where the correction factor v_n is calculated in the Appendix at the end of this chapter (see also [63] for a similar approach to solve the shortest path problem on surfaces). The fact that the contours lie on parallel planes, constant heights with respect to $\hat{l}$, enables us to predict the "deformation" in time of $\tilde{F}$. Define $cos\beta = \langle \hat{z}, \hat{l} \rangle$ (where $\hat{z} \equiv (0,0,1)$), then $\tilde{F}(x,y,t) = F(x - tp_l/\sqrt{1+p_l^2+q_l^2}, y - tq_l/\sqrt{1+p_l^2+q_l^2})$, which means that the image shifts in time along the direction defined by $(-p_l, -q_l)/\sqrt{p_l^2+q_l^2}$ with a constant velocity of $\sin\beta$, see Figure 4.4a.

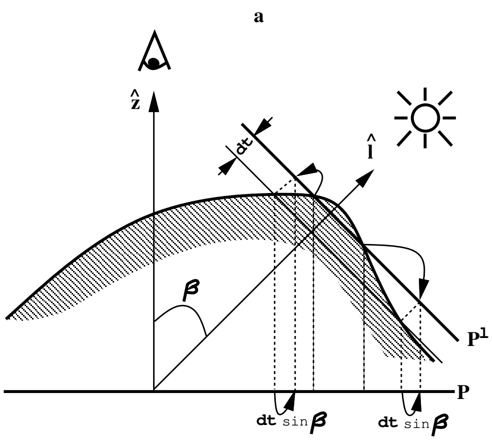


Figure 4.4: **a.** When propagating an equal height contour with respect to $\hat{l}$ on P^l, the underlying image which specifies the propagation velocity shifts in time with a constant velocity of $\sin\beta$. **b.** The constant velocity of the shifting image may be converted to a constant shifting of the evolving curve.

This formulation forces us to interpolate the image intensity between the pixels as the image moves in time. Another problem is the fact that we calculate the heights with respect to P^l so that in the last stage we should project the heights from P to P^l, or stretch the elevation array given on a uniform grid in P with respect to the light source direction. This stretching results in a non uniform elevation array.

Another manipulation on the evolution equation frees us from the need to take care of these problems. Instead of shifting the image forward it is possible to include a backwards propagation constant velocity term in the curve evolution rule. This way we can keep the image in its place and transfer the constant motion to the evolving curve, see Figure 4.4b. It also frees us from the need to project the heights to the P^l plane, and the height at each grid point may be calculated without any interpolation. The new evolution rule is given by

$$\mathcal{C}_t = F(x,y)v_n\mathcal{N} - \sin\beta \frac{(-p_l, -q_l)}{\sqrt{1+p_l^2+q_l^2}}, \tag{4.5}$$

and again, taking only the normal component of the second term (the tangential components have no effect on traces of the evolving curves), we obtain:

$$\mathcal{C}_t = \left(F(x,y)v_n + \frac{(p_l n_1 + q_l n_2)}{\sqrt{1+p_l^2+q_l^2}} \right) \mathcal{N}. \tag{4.6}$$

where $\mathcal{N} = (n_1, n_2)$. By using the v_n obtained in the Appendix, we define

$$\begin{aligned} G(x,y,\mathcal{N}) &= \frac{1}{\sqrt{1+p_l^2+q_l^2}} \times \\ &\left(F(x,y)\sqrt{n_1^2(1+q_l^2)+n_2^2(1+p_l^2)-n_1 n_2 2 p_l q_l} + (p_l n_1 + q_l n_2) \right). \end{aligned} \tag{4.7}$$

Then, equation (4.6) may by written as

$$\mathcal{C}_t = G(x,y,\mathcal{N})\, \mathcal{N}, \tag{4.8}$$

which is again a "Lagrangian" formulation of the new evolution rule. Note that equation (4.3) is a special case of equation (4.8).

4.4 Solution via the Eulerian Formulation

As shown in Chapter 2, the Eulerian scheme [111] is a recursive procedure that propagates height contours while inherently implementing a so-called *entropy condition*. This entropy condition is necessary to select the "physically correct" propagation [87] among all possible weak solutions.

Introduce a function $\phi(x,y,t)$ initialized so that $\phi(x,y,0) = 0$ yields the curve $\mathcal{C}(0)$. Assume that $\mathcal{C}(0)$ is a closed curve and restrict ϕ to be negative in the interior and positive in the exterior of the level set $\phi(x,y,0) = 0$. Furthermore ϕ has to be smooth and Lipschitz continuous.

The idea is to determine an evolution of the surface $\phi(x,y,t)$ so that the level sets $\phi(x,y,t) = 0$ provide the height contours $\mathcal{C}(t)$ as if propagated by (4.8) and obeying the entropy condition. In our case the velocity is given by the scalar function $G(x,y,\mathcal{N})$ as defined in equation (4.7), which is a function of the local image brightness and the given normal of the propagating curve. Following the derivation in Chapter 2, the Eulerian formulation for our problem is given by

$$\phi_t - G\|\nabla\phi\| = 0, \tag{4.9}$$

or explicitly

$$\phi_t = \frac{1}{\sqrt{1+p_l^2+q_l^2}} \times \tag{4.10}$$

$$\left(F(x,y)\sqrt{\phi_x^2(1+q_l^2)+\phi_y^2(1+p_l^2)-\phi_x\phi_y 2p_l q_l}\;-(p_l\phi_x+q_l\phi_y)\right).$$

Therefore, if we have a surface ϕ propagating according to (4.10) with the level set $\phi(x,y,0)=0$ coinciding with $\mathcal{C}(0)$, then $\phi(x,y,t)=0$ will produce $\mathcal{C}(t)$ propagated according to (4.8) and solving topological problems. In order to drive a numerical scheme for the surface propagation equation which obeys the natural "entropy condition", we modify results obtained in [111] (see also [128]) based on Hamilton Jacobi equations, weak solutions and conservation laws.

Following Chapter 2, define $\Phi_{i,j}^n \equiv \phi(i\Delta x, j\Delta y, n\Delta t)$. We use a simple forward derivative approximation in time

$$\phi_t|_{t=n\Delta t} \approx \frac{\Phi^{n+1}-\Phi^n}{\Delta t},$$

and slope limiter approximations [87] for the spatial derivatives. The squared terms are approximated by

$$\phi_x^2|_{x=i\Delta x, y=j\Delta y} \approx \max(D_+^x\Phi_{i,j}, -D_-^x\Phi_{i,j}, 0)^2,$$

and the same for ϕ_y^2. Where $D_-^x\Phi_i^n$ and $D_+^x\phi_i^n$, and the y derivatives are defined in Chapter 2. The minmod [87] function, as defined in Chapter 2, is used to approximate ϕ_x and ϕ_y:

$$\phi_x|_{x=i\Delta x, y=j\Delta y} \approx \operatorname{minmod}(D_-^x\phi_{i,j}, D_+^x\phi_{i,j}),$$

and the same for ϕ_y.

All the above approximations yield a so–called first order numerical scheme that approximates equation (4.10). This scheme automatically enforces the entropy condition, and frees one from the need to take care of topological changes. In fact this formulation deals with the topology of all up going (or down going) surfaces with respect to the light source direction without any external control or outside interference.

The algorithm also deals with shock formation in the propagating contours which indicates sharp corners in the reconstructed surfaces within the numerical flow. One of the great advantages of the Eulerian formulation is that the coordinate system of the propagated ϕ function is fixed, thereby avoiding the stability problems of the Lagrangian formulation. More sophisticated higher order schemes are presented in [111, 108].

In our problem the velocity G is position and normal dependent. In order to avoid numerical problems in which $F\to\infty$, we truncate the brightness function to the maximum values of $I_{max}<1$, which yields $F_{max}=\frac{I_{max}}{\sqrt{1-I_{max}^2}}$.

4.4.1 Initialization and height assignment

Every ϕ function which obeys the demands described earlier provides a good initialization. We present a simple way to initiate the ϕ function, obeying smoothness, continuity so that $\phi(x,y,0)=0$ gives the initial contour. Let us define the initial height contour by first thresholding the gray–levels in the picture and separating all the "singular" areas (the *singular areas* is the set of points in the image for which $I(x, y) \approx 1$, at such points the normal is known to be $\hat{l}$ and therefore may be used as boundary conditions or candidates to start the propagation). The "singular" points are the saddle, minima and maxima points with respect to the light source direction. Then use the gray–level function to initiate the ϕ function in a simple way. For example, if the singular areas are defined by $\{(x, y) : I(x, y) > T\}$ (where $T \in (0, 1)$ is the selected threshold), then the first level contour can be approximated as the level set $I(x, y) = T$. In this case we can take $\phi(x, y, 0) = I(x, y) - T$ near the selected singular areas, as the required initialization, making direct use of the continuity of the gray levels in the picture, without any extra calculations. This is the initialization method used in our examples. Observe that the set $I(x, y) = T$ is not a set of discrete points, but an implicit representation of a curve given by the zero level set of $\phi(x, y, 0)$.

After initialization has been completed the ϕ is propagated according to the above described algorithm. Our goal is finding the elevation at each grid point while the ϕ function is propagated on the grid. A way to achieve accurate results using a simple linear interpolation is as follows:

At every iteration step, for each grid point, check
If $(\Phi^n_{i,j} \cdot \Phi^{n-1}_{i,j} < 0)$
then $height_{i,j} = -[\Delta t(n - \frac{\Phi^n_{i,j}}{\Phi^n_{i,j} - \Phi^{n-1}_{i,j}})]\sqrt{1 + p_l^2 + q_l^2} + i\Delta x p_l + j\Delta y q_l.$

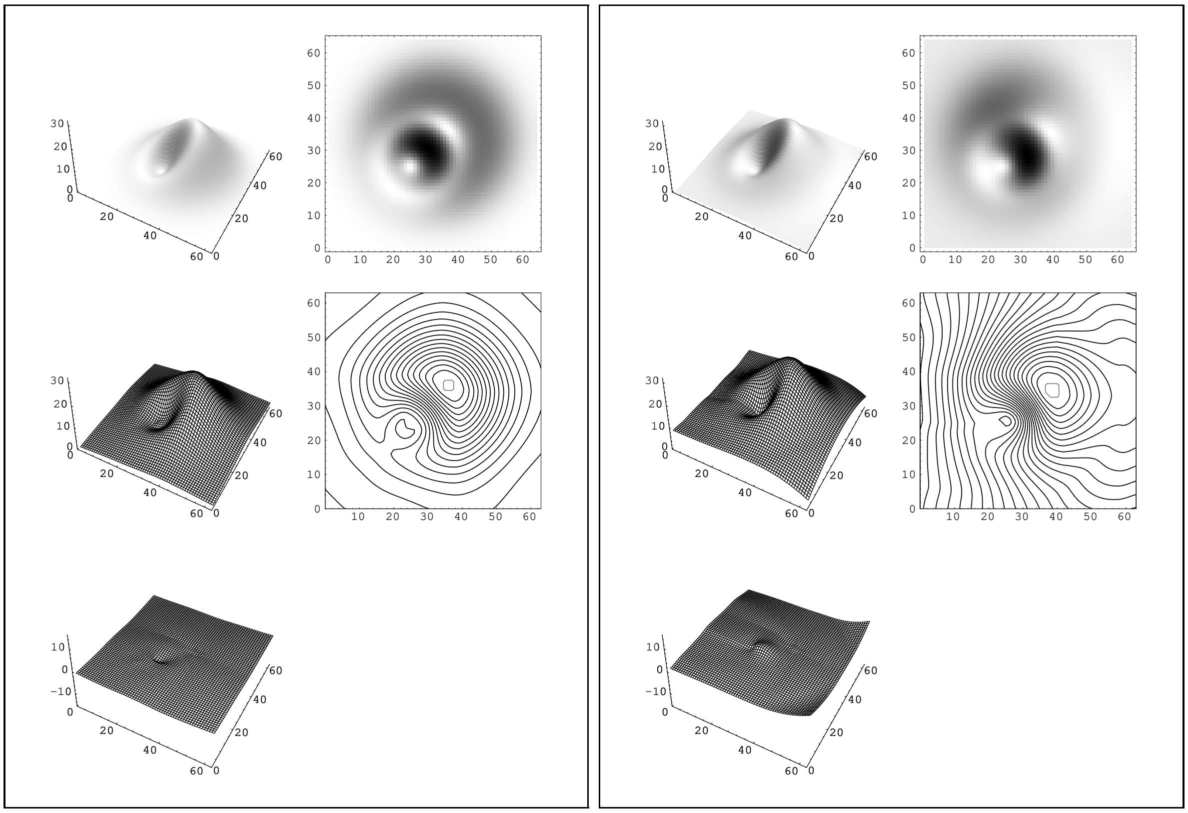
Using the above procedure each grid point gets its height at the "time" when the ϕ functions'–zero level passes through it.

4.5 Examples and Results

We demonstrate the performance of the proposed algorithm by applying it to several synthetic and real shaded images. The synthetic images were generated for surfaces assumed to be Lambertian, the size of these images being quite small (64×64 pixels). The real images were taken by a simple CCD camera and scaled to 128×128 pixels. The initialization is achieved by using gray–level thresholding to specify initial "singular areas".

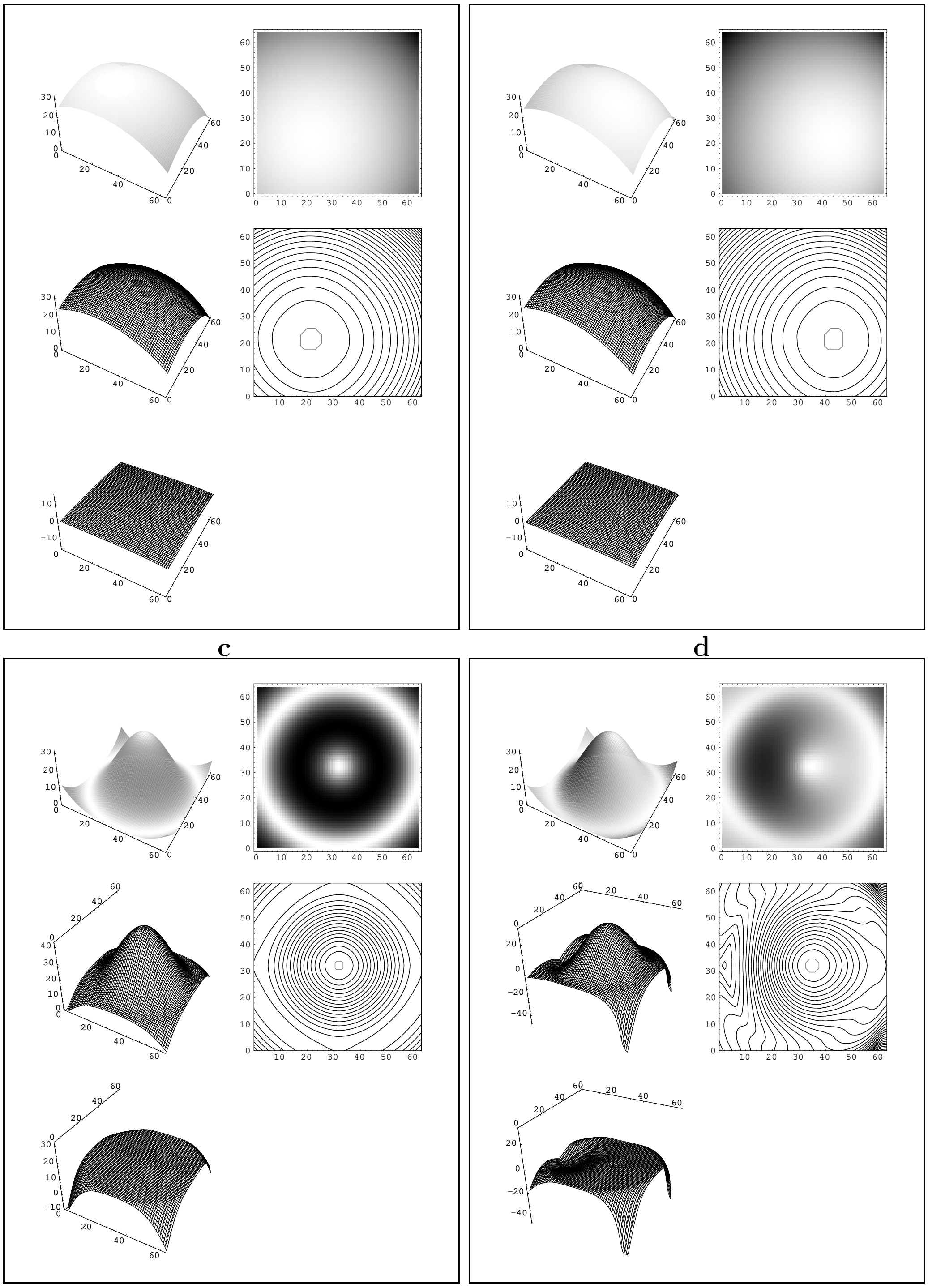
The reconstruction of the synthetic models is displayed in the following Figures where the upper right is the given shading image, the upper left surface is the original model, the middle left surface is the reconstructed surface, and the middle right is a description of the curve evolution starting from the gray level contour (dotted line). An error surface for each experiment, computed by subtracting the reconstructed surface from the original one, is displayed on the lower left. For each surface model two light direction were applied, the first is the vertical $\hat{l} = (0, 0, 1)$ (that is $\beta = 0^o$), and the second is a tilted light source direction of $l = (-0.364, 0, 1)$ (that is $\beta = 20^o$). The propagation is performed from the singular area outward, outside the singular region surrounded by the initial contour. This causes the singular areas to appear flat in the reconstructed surfaces.

Figures 4.5a and 4.5b show two examples of reconstruction for a “volcano” surface, starting from a small curve around the singular area at the top of the mountain, shown as a dotted line. In the equal height contours picture of the reconstructed surface one can observe the way topological problems like the saddle on the lowest left corner are inherently solved through this “down–going” process. Note however that starting at the base of the volcano would require special treatment to proceed beyond the saddle point, such issues are discussed in the next chapter [62, 66]. Figures 4.5c and 4.5d, demonstrate the reconstruction of part of a sphere surface. Figures 4.5e and 4.5f, show the topological mistake that happens when trying to reconstruct a “Mexican hat” like object. This topological error may be avoided if one chooses the singular strip surrounding the middle hill as a starting area, see Figure 4.5g. In this case there are two initial contours segmenting the singular selected area from the rest of the image, the first is propagating outwards and the second inwards yielding the desired reconstruction.



a **b**

Figure 4.5: Reconstruction of synthetic images given on 64×64 grid. **a.** The light source direction is from above ($\hat{l} = \hat{z}$). **b.** A tilted light source direction $l = (-0.364, 0, 1)$ ($\beta = 20^o$).



c d

e f

Figure 4.5 (Continue): **c.** The light source direction is from above ($\hat{l} = \hat{z}$). **d.** A tilted light source direction $l = (-0.364, 0, 1)$ ($\beta = 20^o$). **e.** The light source direction is from above ($\hat{l} = \hat{z}$). **f.** A tilted light source direction $l = (-0.364, 0, 1)$ ($\beta = 20^o$).