Chapter 5

Surface estimation from range images

The second part of this thesis is concerned with building computer models from range images such as might be generated using the method described in the first part of the thesis.

As stated in Chapter 1, the problem of reconstructing a surface from range images can be stated as follows:

> Given a set of p aligned, noisy range images, $\hat{f}_1, \cdots, \hat{f}_p$, find the 2D manifold that most closely approximates the points contained in the range images.

where each function, $\hat{f}_i$ is a sampling of the actual surface f, and each sample, $\hat{f}_i(j,k)$ is the observed distance to the surface as seen along the line of sight indexed by (j,k). The problem of finding this manifold hinges on defining how well any manifold approximates the range images, and then solving for the "best" manifold. In this chapter, we begin with a discussion of prior work in reconstructing surfaces from range data. Next, we describe range imaging in more detail and then construct a probabilistic model relating possible manifolds to range image samples. In Section 5.4, we define the best surface as the Maximum Likelihood surface, and we define the integral that this surface must minimize. In Section 5.5, we show how to bring the integral under a unified domain using a vector field analogy for range image sampling. In Section 5.6, we derive some results from the calculus of variations needed for the solution of this minimization problem. Finally, in Section 5.7, we solve the minimization equation for surface reconstruction from range images. As a special case, we relate the least squares minimization (corresponding to Gaussian statistics) to the search for a zero crossing of weighted sums of signed distance functions.

5.1 Prior work in surface reconstruction from range data

Surface reconstruction from dense range data has been an active area of research for several decades. The strategies have proceeded along two basic directions: reconstruction from unorganized points, and reconstruction that preserves the underlying structure of the acquired data. These two strategies can be further subdivided according to whether they operate by reconstructing polygonal surfaces or by reconstructing an implicit function.

5.1.1 Unorganized points: polygonal methods

A major advantage of the unorganized points algorithms is the fact that they do not make any prior assumptions about connectivity of points. In the absence of range images or contours to provide connectivity cues, these algorithms are the only recourse. Among the polygonal surface approaches, Boissonnat [1984] describes a method for Delaunay triangulation of a set of points in 3-space. Edelsbrunner & Mücke [1992] generalize the notion of a convex hull to create surfaces called alpha-shapes.

5.1.2 Unorganized points: implicit methods

Examples of implicit surface reconstruction include the method of Hoppe et al. [1992] for generating a signed distance function followed by an isosurface extraction. More recently, Bajaj et al. [1995] used alpha-shapes to construct a signed distance function to which they fit implicit polynomials.

Although unorganized points algorithms are widely applicable, they discard useful information such as surface normal and reliability estimates. As a result, these algorithms are well-behaved in smooth regions of surfaces, but they are not always robust in regions of high curvature and in the presence of systematic range distortions and outliers.

5.1.3 Structured data: polygonal methods

Among the structured data algorithms, several polygonal approaches have been proposed. Soucy & Laurendeau [1992] describe a method using Venn diagrams to identify overlapping data regions, followed by re-parameterization and merging of regions. Turk & Levoy [1994] employ an incremental algorithm that updates a reconstruction by eroding redundant geometry, followed by zippering along the remaining boundaries, and finally a "consensus" step that reintroduces the original geometry to establish final vertex positions. Rutishauser et al. [1994] use errors along the sensor's lines of sight to establish consensus surface positions followed by a re-tessellation that incorporates redundant data. These algorithms typically perform better than unorganized point algorithms, but they can still fail catastrophically in areas of high curvature.

5.1.4 Structured data: implicit methods

Several algorithms have been proposed for integrating structured data to generate implicit functions. These algorithms can be classified as to whether voxels are assigned one of two (or three) states or are samples of a continuous function.

Discrete-state voxels

Among the discrete-state volumetric algorithms, Connolly [1984] casts rays from a range image accessed as a quad-tree into a voxel grid stored as an octree, and generates results for synthetic data. Chien et al. [1988] efficiently generate octree models under the severe assumption that all views are taken from the directions corresponding to the 6 faces of a cube. Li & Crebbin [1994] and Tarbox & Gottschlich [1994] also describe methods for generating binary voxel grids from range images. None of these methods has been used to generate surfaces. Further, without an underlying continuous function, there is no mechanism for representing range uncertainty in the volumes.

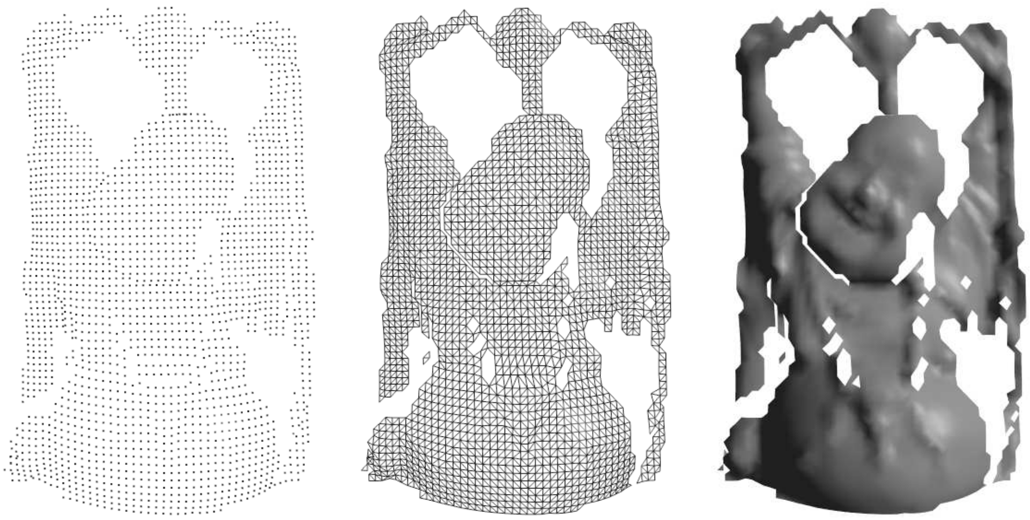
Continuous-valued voxels

The last category of our taxonomy is implicit function methods that use samples of a continuous function to combine structured data. Our method falls into this category. Previous efforts in this area include the work of Grosso et al. [1988], who generate depth maps from stereo and average them into a volume with occupancy ramps of varying slopes corresponding to uncertainty measures; they do not, however, perform a final surface extraction. Succi et al. [1990] create depth maps from stereo and optical flow and merge them volumetrically using a straight average of estimated voxel occupancies. The reconstruction is an isosurface extracted at an arbitrary threshold. In both the Grosso and Succi papers, the range maps are sparse, the directions of range uncertainty are not characterized, they use no time or space optimizations, and the final models are of low resolution. Recently, Hilton et al. [1996] have developed a method similar to ours in that it uses weighted signed distance functions for merging range images, but it does not address directions of sensor uncertainty, incremental updating, space efficiency, and characterization of the whole space for potential hole filling, all of which we believe are crucial for the success of this approach.

5.1.5 Other related work

Other relevant work includes the method of probabilistic occupancy grids developed by Elfes & Matthies [1987]. Their volumetric space is a scalar probability field which they update using a Bayesian formulation. The results have been used for robot navigation, but not for surface extraction. A difficulty with this technique is the fact that the best description of the surface lies at the peak or ridge of the probability function, and ridge finding is not a problem with robust solutions [Eberly et al. 1994].

The discrete-state implicit function algorithms described above also have much in common with the methods of extracting volumes from silhouettes [Connolly 1974] [Martin & Aggarwal 1983] [Hong & Shneier 1985] [Potmesil 1987] [Szeliski 1993]. The idea of using backdrops to help carve out the emptiness of space is one we demonstrate in Chapter 6.



(a) (b) (c)

Figure 5.1: From range image to range surface. The range image, $\hat{f}(i,j)$ in (a) is a set of points in 3D acquired on a regular sampling lattice. (b) shows the reconstruction of a range surface, $\tilde{f}(x,y)$, using triangular elements. (c) is a shaded rendering of the range surface.

5.2 Range images, range surfaces, and uncertainty

In the remainder of this chapter, we focus on the problem of reconstructing a surface exclusively from range images. We begin by describing range imaging in more detail.

Numerous range scanning technologies acquire the shape of an object by generating range images. A *range image* is a 2D lattice of range samples, where each sample corresponds to a distance from the sensor along a single line of sight. A *range surface*, $\tilde{f}$, is a piecewise continuous function reconstructed from a range image. Figure 5.1 shows a range surface reconstructed using triangular elements to connect nearest neighbors in the range image. Such a range surface is a piecewise linear reconstruction, and corresponds to a partial estimate of the shape and topology of the object. To avoid making topological errors such as bridging depth discontinuities, researchers typically set an edge length or surface orientation threshold when establishing connectivity of samples.

Rangefinders might have lines of sight distributed across a range image in any one of a number of different configurations. Figure 5.2 depicts several range scanning geometries and their corresponding viewing frustums. An orthographic imaging frustum may arise from a 2-axis translational scanning configuration as depicted in Figures 5.2a and 5.2b, though no such commercial system exists to our knowledge. Figure 5.2c shows a time of flight scanner with two rotating mirrors; the lines of sight for such a configuration are nearly perspective, assuming the separation between the mirrors is small. Rangefinders are in general neither precisely orthographic nor precisely perspective. Optical distortions in real range imaging systems typically lead to violations of paraxial assumptions about the imaging optics. In addition, some scanners deviate significantly from orthographic or perspective. For instance, some light stripe triangulation scanners sample an object with a perspective projection within the plane of light, but translate horizontally to fill out the range image. Such a projection is a cylindrical or "line perspective" projection; i.e., it is orthographic in the direction of scanning, and perspective within the scanning plane. This last scanning geometry corresponds to the Cyberware scanner used for this thesis (see Figures 5.2e and 5.2f). When applying traditional methods of optical triangulation, samples lie along rays that correspond to the projection of each CCD scanline onto the laser sheet. These sampling lines of sight need *not* coincide with the lines of sight of the laser (though they do happen to coincide in the Cyberware scanner). Figure 5.3 illustrates this point. When applying the spacetime analysis for triangulation, however, the sampling lines of sight follow the lines of sight of the CCD.

While their range imaging frustums are diverse, most rangefinders do have in common that the lines of sight of the sensor are well calibrated and the range uncertainty lies predominantly in determining depth along these lines of sight. For a time of flight rangefinder, for example, the primary source of uncertainty is in determining the time of arrival of a light pulse relative to its time of emission, resulting in depth errors along the direction of the emitted pulse. In optical triangulation, traditional analysis of imaged laser reflections leads to uncertainties in depth along the laser beam or within the laser sheet, as shown in Figure 2.6. Using the spacetime analysis for optical triangulation, the errors in determining the location of a peak in time, lead to range errors along the camera's lines of sight, as shown in Figure 3.5. Other researchers have also characterized range errors for optical triangulation, and have found the error to be ellipsoidal, i.e., non-isotropic, about the range points [Hebert et al. 1993] [Rutishauser et al. 1994].

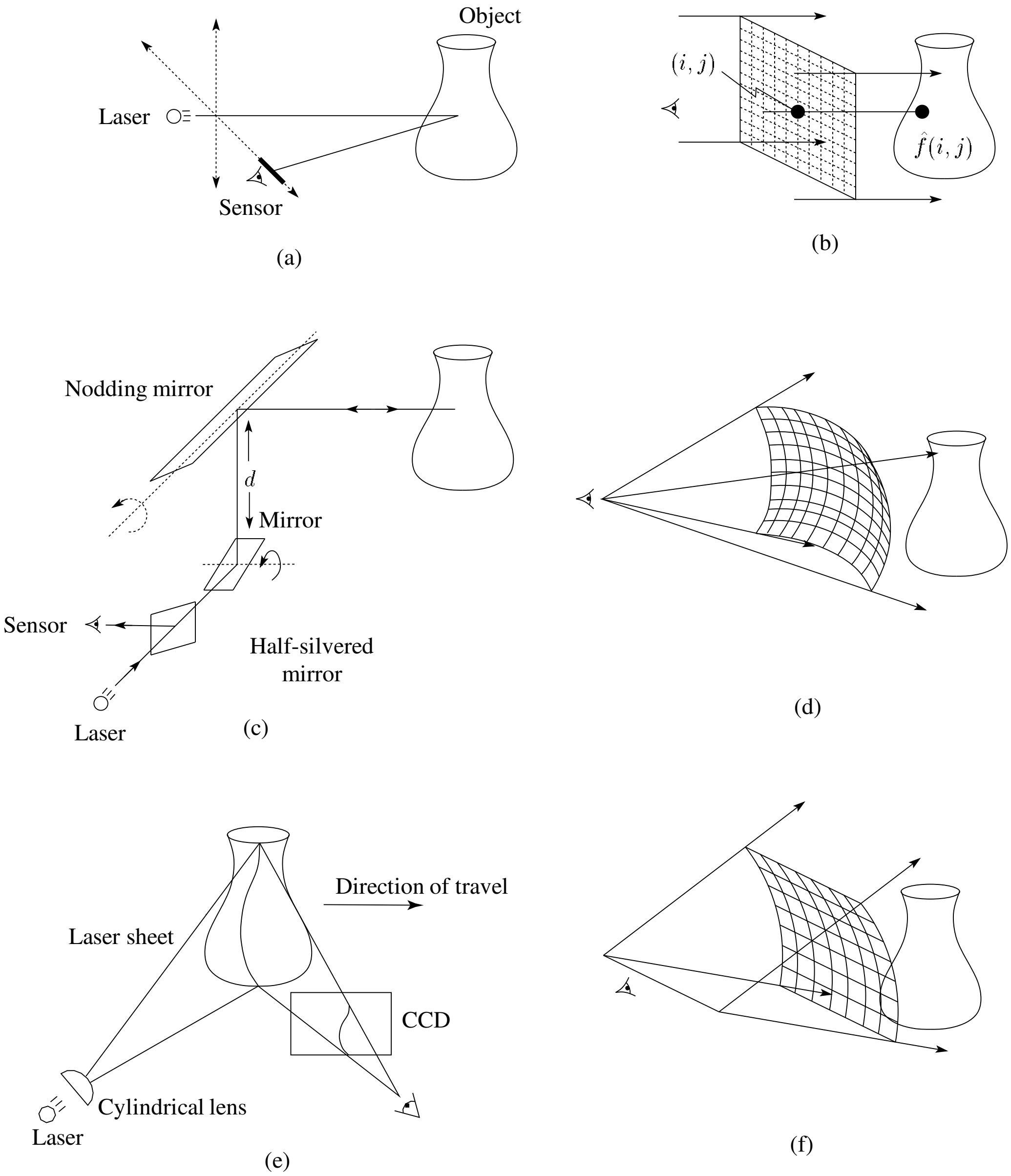


Figure 5.2: Range imaging frustums. (a) An orthographic triangulation scanner using a beam of light and a linear array sensor and (b) its orthographic viewing frustum. The sensor and light move together, stepping in fixed increments horizontally and vertically to traverse its field of view. (c) A time-of-flight scanner with rotating mirrors to scan the beam and sensor line of sight over the object and (d) its approximately perspective viewing frustum (for small mirror separation, d). (e) A translating laser stripe triangulation scanner with an area sensor and (f) its cylindrical projection viewing frustum when using traditional (*not* spacetime) triangulation analysis.

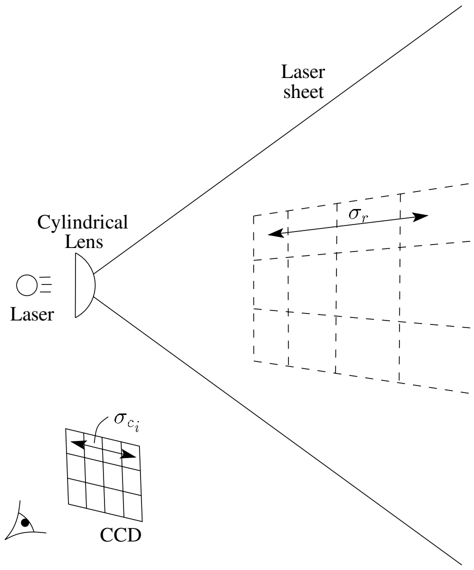


Figure 5.3: Range errors for traditional triangulation with two-dimensional imaging. A typical optical triangulation system uses a laser beam spread into a diverging sheet of light with a cylindrical lens and a CCD that images reflections from within this sheet. Range is usually determined by computing the centers of the imaged laser reflections on each scanline. Thus, the errors in determining the centers correspond to errors in range that vary along the projections of the scanlines onto the laser sheet. Notice that the divergence of the laser sheet need not coincide with the divergence of the projected scanlines.

5.3 A probabilistic model

Before applying a statistical method to estimate the best manifold corresponding to a set of range images, we first need to define a probabilistic model that describes the likelihood of any possible manifold. To this end, we seek the probability distribution function (pdf):

$$\mathrm{pdf}(f \mid \hat{f}_1, \cdots, \hat{f}_p)$$

where f is a possible manifold approximating the range images $\{\hat{f}_i\}$. In order to make the analysis feasible, we make some assumptions about the nature of this pdf. At the end of this chapter, we will discuss the validity of these assumptions. First, we assume that the uncertainties in the range images are independent of one another, giving us:

$$\mathrm{pdf}(f \mid \hat{f}_1, \cdots, \hat{f}_p) = \prod_{i=1}^{p} \mathrm{pdf}(f \mid \hat{f}_i) \tag{5.1}$$

Next, we assume that sample errors within a range image are independently distributed:

$$\mathrm{pdf}(f \mid \hat{f}_i) = \prod_{j=1}^{m} \prod_{k=1}^{m} \mathrm{pdf}(f \mid \hat{f}_i(j,k))$$

where the range images are of dimension $m \times m$. This independence assumption also requires that the registration of the range images is very accurate. Further, if we consider the sampling errors to be distributed along the lines of sight of the sensor, then:

$$\mathrm{pdf}(f \mid \hat{f}_i) = \prod_{j=1}^{m} \prod_{k=1}^{m} \mathrm{pdf}(f_i(j,k) \mid \hat{f}_i(j,k)) \tag{5.2}$$

where $f_i(j,k)$ is the reparameterization of f over the domain of the i'th range image, resampled along the sensor direction at (j,k). Combining Equations 5.1 and 5.2, we have:

$$\mathrm{pdf}(f \mid \hat{f}_1, \cdots, \hat{f}_p) = \prod_{i=1}^{p} \prod_{j=1}^{m} \prod_{k=1}^{m} \mathrm{pdf}(f_i(j,k) \mid \hat{f}_i(j,k)) \tag{5.3}$$

Choosing a single function that "best" approximates the range surfaces takes us from the realm of probability into the realm of statistics.

5.4 Maximum likelihood estimation

One of the most common statistical methods for estimating parameters is the method of Maximum Likelihood (ML), which chooses the parameter values that maximize a pdf [Papoulis 1991]. In our case, we seek the surface function, f, that maximizes the pdf described in Equation 5.3. Note that maximizing the pdf is equivalent to minimizing $-\log(\mathrm{pdf})$, due to the strictly monotonic nature of the logarithm function. Thus, we can transform our problem to solving for the f that minimizes:

$$E(f) = -\sum_{i=1}^{p} \sum_{j=1}^{m} \sum_{k=1}^{m} \log\left[\mathrm{pdf}(f_i(j,k) \mid \hat{f}_i(j,k))\right]$$

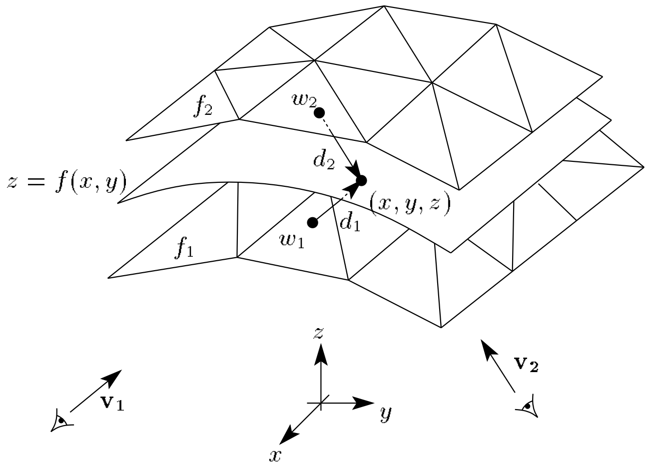


Figure 5.4: Two range surfaces, f_1 and f_2, are tessellated range images acquired from directions v_1 and v_2. In the case of Gaussian statistics, the possible range surface, $z = f(x, y)$, is evaluated in terms of the weighted squared distances to points on the range surfaces taken along the lines of sight to the sensor. A point, (x, y, z), is shown here being evaluated to find its corresponding signed distances, d_1 and d_2, and weights, w_1 and w_2.

A difficulty with this formulation, however, is the fact that the function f need only minimize the function $E(f)$ for the points that project onto the surface for each sampled range image. In between these points, the surface need not even be continuous. In fact, the ML estimate of f would simply be the original points in the range images. Even if we required continuity, the function f could simply interpolate the points in the range images and be almost arbitrarily ill-behaved between points. To remedy this problem, we could impose some continuity and smoothness constraints on the function f. Alternatively, we can interpret the range images as range surfaces; i.e., by suitable interpolation of the range images, we can generate projected *surface* estimates of f. Figure 5.4 illustrates this principle. In this case, the function to minimize becomes:

$$E(f) = \sum_{i=1}^{p} \iint_{A_i} g(f_i(u, v), \tilde{f}_i(u, v)) du dv \tag{5.4}$$

where $\tilde{f}_i$ is the interpolated range surface corresponding to $\hat{f}_i$, we have replaced $-\log[\mathrm{pdf}()]$ with $g()$, and we have taken the limit of the summation to be an integral. Further, the (u, v) notation is a parameterization of the sensor's sampling rays. This parameterization could pertain to, for example, an (x, y) parameterization in the orthographic case or a (θ, ϕ) parameterization in the perspective case.

One familiar pdf to choose is the Gaussian (normal) distribution:

$$\mathrm{pdf}(f_i(u,v) \mid \tilde{f}_i(u,v)) = c_i(u,v) \exp\left[-\frac{1}{2} \left(\frac{\tilde{f}_i(u,v) - f_i(u,v)}{\sigma_i(u,v)} \right)^2 \right]$$

where c_i is the normalization factor and σ_i is the standard deviation. The numerator in the squared term is simply the distance between reconstructed range surface points and points on the candidate surface, where the distance is taken along the sensing direction:

$$d_i(u,v,f_i) = \tilde{f}_i(u,v) - f_i(u,v)$$

Substituting and evaluating $-\log(\mathrm{pdf})$, we obtain:

$$g(f_i(u,v), \tilde{f}_i(u,v)) = \frac{1}{2} w_i(u,v) d_i(u,v,f_i)^2 - \log(c_i(u,v))$$

where we have set $w_i(u,v) = 1/\sigma_i(u,v)^2$. Neglecting the additive term that does not depend on f (and thus has no bearing on the minimization of $E(f)$ with respect to f), the error equation becomes:

$$E(f) = \frac{1}{2} \sum_{i=1}^{p} \int\!\!\int_{A_i} w_i(u,v) d_i(u,v,f_i)^2 du dv$$

Thus, the assumption of Gaussian statistics leads to a weighted least squares minimization, where distances are taken between range surfaces and the minimizing function along range image viewing directions.

Note that the domain of integration in the previous equations depends on the viewpoint and ray distribution of the sensor. By choosing a common domain of integration, we can bring the summation under the integral and attempt to minimize $E(f)$ by applying the calculus of variations.

5.5 Unifying the domain of integration

In order to solve for the optimal surface, we would like to put the total error (Equation 5.4) in a form that resembles:

$$E(f) = \iint_A \sum_{i=1}^{p} e_i(x, y, f)dxdy$$

where we have chosen the domain of integration to be the $x - y$ plane over which our candidate surface, $f(x, y)$, is a function. The function $e_i(x, y, f)$ is the error associated with the point, (x, y, f), on the surface as measured with respect to the i'th range image.

To help see the connections we need to make, we can rewrite Equation 5.4 as:

$$E(f) = \sum_{i=1}^{p} \iint_{A_i} g_i(u, v, f_i)dudv$$

where we define g_i to be:

$$g_i(u, v, f_i) \equiv g(f_i(u, v), \tilde{f}_i(u, v))$$

Our goal then is to find a relationship between e_i and g_i through their respective domains:

$$e_i(x, y, f)dxdy \xleftrightarrow{?} g_i(u, v, f_i)dudv$$

In fact, part of the connection is already made, because e_i and g_i are simply reparameterizations of one another; e_i is the error at a point, (x, y, f), on the surface, and g_i is the error at the same point, expressed in the coordinates of the sensor, (u, v, f_i).

The differentials $dxdy$ and $dudv$ are generally related to one another through the Jacobian. However, rather than compute the Jacobian directly, we prefer to derive the relationship between the differentials explicitly using geometric arguments. The intuitions gained help in generalizing from the simplest case, orthographic projection, to more arbitrary viewing frustums such as line perspective. Thus, we will first consider the case of an orthographic sensor.

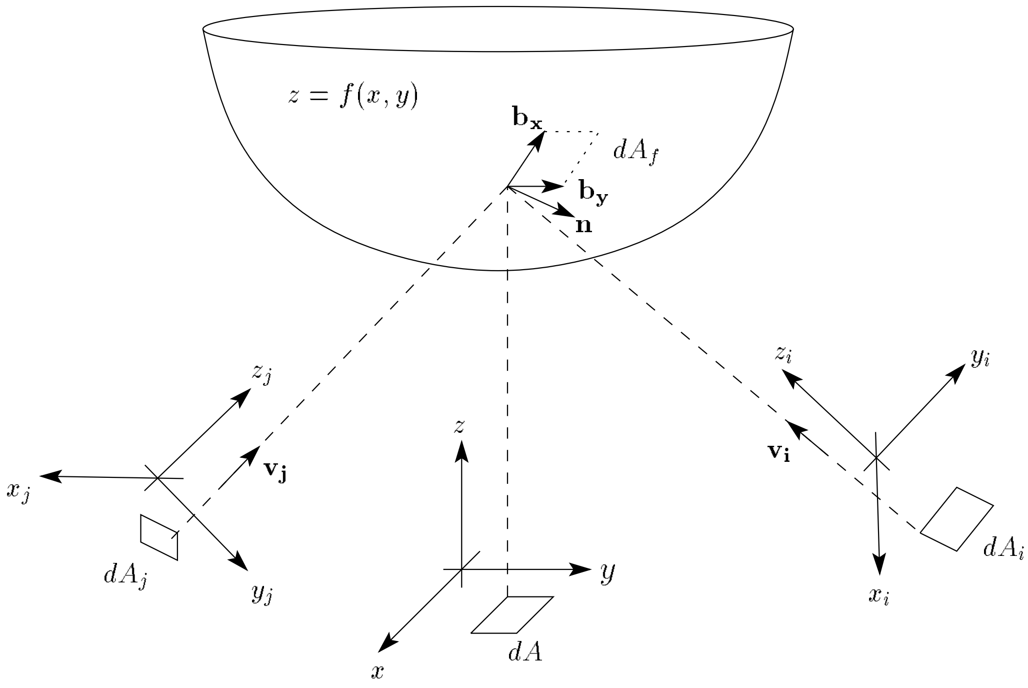


Figure 5.5: Differential area relations for reparameterization. Orthographic range images are acquired from different viewpoints, yielding surface parameterizations over domains such as (x_i, y_i) and (x_j, y_j). Instead, we want to estimate a surface over a canonical domain (x, y). When integrating over this canonical domain, we must relate differential areas in this domain to areas in the domains of the original range images.

For an orthographic sensor, we must project the area $dA = dxdy$ onto the function $f(x, y)$, and then re-project it onto the i'th range image's viewing plane, as indicated in Figure 5.5. We use Cartesian (x_i, y_i, z_i) coordinates to represent (u, v, f_i). First, we note that moving a unit distance in the x-direction in the (x, y) domain corresponds to moving along the vector $\mathbf{b_x}$ on the surface f, where:

$$\mathbf{b_x} = \left(1, 0, \frac{\partial z}{\partial x}\right)$$

Similarly, moving in the y-direction corresponds to moving along $\mathbf{b_y}$ on the surface:

$$\mathbf{b_y} = \left(0, 1, \frac{\partial z}{\partial y}\right)$$

If we define the vector, $\mathbf{a}$, as:

$$\begin{aligned} \mathbf{a} &= \mathbf{b_y} \times \mathbf{b_x} \\ &= \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1 \right) \end{aligned}$$

then the projected area, dA_f, on the surface is:

$$dA_f = |\mathbf{a}| dA$$

and the normal, $\mathbf{n}$ to the surface[1] is:

$$\mathbf{n} = \frac{\mathbf{a}}{|\mathbf{a}|}$$

The projected area, $dA_i = dx_i dy_i$ on the i'th range image plane is then:

$$\begin{aligned} dA_i &= (-\mathbf{n} \cdot \mathbf{v_i}) dA_f \\ &= -\left(\frac{\mathbf{a}}{|\mathbf{a}|} \cdot \mathbf{v_i} \right) |\mathbf{a}| dA \\ &= -(\mathbf{a} \cdot \mathbf{v_i}) dA \end{aligned}$$

This leads us to a reformulation of the equation to be minimized:

$$E(f) = \iint_A \sum_{i=1}^{p} e_i(x, y, z) \left[\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1 \right) \cdot \mathbf{v_i} \right] dx dy \qquad (5.5)$$

where we have replaced $f(x, y)$ with z on the right hand side. This equation is now ready for minimization under the calculus of variations, but it applies only to orthographic equations in this form.

How does this equation change when we handle more complex projections, such as perspective or line perspective? In these cases, we would find that the direction of projection is

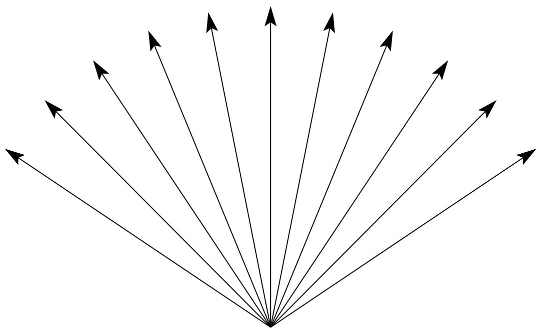
[1] See [Horn 1986] for a similar derivation of the normal to a surface.

not constant as in the orthographic case, but varies over space. Further, the contribution of a surface element varies not only with the orientation of the element, but also with distance from the center of projection. For example, as an element moves away from the center of projection for the perspective case, its apparent size falls off with $1/r^2$, where r is the distance from the center.

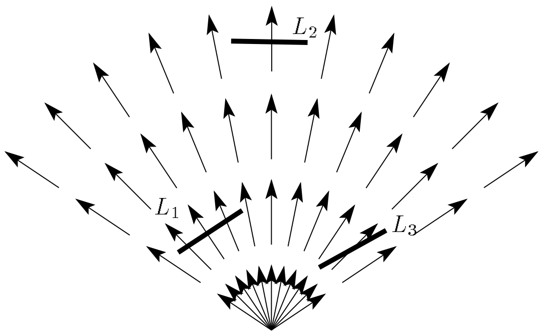
One approach to the problem might be to define a suitable projection surface that captures the variation in orientation of projection rays and onto which we can project a surface element to compute its contribution to the error integral. For example, we could define a unit sphere about the center of projection for the perspective case. The directions of projection would run radially, normal to the surface of the sphere, and we could project a surface element dA_f onto the sphere and use this area as the measure of the contribution of the element. This definition leads to the measurement dA_f in terms of the solid angle it subtends. This is exactly what we would expect, and it will yield the desired $1/r^2$ fall-off. For the line perspective, the projection surface would be a cylinder of unit radius, yielding projection rays that emanate radially from the center line, and the contribution of a surface element would fall off as $1/r$, where r is the distance from the center line.

While the use of these projection surfaces will work in a number of cases, it is too restrictive a definition and will not work in all cases. Instead, we propose an alternative and more general way of looking at these projections. Consider the two-dimensional case of a perspective projection as shown in Figure 5.6. The projection rays radiate from the center of projection. For every point in space (except the center of projection), a single ray passes through that point in a particular direction. The directions evaluated throughout space define a vector field as shown in Figure 5.6b. We could then compute the projected length of a unit line element by dotting the normal of the element with the projection direction at the center of the element.

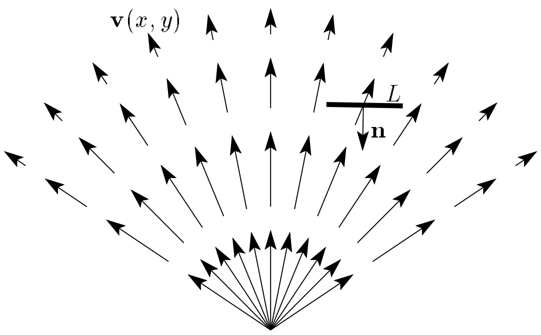
The direction field alone, however, does not tell us the contribution the element would have to an integral over all directions in the projection. This contribution should be a measure of how many rays cross the a unit line segment in the direction of projection. In our example, this "ray density" should fall of with the inverse distance from the center of projection. By weighting our direction field by the "ray density" measure, we arrive at a new



(a)



(b)



(c)

Figure 5.6: Ray density and vector fields in 2D. (a) For a perspective projection in 2D, the sensor is effectively a point source casting rays to sample space. (b) We can view this set of rays as defining a vector field, where each point in space has an associated ray direction. The contribution of elements in the field for the purposes of integration is defined by how many rays pass through the elements; element L_1 receives more rays than element L_2, because it is closer to the sensor. It also receives more rays than L_3, because it is more normal to the ray direction. (c) To encode the variations in ray density, we multiply the direction field in (b) by the ray density. Now the contribution of an element is $|L|\mathbf{n} \cdot \mathbf{v}(x, y)$.

vector field, depicted in Figure 5.6c. For a unit line segment, dotting the normal of the segment with the strength of this vector field will yield the total contribution of the element.

This "ray density" field has a familiar physical interpretation. If we think of the center of projection as being a point source sending equal amounts of particles in all directions, then the "ray density" field is the vector flow field describing the direction and rate of particle flow across unit projected lengths. If we think of these particles as photons, we arrive at a description of the light field about the point source, and we can measure the contribution of a surface as the amount of light that flows through the element. We assume the projections are one-to-one, so the light fields will have a single flow direction at every point.

One of the most important elements of this physical analogy, is the fact that the projection rays emanate from the sensor, but they flow forever without being extinguished or created anew in space. In other words, this flow field is *conservative* at all points outside of the regions where the rays originate. Further, the projection rays are fixed for each range scan; i.e., the flow field is *static*. These conservative and static properties will prove very useful in the next section.

In general, any range imaging frustum will define a vector field whose directions are consistent with the projection rays and whose magnitude is equal to the flux of rays crossing a unit projected area. For the case of an orthographic projection, this vector field is simply a unit vector in the direction of projection:

$$\mathbf{v}(x, y, z) = \hat{\mathbf{v}}$$

A perspective projection yields a radial field with $1/r^2$ fall-off:

$$\mathbf{v}(r, \theta, \phi) = \frac{\hat{\mathbf{r}}}{r^2}$$

where $\hat{\mathbf{r}} = \mathbf{r}/|r|$.

The line perspective results in a cylindrically radial field with $1/r$ fall-off:

$$\mathbf{v}(r, \theta, x) = \frac{\hat{\mathbf{r}}}{r}$$

In the latter two cases, we have expressed the vector fields in spherical and cylindrical coordinates, but they are all equivalently expressible in Cartesian coordinates.

With this new vector flow field analogy, we can now express the relation between $dA = dxdy$ and the ray differential $dA_i = du_i dv_i$ as:

$$dA_i = \left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \mathbf{v_i}(x, y, z) dA$$

Similarly, the error equation becomes:

$$E(f) = \iint_A \sum_{i=1}^{p} e_i(x, y, z) \left[\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \mathbf{v_i}(x, y, z)\right] dxdy \tag{5.6}$$

This equation is almost identical to Equation 5.5, except that the constant vector $\mathbf{v_i}$ has been replaced with a vector field $\mathbf{v_i}(x, y, z)$. For the special case of Gaussian statistics, this integral becomes:

$$E(f) = \iint_A \sum_{i=1}^{p} w_i(x, y, z) d_i(x, y, z)^2 \left[\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \mathbf{v_i}(x, y, z)\right] dxdy \tag{5.7}$$

Note how the integration over a common domain brings the summation under the integral. This new formulation allows us to apply the calculus of variations to find the $z = f(x, y)$ that minimizes the integral.

5.6 Calculus of variations

One of the primary objectives of the calculus of variations is to take an integral of the form:

$$I = \iiint h\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) dxdydz$$

and compute the function:

$$z = f(x, y)$$

that minimizes the integral. A central result of the calculus of variations states that the minimizing function must obey a partial differential equation known as the Euler-Lagrange equation [Weinstock 1974]:

$$\frac{\partial h}{\partial z} - \frac{\partial}{\partial x}\frac{\partial h}{\partial(\partial z/\partial x)} - \frac{\partial}{\partial y}\frac{\partial h}{\partial(\partial z/\partial y)} = 0$$

We can easily extend this result to the case of minimizing integrals of sums of functions:

$$I = \iiint \sum_{i=1}^{p} h_i\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) dxdydz$$

Substituting into the Euler-Lagrange equation, we get:

$$\frac{\partial \sum_{i=1}^{p} h_i}{\partial z} - \frac{\partial}{\partial x}\frac{\partial \sum_{i=1}^{p} h_i}{\partial(\partial z/\partial x)} - \frac{\partial}{\partial y}\frac{\partial \sum_{i=1}^{p} h_i}{\partial(\partial z/\partial y)} = 0$$

and from the linearity of partial differentiation, we arrive at the result:

$$\sum_{i=1}^{p}\left[\frac{\partial h_i}{\partial z} - \frac{\partial}{\partial x}\frac{\partial h_i}{\partial(\partial z/\partial x)} - \frac{\partial}{\partial y}\frac{\partial h_i}{\partial(\partial z/\partial y)}\right] = 0$$

In order to solve an equation such as Equation 5.6 described in the previous section, we must first propose the following theorem.

Theorem 5.1 *Given the integral:*

$$I = \iiint h\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) dxdydz$$

where:

$$z = f(x, y)$$

and the function h is of the form:

$$h\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) = e(x, y, z)\left[\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \mathbf{v}(x, y, z)\right]$$

the function z that minimizes the integral satisfies the relation:

$$\mathbf{v} \cdot \nabla e + e \nabla \cdot \mathbf{v} = 0 \tag{5.8}$$

See Appendix A for the proof. Under the conditions that $\mathbf{v}(x, y, z)$ corresponds to a static, conservative flow field, then the following corollary obtains:

Corollary 5.1 *Given the integral:*

$$I = \iiint h\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) dxdydz$$

where:

$$z = f(x, y)$$

and the function h is of the form:

$$h\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) = e(x, y, z)\left[\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \mathbf{v}(x, y, z)\right]$$

where $\mathbf{v}(x, y, z)$ corresponds to a conservative *and* static *vector flow field, then the function z that minimizes the integral satisfies the relation:*

$$\mathbf{v} \cdot \nabla e = 0 \tag{5.9}$$

The proof is straightforward: the divergence of a conservative vector flow field ($\nabla \cdot \mathbf{v}$) must be exactly zero if the field is not changing with time [Schey 1992]. Thus, the result of Theorem 5.1 is simplified by removing the divergence term.

Corollary 5.1 has an intuitive interpretation. Consider minimizing the same integral for a constant vector field (corresponding to an orthographic projection) oriented in the $+z$ direction. The integrand would become $h = e(x, y, z)$ and would not depend on any partials of z. In this case, the obvious solution would be to minimize with respect to z only; i.e., compute:

$$\frac{\partial e}{\partial z} = 0$$

The only reason h becomes dependent on partials in the theorem, is so that we can solve the problem over a new domain, rotated with respect to the original domain (for the orthographic case). The solution in this new domain is actually the same; we take the derivative along the direction in which the function can vary. We can see this by noting that the dot product in Equation 5.9 is equivalent to a directional derivative taken along the line of sight for the original domain.

In order to solve the minimization equation posed in the previous section (Equation 5.6), we need to extend Theorem 5.1 to minimization over sums of functions. This leads to the following corollary:

Corollary 5.2 *Given the integral:*

$$I = \iiint h\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) dxdydz$$

where:

$$z = f(x, y)$$

and the function h is of the form:

$$h\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) = \sum_{i=1}^{n} e_i(x, y, z)\left[\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \mathbf{v_i}(x, y, z)\right]$$

where the $\mathbf{v_i}(x, y, z)$ correspond to conservative *and* static *vector flow fields, then the function z that minimizes the integral satisfies the relation:*

$$\sum_{i=1}^{p} \mathbf{v_i} \cdot \nabla e_i = 0$$

The proof is a straightforward application of the linearity of the Euler-Lagrange equation in combination with Corollary 5.1.

5.7 A minimization solution

We can now derive an equation for the function $z = f(x, y)$ that minimizes Equation 5.6. Namely, z must satisfy the relation:

$$\sum_{i=1}^{p} \mathbf{v_i}(x, y, z) \cdot \nabla e_i(x, y, z) = 0$$

Noting that a unit vector dotted with the gradient of a function is the same as taking the derivative in the direction of that vector, we can re-write the equation as:

$$\sum_{i=1}^{p} |\mathbf{v_i}| D_{\hat{\mathbf{v}}_\mathrm{i}}[e_i] = 0$$

where $D_{\hat{\mathbf{v}}_\mathrm{i}}$ represents the derivative in the direction of the sensor line of sight, $\hat{\mathbf{v}}_\mathbf{i}$, and we have dropped the (x, y, z) notation.

Consider the case where e_i corresponds to the $-\log(\mathrm{pdf})$ with Gaussian statistics. The solution equation is then:

$$\sum_{i=1}^{p} |\mathbf{v_i}| D_{\hat{\mathbf{v}}_\mathrm{i}}[w_i d_i^2] = 0$$

To simplify this relation, we note that the weighting function varies across the range image, but not along the sensor lines of sight. Accordingly:

$$D_{\hat{\mathbf{v}}_\mathrm{i}}[w_i d_i^2] = 2 w_i d_i D_{\hat{\mathbf{v}}_\mathrm{i}}[d_i]$$

Next, we note that the distance from the range surface is a linear function of constant slope along the viewing direction:

$$D_{\hat{\mathbf{v}}_\mathrm{i}}[d_i] = 1$$

Putting these results together, we obtain:

$$\sum_{i=1}^{p} |\mathbf{v_i}| w_i d_i = 0 \tag{5.10}$$

Thus, we arrive at a surprisingly simple result: the weighted least squares surface is determined by the zero-isosurface of the sums of the weighted signed distance functions.

In the orthographic case, the value of $|\mathbf{v_i}|$ is unity, so the minimization equation reduces to:

$$\sum_{i=1}^{p} w_i d_i = 0$$

while for the perspective case, we arrive at:

$$\sum_{i=1}^{p} \frac{w_i d_i}{r_i^2} = 0$$

where the center of projection of the i'th range image is at (x_i^c, y_i^c, z_i^c), and:

$$r_i = \sqrt{(x - x_i^c)^2 + (y - y_i^c)^2 + (z - z_i^c)^2}$$

Note the distinction here between d_i and r_i: d_i is the distance from a point in 3-space to the ith range image along the sensor's line of sight, while r_i is the distance from the same point to the center of projection of the sensor.

5.8 Discussion

We have shown that, under a set of assumptions, the surface of Maximum Likelihood derived from a set of range images can be found as a zero isosurface of a function over 3-space. The function defined in this domain is a sum of functions generated by each range image, and thus may be updated one range image at a time. For the case of a weighted least squares minimization, the optimal surface is the zero isosurface of the sum of the weighted signed distances along the lines of sight of the range images. Here we review the assumptions and evaluate the validity of each.

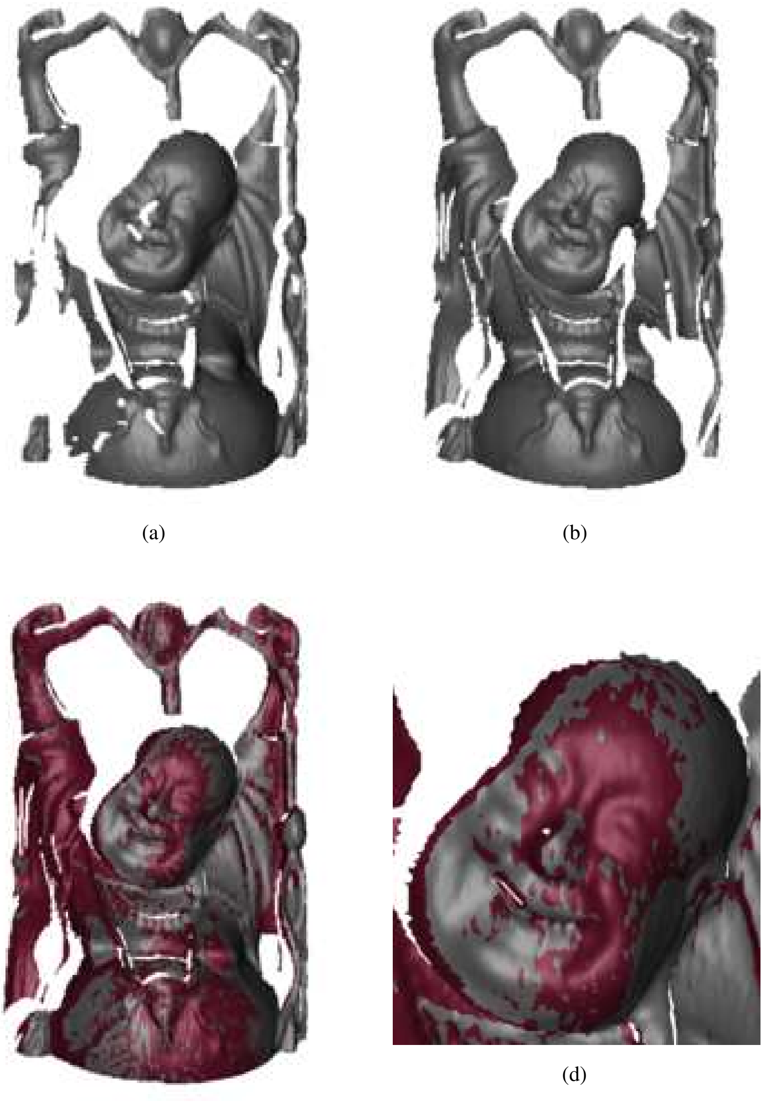
All range image samples are statistically independent. Each range image is likely to have little dependence on the other range images, since they are taken as separate events. However, whether or not samples *within* a range image exhibit independence depends on the range scanning device. Consider a scanner that projects an illuminant on one part of the surface, records the range, and moves the illuminant to the next sample position. If the illuminant projected onto the surface does not overlap with adjacent samples, then we would expect there to be no statistical correlation among samples. On the other hand, if the illuminant does overlap at different positions, then we might expect some correlations. For a well designed scanner, the illuminant is usually quite narrow, and/or the sensor lines of sight have little overlap (held tightly in focus), so that any dependence among samples is highly local, restricted to nearby samples. In this case, we can regard the assumption of statistical independence as an approximation to the more exact solution that accounts for local sample interdependence. Still, scanners are likely to have some element of uncertainty that *is* independent among samples. Thermal noise in electronic circuitry, for example, will exist regardless of the proximity of samples, leading to some independent range errors.

Range uncertainties are purely directional and are aligned along the lines of sight of the sensor which are known to high accuracy. To some extent, this statement may seem to be a tautology; we have defined a range imaging sensor as a device that returns depth values along specified lines of sight. When the sensor errs, it returns an erroneous *depth*, which must be along the line of sight of the sensor. In the previous chapters, we discussed the errors in optical triangulation, and these errors manifested themselves along either the lines of sight of the illuminant or the imaging device, depending what kind of analysis was used in determining depth. However, a sensor may also have some uncertainty in its lines of sight. For example, when performing traditional optical triangulation, if a sweeping illuminant has some "wobble" in its path, then its lines of sight may not be consistent. This inconsistency would lead to errors in depth along the line of sight, as well as errors that vary side to side, perpendicular to the line of sight. The error might then take an elliptical form, rather than a purely directional form. Nonetheless, the devices that control motion can be very precise, and sensors such as CCD's have been shown to exhibit excellent geometric stability. As a result, range imaging sensors can be designed to have excellent accuracy in the lines of sight, leaving depth uncertainty as the dominant error in the system.

Errors in alignment are much smaller than errors in range samples. The problem of aligning range images has received a fair amount of attention, and the results indicate that a number of algorithms can achieve accurate alignment among pairs of range images [Gagnon et al. 1994] [Turk & Levoy 1994]. Figure 5.7 shows the results of performing alignment with one of these pairwise methods. The problem of aligning a set of range images simultaneously, however, requires new alignment algorithms. Consider a set of range images taken from 8 points on the compass around an object. By aligning the range images in a pairwise fashion, we can walk around the object. But by the time we have come full circle, we find that the alignment errors have accumulated, and the first and last range images are now significantly misaligned. A better approach is to define an error function that relates all range images to one another, and then minimize this function all at once. Only recently are solutions to this total alignment problem emerging [Bergevin et al. 1996] [Stoddart & Hilton 1996]. Nonetheless, through a combination of good initial alignment and some heuristics for choosing how to perform the pairwise alignments, researchers have demonstrated that it is still possible to obtain reasonably accurate results. Our results, which we discuss in Chapter 8, show that the average distance between range samples and their projections onto the reconstructed surface are roughly the same as the uncertainty in the individual samples. This fact indicates that range errors dominate the alignment errors.

The range images yield viable range surfaces. While many range scanners generate a dense grid of depth samples, the analysis in this chapter requires the use of (piecewise) continuous surfaces. As described earlier, we can use the depth samples to reconstruct the desired range surfaces. In the case of the spacetime analysis described in the previous chapters, continuous reconstruction of the spacetime volume can indeed lead directly to range surfaces[2]. If the surface being sampled is smooth relative to the sampling rate, then the resulting range surface will be a reasonable estimate of the object's shape. However, the apparent smoothness of a surface, changes with viewing direction. A surface viewed at a grazing angle will be sampled far less than a surface that is facing the sensor. The grazing angle surface may even be undersampled, yielding a questionable, possibly aliased reconstruction. To ameliorate this problem, researchers avoid reconstructing the range surface

[2] In practice, we perform the reconstruction at a fixed resolution, leading to range images that still must be converted to range surfaces.



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

Figure 5.7: Aligned range images. Two range images were taken 30 degrees apart with an optical triangulation scanner. (a) and (b) show the corresponding range surfaces. (c) shows a rendering of the two range surfaces after alignment; the surface from (b) is rendered with a darker (red) color. (d) shows a “blow-up” of the head of the model. Note that the range surfaces are so close as to be inter-penetrating across the surface. The RMS error in distances between nearest points on the two surfaces is on the order of the error in individual range samples, indicating that the majority of the uncertainty remains in sampling error rather than alignment. Note that in some areas of overlap in (c) and (d), one surface is in front of the other in large regions, suggesting correlated range errors. However, these regions are significantly larger than the neighborhoods spanned by the illumination of the range scanner, making it unlikely that correlation in range errors is the cause. We suspect these coherent overlap areas result from small miscalibrations in the range scanner.

in areas that are known to be at sufficiently grazing angles and downweight contributions from surfaces that are at moderately grazing angles. For example, when using the Gaussian model with weighted signed distances, we can lower the weights in accordance with the orientation of the surface with respect to the sensor. While this is not strictly consistent with the notion of variance in Gaussian statistics, it yields excellent results in practice, as we will show in the following chapters.

In addition, points on the range surface that do not coincide with range samples will not appear to be statistically independent, even if the range samples are. Consider a 2D world where our sensor has acquired two neighboring range samples. We can reconstruct the range surface by linearly interpolating between the two samples, i.e., by connecting them with a line segment. We may regard the two samples as statistically independent, but the points on the line segment adjoining them clearly depends on the original samples. While this is not consistent with the assumption that points on the range surfaces are statistically independent, the results do indicate that regarding them as independent is a reasonable approximation.

The range images are functions over the domain where a surface is sought. When computing the ML optimal surface, we assume that each range image is sampling the same surface from different points of view. However, an object is generally not a single-valued function with respect to any single point of view. If we trace a ray from the sensor and intersect it with the object, we will see the nearest point on the "front" side of the object, but if we imagine continuing the ray beyond the first intersection, it will eventually intersect the "back" side of the object. Two different range images will sample the front and back sides of the object, and clearly they are not seeing the same points. Care must be taken so that when determining the optimal surface, we do not presume that these two range images are measuring the same surface. In the next chapter, we discuss efforts to make sure that opposing surfaces are not treated as the same surface.

In the case of the least squares formulation, the range uncertainties obey Gaussian statistics. The development in the previous sections is suited to any statistical model for directional range uncertainty. Nonetheless, Gaussian statistics are appealing because of their simplicity; in our case, they lead to signed distance functionals. When might this apply? In some scanners, individual samples are too noisy, so the sampling is repeated along a single line of sight, and the results are averaged. Almost regardless of the statistics of the errors in the individual samples, averaging them together tends to make the statistics take on a Gaussian character. Indeed, the central limit theorem tells us that the distribution does indeed become "more Gaussian" with each additional sample included in the average. In addition, our studies of errors due to laser speckle in optical triangulation indicate that the error distributions actually do have a Gaussian appearance, as indicated by Figure 4.13.