Generalized Camera Calibration Including Fish-Eye Lenses

DONALD B. GENNERY
1881 Alpha Road #17, Glendale, CA 91208
(under consulting contract to the Jet Propulsion Laboratory, California Institute of Technology)
dgennery@earthlink.net

Abstract. A method is described for accurately calibrating cameras including radial lens distortion, by using known points such as those measured from a calibration fixture. A way of adding decentering distortion also is described. Both the intrinsic and extrinsic parameters are calibrated in a single least-squares adjustment, but provision is made for including old values of the intrinsic parameters in the adjustment. The distortion terms are relative to the optical axis, which is included in the model so that it does not have to be orthogonal to the image sensor plane. These distortion terms represent corrections to the basic lens model, which is a generalization that includes the perspective projection and the ideal fish-eye lens as special cases. The position of the entrance pupil point as a function of off-axis angle also is included in the model. A priori standard deviations can be used to apply weight to given initial approximations (which can be zero) for the distortion terms, for the difference between the optical axis and the perpendicular to the sensor plane, and for the terms representing movement of the entrance pupil, so that the solution for these is well determined when there is insufficient information in the calibration data. For the other parameters, initial approximations needed for the nonlinear least-squares adjustment are obtained in a simple manner from the calibration data and other known information. (Weight can be given to these also, if desired.) Wild points among the calibration data are removed by means of automatic editing based on analysis of the residuals. The use of the camera model also is described, including partial derivatives for propagating both from object space to image space and vice versa.

Keywords: camera calibration, lens distortion, fish-eye lens, entrance pupil

1. Introduction

1.1. Background

Geometric camera calibration is concerned with the problem of determining the accurate mapping between the three-dimensional coordinates of viewed points and the two-dimensional coordinates of the corresponding points in an image sensor. A camera model consists of parameters that define this mapping. They often are divided into extrinsic parameters, which depend upon the pose (position and orientation in three dimensions) of the camera as a whole, and intrinsic parameters, which depend upon the internal nature of the lens and sensor. (Usually in photogrammetry, the former are called "exterior orientation" and the latter are called "calibration," but in computer vision the term "calibration" often includes both, and the term is used that way here.) Many camera calibration techniques have been used; some of them are described in a recent book (Gruen and Huang, 2001) and in a fairly recent historical survey with emphasis on photogrammetry (Clarke and Fryer, 1998).

The basic camera model that has been used since the 1970s at the Jet Propulsion Laboratory (JPL) for robotics work was originally developed at JPL by Yakimovsky and Cunningham (1978). It included a perspective (central) projection and an arbitrary affine transformation in the image plane, but it did not include lens distortion. In 1986, a better method of calibrating that model was developed, using a rigorous least-squares adjustment (Gennery et al., 1987). In 1990, that camera model and the method for its calibration were extended to include radial lens distortion (Gennery, 1991). Further improvements were made in 1992. That improved version has been become fairly widely used at JPL, and it has been described in a recent publication (Gennery, 2001). In 1994, the present author devised a way of including old information about the intrinsic camera model (with appropriate weight) when a camera model adjustment is done with a new camera pose. In 1998, Yalin Xiong devised a variation of the camera model for fish-eye lenses. Recently, the present author devised a way of incorporating movement of the entrance pupil into the camera model, which necessitated revising the mathematical formulation. In the process, the distortion model was generalized to include normal lenses, fish-eye lenses, and others as special cases. This paper describes the generalized model, the adjustment algorithm, and the mathematics for the use of the camera model. It also shows how it reduces to the older version as a special case. A method of measuring the calibration data (finding the dots in images of a calibration fixture) was previously described (Gennery et al., 1987).

The methods described herein will be used to calibrate the cameras to be used on the Mars Exploration Rovers planned for launch in 2003. Several cameras will be used on each rover (Eisenman et al., 2001; Smith et al., 2001). Some of these cameras adhere closely to the linear

perspective model, with very low distortion. However, the hazard avoidance cameras (HazCams) adhere fairly closely to the ideal fish-eye model, in which off-axis distance in image space is proportional to off-axis angle in object space. These HazCams have a field of view of 180°, and they use a 1024-by-1024 CCD which covers the full field of view on the diagonal. Since they will be used for stereo vision, it is important to calibrate them accurately.

The camera model used in the main part of this paper should be adequate for producing accurate geometric data, unless there is appreciable nonradial distortion, such as might be produced by distortion in the image sensor itself or by a lens with badly misaligned elements. For example, if there is decentering in the lens (noncollinearity of the separate lens elements) a more general type of distortion would be produced (Brown, 1966). However, a small amount of decentering mostly causes effects that can be taken up by the other camera model parameters. Therefore, at least for well-built narrow-angle lenses, the accuracy needed for computer vision can be achieved. However, the angular accuracy needed for photogrammetry typically is around 10 to 100 times greater, so allowance for decentering there can be important. Also, as CCDs increase in size, the time may come when much higher resolutions are routinely used in computer vision, so that decentering may become important there too, especially for wide-angle lenses. A way of including decentering distortion in the model is described in Section 7.2.

CCDs (unlike older types of television cameras) are not subject to appreciable general geometric distortion. A bias in the timing of the digitization of the image from a CCD merely shifts the apparent projection of the optical center in the image sensor plane, and this point is adjusted in the model. The model also allows for different scale factors along the different image coordinates. In fact, any error that is a linear function of the position in the image is subsumed by terms included in the camera model.

The calibration method presented here applies at only one lens setting. For a zoom lens, a separate calibration would have to be done at each zoom setting. Similarly, if focus is changed, a separate calibration is needed at each focus setting. (Brown (1971) discusses the effects on distortion of changing focus.) These factors are not a concern for the Mars rover cameras, since their focus is fixed in the design. If there is appreciable lateral chromatic aberration in the lens and it is desired to take images in different colors, separate calibrations might need to be done at different wavelengths. However, the Mars rover cameras have fairly low chromatic aberration, and all except the panoramic cameras use only one fixed wavelength band. If there is appreciable coma and the aperture is changed, the calibration might be affected slightly, but the Mars rover cameras have a fixed aperture. (Willson and Shafer (2001) discuss the modeling of all of these effects.)

The lens elements in the Mars rover cameras will be accurately aligned. Therefore, there should not be much decentering. (Section 8 shows some ray tracing results including

decentering in the HazCam.)

Other potential concerns for operations on Mars are that the calibration might change at widely different temperatures, due to thermal expansion, or at low atmospheric pressure, due to the different ambient index of refraction. Simulations using ray tracing for the HazCams have shown that the effects in the corners of the image will be about 2/3 of a pixel from temperature and about 1/3 of a pixel from pressure. The total effect is one about pixel at -55°C on Mars compared to 23°C on Earth. Since these changes are so small, the ray tracing results could be used to change the calibration from that obtained in a laboratory on Earth to make it appropriate for conditions on Mars at particular temperatures (if it is decided that the effect is large enough to bother with at all). That would be much easier than doing separate calibrations on Earth in a laboratory that reproduces the extreme conditions on Mars.

1.2. Notation

Column matrices (general vectors) are denoted by boldface lower-case letters, and rectangular matrices are denoted by boldface capital letters. (Physical vectors in three-dimensional space are a special case of column matrices.) The derivative of a column matrix with respect to a scalar is a column matrix, the derivative of a scalar with respect to a column matrix is considered to be a row matrix, and the derivative of a column matrix with respect to a column matrix is a rectangular matrix. Scalar variables are denoted by italics (or Greek) letters, either capital or lower-case. Since vectors are represented as column matrices, for any two 3-vectors $\mathbf{u}$ and $\mathbf{v}$ the inner (dot) product $\mathbf{u}\cdot\mathbf{v}$ is equivalent to $\mathbf{u}^T\mathbf{v}$, and the outer product (which produces a 3-by-3 matrix) is written $\mathbf{u}\mathbf{v}^T$. The vector cross product is written $\mathbf{u}\times\mathbf{v}$. For any 3-vector $\mathbf{v}$, its length ($\sqrt{\mathbf{v}\cdot\mathbf{v}}$) is represented by $|\mathbf{v}|$, and the unit vector in its direction ($\mathbf{v}/|\mathbf{v}|$) is represented by unit($\mathbf{v}$). $\mathbf{I}$ denotes the identity matrix of appropriate size. $\mathbf{O}$ denotes a zero rectangular matrix of size appropriate to the context, and $\mathbf{0}$ denotes a zero column matrix of appropriate size.

The subscript i usually will be used on quantities that are associated with individual three-dimensional points, in order to distinguish them from quantities that depend only upon the camera model. A prime (′) is used to denote a value for an apparent point that is shifted from an actual three-dimensional point because of effects such as distortion, and for an entrance-pupil that has shifted from its nominal position. (These effects are described in Section 2.) A tilde (˜) denotes values for the intrinsic camera model from a previous camera model adjustment (as explained in Section 4). A circumflex (ˆ) denotes computed values, as opposed to measured values, of the image coordinates x and y (as explained in Section 2). Symbols that use the same letter but differ in whether they are boldface or whether they are capital do not represent related

quantities unless they are so defined explicitly.

Symbols for individual quantities will be defined as needed. The symbols used here are chosen to agree for the most part with those in the previous publication (Gennery, 2001). Some of the important ones related to the camera model are shown in Fig. 1. Some of the ones related to least-squares adjustments (**N** for the coefficients in the normal equations, **t** for the "constants" in the normal equations, and **M** for a matrix involved in the constraints) are the same as in the cited least-squares reference (Mikhail, 1976), but the others are not. Especially note the following substitutions from Mikhail to here: $\mathbf{B} \rightarrow \mathbf{A}$ for the coefficients in the linearized condition equations, and $\mathbf{C} \rightarrow \mathbf{K}$ for the coefficients in the linearized constraint equations. (The **A** matrix in Mikhail reduces to the identity matrix in the situation here.)

The weight matrices used here are the inverses of the corresponding covariance matrices, instead of having arbitrary scaling. (Such matrices sometimes are called information matrices.)

2. Definition of Camera Model

2.1. Overview of Camera Model

The camera model used here includes three effects that depart from a linear perspective projection model and that can become important as the angle between the optical axis and an incoming ray increases. These include the basic lens model, radial distortion, and movement of the entrance pupil. (The effect of the basic lens model often is included as part of radial distortion, but it is convenient to separate them here.)

Computations that take into account the basic lens model and radial distortion are done to convert the actual position $\mathbf{p}_i$ of a point in three-dimensional space into an apparent point $\mathbf{p}'_i$ that would produce the same result with the Yakimovsky and Cunningham (1978) perspective model as does the actual point with the full lens model. The steps in this conversion are shown in Fig. 1, and they will be discussed in further in Sections 2.3 and 2.4.

One of the important properties of any lens is the position of the entrance pupil point. This is the point towards which incoming chief rays (each of which is the center ray of a bundle of rays that make it through the lens and aperture stop) are headed (extended as a straight lines). Therefore, this is the position from which the camera seems to view the world. This position often is referred to as the perspective center. Sometimes, this is assumed to be the first nodal point of the lens (the same as the first principal point if the medium is the same on both sides of the lens). For many lenses, the pupil points are close to the nodal points, but in general they are not. Of course, mathematically the camera model is defined by the equations, so the physical

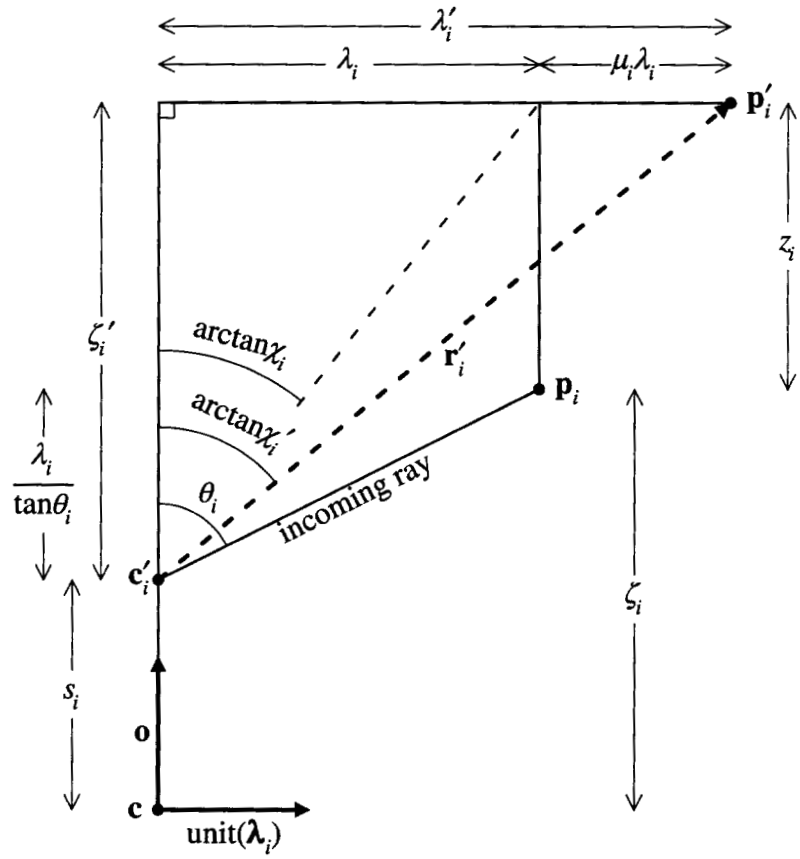

Figure 1. Illustration of the definitions of important quantities. The projected entrance pupil point shifts outwards along the optical axis by the amount s_i. The apparent 3D point shifts outwards by the amount z_i according to the basic lens model ($z_i < 0$ if $L > 1$), and sideways by the amount $\mu_i\lambda_i$ because of other radial distortion. (The diagram is scaled for the 3D-to-2D projection or for the camera model adjustment. In the 2D-to-3D projection, $\zeta_i' \approx 1$ if $\mathbf{a} \approx \mathbf{o}$, a unit vector $\mathbf{r}_i$ is generated along $\mathbf{p}_i - \mathbf{c}_i'$, and $\mathbf{p}_i$ is not used.)

meaning of the parameters does not matter for calibration purposes, as long as they are able to capture the degrees of freedom in the physical situation, and as long as the same equations are used in the camera model adjustment and in the use of the camera model. (For more information about the above terms, see any optics textbook, for example Jenkins and White (1976)).

However, the entrance pupil is not necessarily at a constant point. Therefore, we must convert the nominal entrance pupil point $\mathbf{c}$ into an actual point $\mathbf{c}_i'$ that can be given to the Yakimovsky and Cunningham model. This effect is shown in Fig. 1, and it will be discussed further in Section 2.2.

In Fig. 1, the symbol $\mathbf{o}$ denotes a unit vector along the optical axis of the lens, $\mathbf{c}$ denotes the position of the entrance pupil for on-axis rays, $\mathbf{p}_i$ denotes the three-dimensional position of any point being viewed, and θ_i denotes the angle between the optical axis and an actual incoming ray (approximately a chief ray, but actually whatever ray corresponds to the point within the point-spread function that is determined by the detection algorithm that is used). The orthogonal displacement of the point $\mathbf{p}_i$ from the optical axis is denoted by the vector $\boldsymbol{\lambda}_i$. The magnitude of this vector is λ_i, which thus is the component of $\mathbf{p}_i - \mathbf{c}$ orthogonal to $\mathbf{o}$. The component of $\mathbf{p}_i - \mathbf{c}$ parallel to $\mathbf{o}$ is called ζ_i.

The shifted points $\mathbf{p}_i'$ and $\mathbf{c}_i'$, in the form of the vector $\mathbf{r}_i' = \mathbf{p}_i' - \mathbf{c}_i'$, are given to the Yakimovsky and Cunningham equations, which use the model vectors $\mathbf{a}$, $\mathbf{h}$, and $\mathbf{v}$ to project $\mathbf{p}_i'$ into the image. This process will be discussed in Section 2.5.

The $\mathbf{a}$ vector in the Yakimovsky and Cunningham model and the $\mathbf{o}$ vector usually are nearly equal, since the optical axis usually is nearly perpendicular to the plane of the image sensor. In such a case, if the amount of lens distortion is small, so that the lens nearly corresponds to the ideal perspective projection, the difference between these two vectors could be ignored, as it often is in camera calibration. However, if there is a large amount of distortion, the difference between these two vector can become very important in the calibration, even if it is small.

2.2. Moving Entrance Pupil

Although for an ideal lens the entrance pupil is fixed, for actual lenses the position of the entrance pupil can change as the off-axis angle of an incoming ray increases. For most lenses this effect is small, and thus it usually is ignored in camera calibration; but for fish-eye lenses it is quite large (on the order of the size of the lens), and this shift can be very important for objects that are close to the lens. (For the HazCams on the Mars rover, the forward shift of the entrance pupil is about 7 mm, and it is planned to view objects as close as 100 mm. Therefore, the error could be around 0.07 radians if not corrected.) Fig. 2 illustrates this effect. The solid lines represent incoming rays that pass through the center of the entrance pupil for different incident

angles. The caustic formed by these rays (heavy dashed line in the figure) is the locus of the actual entrance pupil points. However, all that we care about is the projection of these points onto the optical axis, i.e. the intersection of the rays with the optical axis. (This intersection must exist, since we assume that the pupil shift is radially symmetrical, even though that assumption is not exactly true in the presence of lens decentering.) Therefore, we will model the shift of this intersection point (relative to the limiting point for zero angle) as a function of angle, the magnitude of this shift will be called s_i, and the projected shifted entrance pupil point will be called $\mathbf{c}_i'$. Therefore,

$$\mathbf{c}_i' = \mathbf{c} + s_i\mathbf{o} \tag{1}$$

It might be thought that, to represent s as a function of the off-axis angle θ_i, a simple polynomial could be used. However, as the angle becomes greater than 90°, the shift becomes very large, and it becomes infinite as θ_i approaches 180°. This behavior is difficult to model as a polynomial. What is needed is a simple function that approximates the actual behavior fairly well, so that a polynomial is needed only to apply a small correction. A few functions were tried and are shown in Fig. 3. In the limit for small θ_i, the shift is proportional to θ_i^2, and all of the functions shown obey this limit. Therefore, to see the effects more clearly, the shifts on the plot have been divided by θ_i^2. This reduces the dynamic range on the plot by causing the plotted values to become constant at small angles.

Also shown on the plot are some actual published wide-angle lens designs (Laikin, 2001) with the stated fields of view, the design for the HazCam for the 2003 Mars rovers, and two hypothetical designs. The "spherical mirror" plot represents the effect of viewing the reflection in a spherical mirror from an infinite distance (which can produce a 360° field of view), and this produces the equal-area projection. The "Snell's law" plot represents the effect of having a flat slab of refracting material (whose index of refraction n_{refr} here is 1.6), with a hemisphere on the back surface centered on an aperture, so that the sine of the exit angle is equal to the sine of the entrance angle divided by n_{refr} (which produces a 180° field of view). (Lenses approximating this effect have actually been built.)

One of the functions shown in the figure is simply θ_i^2, which becomes a constant after the division. Therefore, this plot coincides with the horizontal axis. It can be seen how inappropriate this is for matching the actual designs, in agreement with the above statement about using only a polynomial. The other three functions tried all become infinite at 180°, as desired. Of these, the two upper ones ($\tan^2\theta_i/2$ and $\theta_i/\sin\theta_i - 1$) seemed to be the best, especially the latter for the spherical mirror and the Laikin lens with the widest field of view. The latter function was chosen, for this reason and because all of the actual designs could be matched easily with polynomial corrections to this model, as shown in Fig. 4. (Note that the limit of

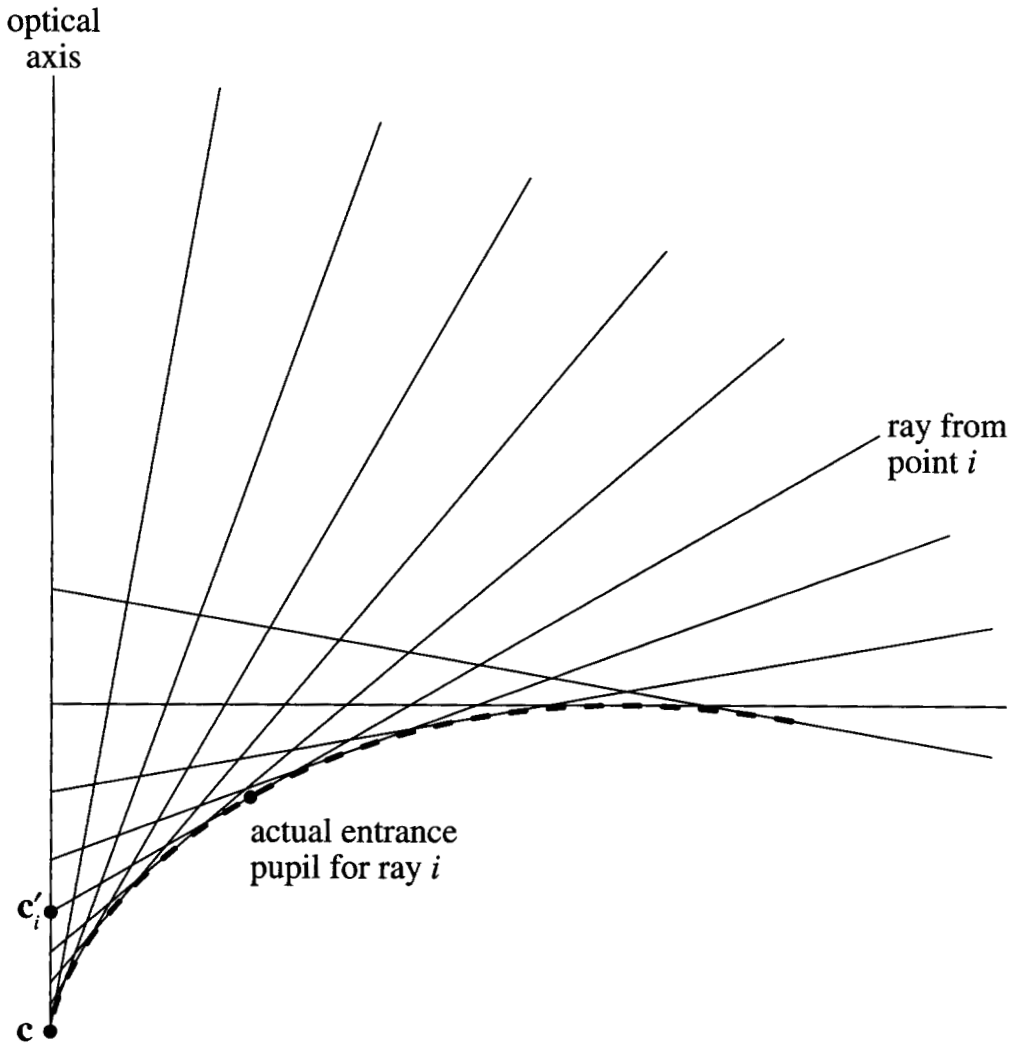

Figure 2. Illustration of shift of entrance pupil. The point **c** is the center of the entrance pupil for on-axis rays. The thick dashed line represents the locus of the entrance pupil as the angle of the incident chief ray changes. The intersection of the actual ray (shown for every 10° up to 100°) with the optical axis is used as the shifted point $\mathbf{c}'_i$.

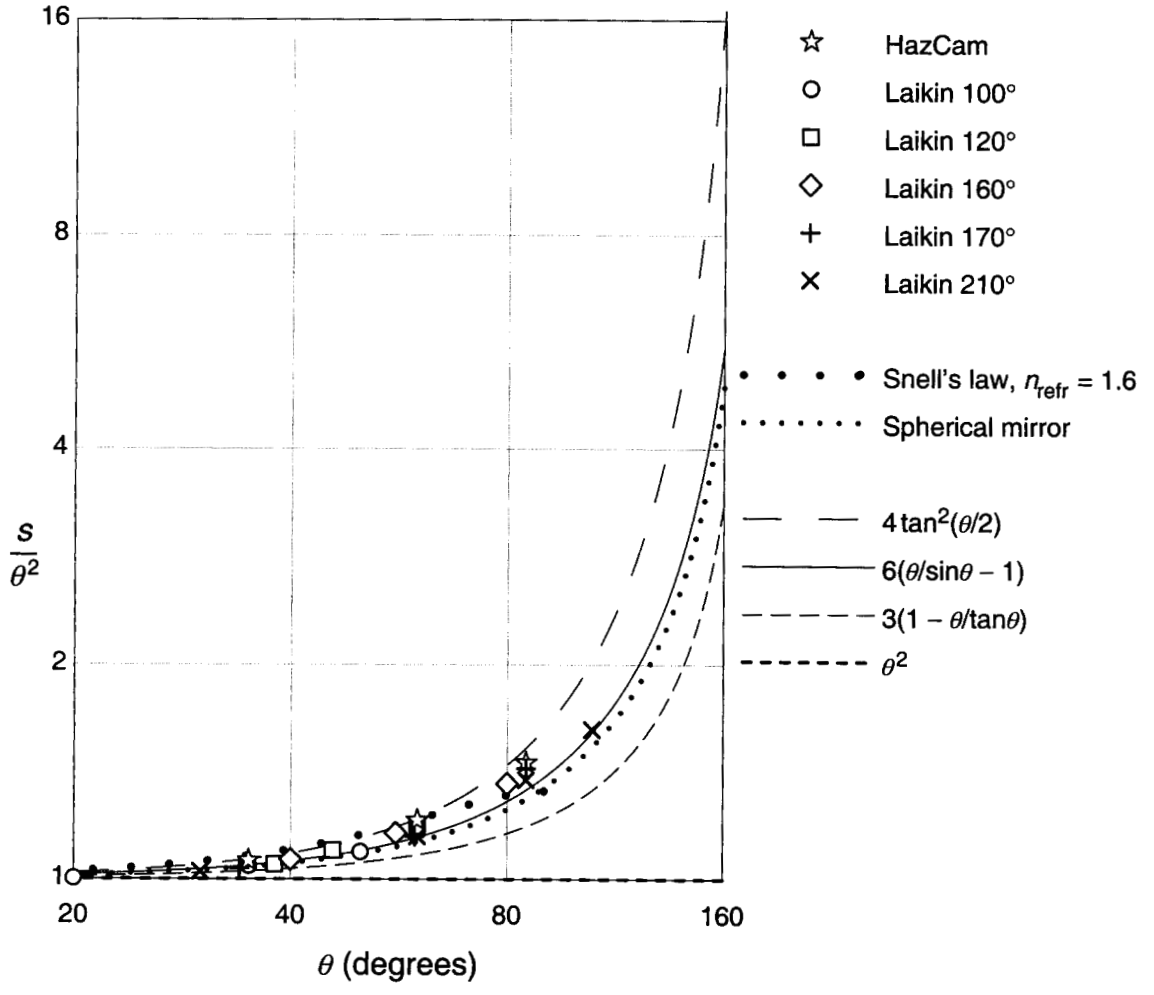

Figure 3. Logarithmic plot of a few functions examined for the basic pupil-shift model. The shift of the projection of entrance pupil along the optical axis is shown, divided by the square of the angle in radians (normalized to equal unity at small angles). Also shown are points for some specific wide-angle designs.

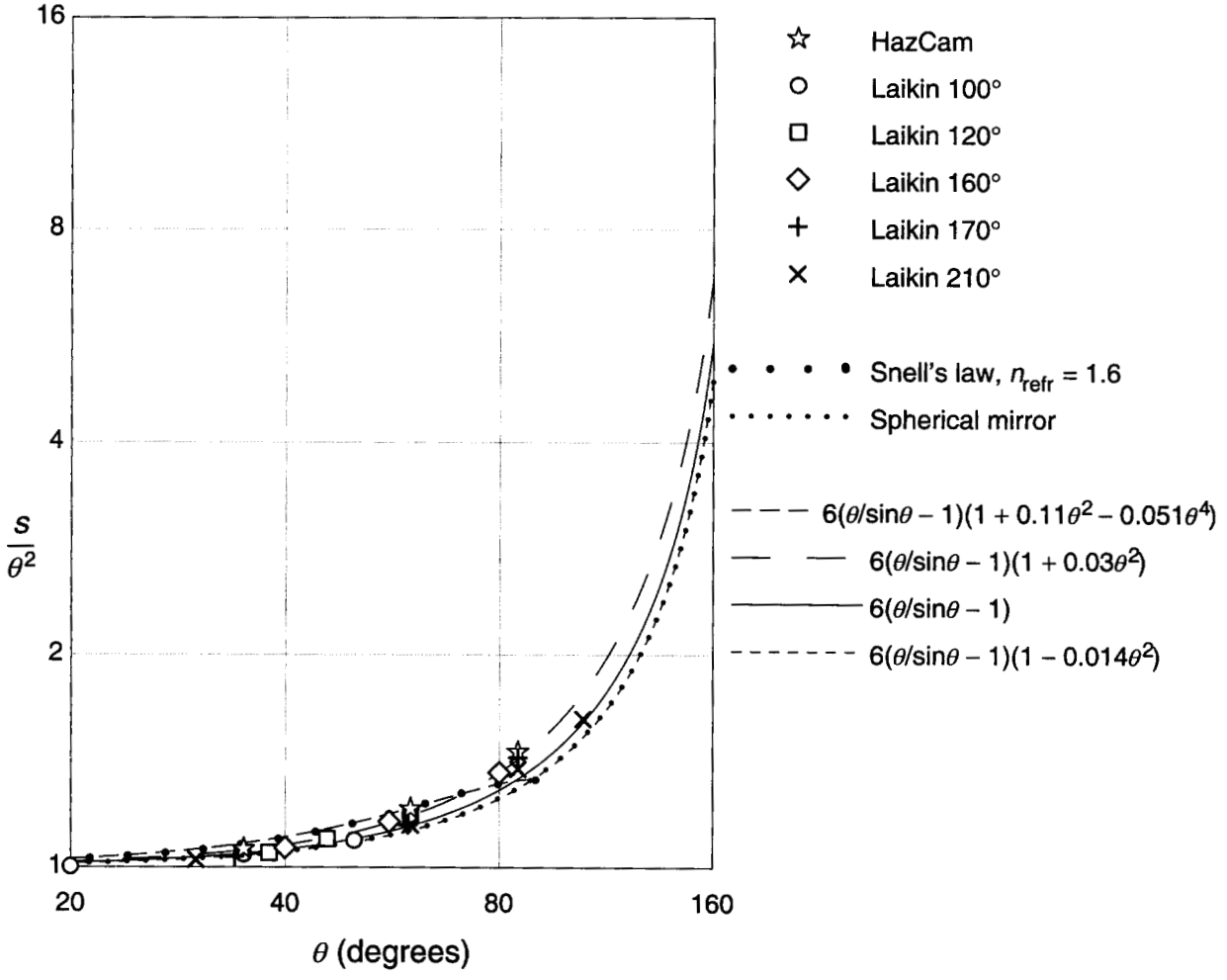

Figure 4. Logarithmic plot of a few examples of polynomial corrections to the chosen basic pupil-shift model, shown as in Fig. 3, with the same specific wide-angle designs.

$\theta_i/\sin\theta_i - 1$ as θ_i approaches zero is $\theta_i^2/6$. This is the reason for the factor of 6 shown on the plots. In practice, this factor is subsumed by the zero-degree term in the polynomial correction.)

In applying a polynomial correction to the chosen function, it was found that multiplying by a polynomial in θ_i was much better than using a polynomial in the chosen function. (Otherwise, the increase at large angles was excessive.) Therefore, the selected model for the pupil shift is as follows:

$$s_i = \left(\frac{\theta_i}{\sin\theta_i} - 1 \right) (\varepsilon_0 + \varepsilon_1 \theta_i^2 + \varepsilon_2 \theta_i^4 + \ldots) \tag{2}$$

where only even-order terms are used because of symmetry, and where the epsilons are camera calibration parameters to be determined. The number of epsilons that are used is called E, and this is a manually set value. The highest-order epsilon is ε_{E-1}, and the polynomial is of degree $2E-2$, which is equivalent to degree $2E$ when combined with the second-degree nature of the basic function. (Currently we use $E = 3$ for fish-eye lenses, which is more than enough to achieve high accuracy with the HazCam. In fact, simulations indicate that $E = 1$ would produce an error of only around a tenth of a pixel at a distance of 100 mm with this camera.)

Fig. 4 shows some examples of the combined pupil-shift model. It can be seen that $E = 2$ can match most of the designs fairly well, but $E = 3$ is needed for the Snell's law lens.

From Fig. 1 it can be seen that

$$s_i = \zeta_i - \frac{\lambda_i}{\tan\theta_i} \tag{3}$$

This relationship will useful later.

2.3. Basic Lens Model

Usually, radial distortion is represented by a polynomial that represents a departure from the perspective model (e.g. Shah and Aggarwal, 1996). However, some wide-angle lenses depart very far from this model, so that a high-degree polynomial would be needed; and, if the field of view reaches 180°, the distortion defined in this way can become infinite. (Fish-eye lenses can have a field of view that exceeds 180°.) In such a case, it is better to adopt a simple model that more closely approximates the actual lens and to let the polynomial represent a departure from this ideal model. For example, for an ideal fish-eye lens, the off-axis image coordinate is proportional to the incoming off-axis angle, instead of the tangent of this angle.

Here, we adopt a generalization of the basic lens model that includes the perspective

projection and the ideal fish-eye lens as special cases, as controlled by the parameter L (for "linearity"). As above, θ_i is the angle that the incoming ray makes with the optical axis. Let χ_i be the tangent of the corresponding angle of an apparent ray that would produce the equivalent result in a perspective projection. From Fig. 1, it can be seen that the effect of the basic lens model is to shift the apparent three-dimensional point outwards parallel to the optical axis by an amount z_i, where

$$z_i = \frac{\lambda_i}{\chi_i} - \frac{\lambda_i}{\tan\theta_i} \tag{4}$$

Then the basic lens model is defined by the following convenient expression:

$$\chi_i = \frac{\sin L\theta_i}{L\cos(\max(0, L\theta_i))} \tag{5}$$

which was chosen because in practice it reduces to the following three cases:

$$\begin{aligned} \chi_i &= \frac{\sin L\theta_i}{L} && \text{if } L < 0 \\ \chi_i &= \theta_i && \text{if } L = 0 \\ \chi_i &= \frac{\tan L\theta_i}{L} && \text{if } L > 0 \end{aligned} \tag{6}$$

where the middle case has been obtained by taking the limit as L approaches zero.

Several special values of L cause the above to reduce to certain standard projections. $L = 1$ produces the usual perspective projection, $L = 0.5$ produces the stereographic projection, $L = 0$ produces the equidistant projection (ideal fish-eye lens), $L = -0.5$ produces what can be called the equal-area or equi-solid-angle projection (since equal solid angles in three-dimensional space project into equal areas in the image plane), and $L = -1$ produces what has been called the sine-law projection. (For more information about these special projections, see Miyamoto (1964) and Stevenson and Fleck (1995).) The case of $L > 1$ represents pincushion distortion, and $L < 1$ represents barrel distortion.

Fig. 5 shows plots representing several values of L. Also shown are points from the same designs shown in Fig. 3. It can be seen in the plot that, by choosing an appropriate value of L, any particular lens design of those shown can be approximated fairly well, except for the Snell's law case. We will see shortly (in Fig. 6) that even that case can be handled when the radial distortion parameters are included.

The mathematics to be presented here works for any value of L. A value of L can be chosen that is suitable for the lens being used, and then the radial distortion parameters can be adjusted automatically to correct for departures from this ideal. (In most cases, just choosing either 0 or 1 for fish-eye lenses or normal lenses, respectively, should suffice.) Also, the mathematics could

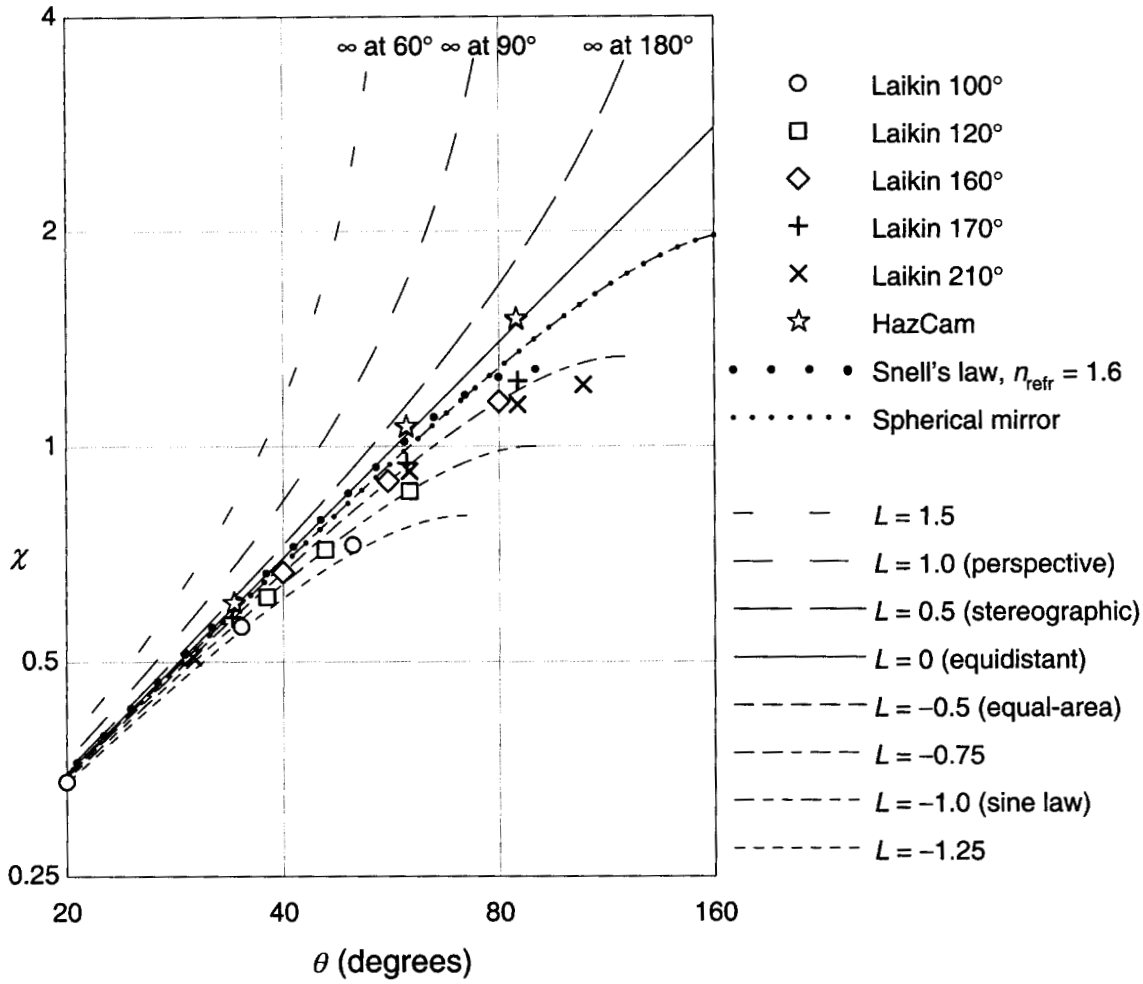

Figure 5. Logarithmic plot of basic lens model $\chi = \sin L\theta / L\cos(\max(0, L\theta))$ for various values of the parameter L, showing the image coordinate (normalized to equal θ in radians at small angles) as a function of incident off-axis angle, for ideal lenses conforming to the model. Some of the values of L are labeled with the names of corresponding standard projections. Also shown are points for some specific wide-angle designs.

include adjusting L automatically to get a better fit with less radial distortion correction (or an approximate fit with no additional distortion correction), if that is desired. However, so far there does not appear to be much need for this, so it has not been implemented, but the way to do it is described in Section 7.1.

2.4. Radial Lens Distortion

It is a common practice to model radial lens distortion as a polynomial that gives the departure of the off-axis coordinate from its ideal value as a function of the off-axis coordinate (e.g. Brown, 1971; Shah and Aggarwal, 1996). Because of symmetry, this polynomial contains only odd-order terms (or only even-order terms if it represents a proportionality correction). Usually this is a polynomial in image coordinates, but the distortion polynomial must be defined relative to the optical axis of the lens, in order for the distortion to be radial. If the image sensor plane is perpendicular to the optical axis, this is the **a** vector in the Yakimovsky and Cunningham model. However, to allow for the possibility that they are not exactly perpendicular, a separate unit vector **o** is used here for the optical axis. Note that, if there is no distortion (with $L = 1$) and the pupil points coincide with the nodal points, it is impossible for the calibration to determine the optical axis. (A perspective projection is equivalent to using a pinhole camera, and a pinhole has no axis.) Therefore, some a priori weight will be applied to tend to make the **o** vector equal to the **a** vector in the Yakimovsky and Cunningham model (which nominally is the perpendicular to the image sensor plane), so that it will be well determined when there is not much distortion. However, when there is a large distortion, the data will outweigh this a priori information, and the two vectors can differ.

The effect of radial distortion will be modeled as an apparent shift of a three-dimensional point in a direction orthogonal to the optical axis, by an amount $\mu_i\lambda_i$, so that the distance from the optical axis of the apparent point is

$$\lambda_i' = (1+\mu_i)\lambda_i \tag{7}$$

as shown in Fig. 1. The distortion polynomial is

$$\mu_i = \rho_0 + \rho_1\chi_i^2 + \rho_2\chi_i^4 + \ldots \tag{8}$$

where the rhos are camera calibration parameters to be determined. Also, we can define χ_i' as being χ_i that has been modified by radial distortion, so that

$$\chi_i' = (1+\mu_i)\chi_i = (1+\rho_0)\chi_i + \rho_1\chi_i^3 + \rho_2\chi_i^5 + \ldots \tag{9}$$

The number of rhos that are used is called R, and this is a manually set value. The highest-order rho is ρ_{R-1}, and the polynomial for μ_i is of degree $2R-2$, which is equivalent to degree $2R-1$ in (9). (Currently, we use $R = 3$.)

When $L = 1$, the type of distortion depends on the sign of μ_i for most points. The case of $\mu_i > 1$ represents pincushion distortion, and the case of $\mu_i < 1$ represents barrel distortion. (If $L \neq 1$, L is the main determinant of the type of distortion, and μ_i usually applies only a small correction.)

Note that ρ_0 does not actually represent radial distortion in the usual sense, but is merely a scale factor in the plane orthogonal to **o**, whereas the scale factors included in **h** and **v** in the Yakimovsky and Cunningham model are in a plane orthogonal to **a**. The zero-degree term represented by ρ_0, which represents the first-degree term after multiplying by λ_i as in (7) or by χ_i as in (9), is subsumed by the scale factors included in **h** and **v** if **a** is along the optical axis of the lens or if the pupil points coincide with the respective nodal points. Therefore, ρ_0 would not be needed in these cases, but it must be included in the general case.

Some a priori weight (usually a small amount) will be applied to tend to make the ρ's equal to zero, so that they will be well determined when there is not enough information in the calibration images. (This is especially important for high-order coefficients when there are not many calibration points, and for ρ_0 if **o** and **a** are nearly equal or if the pupil points nearly coincide with the nodal points.)

Fig. 6 shows the same actual and hypothetical wide-angle lens designs as Fig. 5. However, the curves are drawn only for the case of $L = 0$ (ideal fish-eye lens), but now various radial distortion polynomials are used to produce departures from this basic model. Here the independent variable is χ_i, which results from the basic lens model; but, since $L = 0$ here, $\chi_i = \theta_i$. The dependent variable is shown as χ_i', which includes the effect of the distortion polynomial according to (9).

It can be seen that using $R = 2$ (second-order radial-distortion proportionality correction) can produce a fairly good fit to the Laikin lenses and spherical mirror, and $R = 3$ (fourth-order proportionality correction) can produce a good fit to the Snell's law case. In fact, it appears that there is not much point in using values of $L < 0$ if a radial distortion polynomial is used, since the latter can produce practically the same results. However, if $L > 0$, the image coordinate becomes infinite at certain values of the angle that depend on L, as shown in Fig. 5, and this effect would be difficult to approximate with a polynomial alone. In most practical cases with reasonable fields of view, the value $L = 1$ should be able to handle these latter cases if a radial distortion polynomial is used; but the distinction between normal lenses and fish-eye lenses is large, so at least the choice between $L = 1$ and $L = 0$ is needed. (It is interesting that simulated results using ray-tracing for the HazCam indicate that $L = 0.37$ with $R = 3$ produces a better fit than does $L = 0$

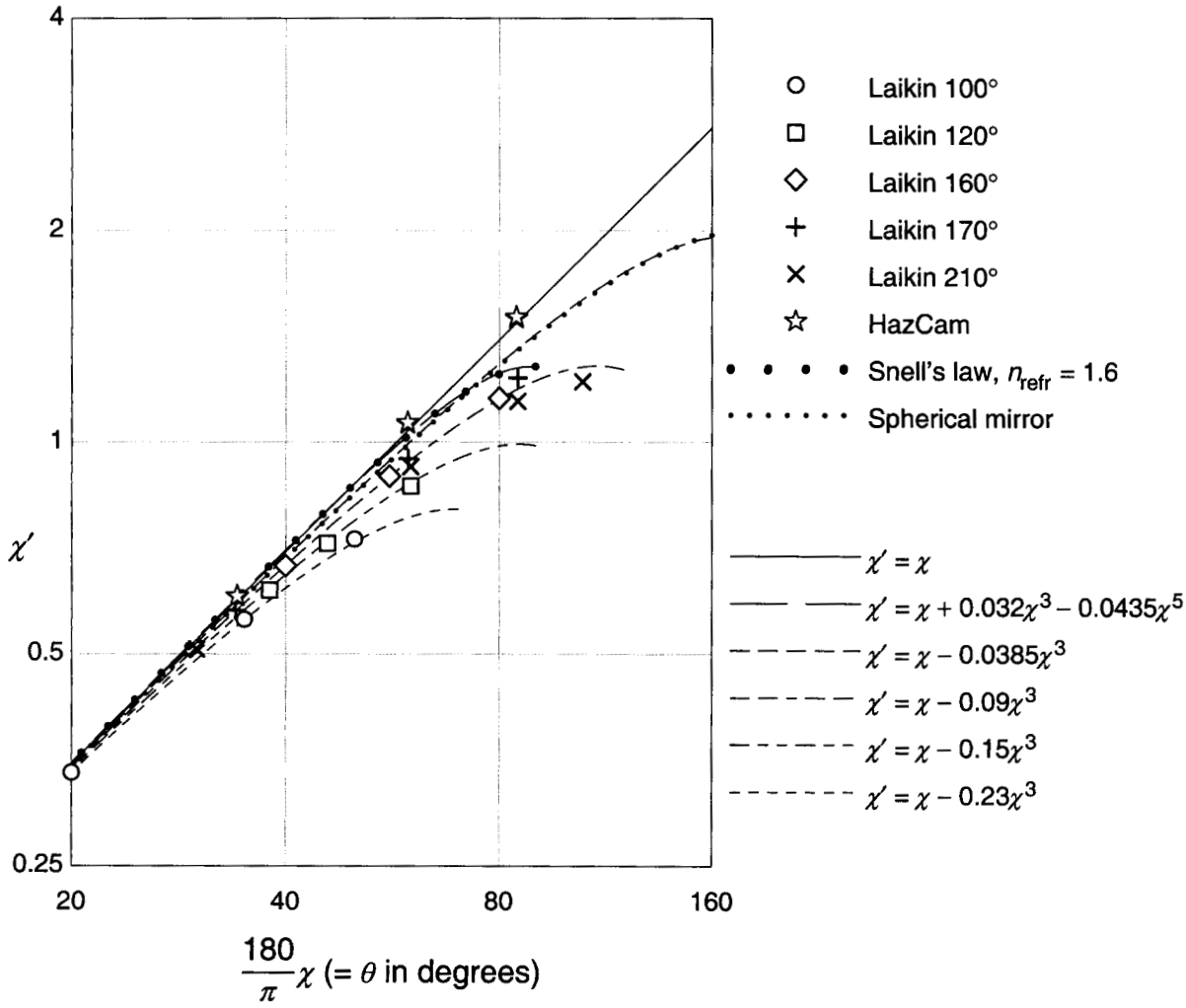

Figure 6. Logarithmic plot of a few examples of the distortion model for $L = 0$ (representing departures from the basic fish-eye model). Also shown are points for the same specific wide-angle designs shown in Fig. 5.

with $R = 4$, even though this camera is fairly close to having an ideal fish-eye lens, as can be seen in Fig. 5. Apparently this is because its trend at small angles corresponds closely to $L = 0.37$, but at large angles distortion effects push it towards the line for $L = 0$.)

The component of $\mathbf{r}_i'$ parallel to $\mathbf{o}$ is called ζ_i', and the orthogonal component is the previously mentioned λ_i'. From Fig. 1 it can be seen that

$$\chi_i = \frac{\lambda_i}{\zeta_i'} \tag{10}$$

$$\chi_i' = \frac{\lambda_i'}{\zeta_i'} \tag{11}$$

These relationships will be useful later.

2.5. Yakimovsky and Cunningham Camera Model

The Yakimovsky and Cunningham (1978) camera model consists of the four 3-vectors **c**, **a**, **h**, and **v**, expressed in object (world) coordinates. The meaning of each of these now will be described for the simple case of a pinhole camera.

The perspective center (the pinhole) normally is at position **c**. For a real lens, **c** normally represents the entrance pupil point. However, since here we consider the fact that the entrance pupil point is a function of off-axis angle, we replace **c** with $\mathbf{c}_i'$ for any actual data point.

Let a perpendicular be dropped from the perspective center (actually the exit pupil point for a real lens) to the image sensor plane, and let its point of intersection with this plane be denoted by x_c and y_c in image coordinates. Then **a** is a unit vector parallel to this perpendicular and pointing outwards from the camera to the scene. (If the image sensor is tilted relative to the lens, **a** is not the same as the **o** vector, which denotes the optical axis of the lens.)

The **h** and **v** vectors contain information about the image coordinate system in a rather complicated way. If the image axes are orthogonal and if the image origin is at x_c,y_c, then **h** and **v** merely point along the image x (horizontal) and y (vertical) axes, and the magnitude of each vector is equal to the distance from the perspective center to the sensor plane expressed in x or y image units (usually pixels). Thus they define an image scale often called the "camera constant." However, in general, $x_c\mathbf{a}$ is added to **h**, $y_c\mathbf{a}$ is added to **v**, and the fact that the image axes are not necessarily orthogonal causes further complications. This formulation was chosen by Yakimovsky and Cunningham because convenient expressions result for the projection. This projection includes not only the perspective projection itself, but also an arbitrary affine transformation in the image plane, caused by the fact the two image axes do not have to be orthogonal and the fact that the image scales in the two dimensions do not have to be equal. This

affine transformation allows for such things as nonsquare pixels and a CCD synchronization error in digitization that is a linear function of the position in the image.

More complete descriptions of these camera model parameters can be found elsewhere (Yakimovsky and Cunningham, 1978; Gennery, 2001). Also, Section 4.3 here describes how to extract more intuitively meaningful information from the parameters.

The resulting equations for the projection, as used here, are as follows:

$$\hat{x}_i = \frac{\mathbf{r}'_i \cdot \mathbf{h}}{\mathbf{r}'_i \cdot \mathbf{a}} \tag{12}$$

$$\hat{y}_i = \frac{\mathbf{r}'_i \cdot \mathbf{v}}{\mathbf{r}'_i \cdot \mathbf{a}} \tag{13}$$

since the apparent ray is along the vector $\mathbf{r}'_i = \mathbf{p}'_i - \mathbf{c}'_i$. The reason for using the circumflex over x and y is to represent computed values, to distinguish them from measured values that will be used later.

2.6. Summary of Camera Model

In addition to the manually set parameter L, the complete camera model consists of the five 3-vectors **c**, **a**, **h**, **v**, and **o**, where **a** and **o** are unit vectors; the distortion coefficients $\boldsymbol{\rho} = [\rho_0 \ \rho_1 \ \rho_2 \ \ldots]^T$; and the entrance pupil coefficients $\boldsymbol{\varepsilon} = [\varepsilon_0 \ \varepsilon_1 \ \varepsilon_2 \ \ldots]^T$. This is $15 + R + E$ adjusted parameters in all, of which two are redundant because of the unit vectors. Note that **c** and the epsilons are in terms of the units of distance used in object space, **h** and **v** are in terms of the units used in image space (usually pixels), and **a**, **o**, and the rhos are dimensionless. The exact meanings of these parameters are defined by equations (2), (5), (12), and (13), together with the relationships shown in Fig. 1.

Where convenient, the camera model parameters will be assembled into the column matrix ($15+R+E$-vector) **g**, as follows:

$$\mathbf{g} = \begin{bmatrix} \mathbf{c} \\ \mathbf{a} \\ \mathbf{h} \\ \mathbf{v} \\ \mathbf{o} \\ \boldsymbol{\rho} \\ \boldsymbol{\varepsilon} \end{bmatrix} \tag{14}$$

The $15+R+E$ by $15+R+E$ covariance matrix $\mathbf{C}_{gg}$ of the parameters also will be produced.

3. Basic Computations

3.1. General Case

Several important quantities were defined in Section 2, but we must now see exactly how these are used when it is desired to compute the image coordinates of a point from the three-dimensional position $\mathbf{p}_i$ of the point and a set of camera model parameters, either during a camera model adjustment or during an actual projection from object space to image space.

First, the following equations are used, which follow immediately from the definitions of the quantities (see Fig. 1):

$$\zeta_i = (\mathbf{p}_i - \mathbf{c})\cdot\mathbf{o} \tag{15}$$

$$\boldsymbol{\lambda}_i = \mathbf{p}_i - \mathbf{c} - \zeta_i\mathbf{o} \tag{16}$$

$$\lambda_i = |\boldsymbol{\lambda}_i| \tag{17}$$

since $\mathbf{o}$ is a unit vector.

The direction of the incoming ray depends upon the position of $\mathbf{c}_i'$, but $\mathbf{c}_i'$ is shifted by an amount that depends on the direction of the incoming ray. However, we have two equations for s_i, (2) and (3). Equating these two expressions for s_i produces an equation that can be solved for θ_i. By multiplying by $\sin\theta_i$ and rearranging terms, this equation becomes the following:

$$\zeta_i\sin\theta_i - \lambda_i\cos\theta_i - (\theta_i - \sin\theta_i)(\varepsilon_0 + \varepsilon_1\theta_i^2 + \varepsilon_2\theta_i^4 + \ldots) = 0 \tag{18}$$

The angle θ_i is computed as the smallest non-negative real root of (18). Since this is a transcendental equation, it must be solved iteratively. A good initial approximation for θ_i is $\arctan(\lambda_i/\zeta_i)$, since this is the exact value for a point at an infinite distance. (The double-argument arctangent must be used; if ζ_i is negative, the angle is greater than 90°.) Then Newton's method can be used to solve the equation, where the needed derivative of the left side of (18) with respect to θ_i is

$$\upsilon_i = \zeta_i\cos\theta_i + \lambda_i\sin\theta_i - (1 - \cos\theta_i)(\varepsilon_0 + \varepsilon_1\theta_i^2 + \varepsilon_2\theta_i^4 + \ldots) - (\theta_i - \sin\theta_i)(2\varepsilon_1\theta_i + 4\varepsilon_2\theta_i^3 + \ldots) \tag{19}$$

The convergence criterion currently used is 10^{-8}. Because Newton's method produces quadratic convergence, the resulting error is about 10^{-16}, which achieves double-precision accuracy. This is important because of the limiting values used at small angles below.

The value of θ_i resulting from the above should always be less than π. However, θ_i should

be tested to see that it is less than $\pi/(2|L|)$. If it fails this test, it means that the point is too far off axis to be compatible with the value of L that is being used.

In addition to θ_i, the value of υ_i is needed below. It is recomputed from (19) by using the value of θ_i from the last iteration.

Then χ_i is computed by (6). By definition,

$$\psi_i = \frac{\partial \chi_i}{\partial \theta_i} \tag{20}$$

which, by (6), produces

$$\begin{aligned} \psi_i &= \cos L\theta_i \quad \text{if } L < 0 \\ \psi_i &= 1 \quad \text{if } L = 0 \\ \psi_i &= \frac{1}{\cos^2 L\theta_i} \quad \text{if } L > 0 \end{aligned} \tag{21}$$

The quantity ζ_i' is computed as follows, from (10):

$$\zeta_i' = \frac{\lambda_i}{\chi_i} \tag{22}$$

and μ_i is computed by (8). We also need the derivative of μ_i, as follows:

$$\nu_i = \frac{\partial \mu_i}{\partial \chi_i} = 2\rho_1\chi_i + 4\rho_2\chi_i^3 + \ldots \tag{23}$$

The vector $\mathbf{r}_i' = \mathbf{p}_i' - \mathbf{c}_i'$ from the projected shifted entrance pupil to the apparent object point is conveniently computed as follows:

$$\mathbf{r}_i' = \zeta_i'\mathbf{o} + (1 + \mu_i)\boldsymbol{\lambda}_i \tag{24}$$

We also need the partial derivatives of $\mathbf{r}_i'$ with respect to $\mathbf{o}$, $\mathbf{c}$, and $\mathbf{p}_i$. These can be obtained from several intermediate partial derivatives. Let $\mathbf{u}$ represent either $\mathbf{o}$, $\mathbf{c}$, or $\mathbf{p}_i$. Differentiating (18) produces

$$\upsilon_i \frac{\partial \theta_i}{\partial \mathbf{u}} + \frac{\partial \zeta_i}{\partial \mathbf{u}} \sin\theta_i - \frac{\partial \lambda_i}{\partial \mathbf{u}} \cos\theta_i = 0 \tag{25}$$

since υ_i is the derivative of the left side of (18) with respect to θ_i. Solving (25) produces

$$\frac{\partial \theta_i}{\partial \mathbf{u}} = \frac{\frac{\partial \lambda_i}{\partial \mathbf{u}} \cos\theta_i - \frac{\partial \zeta_i}{\partial \mathbf{u}} \sin\theta_i}{\upsilon_i} \tag{26}$$

Differentiating (22) and using (20) produces

$$\frac{\partial \zeta_i'}{\partial \mathbf{u}} = \frac{1}{\chi_i}\frac{\partial \lambda_i}{\partial \mathbf{u}} - \frac{\lambda_i}{\chi_i^2}\psi_i \frac{\partial \theta_i}{\partial \mathbf{u}}$$

$$= \left(\frac{1}{\chi_i} - \frac{\psi_i \lambda_i \cos\theta_i}{\chi_i^2 \upsilon_i}\right)\frac{\partial \lambda_i}{\partial \mathbf{u}} + \frac{\psi_i \lambda_i \sin\theta_i}{\chi_i^2 \upsilon_i}\frac{\partial \zeta_i}{\partial \mathbf{u}} \tag{27}$$

where the second form has been obtained by substituting (26). Differentiating (15) produces

$$\frac{\partial \zeta_i}{\partial \mathbf{p}_i} = \mathbf{o}^{\mathrm{T}} \tag{28}$$

$$\frac{\partial \zeta_i}{\partial \mathbf{o}} = (\mathbf{p}_i - \mathbf{c})^{\mathrm{T}} \tag{29}$$

Differentiating (16) produces

$$\frac{\partial \boldsymbol{\lambda}_i}{\partial \mathbf{p}_i} = \mathbf{I} - \mathbf{o}\mathbf{o}^{\mathrm{T}} \tag{30}$$

$$\frac{\partial \boldsymbol{\lambda}_i}{\partial \mathbf{o}} = -\zeta \mathbf{I} - \mathbf{o}(\mathbf{p}_i - \mathbf{c})^{\mathrm{T}} \tag{31}$$

Then, using (30), (31), the identity that

$$\frac{d\lambda_i}{d\boldsymbol{\lambda}_i} = \mathrm{unit}(\boldsymbol{\lambda}_i)^{\mathrm{T}} \tag{32}$$

and the fact that $\mathbf{o}$ and $\boldsymbol{\lambda}_i$ are orthogonal, produces the following:

$$\frac{\partial \lambda_i}{\partial \mathbf{p}_i} = \mathrm{unit}(\boldsymbol{\lambda}_i)^{\mathrm{T}} \tag{33}$$

$$\frac{\partial \lambda_i}{\partial \mathbf{o}} = -\zeta_i \mathrm{unit}(\boldsymbol{\lambda}_i)^{\mathrm{T}} \tag{34}$$

which also can be obtained by inspection of Fig. 1. Finally,

$$\frac{\partial \chi_i}{\partial \mathbf{u}} = \frac{\partial \chi_i}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mathbf{u}}$$

$$= \psi \frac{\dfrac{\partial \lambda_i}{\partial \mathbf{u}}\cos\theta_i - \dfrac{\partial \zeta_i}{\partial \mathbf{u}}\sin\theta_i}{\upsilon_i} \tag{35}$$

which follows by substitution from (20) and (26).

Differentiating (24) produces

$$\frac{\partial \mathbf{r}_i'}{\partial \mathbf{u}} = \zeta_i' \frac{\partial \mathbf{o}}{\partial \mathbf{u}} + \mathbf{o}\frac{\partial \zeta_i'}{\partial \mathbf{u}} + (1+\mu_i)\frac{\partial \boldsymbol{\lambda}_i}{\partial \mathbf{u}} + \nu_i \boldsymbol{\lambda}_i \frac{\partial \chi_i}{\partial \mathbf{u}} \tag{36}$$

Now the specific derivatives with respect to $\mathbf{o}$ and $\mathbf{p}_i$ derived above can be substituted for those with respect to $\mathbf{u}$ in (36). Then, using the facts that $\partial\mathbf{o}/\partial\mathbf{o} = \mathbf{I}$ and $\partial\mathbf{o}/\partial\mathbf{p}_i' = \mathbf{O}$ and rearranging produces the following:

$$\begin{aligned}\frac{\partial\mathbf{r}_i'}{\partial\mathbf{o}} &= (\zeta_i' - (1+\mu_i)\zeta_i)\mathbf{I} \\ &+ \left(\frac{\zeta_i'\psi_i\sin\theta_i}{\chi_i\upsilon_i} - (1+\mu_i)\right)\mathbf{o}\,(\mathbf{p}_i - \mathbf{c})^{\mathrm{T}} - \frac{\nu_i\psi_i\sin\theta_i}{\upsilon_i}\boldsymbol{\lambda}_i(\mathbf{p}_i - \mathbf{c})^{\mathrm{T}} \\ &- \zeta_i\left(\frac{1}{\chi_i\lambda_i} - \frac{\psi_i\cos\theta_i}{\chi_i^2\upsilon_i}\right)\mathbf{o}\boldsymbol{\lambda}_i^{\mathrm{T}} - \zeta_i\frac{\nu_i\psi_i\cos\theta_i}{\lambda_i\upsilon_i}\boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^{\mathrm{T}}\end{aligned} \tag{37}$$

$$\begin{aligned}\frac{\partial\mathbf{r}_i'}{\partial\mathbf{p}_i} &= (1+\mu_i)\mathbf{I} \\ &+ \left(\frac{\zeta_i'\psi_i\sin\theta_i}{\chi_i\upsilon_i} - (1+\mu_i)\right)\mathbf{o}\mathbf{o}^{\mathrm{T}} - \frac{\nu_i\psi_i\sin\theta_i}{\upsilon_i}\boldsymbol{\lambda}_i\mathbf{o}^{\mathrm{T}} \\ &+ \left(\frac{1}{\chi_i\lambda_i} - \frac{\psi_i\cos\theta_i}{\chi_i^2\upsilon_i}\right)\mathbf{o}\boldsymbol{\lambda}_i^{\mathrm{T}} + \frac{\nu_i\psi_i\cos\theta_i}{\lambda_i\upsilon_i}\boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^{\mathrm{T}}\end{aligned} \tag{38}$$

Since $\mathbf{p}_i$ and $\mathbf{c}$ always appear in the combination $\mathbf{p}_i - \mathbf{c}$, the following is true:

$$\frac{\partial\mathbf{r}_i'}{\partial\mathbf{c}} = -\frac{\partial\mathbf{r}_i'}{\partial\mathbf{p}_i} \tag{39}$$

Note that the differences between (37) and (38) consist of the different coefficients for $\mathbf{I}$, the presence of $(\mathbf{p}_i - \mathbf{c})^{\mathrm{T}}$ instead of $\mathbf{o}^{\mathrm{T}}$ in the former, and the presence of $-\zeta_i$ in the last two terms of the former.

We also need the partial derivatives with respect to the distortion parameters and the entrance pupil parameters. Differentiating (24) with respect to $\boldsymbol{\rho}$ produces the following:

$$\frac{\partial\mathbf{r}_i'}{\partial\boldsymbol{\rho}} = \boldsymbol{\lambda}_i\begin{bmatrix}1 & \chi_i^2 & \chi_i^4 & \cdots\end{bmatrix} \tag{40}$$

Differentiating (18), (22), and (24) with respect to $\boldsymbol{\varepsilon}$ and combining the results produces the following:

$$\frac{\partial\mathbf{r}_i'}{\partial\boldsymbol{\varepsilon}} = \frac{\psi_i(\theta_i - \sin\theta_i)}{\upsilon_i}\left(\nu_i\boldsymbol{\lambda}_i - \frac{\zeta_i'}{\chi_i}\mathbf{o}\right)\begin{bmatrix}1 & \theta_i^2 & \theta_i^4 & \cdots\end{bmatrix}\} \tag{41}$$

The derivatives of $\mathbf{r}_i'$ with respect to $\mathbf{p}_i$ will be needed for both the camera model least-squares adjustment and the 3D-to-2D projection from object space to image space. The other derivatives of $\mathbf{r}_i'$ are not needed for the 3D-to-2D projection.

Since the vector $\mathbf{r}_i'$ obtained above includes the corrections for distortion and entrance pupil movement, it can be used in the Yakimovsky and Cunningham equations, as in (12) and (13). However, for efficiency these equations are expressed in terms of the intermediate quantities α_i, β_i, and γ_i, which will be needed again later, as follows:

$$\alpha_i = \mathbf{r}_i' \cdot \mathbf{a} \tag{42}$$

$$\beta_i = \mathbf{r}_i' \cdot \mathbf{h} \tag{43}$$

$$\gamma_i = \mathbf{r}_i' \cdot \mathbf{v} \tag{44}$$

$$\hat{x}_i = \frac{\beta_i}{\alpha_i} \tag{45}$$

$$\hat{y}_i = \frac{\gamma_i}{\alpha_i} \tag{46}$$

The above equations can be differentiated to produce the following:

$$\frac{\partial \hat{x}_i}{\partial \mathbf{r}_i'} = \frac{\mathbf{h}^T}{\alpha_i} - \frac{\beta_i \mathbf{a}^T}{\alpha_i^2} = \frac{\mathbf{h}^T - \hat{x}_i \mathbf{a}^T}{\alpha_i} \tag{47}$$

$$\frac{\partial \hat{y}_i}{\partial \mathbf{r}_i'} = \frac{\mathbf{v}^T}{\alpha_i} - \frac{\gamma_i \mathbf{a}^T}{\alpha_i^2} = \frac{\mathbf{v}^T - \hat{y}_i \mathbf{a}^T}{\alpha_i} \tag{48}$$

$$\frac{\partial \hat{x}_i}{\partial \mathbf{a}} = -\frac{\beta_i \mathbf{r}_i'^T}{\alpha_i^2} = -\frac{\hat{x}_i \mathbf{r}_i'^T}{\alpha_i} \tag{49}$$

$$\frac{\partial \hat{y}_i}{\partial \mathbf{a}} = -\frac{\gamma_i \mathbf{r}_i'^T}{\alpha_i^2} = -\frac{\hat{y}_i \mathbf{r}_i'^T}{\alpha_i} \tag{50}$$

3.2. Case of Zero Angle

It can be seen that some of the above equations become indeterminate when $\theta_i = 0$. Appropriate forms to use in this case can be obtained by taking the limit as θ_i approaches zero. This produces the following:

$$\mathbf{r}_i' = \mathbf{p}_i - \mathbf{c} \tag{51}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \boldsymbol{\varepsilon}} = \mathbf{O} \tag{52}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \mathbf{o}} = -\rho_0 \zeta_i \mathbf{I} - \rho_0 \mathbf{o}(\mathbf{p}_i - \mathbf{c})^{\mathrm{T}} \tag{53}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \mathbf{p}_i} = (1+\rho_0)\mathbf{I} - \rho_0 \mathbf{o}\mathbf{o}^{\mathrm{T}} \tag{54}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \mathbf{c}} = -\frac{\partial \mathbf{r}_i'}{\partial \mathbf{p}_i} \tag{55}$$

$$\chi_i = 0 \tag{56}$$

In the implemented program, (51)-(56) are used instead of the corresponding equations in Section 3.1 when $\theta_i < 10^{-8}$. (The rest of the equations are unchanged.) Since double precision is used in the computations, this insures that at least single-precision accuracy is maintained even if θ_i is slightly less than or slightly greater than 10^{-8} radians, without the necessity of using series expansions. (Single-precision accuracy is sufficient for the input and output quantities. Within the least-squares adjustment in Section 5, double precision is used because of the loss of precision that can occur in that process.)

3.3. Case of Perspective Lens Model without Pupil Movement

If $L = 1$ and $\boldsymbol{\varepsilon} = \mathbf{0}$, the camera model reduces to the case previously reported (Gennery, 2001). Equations (8), (15), (16), (23), and (40) are unchanged. Equation (6) reduces to $\chi_i = \tan\theta_i$. The other equations in Sections 3.1 and 3.2 also still work, but it is more efficient to use simplified forms for the partial derivatives of $\mathbf{r}_i'$ that result from (37) and (38) by making the appropriate substitutions. These are as follows:

$$\frac{\partial \mathbf{r}_i'}{\partial \mathbf{o}} = -\mu_i \zeta_i \mathbf{I} - \mu_i \, \mathbf{o} \, (\mathbf{p}_i - \mathbf{c})^{\mathrm{T}} - \frac{\nu_i \chi_i}{\zeta_i} \boldsymbol{\lambda}_i (\mathbf{p}_i - \mathbf{c})^{\mathrm{T}} - \frac{\nu_i}{\zeta_i \chi_i} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^{\mathrm{T}} \tag{57}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \mathbf{p}_i} = (1+\mu_i)\mathbf{I} - \mu_i \, \mathbf{o}\mathbf{o}^{\mathrm{T}} - \frac{\nu_i \chi_i}{\zeta_i} \boldsymbol{\lambda}_i \mathbf{o}^{\mathrm{T}} + \frac{\nu_i}{\zeta_i^2 \chi_i} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^{\mathrm{T}} \tag{58}$$

Note that, from (23), ν_i/χ_i can be computed without any indeterminacy.

Equations (57) and (58) are mathematically equivalent to the corresponding equations previously published (Gennery, 2001), except that $\mathbf{p}_i'$ was used there instead of $\mathbf{r}_i'$, since there was no entrance pupil movement, and τ_i there represents χ_i^2 here. However, not all of the possible algebraic simplification was done there, so that (57) and (58) appear simpler here.

4. Intrinsic Camera Model

4.1. Motivation

Often it is desired to have explicit values for the intrinsic camera model, which depend only upon the internal state of the camera but not its pose (position and orientation) relative to the external world. One reason for this desire is so that previous values of the intrinsic model can be incorporated easily into a camera model adjustment when the camera is moved. (The new adjustment to determine the extrinsic parameters then would need less complete calibration data. For example, the calibration points then could all be in one plane.) Another reason is to have parameters that have more intuitive meaning than do the Yakimovsky and Cunningham parameters, in order to better understand what a camera model adjustment has produced. Two approaches are described here, one for each of these two goals.

In either case, $\boldsymbol{\rho}$ for distortion and $\boldsymbol{\varepsilon}$ for entrance pupil movement are invariant with respect to pose, and thus are suitable as part of the intrinsic model. On the other hand, $\mathbf{c}$ denotes the camera position, and thus it obviously is part of the extrinsic model. The question then is how to extract the desired information from the $\mathbf{a}$, $\mathbf{h}$, $\mathbf{v}$, and $\mathbf{o}$ vectors.

4.2. Dot-Product Intrinsic Model

The method used here for incorporating previous intrinsic camera models into new adjustments utilizes the fact that dot products among the $\mathbf{a}$, $\mathbf{h}$, $\mathbf{v}$, and $\mathbf{o}$ vectors are invariant with respect to rotation and translation. The following dot products are used: $\mathbf{h}\cdot\mathbf{h}$, $\mathbf{v}\cdot\mathbf{v}$, $\mathbf{h}\cdot\mathbf{v}$, $\mathbf{a}\cdot\mathbf{h}$, $\mathbf{a}\cdot\mathbf{v}$, $\mathbf{o}\cdot\mathbf{h}$, and $\mathbf{o}\cdot\mathbf{v}$. The other possible combinations are not used for the following reasons: $\mathbf{a}\cdot\mathbf{a}$ and $\mathbf{o}\cdot\mathbf{o}$ are superfluous because $\mathbf{a}$ and $\mathbf{o}$ are unit vectors, $\mathbf{a}\cdot\mathbf{o}$ would be numerically very poor because $\mathbf{a}$ and $\mathbf{o}$ usually are nearly collinear, and the seven dot products chosen above suffice to constrain the four vectors into a rigid group. (The four vectors have only ten degrees of freedom, since two of them are unit vectors. Three of these degrees of freedom are for orientation of the entire camera. This leaves seven degrees of freedom, which are accounted for by the seven dot products used here.)

Therefore, this form of the intrinsic camera model consists of the following $7+R+E$-vector:

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{h}\cdot\mathbf{h} \\ \mathbf{v}\cdot\mathbf{v} \\ \mathbf{h}\cdot\mathbf{v} \\ \mathbf{a}\cdot\mathbf{h} \\ \mathbf{a}\cdot\mathbf{v} \\ \mathbf{o}\cdot\mathbf{h} \\ \mathbf{o}\cdot\mathbf{v} \\ \boldsymbol{\rho} \\ \boldsymbol{\varepsilon} \end{bmatrix} \tag{59}$$

The $7+R+E$ by $15+R+E$ matrix of partial derivatives of the intrinsic camera model with respect to the complete camera model is the following:

$$\mathbf{J} = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{g}} = \begin{bmatrix} \mathbf{0}^T & \mathbf{0}^T & 2\mathbf{h}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & 2\mathbf{v}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{v}^T & \mathbf{h}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{h}^T & \mathbf{a}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{v}^T & \mathbf{0}^T & \mathbf{a}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{o}^T & \mathbf{0}^T & \mathbf{h}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{o}^T & \mathbf{v}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I} \end{bmatrix} \tag{60}$$

where the identity matrices, zero matrices, and zero vectors in the last two rows and last two columns are there because $\boldsymbol{\rho}$ and $\boldsymbol{\varepsilon}$ are present in both the complete model and the intrinsic model, and where the zero vectors in the first column are there because the $\mathbf{c}$ vector does not influence the intrinsic model. (The vectors in $\mathbf{J}$ are shown transposed to denote that they are used here as rows instead of as columns.)

We use the notation that a tilde ($\tilde{}$) over a symbol denotes a value from a previous calibration whose results are desired to be included in a new calibration (with different camera pose).

The $7+R+E$ by $7+R+E$ weight matrix $\tilde{\mathbf{W}}$ of the old intrinsic camera model will be needed. If it is desired to constrain the intrinsic camera model to be exactly that from the old calibration, $\tilde{\mathbf{W}}$ could be set to a diagonal matrix with very large values on the main diagonal. However, it

makes more sense to compute $\tilde{\mathbf{W}}$ from the covariance matrix $\tilde{\mathbf{C}}_{gg}$ of the old calibration, since the new information can then combine with the old information optimally and may possibly improve the internal camera model. Therefore,

$$\tilde{\mathbf{W}} = (\tilde{\mathbf{J}}\tilde{\mathbf{C}}_{gg}\tilde{\mathbf{J}}^T)^{-1} \tag{61}$$

(Note that the tilde over **J** means that the camera model parameters from the old calibration are used in its computation, as described in above.)

4.3. Intuitive Intrinsic Model

As previously reported (Gennery et al., 1987), the scale factors in each image dimension, the "center" of the image, and the angle between the image axes can be computed as follows:

$$h_s = |\mathbf{a}\times\mathbf{h}| \tag{62}$$

$$v_s = |\mathbf{a}\times\mathbf{v}| \tag{63}$$

$$x_c = \mathbf{a}\cdot\mathbf{h} \tag{64}$$

$$y_c = \mathbf{a}\cdot\mathbf{v} \tag{65}$$

$$\phi = \arctan\frac{\mathbf{v}\times\mathbf{h}\cdot\mathbf{a}}{(\mathbf{a}\times\mathbf{v})\cdot(\mathbf{a}\times\mathbf{h})} \tag{66}$$

The scale factors h_s and v_s denote the ratio of a change in image coordinate to the change in off-axis object angle when on axis. They usually are approximately equal to the focal length, expressed in image units (usually pixels). The quantities x_c and y_c are as described in Section 2.5. Propagation of the covariance matrix from the complete camera model can also be done (Gennery et al., 1987).

To complete this form of the intrinsic model, the direction of the **o** vector relative to the **a** vector also would be needed, but this has not been implemented.

5. Adjustment of Camera Model

5.1. Data for Adjustment

The calibration data consists of a set of points, with for each point i its three-dimensional position $\mathbf{p}_i$ in object coordinates and its measured two-dimensional position x_i and y_i in image

coordinates.

Other given information consists of the following: the parameter L that defines the basic type of camera model, the minimum allowed estimate of the standard deviation of measured image-coordinate positions σ_{min}, the nominal focal length of the camera f, the nominal pixel spacings p_{hor} and p_{vert} (in the same units as f), the number of columns s_{hor} and rows s_{vert} of pixels in the camera, and the approximate position of the camera $\mathbf{c}_o$ in object coordinates. (The quantities f, p_{hor}, p_{vert}, s_{hor}, s_{vert}, and $\mathbf{c}_o$ are used primarily in obtaining an initial approximation for iterating, and thus their exact values are not important.) In addition, either a previous intrinsic model, represented by $\tilde{\boldsymbol{\xi}}$ from (59) and $\tilde{\mathbf{W}}$ from (61), or the following information is given: the a priori standard deviation σ_d (in radians) of the difference between the optical axis $\mathbf{o}$ and the $\mathbf{a}$ vector; the a priori standard deviations about zero of the distortion parameters, represented here by the covariance matrix $\mathbf{C}_{\rho\rho}$; and the a priori standard deviations about zero of the pupil movement parameters, represented here by the covariance matrix $\mathbf{C}_{\varepsilon\varepsilon}$.

The desired result consists of the camera model parameters $\mathbf{g}$ and their covariance matrix $\mathbf{C}_{gg}$ (which indicates the accuracy of $\mathbf{g}$), as defined in Section 2.6.

5.2. Initialization

In order to obtain initial values for iterating, first a point close to the camera axis is found by selecting $\mathbf{p}_{axis}$ to be the $\mathbf{p}_i$ for which x_i and y_i are closest to $s_{hor}/2$ and $s_{vert}/2$, respectively. Then this and the given data are used to compute the initial approximations to the camera model, as follows:

$$\mathbf{c}_o \text{ is given} \tag{67}$$

$$\mathbf{a}_o = \text{unit}(\mathbf{p}_{axis} - \mathbf{c}_o) \tag{68}$$

$$\mathbf{h}_o = \frac{f}{p_{hor}} \text{unit}(\mathbf{a}_o \times \mathbf{u}) + \frac{s_{hor}}{2} \mathbf{a}_o \tag{69}$$

$$\mathbf{v}_o = \frac{f}{p_{vert}} \text{unit}(\mathbf{a}_o \times \mathbf{h}_o) + \frac{s_{vert}}{2} \mathbf{a}_o \tag{70}$$

$$\mathbf{o}_o = \mathbf{a}_o \tag{71}$$

$$\boldsymbol{\rho}_o = \mathbf{0} \tag{72}$$

$$\boldsymbol{\varepsilon}_o = \mathbf{0} \tag{73}$$

where $\mathbf{u}$ is a vector pointing upwards in object space, and where it is assumed that the image x

axis points to the right and the image y axis points down (from the upper left corner of the image). If the y axis points up in the image, the sign of $\mathbf{v}_o$ should be reversed. (In the existing program, $\mathbf{a}_o$ can be supplied by the user instead of using (68).) The above vectors are assembled into $\mathbf{g}_o$, analogously to (14). Also, as an initial approximation of measurement standard deviation, $\sigma_o = 1$ pixel.

Alternatively, if $\check{\boldsymbol{\xi}}$ is present (from a previous solution), initial approximations could be obtained from it, especially for $\boldsymbol{\rho}$ and $\boldsymbol{\varepsilon}$, and possibly (when the physical orientation is approximately known) for $\mathbf{a}$, $\mathbf{h}$, $\mathbf{v}$, and $\mathbf{o}$ by rotating $\tilde{\mathbf{a}}$, $\tilde{\mathbf{h}}$, $\tilde{\mathbf{v}}$, and $\tilde{\mathbf{o}}$ by premultiplying by the approximately known rotation matrix. (At present, this is not done in the existing program.) However, the position $\mathbf{c}_o$ must still be supplied.

The a priori weight matrix can be computed as follows:

$$\mathbf{N}_o = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sigma_d^2}\mathbf{I} & \mathbf{0} & \mathbf{0} & -\frac{1}{\sigma_d^2}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\frac{1}{\sigma_d^2}\mathbf{I} & \mathbf{0} & \mathbf{0} & \frac{1}{\sigma_d^2}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{\rho\rho}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{\varepsilon\varepsilon}^{-1} \end{bmatrix} \tag{74}$$

where the fact that the off-diagonal terms for $\mathbf{a}$ and $\mathbf{o}$ are the negative of the main-diagonal terms causes the standard deviation σ_d to apply to the difference of $\mathbf{a}$ and $\mathbf{o}$. Additional weight can be applied to any other of the initial approximation values, by adding the reciprocal of the variance to the appropriate main diagonal element of $\mathbf{N}_o$. For example, the position of the camera may be known sufficiently well to enter this information into the solution by adding an appropriate 3-by-3 weight matrix in the upper left corner of $\mathbf{N}_o$. (If a previous intrinsic model represented by $\check{\boldsymbol{\xi}}$ is used, the rest of $\mathbf{N}_o$ usually would be set to zero.)

5.3. Iterative Solution

The method does a rigorous least-squares adjustment in which the camera model parameters are adjusted to minimize the sum of the squares of the residuals (differences between measured and adjusted positions of points) in the image plane. Since the problem is nonlinear, this requires

an iterative solution. The standard Gauss method is used, in which the problem is linearized by means of partial derivatives, the linear least-squares problem is solved, and the process repeats on each iteration (Bard, 1974; Mikhail, 1976). Constraints (Mikhail, 1976) are included in order to force **a** and **o** to be unit vectors. For the problem here, the method converges fairly rapidly if started close to the correct solution, and a reasonably good approximation to start the iterations was obtained in Section 5.2.

Removing wild points that disagree with the rest of the solution is done by a previously developed general method (Gennery, 1980). This editing method is optimum if there is only one erroneous point to remove, and it is nearly optimum if there are only a few erroneous points out of many. (Since finding dots or corners on a calibration fixture is a reliable process, usually there are few, if any, points to remove.) There is an inner loop for iterating the nonlinear solution and an outer loop for editing, as shown in Fig. 7.

The inner loop is initialized using the values in Section 5.2. Then each iteration of the nonlinear least-squares adjustment proceeds as follows. The 2-by-15+R+E matrix of partial derivatives of the constraints (unit(**a**) = 1 and unit(**o**) = 1) relative to **g** (the parameters) is

$$\mathbf{K} = \begin{bmatrix} \mathbf{0}^T & \text{unit}(\mathbf{a})^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \text{unit}(\mathbf{o})^T & \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \tag{75}$$

For each point i currently retained, $\hat{x}_i$, $\hat{y}_i$, the preliminary quantities, and the partial derivatives of $\mathbf{r}'_i$, $\hat{x}_i$, and $\hat{y}_i$ are computed as in Section 3. Then the 2 by 15+R+E matrix of partial derivatives of $\hat{x}_i$ and $\hat{y}_i$ relative to **g** is

$$\mathbf{A}_i = \begin{bmatrix} \frac{\partial \hat{x}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \mathbf{c}} & \frac{\partial \hat{x}_i}{\partial \mathbf{a}} & \frac{\mathbf{r}'^T_i}{\alpha_i} & \mathbf{0}^T & \frac{\partial \hat{x}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \mathbf{o}} & \frac{\partial \hat{x}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \boldsymbol{\rho}} & \frac{\partial \hat{x}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \boldsymbol{\varepsilon}} \\ \frac{\partial \hat{y}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \mathbf{c}} & \frac{\partial \hat{y}_i}{\partial \mathbf{a}} & \mathbf{0}^T & \frac{\mathbf{r}'^T_i}{\alpha_i} & \frac{\partial \hat{y}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \mathbf{o}} & \frac{\partial \hat{y}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \boldsymbol{\rho}} & \frac{\partial \hat{y}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \boldsymbol{\varepsilon}} \end{bmatrix} \tag{76}$$

and the discrepancies between measured and computed data are

$$\mathbf{e}_i = \begin{bmatrix} x_i - \hat{x}_i \\ y_i - \hat{y}_i \end{bmatrix} \tag{77}$$

The solution then is obtained as follows:

$$\mathbf{N} = \mathbf{N}_o + \mathbf{J}^T\tilde{\mathbf{W}}\mathbf{J} + \frac{nf^2}{\sigma^2 p_{hor} p_{vert}}\mathbf{K}^T\mathbf{K} + \frac{1}{\sigma^2}\sum_i \mathbf{A}_i^T\mathbf{A}_i \tag{78}$$

$$\mathbf{t} = \mathbf{N}_o(\mathbf{g}_o - \mathbf{g}) + \mathbf{J}^T\tilde{\mathbf{W}}(\tilde{\boldsymbol{\xi}} - \boldsymbol{\xi}) + \frac{nf^2}{\sigma^2 p_{hor} p_{vert}}\mathbf{K}^T\begin{bmatrix} 1 - |\mathbf{a}| \\ 1 - |\mathbf{o}| \end{bmatrix} + \frac{1}{\sigma^2}\sum_i \mathbf{A}_i^T\mathbf{e}_i \tag{79}$$

$$\mathbf{M} = \mathbf{K}\mathbf{N}^{-1}\mathbf{K}^T \tag{80}$$

$$\mathbf{d} = \mathbf{N}^{-1}\mathbf{t} + \mathbf{N}^{-1}\mathbf{K}^T\mathbf{M}^{-1}\left(\begin{bmatrix} 1 - |\mathbf{a}| \\ 1 - |\mathbf{o}| \end{bmatrix} - \mathbf{K}\mathbf{N}^{-1}\mathbf{t}\right) \tag{81}$$

$$\mathbf{C}_{gg} = \mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{K}^T\mathbf{M}^{-1}\mathbf{K}\mathbf{N}^{-1} \tag{82}$$

$$q = \sum_i \mathbf{e}_i^T\mathbf{e}_i - \mathbf{d}^T(2\mathbf{t} - \mathbf{N}\mathbf{d}) \tag{83}$$

where the summations are over all points currently retained, n is the total number of these points, $\mathbf{g}$ represents the current parameter values, and $\mathbf{g}_o$ represents the a priori parameter values (initial approximations). These equations represent a standard constrained least-squares adjustment (Mikhail, 1976), except for additional terms before the last in (78) and (79). The first term in (78) and (79) applies the a priori weight to the initial values. The second term applies the information from a previous intrinsic camera model solution, as described in Section 4.2. It would not be present (equivalently, $\tilde{\mathbf{W}} = 0$) if there is no such information to be included. (The fact that a tilde is not used on $\mathbf{J}$ above indicates that it is computed from the camera model parameters of the current iteration.) The third term applies some weight to the constraints. These constraint terms in (78) and (79) mathematically have no effect on the solution, since the exact constraints are applied in (81) and (82). However, they are necessary to prevent the solution without the constraints from being singular, and that is computed first (as $\mathbf{N}^{-1}\mathbf{t}$ in (81)) before the constraints are applied. The scale factor chosen for these terms above cause them to have about the same magnitude as the result of the main summation for $\mathbf{N}$, so that numerical accuracy is preserved. The last term in (78) and (79) is the main contribution from the linearized least-squares solution.

Note that $\mathbf{a}$ and $\mathbf{o}$ are unit vectors on the first iteration because of the initial approximations, and they are unit vectors on the last iteration within the convergence tolerance, but on intermediate iterations they in general are not. The comparison of their magnitudes with unity in (81) is what causes them to converge to unit vectors. Also note that $\mathbf{C}_{gg}$ is needed only on the

last iteration.

Also computed on each iteration is an improved estimate of measurement variance σ^2 (for use on the next iteration), obtained by dividing q from (83) by the number of degrees of freedom in the adjustment (the number of measurements minus the number of unconstrained parameters), but limited to be no less than the given minimum variance, as follows:

$$\sigma^2 = \max\left(\frac{q}{2n - 13 - R - E + \kappa}, \sigma^2_{\min} \right) \tag{84}$$

where n is the number of points currently retained (not rejected) (that is, the number of points actually used to obtain q), and where κ is the number of camera model parameters that have been effectively constrained by having either large a priori weights or large weights ($\tilde{\mathbf{W}}$) from a previous intrinsic model. (13 occurs in (84) instead of 15, because of the two exact constraints that force **a** and **o** to be unit vectors.) We usually just use $\kappa = 3$, based on the fact that there usually are fairly large a priori weights forcing **o** to be nearly equal to **a** and forcing ρ_0 to be nearly zero, but not much a priori weight on the other $\boldsymbol{\rho}$ and $\boldsymbol{\varepsilon}$ parameters. The exact value of κ is not very important, since the number of measurements $2n$ usually is much larger than $13+R+E$, and usually not much accuracy is needed or is attainable with variances, anyway.

The term $\mathbf{d}^T(2\mathbf{t} - \mathbf{N}\mathbf{d})$ in (83) would reduce to $\mathbf{d}^T\mathbf{t}$ if there were no constraints. Subtracting this term from the sum of squares of the discrepancies produces the sum of squares of the residuals. On the last iteration, there is no difference between the two (**d** converges to zero), so this term could be left out (as it was in previous versions of the program) without affecting the final results, but including it can help convergence on early iterations because of keeping the correct relative weight between the measurements and the a priori information in (78) and (79).

The corrections **d** from (81) are added to the current estimate of the parameters **g**, to obtain **g** for the next iteration. When each component of **d** is sufficiently small, convergence is declared.

The convergence criterion currently used is 10^{-8}. However, this criterion is multiplied by an appropriate value for each portion of **d** in order to scale it to a meaningful size for that parameter. These scale factors are as follows: $|\mathbf{p}_{axis} - \mathbf{c}_o|$ for **c**, unity for **a**, f/p_{hor} for **h**, f/p_{vert} for **v**, unity for **o**, unity for each element of $\boldsymbol{\rho}$, and a given characteristic lens dimension, such as the lens diameter or f, for each element of $\boldsymbol{\varepsilon}$. The convergence criterion of 10^{-8} guarantees single-precision accuracy in the results, even though the solution undergoes only linear convergence. The use of double-precision computations assures that this tolerance can be achieved, even if the solution is rather ill-conditioned.

In order to determine whether a point should be rejected in the automatic editing, the two

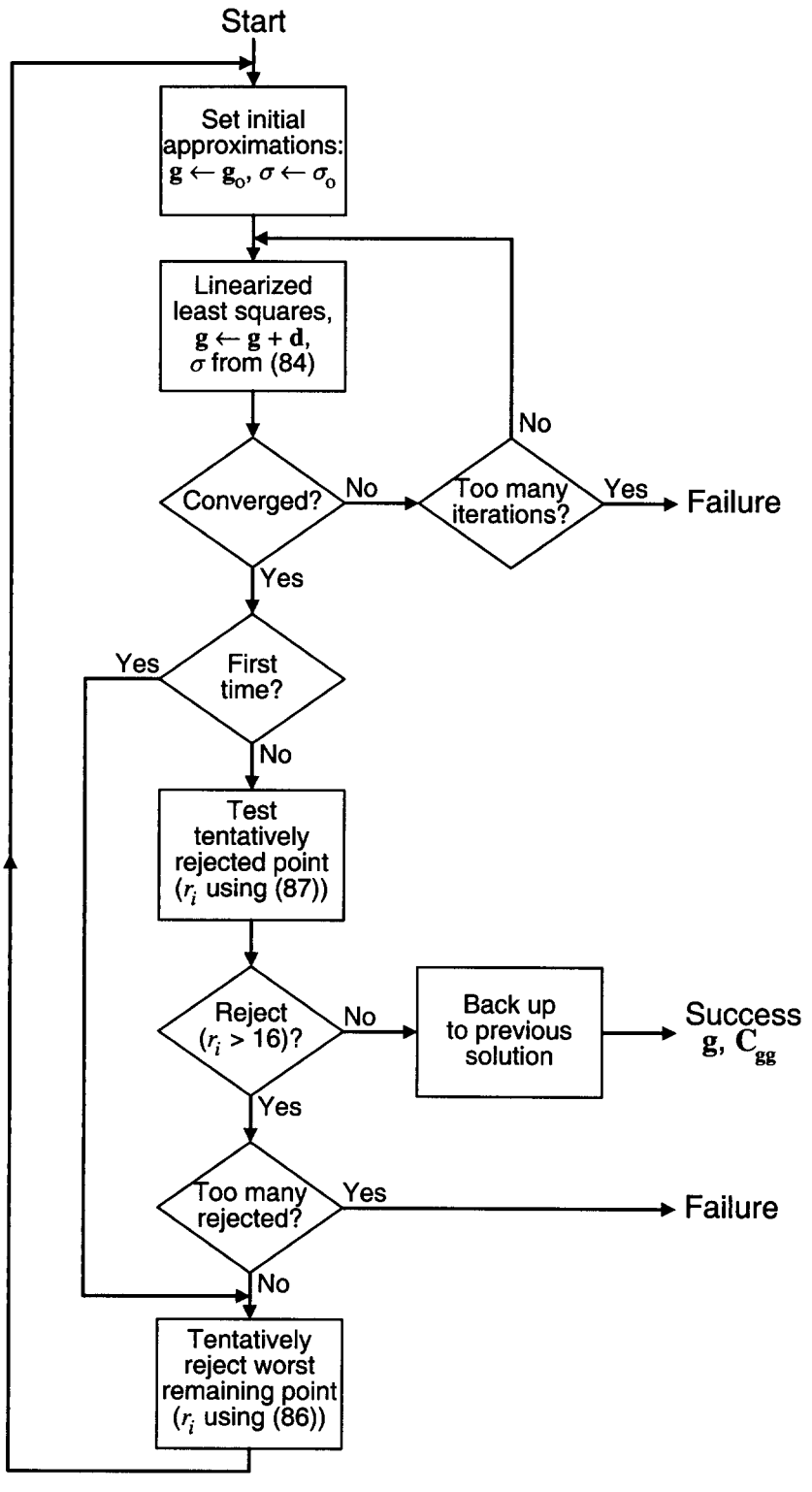

Figure 7. Flowchart of the principal steps in the camera model adjustment.

residuals for a point are compared to their covariance matrix by computing the corresponding quadratic form, as follows (using the results from the last iteration above, so that $\mathbf{e}_i$ has converged to the residuals for point i):

$$r_i = \mathbf{e}_i^T \mathbf{C}_{ee,i}^{-1} \mathbf{e}_i \tag{85}$$

(which is a generalization to multiple dimensions of the ratio of the square of a deviation to its variance.) The 2-by-2 covariance matrix $\mathbf{C}_{ee,i}$ is either

$$\mathbf{C}_{ee,i} = \sigma^2 \mathbf{I} - \mathbf{A}_i \mathbf{C}_{gg} \mathbf{A}_i^T \tag{86}$$

or

$$\mathbf{C}_{ee,i} = \sigma^2 \mathbf{I} + \mathbf{A}_i \mathbf{C}_{gg} \mathbf{A}_i^T \tag{87}$$

according to whether or not point i is in the current solution. (If it is not, (87) follows because the solution presumably is uncorrelated with this point, so the covariance matrices add. If it is in the solution, (86) follows because the covariance matrix of the residuals is the difference between the covariance matrices of the observations and of the adjusted observations (Mikhail, 1976).)

In order to find the most likely candidate for rejecting of those points currently used, the one with the largest value of r_i from (86) and (85) is found (since the large residual compared to its variance indicates that this point disagrees the most with the others). But this point is only tentatively rejected, since it is influencing the variance estimate. The solution then is recomputed without this point, (87) and (85) are used, and, if $r_i > 16$, the point is rejected. (This is equivalent to using a 4-sigma threshold. This is practical because the variance estimate now is not corrupted by this point if it is erroneous, and it is reasonable in two-dimensions when not many erroneous point are expected.) Fig. 7 illustrates this process.

6. Use of Camera Model

6.1. Projecting from Object Space to Image Space

When a point $\mathbf{p}_i$ in three-dimensional space is available and it is desired to compute its projection into an image, $\mathbf{p}_i$ and the camera model computed in Section 5 are used in the equations in Section 3 to compute $\hat{x}_i$ and $\hat{y}_i$.

Often the partial derivatives of the above projection are needed (for error propagation or in a least-squares adjustment). These are obtained as follows:

$$\frac{\partial \hat{x}_i}{\partial \mathbf{p}_i} = \frac{\partial \hat{x}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \mathbf{p}_i} \tag{88}$$

$$\frac{\partial \hat{y}_i}{\partial \mathbf{p}_i} = \frac{\partial \hat{y}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial \mathbf{p}_i} \tag{89}$$

in terms of the partial derivatives defined in Section 3.

For most applications not much accuracy is needed in partial derivatives. Therefore, when a normal lens is used ($L \approx 1$), the radial distortion is small, and the effect of entrance pupil movement is small, sufficient accuracy may be obtained by assuming that $\partial \mathbf{r}'_i / \partial \mathbf{p}_i$ is the identity matrix, so that the following results:

$$\frac{\partial \hat{x}_i}{\partial \mathbf{p}_i} \approx \frac{\partial \hat{x}_i}{\partial \mathbf{r}'_i} \tag{90}$$

$$\frac{\partial \hat{y}_i}{\partial \mathbf{p}_i} \approx \frac{\partial \hat{y}_i}{\partial \mathbf{r}'_i} \tag{91}$$

Using these approximations where appropriate may result in significant savings in time if many points are to be projected, since the computation of $\partial \mathbf{r}'_i / \partial \mathbf{p}_i$ is more involved than is the computation of $\partial \hat{x}_i / \partial \mathbf{r}'_i$ and $\partial \hat{y}_i / \partial \mathbf{r}'_i$, as can be seen in Section 3.

6.2. Projecting from Image Space to Object Space

Sometimes a point in the image (x_i and y_i) is given and it is desired to project it as a ray in space, represented by the unit vector $\mathbf{r}_i$ emanating from the point $\mathbf{c}'_i$. This can be done by the method in this section.

First, the ray is projected according to the Yakimovsky and Cunningham model (neglecting distortion and the other complications), as follows:

$$\mathbf{r}'_i = \frac{(\mathbf{v} - y_i\mathbf{a}) \times (\mathbf{h} - x_i\mathbf{a})}{\mathbf{a} \cdot \mathbf{v} \times \mathbf{h}} \tag{92}$$

This ray is not a unit vector. Instead, the component of $\mathbf{r}'_i$ parallel to $\mathbf{a}$ is 1. This temporary normalization is caused by the presence of the denominator above, which is included to cause the vector to point outwards regardless of whether the camera-model vectors form a left-handed or right-handed system.

Then the distortion, fish-eye effect, and entrance pupil movement all are computed by the following method. From Fig. 1 we have

$$\zeta'_i = \mathbf{r}'_i \cdot \mathbf{o} \tag{93}$$

$$\boldsymbol{\lambda}_i' = \mathbf{r}_i' - \zeta_i' \mathbf{o} \tag{94}$$

$$\lambda_i' = |\boldsymbol{\lambda}_i'| \tag{95}$$

$$\chi_i' = \frac{\lambda_i'}{\zeta_i'} \tag{96}$$

The following equation from (9) is solved for χ_i by using Newton's method (with χ_i' as the initial approximation and 10^{-8} as the convergence criterion):

$$(1+\rho_0)\chi_i + \rho_1\chi_i^3 + \rho_2\chi_i^5 + \ldots = \chi_i' \tag{97}$$

Then (6) is solved for θ_i to produce

$$\begin{aligned} \theta_i &= \frac{\arcsin L\chi_i}{L} && \text{if } L<0 \\ \theta_i &= \chi_i && \text{if } L=0 \\ \theta_i &= \frac{\arctan L\chi_i}{L} && \text{if } L>0 \end{aligned} \tag{98}$$

The ray is considered to be projected from the point $\mathbf{c}_i'$, obtained according to (2) and (1). The desired unit vector along the ray is

$$\mathbf{r}_i = \text{unit}(\boldsymbol{\lambda}_i') \sin\theta_i + \mathbf{o}\cos\theta_i \tag{99}$$

The partial derivatives of the projection into a ray can be obtained as follows. Differentiating (92) produces the following:

$$\frac{\partial \mathbf{r}_i'}{\partial x_i} = -\frac{\mathbf{v}\times\mathbf{a}}{\mathbf{a}\cdot\mathbf{v}\times\mathbf{h}} \tag{100}$$

$$\frac{\partial \mathbf{r}_i'}{\partial y_i} = \frac{\mathbf{h}\times\mathbf{a}}{\mathbf{a}\cdot\mathbf{v}\times\mathbf{h}} \tag{101}$$

Some intermediate derivatives are computed, including ν_i according to (23), ψ_i according to (21), and the following (obtained by using (9), (23), and (2)):

$$\begin{aligned} \omega_i &= \frac{\partial \chi_i'}{\partial \chi_i} \\ &= 1 + \rho_0 + 3\rho_1\chi_i^2 + 5\chi_i^4 + \ldots \\ &= \nu_i\chi + \frac{\chi_i'}{\chi_i} \end{aligned} \tag{102}$$

$$
\begin{aligned}
t_i &= \frac{\partial s_i}{\partial \theta_i} \\
&= \left(\frac{1}{\sin\theta_i} - \frac{\theta_i \cos\theta_i}{\sin^2\theta_i} \right)(\varepsilon_0 + \varepsilon_1\theta_i^2 + \varepsilon_2\theta_i^4 + \ldots) + \left(\frac{\theta_i}{\sin\theta_i} - 1 \right)(2\varepsilon_1\theta_i + 4\varepsilon_2\theta_i^3 + \ldots) \\
&= \frac{\lambda_i}{\sin^2\theta_i} - \frac{\upsilon_i}{\sin\theta_i}
\end{aligned}
\tag{103}
$$

(The last form of (103) shows a relationship to some previously defined quantities for informational purposes, but it may not be useful for computation here). The following can be derived from (93), (94), and (95) analogously to the way that (30) and (33) were derived from (15), (16), and (17):

$$
\frac{\partial \boldsymbol{\lambda}_i'}{\partial \mathbf{r}_i'} = \mathbf{I} - \mathbf{o}\mathbf{o}^{\mathrm{T}} \tag{104}
$$

$$
\frac{\partial \lambda_i'}{\partial \mathbf{r}_i'} = \mathrm{unit}(\boldsymbol{\lambda}_i')^{\mathrm{T}} \tag{105}
$$

Differentiating (96) produces

$$
\begin{aligned}
\frac{\partial \chi_i'}{\partial \mathbf{r}_i'} &= \frac{1}{\zeta_i'}\frac{\partial \lambda_i'}{\partial \mathbf{r}_i'} - \frac{\lambda_i'}{\zeta_i'^2}\frac{\partial \zeta_i'}{\partial \mathbf{r}_i'} \\
&= \frac{1}{\zeta_i'}\mathrm{unit}(\boldsymbol{\lambda}_i')^{\mathrm{T}} - \frac{\lambda_i'}{\zeta_i'^2}\mathbf{o}^{\mathrm{T}}
\end{aligned}
\tag{106}
$$

by using (93) and (105). The definition of the unit vector produces the following identity:

$$
\frac{\partial\, \mathrm{unit}(\boldsymbol{\lambda}_i')}{\partial \boldsymbol{\lambda}_i'} = \frac{1}{\lambda_i'}(\mathbf{I} - \mathrm{unit}(\boldsymbol{\lambda}_i')\,\mathrm{unit}(\boldsymbol{\lambda}_i')^{\mathrm{T}}) \tag{107}
$$

Differentiating (1) and (99) produces the following two equations:

$$
\frac{\partial \mathbf{c}_i'}{\partial \mathbf{r}_i'} = \mathbf{o}\frac{\partial s_i}{\partial \theta_i}\frac{\partial \theta_i}{\partial \chi_i}\frac{\partial \chi_i}{\partial \chi_i'}\frac{\partial \chi_i'}{\partial \mathbf{r}_i'} \tag{108}
$$

$$
\frac{\partial \mathbf{r}_i}{\partial \mathbf{r}_i'} = \sin\theta_i \frac{\partial\, \mathrm{unit}(\boldsymbol{\lambda}_i')}{\partial \mathbf{r}_i'} + (\mathrm{unit}(\boldsymbol{\lambda}_i')\cos\theta_i - \mathbf{o}\sin\theta_i)\frac{\partial \theta_i}{\partial \chi_i}\frac{\partial \chi_i}{\partial \chi_i'}\frac{\partial \chi_i'}{\partial \mathbf{r}_i'} \tag{109}
$$

The needed derivative of $\mathrm{unit}(\boldsymbol{\lambda}_i')$ can be obtained as follows:

$$\frac{\partial \text{unit}(\boldsymbol{\lambda}_i')}{\partial \mathbf{r}_i'} = \frac{\partial \text{unit}(\boldsymbol{\lambda}_i')}{\partial \boldsymbol{\lambda}_i'} \frac{\partial \boldsymbol{\lambda}_i'}{\partial \mathbf{r}_i'}$$

$$= \frac{1}{\lambda_i'}(\mathbf{I} - \text{unit}(\boldsymbol{\lambda}_i') \text{unit}(\boldsymbol{\lambda}_i')^{\mathrm{T}})(\mathbf{I} - \mathbf{o}\mathbf{o}^{\mathrm{T}})$$

$$= \frac{1}{\lambda_i'}(\mathbf{I} - \text{unit}(\boldsymbol{\lambda}_i') \text{unit}(\boldsymbol{\lambda}_i')^{\mathrm{T}} - \mathbf{o}\mathbf{o}^{\mathrm{T}}) \tag{110}$$

by substitution from (107) and (104), and by using the fact that $\boldsymbol{\lambda}_i'$ and $\mathbf{o}$ are orthogonal. Then, substituting (106) and (110) into (108) and (109), substituting the symbols for the other derivatives from (103), (20), and (102), and using the relationship from (96) and the fact that $\text{unit}(\boldsymbol{\lambda}_i') = \boldsymbol{\lambda}_i'/\lambda_i'$ produces the following desired results:

$$\frac{\partial \mathbf{c}_i'}{\partial \mathbf{r}_i'} = \frac{t_i}{\psi \zeta_i' \lambda_i' \omega_i} \mathbf{o}\boldsymbol{\lambda}_i'^{\mathrm{T}} - \frac{t_i \chi_i'}{\psi \zeta_i' \omega_i} \mathbf{o}\mathbf{o}^{\mathrm{T}} \tag{111}$$

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{r}_i'} = \frac{\sin\theta_i}{\lambda_i'} \mathbf{I} + \left(\frac{\lambda_i' \sin\theta_i}{\psi_i \zeta_i'^2 \omega_i} - \frac{\sin\theta_i}{\lambda_i'} \right) \mathbf{o}\mathbf{o}^{\mathrm{T}}$$
$$- \frac{\cos\theta_i}{\psi_i \zeta_i'^2 \omega_i} \boldsymbol{\lambda}_i' \mathbf{o}^{\mathrm{T}} - \frac{\sin\theta_i}{\psi_i \zeta_i' \lambda_i' \omega_i} \mathbf{o}\boldsymbol{\lambda}_i'^{\mathrm{T}}$$
$$+ \left(\frac{\cos\theta_i}{\psi_i \zeta_i' \lambda_i'^2 \omega_i} - \frac{\sin\theta_i}{\lambda_i'^3} \right) \boldsymbol{\lambda}_i' \boldsymbol{\lambda}_i'^{\mathrm{T}} \tag{112}$$

As in Section 3.2, indeterminacy at zero angle is avoided by taking the limit, which produces the following to be used instead of the above:

$$\mathbf{c}_i' = \mathbf{c} \tag{113}$$

$$\mathbf{r}_i = \mathbf{o} \tag{114}$$

$$\frac{\partial \mathbf{c}_i'}{\partial \mathbf{r}_i'} = \mathbf{O} \tag{115}$$

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{r}_i'} = \frac{1}{|\mathbf{r}_i'|(1+\rho_0)} (\mathbf{I} - \mathbf{o}\mathbf{o}^{\mathrm{T}}) \tag{116}$$

These equations are used when $\chi_i' < 10^{-8}$.

Some computation time can be saved by using the following approximations, which can be obtained by setting $L = 1$, $\boldsymbol{\rho} = \mathbf{0}$, and $\boldsymbol{\varepsilon} = \mathbf{0}$ above, and which therefore might be suitable if a normal lens is used, the radial distortion is small, and the effect of entrance pupil movement is small:

$$\frac{\partial \mathbf{c}_i'}{\partial \mathbf{r}_i'} \approx \mathbf{O} \tag{117}$$

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{r}'_i} \approx \frac{1}{|\mathbf{r}'_i|}(\mathbf{I} - \cos^2\theta_i\, \mathbf{o}''_i\mathbf{o}''^{\mathrm{T}}_i - \sin\theta_i \cos\theta_i\, \boldsymbol{\lambda}''_i\mathbf{o}''^{\mathrm{T}}_i - \sin\theta_i \cos\theta_i\, \mathbf{o}''_i\boldsymbol{\lambda}''^{\mathrm{T}}_i - \sin^2\theta_i\, \boldsymbol{\lambda}''_i\boldsymbol{\lambda}''^{\mathrm{T}}_i) \quad (118)$$

(These approximations would not be appropriate for fish-eye lenses.)

In any case, the partial derivatives are combined as follows:

$$\frac{\partial \mathbf{c}'_i}{\partial x_i} = \frac{\partial \mathbf{c}'_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial x_i} \quad (119)$$

$$\frac{\partial \mathbf{c}'_i}{\partial y_i} = \frac{\partial \mathbf{c}'_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial y_i} \quad (120)$$

$$\frac{\partial \mathbf{r}_i}{\partial x_i} = \frac{\partial \mathbf{r}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial x_i} \quad (121)$$

$$\frac{\partial \mathbf{r}_i}{\partial y_i} = \frac{\partial \mathbf{r}_i}{\partial \mathbf{r}'_i}\frac{\partial \mathbf{r}'_i}{\partial y_i} \quad (122)$$

The 6-by-2 matrix of partial derivatives of the projection is then

$$\mathbf{B}_i = \begin{bmatrix} \dfrac{\partial \mathbf{c}'_i}{\partial x_i} & \dfrac{\partial \mathbf{c}'_i}{\partial y_i} \\ \dfrac{\partial \mathbf{r}_i}{\partial x_i} & \dfrac{\partial \mathbf{r}_i}{\partial y_i} \end{bmatrix} \quad (123)$$

The quantities ζ_i and υ_i above correspond to those in Section 3, provided that the 3D point is compatible with $\mathbf{r}'_i$ as used above. The value of $\mathbf{r}_i$ above does not correspond to $\mathbf{p}_i - \mathbf{c}'_i$ in Section 2, since here $\mathbf{r}_i$ has been normalized to be a unit vector.

6.3. Propagating Camera Model Uncertainty

Partial derivatives for the 3D-to-2D projection were presented in Section 6.1, and partial derivatives for the 2D-to-3D projection were presented in Section 6.2. Usually, these are what is wanted, since the measurement errors of individual points ordinarily predominate. However, occasionally it may be desired to know how uncertainty in the camera model parameters affects the projection. For this, the partial derivatives of the result of the projection with respect to the camera model parameters are needed. These will be described in this section.

For projecting from object space to image space (3D to 2D), the desired partial derivative matrix is the $\mathbf{A}_i$ matrix from (76) in the camera model adjustment, since it contains the derivatives of the image coordinates with respect to the camera model parameters.

The case of projecting from image space to object space (2D to 3D) is slightly more

complicated. It uses the same $\mathbf{A}_i$ matrix, but it also uses the $\mathbf{B}_i$ matrix computed by (123) for the 3D to 2D projection. These are multiplied together as follows, since the propagation can be considered to be first from the camera model parameters to the 2D image, and then from the image to the 3D ray:

$$\begin{bmatrix} \dfrac{\partial \mathbf{c}_i'}{\partial \mathbf{g}} \\ \dfrac{\partial \mathbf{r}_i}{\partial \mathbf{g}} \end{bmatrix} = -\mathbf{B}_i\mathbf{A}_i \tag{124}$$

where $\mathbf{g}$ denotes the camera model parameters. The minus sign occurs because, in order to keep the image coordinates constant, the changes in the parameters and in the ray must have the opposite effect in the image.

7. Possible Future Improvements

7.1. Adjusting the Basic Lens Model Parameter

Formally, it would be easy to add the automatic adjustment of the parameter L as part of the camera model solution. It would be added as another element in $\mathbf{g}$, and the partial derivatives of $\mathbf{r}_i'$ with respect to L would be needed, so that an appropriate column could be added in the $\mathbf{A}_i$ matrix in (76), as with the other parameters.

From (24) and (22),

$$\begin{aligned} \frac{\partial \mathbf{r}_i'}{\partial L} &= \frac{\partial \zeta_i'}{\partial L}\mathbf{o} \\ &= -\frac{\zeta_i'}{\chi_i}\frac{\partial \chi_i}{\partial L}\mathbf{o} \end{aligned} \tag{125}$$

From (6),

$$\begin{aligned} \frac{\partial \chi_i}{\partial L} &= \frac{L\theta_i \cos L\theta_i - \sin L\theta_i}{L^2} && \text{if } L < 0 \\ \frac{\partial \chi_i}{\partial L} &= 0 && \text{if } L = 0 \\ \frac{\partial \chi_i}{\partial L} &= \frac{L\theta_i - \sin L\theta_i \cos L\theta_i}{L^2 \cos^2 L\theta_i} && \text{if } L > 0 \end{aligned} \tag{126}$$

However, the fact that the derivatives with respect to L go to zero when $L = 0$ would cause difficulty when L is close to zero. This problem can be solved by replacing L in $\mathbf{g}$ with a new

parameter Λ, such that

$$\begin{aligned} L &= -\sqrt{-2\Lambda} \quad \text{if } \Lambda < 0 \\ L &= 0 \quad\quad\;\; \text{if } \Lambda = 0 \\ L &= \sqrt{\Lambda} \quad\;\; \text{if } \Lambda > 0 \end{aligned} \tag{127}$$

The reason for the factor of 2 in the first case is to avoid a discontinuity in the above derivatives at $L = 0$. By using (126) and (127), it can be seen that $\partial\chi_i/\partial\Lambda = \theta_i^3/3$ if $\Lambda = 0$, regardless of whether the limit is taken from the positive direction or the negative direction.

Combining the above produces the following:

$$\begin{aligned} \frac{\partial \mathbf{r}_i'}{\partial \Lambda} &= \frac{\zeta_i'(L\theta_i \cos L\theta_i - \sin L\theta_i)}{\chi_i L^3}\mathbf{o} \qquad \text{if } \Lambda \le 0 \\ \frac{\partial \mathbf{r}_i'}{\partial \Lambda} &= -\frac{\zeta_i'(L\theta_i - \sin L\theta_i \cos L\theta_i)}{2\chi_i L^3 \cos^2 L\theta_i}\mathbf{o} \qquad \text{if } \Lambda \ge 0 \end{aligned} \tag{128}$$

Where needed, a value for L is available from (127).

Manual tests indicate, at least in some cases, that the least-squares minimum with respect to L (or Λ) is fairly local when the distortion parameters $\boldsymbol{\rho}$ also are being adjusted. Therefore, some search may be needed to find a good initial approximation, or $\boldsymbol{\rho}$ could be held to zero until Λ has initially converged. Perhaps adjusting L is better left as an alternative to using $\boldsymbol{\rho}$, when high accuracy is not needed.

7.2. Decentering Distortion

Decentering distortion is not radially symmetrical. Therefore, if we want to include it in the camera model, a direction orthogonal to the optical axis must be adopted that is fixed relative to the camera, so that it can serve as a reference direction for parameters that describe the decentering distortion. For this purpose, the cross product between **o** and either **h** or **v** might be used. However, if the image sensor is grossly tilted relative to the lens, the chosen cross product could be zero, depending on the value of x_c or y_c. In order to avoid this possibility, the combination $(\mathbf{a}\times\mathbf{h})\times\mathbf{o}$ should be used instead. This combination cannot be zero except in the unrealistic case where **o** is parallel to the sensor plane. (By the usual rule for the vector triple product, $(\mathbf{a}\times\mathbf{h})\times\mathbf{o} = (\mathbf{a}\cdot\mathbf{o})\mathbf{h} - (\mathbf{o}\cdot\mathbf{h})\mathbf{a}$.)

Thus we have the following two unit vectors that are orthogonal to each other and to **o**:

$$\mathbf{k} = \text{unit}\left((\mathbf{a}\times\mathbf{h})\times\mathbf{o}\right) \tag{129}$$

$$\mathbf{l} = \mathbf{o}\times\mathbf{k} \tag{130}$$

Then the dot products of these vectors with either $\mathbf{p}_i - \mathbf{c}$ or $\boldsymbol{\lambda}_i$ produces the components of

distance to point i along each these vectors, and dividing each of these by ζ_i' produces the tangent of the angle subtended by each of the components at $\mathbf{c}_i'$, as follows:

$$k_i = \frac{\mathbf{k}\cdot\boldsymbol{\lambda}_i}{\zeta_i'} \tag{131}$$

$$l_i = \frac{\mathbf{l}\cdot\boldsymbol{\lambda}_i}{\zeta_i'} \tag{132}$$

Note that $k_i^2 + l_i^2 = \chi_i^2$. (We need both of these components, since decentering distortion has both radial and tangential components.)

For general distortion, all powers and products of k_i and l_i up to a desired degree would be needed for each of two components. However, Brown (1966) has shown that decentering distortion can be well modeled more simply (although he was not concerned with fish-eye lenses). Adapting his method from the use there in the image plane to the use here in the plane defined by **k** and **l** produces the following for the decentering correction:

$$\begin{aligned}\boldsymbol{\eta}_i = {} & \zeta_i'\left((3k_i^2 + l_i^2)\delta_0 + 2k_i l_i \delta_1\right)(1 + \chi_i^2\delta_2 + \chi_i^4\delta_3 + \ldots)\,\mathbf{k} \\ & + \zeta_i'\left(2k_i l_i \delta_0 + (k_i^2 + 3l_i^2)\delta_1\right)(1 + \chi_i^2\delta_2 + \chi_i^4\delta_3 + \ldots)\,\mathbf{l}\end{aligned} \tag{133}$$

where the deltas are the decentering parameters, which are assembled into $\boldsymbol{\delta} = [\delta_0\ \delta_1\ \delta_2\ \ldots]^T$ and become part of the camera model. The number of deltas is called D, and we must have $D \neq 1$, since both δ_0 and δ_1 must be present if the decentering correction is done. (Note that only terms of even degree are used.)

The vector $\boldsymbol{\eta}_i$ from (133) would be an additional correction included in $\mathbf{r}_i'$. Therefore, instead of (24), the following would be used:

$$\mathbf{r}_i' = \zeta_i'\mathbf{o} + (1 + \mu_i)\boldsymbol{\lambda}_i + \boldsymbol{\eta}_i \tag{134}$$

Partial derivatives with respect to $\boldsymbol{\delta}$ can be obtained from (133) and (134), to produce the following:

$$\frac{\partial \mathbf{r}_i'}{\partial \delta_0} = \zeta_i'(3k_i^2 + l_i^2)(1 + \chi_i^2\delta_2 + \chi_i^4\delta_3 + \ldots)\,\mathbf{k} + 2\zeta_i' k_i l_i (1 + \chi_i^2\delta_2 + \chi_i^4\delta_3 + \ldots)\,\mathbf{l} \tag{135}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \delta_1} = 2\zeta_i' k_i l_i (1 + \chi_i^2\delta_2 + \chi_i^4\delta_3 + \ldots)\,\mathbf{k} + \zeta_i'(k_i^2 + 3l_i^2)(1 + \chi_i^2\delta_2 + \chi_i^4\delta_3 + \ldots)\,\mathbf{l} \tag{136}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \delta_2} = \zeta_i'\left((3k_i^2 + l_i^2)\delta_0 + 2k_i l_i \delta_1\right)\chi_i^2\,\mathbf{k} + \zeta_i'\left(2k_i l_i \delta_0 + (k_i^2 + 3l_i^2)\delta_1\right)\chi_i^2\,\mathbf{l} \tag{137}$$

$$\frac{\partial \mathbf{r}_i'}{\partial \delta_3} = \zeta_i'\left((3k_i^2 + l_i^2)\delta_0 + 2k_i l_i \delta_1\right)\chi_i^4\,\mathbf{k} + \zeta_i'\left(2k_i l_i \delta_0 + (k_i^2 + 3l_i^2)\delta_1\right)\chi_i^4\,\mathbf{l} \tag{138}$$

and so forth.

In the camera model adjustment, zero initial approximations can be used for all of the deltas. However, as Brown has pointed out, some a priori weight must be applied to at least δ_2 and those beyond (if they are used), because otherwise they would be indeterminate on the first iteration, when δ_0 and δ_1 are zero.

Strictly speaking, the partial derivatives with respect to the other camera model parameters should be modified to include the effect that they have on $\mathbf{r}'_i$ through (27) and (129-134). However, since decentering usually is a small effect, and since high accuracy is not needed in partial derivatives, this effect can be ignored for most purposes.

Decentering also affects the position of the entrance pupil, so that its shift with off-axis angle is no longer symmetrical. However, the resulting effect in the image is very small except for objects very close to a fish-eye lens. (For the Mars rover HazCam, the effect is roughly estimated to have a maximum of about a tenth of a pixel.) Therefore, we ignore it here, but a highly accurate calibration might need to include it.

7.3. Other Possible Improvements

In addition to the above changes to the camera model itself, some improvements are possible in the adjustment of the camera model and in the type of data that it uses.

One possibility is to improve the initial approximation for the iterative solution. An approximate closed-form camera model solution that uses all of the input points and produces results that include the camera position could be used to produce an initial approximation (e.g. Zhang, 1999). This would eliminate the need for any manual input, and it might produce a better approximation that would reduce the number of iterations required in the main solution.

Another possibility is to use images of a calibration fixture in several unknown locations (e.g. Zhang, 1999). In some cases this could simplify the calibration process by eliminating the need for measuring the pose of the fixture or using a fixture with multiple planes, although probably with reduced accuracy. For each position other than one that serves as a reference, six more parameters representing the pose of the fixture would be included in the $\mathbf{g}$ vector to be solved for. Partial derivatives relative to these can be obtained easily by multiplying the results of (88) and (89) by the matrix of partial derivatives of the coordinates of the observed points on the fixture relative to the parameters that represent the pose of the fixture. (Alternatively to considering one fixture pose to be known, the camera pose could be considered to be known.)

A more extreme possibility is to use many unknown points that are viewed by the camera in several different poses of either the camera or a rigid structure containing the points. This technique often is called "self-calibration" or "analytical self-calibration" (Clarke and Fryer, 1998; Kenefick et al., 1972), although the former term is used by some with a different meaning.

In this case there would be three parameters for each point (for its position relative to a reference point) and six parameters for each pose, all to be included in **g**. In some cases a good solution still is possible if a distance is supplied to determine the overall scale. The number of unknowns is large, but by suitable partitioning, the sparseness of some of the matrices can be utilized in order to reduce the computational burden, as in some similar cases (Gennery, 1986; Mikhail, 1976); however, this would necessitate reformulating some of the equations in Section 5.

8. Examples

Calibration data for the cameras of the 2003 Mars rover will be obtained by using a calibration fixture designed by Mark Maimone. (This fixture contains dots on three orthogonal planes. One position of it suffices in principle, but for greater accuracy it can be moved to several precisely measured positions.) However, in order to evaluate the effect of decentering, some ray tracing experiments were done using the design for the HazCam. (Since this camera uses a fish-eye lens, the effect of decentering on it is apt to be the most severe among the Mars rover cameras.)

For this purpose, simulated calibration data was produced for the HazCam, of a simpler nature than that from the actual fixture. An 11-by-11 array of points was generated in the image plane (approximately, but not exactly, equally spaced). For each point, three-dimensional points were generated at five different ranges (approximately 0.1, 0.3 0.5, 0.7, and 0.9 meters), so that there were 605 calibration points in all. Several sets of such data were generated, using the nominal lens design and using the different lens elements misadjusted. Decentering the third element seemed to produce the largest effect.

When a camera-model adjustment was done using the generated data with no decentering, the largest residual was 0.019 pixels. (This value is the result of the higher-order distortion and pupil-movement terms not included in the model, which uses $R = 3$ and $E = 3$).

Fig. 8 shows the effect of decentering the third lens element by 0.025 mm, which is the design tolerance. The square outline denotes the area of the 1024-by-1024 CCD. In this figure, the positions of the images of the calibration points with decentering are compared to the position computed by the adjusted camera model without decentering (shown by the small circles), and the differences (magnified times 10) are shown as the straight lines emanating from the centers of the circles. The largest difference is 5.608 pixels.

In the same way (but with a magnification of 50), Fig. 9 shows the effect of allowing the position of the center pixel to be adjusted in the camera model when using the decentered data, but keeping the other camera model parameters fixed to their values from the adjustment to the

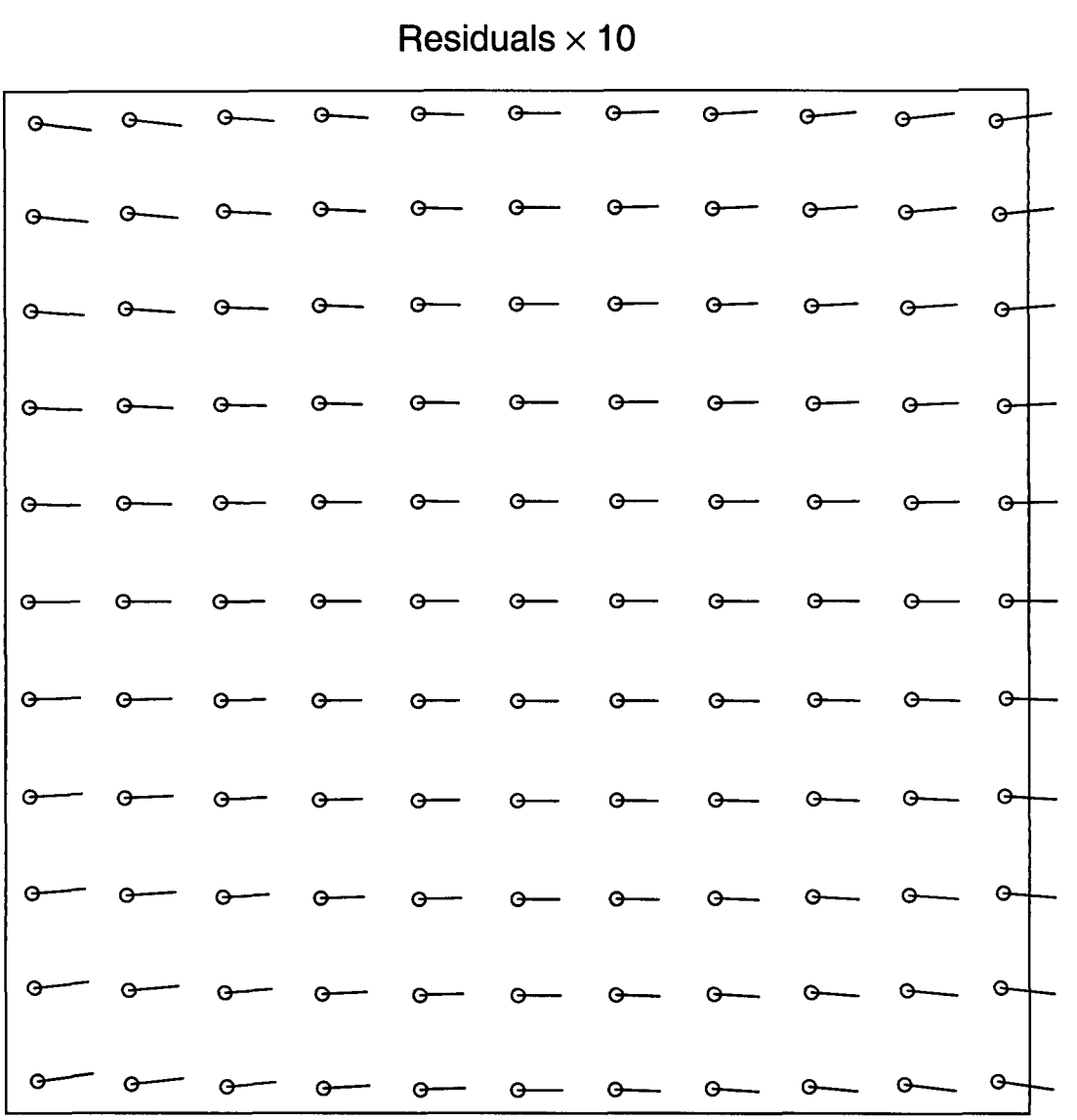

Figure 8. Magnified residuals when ray-tracing results from decentering the third lens element of the HazCam by 0.025 mm are compared to the camera model fit to data without decentering.

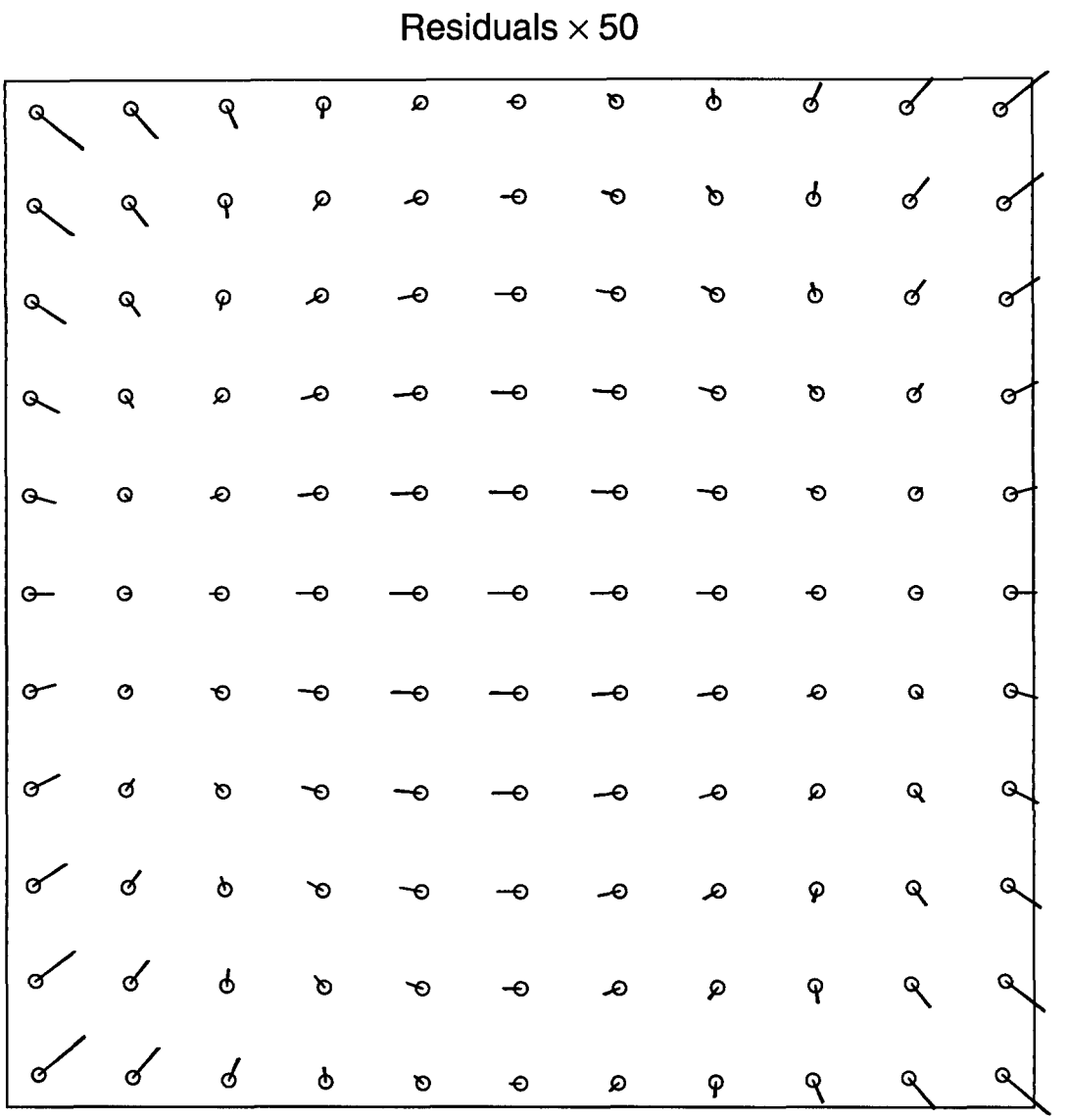

Figure 9. Magnified residuals when ray-tracing results from decentering the third lens element of the HazCam by 0.025 mm are compared to the camera model fit to data without decentering, with bias (center pixel) adjusted to minimize the sum of square of residuals.

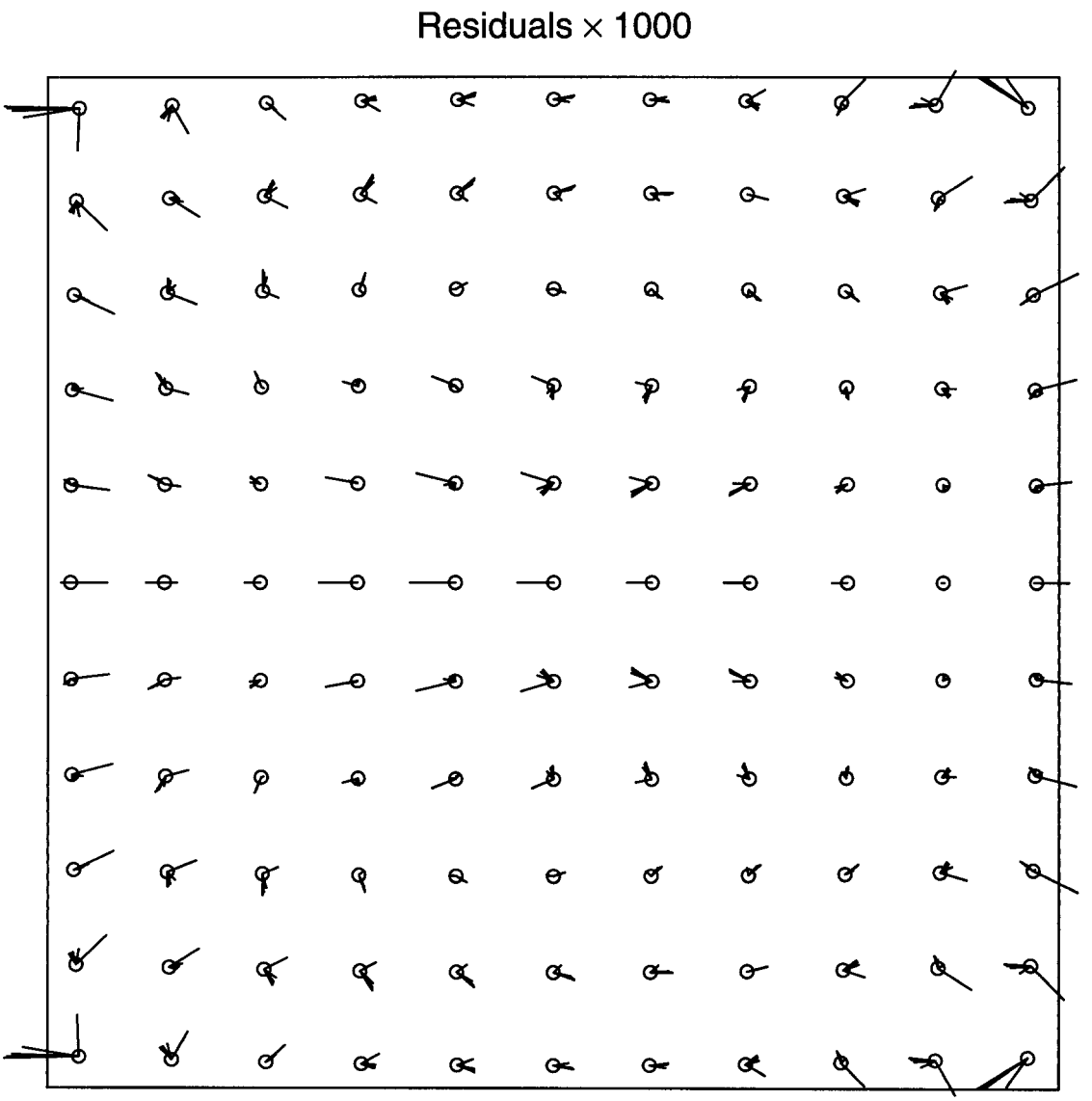

Figure 10. Greatly magnified residuals of the camera model fit to ray-tracing results from decentering the third lens element of the HazCam by 0.025 mm.

data without decentering. In this case the largest difference is 1.226 pixels.

Fig. 10 shows the effect of allowing a full camera model adjustment (which does not include decentering) to the same decentered data. Now the largest residual is only 0.075 pixels. (In this figure, the residuals are magnified times 1000, and the different residuals for the points at different distances can be distinguished.) In this case the adjustment changed the estimates of not only the center pixel but also the orientations of the lens and image sensor plane, in order to compensate for the decentering.

Because of the small residuals in this last case, there are no plans to include decentering distortion in the camera models for the 2003 Mars rover.

9. Summary and Conclusions

A camera model that includes both fish-eye and normal lenses as special cases, additional radial distortion terms, and entrance pupil movement has been described, along with algorithms for calibrating and using the model. These will be used for the 2003 Mars Exploration Rovers. If the improvements in Section 7 concerning decentering distortion and a more general type of calibration data are included, the camera model and methods described in this paper should be appropriate for a wide variety of camera calibration tasks for many years to come.

Acknowledgements

The research described herein was carried out for the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The camera calibration program and related software were written by Todd Litwin, and he has produced many calibration runs. Yalin Xiong made useful suggestions concerning fish-eye lenses and the movement of entrance pupils. Other helpful suggestions were made by Reg Willson and Mark Maimone. Reg Willson did the simulated calibration runs used as examples here. The ray tracing simulations for temperature effects were done by Larry Scherr.

References

Bard, Y. 1974. *Nonlinear Parameter Estimation.* Academic Press.

Brown, D. C. 1966. Decentering distortion of lenses. *Photogrammetric Engineering*, 32:444-462.

Brown, D. C. 1971. Close-range camera calibration. *Photogrammetric Engineering*, 37:855-866.

Clarke, T. A. and Fryer, J. G. 1998. The development of camera calibration methods and models. *Photogrammetric Record*, 16:51-66.

Eisenman, A. R., Liebe, C. C., Maimone, M. W., Schwochert, M. A., and Willson, R. G. 2001. Mars exploration rover engineering cameras. In *Proceedings of the SPIE #4540: Sensors, Systems, and Next-Generation Satellites V*, Toulouse, France, pp. 288-297.

Gennery, D. B. 1980. Modelling the environment of an exploring vehicle by means of stereo vision. Computer Science Dept., Stanford University, AIM-339 (STAN-CS-80-805).

Gennery, D. B. 1986. Stereo vision for the acquisition and tracking of moving three-dimensional objects. In *Techniques for 3-D Machine Perception* (A. Rosenfeld, ed.). Elsevier Science Publishers B. V. (North-Holland). Pp. 53-74.

Gennery, D. B. 1991. Camera calibration including lens distortion. Jet Propulsion Laboratory, Pasadena, CA, JPL internal report D-8580.

Gennery, D. B. 2001. Least-squares camera calibration including lens distortion and automatic editing of calibration points. In *Calibration and Orientation of Cameras in Computer Vision* (A. Gruen and T. S. Huang, eds.). Springer-Verlag. Pp. 123-136.

Gennery, D. B., Litwin, T., Wilcox, B., and Bon, B. 1987. Sensing and perception research for space telerobotics at JPL. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, pp. 311-317.

Gruen, A. and Huang, T. S. (eds.) 2001. *Calibration and Orientation of Cameras in Computer Vision*. Springer-Verlag.

Jenkins, F. A. and White, H. E. 1976. *Fundamentals of Optics*, Fourth Edition. McGraw-Hill.

Kenefick, J. F., Gyer, M. S., and Harp, B. F. 1972. Analytical self-calibration. *Photogrammetric Engineering*, 38:1117-1126.

Laikin, M. 2001. *Lens Design*, Third Edition. Marcel Dekker.

Mikhail, E. M. (with contributions by F. Ackermann) 1976. *Observations and Least Squares*, Harper and Row.

Miyamoto, K. 1964. Fish eye lens. *Journal of the Optical Society of America*, 54:1060-1061.

Shah, S. and Aggarwal, J. K. 1996. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition*, 29:1775-1788.

Smith, G. H., Hagerott, E. C., Scherr, L. M., Herkenhoff, K. E., and Bell, J. F. III. 2001. Optical designs for the Mars '03 rover cameras. In *Proceedings of the SPIE #4441: Current Developments in Lens Design and Optical Engineering II*, San Diego CA, pp. 118-131.

Stevenson, D. E. and Fleck, M. M. 1995. Nonparametric correction of distortion. Computer Science Dept., University of Iowa, Technical Report 95-07.

Willson, R. and Shafer, S. A. 2001. Modeling and Calibration of Variable-Parameter Camera Systems. In *Calibration and Orientation of Cameras in Computer Vision* (A. Gruen and T. S. Huang, eds.). Springer-Verlag. Pp. 137-161.

Yakimovsky, Y. and Cunningham, R. T. 1978. A System for Extracting three-dimensional measurements from a stereo pair of TV cameras. *Computer Graphics and Image Processing*, 7:195-210.

Zhang, Z. 1999. Flexible camera calibration by viewing a plane from unknown orientations. In *Proc. 7th Int. Conf. on Computer Vision*, Kerkyra, Greece, pp. 666-673.