RESECTION AND MONTE CARLO COVARIANCE FROM VANISHING POINTS FOR IMAGES OF UNKNOWN ORIGIN

Reuben Settergren

BAE Systems, Geospatial eXploitation Products™ (GXP®)
10920 Technology Drive, San Diego CA – reuben.settergren@baesystems.com

KEY WORDS: Photogrammetry, vanishing points, parallel lines, resection, 3D scene reconstruction

ABSTRACT:

Photogrammetric analysis requires camera metadata (position, attitude, interior orientation, etc.), which is not available for all images. Modern commercial solutions for 3D reconstruction from images typically assume large amounts of purposefully-collected, highly-overlapping imagery. In this article, a system is demonstrated for recovering a 3D scene from images of unknown origin, by marking ground space axes, resecting a camera consistent with the markings, and then using the solved camera to collect 3D measurements. The capability works with close-range (vanishing points) and long-range (parallel axes) imagery. Monte Carlo analysis is used to propagate measurement uncertainty of the vanishing lines to uncertainty of exterior and interior camera parameters, which can then be used to quantify uncertainty of measurements in the 3D scene.

1. INTRODUCTION

Given information about the state of a camera (position, attitude, optical characteristics) at the moment of image collection, 2D photographs can be used for 3D reconstruction, through the application of standard techniques of projective geometry (Hartley, Zisserman, 2003). 3D reconstruction of a bank from security footage could support forensic estimation of the height of the suspects, or the length of their weapons. Reconstruction of a vehicular accident scene would enable measurement of distances between objects, or lengths of skid marks.

Before the advent of photogrammetric software that enabled exploitation of digital/digitized images, this kind of photogrammetric analysis was conducted in hardcopy. Image prints were blown up and covered with table-size sheets of acetate, and the techniques detailed in (Williamson, Brill, 1990) were applied by carving classical straightedge/compass geometric constructions (vanishing lines, vanishing points, perpendicular lines, circular arcs, etc.) into the acetate with a sharp stylus. Camera parameters such as principal point, focal length, attitude angles, and camera position can then be extracted from certain points, lines, and angles in the construction. Along with the constructive geometric method for recovering generic camera parameters, analysts would have specific procedures for transferring known scale information from one element of the scene to other locations and dimensions. For instance, a known horizontal width of a window could be used to estimate the heights of various persons standing in a room.

These hardcopy techniques for camera resection can be replicated more efficiently with computer software. In the 1980s, as computers became sophisticated enough to process and display digitized imagery, the field of computer vision developed techniques for automating this analysis (Magee, Aggarwal, 1984, Williamson, Brill, 1989, Williamson, Brill, 1990, Caprile, Torre, 1990, Hartley, Zisserman, 2003). In subsequent decades, as computers grew more powerful, academic and industrial focus shifted to reconstruction of 3D scenes from large numbers of purposefully-collected, highly-overlapping, same-camera, fixed-zoom images, using techniques such as Simultaneous Location and Mapping (Durrant-Whyte, Bailey, 2006), and Structure-from-Motion (Ullman, 1979, Forsyth, Ponce, 2003). A standout exception is Make3d (Saxena, Ng, 2008), which reconstructs the camera and scene from 'superpixel' groups of coplanar pixels, but is more geared towards aesthetic and qualitative reconstruction of a 3D scene, rather than quantitative (precision mensuration).

Metric cameras (which are internally calibrated and collect camera position and orientation with high precision) make the task of 3D reconstruction much more efficient and accurate. But even a metric camera would not support reconstruction of an object in motion, such as a car on a revolving display platform at an auto show, or an airplane during takeoff; the camera's pose information would support 3D reconstruction only of the static environment surrounding the object in question. Reconstruction of the moving object would require camera models that are situated in the moving frame of reference of the body of the moving object. Additional complexity arises with photographs from different cameras, or even of different instances of the same type of object.

The goal of this work is to support precision mensuration of objects images of unknown origin. Given user measurements of parallel lines, a Python tool estimates vanishing points and resects a camera and pose. Monte Carlo analysis is used to perturb and iterate the resection, yielding meaningful estimates of the uncertainty of the resulting camera parameters. The camera parameters can be imported for use in a tool that exploits digital imagery, such as SOCET GXP® geospatial intelligence software,[1] enabling 3D navigation throughout the ground scene, and determination of the 3D locations of pixels by projecting to ground planes or triangulation with multiple images.

2. METHOD

The first step in recovering camera models is to select a coordinate system. A right-handed, orthogonal coordinate system should be chosen, for which ground-parallel lines are observable in all images. The conventions of this work are illustrated in Figure 1, with ground axes chosen so that in the image perspective, X will be to the right, Y will be up, and Z will be most into the camera. The image axes x, y are also indicated in parallel in ground space. The image plane is shown in positive, in front of the focal center (rather than behind C in mirror image). The perpendicular line segment from image plane 'principal point' (c_x, c_y) to C is the focal length f, in units of pixels. Image z, out of the camera, completes a right-handed orthogonal system with x and y. Camera rotation matrix R describes the rotation between XYZ and xyz.

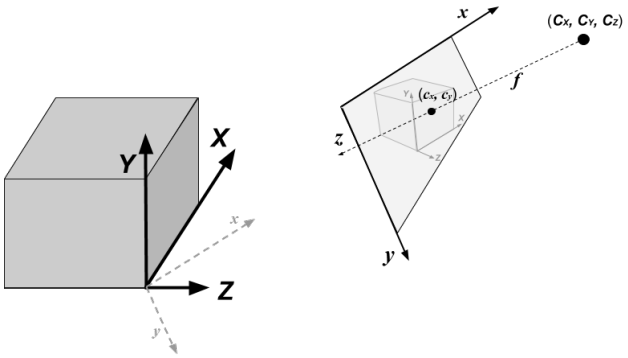

Figure 1. The scene has right-handed axes X, Y, Z, with Z most into the image, and X and Y projecting into image space as nearly right and up. The image axes x and y are right and down from the upper-left of the image, can be seen paralleled in ground space. Image z is out of the camera.

3D points in the scene will be indicated with capital coordinates like (X, Y, Z), and 2D image coordinates (pixels from the image upper-left corner) will be indicated like (x, y). It is convenient to present coordinates as row-vectors in inline text, but context may dictate they are actually used as column-vectors.

It is possible to observe the origin and lines parallel to the X, Y, and Z ground axes, as they appear in perspective in 2D image coordinates. In long-range imagery (essentially infinite focal length), ground-parallel lines will also be parallel in image space. In close-range imagery, ground-parallel lines will converge to vanishing points. In either case, the configuration of ground-space lines parallel to coordinate axes, as projected into 2D vectors in image space, uniquely determine the interior orientation (focal length and principal point) and relative exterior orientation (viewing direction and orientation) of the camera with respect to the chosen coordinate system, leaving a degree of freedom of unknown scale, or equivalently, unknown range from the camera to the ground origin. Additional inputs such as scale bars, or known/assumed focal ratio or range, suffice to determine a complete pinhole camera.

2.1 Camera Matrix

The standard (Hartley, Zisserman, 2003) matrix representation of a projective camera (equivalent to the photogrammetric collinearity equations (ASPRS, 1980)) uses a calibration matrix:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

On the diagonal of K, f represents the focal ratio, in units of pixels (the ratio between the focal length and the size of an individual pixel on the focal plane). PP=(c_x, c_y) are the image coordinates of the principal point (where imaging rays are perpendicular to the focal plane). K summarizes the calibration or 'interior orientation' of the camera.

The pose or 'exterior orientation' is captured in a rotation-translation matrix

$$Rt = \begin{bmatrix} R|t \end{bmatrix},$$

where R is a 3×3 rotation matrix, and t is the 3×1 focal center, in the camera's xyz frame of reference (note how the image axes in Figure 1 are paralleled in ground space); if the camera position in ground scene coordinates is (C_X, C_Y, C_Z), then $t = -RC$ and $C = -R't$.

Multiplied together, $K \cdot Rt$ is the 'camera matrix.' The projection of ground coordinates into image space is

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = K \cdot Rt \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

The 3-dimensional ground coordinates of a scene point (X, Y, Z) are augmented with a fourth homographic scale coordinate of 1, and after application of the 3×4 camera matrix, the output is an image coordinate, also augmented with a homographic scale. The actual 2D image coordinate to which the ground coordinate projects is then $(x/w, y/w)$. This homographic unscaling is well-defined everywhere except for the camera center, which is the only point that yields $w = 0$. Points behind the camera yield $w < 0$.

2.2 Vanishing Point Determination

If lines in any ground axis direction are not parallel in image space, all lines corresponding to that axis converge to a vanishing point. Depending on the range and the relationship of the camera focal plane to ground axes, an image may exhibit vanishing points for 0, 1, 2, or 3 dimensions. The most general case is 3-point perspective (3PP), which has all three vanishing points (VPX, VPY, and VPZ); the scope of this paper does not permit full treatment of 2-point and 1-point perspective, but the principles are similar, and the reader is referred to (Williamson, Brill, 1990). Long-range also generalizes 3PP, in the absence of vanishing points.

For determining the image coordinates of vanishing points, if only two line segments are provided, the vanishing point must be estimated simply as the intersection of the line segments extended. Use of multiple line segments per axis adds redundancy and allows averaging down of error. Due to camera imperfections and measurement error, multiple vanishing lines will not intersect in a unique point. When a particular pair of vanishing lines are very nearly parallel, measurement error can tilt them so that their intersection is in the opposite direction. Measuring as many vanishing lines as possible increases the possibility of this problem. This work uses a least-squares solution for the

location of the vanishing point, from all of the measured line segments simultaneously (Williamson, Brill, 1989), rather than from individual pairs, allowing use of as many vanishing lines as possible, for highest redundancy and accuracy.

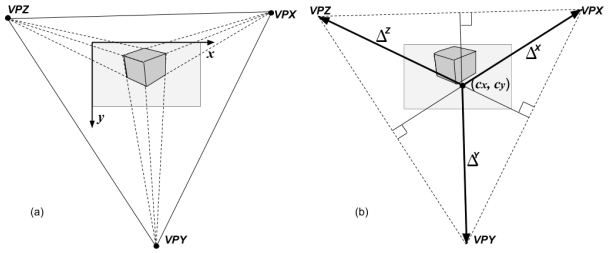

Figure 2. (a) Parallel line segments extend to vanishing lines which intersect at vanishing points. (b) The principal point is located at the center of the vanishing point triangle, and vectors $\Delta^X, \Delta^Y, \Delta^Z$ toward the vanishing points partially determine the rotation matrix.

2.3 Principal Point and Focal Length

The geometry of 3PP is often described using a 'Perspective Pyramid' (See Figure 3). The base of the pyramid is the triangle formed by the three vanishing points. The principal point (PP=(c_x, c_y) in calibration matrix K) can be found at the 'center' of the triangle – the intersection of the three altitudes (lines perpendicular the triangle sides, and through the opposite vertices), as shown in Figure 2.

If the 2D image plane is augmented with a z dimension orthogonal to the image plane (out of the camera towards the scene), then the focal center of camera station lies directly 'above' or 'behind' PP=(c_x, c_y) at CS=$(c_x, c_y, -f)$. Edges from CS to all three vanishing points constitute the Perspective Pyramid. All three angles between the edges emanating from the apex towards the vanishing points are right angles, and the line from CS to the principal point is perpendicular to the vanishing point triangle and the whole image plane.

The vanishing point triangle and the principal point at its center, can be estimated solely from ground-parallel line segments which converge in image space. The height f of the Perspective Pyramid is then determined by the constraint of right angles at the apex. (Williamson, Brill, 1990) provide a geometric method for determining f, by rotating one right-triangular face of the Perpsective Pyramid into the image plane, and using geometrical construction to determine the distance between the principal point and CS.

Alternatively, f can be solved for algebraically using the orthogonality constraint, for instance:

$$[(VPX, 0) - (c_x, c_y, f)] \cdot [(VPY, 0) - (c_x, c_y, f)] = 0 \quad (2)$$

Equation (2) can be solved for f using any two vanishing points.

2.4 Solving for Orientation

The orientation of the camera is determined by forming rotation matrix R from the 3D vectors from the CS to the vanishing points.

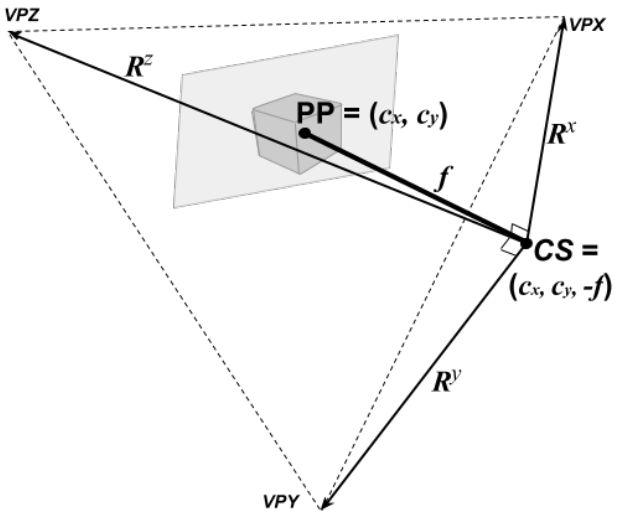

Figure 3. The 'Perspective Pyramid' is formed with the vanishing point triangle as the base, and the apex 'behind' or 'above' the principal point. The apex is at a height such that all three sides are right triangles, and that height is the focal length f. 3D image space vectors R^x, R^y, R^z from CS to the vanishing points form the rotation matrix.

Labelling the individual elements of Rt from (1), we can project X-axis ground coordinate $(X, 0, 0)$ into image space as follows:

$$\begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_x^x & R_x^y & R_x^z & t_x \\ R_y^x & R_y^y & R_y^z & t_y \\ R_z^x & R_z^y & R_z^z & t_z \end{bmatrix} \begin{bmatrix} X \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} f(XR_x^x + t_x) + c_x(XR_z^x + t_z) \\ f(XR_y^x + t_y) + c_y(XR_z^x + t_z) \\ XR_z^x + t_z \end{bmatrix} \quad (4)$$

$$\equiv \frac{f}{R_z^x + t_z/X} \begin{bmatrix} R_x^x + t_x/X \\ R_y^x + t_y/X \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \quad (5)$$

By definition, VPX is $\lim_{X \to \infty}$ of (5), so:

$$\mathrm{VPX} = \frac{f}{R_z^x} \begin{bmatrix} R_x^x \\ R_y^x \end{bmatrix} + \mathrm{PP}$$

At this point, R_z^x is still unknown, but since

$$\begin{bmatrix} R_x^x \\ R_y^x \end{bmatrix} = k(\mathrm{VPX} - \mathrm{PP}) \quad (6)$$

for some constant k, the first two rows of R^x form a vector parallel to VPX-PP. Similarly, the first two elements of rotation matrix columns R^y and R^z are parallel to VPY-PP and VPZ-PP.

The elements in the third row of R are determined by orthogonality constraints ($R^i \cdot R^i = 1$, $R^i \cdot R^{j \neq i} = 0$). As shown in Figure 3, the 3D vectors from the camera station to the vanishing points are mutually orthogonal since they are the rays of the apex of the Perspective Pyramid, so it is clear that augmenting each (R_x^*, R_y^*) with a third dimension of f, and then unitizing, will yield three orthogonal columns for R which retain the property that, for each column, the first two elements form a vector parallel to a vanishing direction. (Caprile, Torre, 1990) uses a similar analysis to determine the relative rotation

of the second image in a stereopair to the first.

Once R is known, orientation angles $\omega\phi\kappa$ can be determined with standard methods (ASPRS, 1980).

2.5 Solving for Camera Position

At this point, K and R of (1) are determined, and all that yet needs to be solved is camera position t.

Let the observed image coordinates of the ground origin be (o_x, o_y), and without loss of generality, temporarily assume that the range of the camera is $t_z = 1$. Projecting the ground origin into camera space by also substituting $X = 0$ into (4), it can be seen that t is the solution of

$$\begin{bmatrix} ft_x + c_x \\ ft_y + c_y \end{bmatrix} = \begin{bmatrix} o_x \\ o_y \end{bmatrix}$$

This is a simple solution of two independent equations.

If $t = (t_x, t_y, 1)$ causes camera KRt to project the ground origin correctly to (o_x, o_y), then it is easily shown that scaling the camera position to $t = (rt_x, rt_y, r)$ continues to project the ground origin into (o_x, o_y). This is a last remaining degree of freedom in the range of the camera, or equivalently, the scale of the 3D scene. The position vector can be scaled continuously to make any particular length in the scene match any known or assumed length. In addition, given camera position t in the camera's frame of reference, the camera's position in the ground frame of reference can be as $C = -R't$.

2.6 2-Point Perspective and 1-Point Perspective

The scope of this article does not permit a full treatment of 2-Point and 1-Point Perspective. They can be handled with techniques similar to 3PP, except that they require additional information to compensate for additional degrees of freedom. For instance 3PP is ambiguous only in scale, but 2PP has a degree of freedom in one orientation angle. A photograph taken with a perfectly horizontal camera results in vertical ground lines being parallel in image space, so vanishing point VPY retreats to infinity. In this limiting situation, PP as the center of the triangle has moved to a point on the horizon between finite vanishing points VPX and VPZ; but where exactly PP belongs on that horizon cannot be determined without knowledge of the orientation angle of some vertical or horizontal line segment. Typically, this degree of freedom is resolved by constructing a square around a known circular object in a principal plane (such as a wheel or a manhole cover), and a 45-degree diagonal drawn across the square. (Williamson, Brill, 1990) provides geometric contructions that use the line segment of known orientation to precisely locate PP on the VPX-VPZ horizon.

2.7 Long-Range Camera Determination

Recall that the apex of the Perspective Pyramid is the meeting of three right angles (Figure 3). Increasing the focal length – while maintaining the apex's rigid cluster of right triangles – causes the vanishing points that form the base of the pyramid to retreat further outward, away from the finite extent of image space encompassed by the image. As a result, ground lines parallel to the XYZ axes, projected into image space, become nearly parallel when viewed from within the bounds of the image. With a long enough focal length, ground-parallel lines are *practically* parallel in image space – within human ability to locate endpoints of line segments (at best 0.1 pixels). In long-range cases like this, it is not critical to know the exact focal length; an arbitrarily long focal length that produces practically-parallel ground axes in image space is sufficient, as long as range is also set to yield appropriate scale. The PP can be arbitrarily chosen at the image center, since there are no differences in perspective when moving around the image, and any required offset can be taken up in camera position.

Since ground axes appear parallel in image space, there are no vanishing points to measure, but the common direction of all lines for the same ground axis serves the same purpose as VP[XYZ]-PP in (6), allowing the same solution for R and $\omega\phi\kappa$ as in section 2.4. To determine each of those VP[XYZ]-PP vectors from multiple marked lines, line segments can be summed as vectors and then unitized; this naturally weights longer lines more and shorter lines less, which is appropriate because the same amount of endpoint measurement error in a shorter line segment causes more inaccuracy in determining the true direction of the vector.

Camera horizontal offset and range can be determined as in section 2.5, ensuring that the ground origin projects to the chosen/marked image coordinates, and the camera has the desired scale.

2.8 Monte Carlo Determination of Covariance

All of the calculations above (vanishing points, determination of PP and f from vanishing points, determination of attitude from PP and the vanishing points, and horizontal shift and range adjustment to match origin and scale) are subject to error, tracing back to measurement error at the endpoints of the line segments which must be extended and intersected to locate the vanishing points. In a real-world photographic scene, however, pixel quantization and viewing interpolation limit the accuracy of human image coordinate measurement to no better than 0.1 pixels, in optimum conditions. Thus, it is desirable to propagate this significant source of error into the space of the resulting camera parameters.

This is done using a Monte Carlo simulation.[2] Given a set of line segment endpoint coordinates, and an estimate[3] of measurement error σ, the endpoints are all randomly perturbed using a gaussian $N(0, \sigma)$ distribution, and all of the calculations repeated with the perturbed measurements, yielding one sample – a 9-vector of camera parameters $(f, c_x, c_y, \omega, \phi, \kappa, t_x, t_y, t_z)$. The perturbation and calculation are repeated many times, and the empirical 9×9 joint variance-covariance matrix is computed for all of the samples.

Two notes are in order: First, because a measurement error in the direction of the line segment has no effect on the line segment's extension or the solution of vanishing points, all gaussian errors are applied in the direction perpendicular to the line segments. Second, the computed covariance matrix is completely dense, capturing important correlation information

[2] From the Latin *Monte*=Mountain, *Carlo*=Carl, as in "Even Carl Friedrich Gauss would consider an analytical solution to be mountainously hard." A modern equivalent would be "I can't be bothered to do it analytically."

[3] Measurement error can be assessed by repeatedly moving and replacing an endpoint, and computing the sample variance of the resulting coordinates.

between camera parameters. The resulting dense parameter covariance matrix can be used by both for propagation of mensuration error into ground space for mono or stereo exploitation, or as an input to least-squares bundle-adjustment, ensuring that all parameter corrections are assigned rigorously meaningful weights.

2.9 Implementation

The method of camera recovery outlined above was implemented in Python.

The inputs for a 3PP resection are:

- A line segment (two pairs of image coordinates of endpoints) to serve as a scale bar along the X axis, and the ground length l of that scale bar.
  - The first endpoint of the scale bar is taken to be the ground origin $(0, 0, 0)$.
  - The other end point is $(l, 0, 0)$.
  - This could be generalized to be a scale bar between any specified 3D ground coordinates.
- Image coordinate endpoint pairs for at least two line segments in each axis.
- Measurement sigma and number of iterations for Monte Carlo covariance estimation.

The outputs are:

- Vanishing point coordinates VPX, VPY, VPZ.
- Camera parameters PP, f, orientation angles ω, ϕ, κ, and camera position (C_X, C_Y, C_Z).
- 9×9 covariance matrix estimated by Monte Carlo simulation.

For 2PP cases, the input also requires a pair of endpoints of a line segment known to be 45 degrees in the XY plane – again, this could be generalized to a line segment of any known angle in any known plane, but is most likely the diagonal of a square circumscribed around the circular rim of a wheel.

For long-range cases, a minimum of only one line segment per axis is required. A long-range resection also accepts the desired long focal ratio.

The Python script is sufficiently quick to enable relatively large Monte Carlo simulations, even single-threaded. Without Monte Carlo covariance estimation, a single camera solution is practically instantaneous. For analytical cube tests (four ground-parallel cube edges for each of the three vanishing points), the covariance matrix for 1000 samples can be generated on a Lenovo P50 laptop with an Intel Core i7 2.7GHz processor, in 1.30 seconds.

3. RESULTS

3.1 Simulated Images

Synthetic images are generated of a 1m cube, and camera parameters are varied to understand their effect on the accuracy of the resection process. The baseline image is seen in Figure 4. The cube is positioned from ground (0,0,0) to (1,1,1), and is rendered into a 1000x800 image using a camera set up as $f = 1000$pix, center PP, $\omega = 10°$, $\phi = 20°$, $\kappa = 30°$, from a position of $(C_X, C_Y, C_Z) = (2, -\frac{1}{2}, 5)$ (this is $t \approx (-0.33, 0.24, 5.39)$ in the camera's frame of reference).

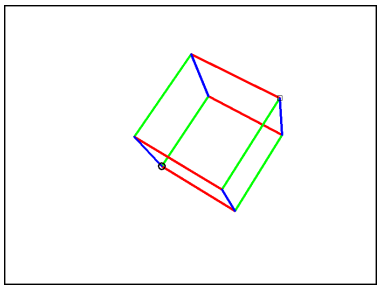

Figure 4. Baseline image of a 1m cube. Edges are color-coded by their axis direction, XYZ∼RGB. The origin is circled, and the square around the opposite corner (1,1,1) illustrates the extent of Figure 5.

Monte Carlo covariance estimation yields a 9×9 covariance matrix. To distill that into a more succinct metric of error, partial derivatives are computed of the ground-to-image projection function, with respect to the sensor parameters, at the opposite cube corner (1,1,1). The 9×9 covariance matrix in sensor parameter space is reduced to image space by outer-multiplying it by the 2×9 partial-derivative matrix, yielding 2×2 covariance in image space. For the baseline cube case, a reasonable 0.3 pixel measurement error sigma thus yields image space error:

$$\begin{bmatrix} 0.4088 & -0.4602 \\ -0.4602 & 2.2898 \end{bmatrix} \tag{7}$$

Figure 5 shows a scatterplot of 1000 reprojections, illustrating the magnitude of covariance matrix (7). Note the center ellipse captures 1σ error, with semimajor radius $\sigma_y = \sqrt{2.2898} \approx 1.54$, semiminor radius $\sigma_x = \sqrt{0.4088} \approx 0.64$, and tilt due to nonzero covariance/correlation. The outer two ellipses enclose 2σ and 3σ error regions.

Note that, while the scatter in Figure 5 is generated by directly exercising the perturbed cameras, the ellipses are generated from the Monte Carlo camera parameter covariance, and partial derivatives computed with the unperturbed camera. The coherence of the ellipses with the scatter is evidence of correctness for the Monte Carlo method.

The area of the 1σ ellipse is $1.54 \cdot 0.64 \cdot \pi \approx 1 \times \pi$ – equivalent to a circle of radius 1, so we will use this metric (the geometric mean of the x and y error sigmas) as a 1-dimensional summary of the error in the process, and call it an approximately 1-pixel reprojection error (a serendipitously round value) for the baseline case.

Starting from the baseline configuration described above, various settings are adjusted to examine their effect on accuracy.

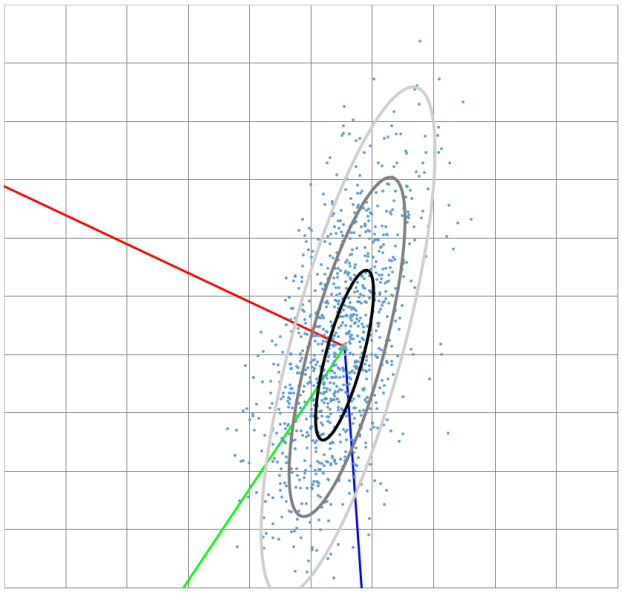

Figure 5. 1000 reprojections using Monte Carlo perturbations of the baseline model. The grid lines are individual pixels. The ellipses are 1σ, 2σ, and 3σ ellipses for covariance matrix (7).

Figure 6 presents the effect on reprojection error of these deviations from the baseline setup. Three different error response curves are presented with a unified X axis, which indicates the multiple of the baseline value that is used for the corresponding parameter. The baseline case is indicated at the intersection of the three curves (all three parameters at $1.0\times$baseline).

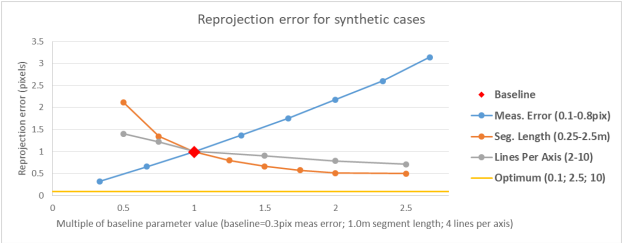

Figure 6. Simulated reprojection error for various configurations of measurement error and vanishing line segments. The error depicted in Figure 5 is the baseline case at the intersection of the three curves.

Obviously, increased measurement error leads to increased reprojection, which is illustrated in the increasing curve in Figure 6. Cutting measurement error to 1/3 of the baseline value (to $\sigma = 0.1$ pixels) decreases reprojection error from 1.0pix to 0.32pix, and tripling to $\sigma = 0.9$ more than triples the reprojection error to 3.14.

The natural 1m length of the vanishing line segments is ranged from 0.25 to 2.5m; just as a rifle offers smaller targeting error than a pistol, longer line segments yield less reprojection error than shorter; bracketing the 1pix baseline error with 2.1pix for 0.25m line segments, to 0.5pix for 2.5m segments.

The baseline number of line segments per vanishing point is the four in each dimension that are naturally present in a cube. This number is tested from the bare minimum of 2 ($0.5\times$baseline) to 10 ($2.5\times$baseline), demonstrating that measuring additional vanishing line segments also reduces error; however, doubling or halving the number of line segments per axis exerts less of an effect on reprojection error than line segment length, ranging from 0.72 to 1.45 pixels of error.

The line labeled 'Optimum' uses the most error-reducing value for each of curves in the Figure: $\sigma = 0.1$ pixel measurement error, and ten vanishing line segments per axis, each of ground length 2.5m. This most-optimistic configuration yielded a reprojection error of 0.097 pixels.

3.2 Real Images

3.2.1 Close-Range Example: Multi-View from Images of Different Volkswagen Things The Volkswagen Thing is perfectly suited to this analysis, with easily visible exterior lines in three orthogonal directions. Two VW Things, and vanishing lines, are displayed in Figure 7. In total, 35 images of VW Things were chosen from the internet (Wikimedia commons, 2020). Images from the right side of the car were flipped to simulate an equivalent image of the left side of the car. Cameras were resected for each according to the methods above. Scale was set to a 768mm door-to-door distance (based on a spec value of 727mm for the width of the door opening in the body, and scaled up using a 1PP image). The cameras were marked with min/avg/max 15/26/39 vanishing lines and Monte Carlo covariance took 1.4/3.4/11.0 seconds.

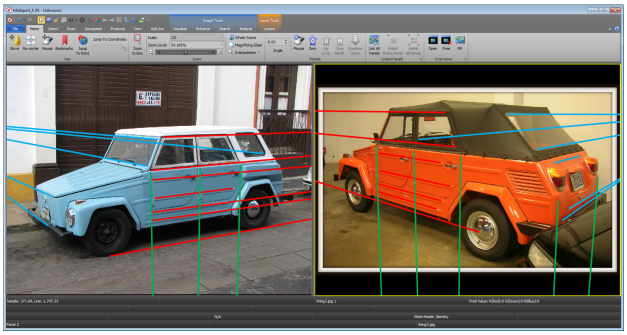

Figure 7. Two Volkswagen Things, viewed in 3-point perspective, with ground-parallel lines marked. Z is chosen as the dimension most into the image, completing a right-handed coordinate system with X towards image-right, and Y towards image-up.

Benchmark measurements for comparison are detailed in Table 1. It was not possible to identify both endpoints of each benchmark distance in all 35 images, because of perspective (images from the back of the car cannot see the windshield in the front), non-stock parts (custom bumpers or no bumpers at all), configuration (windshield folded down), blurriness, etc. The number, average, and sigma of estimates for each benchmark are detailed in Table 2. The L1 and L2 door width measurements are almost proxies for the door-to-door distance (including the pillar between the doors) that was used to set the scale of the project; but the door-to-door scale bar was established between the bottom corners of the doors, and the L1 and L2 door widths were measured across the top of the doors.

Lengths (distances along the chosen X-axis) and heights (Y-axis) are measured by simply projecting image coordinates to a depth of $Z = 0$ in the ground scene, and computing the distance between the ground points.

Widths (distances along the Z-axis) require an assumed ground X value for the desired line. First, the desired line is extended in

Label	mm	Description
L1	727	Max width of door opening (front)
L2	727	Max width of door opening (rear)
L3	3780	Length of car (bumper to bumper)
H1	345	Distance between door hinges (front)
H2	345	Distance between door hinges (rear)
W1	1495	Width of car body
W2	1353	Inner width of windshield frame (bottom)
W3	1270	Inner width of windshield frame (top)

Table 1. Benchmark VW Thing dimensions, categorized as Lengths, Heights, and Widths.

image space. Then the Y-axis ground line $(X, 0, 0) + c(0, 1, 0)$ is rendered in the image, and the intersection of the two lines determines the X, Y values of the Z-axis line in question. The extended line is traversed to determine the values Z_{lo}, Z_{hi} nearest the endpoints of the desired line segments, from which we compute the desired length.

Known	N	Avg	Err%	Sig	Multi	Err%
L1 (727)	28	726	0.13	7.4	726	0.13
L2 (727)	20	719	1.10	13.1	726	0.13
L3 (3780)	30	3827	1.24	86.9	3730	1.23
H1 (345)	31	353	2.31	8.1	351	1.73
H2 (345)	31	354	2.60	9.5	351	1.73
W1 (1495)	15	1496	0.07	53.9	1520	1.67
W2 (1353)	15	1393	2.96	37.0	1393	2.95
W3 (1270)	10	1283	1.02	26.9	1294	1.89

Table 2. Comparison of monoscopic (average) and multiscopic VW Thing measurements.

The 35 resected cameras were then formed into Frame sensor models for exploitation in SOCET GXP geospatial intelligence software, and then bundle-adjusted in Multi Sensor Triangulation (MST). The bundle adjustment involved 41 tie points. The two endpoints of the chosen scale were set to 3D control points at ground positions (0,0,0) and (0.768,0,0). All of the endpoints of the 8 benchmark measurements in Table 1 were included as tie points. The remaining 41-2-16=23 tie points were distributed around the vehicle and along all three dimensions. All nine camera parameters (three position, three orientation, focal length and principal point) were adjusted for all images, weighted by the Monte Carlo covariance at σ=0.3pixels. The aposteriori ground locations of the tie points were used to compute the distances in Table 2.

Since Table 2 lists only average values of monoscopic measurements, the spread of all the measurements are illustrated in Figures 8 and 9.

3.2.2 Long-Range Example: Stereo from Chips of WorldView-1 Satellite Imagery Small extents of imagery in Las Vegas, around the Stratosphere Tower were chipped out of WorldView-1 satellite imagery, stripping all collection metadata, and yielding dumb images with no geolocation or perspective information.

Three ground axis directions were chosen, corresponding to strongly visible features assumed to be orthogonal (given how buildings are most usually constructed), as depicted in Figure 10. The Z dimension of interest is most clearly visible in the tower, but also in other vertical building edges. The other chip is not shown, but the tower (and all vertical structures) exhibit a layover angle to the right, so the two images together are good

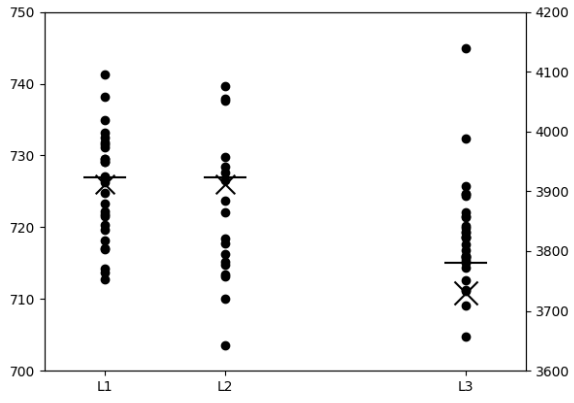

Figure 8. Monoscopic length (X-axis) estimates, in mm. L1 and L2 use the left axis, L3 the right. Benchmark values are indicated with —, and triangulated multiscopic estimates with $\times$.

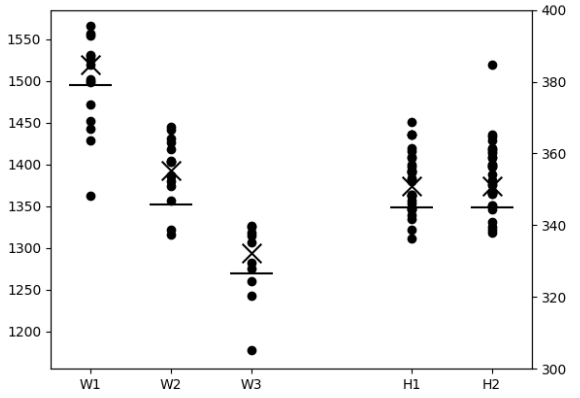

Figure 9. Monoscopic width (Z-axis) and height (Y-axis) estimates, in mm. Widths use the left axis, heights the right. Benchmark values are indicated with —, and triangulated multiscopic estimates with $\times$.

candidates for stereo exploitation. The published height of the Stratosphere Tower is 350.2m.

The origin is chosen to be at a point that can reasonably be assumed to be at the same elevation as the base of the tower. Actual WorldView-1 design parameters are ignored, and instead, as mentioned above, an arbitrarily long focal length of 1000mm is used together with a pixel size of 1μm, for a focal ratio of 10^6 pixels. The principal point is set arbitrarily (without loss of generality) at the center of the chip.

Orientation is solved for using the technique in section 2.4. The parallel vectors in each axis direction are used in place of vectors from the principal point towards the vanishing points. Section 2.5 is used to solve for camera horizontal offsets t_x, t_y. Range t_z is adjusted so that horizontal scale matches measurements of the area from online maps (the two chips require different ranges, since they were collected at different points along the orbit of the WorldView-1 spacecraft, with different slant ranges to the scene). Given a proper solution for the cameras, vertical scale will thus also be correct.

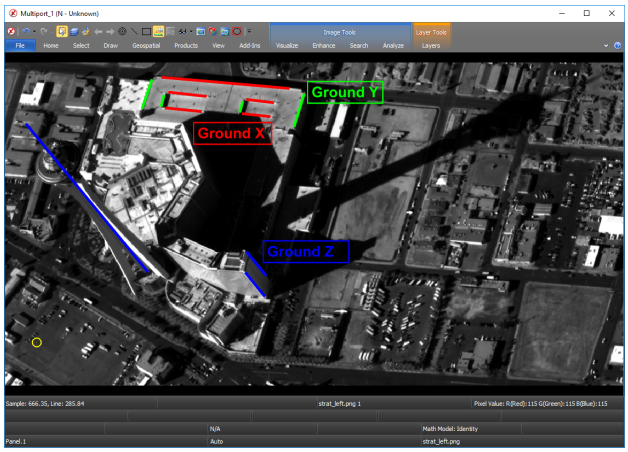

Figure 10. Stratosphere Tower, imagery chipped from WorldView-1 and marked with chosen ground origin (yellow marker) and coordinate axes. A second chip is also used, collected from a perspective such that the layover of the tower is to the right.

The height of the tower is measured in each of the resected images, using SOCET GXP geospatial intelligence software's monoscopic height measurement tool. In the image with the left-leaning tower (shown in Figure 10), the height is measured at 371.9m, an error of +6.20%. In the image with the tower leaning right (not depicted), the height of the tower is significantly underestimated at 309.5m (-11.6%).

After both long-range cameras are solved for, they are bundle-adjusted together in SOCET GXP geospatial intelligence software MST to improve stereo alignment for 3D exploitation. To ensure the images align around a common 3D model, and not just on the ground plane, tie points are included at a wide range of heights; particularly at the tip of the tower's spire, and around the observation deck. After bundle-adjustment, the tip of the spire is at Z=362.92m – that is, above the arbitrarily-chosen origin at street level. That is 3.6% above the known height of 350.2m. Part of that error might be extra height delta to the arbitrary origin, part will be from imperfect camera pose and scale.

4. CONCLUSION

In this work, camera models are developed for images using orthogonal ground axes evident in the scene. The technique generalizes to various close-range cases, in which ground-parallel lines converge to vanishing points in image space, and long-range cases, in which ground-parallel lines are also parallel in image space. The joint, correlated covariance of camera parameters is estimated with a Monte Carlo method based on measurement uncertainty of the user-marked line segment endpoints. The 3-point perspective case is examined with synthetic images, verifying that the estimated Monte Carlo covariance is consistent with simulated error. The techniques are then applied to real images. In the close-range situation, thirty-five 3-point perspective images of different Volkswagen Thing automobiles are used, and various recovered dimensions are within 3% or better of the published specifications. A long-range case is considered using overhead views of the Stratosphere Tower in Las Vegas, recovering the height of the tower to within 3.6% of the published value. These techniques bear much promise for 3D measurement from images of unknown provenance.

REFERENCES

ASPRS, 1980. *Manual of Photogrammetry, 4th ed.* ASPRS, Falls Church, VA.

BAE Systems Inc., 2019. SOCET GXP Software, version 4.4.0.3. `www.GeospatialExploitationProducts.com`.

Caprile, B., Torre, V., 1990. Using Vanishing Points for Camera Calibration. *Intl J of Computer Vision*, 4, 127-140.

DigitalGlobe, 2008. Stereo WorldView-1 imagery of Las Vegas, collected 2018-02-11. `DigitalGlobe.com`.

Durrant-Whyte, H., Bailey, T., 2006. Simultaneous Localization and Mapping (Parts I,II). *IEEE Robotics & Automation Magazine*, 13(2), 99-117.

Flickr user `jacksnell11707`, 2012. "1973 Volkswagen Thing 4 Door Convertible 2". `flickr.com/photos/jacksnell707/7068809657`.

Flickr user `squeakymarmot`, 2008. "powder blue VW thing". `flickr.com/photos/squeakymarmot/3178413638/`.

Forsyth, D., Ponce, J., 2003. *Computer Vision: A Modern Approach*. Prentice Hall.

Hartley, R., Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. 2 edn, Cambridge University Press, Cambridge.

Magee, M. J., Aggarwal, J. K., 1984. Determining Vanishing Points from Perspective Images. *Computer Vision, Graphics, and Image Processing*, 26, 256-267.

Saxena, A., M. S., Ng, A. Y., 2008. Make3d: Depth perception from a single still image. *In: AAAI Conference on Artificial Intelligence*, 1571-1576.

Ullman, S., 1979. The interpretation of structure from motion. *Proc Royal Society of London*, 203(1153), 405-426.

Wikimedia commons, 2020. Category:Volkswagen Type 181, (Accessed 2020-04-18). `commons.wikimedia.org/wiki/Category:Volkswagen_Type_181`.

Williamson, J. R., Brill, M. H., 1989. Dominant Geometry Combinations of Two- and Three-Point Perspective in Close-Range Applications. *Photogrammetric Engineering and Remote Sensing*, 55(2), 223-230.

Williamson, J. R., Brill, M. H., 1990. *Dimensional Analysis from Perspective: A Reference Manual*. ASPRS, Falls Church, VA.