A General Near-Zone Light Source Model and its Application to Computer Automated Reflector Design

M. W. Siegel[*] *and R. D. Stock*[**]

Keywords: near-zone light source model, computer automated reflector design

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
phone: (412) 268-8802
fax: (412) 621-5477
e-mail: mws+@isl1.ri.cmu.edu
e-mail: rstock+@sensor.ri.cmu.edu

[*] The Robotics Institute

[**] Department of Physics

ABSTRACT

The goal of this work is to develop a general light source model that is physically accurate, intuitively descriptive, computationally convenient, and applicable to real sources.

This paper solves two major problems in illumination optics: near-zone general extended light source characterization for accurate image rendering and raytracing, and computer-automated reflector design using the resulting near-zone model. The main approach is to combine measurements with Fourier analysis, using judiciously chosen coordinate systems and orthogonal fitting functions.

This approach has several advantages over standard raytracing: it provides for natural data compression and interpolation, it bypasses the problem of computing the radiance distribution of a real source by using actual pinhole CCD camera measurements, and it eliminates the computationally intensive ray-filament intersection problem by transforming the source into an equivalent nonuniform spherical radiator. A method for treating the occlusion of rays by the extended filament, with only spherical intersection calculations, is also discussed.

The application of the method is illustrated by the problem of designing a smooth specular reflector to cast a desired intensity distribution on a distant screen for a given source. The problem becomes a straighforward numerical optimization of the perturbations to a base reflector shape. An algorithm to provide a first guess for the perturbations based on the shape of the image perimeter is also described.

1. INTRODUCTION

1.1 Motivation

Raytracing[1, 2, 3, 4], the method of light transport favored for handling specular surfaces, suffers several serious shortcomings when applied to real-world problems. One such problem concerns the difficulty of ray and surface intersection calculations in ray tracing for design, rendering, etc; another is the difficulty of characterizing any real luminaire that one might want to include as an object in a ray traced scene. Previous models[5, 6] were therefore forced to make simplifying assumptions regarding light source shape and composition, often employed empirical or idealized relations, and provided no standard procedure for characterizing the light distribution of a given real source. Thus a physically correct and computationally convenient treatment of a real light source in the near-zone,[1] where the extended nature of the light source matters, represented a nearly intractable problem for standard methods.

Over the last few years the notion that the radiance distribution of an extended source could be projected onto a sphere surrounding the body[7, 8, 9] (rendering the intersection problem trivial), and stored as a set of coefficients from an expansion in orthogonal polynomials[10, 11] (solving the storage problem), began to appear. Levin's presentation of this idea was in the form of brief observation; the others that reported actually working on the idea tended to use inconvenient interpolation or data compression schemes. None of this early work addressed the problem of occlusion by the source of secondary reflected rays or of rays from other sources. Although the need for accurate near-zone lighting calculations has been demonstrated[12], the field of near-zone source characterization is in its infancy.

Accurate near-zone rendering with real sources could be applied not only to computer graphics, but also to the tedious problem of luminaire reflector design. The task of designing a reflector for a given light source to create a desired illumination pattern on a screen has attracted interest among luminaire designers in the automotive and architectural fields[13, 14]. As a concrete

[1] As a rule of thumb, the near-zone is defined by illumination engineers as the volume within five characteristic lengths of the source.

example, automotive headlamps must have illumination patterns that satisfy both aesthetic and regulatory requirements. Styling, space, and cost considerations add further constraints. Customer preference is also leading designers away from reliance on lenses; more and more the job must be done with the reflector alone. The design of satisfactory reflectors, however, is an expensive and time consuming process, consisting of iteratively raytracing, prototyping (using libraries of standard lens and/or reflector components in various combinations), and measuring: i.e., trial and error. This technique leads to the astonishing fact that headlights require the longest development time of any automobile component; according to discussions with industry experts, the process can require up to four years!

Currently, computers have been of little help in this field, because most models for real light sources lack prescriptions for physically characterizing the true radiance distribution of the source, and rely on empirical or *ad hoc* relations. In addition, expensive filament and/or reflector intersection problems often lead to inefficient ray tracing. Near-zone source characteristics are usually ignored, as such models have only recently begun to appear[12, 10, 11, 9]. For instance, a technique developed by researchers at Mitsubishi Motors treats the source as a set of radiating pieces, and finds a 2-dimensional reflector by "growing" reflector segments with an orientation that produces the best match to a desired illumination pattern[13]. This is essentially the process currently carried out by human designers[15].

Our goal then is to save time and money by developing a combined near-zone light source and reflector model which can provide accurate computer-simulated rendering[2] to minimize the need for actual prototypes, giving the designer a valuable new tool. It may also be part of a computer-driven optimization system,[3] the result of which could be a starting point for fine adjustment by a human expert, perhaps reducing days or weeks of analysis and design to a few hours.

[2] computer-*aided* design

[3] computer-*automated* design

1.2 Overview of Approach

The application of a computer to the problem of reflector design requires (1) a near-zone light source model, (2) a light transport method, (3) a reflector model, and (4) a figure of merit. With these four elements, the reflector design problem in principle becomes one of optimization, which can be solved by a straightforward search of the reflector model parameter space by, for example, steepest descent, conjugate variables, conjugate gradient, simulated annealing, or genetic algorithms.

Because real light sources (extended bodies with nonuniform radiance distributions) still lack a physics-based near-zone model that is descriptive, practical, and general, we first develop such a model by providing a prescription for combining actual measurements with expansions in orthogonal functions. This model is then applied to the problem of designing a pseudo-parabolic reflector for a real light source to create a desired illumination pattern on a screen.

Operationally the model depends on measurements taken at points on a sphere surrounding the light source, using a pinhole camera with a charge-coupled device (CCD) on the image plane to obtain a series of 2-dimensional images for the extraction of the directional radiance information necessary for near-zone ray tracing. The measurements are then encoded as a set of Fourier coefficients of appropriate orthogonal functions.

The model is practical for several reasons. One is the ease with which raytracing calculations can be performed, because the light source is transformed by our procedure into a spherical source of secondary wavelets and because the fitting process allows interpolation between measurement points; the intersection problem, which can be expensive in standard ray tracing, becomes trivial. For example, the standard alternative would be to model the filament as a polygonal mesh, which requires measuring the filament's exact location in space. Another reason is in terms of computer memory requirements: intensity measurements (at the density with which we record the data) can be encoded by coefficients at a ratio of roughly a thousand to one, unlike typical ray tracing models which would demand a large number of points or polygons to be used.

The model is general because it is based on measurement. It is therefore applicable to any type of light source, unlike standard physics-based or computer graphics-based models which are forced to make simplifying assumptions regarding light source shape and composition, often employ *ad hoc* or empirical relations, and provide no standard procedure for characterizing the radiance distribution of a given real source. Our model provides a common language for conveniently quantifying the illumination patterns and radiance distributions of real light sources, regardless of their nature.

Using this generally applicable measurement-based near-zone light source model, we can formulate the reflector design problem in a way that is simultaneously computationally efficient and physically correct. The reflectors considered here are smooth paraboloids of revolution to which a few physically-based higher order perturbations are added; thus the number of terms to be optimized is greatly reduced compared with facetted mirror models. No restrictions are placed on the light source; real, extended sources can be correctly treated even in the near-zone.

Because of prevailing design and manufacturing techniques, most reflector models consist of a set of facets, which means a very large number of parameters is required to specify the surface, and the extra problem of enforcing smoothness arises. Because there are too many parameters in such a model for standard optimization algorithms, Ashdown outlines how genetic algorithms may be used for computer-automated reflector design[14]. The difficulties of standard reflector models are solved in our approach by eliminating the use of facets in favor of a polynomial representation, reducing the number of parameters to a dozen or so and insuring smoothness. Though local control of the surface is lost, the simplicity gained allows the use of standard numerical optimization techniques. Furthermore, the reflector surface coefficients have understandable physical meanings, such as astigmatism and focal length. Thus, adjustments can be made by a human designer with a firmer intellectual basis than the simple trial and error practice of manipulating a vast array of facets.

In summary, we provide a compact method for real source characterization in both the far- and near-zones, and combine the computationally practical near-zone model with a smooth reflector representation to facilitate both computer-aided and computer-automated reflector design. The source model can also be applied to accurate rendering in computer graphics.

2. A GENERAL NEAR-ZONE LIGHT SOURCE MODEL

2.1 Radiance From Pinhole Camera Measurements

The source model begins with data acquired by using a goniometer to scan the measurement sensor over the surface of a sphere that encompasses the radiant source.

Goniometer-based measurements have previously been used in computer graphics for simulating extended light sources[8]. These measurements have been of radiant intensity integrated over incoming directions. They are thus useless for raytracing *in the near-zone*:[4] the light flux is not entirely radial, so the radiance distribution is a function of four variables (three for position, two for direction of view, and less one for the constancy constraint along lines[16]) and the scaling of intensity is therefore **not** $1/r^2$. In fact, in the near-zone of a real light source there is really no unique "center" (or line, plane, etc) from which to measure a distance with which radiant intensity might scale!

To correctly model extended sources in the near-zone we need measurements with additional detail: we need to measure the differential intensity per unit directed solid angle, not just the integral intensity over all directions. We obtain the required differential data from the goniometer by mounting a CCD pinhole camera on it, placing the source near the center of the goniometer, and collecting images. The pinhole defines latitudes and longitudes on the sphere, each pixel-pinhole pair defines a direction, and the gray level reported by each pixel is proportional to the radiant intensity transmitted by the pinhole into the solid angle subtended by that pixel as seen from the pinhole. Each pixel-pinhole pair thus acts as a miniature radiance meter, and the many pixels effectively allow many measurements to be taken at once.

An Electrim EDC-1000 digitally controlled monochrome CCD camera was used to take the filament images. This camera has pixels of 13.75 µm by 16 µm in an array of 192 by 165, giving a sensing area of 2.64 mm by 2.64 mm. Five pinholes were obtained from Edmund Scientific, with diameters specified only as being in the ranges 5-12 µm, 20-35 µm, 40-70 µm,

[4] They are useful in the far zone.

85-120 μm, and 200-300 μm. The 40-70 μm pinhole produced the least image crosshatching, [5] so it was used for all subsequent image gathering. To reduce blooming, an infrared blocking filter and an 0.8 neutral density filter were also placed in front of the pinhole. Measurements were typically taken at goniometer radii between 2.22 and 5.22 inches.

Four different coiled-filament headlamps[6] were investigated. They are a horizontal filament "spot quality" lamp surrounded by a carefully manufactured nearly spherical glass bulb; a vertical axial filament "high beam" lamp with a cylindrical bulb that becomes conical on top; a vertical filament axial "low beam" lamp with a cylindrical bulb and a light-blocking cap on top; and a "dual beam" lamp with two horizontal filaments for high and low beams, with a cap on top. The spot quality bulb is used in the manufacturer's reflector design process to evaluate reflector prototypes, to hand finish dies, etc. The other three bulbs are standard headlamp bulbs, with part numbers 9005, 9006, and 9004 respectively (see appendix V for photographs). The pinhole camera image of a spot quality light bulb filament, shown as a negative for clarity, is typically a "blob" like the one shown in figure 2-1.

Figure 2-1: a "blob" (a pinhole CCD camera negative)

Because an aperture uniformly illuminated in a given direction by a radiance *B* from one side acts just like an area source of the same radiance with respect to objects on the other side, a

[5] The term "crosshatching" here refers to lines or distortions in the image. This determination was performed by eye.

[6] Provided by a major automotive lighting supplier.

single pinhole camera picture can measure the radiance distribution through a small patch of area corresponding in location, size, and orientation with the camera's aperture. Many such pictures can be used to characterize the full radiance distribution of the source over some closed (conveniently spherical) surface. The entire dataset of all the pinhole camera pictures taken over the measurement sphere is shown in figure 2-2. A full dataset of the low beam in a dual filament bulb is shown in figure 2-3 for comparison. Obviously, it is important to adjust the geometry to ensure that each blob fits entirely within the CCD array.

If, as in figure 2-4, the aperture is placed on a sphere at coordinates (r_0,θ_0,ϕ_0), the radiance B passing through this point in the direction of pixel (x_0,y_0) is

$$B(r_0,\theta_0,\phi_0;x_0,y_0) = I(r_0,\theta_0,\phi_0;x_0,y_0)\, f^2 / (A_p A_q \cos^4\beta)$$

Here I is the power measured at the pixel, f is the distance to the image plane of the camera, β is the angle of incidence, and A_p and A_q are the pixel and aperture areas, respectively. Two cosine factors appear for the projected areas, and two are due to the squared pinhole-to-aperture distance.

Ashdown has outlined a similar data gathering procedure[9]. He too records the angular distribution of light at multiple points on the sphere. However his methods of reducing, representing, and using the data in downstream computations appear to be less developed than ours.

2.2 Data Analysis and Representation

2.2.1 The Harmonic Oscillator Wavefunction Expansion

First we decompose each 2-dimensional image into a multi-parameter representation in which the parameters are intimately connected with our intuitive perception about how best to convey the description of a "blob":

- *Parameters x_{00} and y_{00}:* the x and y coordinates of the center-of-intensity;
- *Parameters σ_x and σ_y:* the standard deviations of the intensity distributions in a frame centered on the center-of-intensity and oriented along and perpendicular to the tilt-axis;
- *Parameter a_{00}:* the tilt-angle, with respect to the x-axis, of the elliptical cross-section of the bi-Gaussian that "best-fits" the blob, derived analytically from a

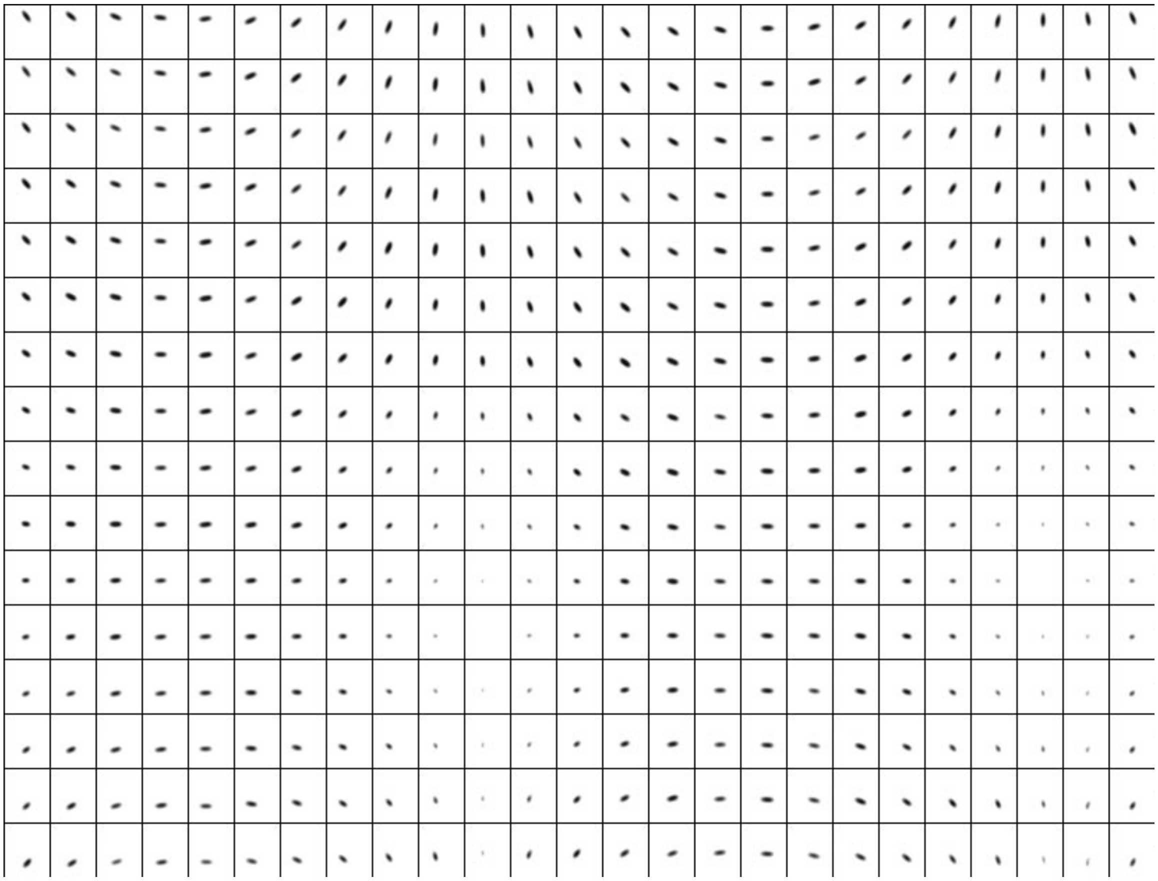

Figure 2-2: a set of spot quality "blobs" (vertical θ, horizontal ϕ)

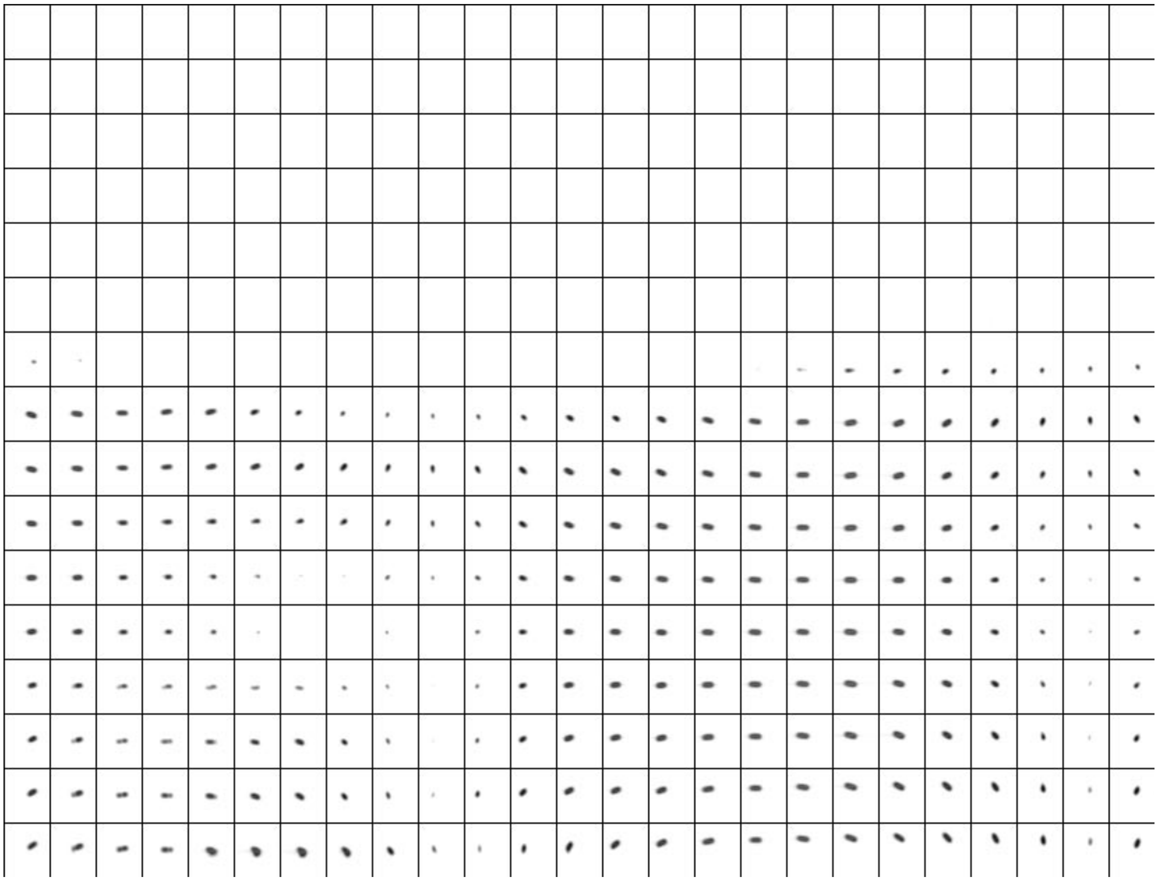

Figure 2-3: a set of dual low "blobs" (vertical θ, horizontal ϕ)

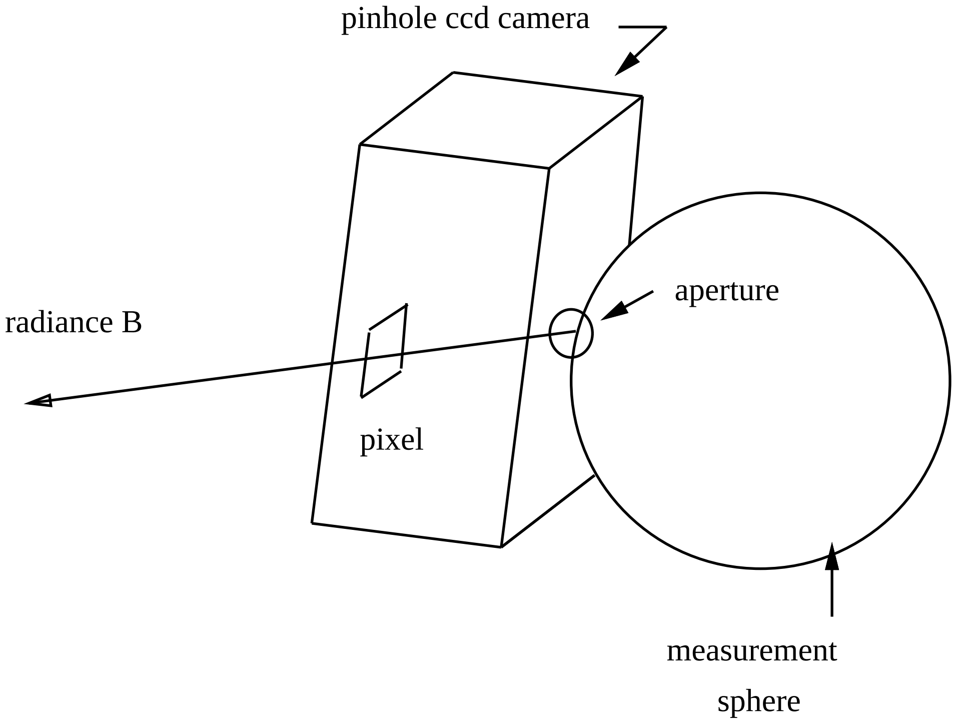

Figure 2-4: illumination of an aperture

function of σ_x, σ_y, and σ_{xy} (see appendix I);

- *Higher Parameters m_{ij}:* coefficients of the Fourier expansion of the intensity distributions along and perpendicular to the tilt axis, using basis functions well known in quantum mechanics as "the wavefunctions of the harmonic oscillator" (they are not known by, e.g., a person's name), scaled by parameters σ_x and σ_y respectively.

The harmonic oscillator wavefunctions are an orthonormal set on the interval $-\infty$ to $+\infty$. They are products of a Gaussian with a Hermite polynomial scaled to the standard deviations of the bi-Gaussian that best-fits the blob. Appendix I reviews the mathematical details.

The image taken at camera pinhole position co-latitude θ and longitude ϕ thus has the intensity distribution

$$I[\theta,\phi;x,y] = \sum_{ij} m_{ij}[\theta,\phi]\,\Phi_i[x]\,\Phi_j[y]$$

where $[x,y]$ is a position, in units of σ_x and σ_y, with respect to the center-of-intensity $[x_{00},y_{00}]$, along axes tilted by a_{00} from horizontal.

We typically include terms in i and j from zero to six; higher order terms tend to "fit the noise" for our data rather than lend improved understanding of the underlying light distribution. For very regular spot quality filament images, a maximum index of four is often acceptable if regeneration speed is an issue.

The effectiveness of this parameterization is illustrated by figure 2-5, which is the image shown in figure 2-1 "regenerated" from its 2D parametric representation in the image plane. Note how the noise (consisting of the dots at 2 o'clock and 4 o'clock about 2 major diameters from the center of blob 2-1) due to bad pixels has been "smoothed" away. The *rms* fitting error (over pixels) for the regenerated image (figure 2-5) is 1.6 grayscales over a dynamic range of 245; this is well within the typical CCD pixel non-uniformity of 10% (according to the manufacturer's spec sheet).

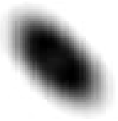

Figure 2-5: figure 2-1 regenerated from its 2D m_{ij} coefficients

2.2.2 Expanding the Expansion

By the preceeding method we obtain a set of coefficients for each picture taken over the sphere:

$$s[\theta,\phi] = \{x_{00}, y_{00}, a_{00}, \sigma_x, \sigma_y, m_{ij}\}[\theta,\phi] \quad \text{for } i,j = 0 \ldots mmax$$

So far we have concentrated on each image individually: θ and φ are just labels that identify a particular image. Now it is useful to think about labelling the parts of the data set in another way: recognizing that each element s_k of set s is a smoothly-varying function of position (θ,φ) on the sphere, we can represent each of these elements by another set of coefficients $\{\alpha^k_{lm}\}$ times some convenient set of orthogonal basis functions $\Psi_{lm}[\theta,\phi]$:

$$s_k[\theta,\phi] = \sum_{lm} \alpha^k_{lm} \Psi_{lm}[\theta,\phi]$$

We obtain excellent and efficiently computable fits when we choose the 2-dimensional basis functions $\Psi_{lm}[\theta,\phi]$ to be products of 1-dimensional trigonometric Fourier series[17]:

$$\alpha^k_{lm}\Psi_{lm}[\theta,\phi] = a^k_{lm}\sin[l\theta]\sin[m\phi] + b^k_{lm}\sin[l\theta]\cos[m\phi]$$
$$+ c^k_{lm}\cos[l\theta]\sin[m\phi] + d^k_{lm}\cos[l\theta]\cos[m\phi]$$

The set of coefficients $\{\alpha^k_{lm}\}$ is shorthand for the larger set $\{a^k_{lm}, b^k_{lm}, c^k_{lm}, d^k_{lm}\}$.[7]The coefficients are found by the standard Fourier integral on a discrete space.

[7] Spherical Harmonics would be inappropriate here because they lose their φ dependence at the poles, where the pictures still vary with φ.

The effectiveness of this parameterization is illustrated by figure 2-6, which is the image shown in figure 2-1 "regenerated" from its 4D parametric representation over the sphere. The *rms* fitting error (over pixels) for the regenerated image (figure 2-6) is 9.2 grayscales over a dynamic range of 245, again within the typical CCD pixel non-uniformity of 10%.

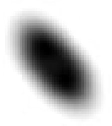

Figure 2-6: figure 2-1 regenerated from its 4D α_{lm}^{k} coefficients

Thus the dataset over the sphere is treated as if it were on a rectangular grid, and the data interval in θ is mapped to the range 0 to 2π.[8] The number of θ coefficients $(2m+1)$ and ϕ coefficients $(2l+1)$ can be tuned to approximate the ratio of the number of θ data points to the number of ϕ data points.

For efficiency in regeneration, following an observation by Acton[18], the trigonometric functions of multiple angles should not be calculated as function calls, but from the three-term multiple angle recurrence relation. Thus, for example, after merely setting $\cos(0)$ to one and making a single function call to evaluate $\cos(\theta)$, all higher orders can be calculated recursively with a single multiplication and subtraction from the formula

$$\cos(n\alpha) = 2\cos(\alpha)\cos((n-1)\alpha) - \cos((n-2)\alpha)$$

[8] It is actually mapped to one stepsize short of the full range because of possible data discontinuity in θ.

2.2.3 Compression Factor for Blob Images

Five blob-locating parameters and forty-nine m_{ij} coefficients were needed to represent the 196x165 pixel image of figure 2-1. Thus, the representation of a single image was successfully changed from 32,340 one-byte integers to 54 four-byte floats, for a compression ratio of about 150 to 1. For many (typically 425) images over a sphere, each of the 54 blob parameters can be stored by the $\{\alpha^k_{lm}\}$ with reasonable accuracy at compression ratios of five or ten to one. Therefore a compression ratio approaching three orders of magnitude can be obtained for the entire set of images over the measurement sphere.

2.2.4 Complete Representation and Regeneration

Assuming non-pathological behavior (which experimentally is the case), the intensity at any hypothetical pixel (x,y) in a pinhole camera located at any hypothetical point (θ,ϕ) on the measurement sphere can then be represented (and interpolated) via the two-step process:

$$s_k(\theta,\phi) = \sum_{lm} \alpha^k_{lm} \Psi_{lm}(\theta,\phi)$$

$$I(\theta,\phi;x,y) = \sum_{ij} m_{ij}(\theta,\phi)\,\Phi_i(x)\,\Phi_j(y)$$

where again (x,y) refers to a coordinate system described by the regenerated s_k parameters described in section 2.2.1.

With this represention entire pinhole camera pictures can be reconstructed at *and interpolated between* latitudes and longitudes at which pictures were actually taken. Figures 2-7 and 2-9 show the pictures (real data from figure 2-2) of the spot quality bulb filament taken at co-latitude $\theta = 60$ grads at two values of longitude ϕ separated by 16 grads. Figure 2-8 shows the picture computed (interpolated) midway between them via the coefficients $\{\alpha^k_{lm}\}$.

Figure 2-7: actual image at (60,-8) grads

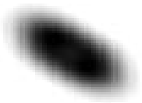

Figure 2-8: interpolated image at (60,0) grads

Figure 2-9: actual image at (60,8) grads

2.3 Light Transport

To transport light, the power incident on any distant surface patch can be found by sampling small solid angles in many directions from the surface, checking for intersection with the measurement sphere, regenerating the radiance of the source, and re-converting the radiance to incident power. This calculation is possible because the radiance is constant along lines[16], so the regenerated radiance emanating from the source is the same as that incident upon the receiving surface from any particular direction.

2.3.1 Efficient Sampling Limits

Knowledge of the size and location of the measurement sphere (or spheres), however, allows the sampling to be efficiently restricted to the geometrical projection of the measurement sphere onto the image plane of the sampling hemicube (figure 2-10).

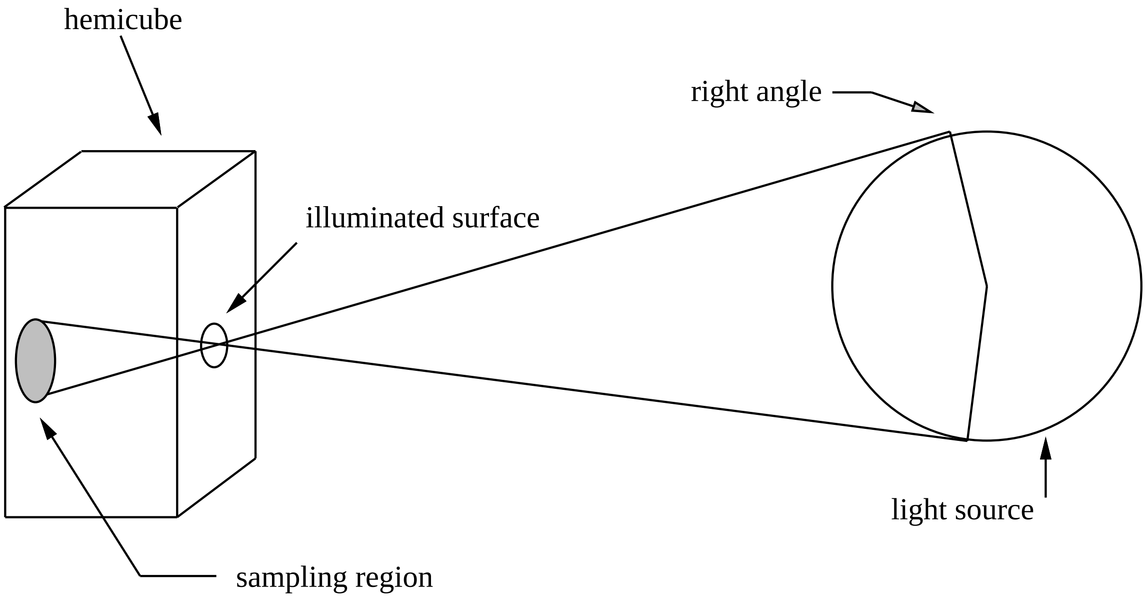

Figure 2-10: efficient sampling limits

2.3.2 Self-Occlusion

The occlusion of rays by light-emitting portions of the source can easily be checked within the framework of this model. If at some point on a surface (over which the radiance distribution is known) and in some direction the radiance is non-zero, then the line through that point, in that direction, *must intersect* a light-emitting part of the source with that radiance. Conversely, if at some point, in some direction, the radiance is zero, then the line through that point, in that direction, *does not intersect* a light-emitting part of the source. We assume, of course, the source is a simple emitter that does not have, for example, a frosted or refractive glass envelope.

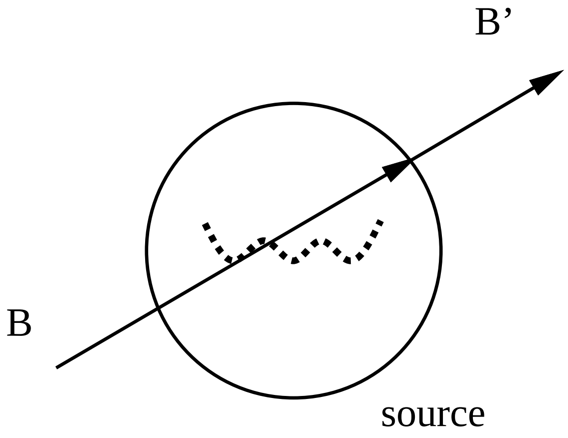

Figure 2-11: occlusion

As shown in figure 2-11, the exit position and direction coordinates of a ray *B* that intersects the measurement sphere surrounding the source can easily be calculated by solving a quadratic, and the source's radiance *B′* at these coordinates can be computed from the coefficients $\{\alpha^k_{lm}\}$. If *B′* is nonzero, then there must exist a light-emitting surface within the measurement sphere that lies in the path of the ray, thereby occluding that ray. If the radiance is zero, then the ray will not be occluded by any light-emitting portion of the source.[9] The problem of intersecting, say, a complicated coiled filament to test for occlusion, is thus rendered trivial by the measurement sphere concept.

[9] "Dark" portions of the source (posts, baffles, etc) may still block such a ray, so in this model there is no way around explicitly testing for intersections with these structures. Often they are small, however, and can reasonably be handled via highly simplified geometrical models that are easy to intersect. A series of pinhole camera measurements, with the filament turned off but backlit, might provide a way for describing the ray-blocking and transmitting portions of the source.

3. COMPUTER-AUTOMATED REFLECTOR DESIGN

3.1 Reflector Model

For illustration we present a relatively simple but nevertheless realistic parametric model of a reflector. The aim is to find values of the reflector parameters that, given the light source coefficients, produce (within tolerance) a desired illumination pattern on, for example, a distant wall. The reflector is assumed to be perfectly specular (this assumption is used for computational simplicity but is not required). The base shape, to which higher order perturbations are sought, is a parabola of revolution with unit focal length aligned along the z axis with its focal point at the origin and its apex at $z=-1$. The reflector is therefore represented by a polynomial expansion of the form:

$$\begin{aligned} z = {} & \frac{x^2+y^2}{4} - 1 + v + r(x^2+y^2)^{1/2} + l_1 x + l_2 y \\ & + a_1(x^2-y^2) + a_2 xy + c_1 x^3 + c_2 y^3 + c_3 x^2 y + c_4 x y^2 \\ & + q_1 x^4 + q_2 x^3 y + q_3 x^2 y^2 + q_4 x y^3 + q_5 y^4 \end{aligned}$$

Because x, y, and z are measured in units of the focal length of the base parabola, the perturbation coefficients are dimensionless.

These perturbations represent

$v \rightarrow$ vertical offset
$r \rightarrow$ conical term
$l_1, l_2 \rightarrow$ linear perturbations
$a_1, a_2 \rightarrow$ astigmatism terms
$c_1, c_2, c_3, c_4 \rightarrow$ cubic perturbations
$q_1, q_2, q_3, q_4, q_5 \rightarrow$ quartic perturbations

These coefficients have understandable physical meanings. The offset, for example, is like the "object distance" in the Newtonian lens equation, and thus controls the convergence or divergence of the rays. The linear terms shift the paraboloid both horizontally and vertically, without actually changing its shape, causing a shift in the transverse position of the image.[10] The three terms v, l_1, and l_2 thus control the horizontal and vertical position of the paraboloid.

[10] The use of linear perturbations to a paraboloid to aim the light has appeared in the literature[19].

The terms r, c_1 and c_2 have effects qualitatively similar to the offset and linear terms respectively, but with attendant changes in reflector shape. Each mixed cubic term also acts like linear term, shifting the image, but along a pair of axes crossing at approximately 53°. The astigmatism term a_1 stretches or compresses the image along the x axis, and does the reverse to the image along the y axis. The other astigmatism term does the same, but along axes 45° to the coordinate axes of the image. The quartic terms q_1 and q_5 stretch or compress the image sharply along the coordinate axes. Higher-order terms may be added as necessary, but the optimization will become slower and more local minima may appear.

The representation of the reflector surface as a polynomial expansion has several additional advantages, including enforcing smoothness and giving the surface normal for any (x,y) by an analytic derivative. The automated machining tools that would fabricate the mirror could therefore easily interpret this model (but terms like z^n would generally be ruled out as $z(x,y)$ would not generally be known analytically). Furthermore, the smooth model, as opposed to the standard facetted approach, greatly reduces the number of terms to be optimized. It does, however, require foregoing purely local control over the light. The computer-generated mirror, however, can be used as a good starting point for an expert's "tweaking".

We usually assume f (the focal length of the base-shape paraboloid reflector) is given, and do not allow it to vary (although it is possible to let the search algorithm perturb it if so desired). There are two main reasons for this approach, both related to the observation that f sets the scale of the problem. The first reason is that it is natural to measure all distances in units with $f=1$; a change of f would suddenly change the meaning of all the other perturbations, resulting in a large jump in perturbation space and a less than systematic search. The second reason is that the size of the reflector is often constrained, and f is the single most important parameter in determining the overall size. It should be reasonable, therefore, to think of f as a given constant.

Finally, a design rule must be used to truncate the reflector. For this example we arbitrarily chose to use a rule that truncates the reflector above $z=0$. Other rules based on overall width, depth, or both may also of course be used, depending on the problem at hand.

3.2 Image Rendering

The ray tracing space for this problem consists of three objects: the equivalent spherical source, the pseudo parabolic reflector, and the image plane. Because both the image plane and the source sphere are easy to intersect (algebraic calculation versus search), it is computationally feasible to represent the reflector surface $z(x,y)$ as a polynomial expansion as explained in section 3.1. Normally such an expression would be avoided within a raytracing framework because the intersection problem would require solving for the zeros of a high order polynomial; the standard approach is to treat the reflecting surface as a set of easily intersectible facets. With our spherical source model, however, we are free to parametrically step over the surface (by sampling points in x and y), and to cast specularly-related pairs of rays toward the easily-intersectible image plane and source sphere. Thus we use a combination of forward and backward raytracing, in which the entire reflector surface and the entire image plane are sampled.

The procedure for building up an image for a given light source is performed by parametrically stepping over the reflector surface by taking equal steps in x and y, and using the current coefficients to find the height $z(x,y)$ and normal $\boldsymbol{n}$ of each corresponding planar differential reflector element. From this reflector point, a ray $\boldsymbol{r}_1$ is cast to each pixel (x',y') on the image plane $z=D$. For each such ray, the corresponding specularly-reflected ray $\boldsymbol{r}_2$ is checked for intersection with the spherical light source (figure 3-1).

For a set of reflector coefficients, the pseudocode for the rendering algorithm is:

```
• step over reflector plane
      • calculate reflector element position, orientation, and area
      • step over image plane
            • send ray from image to reflector
            • find specular ray $S$
            • if $S$ intersects measurement sphere:
                  • regenerate radiance along $S$
                  • calculate image pixel power
```

To decrease the computation time, the rendering can be performed on a regular grid for only a

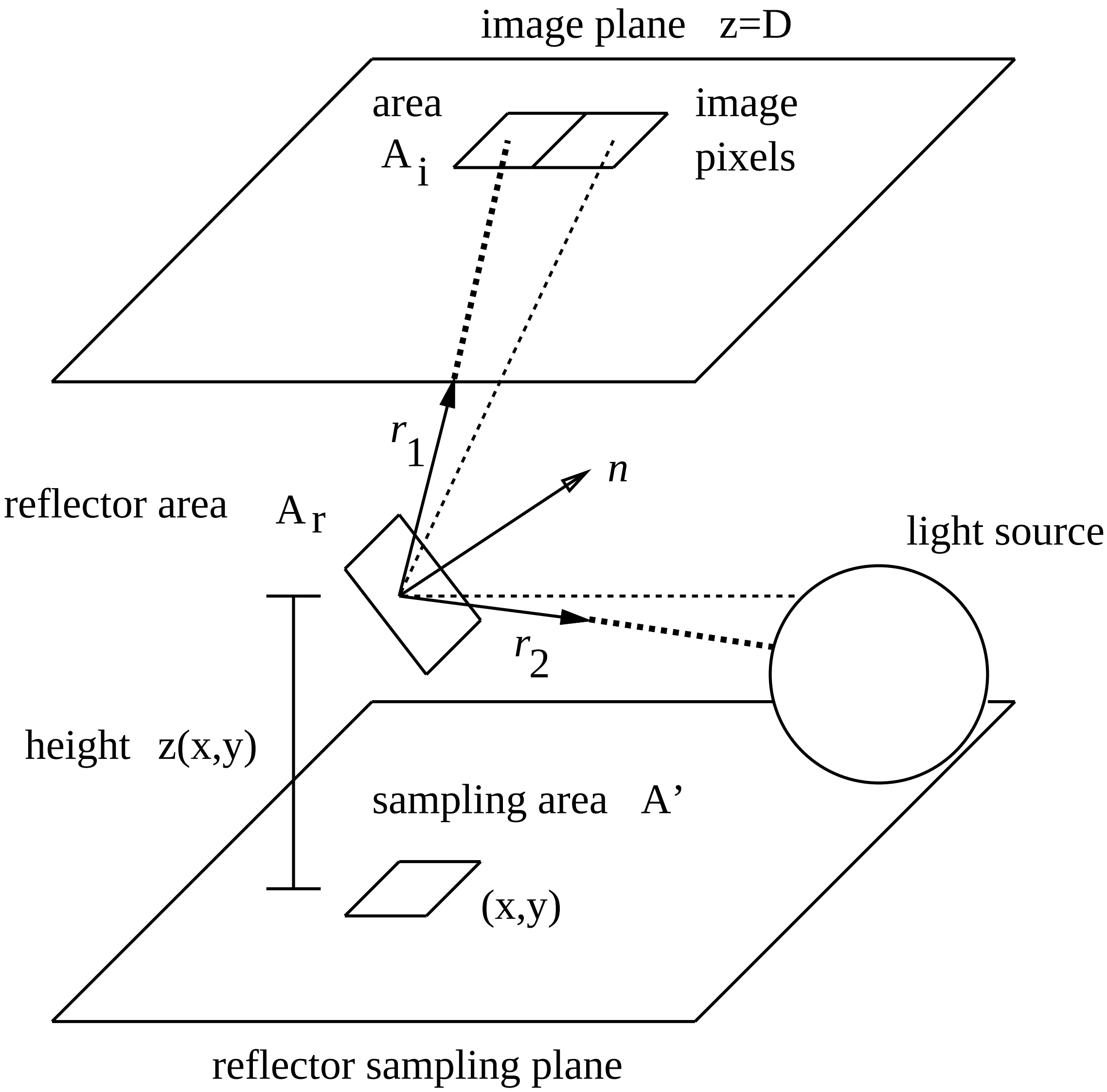

Figure 3-1: sampling the reflector surface

fraction of the total image pixels (typically 1/4 or 1/9), with the remaining intensity values filled in by linear interpolation. Because the time required for the linear interpolation is negligible compared with actual rendering, the speed-up is often well worth the slight decrease in accuracy. As a locally optimal solution for the mirror coefficients is approached, the rendering density can be increased to fine-tune the coefficients, but this has not been found to appreciably change the solution.

3.2.1 Comparing Two Images

The resulting image can be built up pixel-by-pixel this way and stored as an array of intensity values. For compactness and generality, the image can alternatively be represented by its Fourier coefficients, for example, with harmonic oscillator wave functions used as the basis functions in an appropriate coordinate system, as described in section 2.2 and appendix I. This representation also provides a robust method of quantifying the "closeness" of two images.

The desired target image is represented as a set of (typically 49) coefficients $\{m_{ij}\}$, plus five additional parameters to specify the coordinate system: two to locate the origin at the "center of intensity" of the image (x_{00},y_{00}), two to set the units of length in the x and y directions equal to the standard deviations of the image (σ_x,σ_y), and one to tilt the x-axis to correspond with the major axis of the elliptical cross-section of the bi-Gaussian which best fits the image (a_{00}).

For a given reflector, the Fourier coefficients $\{m'_{ij}\}$ of the rendered image can be calculated incrementally without the need for ever storing a large array of image intensity values. The coefficients are built up by

$$m'_{ij} = \sum_{xy} \sum_{x'y'} \Delta I'(x',y';x,y)\, \Phi_i(x'/\sigma_x)\, \Phi_j(y'/\sigma_y)$$

where Φ is a harmonic oscillator wave function, and the same coordinate system as the desired image is used. Here x and y refer to the reflector sampling plane, and x' and y' refer to the shifted and rotated coordinates of the image plane.

All optimization procedures essentially iteratively vary the reflector coefficients, calculate test images, and replace the old reflector coefficients with those that minimize some measure of "closeness" between the test image and the desired image. For two image arrays I_{xy} and I'_{xy}, the squared residual is defined as (dropping the primes on the image plane coordinates *x* and *y*):[11]

$$\Delta^2 = \int\int_{-\infty}^{+\infty} dxdy\,(I_{xy} - I'_{xy})^2$$

Thus it appears as if the desired image must be "unpacked" from its $\{m_{ij}\}$ representation for comparison with the current image, which is a memory intensive and time-consuming process.

An alternative definition of Δ^2 is possible, however. If the image I' is also characterized by coefficients $\{m'_{ij}\}$, using the same coordinate system parameters as the image *I*, then orthogonality among the basis functions results in

$$\begin{aligned}\Delta^2 &= \int\int_{-\infty}^{+\infty} dxdy\,\{\sum_{ij} m_{ij}\,\Phi_i(x/\sigma_x)\,\Phi_j(y/\sigma_y) - \sum_{ij} m'_{ij}\,\Phi_i(x/\sigma_x)\,\Phi_j(y/\sigma_y)\}^2 \\ &= \sigma_x\sigma_y \int\int_{-\infty}^{+\infty} dxdy\,\{\sum_{ij} (m_{ij} - m'_{ij})\,\Phi_i(x)\,\Phi_j(y)\}^2 \\ &= \sigma_x\sigma_y \sum_{ijkl} (m_{ij} - m'_{ij})\,(m_{kl} - m'_{kl}) \int_{-\infty}^{+\infty} dx\,\Phi_i(x)\,\Phi_k(x) \int_{-\infty}^{+\infty} dy\,\Phi_j(y)\,\Phi_l(y) \\ &= \sigma_x\sigma_y \sum_{ijkl} (m_{ij} - m'_{ij})\,(m_{kl} - m'_{kl})\,\delta_{ik}\,\delta_{jl}\end{aligned}$$

leaving

$$\Delta^2 = \sigma_x\sigma_y \sum_{ij} (m_{ij} - m'_{ij})^2$$

Dividing by $\sigma_x\sigma_y$, a measure of the characteristic size of the image, and taking the square root, results in the normalized "root mean square" quantity

$$R = (\sum_{ij} (m_{ij} - m'_{ij})^2\,)^{1/2}$$

Because the size has been divided out, *R* can be used as a standard, size-independent characterization of the closeness between two images. We have found this coefficient-based method to be more numerically robust than the pixel-based method.

Once the Fourier coefficients are found, the work required to compare two compressed images is greatly reduced; for a 200x200 pixel image represented by 49 coefficients, for example, the number of squaring operations required to calculate *R* is decreased by a factor of 800. The

[11] Because the filament image is completely contained within the camera's field of view, the intensity outside of the camera's boundaries can be set to zero, allowing the integral to proceed to infinity.

Fourier representation also has the desireable property of spreading out any noise (or outlier intensity values) localized in individual pixels over the entire image.

3.3 Automated Optimization Procedure

Working within this new model, we develop an algorithm for automated reflector design. The user supplies the position of the measurement sphere for a light source characterized by the α_{lm}^{k} coefficients, a desired illumination pattern on a screen a distance D from the light source, and a base focal length f for the initially parabolic mirror. The origin of the reflector's xyz coordinate system is chosen to coincide with the center of the light source's measurement sphere.

A judicious choice of coordinate axes with which to represent the reflector can minimize the final values of the perturbation coefficients and hence reduce the solution time. For instance, the linear perturbation terms to the reflector base shape which are responsible for pointing the image can be minimized by requiring the z axis to tilt from the vertical in such a way as to pass through the "center of intensity" of the desired image. Furthermore, the astigmatism terms can be "diagonalized" if the x and y axes are rotated such that their projections onto the image plane in the direction of the z axis are parallel to the major and minor axes of the ellipse that best fits the desired image; in this way the "cross-astigmatism" term a_2 will become small.

The x and y reflector axes are first rotated about the vertical by a_{00}, and then appropriate rotations of the coordinate system about these two axes are made to align the z axis of the paraboloid with the point (x_{00}, y_{00}) on the image plane, which describes the location of the center-of-intensity of the desired image (see section 2.2 and appendix I).

For any given set of reflector coefficients, an image can be rendered. The "closeness" of this image to the desired image can then be measured. This measure of closeness R (section 3.2.1) is then a numerical function of the reflector coefficients. The problem has now been cast as a straightforward exercise in numerical optimization[20].

Any reasonable method that samples the reflector parameter space (such as steepest descent, simulated annealing, multidimensional binary search, or genetic algorithms) will suffice. All optimization/minimization methods, however, must overcome two major obstacles: the presence of narrow, twisting ravines (or ridges), and local minima, especially for residual functions such as this one which consist of several parameters and contain no analytical derivative information. The lack of analytic knowledge of the slope of the residual function unfortunately excludes many of the most efficient minimization algorithms.

Simulated annealing robustly deals with these problems but can be time consuming. The gradient-based steepest descent method often overshoots the minimum or misses twists in ravines, and convergence is slow near a minimum; furthermore, derivatives are required. The multidimensional binary search[18], or "star search", is single-minded but inefficient. Ashdown has described the application of genetic algorithms to this problem, using Bezier and B-splines surfaces to model the reflector surface[14]. We choose to use Brent’s modification[21, 22] to Powell’s improvement[23] of the "conjugate variables" approach.

The conjugate variables method is essentially a multidimensional binary search without derivatives that is made more efficient by adaptively changing the search axes in order to be sure of finding the minimum of a quadratic form in a finite number of steps. Since all functions approach a quadratic form near a minimum, convergence should be rapid if a good initial guess is available. This optimization procedure is discussed more fully in appendix II.

Although conjugate variables provides a robust multidimensional minimization method when no derivatives are available, local rather than global minima can still be found. We deal with this remaining problem in several ways. The first is the realization that a true global minimum is not actually required; all we need is a minimum that is good enough to satisfy the design tolerance. Another strategy is to use more than one starting point in parameter space. Choosing an initial set of reflector coefficients that is close to the global minimum (or a "good enough" minimum) can also help; in appendix III we therefore present a "first guess" algorithm that chooses a reasonable set of starting parameters based on the shape of the perimeter of the desired pattern. The problems of efficiently searching a tortuous, numerically determined surface have thus been addressed.

3.4 Results - Simulated Source

Reflectors were designed by this process using a 15x15 mirror plane grid with a step size of 0.4 *f*, and a 51x51 image plane grid with a step size of 1.0 *f*. These numbers were chosen to avoid visually obvious sampling errors in the images. The distance to the image plane was arbitrarily taken as 320 *f*, and the light source measurement sphere radius as 0.5 *f*. In order to avoid introducing artifacts due to light source anisotropy, and to more easily understand the results, we used a simplified synthetic light source with the property that any radial pinhole camera picture on the measurement sphere would result in a bi-Gaussian intensity distribution (i.e. all of the m_{ij} were set to zero except for m_{00}). Coincidently, to the degree that a bi-Gaussian approximates cosine[4], the light source approximates a Lambertian radiating sphere.

Images (to become the target "desired" images) were created using this synthetic light source model so that the desired image size and intensity would be reasonable in terms of energy conservation for this source. The mirror used for creating the image was a vertical paraboloid with randomly chosen perturbation terms.

When numerically searching for a good minimum in the presence of several local minima, various starting points should be used for the search. Thus, each example will show the results for two different starting points: an unperturbed paraboloid that is automatically oriented to point at the center of intensity of the image as described in section 3.3, and a paraboloid oriented the same way but with the results of the "first guess" algorithm of appendix III used as initial perturbations.

As a general rule of thumb, we classify solutions with final residuals[12] less than 10.0 as absolute successes, between 10.0 and 20.0 as marginally satsifactory, and greater than 20.0 as clear failures (an *R* of about 10.0 corresponds to an *rms* error of about 2.0 greyscales).

Each iteration (minimizing in each of the coordinate directions) required approximately five minutes on a Silicon Graphics Indigo workstation running a 150 MHZ processor with a MIPS R4400 CPU, and included 30-35 renderings. The rendering was done on a regular grid

[12] The quantity *R* of section 3.2.1.

containing 1/9 of the pixels, with the rest filled in by linear interpolation.

3.4.1 A Tri-Lobed Example

An image with three lobes was created using six perturbations: a converging offset, a linear term, an astigmatism, two quartics, and a diverging conical term. This example should also be difficult because of possible confusion between the proper contributions of the offset and radial terms (both having similar light-steering properties) in the solution. Furthermore, the first guess algorithm is not equipped to guess the offset term, which is important in this example.

A summary of the residual for each of the two starting points is shown in figure 3-2. Without the guess the residual began at 119.5 and ended at 14, but with the guess the initial residual was 103.3 and ended at 4.9.

Figure 3-3 shows the initial image of the second starting point (tilted paraboloid with first guess algorithm) and figure 3-4 shows the best solution found. Figure 3-5 shows a plot of the absolute difference between the desired and best-fit images. Figures 3-6 and 3-7 show the desired and derived images side by side for comparison. Each contour line in this section represents a change in intensity of five greyscales, except for figure 3-5 where it represents 3.5 greyscales.

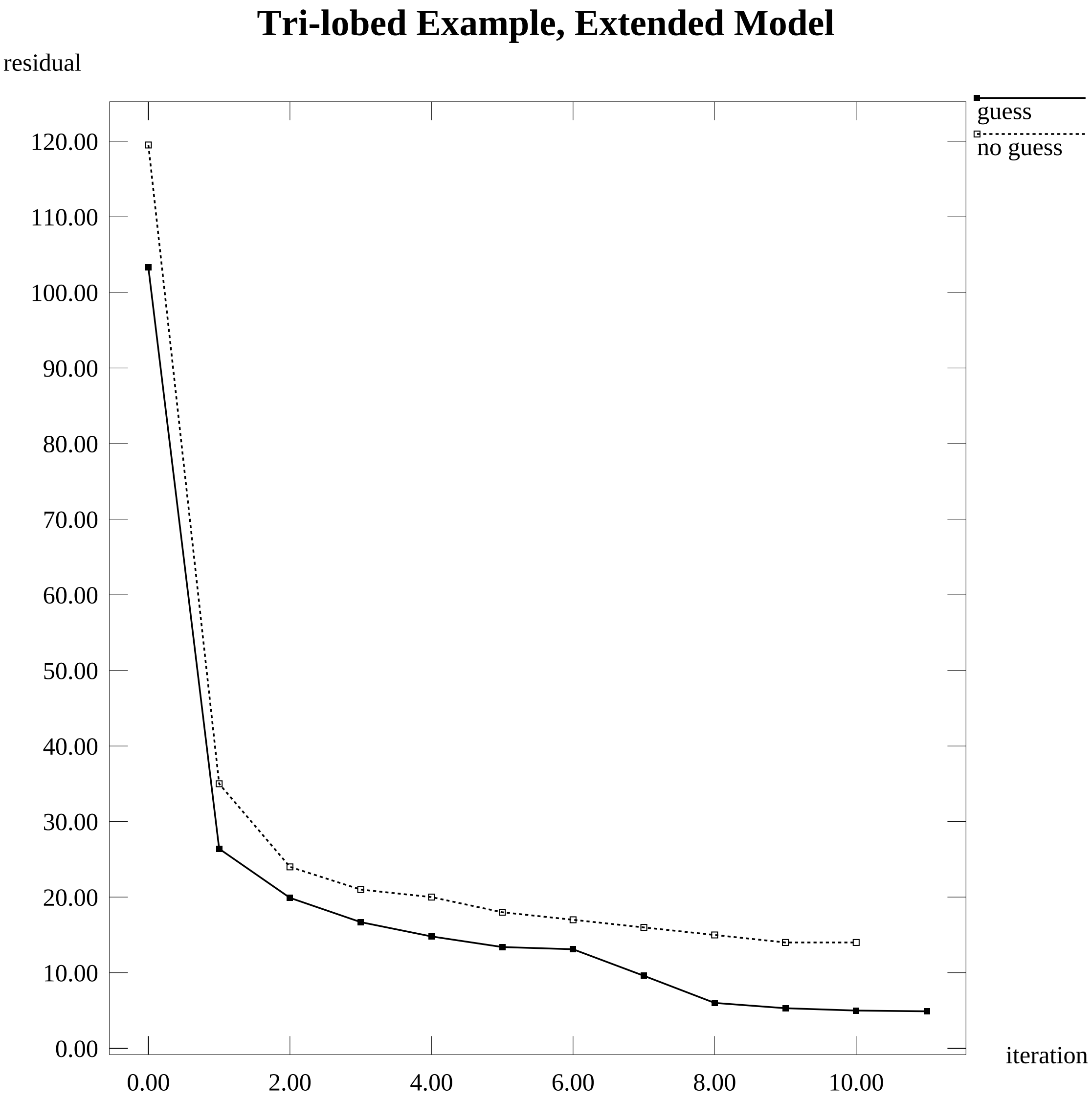

Figure 3-2: residual summary

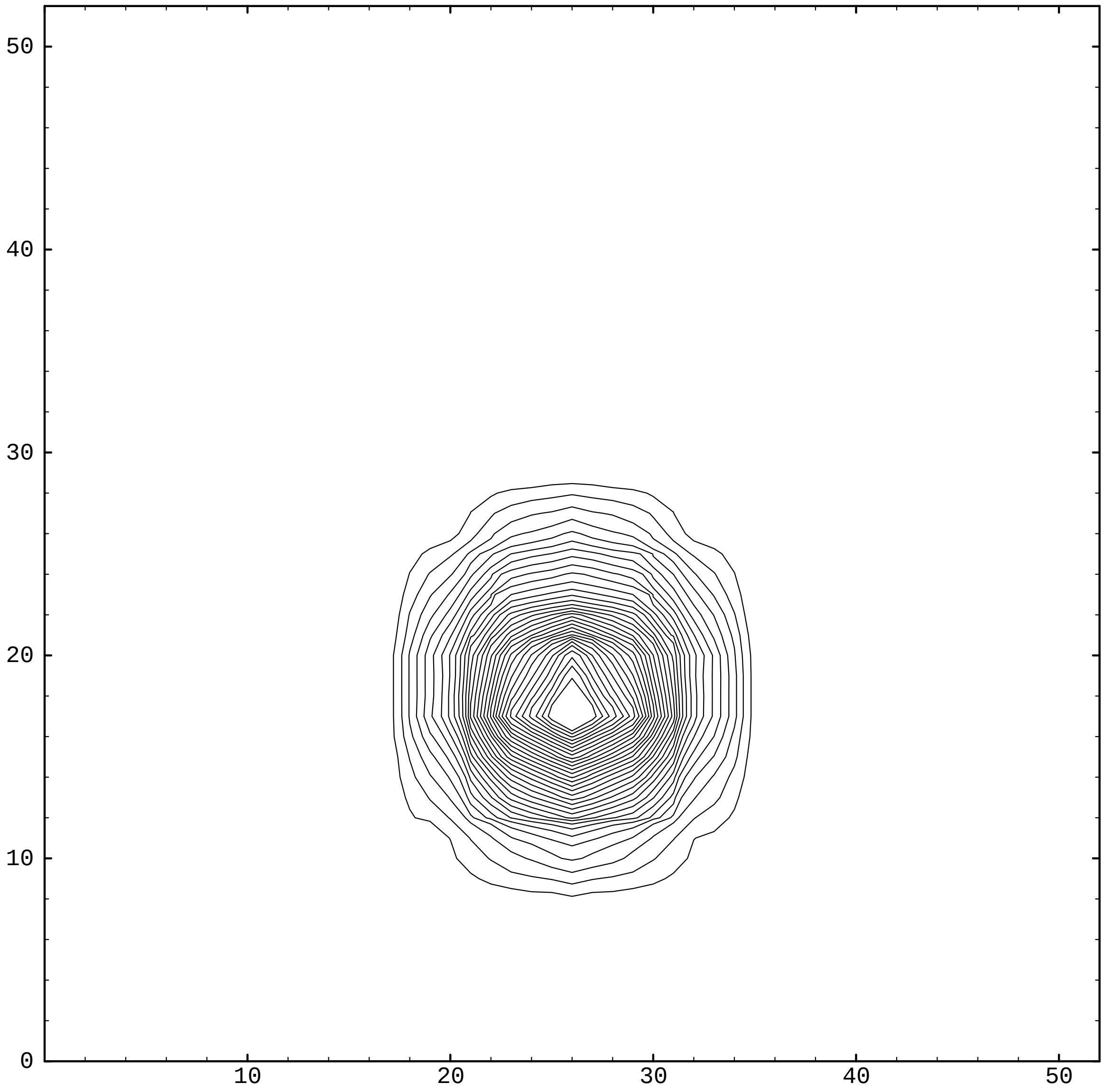

Figure 3-3: initial image, guess algorithm (tri-lobed example)

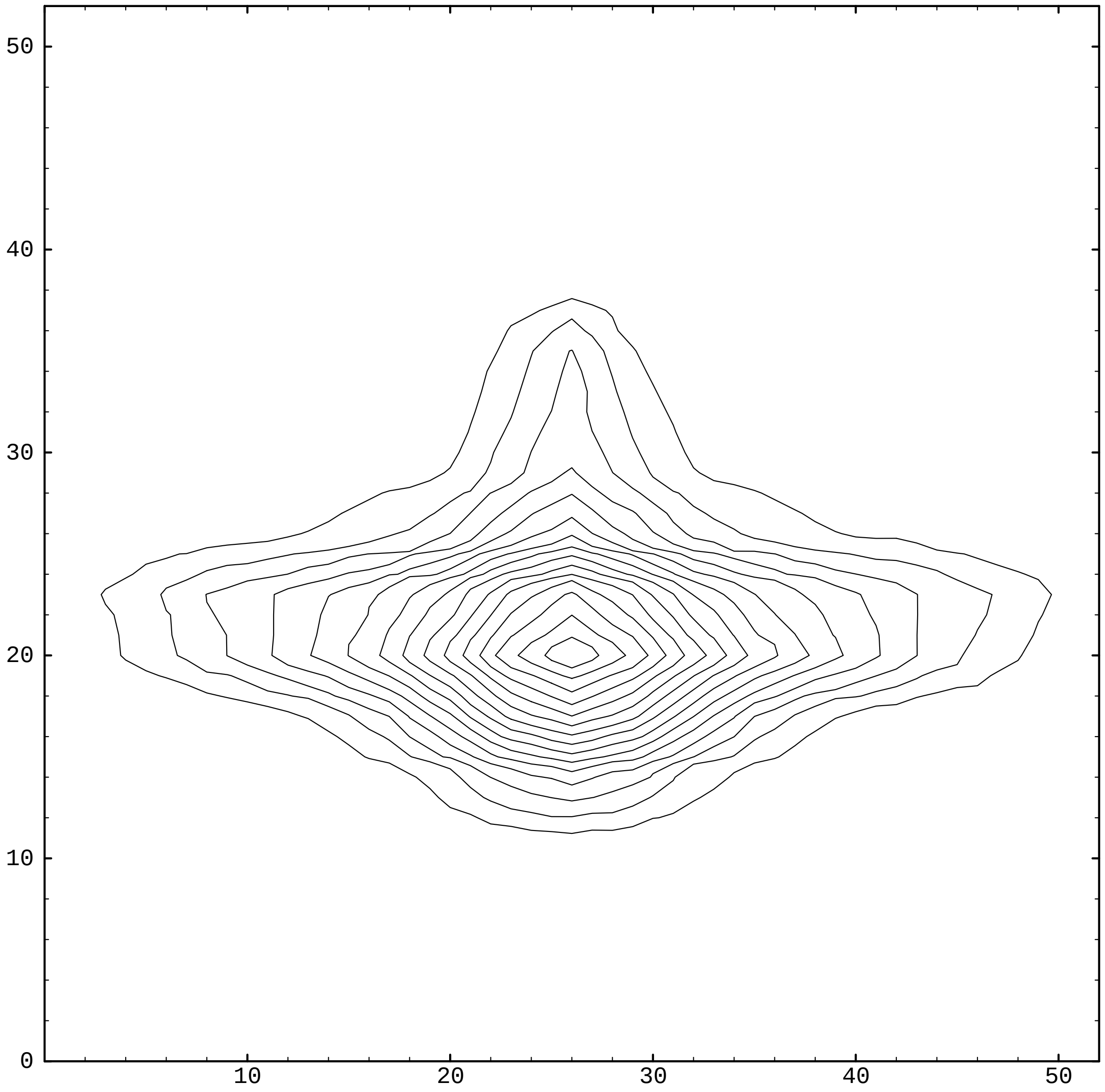

Figure 3-4: best solution, guess algorithm (trilobed example)

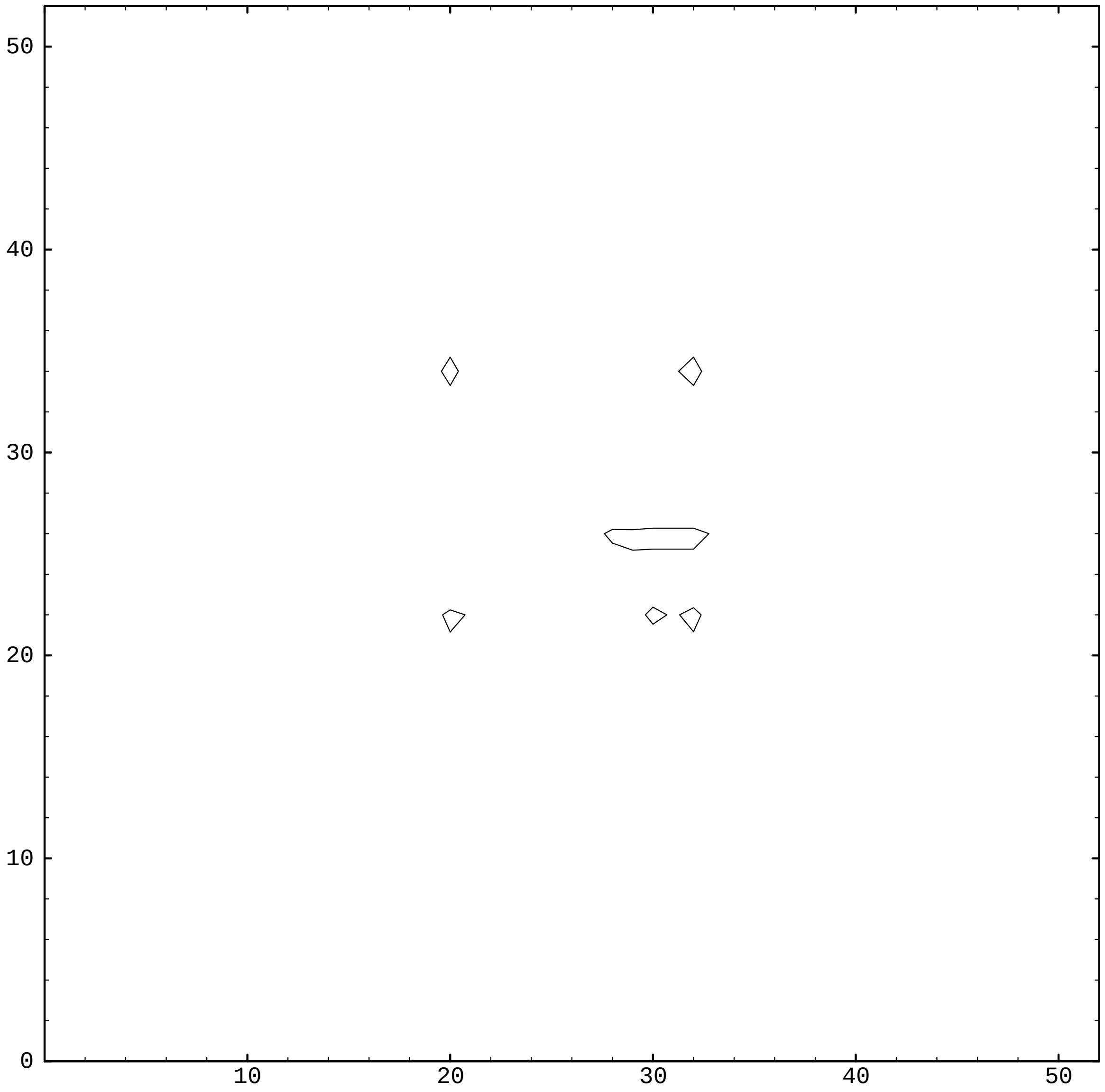

Figure 3-5: absolute difference plot (trilobed example)

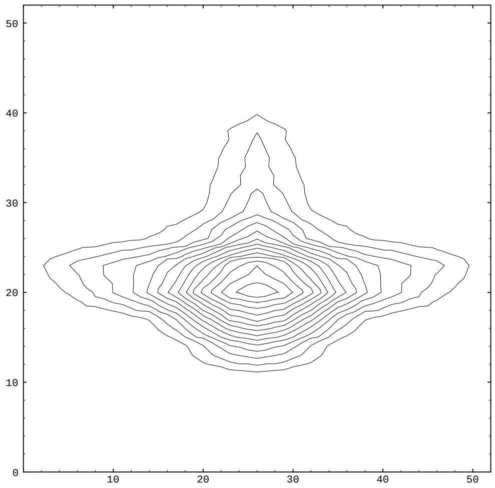

Figure 3-6: desired image (tri-lobed example)

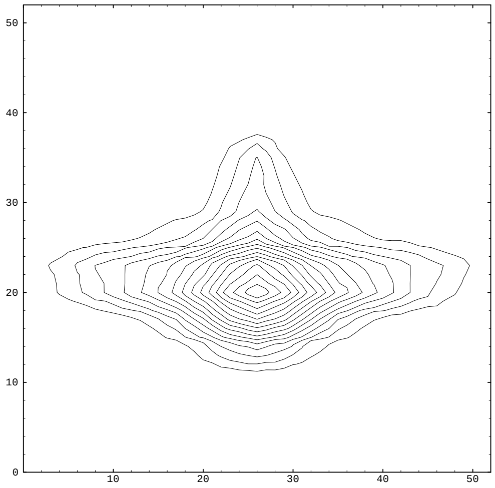

Figure 3-7: best solution, guess algorithm (tri-lobed example)

3.5 Results - Real Source

As a proof of concept, we present a mirror design example that uses actual spot quality bulb data to obtain a lighting pattern from a real source. The bulb dataset used is shown in figure 2-2. Because the CCD camera measurements were not calibrated, a true automotive design is not attempted. Instead, the point of this section is to show that the data compression and handling concepts of chapter 2 can indeed be put into practice.

The image was created on a 48x48 grid, with a pixel width of 0.5 *f*, at a distance of 60 *f* from the source. Every fourth pixel on every fourth row was rendered, with the rest filled in by linear interpolation. The mirror surface was sampled by squares 0.2 *f* on a side on a 27x27 grid.

To increase the rendering speed, each of the 400 pinhole camera pictures representing the source was fit to 25 m_{ij} terms instead of the usual 49; although the regeneration of individual pinhole camera images suffered, the effect on the overall projected image was small (an *rms* difference of less than 2.0 greyscales over a dynamic range of 140).

3.5.1 A Real Source Example

A desired image was created using six perturbations, including a strong linear term. As shown in figure 3-8, the residual was reduced to 10.7 after six tries. Figure 3-10 shows a plot of the absolute difference between the desired and best-fit images. Figure 3-9 shows the final derived image, and figures 3-11 and 3-12 show the desired and final images side by side for comparison. Each contour line in this section represents a change in intensity of seven greyscales, except for figure 3-10 where it represents 3.5 greyscales.

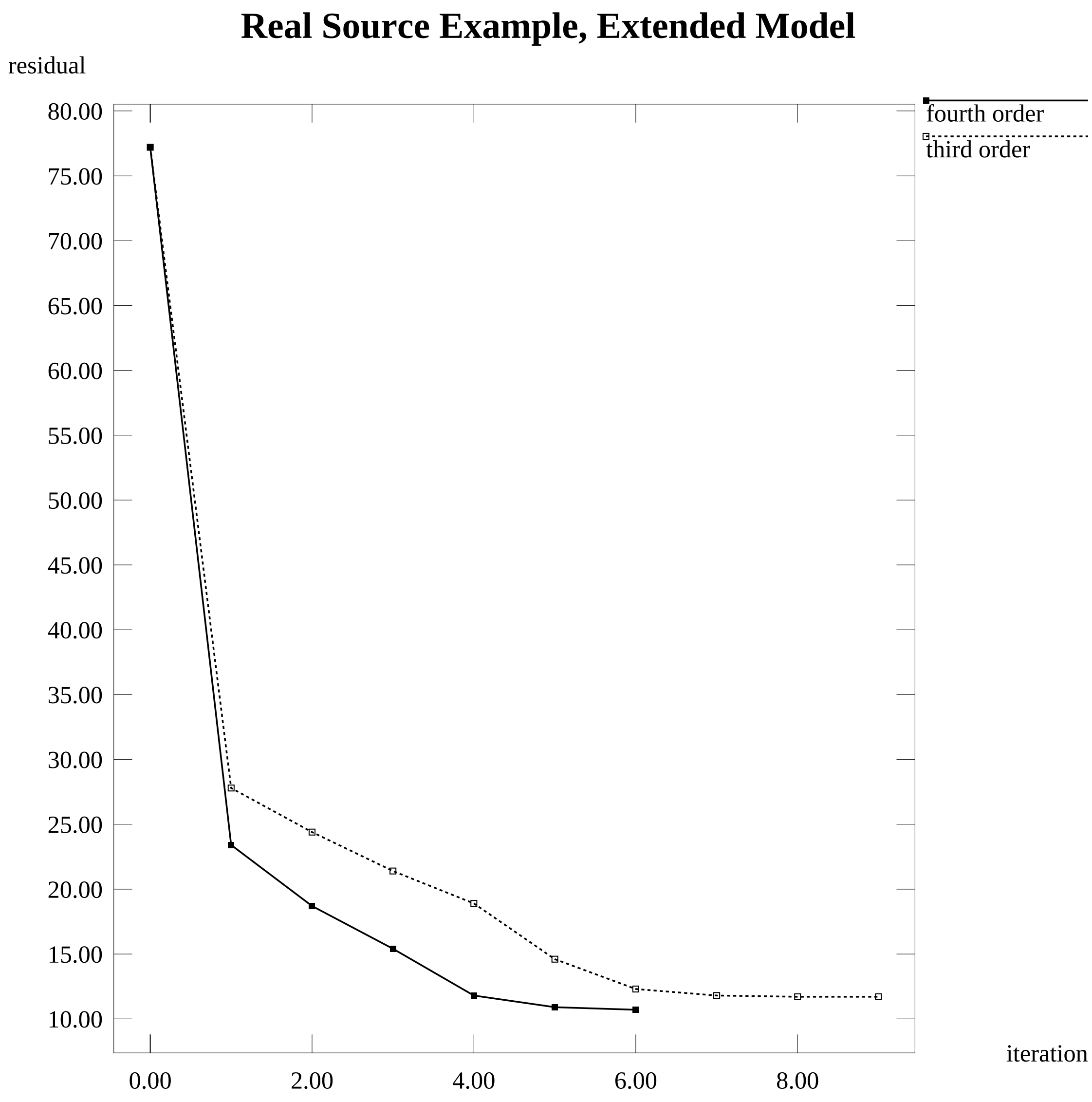

Figure 3-8: residual summary

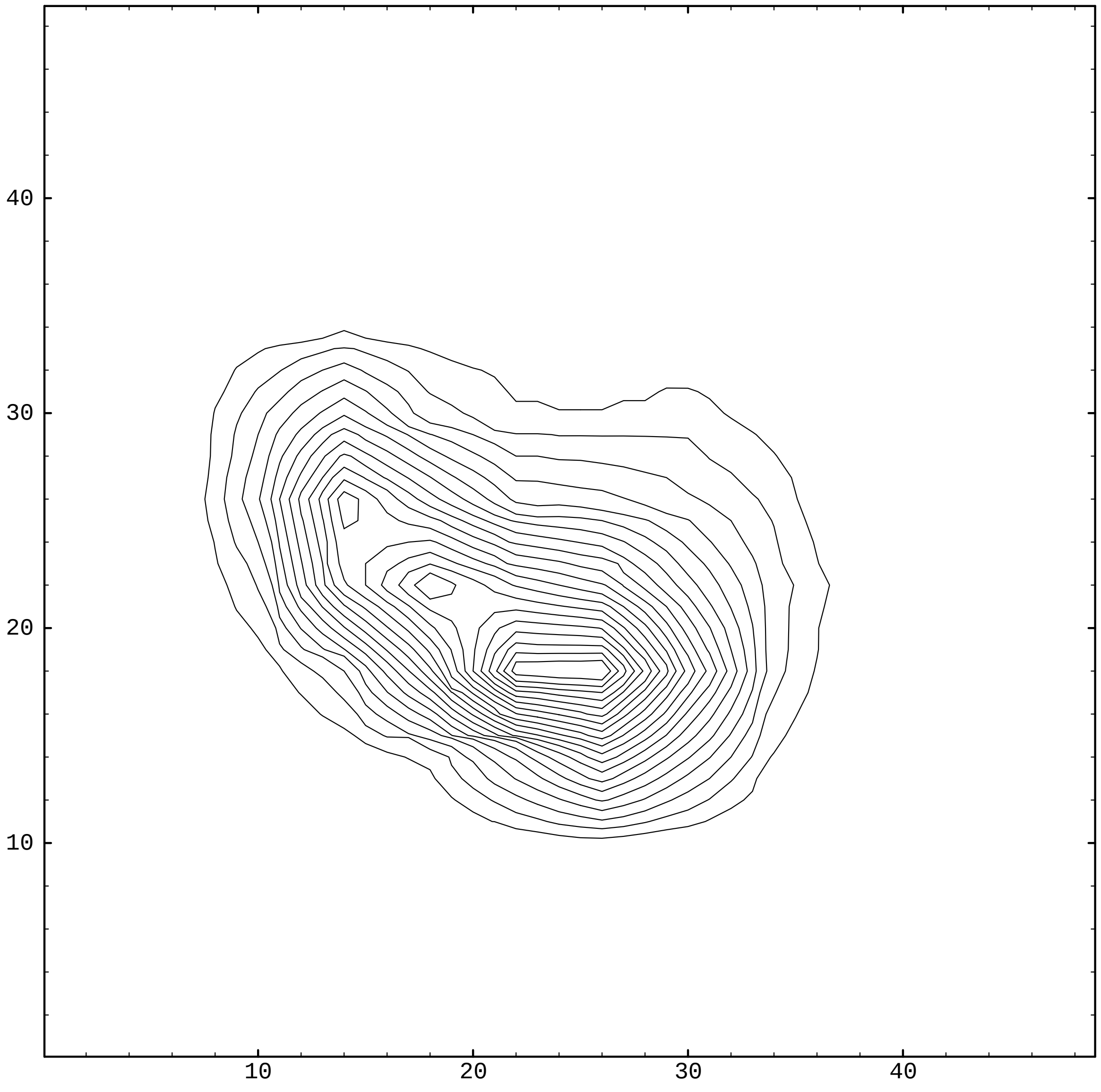

Figure 3-9: best solution, guess algorithm (real source example, extended model)

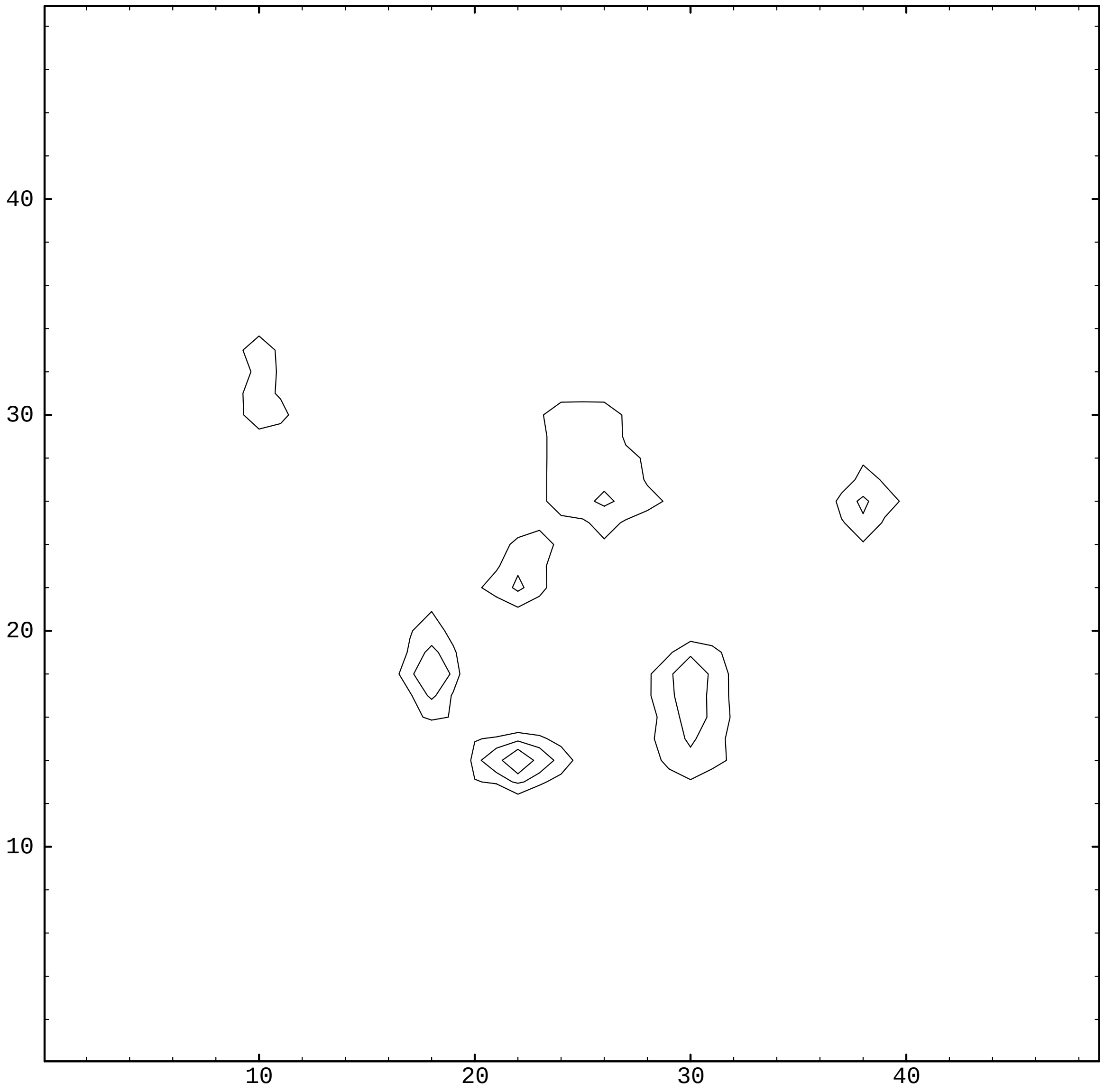

Figure 3-10: absolute difference plot (real source example, extended model)

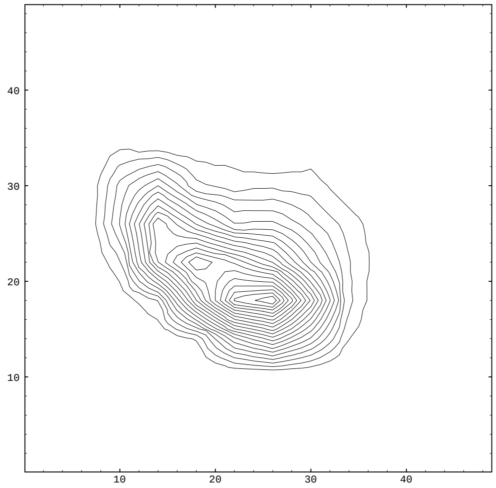

Figure 3-11: desired image (real source example)

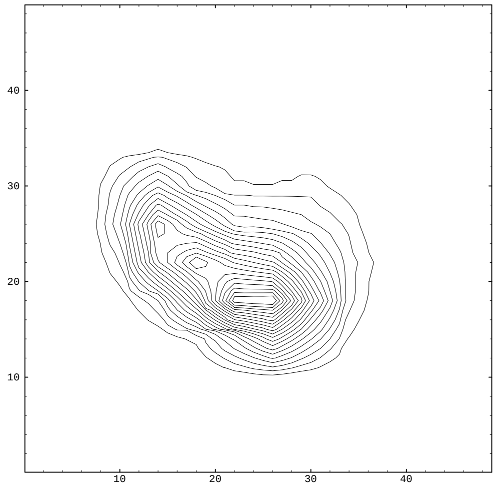

Figure 3-12: best solution, guess algorithm (real source example, extended model)

4. CONCLUSION

4.1 Significance

Fundamentally, the goal of this research is to develop a model of radiometry based on a combination of judicious measurement, efficient data representation, and physics based computation.

The general near-zone light source model (chapter 2) will find applications in the fields of realistic scene rendering, luminaire modeling, and luminaire design. Currently, the Fourier expansion seems to be the most compact and convenient description of the radiance distribution of an emitting surface.

The near-zone model is efficient because the radiance distribution of an extended source is encoded as a set of coefficients of orthogonal functions over a sphere, which solves the data storage problem, the interpolation problem, and the brute-force ray tracing intersection problem. The model is physics-based because it obeys the laws of geometric optics without resorting to the simplifying assumptions concerning the source's shape or radiance distribution often found in computer graphics-based methods. The model applies to real light sources with complete generality because it relies on measurement, thus correctly treating extended bodies with nonuniform radiance distributions.

The general light source model, using a nonuniform, extended filament, is successfully applied to the problem of designing a reflector to produce a desired illumination pattern on a screen (chapter 3). To do this, a smooth reflector model that consists of a set of physically understood perturbations to a parabolic base shape was developed. Because it is easy to calculate the intersection of a ray with our physically correct near-zone light source model, it is possible to parametrically sample the reflector surface and render an image on a distant screen with a combination of forward and backward ray tracing. This system can be used as a design tool with a human in the loop to test various reflector shapes and source types, or it can be used in conjunction with a computer-driven optimization algorithm. A judicious choice of coordinate system and a good first guess at the perturbations based on the shape of the perimeter of the image can often result in a better reflector, or decrease the time required for the search of perturbation space. A numerical optimization method was chosen that is appropriate to this type of numerical problem, in which derivative information is lacking.

If a non-spotlight type of image is desired, the general approach outlined here still holds, but a different reflector representation, especially for the base shape, should be used. A different set of functions for the image representation should also be found. In other words, it is a matter of choosing the proper basis functions for the problem at hand.

4.2 Acknowledgements

This paper is submitted in partial fulfillment of the requirements for R. D. Stock's degree of Doctor of Philosophy at Carnegie Mellon University.

I. BLOB-FITTING DETAILS

I.1 Blobs and Moments

A "blob" is a roughly elliptical, roughly Gaussian 2-dimensional distribution. In a coordinate system in which a perfectly bi-Gaussian blob is centered on the origin with its major and minor axes coincident with the $\overline{x}$- and $\overline{y}$-axes, its intensity distribution can thus be written in the simple form:

$$G(\overline{x},\overline{y}) = A \exp\{-\overline{x}^2/(2\overline{\sigma}_x{}^2)\} \exp\{-\overline{y}^2/(2\overline{\sigma}_y{}^2)\}$$

(In this appendix, coordinates with overlines refer to the rotated and scaled frame, whereas those without are absolute pixel coordinates in the camera frame). The zeroth moment m_{00}, the volume under the bi-gaussian surface, is related to A by a constant factor:

$$m_{00} = A\sqrt{\pi}$$

All of the other m_{ij} are zero for a bi-Gaussian blob.

For a blob that is not exactly bi-Gaussian, but roughly so, m_{00} would still contain most of the intensity information, and the other moments would be relatively small.

To separate details of camera orientation and blob location in the camera's field of view from representation of the inherent structure of the blob, we initially define and compute five parameters: the *x*- and *y*-coordinates of the center-of-illumination, the tilt-angle of the blob with respect to the *x*-axis, and the standard deviations of the blob along and perpendicular to the tilt-axis. Figure I-1 shows the coordinate system used for this analysis.

We name these offset, rotation, and distance scale parameters as follows:

The total intensity is:

$$S_{00} = \sum_{xy} S(x,y)$$

where *S(x,y)* is the intensity measured at pixel *(x,y)*.

The offsets are:

$$x_{00} = \langle x \rangle = \sum_{xy} x\, S(x,y) \,/\, S_{00}$$

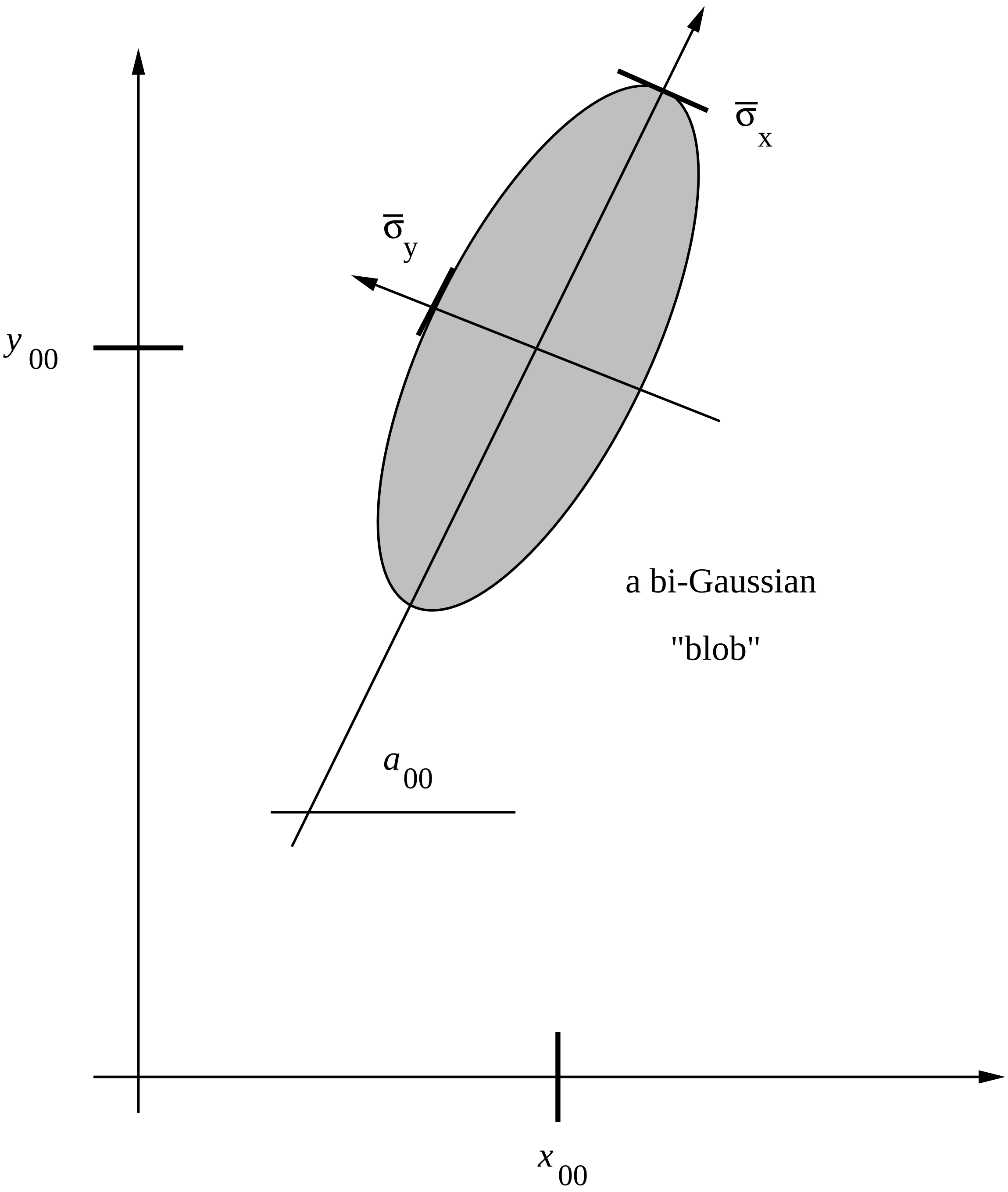

Figure I-1: coordinate system for decomposition of a blob

and

$$y_{00} = \langle y \rangle = \sum_{xy} y\, S(x,y) \,/\, S_{00}$$

The squared standard deviations before rotating by the tilt angle are:

$$\sigma_{xx}{}^2 = \langle x^2 \rangle - \langle x \rangle^2 = (\int\int_{-\infty}^{+\infty} dx\, dy\, S(x,y)\, x^2 \,/\, S_{00}\,) - x_{00}{}^2$$

$$\sigma_{yy}{}^2 = \langle y^2 \rangle - \langle y \rangle^2 = (\int\int_{-\infty}^{+\infty} dx\, dy\, S(x,y)\, y^2 \,/\, S_{00}\,) - y_{00}{}^2$$

and

$$\sigma_{xy}{}^2 = \langle xy \rangle - \langle x \rangle \langle y \rangle = (\int\int_{-\infty}^{+\infty} dx\, dy\, S(x,y)\, x\, y \,/\, S_{00}\,) - x_{00} y_{00}$$

The formula for the tilt angle is derived by noting that for a bi-Gaussian distribution, the expectation value $\langle \bar{x}\,\bar{y} \rangle$ equals zero in a coordinate system whose origin is located at the central maximum and whose $\bar{x}$-axis is aligned with the major axis of the distribution. Writing this expression in the original, unshifted coordinate system results in

$$\langle \{(x - x_{00})\cos(a_{00}) + (y - y_{00})\sin(a_{00})\}\{(x - x_{00})\sin(a_{00}) - (y - y_{00})\cos(a_{00})\}\rangle = 0$$

Expanding and using the above definitions gives:

$$(\sigma_{xx}{}^2 - \sigma_{yy}{}^2)\cos(a_{00})\sin(a_{00}) = \sigma_{xy}{}^2(\cos^2(a_{00}) - \sin^2(a_{00}))$$

Rearranging and using a standard identity produces the tilt angle:

$$a_{00} = 0.5 \text{ arctan}(2\sigma_{xy}{}^2 \,/\, \{\sigma_{xx}{}^2 - \sigma_{yy}{}^2\})$$

After translating, scaling, and rotating the axes, the standard deviations along the major and minor axes of the blob are:

$$\bar{\sigma}_x = (\int\int_{-\infty}^{+\infty} dx\, dy\, S(x,y)\, \bar{x}(x,y)^2 \,/\, S_{00}\,)^{1/2}$$

and

$$\bar{\sigma}_y = (\int\int_{-\infty}^{+\infty} dx\, dy\, S(x,y)\, \bar{y}(x,y)^2 \,/\, S_{00}\,)^{1/2}$$

where

$$\bar{x} = (x - x_{00})\cos(a_{00}) + (y - y_{00})\sin(a_{00})$$

and

$$\bar{y} = (y - y_{00}) \cos(a_{00}) - (x - x_{00}) \sin(a_{00})$$

Or, in terms of a_{00},

$$\bar{\sigma}_x{}^2 = \sigma_{xx}{}^2 \cos^2(a_{00}) + 2\sigma_{xy}{}^2 \sin(a_{00}) \cos(a_{00}) + \sigma_{yy}{}^2 \sin^2(a_{00})$$

and

$$\bar{\sigma}_y{}^2 = \sigma_{xx}{}^2 \sin^2(a_{00}) - 2\sigma_{xy}{}^2 \sin(a_{00}) \cos(a_{00}) + \sigma_{yy}{}^2 \cos^2(a_{00})$$

I.2 Harmonic Oscillator Wave Functions

Because the interval over which a blob extends is arbitrary, generating a unique expansion independent of the number of terms[24, 25] requires the basis functions to be orthogonal over the interval $(-\infty, +\infty)$.

The Gaussian and either its derivatives or its products with simple polynomials have previously been used in the computer vision literature as basis functions for fitting blobs, blurred edges, etc.[26] However it appears that no one in computer vision has yet recognized that the harmonic oscillator wavefunctions are the right way to implement the intent, i.e., to use a set of basis functions that have the requisite mathematical properties *and* whose shapes are well matched to the shape of the data, thereby achieving good fits with a small number of terms.

The harmonic oscillator wavefunctions are:

$$\Phi_n(z) = (1/\sqrt{\pi^{1/2}\, 2^n\, n!})\ \exp(-z^2/2)\, h_n(z)$$

where the $h_n(z)$ are the Hermite polynomials:

$$h_0(z) = 1$$
$$h_1(z) = 2z$$
$$h_2(z) = 4z^2 - 2$$
$$h_3(z) = 8z^3 - 12z$$
$$h_4(z) = 16z^4 - 48z^2 + 12$$
$$\cdots$$
$$h_n(z) = 2z\, h_{n-1}(z) - 2(n-1)\, h_{n-2}(z)$$

They are orthonormal:

$$\int_{-\infty}^{+\infty} dz\, \Phi_i(z)\, \Phi_j(z) = \delta_{ij}$$

and complete:

$$\sum_i \Phi_i(z)\, \Phi_i(z') = \delta(z-z')$$

It is therefore convenient, and it has proven effective, to represent appropriately centered and oriented "blobs" as the coefficients of a Fourier expansion in products of harmonic oscillator functions for each coordinate direction:

$$S(x,y) = \sum_{ij} m_{ij}\, \Phi_i(\overline{x}(x,y)/\overline{\sigma}_x)\, \Phi_j(\overline{y}(x,y)/\overline{\sigma}_y)$$

where $\overline{\sigma}_x$ and $\overline{\sigma}_y$ are the standard deviations in $\overline{x}$ and $\overline{y}$. Using the standard deviation in each direction as the scale factor for distance in that direction ensures that the spatial frequencies contained in $\Phi_i(z)$ are neither so small (the wavelengths so long) that there is danger of undersampling and thus aliasing, nor so large (the wavelengths so short) that there is oversampling and thus loss of computational efficiency.

I.3 Method of Calculating Coefficients

The coefficients m_{ij} are calculated from representative data by

$$m_{ij} = \int\int_{-\infty}^{+\infty} dx\, dy\, S(x,y)\, \Phi_i(\overline{x}(x,y)/\overline{\sigma}_x)\, \Phi_j(\overline{y}(x,y)/\overline{\sigma}_y) \,/\, (\overline{\sigma}_x \overline{\sigma}_y)$$

The values of *i* and *j* range from zero to some predetermined limit *mmax*, typically equal to six.

This discrete Fourier integral "dot product" method for calculating the coefficients works well when the data are (in an appropriate sense) evenly distributed over the space in which the basis functions are orthogonal, and when the sum and the integral are really equivalent (in an appropriate sense). The dot product calculations are much faster than least-squares fitting, and (unlike least-squares fitting with non-orthogonal functions) adding higher order coefficients does not change the values of the previously calculated lower order coefficients. However the statistics that emerge as a side effect of the least-squares calculation are of course not obtained "for free" with the dot product.

II. CONJUGATE VARIABLES

The basic procedure for iteration k of the conjugate variables optimization method is:

1. Let n search directions $\boldsymbol{v}_i^k$ and a starting point $\boldsymbol{x}_0^k$ in reflector coefficient space be given.
2. For $i=1, . . . ,n$, find λ_i to minimize the residual function $R(\boldsymbol{x}_{i-1}^k+\lambda_i\boldsymbol{v}_i^k)$ described in section 3.2.1, and define $\boldsymbol{x}_i^k=\boldsymbol{x}_{i-1}^k+\lambda_i\boldsymbol{v}_i^k$.
3. Minimize from point $\boldsymbol{x}_n^k$ in the direction of $\boldsymbol{v}_{n+1}^k=\boldsymbol{x}_n^k-\boldsymbol{x}_0^k$, and call the resulting point $\boldsymbol{x}_0^{k+1}$.
4. Define a new set of search directions so $\boldsymbol{v}_i^{k+1}=\boldsymbol{v}_{i+1}^k$. The last k directions of the set can be shown to be mutually conjugate.
5. Check halting criteria.
6. Increment k and go to 1.

Search directions $\boldsymbol{p}$ and $\boldsymbol{q}$ are said to be conjugate with respect to the quadratic surface

$$f(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}\cdot\boldsymbol{x}+c$$

if

$$\boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}=0$$

For n variables a complete set of conjugate search directions can theoretically be built up after n full iterations. It can be proved[23] that if the current best solution for the minimum is close enough to the true local minimum such that the quadratic approximation to the residual surface is valid, then the local minimum will be found by simply minimizing exactly once along each of the conjugate search directions in any order. Convergence (within tolerance) for a quadratic form (or near a minimum) is therefore linear in the number of search parameters.

Powell’s improvement for efficiently choosing the search directions, based on numerical realities, is that the actual search direction to be replaced, if any, is the one that will maximize the determinant of the matrix $\boldsymbol{V}$ of search directions $\boldsymbol{v}_i$. This prevents the search directions from becoming too linearly dependent, reducing the efficiency of the search or restricting it to a subspace. Some[27, 28] have suggested that Powell’s method is too stringent, because the replacement of a conjugate direction will slow the convergence. Instead, they propose that the determinant of search directions merely be kept above some minimum value.[13]

Brent discusses a variety of further modifications. For instance, the search direction to be replaced is restricted to the set of non-conjugate directions. Thus, there is some choice to allow the determinant of ***V*** to remain high, but conjugate directions are not thrown away. Next, once a complete set has been found, to avoid having the directions eventually become dependent, ***V*** is periodically reset to some orthogonal matrix. The standard choice is to simply start again with the identity matrix, but in order to preserve information about the surface, however, Brent resets ***V*** to the matrix whose columns are the principal axes of the quadratic form that best fits the residual surface at the current point. In this way the search directions remain conjugate. Scaling of the search directions is also periodically performed to reduce the condition number of ***V*** to continue to fight linear independence. A random step is also incorporated if a valley is suspected; the minimization from a nearby random point should find another point in the valley and then a search in the direction from one of these minima to the other should allow quick traversal.

Brent also describes a halting criterion. The search stops whenever

$$|\boldsymbol{x}_0^{k+1} - \boldsymbol{x}_0^{k}| \leq \varepsilon^{1/2} |\boldsymbol{x}_0^{k+1}| + T$$

holds for some number of iterations (typically one or two) where ε is the machine epsilon and *T* a positive tolerance.

[13] Concerning which search direction to discard in the minimization algorithm, we would like to note that since the "condition number" of a matrix is a better measure of the linear dependence of its columns than the determinant[29], that the best direction to be replaced is the one that keeps *cond*(***V***) small rather than the one that keeps *det*(***V***) large. The condition number can be estimated as the ratio of the largest and smallest singular values of the matrix, found by a Singular Value Decomposition[30, 31, 25, 32]. The time involved for repeated decompositions may be prohibitive, however.

III. FIRST GUESS ALGORITHM

In outline, the method works by comparing the shape of the perimeter of the image cast by an unperturbed parabolic reflector with that of the desired image. The change in radius between the two images at a given angle around the perimeter is attributed entirely to the change in slope of the corresponding reflector element at the parabolic reflector's edge. The procedure is to start from the center of each image, move out along corresponding radial lines, and look for the first edge. The required slope can be written in terms of the coordinates of the appropriate reflector edge element, and hence in terms of the perturbation coefficients. Considering several image radii and reflector edge elements, for small perturbations a set of overdetermined linear equations results, which can be solved for the most consistent set of perturbation coefficients by the method of least-squares.

The perimeters are found with the Roberts edge detector formula[33, 34]. An edge occurs at $(x+1/2, y+1/2)$ when the function of image intensity

$$g(x,y) = |I(x,y) - I(x+1,y+1)| + |I(x+1,y) - I(x,y+1)|$$

is a maximum. Subpixel interpolation can be used to refine the edge calculation. Operating on the image of figure III-1, for example, the Roberts edge detector found the edge points shown in figure III-2.

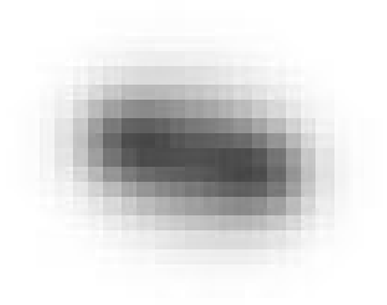

Figure III-1: a sample image

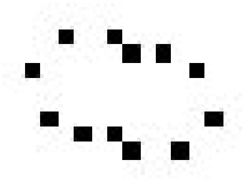

Figure III-2: twelve edge points

The full superimposed edges of the desired and initial images might look like the perimeters shown in figure III-3. Two pairs of corresponding radii are shown.

Figure III-4 shows how the change in perimeter radius along a given line is attributed to the tilt

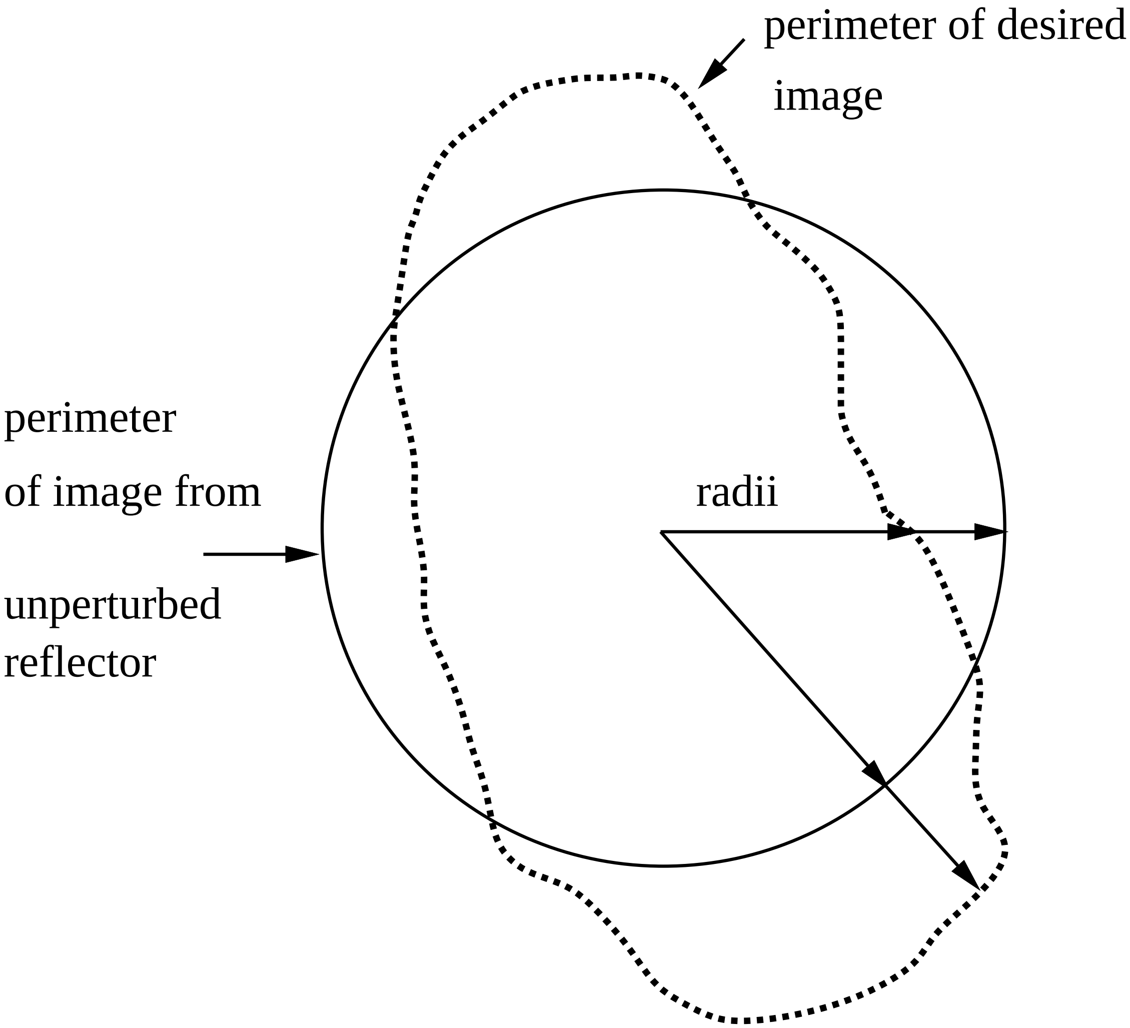

Figure III-3: comparing radii of image perimeter edges

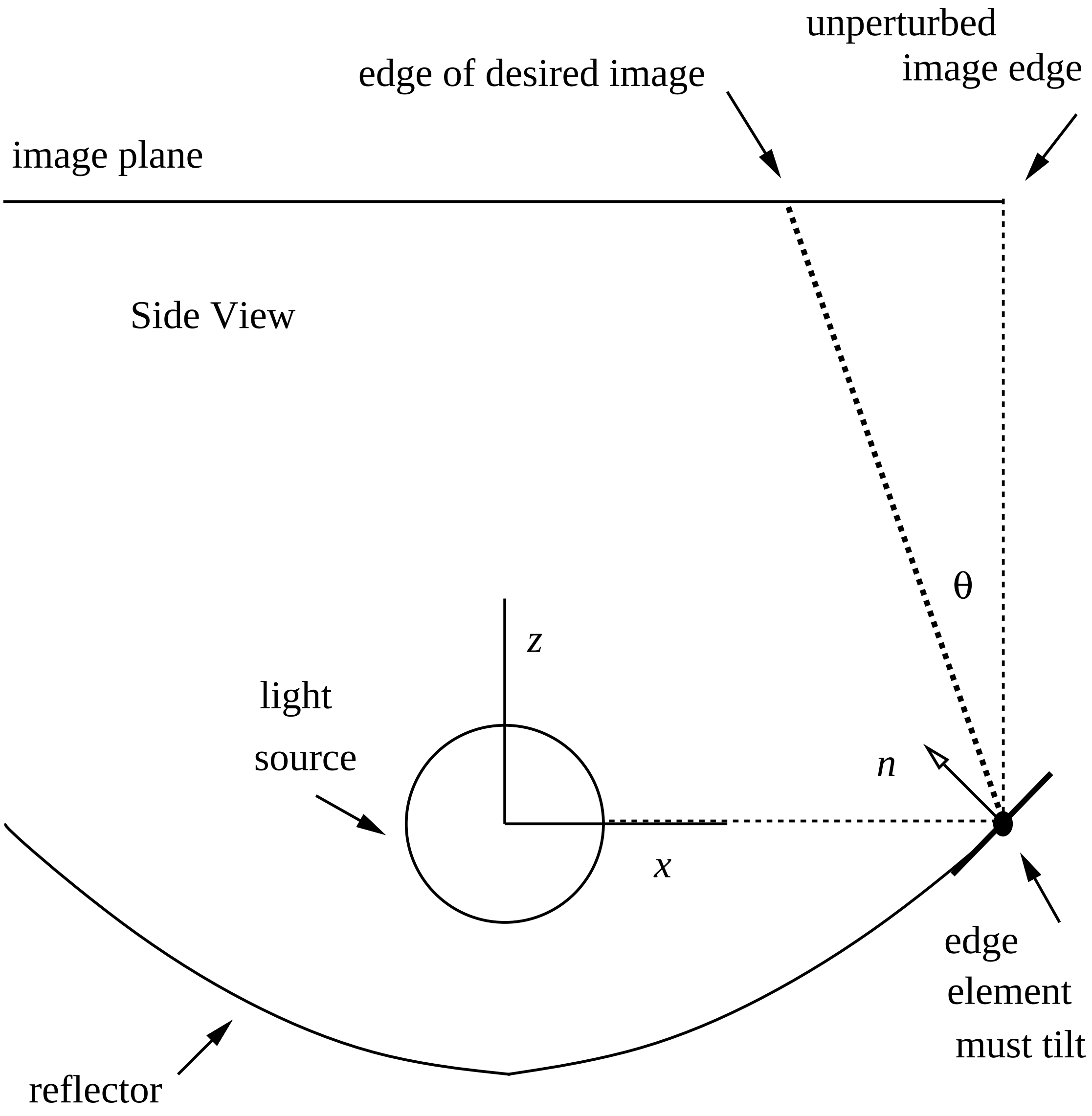

Figure III-4: image edge shape roughly corresponds with reflector edge tilt

of an edge element of the reflector. The z axis corresponds with the axis of the parabolic reflector, as explained in section 3.3 (in the figure the z axis happens to be perpendicular to the image plane, but this is not necessary). The new angle θ from the unit vector $\boldsymbol{z}$ to the required normal vector $\boldsymbol{n'}$ of the edge of the reflector at this point is easily calculated once the coordinates of the image edge points are known, and thus

$$\boldsymbol{n'} \cdot \boldsymbol{z} = 1 = (p^2 + q^2 + 1)^{1/2} \cos\theta$$

where

$$p = \frac{dz}{dx}$$

and

$$q = \frac{dz}{dy}$$

Expanding p^2 and q^2 and assuming the perturbations are small, this can be written as

$$2/\cos^2\theta - 4 = l_1 x + l_2 y + 2a_1(x^2 - y^2) + 2a_2 xy +$$
$$3c_1 x^3 + 3c_2 y^3 + 3c_3 x^2 y + 3c_4 xy^2 + 2r(x^2 + y^2)^{1/2}$$

using the fact that $(x^2 + y^2)/4 = 1$ around the rim of the parabolic reflector and eliminating all terms that are second order or higher in the perturbation coefficients. The extension to the use of additional terms in the reflector model is straightforward.

For several radius pairs, each yielding an x, y, and θ, a system of equations linear in the perturbation coefficients results, which can be quickly solved by a least squares fit. If the perturbations are indeed small, the guess often performs remarkably well. For example, for a desired image with an exact solution of l_1 = -5.0x10^{-4} and a_1 = -3.0x10^{-5}, the guess returns l_1 = -5.3x10^{-4} and a_1 = -2.6x10^{-5}, reducing the residual from 36.7 (for a vertically oriented pure paraboloid) to 2.0. If the axis of the pure parabolic mirror is oriented to point at this image's center of intensity, as outlined in section 3.3, the residual immediately drops to 7.9. Applying the guess to this tilted reflector results in l_1 = -1.7x10^{-5} and a_1 = -3.0x10^{-5}, reducing the residual to 5.1. This is not as good as the original guess in the vertical frame because an exact solution does not exist in the tilted reference frame. Note how the guess for the linear coefficient l_1 in the tilted frame decreases by an order of magnitude; this is because the linear term mainly represents a shift of the image position, which is largely taken care of simply by pointing the reflector the right way. If the perturbations are not in fact small, the guess may not be very useful however.

Trying to guess too many terms (typically eight) by the least-squares method often results in a small pivot and a spurious "inverse" matrix. This is because the dataset cannot distinguish between the contributions of perturbations with similar symmetry. It is therefore necessary to use the covariance matrix to calculate the relative uncertainty of each term[24], and then guess only the most significant terms, varying the number of data points used until the best overall fit can be found. In practice, two to three times as many data points as coefficients resulted in the best guesses.

IV. PARAMETER FITTING DETAILS

This appendix describes how difficulties relating to aliasing and nonperiodicity were handled when the blob parameters were fit over the measurement sphere with the α_{lm}^{k}.

IV.1 Noncontinuity

Because the tilt angle a_{00} is calculated by an arctangent (appendix I), it ranges from –90° to +90°, resulting in large discontinuous jumps from one endpoint to the other. Figure IV-1 shows a set of raw tilt angle data in two-dimensional form; the right-left axis with 25 data points is ϕ_n, the front-back axis with 16 data points is θ_n, and the vertical axis is the tilt angle in radians. The spherical fitting routine then wastes much of its energy fitting to these nonphysical jumps, sometimes resulting in a poor overall fit. In addition, a problem similar to aliasing can result when regenerating in a jump region. For example, assume two neighboring tilt angle data points are +89° and –89°. A proper fit between these two points should range from +89° to +90°, and then from –90° to –89°. The actual fit, however, will range from +89° down through 0° and then on to –89°, going through precisely the wrong points. Because the accuracy of the fit to the eye depends critically on the tilt of the image, this situation is serious.

The aliasing problem was remedied by adding offsets of ±180° to the tilt angle data whenever necessary to smooth it out. The result of this smoothing procedure is shown in figure IV-2. Reconstructed tilt angles that lie outside of the legal range are then "de-shifted" back into range. One question that remained was how to deal with slight fitting errors near ±90°. Is a reconstructed angle that is just over +90° supposed to be shifted by –180°, or is it merely a slight fitting error for an angle that is actually supposed to be just under +90°? The consequence of such a mistake in the fit angle results in a rotation of the image by 180°. Thus "Shannon sampling" interpolation theory[35] is used as a consistency check for angles which reconstruct "close" to the legal endpoints to determine whether or not they should be shifted by ±180°. The only drawback is that the interpolation theory requires the retention of the complete original dataset of tilt angles. The *rms* residual of one dataset of tilt angles was reduced by a factor of four by this method.

IV.2 Nonperiodicity

Not all datasets saw an improvement in the residual, however (though the aliasing problem was solved). For example, for a spot quality bulb dataset the *rms* residual increased from 0.10 to 1.12 after adding the offsets. This is because spot quality bulb filament images rotate fairly regularly and the tilt angles thus always go through +90° in the same direction, causing the offsets to accumulate instead of averaging out to zero. This results in the adjusted tilt angle of the last data point differing vastly from the first data point. The 2-D Fourier fit requires continuity of the data in both ϕ and θ or else convergence will be slow; not only will the offsets therefore prevent the data from "wrapping" in ϕ, but the accumulation of this effect will cause a severe discontinuity in θ. The discontinuity in θ is quite apparent in figure IV-2.

In the one-dimensional case, the standard way[25] to solve the discontinuity problem when using Fourier series is to subtract a line that goes through the first and last data points from the data, and then fitting to the (now continuous) result. Upon reconstruction, the line is added back, requiring the retention of only two extra parameters to specify the line. I extended this trick to the two-dimensional case by subtracting a plane from the data that goes through the first and last data points, and is independent of ϕ. After subtracting this plane, the processed tilt angle data are shown in figure IV-3. Notice the creation of continuity in θ; this decreased the *rms* residual by a factor of six, to 0.19. The resulting regenerated fit, after postprocessing out the plane and offsets, is shown in figure IV-4.

Because the tilt angle data still did not wrap in ϕ, more coefficients were used to fit this parameter than any of the other image parameters (345 coefficients versus 49 coefficients for 425 data points). Other parameters such as m_{00} that do not suffer from ϕ-nonperiodicity are easier to fit; the data and fit for a typical m_{00} dataset are shown in figures IV-5 and IV-6. The subtraction of the plane did not tend to change the *rms* residual of the other parameters very much, partly because on physical grounds they were automatically continuous in ϕ. Although not necessarily continuous in θ, the discontinuities were still not as severe as the case of the tilt angle with its accumulating offsets.

A better solution to the problem of fitting a_{00} could be to dispense with fitting it altogether

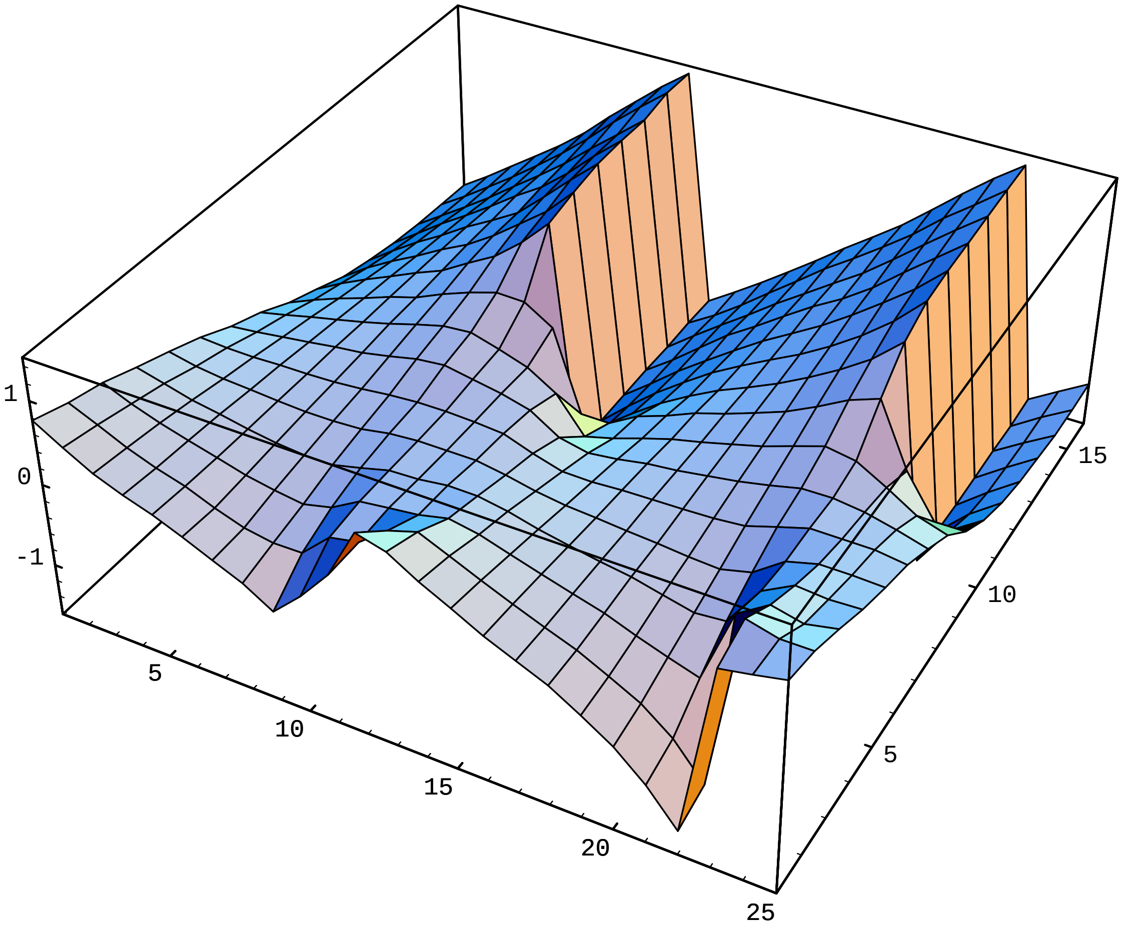

Figure IV-1: raw tilt angle data

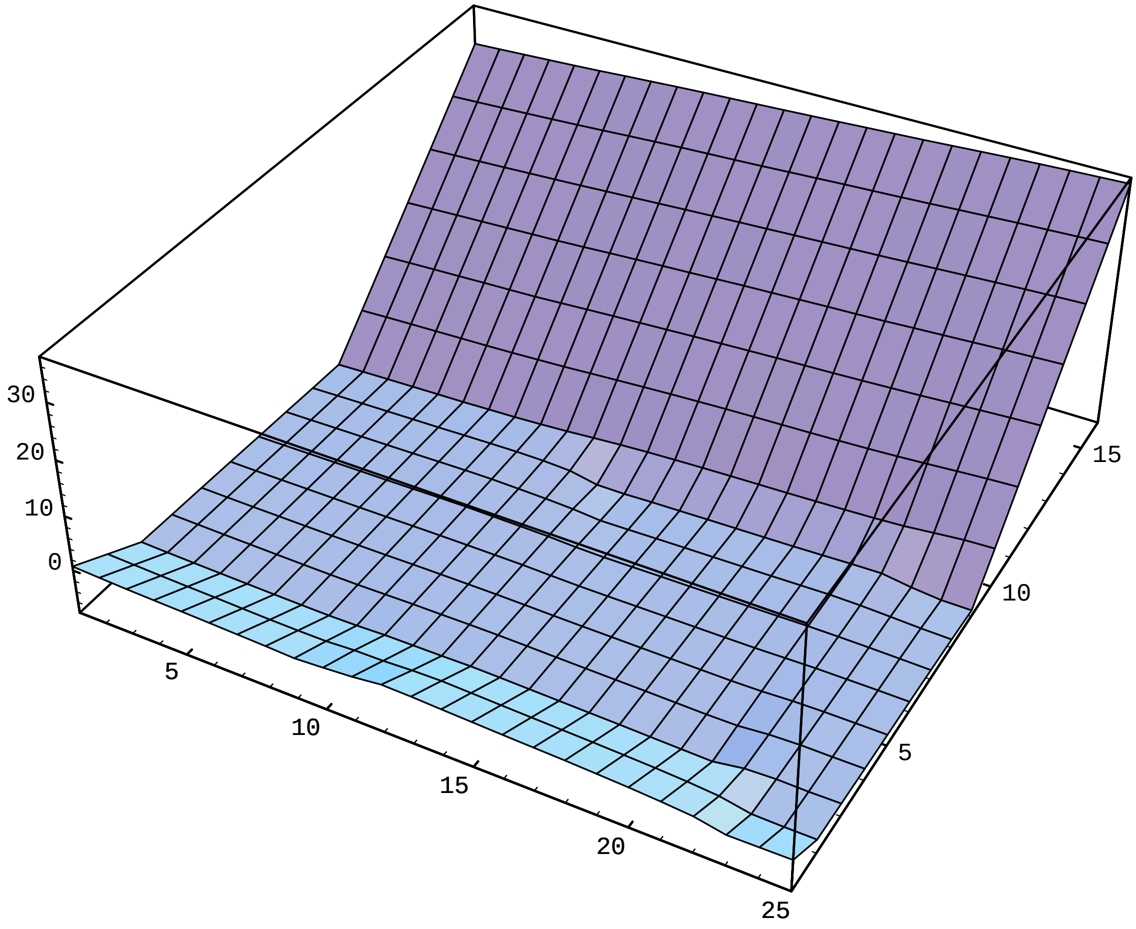

Figure IV-2: tilt angle data smoothed with offsets

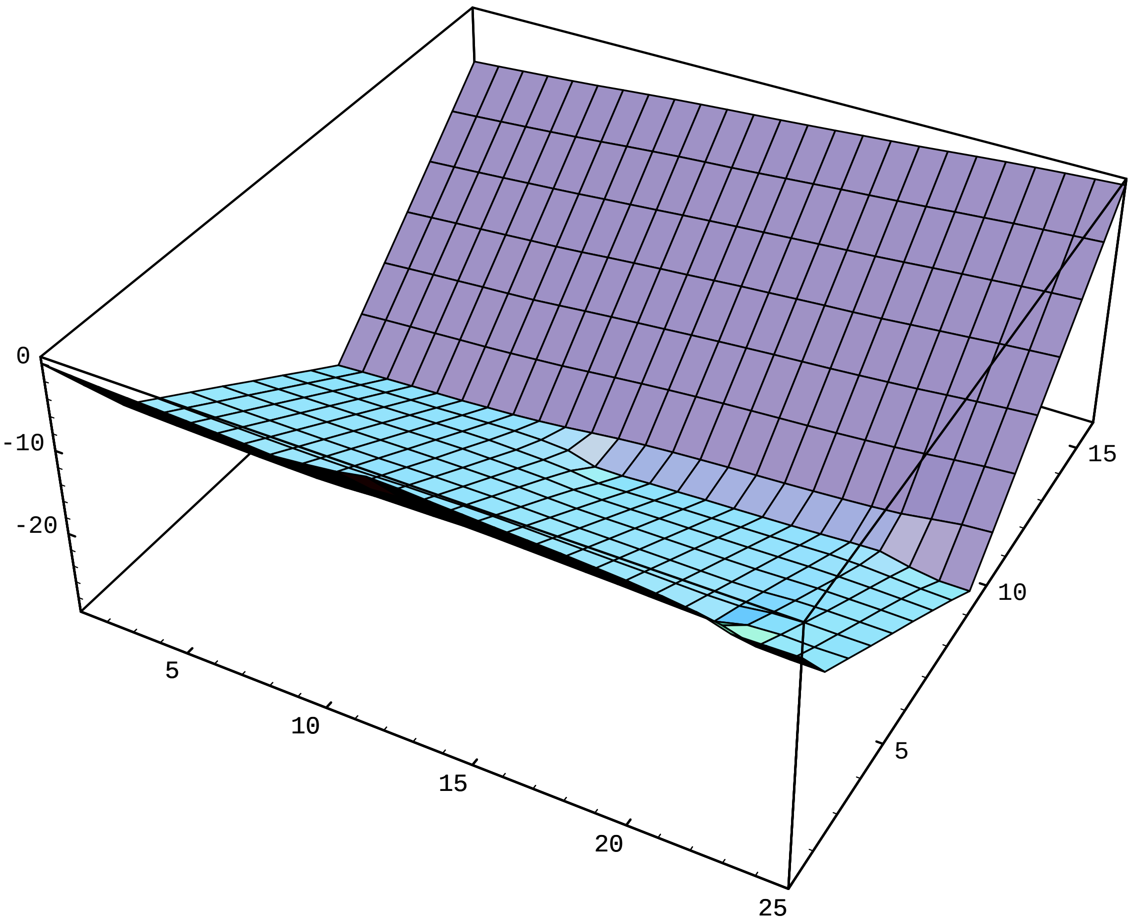

Figure IV-3: tilt angle data with a plane subtracted

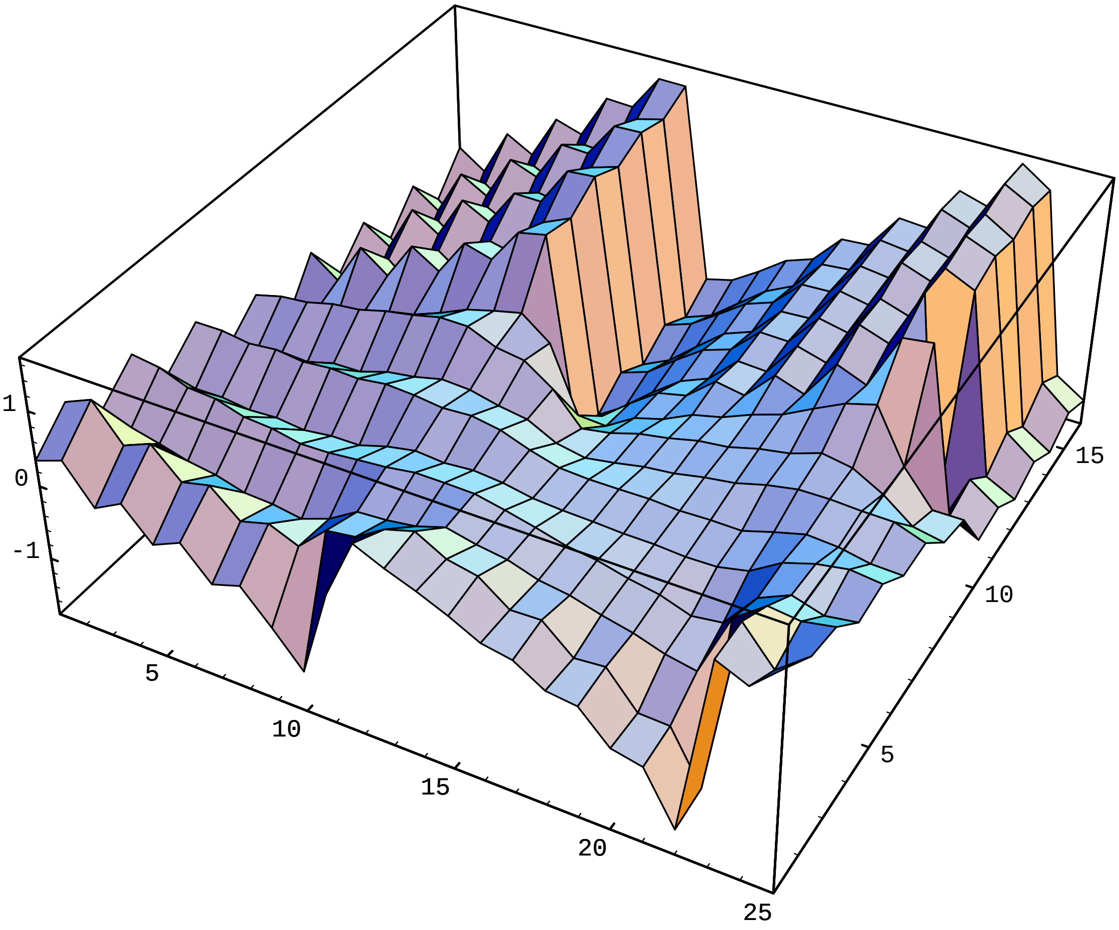

Figure IV-4: regenerated tilt angle data after postprocessing

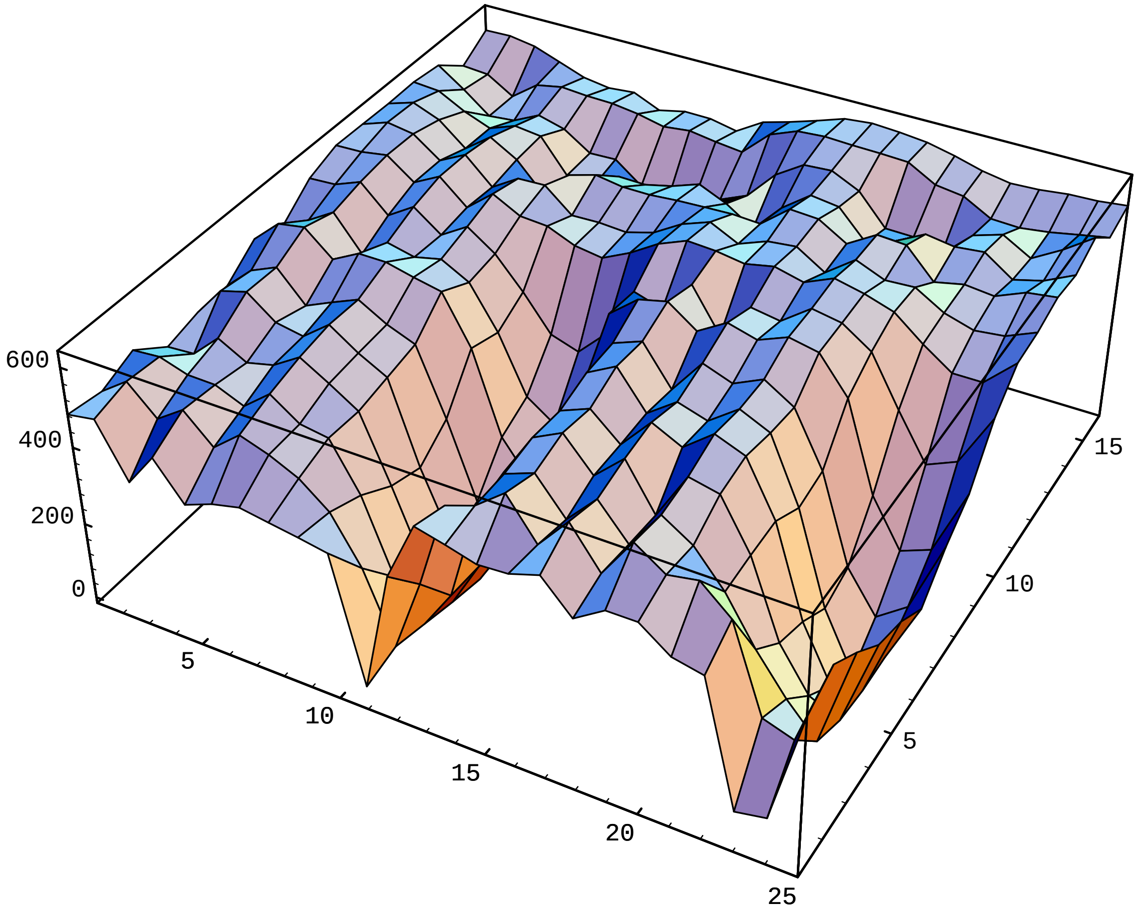

Figure IV-5: typical m_{00} dataset

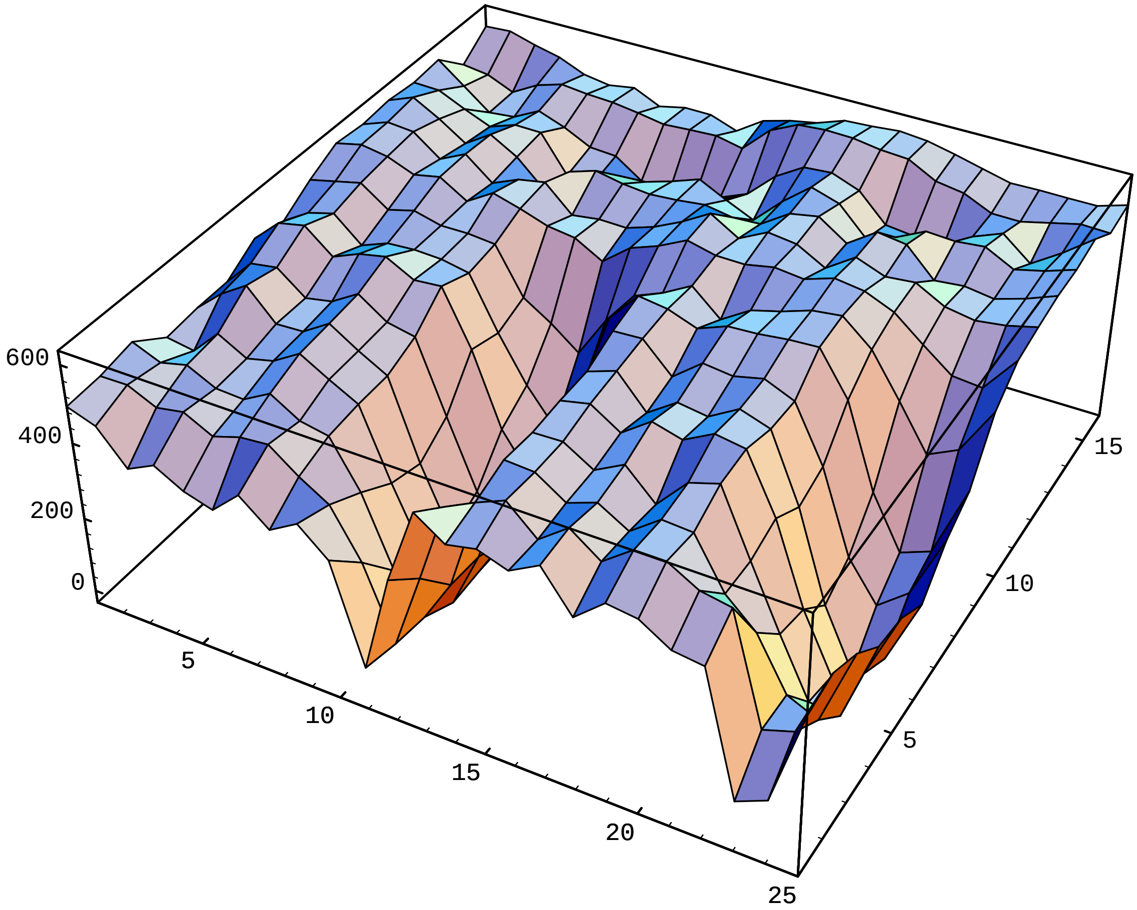

Figure IV-6: regenerated m_{00} dataset

since it can be derived from other parameters. For example, it should be possible to fit the set $\{\sigma_{xx},\sigma_{yy},\sigma_{xy}\}$ over the sphere instead of the currently used set $\{a_{00},\bar{\sigma}_x,\bar{\sigma}_y\}$, and then calculate a_{00}, $\bar{\sigma}_x$, and $\bar{\sigma}_y$ from them using the formulas of appendix I.

V. APPARATUS AND BULBS

Figure V-1 displays a schematic of the goniometer apparatus and details of the two "detector" configurations. The source is attached to the end of a rotating horizontal shaft, which holds the source above the center of a rotating horizontal platter. The shaft is also connected to the platter. The detector is placed on an independent fixed support above the platter at a controllable radius ranging between 2.22 and 5.22 inches. The goniometer operates by turning the source on the shaft in preset steps through ϕ, and collecting data at each step. Once the shaft makes a complete revolution, the platter rotates through a step in θ, moving the source with respect to the detector. The "detector" is the pinhole CCD camera. The different headlight bulb types are shown in figure V-2. In order, they are the spot quality, high beam, low beam, and dual beam bulbs.

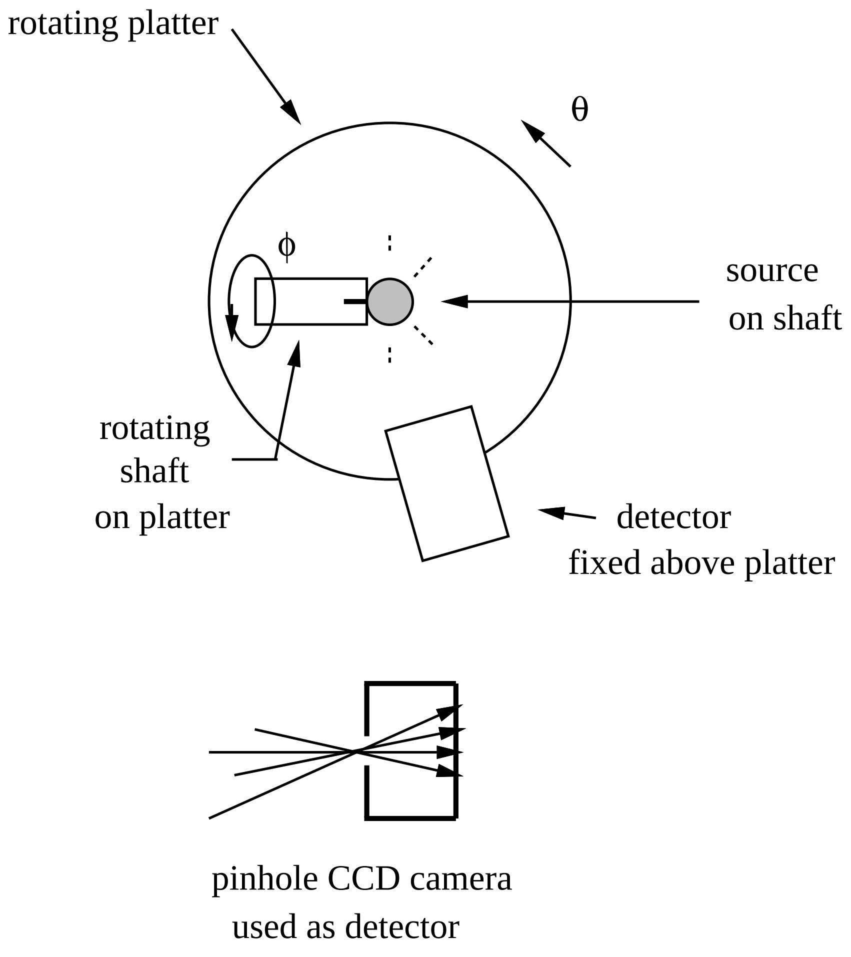

Figure V-1: a sketch of the goniometer and detector apparatus

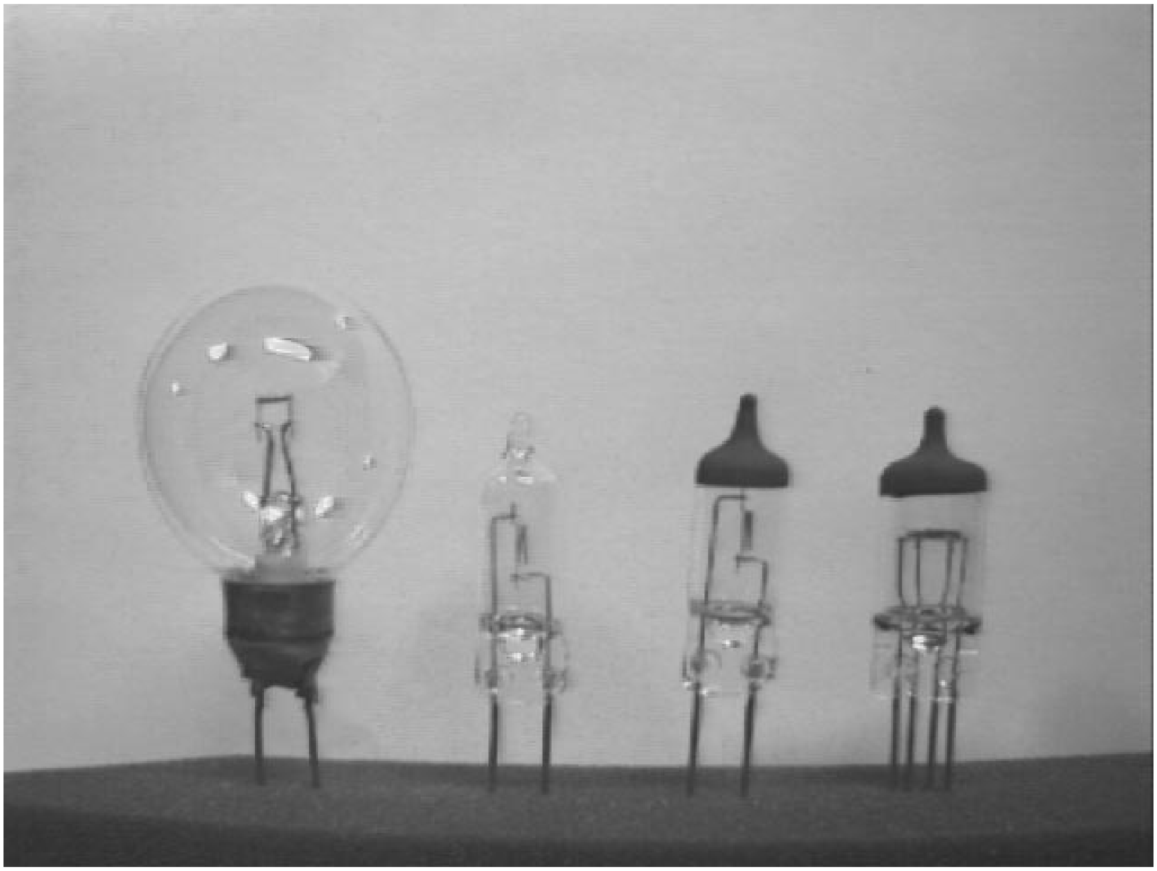

Figure V-2: spot quality, high, low, and dual beam bulbs

References

1. T. Whitted, ‘‘An Improved Illumination Model for Shaded Display’’, *Communications of the ACM*, Vol. 23, No. 6, June 1980, pp. 343-349.

2. A. S. Glassner, editor, *An Introduction to Ray Tracing,* Academic, London, 1989.

3. R. Hall, *Illumination and Color in Computer Generated Imagery,* Springer-Verlag, New York, 1989.

4. D. Rogers, *State of the Art in Computer Graphics,* Springer-Verlag, New York, 1991.

5. R. J. Donohue and B. W. Joseph, ‘‘Computer Design of Automotive Lamps With Faceted Reflectors’’, *Journal of the Illuminating Engineering Society*, Vol. 2, October 1972, pp. 36-42.

6. R. J. Donohue and B. W. Joseph, ‘‘Computer Synthesized Filament Images From Reflectors and Through Lens Elements for Lamp Design and Evaluation’’, *Applied Optics*, Vol. 14, No. 10, 1975, pp. 2384-2390.

7. R. E. Levin, ‘‘Photometric Characteristics of Light-Controlling Apparatus’’, *Illuminating Engineering*, Vol. 66, No. 4, 1971, pp. 205-215.

8. C. Verbeck and D. Greenberg, ‘‘A Comprehensive Light-Source Description for Computer Graphics’’, *IEEE Computer Graphics and Applications*, Vol. 4, July 1984, pp. 66-75.

9. I. Ashdown, ‘‘Near-Field Photometry: A New Approach’’, *Journal of the Illuminating Engineering Society,* Vol. 22, No. 1, 1993, pp. 163-180.

10. P. Y. Ngai, ‘‘On Near-Field Photometry’’, *Journal of the Illuminating Engineering Society*, Vol. 16, No. 2, 1987, pp. 129-136.

11. P. Y. Ngai et al., ‘‘Near-Field Photometry: Measurement and Application for Fluorescent Luminaires’’, *Journal of the Illuminating Engineering Society*, Vol. 21, No. 2, 1992, pp. 68-83.

12. R. G. Mistrick and C. R. English, ‘‘Study of Near-Field Indirect Lighting Calculations’’, *Journal of the Illuminating Engineering Society,* Vol. 19, No. 2, 1990, pp. 103-112.

13. O. Myodo and M. Karino, ‘‘A New Method for Computer Aided Design of Luminaire Reflectors’’, *Journal of the Illuminating Engineering Society*, Vol. 11, No. 1, 1982, pp. 98-105.

14. I. Ashdown, ‘‘Non-Imaging Optics Design Using Genetic Algorithms’’, *Journal of the Illuminating Engineering Society,* Vol. 23, No. 1, 1994, pp. 12-21.

15. W. B. Elmer, *Optical Design of Reflectors,* published by author, Andover, MA, 1974.

16. F. E. Nicodemus, ‘‘Radiance’’, *American Journal of Physics*, Vol. 31, 1963, pp. 368.

17. G. P. Tolstov, *Fourier Series,* Prentice Hall, Inc., Englewood Cliffs, NJ, 1962.

18. F. S. Acton, *Numerical Methods That Work,* Mathematical Association of America, Washington, D.C., 1990.

19. T. Watanabe and H. Kawashima, ‘‘Development of Smoothly Curved Reflector for Headlamps’’, in *Vehicle Lighting Design for Optimal Visibility and Performance (SP-857),* D. Hoffmeister, ed., Society of Automotive Engineers, 1991, pp. 69-75.

20. M. J. D. Powell, ‘‘A Survey of Numerical Methods for Unconstrained Optimization’’, *SIAM Review*, Vol. 12, No. 1, January 1970, pp. 79-97.

21. R. P. Brent, ‘‘Algorithms for Finding Zeros and Extrema of Functions Without Calculating Derivatives’’, Tech. report, Stanford University, 1971.

22. R. P. Brent, *Algorithms for Minimization Without Derivatives,* Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1973.

23. M. J. D. Powell, ‘‘An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives’’, *Computer Journal*, Vol. 7, 1964, pp. 155-162.

24. P. R. Bevington, *Data Reduction and Error Analysis for the Physical Sciences,* McGraw-Hill, New York, 1969.

25. R. W. Hamming, *Numerical Methods for Scientists and Engineers,* McGraw-Hill, New York, 1973.

26. D. Gabor, ‘‘Theory of Communication’’, *Journal of the Institution of Electrical Engineers, part III*, Vol. 93, No. 26, November 1946, pp. 429-457.

27. R. Fletcher, ‘‘Function Minimization Without Evaluating Derivatives - A Review’’, *Computer Journal*, Vol. 8, 1965, pp. 33-41.

28. W. I. Zangwill, ‘‘Minimizing a Function Without Calculating Derivatives’’, *Computer Journal*, Vol. 10, 1967, pp. 293-296.

29. D. Kahaner, *Numerical Methods and Software*, Prentice Hall, Englewood Cliffs, NJ, 1988.

30. G. E. Forsythe, *Computer Methods for Mathematical Computations,* Prentice-Hall, Englewood Cliffs, NJ, 1977.

31. G. H. Golub and C. Reinsch, ‘‘Singular Value Decompsition and Least Squares Solutions’’, *Numerische Mathematik*, Vol. 14, 1970, pp. 403-420.

32. W. H. Press et al., *Numerical Recipes in C,* Cambridge University Press, Cambridge, 1988.

33. L. G. Roberts, ‘‘Machine Perception of 3-Dimensional Solids’’, in *Optical and Electro-optical Information Processing,* J. T. Tippet et al., eds., MIT Press, Cambridge, MA, 1965, pp. 159-197.

34. W. K. Pratt, *Digital Image Processing,* John Wiley & Sons, New York, 1978.

35. R. J. Marks, *Introduction to Shannon Sampling and Interpolation Theory,* Springer-Verlag, New York, 1991.

Table of Contents

List of Figures