Computer Science
Technical Report

Colorado State University

Approximate Image Mappings Between Nearly Boresight Aligned Optical and Range Sensors

J. Ross Beveridge, Zhongfei Zhang, Mike Goss,
Mark R. Stevens, A. Schwickerath

April 16, 1996

Technical Report CS-96-112

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (970) 491-5792 Fax: (970) 491-2466
WWW: http://www.cs.colostate.edu

Approximate Image Mappings Between Nearly Boresight Aligned Optical and Range Sensors *

J. Ross Beveridge
Colorado State University
ross@cs.colostate.edu

Zhongfei Zhang
SUNY Buffalo
zhongfei@cedar.buffalo.edu

Mike Goss
Hewlett Packard Labs
goss@hpl.hp.com

Mark R. Stevens
Colorado State University
stevensm@cs.colostate.edu

A. Schwickerath
Colorado State University
schwicke@cs.colostate.edu

April 16, 1996

Abstract

This technical report summarizes the intrinsic sensor parameters for the range, IR and color sensor used in the Fort Carson data collection and explores sensor-to-sensor image mapping under different assumptions regarding relative sensor placement. The default case is to assume perfectly boresight aligned placement, and then the implications of different deviations from this perfect placement are considered. Included in this report is a description of the calibration process used to recover the color sensor parameters. A key result shown in detail is the relative equivalence of planar sensor translation and small angle pan and tilt for points of known depth. This simplifying approximation has significant implications when fusing data from separate range and optical sensors.

*This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under grants DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office

Contents

List of Tables

List of Figures

1 Introduction

In mapping data from an optical sensor onto a range sensor, such as a LADAR, intrinsic sensor calibration as well as extrinsic sensor pose must be taken into account. This paper looks at various aspects of the relative geometric transformations between sensors which are nearly, but not completely, boresight aligned. The practical motivation for this work is to explore the how these mapping change given different types of deviation from the perfect case and from this exploration draw some conclusions about when certain simplifying assumptions can be used without introducing excessive error into the sensor-to-sensor pixel mapping.

The intention is to present this material at a detailed, nearly tutorial level. All but the section on calibration should be readily understood by any reader who possesses basic knowledge of trigonometry and linear algebra. Consequently, there is a risk the presentation may appear a bit labored to the reader intimately familiar with these issues. Those with such backgrounds either do not need this background or should read quickly.

Several highly practical considerations have brought this paper into existence. First, much of the work at Colorado State on multi-sensor fusion [SB94, BHP95, J. 96, Ant96] has been implicitly using assumptions fleshed out and tested in this paper. The foremost such assumption is that changes in pixel mappings between sensor induced by small rotations of one sensor relative to another may, under a limited set of conditions, be expressed as planar translation of one image relative to another. To many familiar with the geometry involved the conclusion is self evident. However, it is important to understand just how good an approximation this is and thus what magnitude of error to expect under different practical operating conditions. One way to view this technical report is as a long tutorial working up to Section 6 which answers this question.

Another motivation for this technical report is to better record and understand the characteristics of the sensors used in the Fort Carson data collection [BPY94]. In November of 1993 Colorado State University, Alliant Techsystems and Martin Marrietta jointly collected a set of range, IR and color data at the Colorado National Guard Facility at Fort Carson, Colorado. Over 400 range images were collected in such a manner as to approximate 3 boresighted sensors. This technical report contains estimates of the intrinsic sensor parameters for the Fort Carson data and these sensors are used for illustration throughout the report. This data is now publicly available and may be down-loaded from our web site: `http://www.cs.colostate.edu/~vision/`. Anyone using this data may find this report helpful.

2 Overview

Section 3 reviews and defines the intrinsic parameters of a perspective or pin hole camera model. It also presents two ways of deriving these intrinsic parameters. The first and obviously superior way is through calibration and Section 3.2 presents details on the exact calibration technique used to recover the intrinsic parameters for the color data collected at Fort Carson. The second way is to compute them from information commonly provided by a manufacturer, and this is reviewed in Section 3.3.

Section 4 lays to rest a key detail relating to some range sensors including the one used in the Fort Carson data collection. Based upon the physics of the actual range sensor, we have been told that a spherical mapping is a more accurate description of the image, i.e. the pixels spacing uniform in angle. In this section the spherical mapping is compared to the most closely equivalent pin hole camera model and it is concluded the difference in pixel mappings for common points in the world never exceeds 0.15 pixel units for the LADAR used at Fort Carson. Based upon this analysis, we conclude that the pin-hole model is perfectly acceptable for this sensor and that the distinction does not matter for the field of view and pixel resolution in question.

Section 5 takes up the key question of how do different deviations from perfect bore-sight alignment alter the mapping between sensor image planes. If one sensor rotates about the about the optical axis relative to the other, the mapping between sensors remains 2D affine for all points in the world. When the horizontal and vertical scale factors are identical no warping is involved and the rotation angle is preserved in the 2D mapping. If one sensor translates forward or backward relative to the other, there is no single 2D affine

mapping for all 3D points. There is a mapping involving scaling and translation in the image plane where the translation is independent of depth and the scaling is dependent upon depth. If one sensor translates relative to another in a common image plane, again the 2D mapping is not independent of the depth of points in the world. In this case, there is scale change which is independent of depth and the translation term depends upon depth to the points. Finally, the case of one sensor rotating about the horizontal and vertical axes relative to another is considered. Under these conditions, the 2D mapping between images becomes quite complex and dependent upon the full 3D coordinates of the point being viewed. Unlike the previous cases, it is not longer helpful to attempt to derive a 2D mapping between images.

Understanding that minor shifts in pan and tilt angles of one sensor relative to another introduce a quite complicated mapping between image spaces, it becomes interesting to ask under what conditions such rotations may be reasonably approximated by the much simpler case of translation in a common image plane. Section 6 works up an analysis which allows us to answer this question. In this analysis, two sensors are coupled so as to track a common reference point at a fixed depth. To accomplish this, one rotates and the other translates. As is perhaps not surprising, the translation approximates the rotation almost perfectly for points at the tracking depth. When working within a narrow depth of field about the tracking depth, the approximation is likewise very good and actual values are presented in Section 6. Finally, for all depths beyond the tracking depth, the approximation introduces error. However, this error grows quickly and then begins to approach an upper bound. Thus, for points beyond the tracking point, the pixel-to-pixel error for practical purposes is bounded.

3 Optical Sensor Geometry

Let us review the basics of 3D projection as performed with a projective camera. The key mapping is between 3D points and their projection on the 2D image plane. Many texts treat this topic [FD82]. One of the most compact and simplest ways of expressing the 3D to 2D relationship closely follows concepts developed in projective geometry. The following is a general equation for projection.

$$\begin{aligned} I &= PW \\ &= \left| \begin{array}{ccc||c} s_u & 0 & t_u & X \\ 0 & s_v & t_v & Y \\ 0 & 0 & 1 & Z \end{array} \right| \end{aligned} \tag{1}$$

where I is a point on the projective image plane, P is the projection matrix and W is the 3D point being imaged.

There is a marvelous trick implicit in this technique which makes the non-linear perspective mapping amenable to a simple linear algebraic form. This trick is actually quite proper and rigorous in terms of projective geometry and is nicely explained in [Fau93]. From a mechanical standpoint, simply observe that expanding out the matrix multiplication yields:

$$I = \left| \begin{array}{c} s_u X + t_u Z \\ s_v Y + t_v Z \\ Z \end{array} \right| \tag{2}$$

The 2D point I is represented in projective coordinates in which there are an infinite number of ways to express a single point.

$$P = \alpha P \quad \forall \alpha \tag{3}$$

Given this redundancy, a normalized form is selected in which the third element must equal 1. Applying this normalization to I yields

$$I = \left| \begin{array}{c} s_u \frac{X}{Z} + t_u \\ s_v \frac{Y}{Z} + t_v \\ 1 \end{array} \right| \tag{4}$$

The $\frac{X}{Z}$ and $\frac{Y}{Z}$ may now be recognized as the ratios which are commonly used to define or explain perspective projection.

The four terms s_u, s_v, t_u and t_v define the intrinsic characteristics of the optical camera. The terms s_u and s_v are typically called the scale factors and they encode both the focal length of the sensor as well as the pixel sampling dimension. It is equally correct to interpret these parameters as the size of the horizontal and vertical focal lengths measured in pixel units [Fau93]. The terms t_u and t_v represent the coordinates of the point where the optical axis pierces the image plane as measured in pixel units. The issue of recovering these intrinsic parameters from known calibration targets is taken up in in the following section.

3.1 Calibration from Calibration Targets

The calibration work for the color imagery was performed by Zhongfei Zhang at the University of Massachusetts using a method developed earlier at UMass by Yong-Qing Cheng *et al* [CCHR94]. The resulting intrinsic parameters are presented in Table 1. There is a minor inconsistency between the imagery used for calibration and that typically distributed with the Fort Carson dataset: the latter has been cropped to the center 720x480 pixels. This does not alter the scale factors, but does alter the image center by half the cropping margin. This adjustment is reflected in the last row of Table 1.

Image Resolution	Scale Factor		Principal Point	
	s_u	s_v	t_u	t_v
767×512	978.081	947.117	368.536	243.794
720×480	"	"	345.036	227.794

Table 1: Scale factors and image center for Fort Carson Imagery

3.2 Details of The Calibration Method

This section describes briefly how these parameters were estimated. Sensor calibration is a rather complicated topic and this section will not attempt the same level of tutorial presentation used elsewhere in this report. Readers unfamiliar with calibration are encouraged to see [Gan84, LT86, STH80].

As laid out above, the camera model used in this work is assumed to be pinhole and the underlying mathematical model is a perspective transformation. Consider the case with m camera positions and n 3D points $\vec{P}_1(x_1, y_1, z_1)$, ... , $\vec{P}_n(x_n, y_n, z_n)$ in the world coordinate system. For jth position, there are n corresponding image points $\vec{Q}_1^{(j)}(u_1^{(j)}, v_1^{(j)})$, ..., $\vec{Q}_n^{(j)}(u_n^{(j)}, v_n^{(j)})$ (j = 1,2, ... , m).

Assuming that the relationship between the world coordinate system and the camera coordinate system at camera position j is:

$$\vec{P}_{ci}^{(j)} = R^j \vec{P}_i + \vec{t}^j \tag{5}$$

where $\vec{P}_{ci}^j = (x_{ci}^j, y_{ci}^j, z_{ci}^j)^T$ is the 3D coordinate vector of the ith point at jth camera coordinate system, $R^j = (\vec{r}_1^j, \vec{r}_2^j, \vec{r}_3^j)^T$ is the rotation matrix from the world coordinate system to the jth camera coordinate system, and $\vec{t}^j = (t_x^j, t_y^j, t_z^j)^T$ is the translation vector from the world coordinate system to the jth camera coordinate system. Given the above assumptions, together with equation 1, lead to the following set of constraint equations:

$$\begin{aligned} \hat{u}_i^{(j)} &= \frac{(S_u \vec{r}_1^{(j)} + t_u \vec{r}_3^{(j)}) \cdot \vec{P}_i + S_u t_x^{(j)} + t_u t_z^{(j)}}{\vec{r}_3^{(j)} \cdot \vec{P}_i + t_z^{(j)}} \\ \hat{v}_i^{(j)} &= \frac{(S_v \vec{r}_2^{(j)} + t_v \vec{r}_3^{(j)}) \cdot \vec{P}_i + S_v t_y^{(j)} + t_v t_z^{(j)}}{\vec{r}_3^{(j)} \cdot \vec{P}_i + t_z^{(j)}} \end{aligned} \tag{6}$$

where $\hat{u}_i^{(j)}$ and $\hat{v}_i^{(j)}$ are image projections of the corresponding 3D points. The camera internal parameters are S_u, S_v, t_u, t_v, and the external parameters are $\vec{r}_1^j, \vec{r}_2^j, \vec{r}_3^j, t_x^j, t_y^j, t_z^j$. Thus, taking the difference between projected and actual observed image points, we form the following general aggregate sum-of-squares of the residuals over m images:

$$\Phi_m = \sum_{j=1}^{m} \sum_{i=1}^{n} \left\{ \left(\frac{u_i^{(j)} - \hat{u}_i^{(j)}}{\sigma_{u_i^{(j)}}} \right)^2 + \left(\frac{v_i^{(j)} - \hat{v}_i^{(j)}}{\sigma_{v_i^{(j)}}} \right)^2 \right\} \tag{7}$$

where σ_u and σ_v are deviations along the two image axes, and they are set to 1 in all the tests.

The function Φ_m depends nonlinearly on $\vec{P}_i$, $\vec{Q}_i^{(j)}$, and the $4 + 6 \times m$ parameters in $\vec{\alpha}$ where

$$\vec{\alpha} \ = \ (t_u, t_v, S_u, S_v, \vec{r}_1^{(1)}, \vec{r}_2^{(1)}, \vec{r}_3^{(1)}, t_x^{(1)}, t_y^{(1)}, t_z^{(1)}, \ldots \ \vec{r}_1^{(m)}, \vec{r}_2^{(m)}, \vec{r}_3^{(m)}, t_x^{(m)}, t_y^{(m)}, t_z^{(m)})^T \tag{8}$$

Note that although $\vec{r}_1^{(j)}, \vec{r}_2^{(j)}, \vec{r}_3^{(j)}$ has nine parameters in total, they only count as three independent parameters, since there are six constraints imposed on the nine parameters in order to form a rotation matrix. See Appendix A of [Kum92] for the details of those constraints.

The function Φ_m may be expressed as the second-order Taylor series expansion

$$\Phi_m(\vec{\alpha} + \Delta\vec{\alpha}) = \Phi_m(\vec{\alpha}) + \vec{\omega}^T \Delta\vec{\alpha} + \frac{1}{2!} \Delta\vec{\alpha}^T H \Delta\vec{\alpha} \tag{9}$$

where $\vec{\alpha}$ is the initial guess and $\Delta\vec{\alpha}$ is an small correction to $\vec{\alpha}$, $\vec{\omega} = \frac{\partial \Phi}{\partial \vec{\alpha}}$ is the gradient of the objective function Φ_m with respect to $\vec{\alpha}$, and $H = \frac{\partial^2 \Phi}{\partial \vec{\alpha} \partial \vec{\alpha}^T}$ is the second derivative matrix (Hessian matrix) of the objective function Φ_m.

Here, the Levenberg-Marquardt algorithm which is a robust algorithm to solve nonlinear systems developed by Levenberg and Marquardt [Mar63, PFTV88], is used to compute the camera parameters $\vec{\alpha}$. In the Levenberg-Marquardt method, we have

$$\Delta\vec{\alpha} = -(H + \lambda I)^{-1} \vec{\omega} \tag{10}$$

where λ is a conditioning factor and I is an identity matrix.

Due to tracking in the image, without using special patterns, some 2D-3D measurements and correspondences may be incorrect. In these cases, the underlying noise in the 2D and 3D data may not be Gaussian. Hence, gross errors or outliers may occur. In order to deal with gross errors or outliers in the 2D and 3D data, the following least median of squares (LMS) estimator is used. It has been proved that the following minimization always leads to a solution [RL87]

$$\text{Minimize } {\Phi_m}^{(i)} = \text{median}_i \left\{ \left(\frac{u_i^{(j)} - \hat{u}_i^{(j)}}{\sigma_{u_i^{(j)}}} \right)^2 + \left(\frac{v_i^{(j)} - \hat{v}_i^{(j)}}{\sigma_{v_i^{(j)}}} \right)^2 \right\} \tag{11}$$

Since the median is not differentiable, $\Phi_m^{(i)}$ must be minimized using combinatorial methods such as subsampling. The algorithm based on least median of squares technique is proposed as follows:

(a) Select "l" random subsets of size "k" from the input data.

(b) For each subsample S_i, determine the camera parameters α by using the Levenberg-Marquardt algorithm. Estimate the residual error e_i for all "n" points given the camera parameters and find the median square error.

(c) Select the camera parameters which gives the minimum median error $\Phi_m^{(i)}$ and compute the scale "s" using equation:

$$s = \text{median}_i \left\{ \frac{|e_i|}{0.6745} \right\} \tag{12}$$

(d) Filter out points as outliers whose squared residual error from the camera parameters is greater than $(as)^2$; a is an algorithm parameter and is set equal to 1.5 for all tests.

(e) Minimize the error function given in equation 11 on the remaining points using the above algorithm and return the estimated camera parameters as the final output.

The calibration procedure is as follows. We take three images of a fixed calibration pattern with different depth distances from the camera. Then we extract a set of points in each image, and measure the 3D coordinates of those points relative to some arbitrary 3D coordinate system. Finally, the 2D image points and 3D world points are input into this algorithm to compute the internal parameters. In this case (one of the images is shown in Figure 1), the $(0, 0, 0)$ point is the lower corner of the intersection between the two boards, and the known quantities are the radius of the circles and the spacing between the circles (in 3D). The internal parameters estimated for the Fort Carson image data using this method are recorded in Table 1.

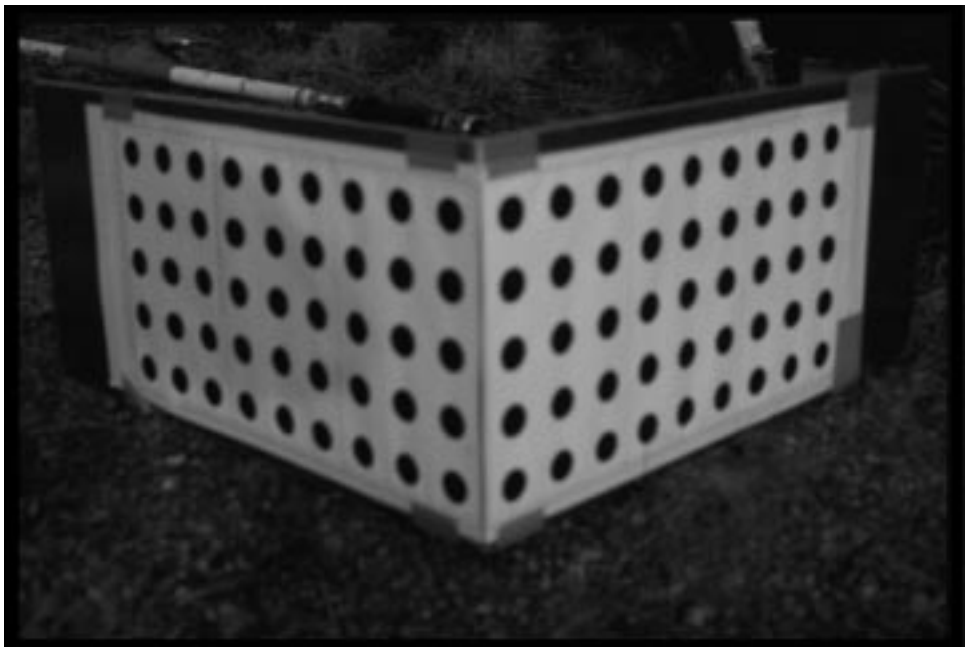


Figure 1: Geometric Camera Calibration Target for Fort Carson Data.

To have a comparison between the performance of this algorithm and those of the others, we use the same data points for Ganapathy's [Gan84] and Crowley et al's [CB93] algorithm. The following table lists the total error projected back to the image plane for each algorithm, respectively. The total error is the sum over all the points used in the calibration procedure ($n = 18$), of the error in each point.

Total Errors for the Same Point Set
unit: pixel

Ours	Ganapathy	Crowley
0.620	1.230	6.399

3.3 Intrinsic Parameters from Field of View and Image Size

Calibration as described above is clearly superior to simply estimating intrinsic parameters based upon general information provided by manufacturers. However, there are times when calibration is impractical

and one must make an intelligent guess based upon commonly provided information such as the vertical and horizontal field of view alone.

The mapping is rather straight forward, but for the sake of completeness it is presented here. Table 2 presents the conclusions drawn from this section relating field of view, image size and intrinsic parameters for each sensor used in the Fort Carson data collection. Table 2 lists the fields of view, pixel dimensions of the images, and the intrinsic parameters defined earlier. For the color and IR sensors, two entries are listed. The first entry for color lists the parameters derived from the calibration procedure described in the previous section. The second are parameters derived from the specifications of the color sensor. Specifically, the knowledge that the film is 36mm wide, 12mm high, and a 50mm lens was used.

	FOV f_u		FOV f_v		Dimensions		Scale		Center	
Sensor	Rad.	Deg.	Rad.	Deg.	d_u	d_v	s_u	s_v	t_u	t_v
Color (from Cal.)	0.705	40.4	0.496	28.4	720	480	-978	947	345.0	227.8
Color (from Specs.)	0.691	39.6	0.471	27.0	720	480	-1000	1000	359.5	239.5
FLIR (Visual Cal.)	0.434	24.9	0.401	23.0	256	256	-580	630	127.5	127.5
FLIR (from Specs)	0.419	24	0.419	24	256	256	-602	602	127.5	127.5
range	0.271	15.5	0.060	3.4	120	24	-440	400	59.5	11.5

Table 2: Intrinsic Sensor Parameters for Fort Carson Color, IR and Rang.

The first of the two entries for the FLIR are based upon the manufacturers specified field of view for the FLIR. The second incorporates a correction generated by hand based upon visual appearance of modeled 3D objects in both range and IR. This was done interactively using our own multi-sensor visualization software [GBSF95, GBSF94]. With this software, it is possible to first align a 3D object model with range data and also with IR using the manufacturers specifications for the IR sensor. Then a user can adjust the scale factors so as to make the projection of the object model more precisely match the appearance of the object in IR. This process is certainly not assured of generating the true intrinsic parameters, but it will generate compatible range and IR parameters for objects at similar depth.

The parameters for the LADAR range sensor are based upon field calibration using calibrated imagery. To calibrate the horizontal angular pixel resolution, the sensor was field tested viewing a pair of survey markers 50 feet apart at 184 feet from the sensor. To calibrate the vertical angular resolution, the senosr viewed two vertical markers 3.7 feet apart at 157 feet. The maximum range measured by the LADAR is 1074 feet, and hence multiplying a raw pixel value by the ratio 1074/4095 yields a range measurement in feet. The σ on the range meaurement is approximately 1 foot [Bel93]. More will be said about the geometry of the LADAR in the following section.

In going back and forth between alternative ways of describing a sensor, a minor point where confusion can arise concerns the exact 'position' of a pixel. The convention used here is that the pixel centers are points on the U, V image plane with **integer** coordinates. When drawing an image as a grid, this convention means the pixels centers fall at intersections of grid lines. This relationship is illustrated in Figure 2.

When going from a specification of the field of view and image dimensions, an assumption about the optical center must be made. Considering an image with dimensions (d_u, d_v), the obvious default assumption is to place the optical center at the center of the image.

$$C = \begin{vmatrix} c_u \\ c_v \\ 1 \end{vmatrix} = \begin{vmatrix} \frac{d_u - 1}{2} \\ \frac{d_v - 1}{2} \\ 1 \end{vmatrix} \tag{13}$$

The terms t_u and t_v in the projection matrix defined in equation 1 are the coordinates of the image center

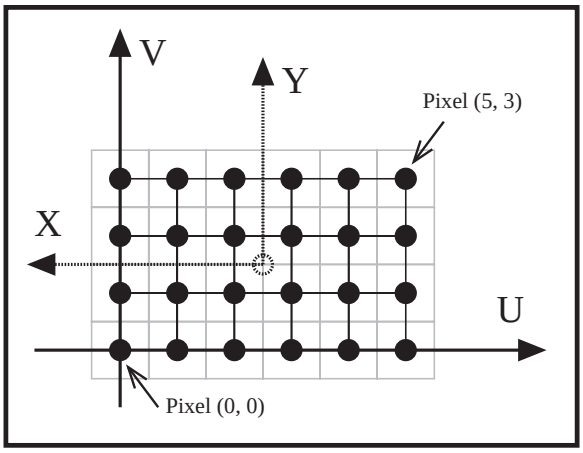


Figure 2: Illustrating pixel coordinate conventions.

and therefore two of the four intrinsic parameters are now known.

$$\begin{vmatrix} t_u \\ t_v \\ 1 \end{vmatrix} = \begin{vmatrix} c_u \\ c_v \\ 1 \end{vmatrix} \tag{14}$$

So long as the optical center and image centers, the scale factors are also easily computed. The key observation is that half the pixels span have the field of view angle, and therefore:

$$f_u = 2 \arctan\left(\frac{d_u}{2s_u}\right) \qquad f_v = 2 \arctan\left(\frac{d_v}{2s_v}\right) \tag{15}$$

Inverting this relationship yields expressions for the scale factors in terms of the dimensions and field of view.

$$s_u = \frac{\frac{d_u}{2}}{\tan\frac{f_u}{2}} \qquad s_v = \frac{\frac{d_v}{2}}{\tan\frac{f_v}{2}} \tag{16}$$

This is the clearest and simplest way to solve for the scale factors. However, it does not completely address the relationships between points in 3D and their mappings to points in the UV image plane. There is an alternative derivation which, while more cumbersome, also adds some additional insight into the relationships involved.

Consider the mapping between the corners of the image and the corresponding 3D points which project to the corners. The field of view determines the minimum and maximum X and Y values visible to the sensor at a given depth Z.

$$\begin{array}{rclrcl} X_{min} & = & -\tan\left(\frac{f_u}{2}\right) Z & Y_{min} & = & -\tan\left(\frac{f_v}{2}\right) Z \\ X_{max} & = & \tan\left(\frac{f_u}{2}\right) Z & Y_{max} & = & \tan\left(\frac{f_v}{2}\right) Z \end{array} \tag{17}$$

The choice of Z in equations 17 will not matter when projecting points to the image plane. For the sake of simplicity, points on the $Z = 1$ plane may be selected as the 3D corners.

The corners in the UV image plane may also be expressed in terms of upper and lower bounds.

$$\begin{array}{rclrcl} u_{min} & = & c_u - \frac{d_u - 1}{2} & v_{min} & = & c_v - \frac{d_v - 1}{2} \\ u_{max} & = & c_u + \frac{d_u - 1}{2} & v_{max} & = & c_v + \frac{d_v - 1}{2} \end{array} \tag{18}$$

To determine the scale factors it is sufficient to map one corner on the $Z = 1$ plane to its corresponding corner on the UV image plane. We will select the bottom left corner (u_{min}, v_{min}) yielding the constraint:

$$\begin{vmatrix} u_{min} \\ v_{min} \\ 1 \end{vmatrix} = \begin{vmatrix} s_u & 0 & t_u & 0 \\ 0 & s_v & t_v & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} \begin{vmatrix} X_{max} \\ Y_{min} \\ 1 \\ 1 \end{vmatrix} \tag{19}$$

Note that the lower bound on U is being mapped to the upper bound on X. This flips the sign of the U axis relative to the X axis. The reversal is made so that objects may be viewed forward of the camera, i.e. with positive Z values while at the same time allowing the image plane to have a natural origin at the lower left corner. This reversal of direction is illustrated in Figure 2.

Solving equation 19 for the scale factor using values from equations 17 and 18 yields

$$s_u = \frac{-(u_{min} - t_u)}{\tan\left(\frac{f_u}{2}\right)} \quad s_v = \frac{(v_{min} - t_v)}{\tan\left(\frac{f_v}{2}\right)} \tag{20}$$

These expression simplify to those already presented above when it is noted that when the optical centers and image centers are coincident

$$u_{min} - t_u = \frac{d_u}{2} \qquad v_{min} - t_u = \frac{d_v}{2} \tag{21}$$

Thus leading finally to

$$s_u = -\frac{\frac{d_u}{2}}{\tan\frac{f_u}{2}} \qquad s_v = \frac{\frac{d_v}{2}}{\tan\frac{f_v}{2}} \tag{22}$$

Equations 16 and 22 are identical up to the change in sign on s_u which reverses the U axes relative to the X axis.

4 LADAR Range Sensor Geometry

The imaging geometry for the LADAR range sensor is similar, but not identical to that of a perspective camera. The key distinction is that the LADAR has a constant sampling **angle** per pixel. This is not true of a projective sensor, in which pixels subtend increasingly large angles as they move away from the image center.

The radial mapping between a pixel (u, v) in a LADAR range image and the point in the world sampled by that pixel depends upon the angles (θ, ϕ) encoded by the pixel coordinate. There is a 2D scale and translation transformation between the angles (θ, ϕ) and pixel coordinates (u, v) which moves the image center and changes the units from radians to pixels.

$$\begin{vmatrix} u \\ v \\ 1 \end{vmatrix} = \begin{vmatrix} r_u & 0 & t_u \\ 0 & r_v & t_v \\ 0 & & 1 \end{vmatrix} \begin{vmatrix} \theta \\ \phi \\ 1 \end{vmatrix} = \begin{vmatrix} r_u\theta + t_u \\ r_v\phi + t_v \\ 1 \end{vmatrix} \tag{23}$$

The mapping from 3D Cartesian coordinates to the image plane involves both this 2D affine transformation as well as a spherical to Cartesian coordinate transformation. The angles (θ, ϕ) are the azimuth and elevation of a ray projecting out from the LADAR focal point to a point in the scene a distance D from the focal point. As such, (θ, ϕ) represent rotation about the vertical and horizontal axes respectively.

To derive the spherical to Cartesian transformations it helps to view (θ, ϕ) as representing rotation about the vertical and horizontal axes respectively. Consider a new measurement specific coordinate system L' in which the Cartesian coordinate of the point being viewed is defined as the point D units out the Z axis.

$$P_{L'} = \begin{vmatrix} 0 \\ 0 \\ D \end{vmatrix} \tag{24}$$

The coordinate transformation we seek maps L' into the canonical LADAR system L. To accomplish this transformation, a point $P_{L'}$ must first be rotated about the X axis by an amount ϕ representing the elevation of the point above the horizontal. Next the point must be rotated about the Y axis by an amount θ representing the panning of the sensor in the XY plane. Composing these two rotation matrices in the proper order defines the relationship between the two coordinate systems.

$$\begin{vmatrix} X \\ Y \\ Z \end{vmatrix} = \begin{vmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{vmatrix} \begin{vmatrix} 0 \\ 0 \\ D \end{vmatrix} = \begin{vmatrix} \cos\phi\sin\theta D \\ \sin\phi D \\ \cos\phi\cos\theta D \end{vmatrix} \tag{25}$$

The inverse mapping from Cartesian coordinates X, Y, Z to pixel coordinate θ, ϕ and depth D is expressed by the following equations:

$$D = \sqrt{X^2 + Y^2 + Z^2} \tag{26}$$

$$\theta = \arctan\frac{X}{Z} \tag{27}$$

$$\phi = \arcsin\frac{Y}{D} \tag{28}$$

Combining these equations with the angular scale factors in equation 23 yields the expression for the UV LADAR coordinates of a 3D point.

$$u = r_u \arctan\left(\frac{X}{Z}\right) + t_u \tag{29}$$

$$v = r_v \arcsin\left(\frac{Y}{\sqrt{X^2 + Y^2 + Z^2}}\right) + t_v \tag{30}$$

4.1 Parameters for the Fort Carson LADAR

The LADAR used to collect the Fort Carson data was calibrated before the data collection by collecting imagery of surveyed features and measuring the angular resolution. This yields measured r values:

$$r_u = -438.6\left(\frac{\text{Pixels}}{\text{radian}}\right) \quad r_v = 383.1\left(\frac{\text{Pixels}}{\text{radian}}\right) \tag{31}$$

As already reported, the LADAR images contain 120x24 pixels. The field of view values reported in Table 2 are derived from the images size and scale factors r_u and r_v as follows:

$$f_u = \frac{120 - 1}{438.6} \tag{32}$$
$$= 0.271 \quad \text{Radians} \quad (15.5 \quad \text{Degrees})$$

$$f_v = \frac{24 - 1}{383.1} \tag{33}$$
$$= 0.060 \quad \text{Radians} \quad (3.4 \quad \text{Degrees})$$

In the absence of precise calibration it is assumed that the image center for the LADAR is just the center of the bounded image plane and hence the translation terms t_u and t_v using the pixel center method of Figure 2, are as follows:

$$t_u = 59.5 \quad \text{Pixels} \tag{34}$$

$$t_v = 11.5 \quad \text{Pixels} \tag{35}$$

4.2 Spherical Versus Perspective Mappings and Image Displacement

While the Spherical Mapping defined in equation 25 for the range sensor is a better model of what actually takes place in LADAR, it is fair to ask whether over typical fields of view and sampling resolutions the difference is significant relative to a perspective mapping.

One way to answer this question is to plot the Euclidean offset between the UV mapping from spherical projection versus the UV mapping from perspective projection for the same 3D points over the sensor field view. Using the perspective matrix from equation 1, the mapping for perspective projection is

$$\begin{array}{rclcrcl} u_p & = & s_u\left(\frac{X}{Z}\right) + t_u & & v_p & = & s_v\left(\frac{Y}{Z}\right) + t_v \\ & = & -435.9\left(\frac{X}{Z}\right) + 59.5 & & & = & 383.0\left(\frac{Y}{Z}\right) + 11.5 \end{array} \tag{36}$$

Substituting the intrinsic spherical parameters in equations 31, 34 and 35 into equations 29 and 30 yields the following spherical mapping from 3D points to the UV LADAR image plane.

$$\begin{array}{rclcrcl} u_s & = & r_u \arctan\left(\frac{X}{Z}\right) + t_u & & v_s & = & r_v \arcsin\left(\frac{Y}{\sqrt{X^2+Y^2+Z^2}}\right) + t_v \\ & = & -438.6\left(\frac{X}{Z}\right) + 59.5 & & & = & 383.1\left(\frac{Y}{\sqrt{X^2+Y^2+Z^2}}\right) + 11.5 \end{array} \tag{37}$$

It is now possible to ask how these two mappings differ over the sensor field of view. For points at $Z = 100$, the visible points have X values ranging from -13.65 to 13.65 and Y values ranging from -3.0 to 3.0. By plotting the Euclidean distance between the two image mappings for corresponding 3D points one can see the extent to which the two mapping differ over different portions of the image. The Euclidean difference Δ may be written as:

$$\Delta(X,Y) = \sqrt{(u_p - u_s)^2 + (v_p - v_s)^2} \tag{38}$$

This difference measured over all visible points is shown as a surface plot in Figure 3. Observe the difference never exceeds 0.15 over the entire image. Hence, while the spherical mapping is more correct given the sensor design, the difference between the two with respect to relative pixel mappings in the image planes never varies more than by 15% of the width of a single pixel.

4.3 Spherical Versus Perspective Mappings and Range Displacement

The previous section showed that relative to the displacement between pixel coordinates, the difference between spherical and perspective projection is not significant for the specific case of the Fort Carson LADAR.

This section takes up a different question: what if range data is back-projected into the scene assuming perspective when it has in fact been collected using spherical projection. To test this case, the same field of planar points at $Z = 100$ will be considered. This time, these will be imaged using the spherical projection, and then back-projected into the scene using perspective.

The mapping for points in the world to pixels using spherical projection was already expressed in equation 37. Let us introduce the convention of writing down a range pixel l as a coordinate-value pair:

$$l = \left(\begin{vmatrix} u \\ v \\ 1 \end{vmatrix}, \; D \right) = \left(\begin{vmatrix} r_u \arctan\left(\frac{X}{Z}\right) + t_u \\ r_v \arcsin\left(\frac{Y}{\sqrt{X^2+Y^2+Z^2}}\right) + t_v \\ 1 \end{vmatrix}, \; \sqrt{X^2+Y^2+Z^2} \right) \tag{39}$$

where u and v are the image coordinates of a point in 3D and D is the recorded depth to that point.

The projected point can then be back-projected into the scene using the perspective rather than spherical mapping. Figure 4 shows the error introduced by mixing the projection schemes. The graph has been generated for pixels at 100m. Note the error is quite small relative to the large change in Z.

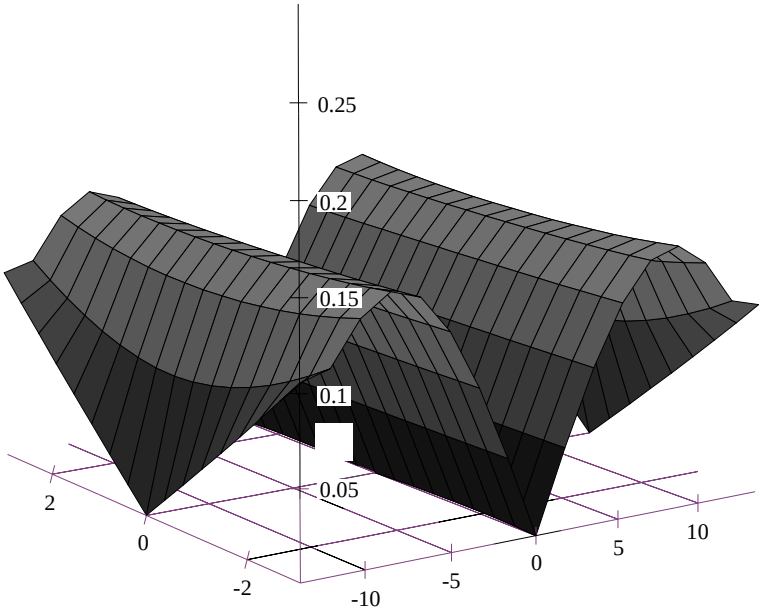


Figure 3: Pixel Mapping Δ over viewable points (X, Y). Δ is Euclidean distance between spherical and perspective projections. The difference is least alone the line $Y = 0$ and never exceeds 0.15.

5 Deviations from Perfect Boresight Alignment

Consider transformations between three distinct 3D coordinate reference frames: a world reference $\mathcal{W}$ and two sensor reference frames $\mathcal{A}$ and $\mathcal{B}$. Define the 3D transformation from frame $\mathcal{W}$ to frame $\mathcal{A}$ as:

$$M_{\mathcal{WA}} = S_{\mathcal{WA}} R_{\mathcal{WA}} T_{\mathcal{WA}} \tag{40}$$

where $S_{\mathcal{WA}}$ is a scale transformation, $R_{\mathcal{WA}}$ is a rotation and $T_{\mathcal{WA}}$ is a translation. Hence, the standard mapping of points from one frame of reference to another is accomplished by pre-multiplying the point by the transformation.

$$A = M_{\mathcal{WA}} W \tag{41}$$

where W is a point in reference frame $\mathcal{W}$ and A is the same point expressed in the range sensor reference frame R. An analogous transformation $M_{\mathcal{WB}}$ maps points from the world to reference frame $\mathcal{B}$.

These transformations are often described as the extrinsic sensor parameters. In the case where $M_{\mathcal{WA}} = M_{\mathcal{WB}}$, the sensors have identical references frames and pixel mappings will differ only as a function of the sensor parameters. Equality of extrinsic parameters between two sensors may be thought of as the condition

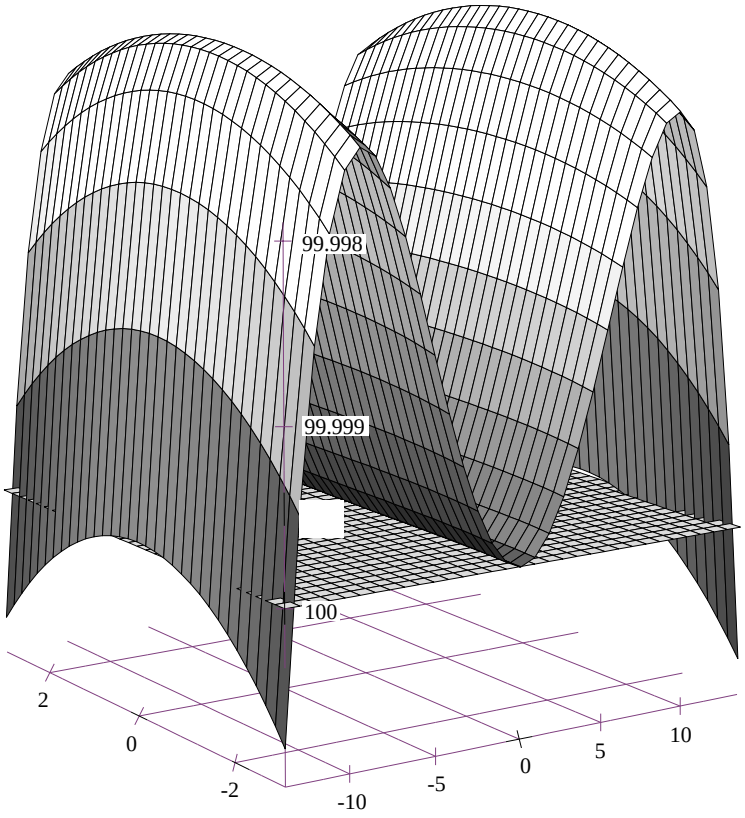


Figure 4: Reconstructed Z' for viewable points $(X, Y, 100)$ when Perspective and Spherical Projection are mixed. Here, the X-axis (on the right) and Y-axis (on the left) are in meters at the target. The Z-axis is Z' in meters, with the plane at 100 meters representing the true depth, Z. Note characteristic shape similar to pixel displacement, but tiny absolute deviation from true 100 meters. Greatest deviation is roughly 2 millimeters in Z over 25 meters in X.

which arises when two sensors are perfectly bore-sight aligned. In other words, they have coincident focal points and coincident optical axes.

How minor deviations from perfect bore-sight alignment alter the relative mapping between sensor pixels is of key interest when doing sensor fusion. The remainder of this section will consider how 3D points map to the respective pixel coordinates of sensors $\mathcal{A}$ and $\mathcal{B}$

5.1 The Base-Case: Perfect Bore-Sight Alignment

In the case of perfect bore-sight alignment, the transformations $M_{\mathcal{WA}} = M_{\mathcal{WB}}$ are equal and hence may be neglected. Therefore, the projection matrixes defined above operating on the same 3D point allow us to derive points A and B in the respective image coordinates of the two sensors $\mathcal{A}$ and $\mathcal{B}$.

$$A' = \left| \begin{array}{ccc||c} s_{\alpha u} & 0 & t_{\alpha u} & X \\ 0 & s_{\alpha v} & t_{\alpha v} & Y \\ 0 & 0 & 1 & Z \end{array} \right| \qquad B' = \left| \begin{array}{ccc||c} s_{\beta u} & 0 & t_{\beta u} & X \\ 0 & s_{\beta v} & t_{\beta v} & Y \\ 0 & 0 & 1 & Z \end{array} \right| \tag{42}$$

The prime notation A' emphasizes that this linear algebraic expression produces a point equivalent up to a scale factor and that the true u_α and v_α coordinates are indicated in the normalized form A where the third element is 1.

Carrying out the indicated linear multiplication yields:

$$A' = \begin{vmatrix} s_{\alpha u}X + t_{\alpha u}Z \\ s_{\alpha v}Y + t_{\alpha v}Z \\ Z \end{vmatrix} \qquad B' = \begin{vmatrix} s_{\beta u}X + t_{\beta u}Z \\ s_{\beta v}Y + t_{\beta v}Z \\ Z \end{vmatrix} \tag{43}$$

and after normalization, the following are the coordinates of a point $(X,\ Y,\ Z)$ in the UV_α and the UV_β planes.

$$\begin{aligned} u_\alpha &= s_{\alpha u}\left(\tfrac{X}{Z}\right) + t_{\alpha u} & u_\beta &= s_{\beta u}\left(\tfrac{X}{Z}\right) + t_{\beta u} \\ v_\alpha &= s_{\alpha v}\left(\tfrac{Y}{Z}\right) + t_{\alpha v} & v_\beta &= s_{\beta v}\left(\tfrac{Y}{Z}\right) + t_{\beta v} \end{aligned} \tag{44}$$

Our goal is a linear mapping from UV_α to UV_β. This can be readily computed by equating the common ratio terms $\frac{X}{Z}$ and $\frac{Y}{Z}$ in the above equation and solving for the UV_β coordinates in terms of the UV_α coordinates.

$$\begin{vmatrix} u_\beta \\ v_\beta \\ 1 \end{vmatrix} = \begin{vmatrix} \frac{s_{\beta u}}{s_{\alpha u}} & 0 & 0 \\ 0 & \frac{s_{\beta v}}{s_{\alpha v}} & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} u_\alpha \\ v_\alpha \\ 1 \end{vmatrix} + \begin{vmatrix} t_{\beta u} - \frac{s_{\beta u}}{s_{\alpha u}} t_{\alpha u} \\ t_{\beta v} - \frac{s_{\beta v}}{s_{\alpha v}} t_{\alpha v} \\ 1 \end{vmatrix} \tag{45}$$

5.2 Rotation About the Optical Axis

Consider now the above case with minor modification: sensor $\mathcal{B}$ is rotated about the Z, the optical axis, by an amount ϕ. The transformation and projection equation are:

$$A = \begin{vmatrix} s_{\alpha u} & 0 & t_{\alpha u} \\ 0 & s_{\alpha v} & t_{\alpha v} \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} \qquad B = \begin{vmatrix} s_{\beta u} & 0 & t_{\beta u} \\ 0 & s_{\beta v} & t_{\beta v} \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} \tag{46}$$

In a manner similar to that shown above, one can equate the ratios $\frac{X}{Z}$ and $\frac{Y}{Z}$ and solve for a linear mapping from UV_α to UV_β.

$$\begin{vmatrix} u_\beta \\ v_\beta \\ 1 \end{vmatrix} = \begin{vmatrix} \frac{s_{\beta u}}{s_{\alpha u}}\cos\phi & \frac{s_{\beta u}}{s_{\alpha v}}\sin\phi & 0 \\ -\frac{s_{\beta v}}{s_{\alpha u}}\sin\phi & \frac{s_{\beta v}}{s_{\alpha v}}\cos\phi & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} u_\alpha \\ v_\alpha \\ 1 \end{vmatrix} - \begin{vmatrix} \frac{s_{\beta u}}{s_{\alpha u}}\cos\phi & \frac{s_{\beta u}}{s_{\alpha v}}\sin\phi & 0 \\ -\frac{s_{\beta v}}{s_{\alpha u}}\sin\phi & \frac{s_{\beta v}}{s_{\alpha v}}\cos\phi & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} t_{\alpha u} \\ t_{\alpha v} \\ 1 \end{vmatrix} + \begin{vmatrix} t_{\beta u} \\ t_{\beta v} \\ 1 \end{vmatrix} \tag{47}$$

In the special case that the horizontal and vertical scale factors for sensor $\mathcal{A}$ are equal, as are the scale factors for sensor $\mathcal{B}$, then and only then does this reduce to a 2D rotation plus a translation. To see this, let

$$\begin{aligned} s_\alpha &= s_{\alpha u} = s_{\alpha v} \\ s_\beta &= s_{\beta u} = s_{\beta v} \end{aligned} \quad \text{and} \quad k = \frac{s_\beta}{s_\alpha} \tag{48}$$

and note that equation 47 simplifies to

$$\begin{vmatrix} u_\beta \\ v_\beta \\ 1 \end{vmatrix} = k \begin{vmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} u_\alpha \\ v_\alpha \\ 1 \end{vmatrix} + \begin{vmatrix} t_{\beta u} - \frac{s_{\beta u}}{s_{\alpha u}}\cos\phi t_{\alpha u} - \frac{s_{\beta u}}{s_{\alpha v}}\sin\phi t_{\alpha v} \\ t_{\beta v} + \frac{s_{\beta v}}{s_{\alpha u}}\sin\phi t_{\alpha u} - \frac{s_{\beta v}}{s_{\alpha v}}\cos\phi t_{\alpha v} \\ 1 \end{vmatrix} \tag{49}$$

When the scale factors are unequal, then the mapping between UV_α and UV_β is a general 2D affine transformation.

$$B = M\ A \tag{50}$$

where

$$M = \begin{vmatrix} \left(\frac{s_{\beta u}}{s_{\alpha u}} \cos\phi\right) & \left(\frac{s_{\beta u}}{s_{\alpha v}} \sin\phi\right) & \left(t_{\beta u} - \frac{s_{\beta u}}{s_{\alpha u}} \cos\phi t_{\alpha u} - \frac{s_{\beta u}}{s_{\alpha v}} \sin\phi t_{\alpha v}\right) \\ \left(-\frac{s_{\beta v}}{s_{\alpha u}} \sin\phi\right) & \left(\frac{s_{\beta v}}{s_{\alpha v}} \cos\phi\right) & \left(t_{\beta v} + \frac{s_{\beta v}}{s_{\alpha u}} \sin\phi t_{\alpha u} - \frac{s_{\beta v}}{s_{\alpha v}} \cos\phi t_{\alpha v}\right) \\ 0 & 0 & 1 \end{vmatrix} \tag{51}$$

The basic conclusion to be drawn from this section is that there is still an exact 2D affine mapping between UV_α and UV_β and subject to equal horizontal and vertical scaling the rotation angle maps directly into 2D. However, in the more general case, the full 6 degrees of freedom associated with the 2D affine transform are needed to represent warping induced by rotation about unequally sampled axes.

5.3 Translation Along the Z axis

Consider again a perfectly bore-sight aligned pair of sensors and now ask what happens if sensor $\mathcal{B}$ is translates ahead or behind sensor $\mathcal{A}$ along the common optical axis. Under these conditions, the projection equations for a common point are:

$$A' = \left|\begin{array}{cccc||c} s_{\alpha u} & 0 & t_{\alpha u} & 0 & X \\ 0 & s_{\alpha v} & t_{\alpha v} & 0 & Y \\ 0 & 0 & 1 & 0 & Z \\ 0 & 0 & 0 & 1 & 1 \end{array}\right| \qquad B' = \left|\begin{array}{cccc||cccc||c} s_{\beta u} & 0 & t_{\beta u} & 0 & 1 & 0 & 0 & 0 & X \\ 0 & s_{\beta v} & t_{\beta v} & 0 & 0 & 1 & 0 & 0 & Y \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & T_z & Z \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array}\right| \tag{52}$$

Clearly the coordinates on the UV_α plane are the same a derived in equation 44. The coordinates of the same point in the UV_β plane are

$$u_\beta = \frac{s_{\beta u} X + t_{\beta u} Z + t_{\beta u} T_z}{T_z + Z} \tag{53}$$

$$v_\beta = \frac{s_{\beta v} Y + t_{\beta v} Z + t_{\beta v} T_z}{T_z + Z} \tag{54}$$

The denominator $T_z + Z$ complicates the task of comparing coordinate systems. However, X and Y can be expressed in terms of the other variables for each transformation and then these common expressions themselves may be equated. Thus, coordinates in UV_β may be expressed in terms of coordinates in UV_α so long as the depth values Z is assumed to be known.

$$B = M\ A \tag{55}$$

where

$$M = \begin{vmatrix} \left(\frac{s_{\beta u}}{(T_z+Z)\, s_{\alpha u}}\right) & 0 & \left(t_{\beta u} - \frac{s_{\beta u} t_{\alpha u} Z}{(T_z+Z)\, s_{\alpha u}}\right) & 0 \\ 0 & \left(\frac{s_{\beta v}}{(T_z+Z)\, s_{\alpha v}}\right) & \left(t_{\beta v} - \frac{s_{\beta v} t_{\alpha v} Z}{(T_z+Z)\, s_{\alpha v}}\right) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{56}$$

The depth of a point Z changes the scaling applied to the point in mapping between UV_α and UV_β. Under the highly restricted case of viewing points all lying in a plane of constant Z, the scaling is constant for all points and the matrix M reduces to a simple 2D affine transformation. However, in general no single 2D affine transformation can capture the UV_α to UV_β mapping if sensor $\mathcal{B}$ is translated ahead of or behind sensor $\mathcal{A}$.

5.4 Translation in a Common Image Plane

Again, starting with the perfectly bore-sight aligned configuration, consider what happens when sensor $\mathcal{B}$ translates in a common XY image plane. The projection equations for this case are:

$$A' = \left| \begin{array}{cccc||c} s_{\alpha u} & 0 & t_{\alpha u} & 0 & X \\ 0 & s_{\alpha v} & t_{\alpha v} & 0 & Y \\ 0 & 0 & 1 & 0 & Z \\ 0 & 0 & 0 & 1 & 1 \end{array} \right| \qquad B' = \left| \begin{array}{cccc||cccc||c} s_{\beta u} & 0 & t_{\beta u} & 0 & 1 & 0 & 0 & T_x & X \\ 0 & s_{\beta v} & t_{\beta v} & 0 & 0 & 1 & 0 & T_y & Y \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & Z \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array} \right| \tag{57}$$

The coordinates of the projection of the point $(X,\ Y\ , Z)$ in the UV_β plane are

$$u_\beta = \frac{s_{\beta u}\,(X + T_x)}{Z} + t_{\beta u} \tag{58}$$

$$v_\beta = \frac{s_{\beta v}\,(Y + T_y)}{Z} + t_{\beta u} \tag{59}$$

Taking the same approach as in the previous section, the coordinates of a common point projected to the UV_β plane expressed in terms of its coordinates in the UV_α plane can be shown to depend upon the depth of the point Z.

$$B = M\ A \tag{60}$$

where

$$M = \left| \begin{array}{ccc} \frac{s_{\beta u}}{s_{\alpha u}} & 0 & \left(\frac{s_{\beta u} T_x}{Z} - \frac{s_{\beta u} t_{\alpha u}}{s_{\alpha u}} + t_{\beta u} \right) \\ 0 & \frac{s_{\beta v}}{s_{\alpha v}} & \left(\frac{s_{\beta v} T_y}{Z} - \frac{s_{\beta v} t_{\alpha v}}{s_{\alpha v}} + t_{\beta v} \right) \end{array} \right| \tag{61}$$

Unlike the translation in Z case, the dependency on depth in this case modifies the translation of one plane relative to the other. In simple and intuitive terms, this means that for points all at a common depth Z there is a 2D affine mapping between UV_α and UV_β.

In general, the 2D scale change between systems is identical to that for perfect bore-sight aligned sensors. The relative 2D translation is similar to the bore-sighted case but with one additional term which depends upon the depth of the points Z and the planar translation T_x, T_y between the two sensors. As should come as no surprise, the apparent 2D translation between UV_α and UV_β gets larger for points near the sensors and less for points far from the sensor.

5.5 Rotation About the Horizontal and Vertical Axes

The case of rotation about the X and Y axes is considerably more complicated than those previously considered. It is no longer practical to solve a direct mapping between the UV_α and UV_β image planes. The mapping is no longer well expressed as a 2D affine transformation, even allowing for simple parameterization in say the depth value Z. The relationship between image spaces is coupled through the angles of rotation and dependent upon all three point coordinates: $(X,\ Y,\ Z)$.

It is relatively simple, though, to express the mapping from points in the world to the UV_β image plane.

$$B' = \left| \begin{array}{cccc||cccc||cccc||c} s_{\beta u} & 0 & t_{\beta u} & 0 & \cos\phi_y & 0 & \sin\phi_y & 0 & 1 & 0 & 0 & 0 & X \\ 0 & s_{\beta v} & t_{\beta v} & 0 & 0 & 1 & 0 & 0 & 0 & \cos\phi_x & \sin\phi_x & 0 & Y \\ 0 & 0 & 1 & 0 & -\sin\phi_y & 0 & \cos\phi_y & 0 & 0 & -\sin\phi_x & \cos\phi_x & 0 & Z \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array} \right| \tag{62}$$

The coordinates of the projection of the point $(X,\ Y,\ Z)$ may now be written as

$$u_\beta = \frac{s_{\beta u}\,(\Phi_y \cdot V_{xzy})}{\Phi_y \cdot V_{zyx}} + t_{\beta u} \tag{63}$$

$$v_\beta = \frac{s_{\beta v}\,(\Phi_x \cdot V_{yz})}{\Phi_y \cdot V_{zyx}} + t_{\beta v} \tag{64}$$

where

$$\Phi_x = \left| \begin{array}{c} \cos\phi_x \\ \sin\phi_x \end{array} \right| \tag{65}$$

$$\Phi_y = \left| \begin{array}{c} \cos\phi_y \\ \sin\phi_y \end{array} \right| \tag{66}$$

$$V_{yz} = \left| \begin{array}{c} Y \\ Z \end{array} \right| \tag{67}$$

$$V_{zy} = \left| \begin{array}{c} Z \\ -Y \end{array} \right| \tag{68}$$

$$V_{xzy} = \left| \begin{array}{c} X \\ \Phi_x \cdot V_{zy} \end{array} \right| \tag{69}$$

$$V_{zyz} = \left| \begin{array}{c} \Phi_x \cdot V_{zy} \\ -X \end{array} \right| \tag{70}$$

6 Approximating Small Pan/Tilt Errors with Translation

Much of the previous development has been working up to answer the following question. Given two sensors, one range and one optical, which are nearly bore-sight aligned, how much error is introduced if small rotations about the X and Y axes are modeled as translation in a common image plane?

To address this question, first there is the matter of whether to model the range sensor as a spherical or a perspective projection sensor. While we know spherical is more consistent with the actual construction of the sensor, Section 4 demonstrated that little error is introduced for a sensor with parameters such as were used in the Fort Carson data collection. The greatest error in relative pixel mappings over the sensor field of view was 0.15 pixels. Therefore, because it will make our task slightly simpler, both the range and optical sensors will be assumed to be projective.

The equations relating a general point $(X,\ Y,\ Z)$ to the image plane of a sensor assuming translation versus rotation have already been presented in equations 58, 59, 63 and 64. Here let us restate these equation with the following change. Let sensor $\mathcal{A}$ rotate about X and Y and let sensor $\mathcal{B}$ translate in the common XY plane.

$$u_\alpha = \frac{s_{\alpha u}\left(\cos\phi_y X + (\cos\phi_x Z - \sin\phi_x Y)\sin\phi_y\right)}{(\cos\phi_x Z - \sin\phi_x Y)\cos\phi_y - \sin\phi_y X} + t_{\alpha u} \tag{71}$$

$$v_\alpha = \frac{s_{\alpha v}\left(\cos\phi_x Y + \sin\phi_x Z\right)}{(\cos\phi_x Z - \sin\phi_x Y)\cos\phi_y - \sin\phi_y X} + t_{\alpha u} \tag{72}$$

$$u_\beta = \frac{s_{\beta u}\left(X + T_x\right)}{Z} + t_{\beta u} \tag{73}$$

$$v_\beta = \frac{s_{\beta v}\left(Y + T_y\right)}{Z} + t_{\beta u} \tag{74}$$

There are two steps involved in analyzing the degree to which the translation mapping can stand in for the rotation mapping. The first addresses the question of sensor parameters. The comparison is most direct if identical sensor parameters are assumed. For the comparison, we will use the following intrinsic parameters

$$\begin{array}{lllllll} s_{\alpha u} & = & s_u & & t_{\alpha u} & = & 0 \\ s_{\alpha v} & = & s_v & & t_{\alpha v} & = & 0 \\ s_{\beta u} & = & s_u & & t_{\beta u} & = & 0 \\ s_{\beta v} & = & s_v & & t_{\beta v} & = & 0 \end{array} \tag{75}$$

the image center translation terms, once equal, will drop out of any comparison, and therefore might as well be set to zero.

The next matter is how to determine what degree of translation $T_x,\ T_y$ to introduce in order to approximate a rotation $\phi_x,\ \phi_y$. The goal shall be to track a common point at depth D. More precisely, consider the point $(0,\ 0,\ D)$ in the coordinate reference frame for sensor $\mathcal{A}$. Determine a translation for sensor $\mathcal{B}$ such that this point projects to the center of the UV_β image plane.

To accomplish this transformation, observe that the coordinates in the world reference frame $\mathcal{W}$ of the point to be tracked P' may be expressed in terms of the rotation of sensor $\mathcal{A}$.

$$\begin{vmatrix} 0 \\ 0 \\ D \\ 1 \end{vmatrix} = \begin{vmatrix} \cos\phi_y & 0 & \sin\phi_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi_y & 0 & \cos\phi_y & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi_x & \sin\phi_x & 0 \\ 0 & -\sin\phi_x & \cos\phi_x & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} X' \\ Y' \\ Z' \\ 1 \end{vmatrix} \tag{76}$$

The world coordinates of the point P' are therefore

$$\begin{vmatrix} X' \\ Y' \\ Z' \\ 1 \end{vmatrix} = \begin{vmatrix} -\sin\phi_y D \\ -\sin\phi_x \cos\phi_y D \\ \cos\phi_x \cos\phi_y D \\ 1 \end{vmatrix} \tag{77}$$

Setting equations 73 and 74 equal to zero for the point to be tracked provides an expression for the translations necessary to keep the point at depth D in the center of view of the translating as well as the rotating sensor.

$$T_x \ = \ \sin\phi_y D \qquad\qquad T_y \ = \ \cos\phi_y \sin\phi_x D \tag{78}$$

Substituting these back into equations 73 and 74 yields a general mapping from world reference frame $\mathcal{W}$ to the UV_β image plane subject to the constraint that the translating sensor track the point at depth D viewed by the rotating sensor $\mathcal{A}$.

$$\tag{79}$$

$$u_\beta \ = \ \frac{s_u\left(X + \sin\phi_y D\right)}{Z} \tag{80}$$

$$v_\beta \ = \ \frac{s_v\left(Y + \cos\phi_y \sin\phi_x D\right)}{Z} \tag{81}$$

Recall also the simplifications from equation 75 have been employed.

It is now possible to define a simple expression which is a function of a points placement in the world, $(X,\ Y,\ Z)$ and sensor $\mathcal{A}$'s rotation (ϕ_X, ϕ_y) which represents the Euclidean distance between projections of equivalent 3D points on the two UV_α and UV_β image planes.

$$\Delta \ = \ \sqrt{\left(u_\alpha - u_\beta\right)^2 + \left(u_\alpha - u_\beta\right)^2} \tag{82}$$

$$\tag{83}$$

Using equation 82 it is possible to examine a variety of scenarios and determine how much pixel error between corresponding 3D points is introduced when XY sensor translation is used to approximate XY sensor rotation.

6.1 Scenario 1: Points at Constant Dept

Consider a plane of points at depth $Z = 100$ meters. The tracking point will therefore be assumed to lie at depth $D = 100$ as well. Set the azimuth ϕ_y and the elevation to ϕ_x specific values and plot Δ as a function of independent variables X and Y. The plots in Figures 5a and 5b shows this plot for the color sensor parameters: row 2 of Table 2. The difference between the two is Figure 5a uses smaller rotations than

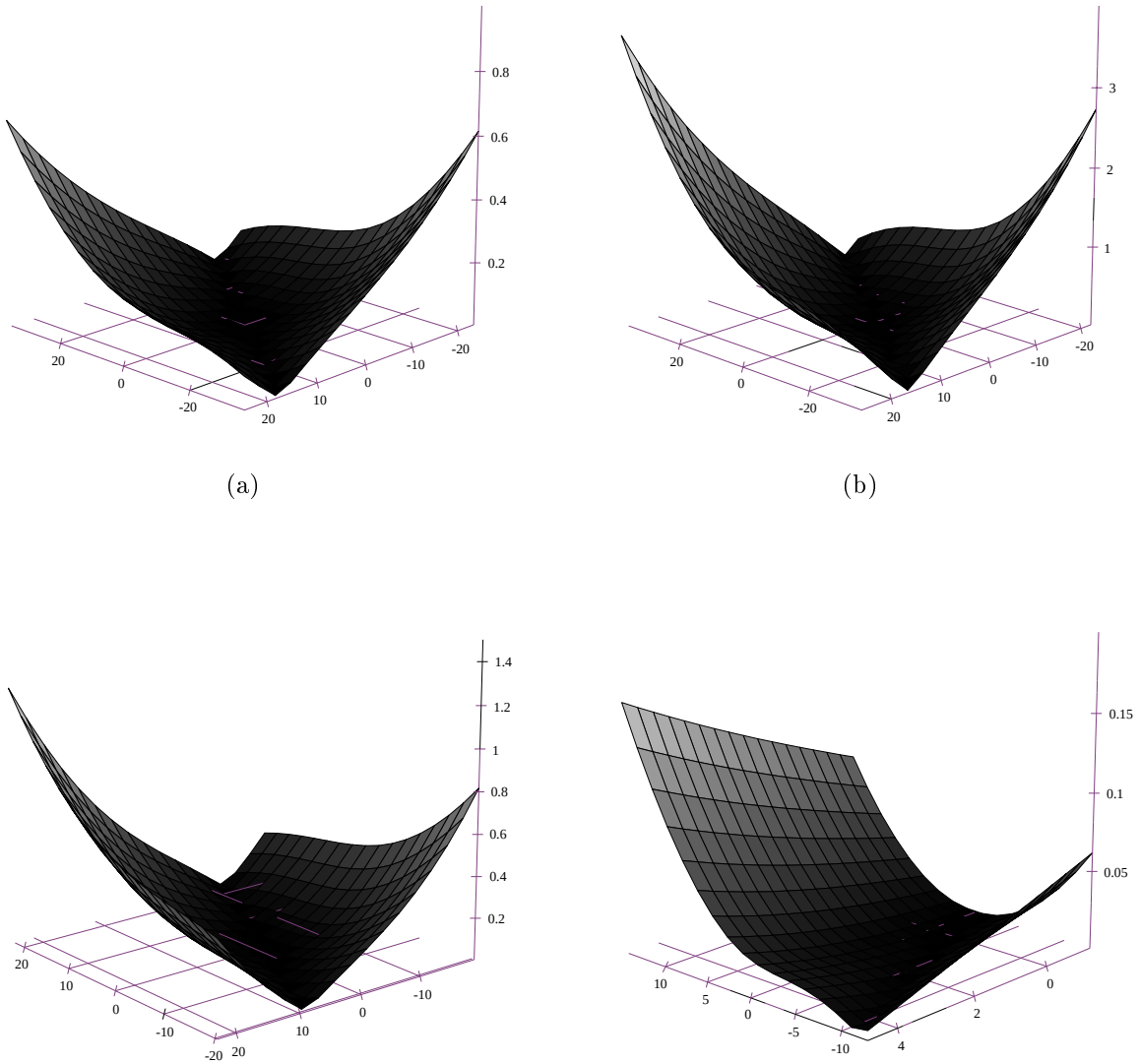


Figure 5: Pixel Coordinate Differences Δ over a plane $Z = 100$. a) color sensor and rotation $(\phi_y,\ \phi_x) = (1/5,\ 1/10)$ degrees, b) color sensor and $(1,\ 1/2)$ degrees, c) FLIR sensor and $(1,\ 1/2)$ degrees, d) LADAR sensor and $(1,\ 1/2)$. Vertical axis Δ, left bottom axis X and right bottom axis Y.

Figure 5b. Figure 5c and 5d show Δ for the FLIR (row 4 of Table 2) and range sensor. The reduced fields of view and pixel resolution lead to much smaller absolute Δ values for these sensors.

To really understand the plots in Figure 5 it is important to understand the choice of bounds for X and Y both in this and other examples. The idea is to select bounds which match that portion of 3D space visible to the sensor. To accomplish this, the bounds must depend upon two things. First they depend upon the width W and height H of the area in view at depth Z. They also depend upon the translation used to track the center point being viewed by both sensors: $(T_x,\ T_y)$.

$$\begin{aligned} W &= 2\tan\frac{f_u}{2}Z \\ H &= 2\tan\frac{f_v}{2}Z \end{aligned}$$

Hence the bounds on X and Y are:

$$\begin{array}{rclcrcl} X_{max} & = & T_x + W/2 & & X_{min} & = & T_x - W/2 \\ Y_{max} & = & T_y + H/2 & & Y_{min} & = & T_y - H/2 \end{array} \tag{84}$$

6.2 Scenario 2: Limited Deviation from Target Tracking Depth

The next question to consider using Δ from equation 82 is how the rotation versus translation approximation holds up over a limited range of depth values centered about the tracking depth. To give this some practical motivation, assume an object 2 meters wide is viewed by sensors $\mathcal{A}$ and $\mathcal{B}$ with $\mathcal{A}$ rotated about the vertical axis by an amount $\phi_y = 1/5$ degree.

Figure 7 shows Δ as a dependent variable of X and Z. For the intrinsic camera parameters, the values for the color sensor in Table 2), row 2, are used. The other variables in equation 82 are constrained as follows.

$$Z \in [99, 101] \tag{85}$$
$$X \in [X_{min}, X_{max}] \tag{86}$$
$$D = 100 \tag{87}$$
$$\phi_x = 0.0 \tag{88}$$
$$\phi_y = 1/5(\text{degrees} \tag{89}$$
$$Y = \kappa Y_{max} + (1-\kappa)\, Y_{min} \tag{90}$$

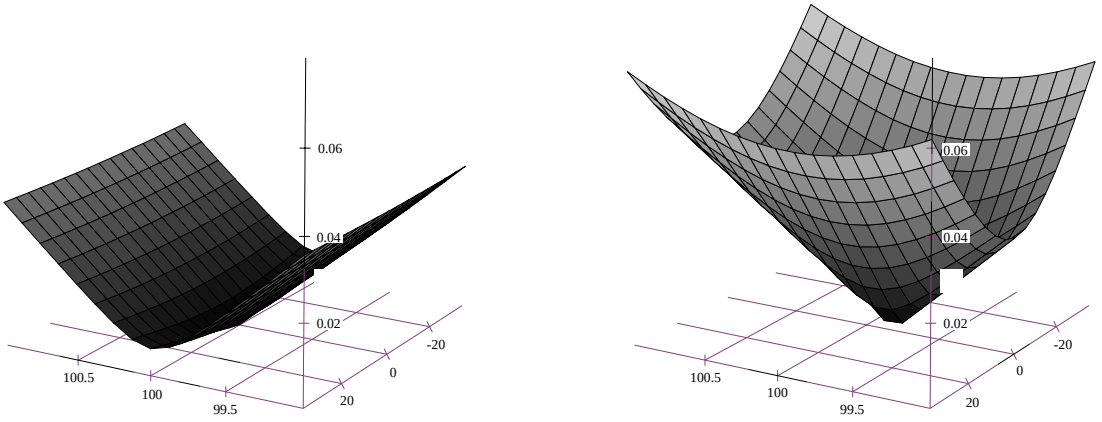
The term κ allows movement of the point being viewed vertically with $\kappa = 0.5$ specifying the center of the image.

6.3 Scenario 3: Wide Variation in Depth

It is clear that translation very well approximates rotation for a depth field about the tracking depth D. Another question is just how significantly does the approximation of Z values differ from D.

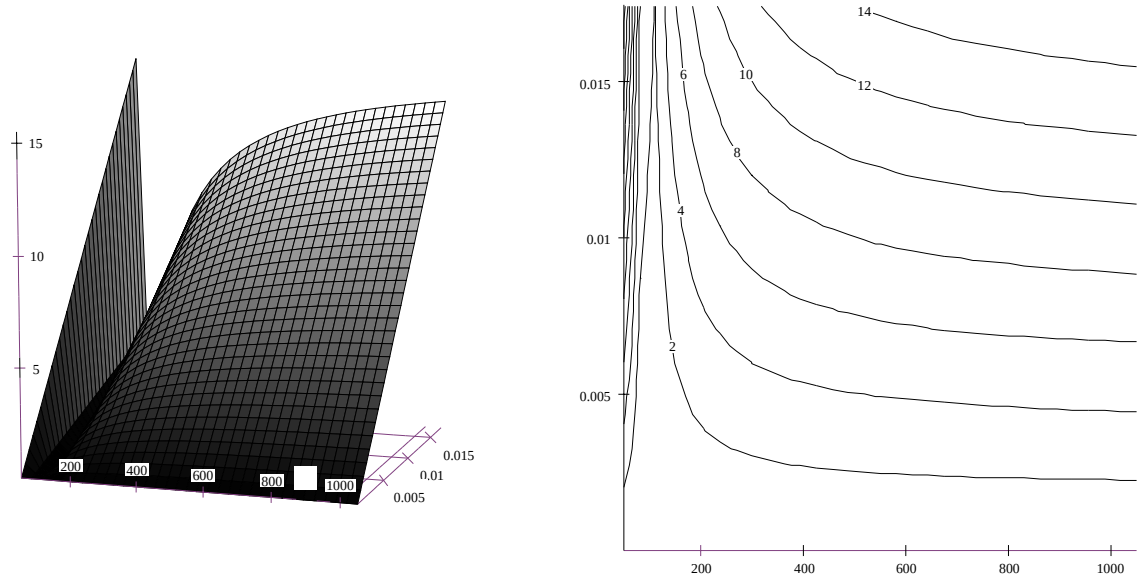
Figure 7 shows Δ as a dependent function of rotation angle ϕ_y and point depth Z. For the intrinsic camera parameters, the values for the color sensor in Table 2), row 2, are again used. The other variables in equation 82 are constrained as follows.

$$Z \in [50, 1050] \tag{91}$$
$$\phi_x \in [0.0, 3.0] \text{ degrees} \tag{92}$$
$$X \in \frac{X_{max} + X_{min}}{2} \tag{93}$$
$$Y \in \frac{Y_{max} + Y_{min}}{2} \tag{94}$$
$$D = 100 \tag{95}$$



(a) (b)

Figure 6: Pixel Differences Δ over small variations in Depth. a) Viewing the image center ($\kappa = 0.5$), b) Viewing a point 60% up the image ($\kappa = 0.6$). Vertical axis Δ, left bottom axis Z and the right bottom axis X.



(a) (b)

Figure 7: Pixel Differences Δ over large variations in Depth for a center pixel as a function of Z and ϕ_y. a) surface plot with Δ vertical, Z facing and ϕ_y back into page. b) contour plot of same data with ϕ_y vertical and Z horizontal.

Thus, Figure 7 show essentially the deviation between the center pixel for sensor $\mathcal{B}$ relative to sensor $\mathcal{A}$ as true depth to the point varies. Figure 7a show presents the data as a surface plot and Figure 7b as a contour plot.

While the surface plot is more suggestive of shape, the contour plot may be thought of as delineating pairs of values $(Z,\ \phi_y)$ such that the maximum pixel error Δ does not exceed a threshold.

Figure 7b shows the accuracy modelling small errors in rotation with planar translations. Using the assumption the sensors are near-boresight aligned, the error introduced while using the translation mapping are quite small. As is to be expected, as the near boresight constraint is relaxed, increasing amounts of error are introduced.

References

[Ant96] Anthony N. A. Schwickerath and J. Ross Beveridge. Coregistration of Range and Optical Images Using Coplanarity and Orientation Constraints. In *1996 Conference on Computer Vision and Patter Recognition*, page (to appear), San Francisco, CA, June 1996.

[Bel93] Mark Bellrichard. Alliant techsystems LADAR field calibration. Personal Correspondence, November 1993.

[BHP95] J. Ross Beveridge, Allen Hanson, and Durga Panda. Model-based Fusion of FLIR, Color and LADAR. In Paul S. Schenker and Gerard T. McKee, editors, *Proceedings: Sensor Fusion and Networked Robotics VIII*, Proc. SPIE 2589, pages 2 – 11, October 1995.

[BPY94] J. Ross Beveridge, Durga P. Panda, and Theodore Yachik. November 1993 Fort Carson RSTA Data Collection Final Report. Technical Report CSS-94-118, Colorado State University, Fort Collins, CO, January 1994.

[CB93] James L. Crowley and Philippe Bobet. Maintaining stereo calibration by tracking image p oints. In *CVPR–93 Proceedings*, pages 483 – 488, New York, NY, 1993.

[CCHR94] Y-Q Cheng, R. Collins, A.R. Hanson, and E.M. Riseman. *Robust camera calibration for a moving camera on a mobile robot.* Master Thesis, Computer Science, University of Massachusetts, 1994.

[Fau93] O.D. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, Cambridge, MA, 1993.

[FD82] J. D. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics.* The Systems Programming Series. Addison–Wesley, Reading, Massachusetts, 1982.

[Gan84] S. Ganapathy. Decomposition of Transformation Matrices for Robot Vision. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 130–139, 1984.

[GBSF94] Michael E. Goss, J. Ross Beveridge, Mark Stevens, and Aaron Fuegi. Visualization and Verification of Automatic Target Recognition Results Using Combined Range and Optical Imagery. In *Proceedings: Image Understanding Workshop*, pages 491 – 494, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[GBSF95] M. E. Goss, J. R. Beveridge, M. Stevens, and A. Fuegi. Three-dimensional visualization environment for multisensor data analysis, interpretation, and model-based object recognition. In *IS&T/SPIE Symposium on Electronic Imaging: Science & Technology*, pages 283 – 291, February 1995.

[J. 96] J. Ross Beveridge and Bruce A. Draper and Kris Siejko. Progress on Target and Terrain Recognition Research at Colorado State University. In *Proceedings: Image Understanding Workshop*, page (to appear), Los Altos, CA, February 1996. ARPA, Morgan Kaufman.

[Kum92] Rakesh Kumar. *Model Dependent Inference of 3D Information From a Sequence of 2D Images.* PhD thesis, University of Massachusetts, COINS TR92-04, Amherst, February 1992.

[LT86] R.K. Lenz and R.Y Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-d machine vision metrology. *Trans. PAMI*, 10(5), 1986.

[Mar63] D. W. Marquardt. An algorithm for least squares estimation of nonli near parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[PFTV88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 1988.

[RL87] P. J. Rousseeuw and A. M. Leroy. *Robust regression & outlier detection.* John Wiley & Sons, New York, 1987.

[SB94] Anthony N. A. Schwickerath and J. Ross Beveridge. Model to Multisensor Coregistration with Eight Degrees of Freedom. In *Proceedings: Image Understanding Workshop*, pages 481 – 490, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[STH80] C. Slama, C. Theurer, and S. Henriksen. *Manual of Photogrammetry, Fourth Edition.* American Society of Photogrammetry, 1980.