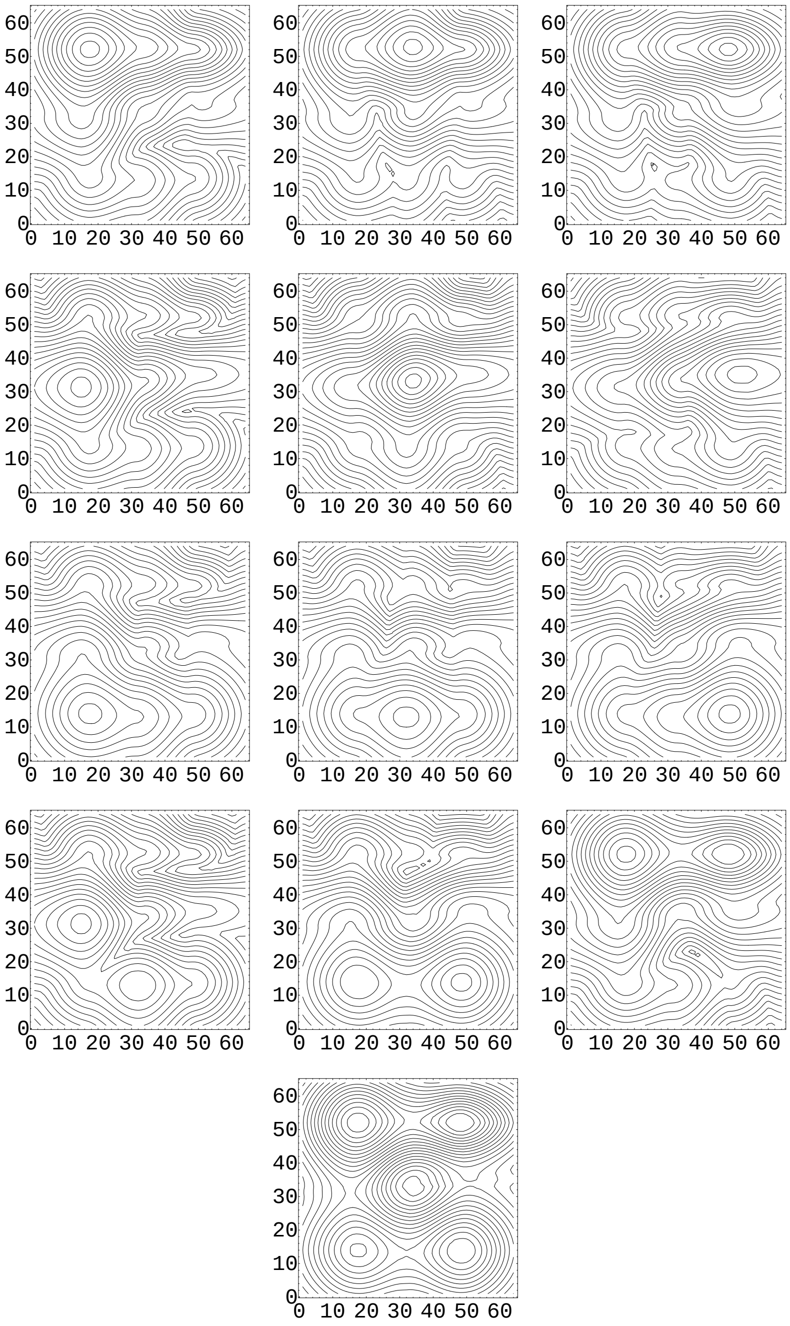

Figure 5.6: The weighted distance transforms from each of the singular points and the merging up to the first legal solution are presented as level contours, numbered 1 to 13, left to right top to bottom.

5.5 Examples

In the following examples the shape from shading via level set algorithm (Chapter 4, [66]) was used as the basic procedure that calculates the weighted distance transform from each singular point. Any other local shape form shading algorithm or weighted distance transform may be used for the same purpose at the first step of the algorithm. The shading images are given as 64×64 gray–level pixel arrays.

Figure 5.7 presents a volcano surface (upside down). In this example it is enough to start form the global minimum point (the 'top' of the upside-down volcano) in order to reconstruct the surface without any merging steps. When starting the propagation from the global minimum point, the saddle topology is of the first type, see Figure 5.3 upper row, and there is no need to perform any topological corrections in the reconstruction procedure. The weighted distance transform is the desired surface.

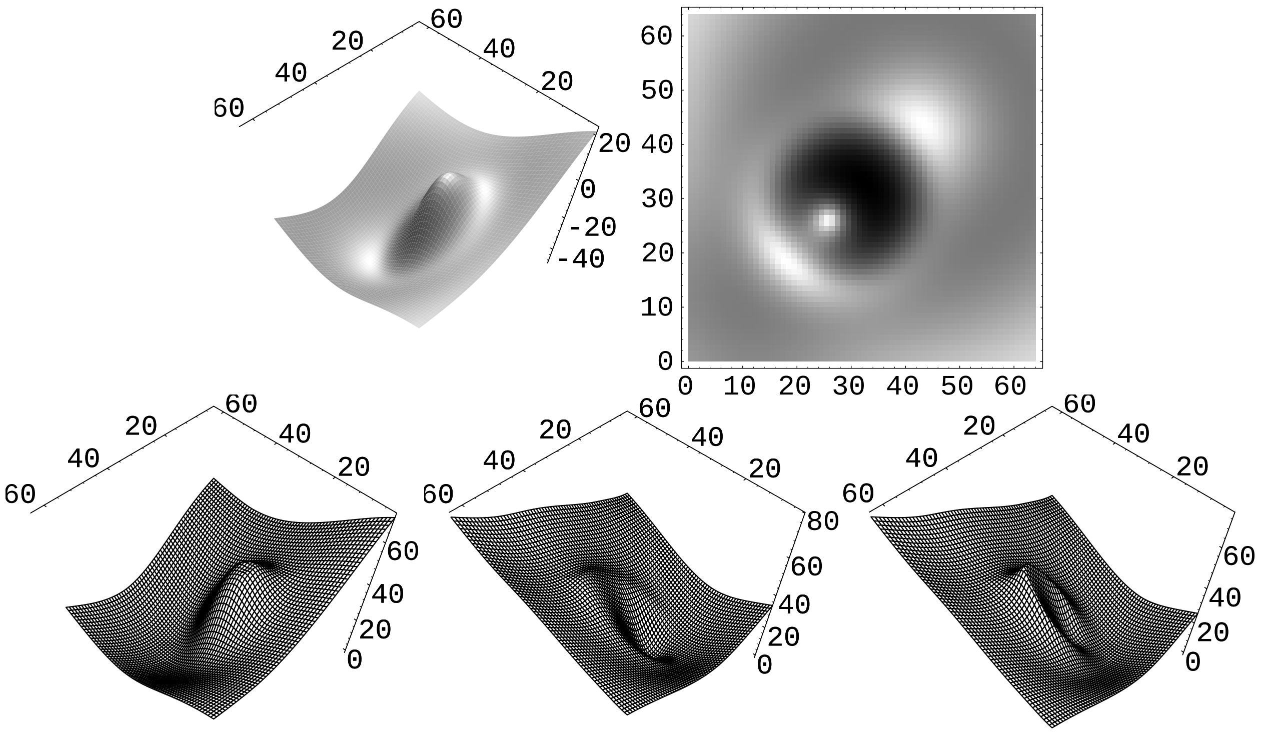

Figure 5.7: A shading image (upper right) of an upside-down volcano model (upper left), is given as an input to the global algorithm. The reconstructed surface (lower left), is the weighted distance transform of the global minimum point. Observe that the saddle is of the first type in this distance map. The other maps on the lower row (middle and right) do not contribute to the global reconstruction process in this case.

In the following examples the surfaces are shown upside down, making it easier to understand the reconstruction process and the surfaces structure.

Figure 5.8 presents a simple surface containing two maximum points and a saddle between them. The upper row displays the surface and its shading image. The three weighted distance maps from each of the singular points are then computed and presented in the middle row of Figure 5.8. The logical combination, as previously defined, of the two minimal points, yields the surface in the lower row of Figure 5.8. This surface is the logical combination of the left and right surfaces in the middle row which share the same saddle point.

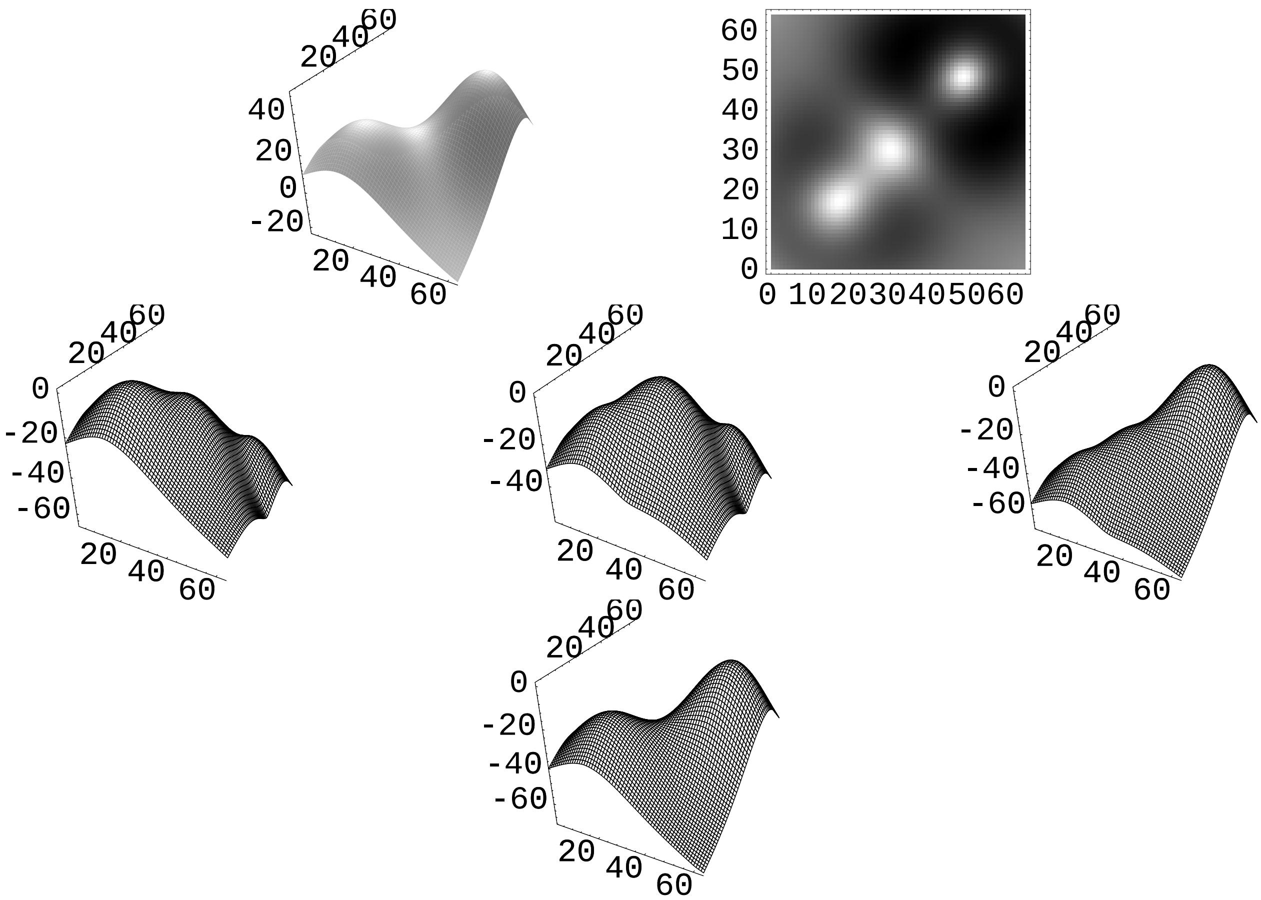

Figure 5.8: **Upper row:** The original surface and the shading image. **Middle row:** The weighted distance transform from each of the singular points (from the two maximal points and the saddle point). **Lower row:** The 'logical combination' of the distance maps from the two maximal points and the saddle as the merging point.

In Chapter 1 a more complicated surface, with 11 singular points of all possible types, is dealt with. Figure 1.3 (upper row on the left) presents the original surface and its shading image (the upper row on the right). The final reconstructed surface and the error are displayed in the lower row.

5.6 Chapter Summary

In this chapter a new approach to solve the global shape from shading problem was presented. The topological properties of simple smooth surfaces were used to construct an algorithm that uses a logical combination of the weighted distance transforms which are calculated form each of the singular points.

It is possible to implement the first step of the algorithm in parallel. The calculation efforts in computing the weighted distance transform of each of the singular points is determined by the maximal weighted distance from the singular point to the image boundary $\partial\Omega \in \mathbb{R}^2$. Define this maximal distance as $g = \sup_{i\in\{1,..,N\}}\{z_i(\mathbf{x}) \ : \ \mathbf{x} \in \partial\Omega\}$.

The second step may also be performed in parallel. The computation complexity in this case is at the order of the number of merging steps that should be performed until the desired result is obtained. The number of iterations in the second step is therefore bounded by the number of saddle points, and according to the mountaineers theorem this number is given by $(N-1)/2$.

The total calculation complexity for a parallel machine is of order $\mathcal{O}(g+(N-1)/2)$. The memory involved in the solution should be at least of order $\mathcal{O}(\Omega N)$, and its upper bound is determined by the number of possible merging steps.

Chapter 6

Finding Shortest Paths on Surfaces

In this chapter we present a new algorithm for determining minimal length paths between two regions on a three dimensional surface [63, 64, 65]. The numerical implementation is based on finding equal geodesic distance contours from a given area. These contours are calculated as zero sets of a bivariate function designed to evolve so as to track the equal distance curves on the given surface. The algorithm produces all paths of minimal length between the source and destination areas on the surface given as height values on a rectangular grid.

6.1 Chapter Introduction

Finding paths of minimal length between two areas on a three dimensional surface is of great importance in many fields such as computer aided neuroanatomy, robotic motion planning (autonomous vehicle navigation), geophysics, terrain navigation, etc. Paths of minimal Euclidean distance between two points on a surface are usually referred to as *minimal geodesics*.

A new approach of dealing with the problem of finding the minimal distance paths, in which the surface is given as height samples on a rectangular grid is introduced. As a first step, a distance map from the source area is calculated. The distance map is computed via *equal distance curve* propagation on the surface. Equal distance curves are calculated as the zero sets of a bivariate function evolving in time, see Chapter 2.

As noted in Chapter 2, the proposed numerical scheme is consistent with the continuous propagation rule. The consistency condition guarantees that the solution converges to the true one as the grid is refined and the time step in the numerical scheme is kept in the right proportion to the grid size. This is known not to be the case in general graph search algorithms that suffer from *digitization bias* due to the *metrication error* when implemented on a grid [99, 83].

The relation between minimal paths, geodesics and equal distance contours may be found in elementary differential geometry textbooks, *e.g.* [38]. Geodesics are locally shortest paths in the sense that any perturbation of a geodesic curve will increase its length. The minimal length paths between two points are the *minimal geodesics* connecting those points. A simple way of determining minimal geodesics is by constructing a so–called *geodesic polar coordinate system* on the surface around the source area. Using such a coordinate system readily provides the *geodesic circle map*, or the map of equal distance contours on the surface.

In the next Section an analytic model for the equal distance contour evolution is discussed. In Section 6.3, a numerical implementation of the analytic propagation is presented. The results of the numerical algorithm are demonstrated by an example in Section 6.4. We conclude with a discussion of some possible extensions of the algorithm and comment on its complexity in Section 6.5.

6.2 Equal Geodesic–Distance Contour Propagation

Let us first define a differential equation describing the propagation of equal geodesic–distance contours on a smooth surface starting from a point or a source region on the surface. Given a source area $S \in \mathbb{R}^3$ (S is not necessarily connected), on a surface $\mathcal{S} \in \mathbb{R}^3$, let the $3D$ equal distance contour of distance t from S be defined as

$$\{\mathbf{p} \in \mathcal{S} | d_{\mathcal{S}}(p, S) = t\} = \alpha(*, t),$$

where $d_{\mathcal{S}}(\mathbf{p}, S)$ is the *minimal geodesic distance* determined by the shortest paths from a point $\mathbf{p}$ to an area S on the surface $\mathcal{S}$.

We shall prove that the $3D$ parametric representations of $\alpha(*, t)$, on $\mathcal{S}$, can be obtained by the equal distance contour propagation

$$\alpha_t = N \times \mathcal{T}^{\alpha}, \quad \text{given} \quad \alpha(u, 0) = \alpha(u), \tag{6.1}$$

where $\mathcal{T}^{\alpha}$ is the tangent unit vector to α, and N is the surface normal.

The traces of constant parameter along the curve evolving according to (6.1) are geodesics, and these geodesics are locally shortest paths. We have the following results:

Lemma 6.1 *Define the curve $\beta(t) = \alpha(u, t)|_{u=u_0}$. Then, for any u_0, the curve $\beta(t)$ is a geodesic.*

Proof. The trace $\beta(t)$ is determined by the evolution of $\alpha_t = N \times \mathcal{T}^{\alpha}$, hence $\beta_t = N \times \mathcal{T}^{\alpha}$. Since $|\beta_t| = |N \times \mathcal{T}^{\alpha}| = 1$, the t parameter is the arclength of β.

In order to show that β is a geodesic we recall the definition of geodesics and prove that $\frac{d^2}{dt^2}\beta = \lambda N$,where $\lambda(t)$ is a scalar function. A geometric interpretation of this formula is that the second derivative of the curve (its normal direction $k\mathcal{N}^{\beta}$) is in the surface normal direction N. To prove the result it is sufficient to verify that

$$\langle \beta_{tt}, \beta_t \rangle = 0 \quad \text{and} \quad \langle \beta_{tt}, \mathcal{T}^{\alpha} \rangle = 0,$$

or explicitly that

$$\langle \frac{d}{dt}[N \times \mathcal{T}^{\alpha}], N \times \mathcal{T}^{\alpha} \rangle = 0 \quad \text{and} \quad \langle \frac{d}{dt}[N \times \mathcal{T}^{\alpha}], \mathcal{T}^{\alpha} \rangle = 0$$

which clearly force $\beta_{tt} = \lambda N$. Let us first define $k, \mathcal{N}^\beta$ and $\mathcal{T}^\beta$ to be the curvature, normal and tangent of $\beta(t)$, respectively. Using the Frenet formulas we first have

$$\langle \frac{d}{dt}[N \times \mathcal{T}^\alpha], N \times \mathcal{T}^\alpha \rangle = \langle \beta_{tt}, \beta_t \rangle = \langle k\mathcal{N}^\beta, \mathcal{T}^\beta \rangle = 0.$$

The second expression may be written as follows

$$\langle \frac{d}{dt}[N \times \mathcal{T}^\alpha], \mathcal{T}^\alpha \rangle = \langle \beta_{tt}, \mathcal{T}^\alpha \rangle = \langle \alpha_{tt}, \frac{\alpha_u}{|\alpha_u|} \rangle.$$

Therefore, we should prove that $\langle \alpha_{tt}, \frac{\alpha_u}{|\alpha_u|} \rangle = 0$. First note that

$$\frac{d}{dt}\langle \alpha_t, \frac{\alpha_u}{|\alpha_u|} \rangle = \frac{d}{dt}\langle N \times \mathcal{T}^\alpha, \mathcal{T}^\alpha \rangle = \frac{d}{dt}0 = 0.$$

Using inner product rules we have

$$\frac{d}{dt}\langle \alpha_t, \frac{\alpha_u}{|\alpha_u|} \rangle = \langle \alpha_{tt}, \frac{\alpha_u}{|\alpha_u|} \rangle + \langle \alpha_t, (\frac{\alpha_u}{|\alpha_u|})_t \rangle.$$

Therefore, in order to show that $\langle \alpha_{tt}, \frac{\alpha_u}{|\alpha_u|} \rangle = 0$, it is enough to show that $\langle \alpha_t, (\frac{\alpha_u}{|\alpha_u|})_t \rangle = 0$.

Define the metric along the curve to be $g \equiv |\alpha_u|$, and compute

$$\frac{d}{dt}\frac{\alpha_u}{|\alpha_u|} = \frac{d}{dt}\mathcal{T}^\alpha = \frac{\alpha_{ut} g - \alpha_u g_t}{g^2} = \frac{\alpha_{ut}}{g} - \frac{\alpha_u g_t}{g^2}.$$

Using this result we proceed

$$\begin{aligned}
\langle \alpha_t, (\frac{\alpha_u}{|\alpha_u|})_t \rangle &= \langle \alpha_t, \frac{\alpha_{ut}}{g} - \frac{\alpha_u g_t}{g^2} \rangle \\
&= \langle \alpha_t, \frac{\alpha_{ut}}{g} \rangle - \langle \alpha_t, \alpha_u \frac{g_t}{g^2} \rangle \\
&= \frac{1}{g}\langle \alpha_t, \alpha_{tu} \rangle - \frac{g_t}{g^2}\langle N \times \mathcal{T}^\alpha, \mathcal{T}^\alpha \rangle \\
&= \frac{1}{g}\frac{1}{2}\frac{d}{du}\langle \alpha_t, \alpha_t \rangle - \frac{g_t}{g^2}0 \\
&= \frac{1}{g}\frac{1}{2}\frac{d}{du}1 \\
&= 0.
\end{aligned}$$

Therefore, $\langle \beta_{tt}, \mathcal{T}^\alpha \rangle = 0$, and this proves that $\beta(t)$ is a geodesic. ■

Let $\alpha(u, t)$ be a $3D$ curve propagating on the surface $\mathcal{S} \subset \mathbb{R}^3$, where u is the parameter and t is the propagation time. Then

Lemma 6.2 *The equal distance contour evolution is given by*

$$\alpha_t = N \times \mathcal{T}^\alpha \quad \textit{given} \quad \alpha(0).$$

Proof. As a first step we shall use *Gauss Lemma* to show that the asserted evolution rule formulates *geodesic polar coordinates* when starting from a point. The *geodesic circles* in this coordinates system are the equal distance contours. This result is then generalized to any given initial curve.

Define the tangent plane to a point p on the surface $\mathcal{S}$ as $T_p(\mathcal{S})$. Let $\vec{w} \in T_p(\mathcal{S}), |\vec{w}| = 1$ be a unit vector indicating a direction from p in the tangent plane. Use a polar coordinate system to define $\vec{w}$ on $T_p(\mathcal{S})$ as $\vec{w}(u) = \sin(u)\hat{x} + \cos(u)\hat{y}$. Define $\gamma^u(t)$ to be the geodesic which starts from p, with $\gamma^u(0) = p$, and $\gamma^u_t(0) = \vec{w}(u)$, where t stands for the arclength.

According to *Gauss Lemma*, see *e.g.* [38] pp. 287, the radial geodesics $\gamma^u(t)$ together with the geodesic circles $\eta(u,t) = \gamma^u(t)|_{t=\text{const},u\in S^1}$, form geodesic polar coordinates, in which the geodesic circles are orthogonal to the radial geodesic.

Considering the constant parameter traces along the evolving contours as the radial geodesics $\beta(t) = \gamma^u(t)$, and the contours themselves as the geodesic circles $\alpha(u) = \eta(u)$, we obtain that the equal distance contour evolution rule is given by the asserted equation, when starting from a point. To be more precise, we start from a given infinitesimal geodesic circle around the point.

We now proceed and generalize the result to any given curve $\alpha(u,0) = \alpha(0)$ on the surface.

Let P be the set of points forming the equal distance contour of distance d from $\alpha(u,0)$ on the given surface. Propagate an equal distance contour $\eta(v,t)$ starting from any point $p \in P$. Stop the propagation when the equal distance contour first touches $\alpha(u)$, let say at $q = \alpha(u_0) = \eta(v_0,\tau)$. According to the construction $\tau = d$, and therefore, $q = \eta(v_0,d)$. See Figure 6.1a.

At q, $\eta(v,d)$ and $\alpha(u)$ osculate, which means that $\mathcal{T}^\alpha|_{u=u_0}$ is parallel to $\mathcal{T}^\eta|_{v=v_0}$, see Figure 6.1b.

We have shown earlier that the shortest path from p to q is given by the radial geodesic $\beta(t) = \gamma^{v_0}(t) = \eta(v,t)|_{v=v_0}$, and where $\beta(t)$ and $\eta(v,t)$ are orthogonal along $v = v_0$ (Gauss lemma). Hence, $\mathcal{T}^\beta|_{t=d} \perp \mathcal{T}^\eta|_{t=d,v=v_0}$, and therefore,

$$\mathcal{T}^\beta|_{t=d} \perp \mathcal{T}^\alpha|_{u=u_0}.$$

We have just proved that the shortest paths from each point in the set P to α, is given by the geodesics starting from $\alpha(u)$ and orthogonal to $\mathcal{T}^\alpha$. This geodesic is the one obtained via the asserted evolution rule; using the continuity of α, the equal distance contour from $\alpha(u,0)$, is obtained by the asserted evolution rule, $\alpha_t = N \times \mathcal{T}^\alpha$. ■

Lemma 6.2 provides the evolution equation of the equal distance contour. Starting from the boundary of the source area $\alpha(0) = \{(x,y,z(x,y))|(x,y,z(x,y)) \in \partial S\}$, it is possible to

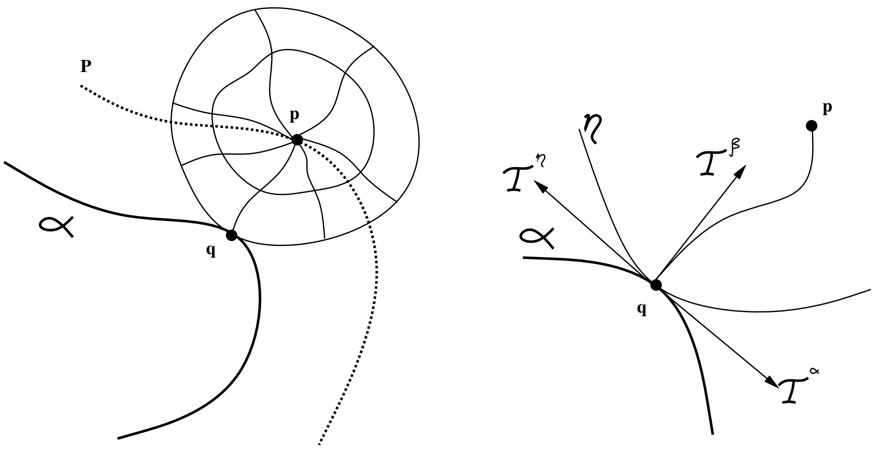

Figure 6.1: a. Equal distance contours are propagated from p, until the contour first touches $\alpha(u)$. b. The tangent point, observe that at q: $\mathcal{T}^{\beta}|_{u=u_0} \parallel \mathcal{T}^{\eta}|_{v=v_0}$.

find the equal distance contour for any desired distance d, by using the evolution equation to calculate $\alpha(u,t)|_{t=d}$. This propagation may be used to build the distance map for each point on the surface.

Implementing the three dimensional curve evolution is quite a complicated task. We are therefore interested in considering the projection of the $3D$ curve on the (x,y)–plane,

$$\mathcal{C}(t) = \pi \circ \alpha \equiv \{(x,y)|(x,y,z(x,y)) \in \alpha(t)\}.$$

A result from the general theory of curve evolution states that the trace of a propagating planar curve may be determined only by its normal velocity [43], see Chapter 2. Let us consider the projection of the above evolution on the (x,y)–plane (Figure 6.2). The knowledge of how this projected contour behaves allows us to construct a simple, accurate and stable numerical algorithm that can be used to produce these equal distance contours.

In [63] we calculate the planar normal component of the projected velocity of the evolving equal distance contour, $V_N = \langle \mathcal{N}, \pi \circ (N \times \mathcal{T}^{\alpha}) \rangle$, see Chapter 4. Using this velocity we construct a differential equation describing the projected equal distance contour evolution of the form

$$\frac{\partial}{\partial t}\mathcal{C} = V_N \mathcal{N} \quad \text{given} \quad \mathcal{C}(0) = \{(x,y)|(x,y,z(x,y)) \in \partial S\} \equiv \partial \tilde{S}, \tag{6.2}$$

where $\mathcal{N}$ is the planar normal direction, and V_N depends on the surface gradient ($p = \partial z/\partial x$ and $q = \partial z/\partial y$) and $\mathcal{N}$. Writing the planar normal as its components $\mathcal{N} = (n_1, n_2)$,

$$V_N = \sqrt{\frac{(1+q^2)n_1^2 + (1+p^2)n_2^2 - (2pq)n_1 n_2}{1+p^2+q^2}}$$

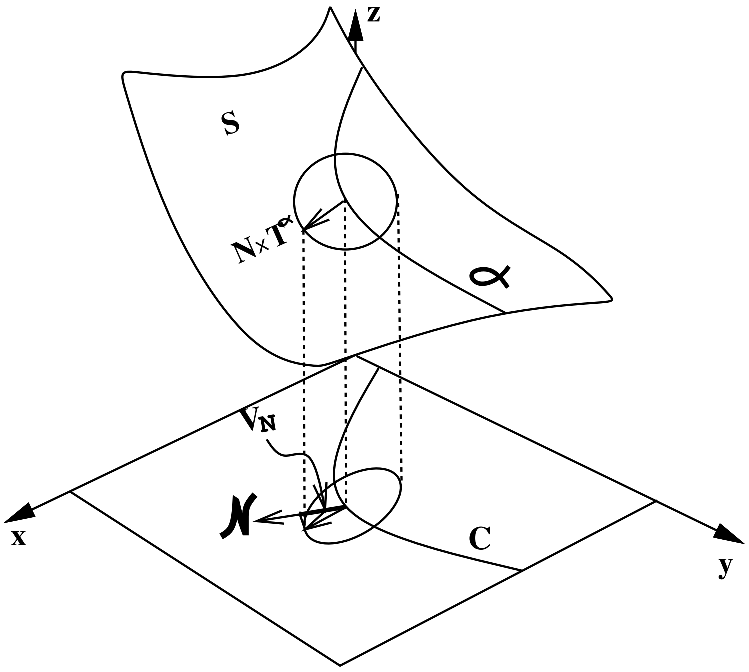

Figure 6.2: The projection of an infinitesimal geodesic circle on the surface forms a tilted ellipse on the (x, y)–plane.

or $V_N = \sqrt{an_1^2 + bn_2^2 - cn_1n_2}$, where the coefficients a, b and c depend on the surface gradient and can be computed once at the initialization step.

6.2.1 Finding the minimal path

The procedure that calculates the equal distance contours allows us to build a Euclidean distance map on the surface, from a given area. Assuming we have reliable distance map procedure in hand, we can construct a simple procedure that finds the minimal path from a source area S to a destination area D (where $S, D \in \mathcal{S}$).

Define $\mathcal{M}_A$ as the distance map of area A as

$$\mathcal{M}_A(x, y) = d_{\mathcal{S}}((x, y, z(x, y)), A).$$

we readily have the following result

Lemma 6.3 *All minimal paths between S and D on $\mathcal{S}$ are given by the set $G \subset \mathcal{S}$,*

$$G = \{(x, y, z(x, y)) | \mathcal{M}_S(x, y) + \mathcal{M}_D(x, y) = g_m\} \tag{6.3}$$

where $g_m \equiv \min_{(x,y)}(\mathcal{M}_S + \mathcal{M}_D)$ is the global minimum of the sum of the source and destination distance maps.