Chapter 6

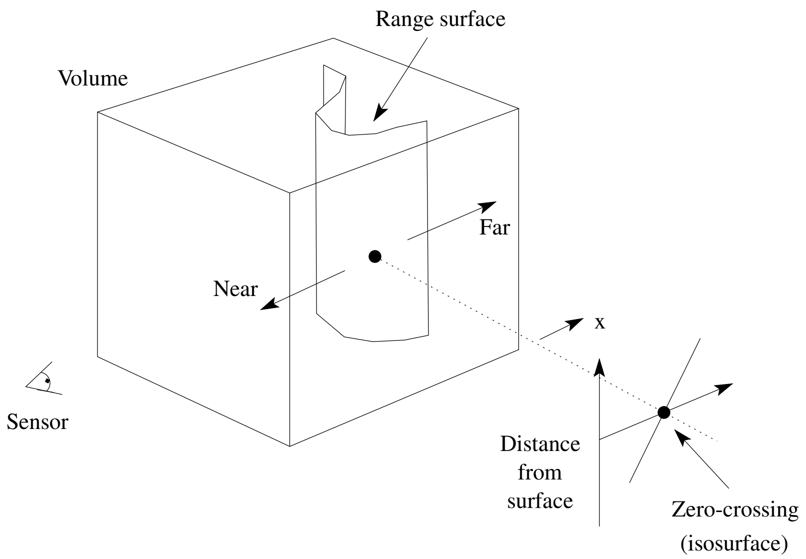
A New Volumetric Approach

The discussion in the previous chapter provides a mathematical framework for merging range images. In this chapter, we begin with a description of how we can implement this framework on a volumetric grid (Section 6.1). Then, we extend the framework to represent knowledge about the emptiness of space around an object (Section 6.2). This additional knowledge leads to a simple algorithm for filling gaps in the reconstruction where no surfaces have been observed. In Section 6.3, we analyze sampling and filtering issues that arise from working with a sampled representation. Finally, we discuss the limitations of the volumetric method described in this chapter and offer some possible solutions (Section 6.4).

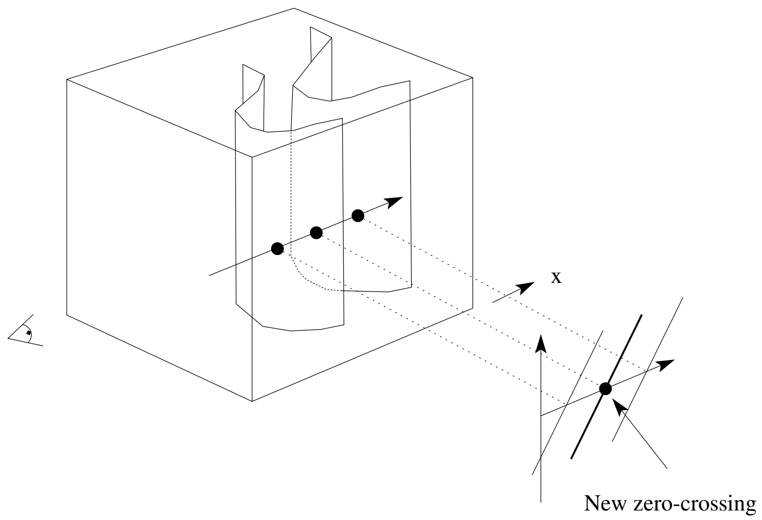
6.1 Merging observed surfaces

Our algorithm employs a continuous implicit function, $D(\mathbf{x})$, represented by samples. The function we represent is the weighted signed distance of each point $\mathbf{x}$ to the nearest range surface along the line of sight to the sensor. We construct this function by combining signed distance functions $d_1(\mathbf{x})$, $d_2(\mathbf{x})$, ... $d_p(\mathbf{x})$ and weight functions $w_1(\mathbf{x})$, $w_2(\mathbf{x})$, ... $w_p(\mathbf{x})$ obtained from range images 1 ... n. Our combining rules give us for each voxel a cumulative signed distance function, $D(\mathbf{x})$, and a cumulative weight $W(\mathbf{x})$. We represent these functions on a discrete voxel grid and extract an isosurface corresponding to $D(\mathbf{x}) = 0$. As shown in the previous chapter, this isosurface is optimal in the least squares sense.

Figure 6.1 illustrates the principle of combining unweighted signed distances for the



(a)



(b)

Figure 6.1: Unweighted signed distance functions in 3D. (a) A range sensor looking down the x-axis observes a range image, shown here as a reconstructed range surface. Following one line of sight down the x-axis, we can generate a signed distance function as shown. The zero crossing of this function is a point on the range surface. (b) The range sensor repeats the measurement, but noise in the range sensing process results in a slightly different range surface. In general, the second surface would interpenetrate the first, but we have shown it as an offset from the first surface for purposes of illustration. Following the same line of sight as before, we obtain another signed distance function. By summing these functions, we arrive at a cumulative function with a new zero crossing positioned midway between the original range measurements.

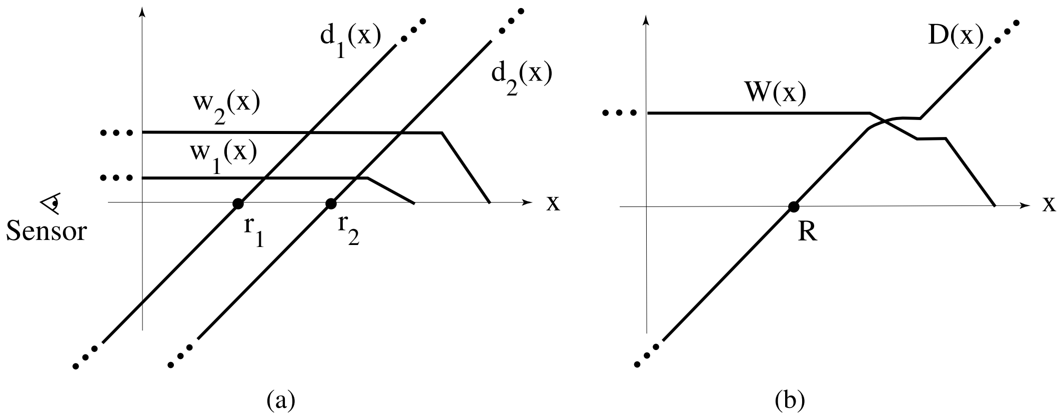


Figure 6.2: Signed distance and weight functions in one dimension. (a) The sensor looks down the x-axis and takes two measurements, r_1 and r_2. $d_1(x)$ and $d_2(x)$ are the signed distance profiles, and $w_1(x)$ and $w_2(x)$ are the weight functions. In 1D, we might expect two sensor measurements to have the same weight magnitudes, but we have shown them to be of different magnitude here to illustrate how the profiles combine in the general case. (b) $D(x)$ is a weighted combination of $d_1(x)$ and $d_2(x)$, and $W(x)$ is the sum of the weight functions. Given this formulation, the zero-crossing, R, becomes the weighted combination of r_1 and r_2 and represents our best guess of the location of the surface. In practice, we truncate the distance ramps and weights to the vicinity of the range points.

simple case of two range surfaces sampled from the same direction. Note that the resulting isosurface would be the surface created by averaging the two range surfaces along the sensor's lines of sight. In general, however, weights will vary across the range surfaces.

6.1.1 A one-dimensional Example

Figure 6.2 illustrates the construction and usage of the signed distance and weight functions in one dimension. In Figure 6.2a, the sensor is positioned at the origin looking down the +x axis and has taken two measurements, r_1 and r_2. The signed distance profiles, $d_1(x)$ and $d_2(x)$ may extend indefinitely in either direction, but the weight functions, $w_1(x)$ and $w_2(x)$, taper off behind the range points for reasons discussed below.

Figure 6.2b is the weighted combination of the two profiles. The combination rules are straightforward:

$$D(\mathbf{x}) = \frac{\Sigma w_i(\mathbf{x}) d_i(\mathbf{x})}{\Sigma w_i(\mathbf{x})} \tag{6.1}$$

$$W(\mathbf{x}) = \Sigma w_i(\mathbf{x}) \tag{6.2}$$

where, $d_i(\mathbf{x})$ and $w_i(\mathbf{x})$ are the signed distance and weight functions from the ith range image. Note that setting $D(\mathbf{x}) = 0$ and solving for $\mathbf{x}$ is equivalent to the least squares solution described in the previous chapter. Equation 6.1 differs from Equation 5.10 in that the former is normalized by the sum of the weights. Since the sum of the weights is always positive, the equations yield the same isosurface. In practice, we have observed that the normalization serves to equalize the gradient of the sampled field, $D(\mathbf{x})$, and thereby reduces artifacts in the isosurface algorithms, which are sensitive to sudden changes in the gradient. We discuss the benefits of using weighted averages in greater detail in Section 6.3.2.

Expressed as an incremental calculation, the signed distance and weight combination rules are:

$$D_{i+1}(\mathbf{x}) = \frac{W_i(\mathbf{x})D_i(\mathbf{x}) + w_{i+1}(\mathbf{x})d_{i+1}(\mathbf{x})}{W_i(\mathbf{x}) + w_{i+1}(\mathbf{x})} \tag{6.3}$$

$$W_{i+1}(\mathbf{x}) = W_i(\mathbf{x}) + w_{i+1}(\mathbf{x}) \tag{6.4}$$

where $D_i(\mathbf{x})$ and $W_i(\mathbf{x})$ are the cumulative signed distance and weight functions after integrating the ith range image.

In the special case of one dimension, the zero-crossing of the cumulative function is at a range, R given by:

$$R = \frac{\Sigma w_i r_i}{\Sigma w_i} \tag{6.5}$$

i.e., a weighted combination of the acquired range values, which is what one would expect for a least squares minimization.

6.1.2 Restriction to vicinity of surface

In principle, the distance and weighting functions should extend indefinitely in either direction. However, to prevent surfaces on opposite sides of the object from interfering with each other, we force the weighting function to taper off behind the surface. There is a trade-off involved in choosing where the weight function tapers off. It should persist far enough behind the surface to ensure that all distance ramps will contribute in the vicinity of the final zero crossing, but, it should also be as narrow as possible to avoid influencing surfaces on the other side. To meet these requirements, we force the weights to fall off at a distance equal to half the maximum uncertainty interval of the range measurements. Similarly, the signed distance and weight functions need not extend far in front of the surface. Restricting the functions to the vicinity of the surface also yields a more compact representation and reduces the computational expense of updating the volume. In Section 6.4.1, we discuss problems of interfering signed distance ramps in greater detail.

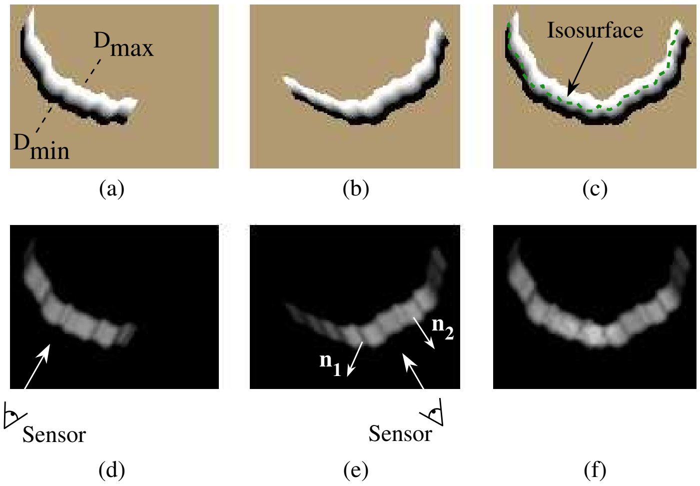


Figure 6.3: Combination of signed distance and weight functions in two dimensions. (a) and (d) are the signed distance and weight functions, respectively, generated for a range image viewed from the sensor line of sight shown in (d). The signed distance functions are chosen to vary between D_{min} and D_{max}, as shown in (a). The weighting falls off with increasing obliquity to the sensor and at the boundaries of the meshes as indicated by the darker regions in (e). The normals, $\mathbf{n_1}$ and $\mathbf{n_2}$ shown in (e), are oriented at a grazing angle and facing the sensor, respectively. Note how the weighting is lower (darker) for the grazing normal. (b) and (e) are the signed distance and weight functions for a range image of the same object taken at a 60 degree rotation. (c) is the signed distance function $D(\mathbf{x})$ corresponding to the per voxel weighted combination of (a) and (b) constructed using Equations 6.3 and 6.4. (f) is the sum of the weights at each voxel, $W(\mathbf{x})$. The dotted green curve in (c) is the isosurface that represents our current estimate of the shape of the object.

6.1.3 Two and three dimensions

In two and three dimensions, the range measurements correspond to curves or surfaces with weight functions, and the signed distance ramps have directions that are consistent with the primary directions of sensor uncertainty. Figure 6.3 illustrates the two-dimensional case for a range curve derived from a single scan containing a row of range samples. In practice, we use a fixed point representation for the signed distance function, which bounds the values to lie between D_{min} and D_{max} as shown in the figure. The values of D_{min} and D_{max} must be negative and positive, respectively, as they are on opposite sides of a signed distance zero-crossing.

For three dimensions, we can summarize the whole algorithm as shown in Figure 6.4. First, we set all voxel weights to zero, so that new data will overwrite the initial grid values.

```
  /* Initialization */
For each voxel {
   Set weight = 0
}

  /* Merging range images */
For each range image {

    /* Prepare range image */
   Tesselate range image
   Compute vertex weights

    /* Update voxels */
   For each voxel near the range surface {
        Find point on range surface
        Compute signed distance to point
        Interpolate weight from neighboring vertices
        Update the voxel's signed distance and weight
   }
}

  /* Surface extraction */
Extract an isosurface at the zero crossing
```

Figure 6.4: Pseudocode for the method of volumetric integration.

Next, we tessellate each range image by constructing triangles from nearest neighbors on the sampled lattice. We avoid tessellating over step discontinuities (cliffs in the range map) by discarding triangles with edge lengths that exceed a threshold. We must also compute a weight at each vertex as described below.

Once a range image has been converted to a triangle mesh with a weight at each vertex, we can update the voxel grid. The signed distance contribution is computed by casting a ray from the sensor through each voxel near the range surface and then intersecting it with the triangle mesh, as shown in figure 6.5. The weight is computed by linearly interpolating the weights stored at the intersection triangle's vertices. Having determined the signed distance and weight we can apply the update formulae described in Equations 6.3 and 6.4.

At any point during the merging of the range images, we can extract the zero-crossing isosurface from the volumetric grid. Isosurface extraction algorithms have been well-explored, and a number of approaches have been demonstrated to produce tessellations without consistency artifacts [Ning & Bloomenthal 1993]. These algorithms typically decompose the volume into cubes or tetrahedra with sample values stored at the vertices, followed by interpolation of these samples to estimate locations of zero-crossings, as shown in

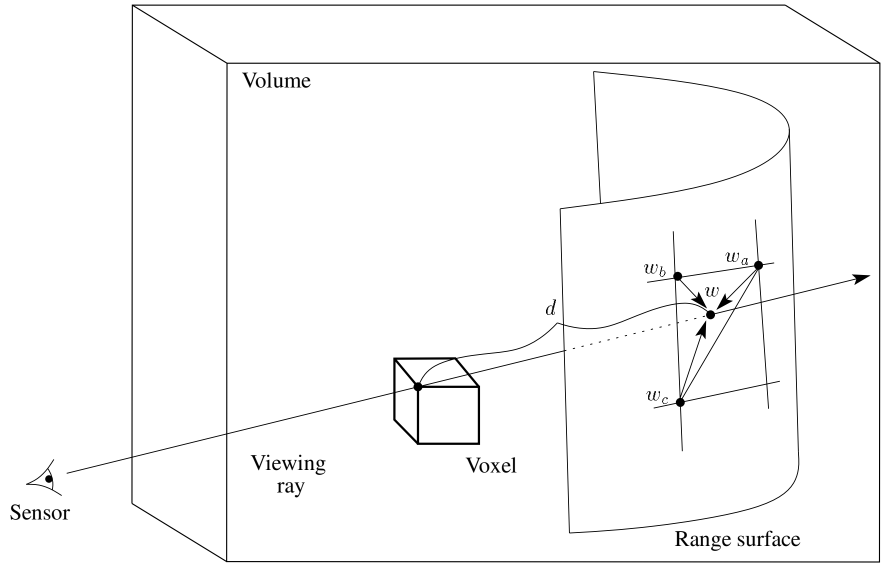


Figure 6.5: Sampling the range surface to update the volume. We compute the weight, w, and signed distance, d, needed to update the voxel by casting a ray from the sensor, through the voxel onto the range surface. We obtain the weight, w, by linearly interpolating the weights (w_a, w_b, and w_c) stored at neighboring range vertices. Note that for a translating sensor (like our Cyberware scanner), the sensor point is different for each column of range points.

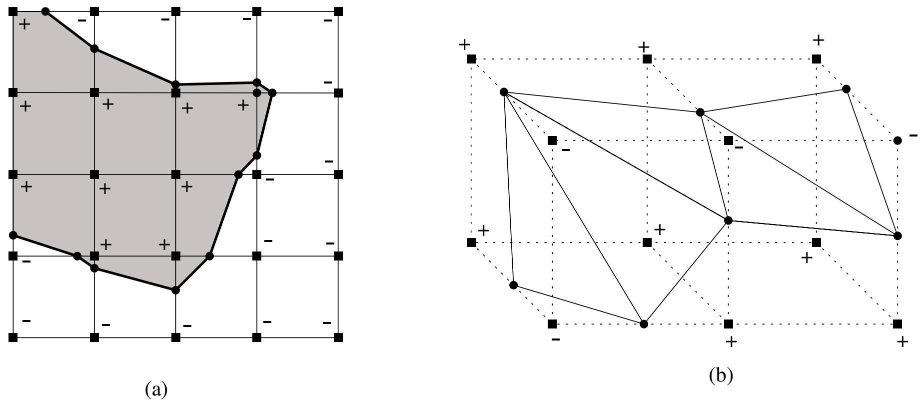


Figure 6.6: Discrete isosurface extraction. (a) In two dimensions, a typical isocontour extraction algorithm interpolates grid values to estimate isovalue crossings along lattice lines. By connecting the crossings with line segments, an isocontour is obtained. (b) In three dimensions, the crossings are connected with triangles. This method corresponds to the marching cubes algorithm described in [Lorensen & Cline 1987] with corrections in [Montani et al. 1994].

Figure 6.6. The gradients at the extracted zero-crossings provide an estimate of the surface normals[1]. We restrict the extraction procedure to skip samples with zero weight, generating triangles only in the regions of observed data. We will relax this restriction in the next section.

6.1.4 Choosing surface weights

The choice of weights is theoretically a function of the uncertainties in the range data, but they can also serve the purpose of manipulating the way range surfaces are merged. As described in the previous chapter, the use of weighted signed distances corresponds to the assumption of independent Gaussian statistics along the sensor's lines of sight. The weight then is the inverse of the squared variance of the pdf at each point on the surface. In this sense, the weight represents the degree of confidence in the data; a larger weight means that the data is more likely to be accurate.

The manner in which weights vary over a surface depend on the scanning technology. For example, in an optical triangulation system, low intensity images of the laser reflection imply a lower signal to noise ratio at the sensor, and thus lower confidence in shape estimates. [Soucy & Laurendeau 1995] and [Turk & Levoy 1994] have used this argument to justify forcing weights to depend on the dot product between the range surface normal and the laser direction, because the reflected radiance from a diffuse Lambertian surface varies with this dot product. When applying the spacetime analysis, the "goodness of fit" to the ideal spacetime Gaussian is also a confidence indicator. If the width of the Gaussian fit to the data deviates from the ideal width, then the data is accordingly downweighted, or even discarded. Additionally, at a step discontinuity, range data may be less reliable. The false edge extensions associated with traditional triangulation illustrate this point. Even in the case of spacetime analysis, the quality of edges is suspect due to filtering and resolution limitations.

[1] If the points $\mathbf{x_s}$ comprise the implicit surface defined by $F(\mathbf{x}) = const$, then the gradient of $F(\mathbf{x})$ at the surface, $\nabla F(\mathbf{x_s})$, corresponds exactly to the normals over the surface [Edwards & Penney 1982].

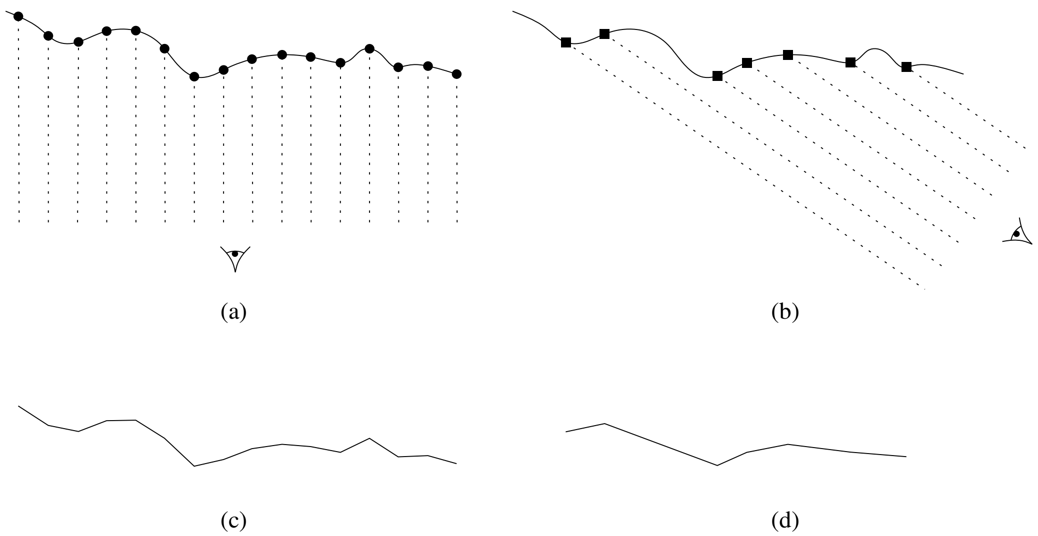


Figure 6.7: Dependence of surface sampling rate on view direction. When a surface is viewed "head on" as in (a), the sampling rate is on average significantly higher than when the surface is viewed from a grazing angle, as in (b). Note the greater detail in the estimated range surfaces, (c) versus (d).

The variations in weights can also be chosen for very practical considerations. If a surface is roughly planar with some moderate surface detail, then we would expect that a direct view of this surface in line with the predominant surface normal would give a higher quality range surface than a grazing angle view of the same surface. Figure 6.7 illustrates this idea. In general, separation between samples on the surface gives an indication of the comparative sampling rates among different range images. This sample separation is well approximated by $1/\cos\gamma$ where γ is the angle between the surface normal and the viewing direction. Thus, by setting the weights to be proportional to $\cos\gamma$ or even a higher power (e.g., $\cos^2\gamma$), the contributions from the range surfaces can be biased in favor of range views with higher surface sampling rates.

Reducing weights near surface boundaries serves a practical purpose as well: smooth blending of range surfaces. Consider the example in Figure 6.8. If the vertex weights are not tapered near the boundary of a range surface, then when it is merged with another range surface, there will be an abrupt jump from the average of two surfaces to a single surface. By tapering the vertex weights to zero in the vicinity of the boundary of a range surface, this surface will blend smoothly with other range surfaces.

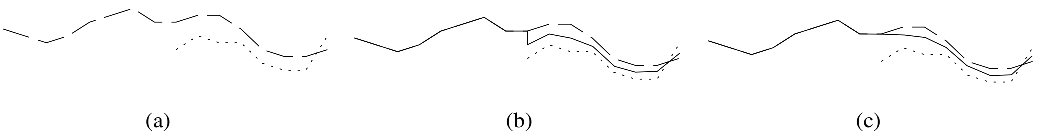


Figure 6.8: Tapering vertex weights for surface blending. (a) Two surfaces overlap, but one surface (shown dotted) has a boundary. (b) If the surface weights are not tapered near the boundary, then an abrupt transition appears when merging the surfaces, as indicated by the merged surface (solid line). (c) By tapering the surface weights near the boundary, the surfaces blend smoothly together.

6.2 Hole filling

The algorithm described in the previous section is designed to reconstruct the observed portions of the surface. Unseen portions of the surface will appear as holes in the reconstruction. While this result is an accurate representation of the known surface, the holes are esthetically unsatisfying and can present a stumbling block to follow-on algorithms that expect continuous meshes. In [Krishnamurthy & Levoy 1996], for example, the authors describe a method for parameterizing patches that entails generating evenly spaced grid lines by walking across the edges of a mesh. Gaps in the mesh prevent the algorithm from creating a fair parameterization. As another example, rapid prototyping technologies such as stereolithography typically require a "watertight" model in order to construct a solid replica [Dolenc 1993].

One option for filling holes is to operate on the reconstructed mesh. If the regions of the mesh near each hole are very nearly planar, then this approach works well. However, holes in the meshes can be (and frequently are) highly non-planar and may even require connections between unconnected components. Instead, we offer a hole filling approach that operates on our volume, which contains more information than the reconstructed mesh.

6.2.1 A hole-filling algorithm

The key to our algorithm lies in classifying all points in the volume as being in one of three states: unseen, empty, or near the surface. Holes in the surface are indicated by frontiers between unseen regions and empty regions (see Figure 6.9). Surfaces placed at these frontiers offer a plausible way to plug these holes (dotted in Figure 6.9). Obtaining this classification and generating these hole fillers leads to a straightforward extension of the algorithm

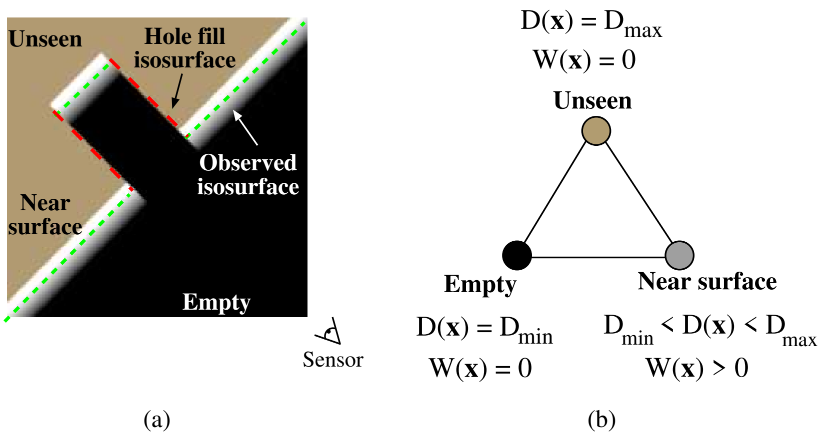


Figure 6.9: Volumetric grid with space carving and hole filling. (a) The regions in front of the surface are seen as empty, regions in the vicinity of the surface ramp through the zero-crossing, while regions behind remain unseen. The green (dashed) segments are the isosurfaces generated near the observed surface, while the red (dotted) segments are hole fillers, generated by tessellating over the transition from empty to unseen. In (b), we identify the three extremal voxel states with their corresponding function values.

described in the previous section:

1. Initialize the voxel space to the "unseen" state.
2. Update the voxels near the surface as described in the previous section. As before, these voxels take on continuous signed distance and weight values.
3. Follow the lines of sight back from the observed surface and mark the corresponding voxels as "empty". We refer to this step as *space carving*.
4. Perform an isosurface extraction at the zero-crossing of the signed distance function. Additionally, extract a surface between regions seen to be empty and regions that remain unseen.

In practice, we represent the unseen and empty states using the function and weight fields stored on the voxel lattice. We represent the unseen state with the function values $D(\mathrm{x}) = D_{max}$, $W(\mathrm{x}) = 0$ and the empty state with the function values $D(\mathrm{x}) = D_{min}$, $W(\mathrm{x}) = 0$, as shown in Figure 6.9b. The key advantage of this representation is that we can use the same isosurface extraction algorithm we used in the previous section this time lifting the prohibition on interpolating voxels of zero weight. This extraction finds both the signed distance and hole fill isosurfaces and connects them naturally where they meet, i.e., at the corners in Figure 6.9a where the dotted red line meets the dashed green line. Note that the triangles that arise from interpolations across voxels of zero weight are distinct from the others: they are hole fillers. We take advantage of this distinction when smoothing surfaces as described below.

Figure 6.9 illustrates the method for a single range image, and provides a diagram for the three-state classification scheme. Figure 6.10 illustrates how the hole filler surfaces connect to the observed surfaces. The hole filler isosurfaces are "false" in that they are not representative of the observed surface, but they do derive from observed data. In particular, they correspond to a boundary that confines where the surface could plausibly exist. Thus, the combination of the observed surfaces and the hole fill surfaces represents the object of maximum volume which is consistent with all of the observations. In practice, we find that many of the hole fill surfaces are generated in crevices that are hard for the sensor to reach.

6.2.2 Carving from backdrops

We have just seen how "space carving" is a useful operation: it tells us much about the structure of free space, allowing us to fill holes in an intelligent way. However, our algorithm only carves back from observed surfaces. There are numerous situations where more carving would be useful. For example, the interior walls of a hollow cylinder may elude digitization, but by seeing through the hollow portion of the cylinder to a surface placed behind it, we can better approximate its geometry. We can extend the carving paradigm to cover these situations by placing such a backdrop behind the surfaces being scanned. By placing the backdrop outside of the voxel grid, we utilize it purely for carving space without introducing its geometry into the model. Figure 6.11 illustrates the utility of space-carving with backdrops.

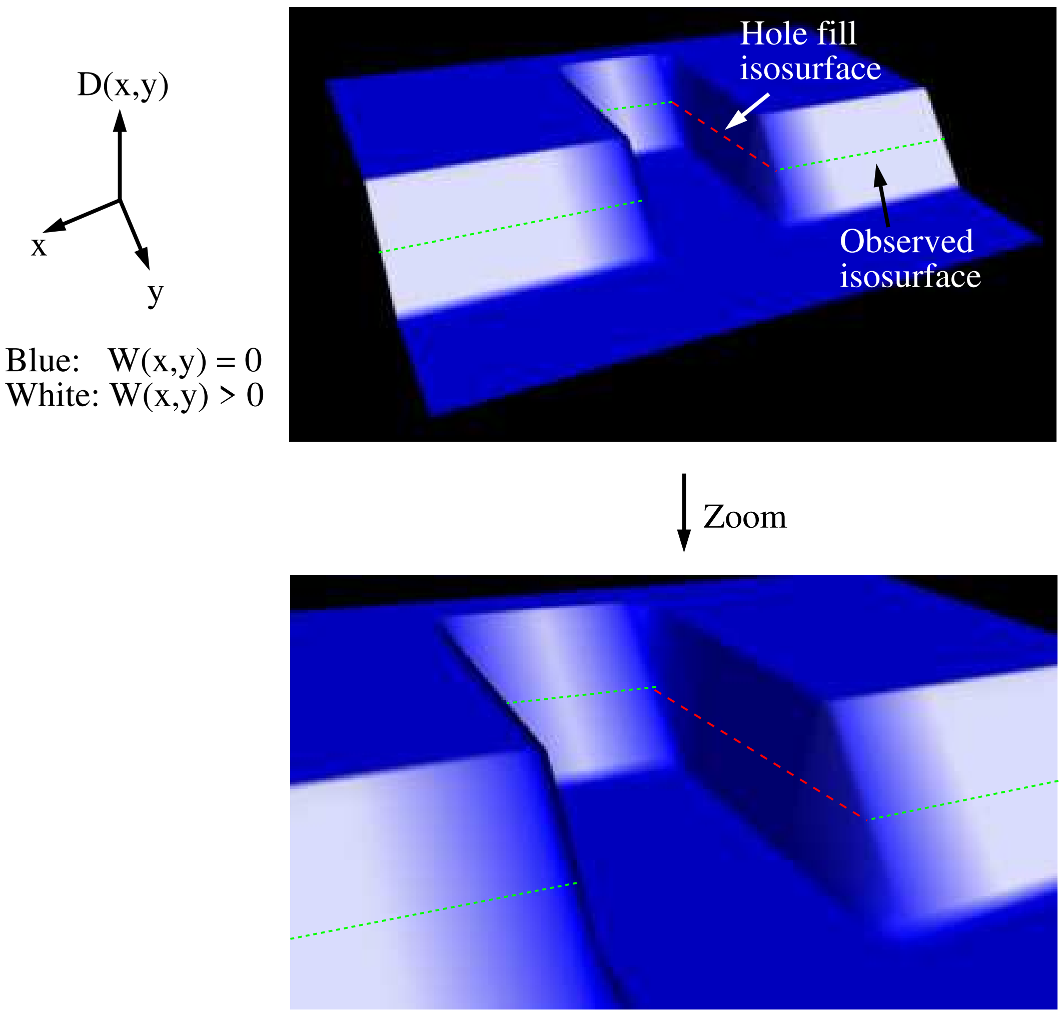


Figure 6.10: A hole-filling visualization. The images represent a slice through the volumetric grid depicted in Figure 6.9. The height of the ramps correspond to the signed distance field, $D(x, y)$, and the shading corresponds to the weight, $W(x, y)$. The isosurface is extracted at the level shown, where the dotted green lines correspond to observed surfaces and the dashed red lines correspond to hole-fill surfaces. Note how the observed and hole-fill surfaces connect seamlessly, with only a change in weights (going to zero at the hole-fill regions) to indicate the transitions.

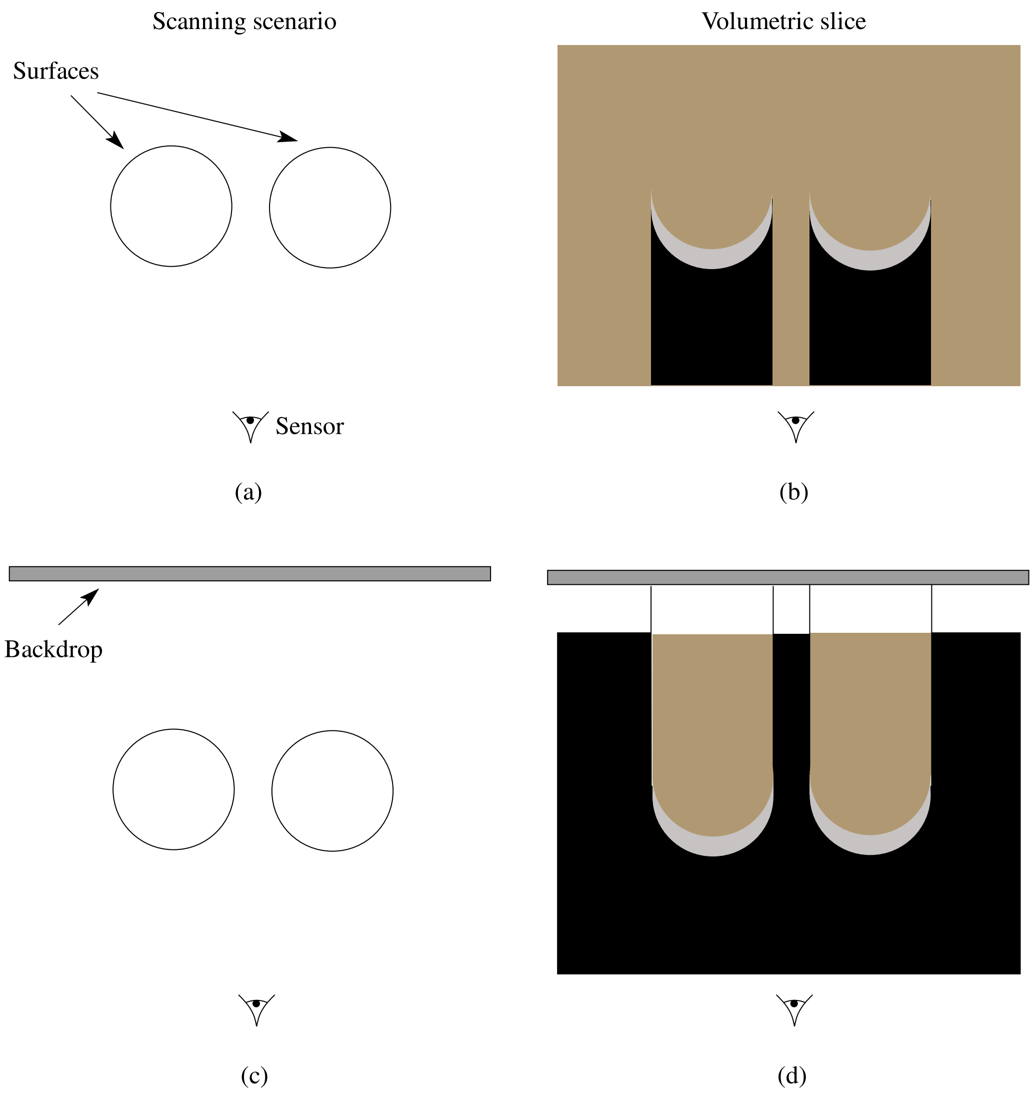


Figure 6.11: Carving from backdrops. (a) An orthographic sensor captures a range image of two cylinders. (b) A slice through the volumetric grid after laying down signed distance ramps near the observed surface and carving out the empty regions in front of the surfaces. It will be difficult to scan the portions of the cylinders that are close together. (c) The same scene as in (a) with the addition of a backdrop. (b) The volumetric slice after scanning the cylinders with a backdrop placed behind them. Much more of the space is carved as empty, and the cylinders are clearly segregated by the empty region between them.

6.3 Sampling, conditioning, and filtering

Operating in a sampled domain requires attention to several details. In this section, we begin by developing criteria for choosing the voxel resolution. Next, we show how the gradient variations can introduce artifacts when extracting an isosurface from a sampled volume, and we develop the notion of a conditioning function to address the problem. Finally, we describe methods for coping with the aliasing artifacts that arise when extracting hole fill surfaces.

6.3.1 Voxel resolution and tessellation criteria

Choosing the appropriate voxel resolution is crucial to avoiding aliasing artifacts in the surface reconstruction. Intuitively, the voxel resolution should be sufficient to sample the distance ramp and should be chosen to match the level of detail of the range surface.

To derive a criterion for the voxel resolution, we examine the problem from a signal processing point of view. Combining multiple range surfaces with weighted averages leads to a complicated analysis that does not offer many insights. Accordingly, we restrict the analysis to the case of determining the volumetric sampling rate required to represent a single range surface as a signed distance function. We will consider the role of weights in the context of isosurfaces in the next section.

In order to estimate the required voxel resolution, we must determine the bandwidth of the signed distance representation for a range surface. We define the symbols s_x, s_y, and s_z to correspond to the spatial frequencies in x, y, and z. The radial spatial frequency is then:

$$\rho_{s_x s_y s_z} = \sqrt{s_x^2 + s_y^2 + s_z^2}$$

and we hope to derive an expression for:

$$\rho_{s_x s_y s_z}^{max} \leq \sqrt{(s_x^{max})^2 + (s_y^{max})^2 + (s_z^{max})^2}$$

where $\rho_{s_x s_y s_z}^{max}$ is the radial bandlimit.

Consider the case of an orthographic sensor looking down the z-axis having acquired a range surface, $f(x, y)$. For this single scan, the volume will store:

$$D(x, y, z) = d(z - f(x, y))$$

where $d(z)$ is the signed distance function centered at $z = 0$. Taking the Fourier transform:

$$\mathcal{F}\{D\} = \iiint d(z - f(x, y)) \exp[-i2\pi(s_x x + s_y y + s_z z)] dx dx dz$$

Regrouping the terms, we obtain:

$$\begin{aligned}
\mathcal{F}\{D\} &= \iint \exp[-i2\pi(s_x x + s_y y)] \left\{ \int d(z - f(x, y)) \exp[-i2\pi s_z z] dz \right\} dx dy \\
&= \iint \exp[-i2\pi(s_x x + s_y y)] \mathcal{F}_z\{d(z)\} \exp[-i2\pi s_z f(x, y)] dx dy \\
&= \mathcal{F}_z\{d(z)\} \mathcal{F}_{xy}\{\exp[-i2\pi s_z f(x, y)]\}
\end{aligned}$$

This equation reveals that the Fourier transform of the signed distance is partially separable; i.e., a product of the Fourier transform in z and the transform in xy. The z transform pertains to a known function, the signed distance ramp, and is thus easily computed. The xy transform has a significantly more complicated argument that includes the z component of frequency, s_z. Ultimately, we would like to derive the bandwidth of $\mathcal{F}\{D\}$ to help decide the necessary sampling rate for accurate shape capture without aliasing. To do so, we need to relate some property of the function $f(x, y)$ to the frequency content of the the xy transform.

One revealing measure that corresponds to bandwidth is the 2D variance of the squared magnitude of the $s_x s_y$ spectrum of a function $p(x, y)$ [Bracewell 1986],

$$(\Delta\rho_{s_x s_y})^2 \equiv \frac{\iint (s_x^2 + s_y^2) |P(s_x, s_y)|^2 ds_x ds_y}{\iint |P(s_x, s_y)|^2 ds_x ds_y} \tag{6.6}$$

where $P(s_x, s_y)$ is the Fourier transform of $p(x, y)$, and we have assumed the first central moment to be zero. A more convenient definition of this variance arises after the application of several relations. The first is Rayleigh's theorem:

$$\iint |G(s_x, s_y)|^2 ds_x ds_y = \iint |g(x, y)|^2 dx dy \tag{6.7}$$

where $G(s_x, s_y)$ is the Fourier transform of a function $g(x, y)$. The second two relations are the rules of Fourier transforms and partial differentiation:

$$s_x G(s_x, s_y) = \frac{-i}{2\pi}\frac{\partial}{\partial x} g(x, y) \tag{6.8}$$

$$s_y G(s_x, s_y) = \frac{-i}{2\pi}\frac{\partial}{\partial y} g(x, y) \tag{6.9}$$

$$\tag{6.10}$$

After re-writing Equation 6.6 as

$$(\Delta\rho_{s_x s_y})^2 = \frac{\iint |s_x P(s_x, s_y)|^2 ds_x ds_y + \iint |s_y P(s_x, s_y)|^2 ds_x ds_y}{\iint |P(s_x, s_y)|^2 ds_x ds_y} \tag{6.11}$$

then we can combine Equations 6.7-6.11 to obtain the following relation:

$$(\Delta\rho_{s_x s_y})^2 = \frac{1}{4\pi^2} \frac{\iint \left|\frac{\partial}{\partial x} p(x, y)\right|^2 + \left|\frac{\partial}{\partial y} p(x, y)\right|^2 dx dy}{\iint |p(x, y)|^2 dx dy}$$

Now we can make the substitution:

$$p(x, y) = \exp[-i2\pi s_z f(x, y)]$$

which leads to the following expression:

$$(\Delta\rho_{s_x s_y})^2 = \frac{s_z^2 \iint_A \left\{ \left| \frac{\partial}{\partial x} f(x,y) \right|^2 + \left| \frac{\partial}{\partial y} f(x,y) \right|^2 \right\} dxdy}{\iint_A dxdy} \tag{6.12}$$

where A is the area over which the range surface is defined. At the bandlimit, s_z obtains its maximum value s_z^{max} as set by the signed distance function. We can approximate the square of the overall bandwidth to be:

$$(\rho_{s_x s_y s_z}^{max})^2 \approx (s_z^{max})^2 + (\Delta\rho_{s_x s_y})^2 \tag{6.13}$$

After noting that $\iint_A dxdy = A$, we substitute Equation 6.12 into Equation 6.13 to obtain:

$$(\rho_{s_x s_y s_z}^{max})^2 \approx \frac{(s_z^{max})^2}{A} \iint_A \left\{ 1 + \left| \frac{\partial}{\partial x} f(x,y) \right|^2 + \left| \frac{\partial}{\partial y} f(x,y) \right|^2 \right\} dxdy \tag{6.14}$$

We can make this relation easier to understand if we define:

$$\langle g(x,y) \rangle \equiv \frac{1}{A} \iint_A g(x,y) dxdy \tag{6.15}$$

to be the area average of $g(x,y)$, and we observe that the normal to the surface in the z direction is (see Section 5.5):

$$n_z(x,y) = \frac{-1}{\sqrt{1 + \left| \frac{\partial}{\partial x} f(x,y) \right|^2 + \left| \frac{\partial}{\partial y} f(x,y) \right|^2}} \tag{6.16}$$

Combining Equations 6.14-6.16, we arrive at the following relation for the bandwidth of the surface convolved with the signed distance function:

$$\rho_{s_x s_y s_z}^{max} = s_z^{max} \sqrt{\left\langle \frac{1}{|n_z(x,y)|^2} \right\rangle}$$

In other words, the bandwidth is set by the average square value of the reciprocal of the component of the surface normal along the viewing direction. It is therefore very sensitive to surfaces at grazing angles to the sensor; i.e., as n_z approaches zero, the bandwidth becomes very large, requiring high sampling rates.

We can derive a simple rule of thumb for the sampling rate if we consider a single planar surface. In this case, the bandwidth expression simplifies to:

$$\rho^{max}_{s_x s_y s_z} \approx \frac{s^{max}_z}{n_z}$$

To make this relation useful, we need to know the value of s^{max}_z. We can model the signed distance function, clipped to have a width of twice the maximum uncertainty interval with the function:

$$d(z) = z \cdot \mathrm{rect}\left(\frac{z}{b}\right)$$

where b is width of the distance ramp. The Fourier transform of this function is:

$$\mathcal{F}\{d\} = \frac{ib}{2\pi}\frac{d(\mathrm{sinc}(bs_z))}{ds_z}$$

where $\mathrm{sinc}(x) = \sin(\pi x)/\pi x$. We can define the first zero crossing of this function to be the bandwidth of the signed distance function. This zero crossing occurs at $s^{max}_z = 1/b$, leading to the relation:

$$\rho^{max}_{s_x s_y s_z} \approx \frac{1}{bn_z}$$

Thus, the spacing between voxels (Δ) should be small enough that the sampling frequency ($1/\Delta$) exceeds the Nyquist sampling rate:

$$\Delta \leq \frac{bn_z}{2} \tag{6.17}$$

This relationship provides several insights. First, the voxel spacing must be small enough to sample the distance ramp adequately, as indicated by the proportionality to the ramp width b. Less noisy range data requires a narrower distance ramp and a smaller voxel spacing. In addition, the dependence on n_z is closely related to range image tessellation. When a range image is converted to a range surface, a tessellation criterion determines whether neighboring range samples should be connected. This criterion typically takes the form of a minimum threshold normal component in the direction of the sensor or a maximum permissible edge length. The edge length criterion is very similar to the normal component criterion. Thus, the range surface tessellation criterion enforces a bound on the maximum voxel spacing.

Equation 6.17 can also be seen as a guideline for modifying the distance ramps and tessellation criteria to satisfy a desired sampling rate. In other words, the distance ramp can be seen as a bandlimiting filter; widening the ramp decreases the required sampling rate. Applying a more restrictive tessellation criterion has a similar effect.

While Equation 6.17 is based on the unweighted signed distance for a single range image, our experience has shown that it serves as an excellent guide for avoiding aliasing artifacts even when using weighted averages of signed distances for multiple range images.

6.3.2 Conditioning the implicit function

The algorithm described in this chapter uses a weighted *average* of the signed distances, yet in the previous chapter we showed that a simple weighted *sum* would suffice. In this section, we show why dividing by the sum of the weights is advantageous.

As shown in the previous chapter, the governing equation for finding a least squares solution is:

$$\sum_{i=1}^{p} |\mathbf{v_i}(x,y,z)| w_i(x,y,z) d_i(x,y,z) = 0$$

Without altering the position of the least squares isosurface, we can multiply this equation by a strictly positive conditioning function, $\kappa(x,y,z)$:

$$\kappa(x,y,z) \sum_{i=1}^{p} |\mathbf{v_i}(x,y,z)| w_i(x,y,z) d_i(x,y,z) = 0$$

One such conditioning function is the reciprocal of the sum of the weights:

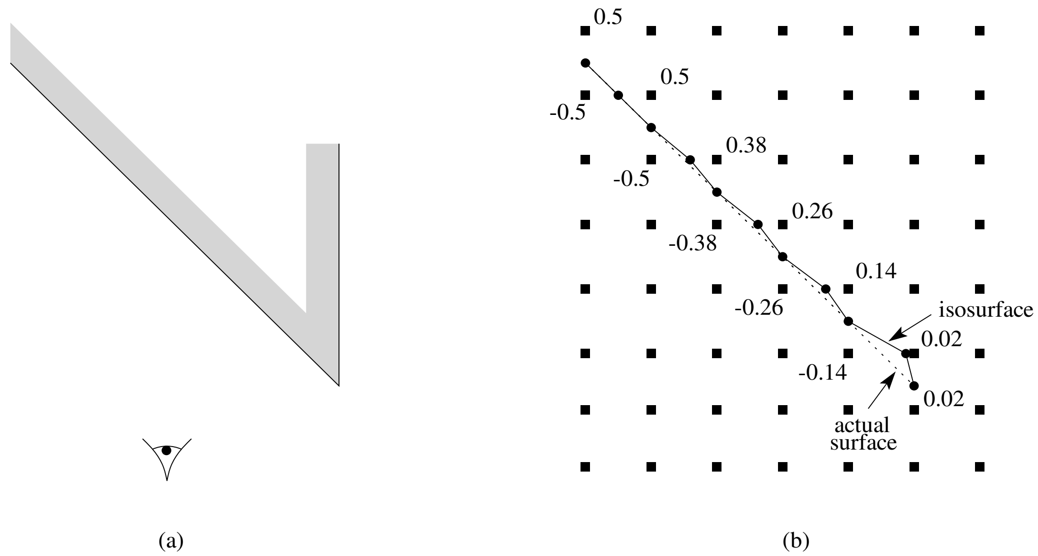


Figure 6.12: Tessellation artifacts near a planar edge. (a) A range sensor scans a planar surface up to an edge. (b) The product of the signed distance and the weights are laid down into the volume. The weights taper near an edge causing the value-weight product to shrink. The square markers are voxels, the dotted line is the actual surface, and the solid line is the extracted isosurface. Note the tessellation artifacts that result from the variations in weight.

$$\kappa(x, y, z) = \frac{1}{\Sigma w_i(x, y, z)} \tag{6.18}$$

which corresponds to computing a weighted average of the signed distances at each voxel.

But what is the utility of a conditioning function? To answer this question, we consider the impact of tapering weights near mesh boundaries. Figure 6.12 depicts a planar surface with a boundary edge after being merged into the volumetric grid. Tapering the weights near the boundary is intended to make for a smoother blend when merging overlapping surfaces, but the decreasing weights, when multiplied by the signed distance, alter the gradient of the implicit function. Isosurface extraction methods such as the Marching Cubes algorithm are very sensitive to variations in the gradient, and the resulting isosurface has artifacts as depicted in the figure. On the other hand, after dividing by the sum of the weights (for a single scan, this amounts to unweighted signed distance), the gradient is constant and no artifacts appear.

Figure 6.13 shows the effect of using sums of weighted distances versus the use of weighted averages of signed distances when merging range images. Clearly, the weighted average conditioning smooths the gradient and avoids artifacts associated with the unconditioned isosurface.

The examples in Figure 6.13 show how the sum of weights conditioning function (Equation 6.18) can help reduce variations in the gradient to produce better isosurfaces. To see how we might design different conditioning functions, we can derive an expression for the gradient magnitude of the conditioned signed distance function:

$$\begin{aligned} |\nabla D| &= \left|\nabla\left[\kappa\sum_{i=1}^{p}|\mathbf{v_i}|w_i d_i\right]\right| \\ &= \left|\nabla\kappa\sum_{i=1}^{n}|\mathbf{v_i}|w_i d_i + \kappa\nabla\sum_{i=1}^{p}|\mathbf{v_i}|w_i d_i\right| \end{aligned}$$

Since we are only interested in the gradient in the immediate vicinity of the isosurface, the first term on the right hand side of the last equation vanishes to give:

$$\begin{aligned} |\nabla D|_{iso} &= \left|\kappa\nabla\sum_{i=1}^{p}|\mathbf{v_i}|w_i d_i\right| \\ &= \kappa\left|\sum_{i=1}^{p}\nabla\left(|\mathbf{v_i}|w_i d_i\right)\right| \end{aligned}$$

where $|\nabla D|_{iso}$ is the gradient magnitude at the isosurface. Thus, we can ensure a uniform gradient magnitude in the vicinity of the isosurface by setting the conditioning function to be:

$$\kappa = \frac{1}{\left|\sum_{i=1}^{p}\nabla\left(|\mathbf{v_i}|w_i d_i\right)\right|}$$

This observation leads to a uniform gradient algorithm. At each voxel we would store a scalar and a *vector*. The scalar would be the cumulative weighted signed distance, and the

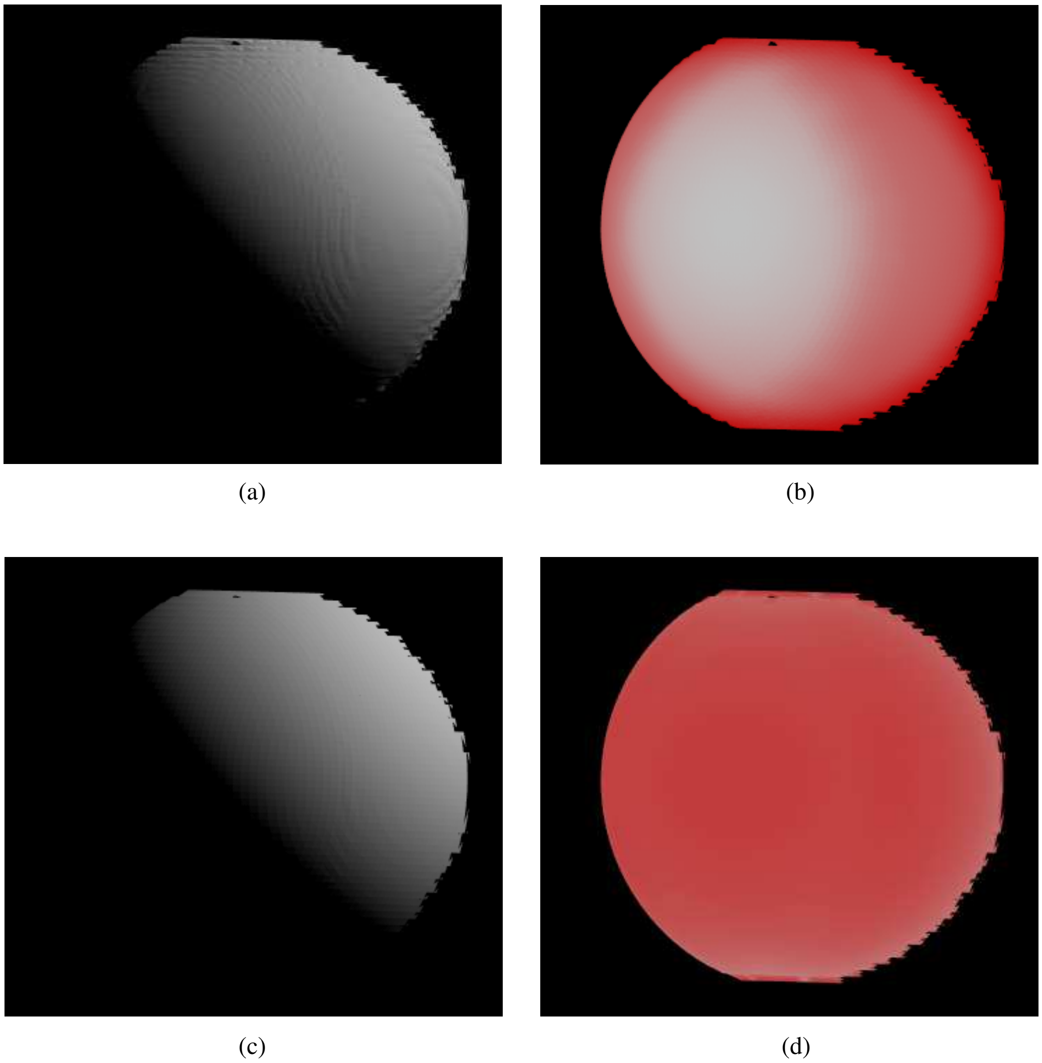


Figure 6.13: Isosurface artifacts and conditioning functions. Noiseless synthetic range images were constructed from a perfect sphere scanned orthographically from two directions spaced 45 degrees apart. The scans were merged using two techniques, one with and one without a conditioning function. (a) and (c) are faceted renderings of the resulting isosurfaces, lit to emphasize artifacts. (b) and (d) are the gradient magnitudes across the isosurface. Red areas have small gradients while white areas have large gradients. In (a) and (b), no conditioning function is used, and “ripples” appear in the rendering (a) due to variations in the gradient evident in (b). In (c) and (d), a weighted average conditioning function is used, and the result is a smooth surface (c) with little variation in the gradient (b). The hole at the tops and the jagged boundaries of the reconstructions are artifacts of the marching cubes algorithm operating on the discrete voxel grid used in this illustration.

vector would be the sum of the gradients computed at each voxel for each scan. After merging a set of scans, we would divide the scalar by the magnitude of the vector at each voxel to yield a volume with gradients normalized in the vicinity of the isosurface. While we have yet to implement such an algorithm, it holds promise for minimizing isosurface artifacts due to gradient magnitude variations.

6.3.3 Mesh filtering vs. anti-aliasing in hole fill regions

Artifacts in the isosurface also arise when applying the method of hole filling by carving space. Because the transition between unseen and empty is discontinuous and hole fill triangles are generated as an isosurface between these binary states, with no smooth transition, we generally observe aliasing artifacts in these areas. These artifacts can be eliminated by pre-filtering the transition region before sampling on the voxel lattice using straightforward methods such as analytic filtering or super-sampling and averaging down. In practice, we have obtained satisfactory results by applying another technique: post-filtering the mesh after reconstruction using weighted averages of nearest vertex neighbors as described in [Taubin 1995]. The effect of this filtering step is to fair the hole fill surface. Since we know which triangles correspond to hole fillers, we need only concentrate the surface filtering on these portions of the mesh. This localized filtering preserves the detail in the observed surface reconstruction. To achieve a smooth blend between filtered hole fill vertices and the neighboring "real" surface, we allow the filter weights to extend beyond and taper off into the vicinity of the hole fill boundaries.

6.4 Limitations of the volumetric method

The volumetric method described in this section has several limitations. The first two concern the ability to determine accurate shape from observed surfaces, while the third is a limitation inherent in the process of space carving.

(a) (b) (c)

Figure 6.14: Limitation on thin surfaces. (a) A surface is scanned from two sides. (b) The volumetric grid shows that the distance ramps overlap as the opposing surfaces come close together. (c) The resulting isosurface shows that the interference of signed distance functions results in a thicker surface.

6.4.1 Thin surfaces

As described earlier in this chapter, the width of each signed distance ramp is made as small as possible while ensuring overlap with ramps taken from the same side of a surface. However, when opposing surfaces come within a ramp width of each other, then the signed distances taken from opposite sides of the surface begin to interfere. Figure 6.14 illustrates this point. In effect, this limitation imposes a minimum thickness of a feature that can be seen from two sides. This is also problematic for sharp corners. At the tip of a corner, the adjoining faces come arbitrarily close together, and opposing scans (i.e., scans that do not see both sides of the corner) will necessarily interfere with one another in the volume, as show in Figure 6.16a-c.

The fundamental problem is the inability of the volumetric representation to distinguish opposing surfaces before combining their distance functions. We may address this problem in several ways. For instance, we could begin by storing estimated surface normals for voxels in the neighborhood of the surface. These normals might arise by blending together estimated normals from individual range scans, though the noisiness of individual normal estimates will affect results. Alternatively, the normals may be derived from the initial reconstruction following a recipe such as: (1) reconstruct the surface, (2) reinitialize the volume, (3) deposit the surface normal of the nearest point on the surface into each

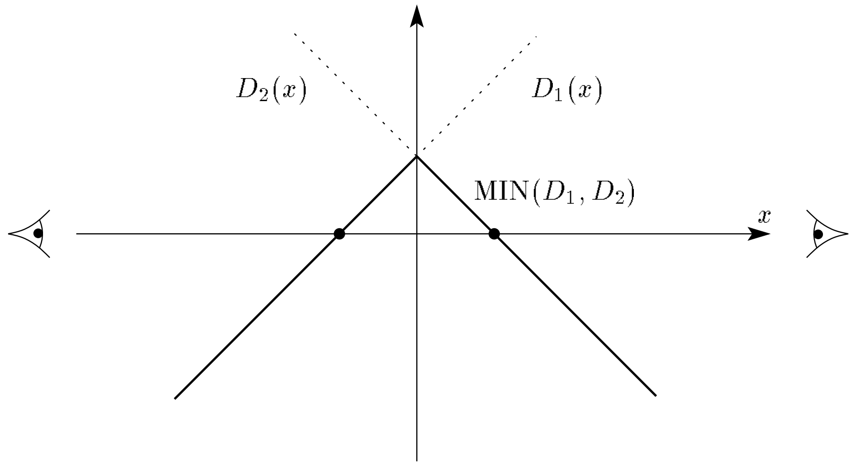


Figure 6.15: Using the $\mathrm{MIN}()$ function to merge opposing distance ramps. Cumulative signed distance functions can be accumulated for differing orientations, as in the case of opposing surfaces. These distance functions are shown here in one dimension. Using the $\mathrm{MIN}()$ function to merge these functions before extracting the isosurface will preserve the desired zero crossings.

voxel within the distance ramp of the surface. This second approach assumes that errors associated with thin surfaces are fairly isotropic; i.e., the erroneous thickening of the surface is roughly a simple dilation of the surface in the areas affected. Once approximate normals are established, then we can modify the method for merging range surfaces into the volume. A simple approach would restrict a range surface to influence only voxels whose normals are within some threshold of what could possibly be visible to the sensor.

These normal techniques can be extended to store more than one estimated surface normal and distance function per voxel. For example, if a voxel is between opposing surfaces, then the algorithm could store the estimated orientations for both of the opposing surfaces and update separate signed distances for range estimates coming from opposite sides. In general, storage requirements for this method would vary with the geometry of the object: in particularly complex regions, storage of multiple estimated normals may be necessary, whereas smoother regions without opposing surfaces would require only a single normal and signed distance. To reconstruct the final range surface, it might be possible to follow the contours of the two signed distances separately, and then merge the resulting geometry. Alternatively, these distances could be blended together using a $\mathrm{MIN}()$ function before reconstruction as indicated in Figure 6.15.

(a) (b) (c) (d)

Figure 6.16: Limitations due to sharp corners. (a) A corner is scanned from two sides. (b) The volumetric grid shows the distance ramps overlapping opposite sides near the apex. (c) The isosurface shows a thickening of the corner and a gap at the apex. (d) Samples on the volumetric grid show that even ideal range data, merged into the volume in a manner avoiding surface thickening will lead to a gap in the reconstruction.

Other potential strategies include modifications to the weight functions. For example, if the weight functions begin to taper off along the sensor's line of sight for voxels behind the range surface, then the interference with the opposing surface will be reduced. However, when combining surface estimates on the same side of the surface, the tapered weights will introduce some bias into the reconstruction. Such an approach would affect the accuracy of the whole reconstruction, but selective application of this strategy might still be of use. By adjusting the weight profiles for range samples near boundaries, it is possible to reduce the surface thickening selectively in at least one of the more troublesome areas, i.e., near corners. Early experiments in this area have yielded promising results.

6.4.2 Bridging sharp corners

Another limitation of the volumetric method is the difficulty in bridging sharp corners, a problem which persists even if interference of opposing signed distances are eliminated. Consider the scenario in Figure 6.16. Two scans of a sharp corner acquire accurate descriptions of the corner up to the apex. However, when merging these scans into the volumetric grid, the restricted marching cubes tessellator cannot "turn the corner" and connect the two surfaces.

This difficulty is avoidable under two circumstances. If an additional scan observes both sides of the corner, then the corner is bridged. Alternatively, if the scans of the corner are observed to be occluding other geometry such as a backdrop, then the space carving step

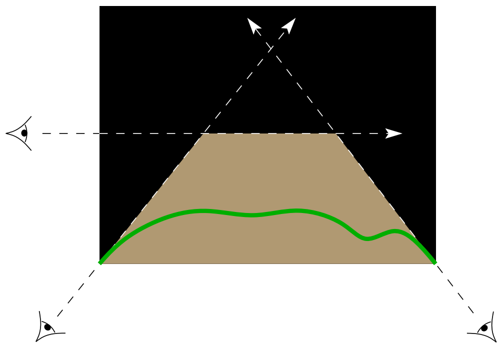


Figure 6.17: Intersections of viewing shafts for space carving. Three scans are taken from different points of view. The solid green curve is the actual surface. Due to occlusions (not shown), the bound on the object is determined primarily by space carving. Notice that in this two-dimensional example, the resulting boundary is an intersection of empty half-spaces and thus consists of large line segments with sizes that do not depend on the voxel resolution.

will declare the region near the apex to be empty. In this case, the hole filling algorithm will bridge the gap.

In the absence of an additional scan or the use of space carving, however, bridging the gap is still desirable. The range samples acquired at the apex from two viewpoints may be arbitrarily close together – even more closely spaced than the samples in individual range images – so the reconstruction algorithm ought to be able to bridge the gap. One possible solution would be a modified marching cubes algorithm that assumes regions of the volume near the surface are empty and performs a local hole-filling only to bridge small gaps. Alternatively, a different hole-filling algorithm could be applied as a post-process, directly filling small gaps in the polygonal reconstruction.

6.4.3 Space carving

In many cases, the space carving algorithm described in Section 6.2 can attain a reasonably tight bound on the object and lead to a plausible hole-free surface. In some cases, however, the hole-fill regions have undesirable properties. They represent intersections of viewing "shafts" that may in fact not achieve a tight bound on the surface and may appear as protrusions from the object, or, worse, they may change the topological genus of the reconstruction by introducing "handles". Figures 6.17, 6.18 and 6.19 illustrate these possibilities.

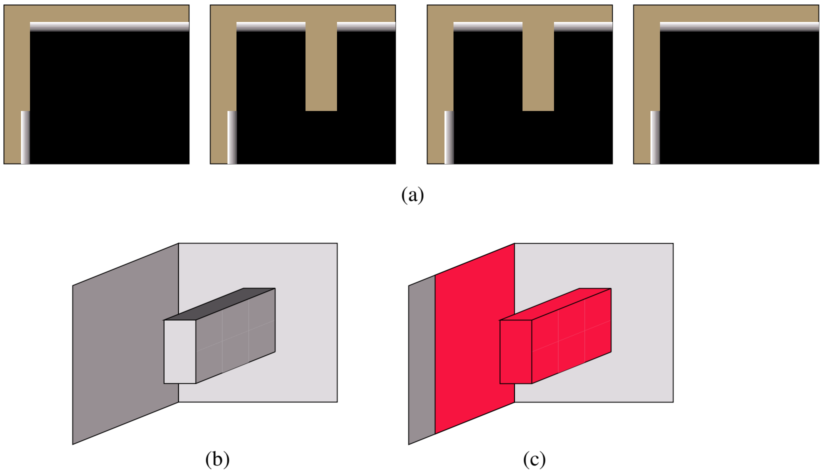


Figure 6.18: Protrusion generated during space carving. (a) Four slices through a volumetric grid show the signed distance ramps due to visible surfaces as well as the resultant space carving. (b) Taking an isosurface that includes hole-filling yields a surface with a protrusion. (c) Red surfaces indicate the hole fill regions of the isosurface.

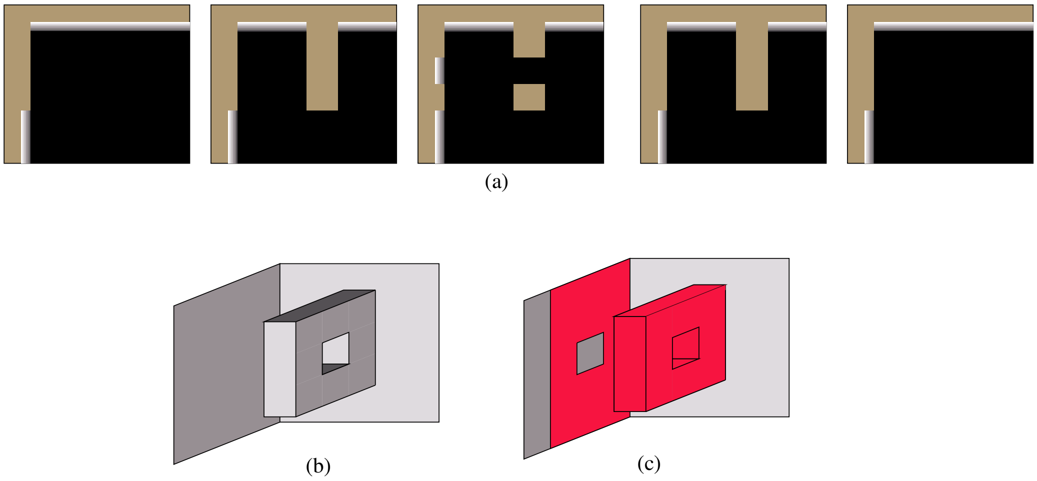


Figure 6.19: Errors in topological type due to space carving. (a) Five slices through a volumetric grid show the signed distance ramps due to visible surfaces as well as the resultant space carving. (b) The resulting isosurface has a "handle". (c) Red surfaces indicate the hole fill regions of the isosurface.

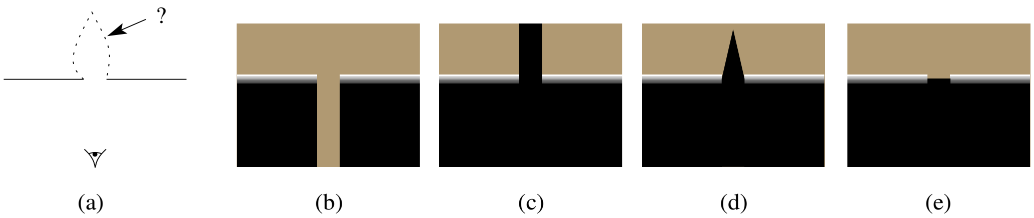


Figure 6.20: Aggressive strategies for space carving with a range sensor using a single line of sight per pixel. (a) A surface is scanned with a range sensor that has a single line of sight per sample. There is a gap in the recorded range data, which could correspond to the dotted surface. (b) A conservative strategy would carve space only from the observed surfaces. (c) A very aggressive strategy would incorporate the notion that missing data implies no surface along the sensor's line of sight. In this case, the carving extends to the limit of the sensor's field of view. (d) Due to sensitivity limitations, surfaces at grazing angles to the sensor may not be detected. This places a limit on how much space may be declared empty and still be consistent with a surface that just evades detection due to oblique orientation. (e) A still less aggressive strategy would simply fill the gap by bounding the emptiness at the line segment connecting the edges of the missing data. Such a strategy might lead to less objectionable hole fill regions such as would be generated by the indentation shown in (d).

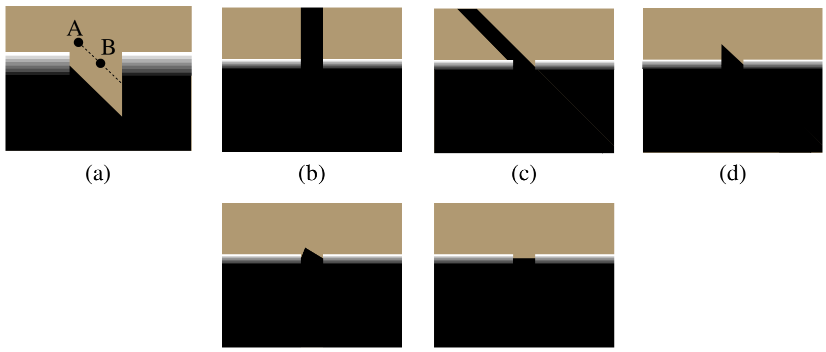
These undesirable hole fillers might be removed at the volumetric level or by operating on the reconstructed surface. One volumetric approach might be to apply image processing operations such as erosions and dilations to modify the hole fill regions and possibly collapse thin unknown regions that lead to "handles" such as the one shown in Figure 6.19 [Jain 1989]. In addition, more aggressive space carving strategies could be employed to empty out more of space as illustrated in Figure 6.20. Consider a very simple range sensor with a single line of sight. If such a sensor were to return no measurement, we would normally draw no conclusions. In fact, barring a surface that is "invisible" to the sensor, one can argue that there is no surface along the sensor's line of sight, or else it would have been detected. The space must therefore be empty in this direction. For a full range imaging sensor, we could follow all lines of sight that returned no range data and declare all these regions of the volume to be empty. In practice, this approach could not be applied indiscriminately. For instance, no data is returned from very dark regions of a surface or shiny portions that deflect the line of sight. Even if the surface were known to have a uniform diffuse reflectance, regions that are oblique to the sensor may not return enough light to be detected. In this case, there would be a limit to how far the space carving would proceed while being consistent with the possibility that the surface is receding with steep slope. Figure 6.20 shows how this more aggressive space carving might proceed.

In the case of a triangulation scanner, space carving is complicated by the requirement that each point must have a clear line of sight to both the laser and the camera. The conservative approach consists of following the lines of sight from the surface to both the laser and the camera and declaring the voxels to be empty as shown in Figure 6.21. For the case of aggressive space carving, we refer to Figure 6.22. If we hypothesize that point A in Figure 6.22a is not empty, then there must be a surface that occludes the line of sight of the laser or the sensor, e.g., point B. But in order for point B to be an occluder, then there must be yet another surface that blocks its visibility. Following this line of reasoning, we eventually determine that a surface must exist in the region of space already conservatively declared to be empty in order to prevent the detection of point A. This contradiction leads to the conclusion that point A must be empty. By induction, we can then argue that all points that are accessible to both the laser and the sensor must be empty [Freyburger 1994]. As with the single line of sight rangefinder, less aggressive strategies are also possible (Figure 6.22e and f).

Another approach to improving the hole fill regions would be to operate directly on the polygonal reconstruction. For example, the tangent planes to the surface at the boundary between the hole fill and the observed surface could be used to coerce the hole fill to relax into a less protruding configuration. Ideally, such a relaxation procedure could be extended to modify the topology and collapse undesirable handles in the hole fill regions.

(a) (b) (c) (d)

Figure 6.21: Conservative strategy for space carving with a triangulation range sensor. (a) A surface is scanned with a triangulation range sensor that has two lines of sight per sample. For the purpose of illustration, the camera is assumed orthographic and the scanning sweep to be linear. (b) The region of space that must be empty from the point of view of the laser. (c) The region of space that must be empty from the point of view of the camera. (d) The union of the empty spaces yields a conservative estimate of the emptiness of space around the observed surfaces.



(a) (b) (c) (d)

(e) (f)

Figure 6.22: Aggressive strategies for space carving with a triangulation range sensor. The sensor geometry is assumed the same as in Figure 6.21a. (a) Magnified view of the hole fill region established by conservative space carving in Figure 6.21d. Following the arguments in the text, all points that could be visible to both the laser and the sensor must be empty. (b) The region of space that could be empty from the point of view of the laser. (c) The region of space that could be empty from the point of view of the camera. (d) The intersection of the empty regions in (b) and (c) yields an aggressive estimate about the emptiness of space around the observed surface. (e) Surfaces at grazing angle to the laser or camera might not be detected, so a less aggressive strategy would taper the boundaries of the space carving region. (f) As in Figure 6.20, a less aggressive strategy would simply fill the gap by bounding the emptiness at the line segment connecting the edges of the missing data.