3D Pose from 3 Corresponding Points under Weak-Perspective Projection

T. D. Alter

Abstract

Model-based object recognition commonly involves using a minimal set of matched model and image points to compute the pose of the model in image coordinates. Furthermore, recognition systems often rely on the "weak-perspective" imaging model in place of the perspective imaging model. This paper discusses computing the pose of a model from three corresponding points under weak-perspective projection. A new solution to the problem is proposed which, like previous solutions, involves solving a biquadratic equation. Here the biquadratic is motivated geometrically and its solutions, comprised of an actual and a false solution, are interpreted graphically. The final equations take a new form, which lead to a simple expression for the image position of any unmatched model point.

1 Introduction

Recognizing an object generally requires finding correspondences between features of a model and an image. Since finding corresponding features often requires trying all possible correspondences, recognition systems frequently use correspondences between minimal sets of features to compute poses of the model. For instance, "alignment" techniques repeatedly hypothesize correspondences between minimal sets of model and image features, and then use those correspondences to compute model poses, which are used to find other model-image correspondences (e.g., [5], [10], [1], [9], [28], [29], [15], [3], [16]-[18], [30], [19]). In addition, "pose clustering" techniques use every correspondence between a minimal set of model and image features to compute a model pose, and then count the number of times each pose is repeated (e.g., [2], [26], [25], [23], [11], [4]).

For computing poses of 3D objects from 2D images, a model of projection must be selected, and typically either perspective or "weak-perspective" projection is chosen. Weak-perspective projection is an orthographic projection plus a scaling, which serves to approximate perspective projection by assuming that all points on a 3D object are at roughly the same distance from the camera. For both perspective and weak-perspective projections, the minimal number of points needed to compute a model pose up to a finite number of solutions is three ([10], [18]). For point features, then, the problem is to determine the pose of three points in space given three corresponding image points. When perspective projection is the imaging model, the problem is known as the "perspective three-point problem" [10]. When weak-perspective is used, I shall call the problem the "weak-perspective three-point problem."

A few methods for solving the weak-perspective three-point problem have been suggested in the past ([20], [8], [17], [18], [12]), and this paper proposes a new method (solution). The major differences with the new solution is that it motivates and explains the solution geometrically, and it does not compute a model-to-image transformation as an intermediate step. As will be demonstrated later, understanding the geometry is useful for seeing under which circumstances the solution simplifies or breaks, and for analyzing where the solution is stable. Furthermore, a geometric understanding may be useful for seeing how the solution is affected by error in the image and the model.

In addition to providing a geometric interpretation, the solution in this paper gives direct expressions for the three matched model points in image coordinates, as well as an expression for the position in the image of any additional, unmatched model point. Earlier methods all require the intermediate computation of a model-to-image transformation. This is meaningful because, as mentioned above, many alignment-based recognition systems calculate the 3D pose solution many times while searching for the correct pose of the model. Consequently, avoiding the intermediate calculation of the transformation could cause such systems to run faster.

To illustrate how significant such a speed-up can be, consider a system that performs 3D recognition by alignment using point features to generate hypotheses. The input to the system is a model and an image, and the goal is to identify all instances of the model in the image. The model is specified by a set of 3D points that can be detected reliably in images, along with any number of extended features whose projections can be predicted using points (e.g., line

segments, some sets of curves, and edges represented point-by-point). From the image, a set of 2D points is extracted by a low-level process that looks for points of the type corresponding to points in the model. The alignment algorithm proceeds as follows:

1. Hypothesize a correspondence between three model points and three image points.

2. Compute the 3D pose of the model from the three-point correspondence.

3. Predict the image positions of the remaining model points and extended features using the 3D pose.

4. Verify whether the hypothesis is correct by looking in the image near the predicted positions of the model features for corresponding image features.

This process is repeated until all pairs of triples of model and image (sensed) points have been tried. For m model points and s sensed points, there are $\binom{m}{3}\binom{s}{3}3!$ distinct pairs of model and image point triples. Consequently, the running time for the algorithm grows with the cubes of the numbers of model and image points. Since these numbers can be large, the model and image points typically are grouped in advance so that only triples of points from the groups have to be tried (e.g., [23], [21], [18]). This can bring the number of pairs of triples into a range where the algorithm is practical. Then a "constant-times" speed-up in the innermost loop of the algorithm, that is, steps 2-4 listed above, could give a substantial improvement in the overall execution time. As already suggested, the solution given in this paper should make steps 2 and 3 significantly faster.

From observing previous solutions, the solution given in this paper most resembles Ullman's ([28], [17]), in that both end up having to solve the same biquadratic equation, although each derives the biquadratic differently. In this sense, the solution given here is an extension of Ullman's, because, unlike Ullman's solution, it resolves which of the two non-equivalent solutions to the biquadratic is correct. In addition, this paper explains graphically why the two solutions arise and to what geometry each corresponds.

There is an intrinsic geometry that underlies the perspective three-point problem; it is shown in Fig. 1. In the figure, the three model points, $\vec{m}_0$, $\vec{m}_1$, and $\vec{m}_2$, are being perspectively projected onto three image points, $\vec{i}_0$, $\vec{i}_1$, and $\vec{i}_2$, via lines through the center of projection (center point), $\vec{p}$. The task is to recover $\vec{m}_0$, $\vec{m}_1$, and $\vec{m}_2$. The essential information is contained in the side lengths and angles of the surrounding tetrahedron.

Similar to the perspective case, there is an intrinsic geometry underlying the weak-perspective three-point problem, shown in Fig. 2. The picture shows the three model points being projected orthographically onto the plane that contains $\vec{m}_0$ and is parallel to the image plane, and then shows them being scaled down into the image. In addition, the picture shows the model points first being scaled down and then projected onto the image plane. In each case, the projection is represented by a solid with right angles as shown. The smaller solid is a scaled-down version of the larger. The relevant information consists of the side lengths of the solids and the scale factor.

In what follows, first the perspective case is discussed (Section 2). Then I summarize how to compute 3D pose from three corresponding points under weak-perspective projection

Figure 1: Model points $\vec{m_0}$, $\vec{m_1}$, and $\vec{m_2}$ undergoing perspective projection to produce image points $\vec{i_0}$, $\vec{i_1}$, and $\vec{i_2}$. a, b, and c are distances from the center point, $\vec{p}$, to the model points.
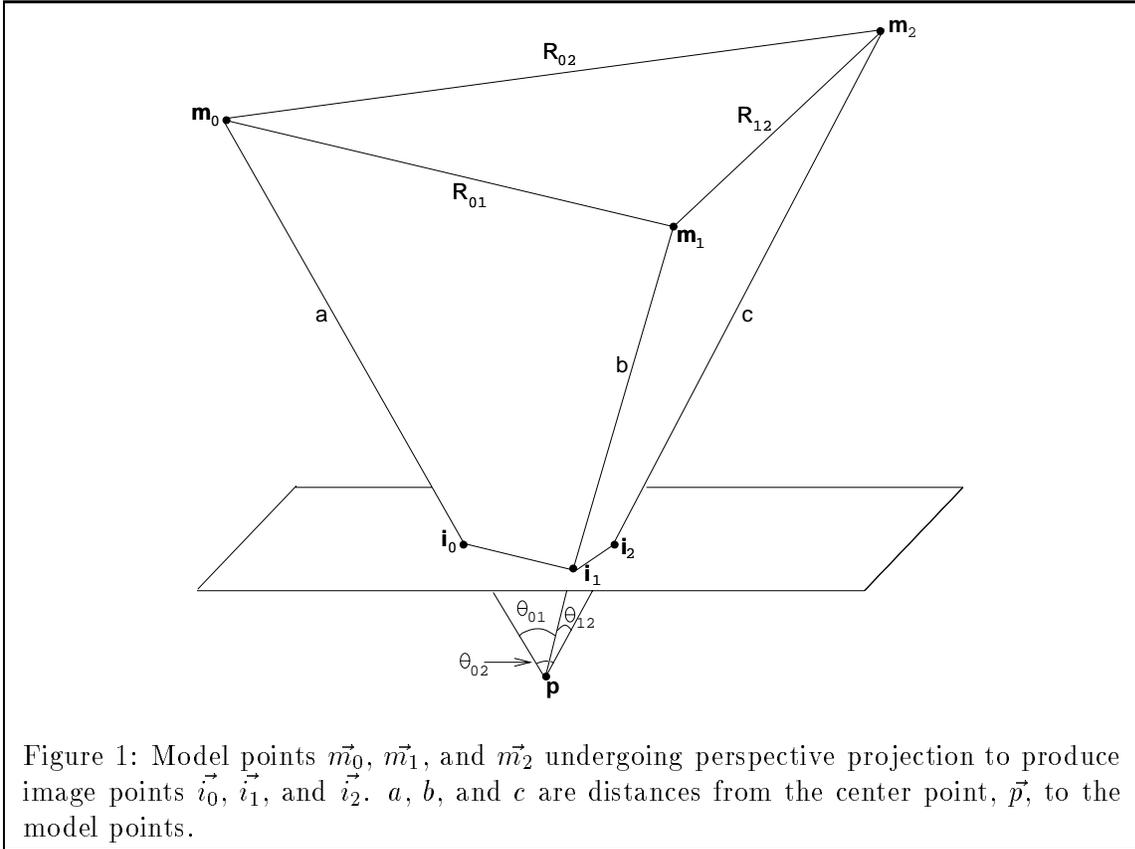
(Section 3). Third the 3D pose solution is shown to exist and be unique, and a geometrical interpretation is provided (Section 4). Next a direct expression is derived for the image position of an unmatched model point (Section 5). Then I review earlier solutions to the problem and present the three most related solutions in detail (Sections 7 and 8). In addition, the new and earlier solutions are examined and compared in terms of their stabilities (Sections 6 and 9).

2   The Perspective Solution

To see the difference between the perspective and weak-perspective cases, first let us observe exactly what is required for the perspective three-point problem. As pictured in Fig. 1, I will work in camera-centered coordinates with the center point at the origin and the line of sight along the z axis. The distances R_{01}, R_{02}, and R_{12} come from the original, untransformed model points. The angles θ_{01}, θ_{02}, and θ_{12} can be computed from the positions of the image points, the focal length, and the center point. To see this, let f equal the focal length, and let the image points $\vec{i_0}$, $\vec{i_1}$, $\vec{i_2}$ be extended as follows: $(x, y) \rightarrow (x, y, f)$. Then

$$\cos \theta_{01} = \widehat{i}_0 \cdot \widehat{i}_1, \qquad \cos \theta_{02} = \widehat{i}_0 \cdot \widehat{i}_2, \qquad \cos \theta_{12} = \widehat{i}_1 \cdot \widehat{i}_2, \tag{1}$$

where in general $\hat{v}$ denotes the unit vector in the direction of $\vec{v}$. The problem is to determine a, b, and c given R_{01}, R_{02}, R_{12}, $\cos\theta_{01}$, $\cos\theta_{02}$, and $\cos\theta_{12}$. From the picture, we see by the law of cosines that

$$a^2 + b^2 - 2ab\cos\theta_{01} = R_{01}^2 \tag{2}$$
$$a^2 + c^2 - 2ac\cos\theta_{02} = R_{02}^2 \tag{3}$$
$$b^2 + c^2 - 2bc\cos\theta_{12} = R_{12}^2 \tag{4}$$

Over time, there have been many solutions to the problem, all of which start with the above equations. The solutions differ in how they manipulate the equations when solving for the unknowns. Recently, Haralick et al. reviewed the various solutions and examined their stabilities [13].

Given a, b, and c, we easily can compute the 3D locations of the model points:

$$\vec{m}_0 = a\,\hat{i}_0, \qquad \vec{m}_1 = b\,\hat{i}_1, \qquad \vec{m}_2 = c\,\hat{i}_2. \tag{5}$$

If a 3D rigid transformation is desired, it can be determined from the original 3D model points and the 3D camera-centered model points just computed. A simple method for doing so is given in Appendix A; for a least-squares solution, see Horn [14].

Although perspective (central) projection is a more accurate model, numerous researchers have used weak-perspective projection instead (e.g., [24], [20], [7], [8], [25], [28], [29], [21], [22], [16]-[18], [3], [30], [19], [12]). The justification for using weak-perspective is that in many cases it approximates perspective closely. In particular, for many imaging situations if the size of the model in depth (distance in $\hat{z}$) is small compared to the depth of the model centroid, then the difference should be negligible [25].

There are some advantages to using weak-perspective instead of perspective. In particular, computations involving weak-perspective often are less complicated. In addition, the weak-perspective math model is conceptually simpler, since it uses orthographic instead of perspective projection. Another advantage is that we do not need to know the camera focal length or center point. Furthermore, there are fewer solutions to deal with—four for perspective and two for weak-perspective ([10], [18]). It should be understood, however, that finding two solutions instead of four is only an advantage if the four solutions actually collapse to two; otherwise, at least two of the solutions are missed.

Lastly, the weak-perspective imaging model can be used without modification to recognize scaled versions of the same object, since the built-in scale factor incorporates object scale. For perspective to handle scale, an additional scale parameter must be used. On the other hand, weak-perspective is unable distinguish objects that differ only in size, since a smaller scale could mean the object is smaller or further away. Nonetheless, in cases where the weak-perspective approximation applies, the perspective solution may be unstable in distinguishing different-sized objects ([27], [29], [18]). In these cases, moving the object further out in depth, that is, past the point where perspective and weak-perspective projections are essentially equivalent, will have the same effect in the image as uniformly scaling the object down in size. Since the perspective solution always distinguishes the depth and size of the object, this suggests that

small variations in the image could lead to very different interpretations for the size as well as the depth.

In sum, there are significant advantages to using weak-perspective in place of perspective, and under many viewing conditions the weak-perspective approximation is close to perspective. As suggested in the introduction, for these situations it would be useful to know how to solve, using weak-perspective projection, the problems of recovering the 3D pose of a model and computing the image position of a fourth model point.

3   Computing the Weak-Perspective Solution

This section provides a summary of the results I will derive in the next two sections. Specifically, it tells how to compute the locations of the three matched model points and the image location of any additional, unmatched model point.

For reference, the geometry underlying weak-perspective projection between three corresponding points, which was described in the introduction, is shown in Fig. 2. All that is pertinent to recovering the 3D pose of the model are the distances between the model and image points. Let the distances between the model points be (R_{01}, R_{02}, R_{12}), and the corresponding distances between the image points be (d_{01}, d_{02}, d_{12}). Then the parameters of the geometry in Fig. 2 are

$$s = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}} \tag{6}$$

$$(h_1, h_2) = \pm \left(\sqrt{(sR_{01})^2 - d_{01}^2}, \ \sigma\sqrt{(sR_{02})^2 - d_{02}^2} \right) \tag{7}$$

$$(H_1, H_2) = \frac{1}{s}(h_1, h_2) \tag{8}$$

where

$$a = (R_{01} + R_{02} + R_{12})(-R_{01} + R_{02} + R_{12})(R_{01} - R_{02} + R_{12})(R_{01} + R_{02} - R_{12}) \tag{9}$$

$$b = d_{01}^2(-R_{01}^2 + R_{02}^2 + R_{12}^2) + d_{02}^2(R_{01}^2 - R_{02}^2 + R_{12}^2) + d_{12}^2(R_{01}^2 + R_{02}^2 - R_{12}^2) \tag{10}$$

$$c = (d_{01} + d_{02} + d_{12})(-d_{01} + d_{02} + d_{12})(d_{01} - d_{02} + d_{12})(d_{01} + d_{02} - d_{12}) \tag{11}$$

$$\sigma = \begin{cases} 1 & \text{if } d_{01}^2 + d_{02}^2 - d_{12}^2 \leq s^2(R_{01}^2 + R_{02}^2 - R_{12}^2), \\ -1 & \text{otherwise.} \end{cases} \tag{12}$$

As the equations show, the solution has a two-way ambiguity except when h_1 and h_2 are zero. The ambiguity corresponds to a reflection about a plane parallel to the image plane. When $h_1 = h_2 = 0$, the model triangle (the triangle defined by the three model points) is parallel to the image triangle (the triangle defined by the three image points). As a note, a and c measure sixteen times the squares of the areas of the model and image triangles, respectively. Further, the solution fails when the model triangle degenerates to a line, in which case $a = 0$; in fact, this is the only instance in which a solution may not exist (for a discussion of this case, see Section 4.5). Note, however, that no such restriction is placed on the image triangle; so the
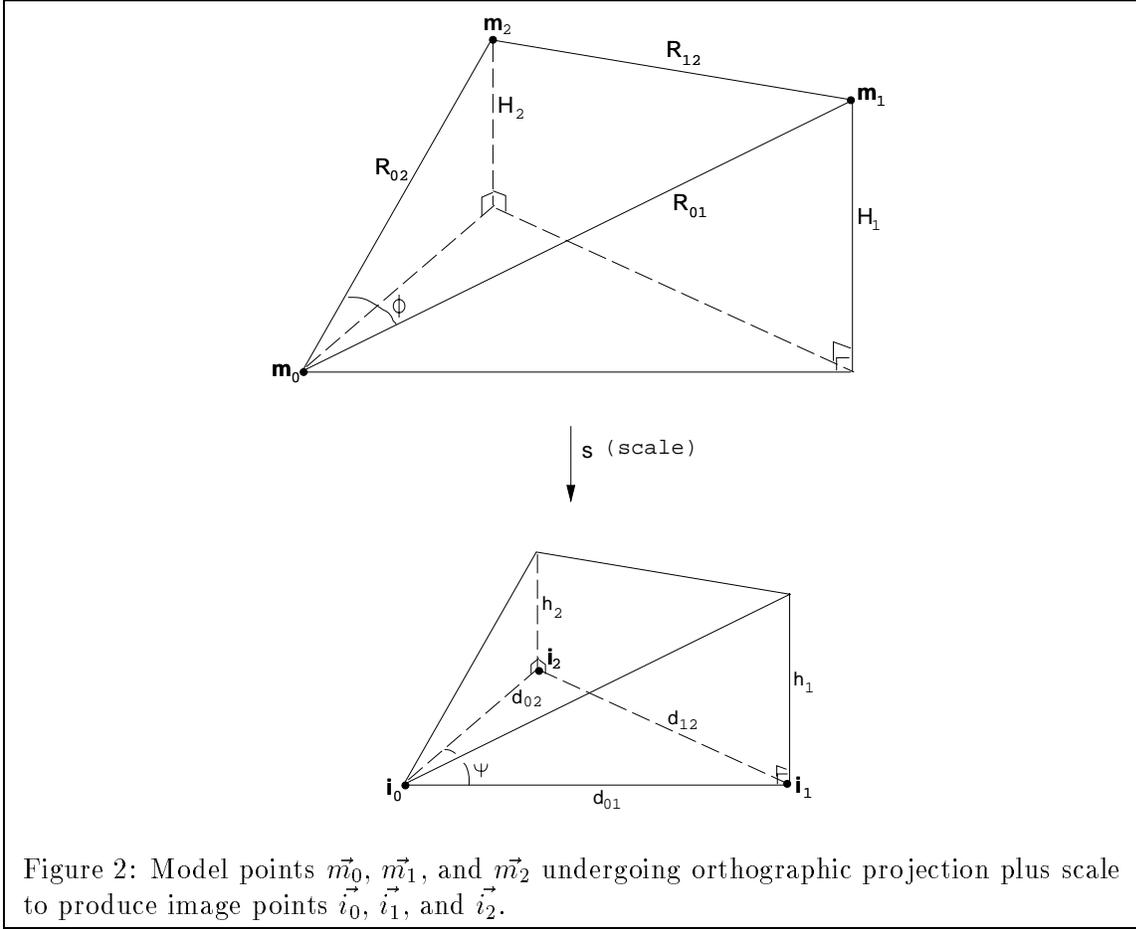
Figure 2: Model points $\vec{m_0}$, $\vec{m_1}$, and $\vec{m_2}$ undergoing orthographic projection plus scale to produce image points $\vec{i_0}$, $\vec{i_1}$, and $\vec{i_2}$.

image points may be collinear. Even so, care should be taken since the solution may be unstable when image points are collinear, when the model points are collinear, or when one of the sides of the model triangle is parallel to the image plane (see Section 6).

Next, I give an expression for the image location of a fourth model point. Originally, the models points are in some arbitrary model coordinate frame. Also, the image points are in a camera-centered coordinate frame in which the image serves as the x-y plane. Denote the original, untransformed model points by $\vec{p_i}$, to distinguish them from the camera-centered model points $\vec{m_i}$ shown in Fig. 2. Using $\vec{p_0}$, $\vec{p_1}$, and $\vec{p_2}$, solve the following vector equation for the "extended affine coordinates," (α, β, γ), of $\vec{p_3}$:

$$\vec{p_3} = \alpha(\vec{p_1} - \vec{p_0}) + \beta(\vec{p_2} - \vec{p_0}) + \gamma(\vec{p_1} - \vec{p_0}) \times (\vec{p_2} - \vec{p_0}) + \vec{p_0} \tag{13}$$

Given image points $\vec{i_0} = (x_0, y_0)$, $\vec{i_1} = (x_1, y_1)$, and $\vec{i_2} = (x_2, y_2)$, let

$$x_{01} = x_1 - x_0, \qquad y_{01} = y_1 - y_0,$$
$$x_{02} = x_2 - x_0, \qquad y_{02} = y_2 - y_0.$$

6

Then the image location of the transformed and projected $\vec{p}_3$ is

$$(\alpha x_{01} + \beta x_{02} + \gamma(y_{01}H_2 - y_{02}H_1) + x_0, \; \alpha y_{01} + \beta y_{02} + \gamma(-x_{01}H_2 + x_{02}H_1) + y_0). \qquad (14)$$

Lastly, the weak-perspective solution can be used to compute the 3D locations of the model points in camera-centered coordinates:

$$\vec{m}_0 \;=\; \frac{1}{s}(x_0, y_0, w) \qquad (15)$$

$$\vec{m}_1 \;=\; \frac{1}{s}(x_1, y_1, h_1 + w) \qquad (16)$$

$$\vec{m}_2 \;=\; \frac{1}{s}(x_2, y_2, h_2 + w), \qquad (17)$$

where w is an unknown offset in a direction normal to the image plane. It is worth noting that if the 3D rigid transform that brings the model into camera-centered coordinates is desired, it can be computed from these three camera-centered model points and the original three model points. The unknown offset w drops out when computing the rotation and remains only in the z coordinate of the translation, which cannot be recovered. As mentioned in Section 2, a simple method for computing the transform is given in Appendix A, and a least-squares solution was given by Horn [14].

4 Existence and Uniqueness of the 3D Pose Solution

In deriving the 3D pose solution, I start with the basic geometry for the weak-perspective three-point problem, shown in Fig. 2. Fig. 3 shows the smaller solid again with more labels. There are three right triangles in the solid, from which three constraints can be generated:

$$h_1^2 + d_{01}^2 \;=\; (sR_{01})^2 \qquad (18)$$

$$h_2^2 + d_{02}^2 \;=\; (sR_{02})^2 \qquad (19)$$

$$(h_1 - h_2)^2 + d_{12}^2 \;=\; (sR_{12})^2 \qquad (20)$$

It should be pointed out that the distances R_{01}, R_{02}, R_{12}, d_{01}, d_{02}, d_{12} and the scale factor s are all positive, but the altitudes h_1, h_2 along with H_1, H_2 are signed. Since h_1 and h_2 are signed, having "$h_1 - h_2$" in the third equation is an arbitrary choice over "$h_1 + h_2$"; it was chosen because, when h_1 and h_2 are positive, it directly corresponds to the picture in Fig. 3.

Multiplying the third equation by -1 and adding all three gives

$$2h_1 h_2 = s^2(R_{01}^2 + R_{02}^2 - R_{12}^2) - (d_{01}^2 + d_{02}^2 - d_{12}^2).$$

Squaring and using the first two equations again to eliminate h_1^2 and h_2^2, we have

$$4(s^2 R_{01}^2 - d_{01}^2)(s^2 R_{02}^2 - d_{02}^2) = \left(s^2(R_{01}^2 + R_{02}^2 - R_{12}^2) - (d_{01}^2 + d_{02}^2 - d_{12}^2) \right)^2, \qquad (21)$$

which leads to a biquadratic in s (for details see Appendix B):

$$as^4 - 2bs^2 + c = 0, \qquad (22)$$

7
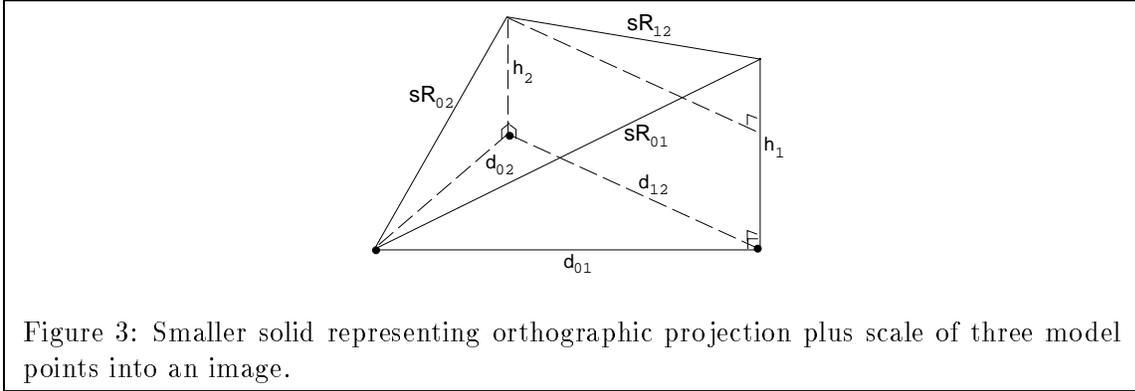
Figure 3: Smaller solid representing orthographic projection plus scale of three model points into an image.

where

$$
\begin{aligned}
a &= 4R_{01}^2 R_{02}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)^2 \\
&= (R_{01} + R_{02} + R_{12})(-R_{01} + R_{02} + R_{12})(R_{01} - R_{02} + R_{12})(R_{01} + R_{02} - R_{12})
\end{aligned}
$$

$$
\begin{aligned}
b &= 2R_{01}^2 d_{02}^2 + 2R_{02}^2 d_{01}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)(d_{01}^2 + d_{02}^2 - d_{12}^2) \\
&= d_{01}^2(-R_{01}^2 + R_{02}^2 + R_{12}^2) + d_{02}^2(R_{01}^2 - R_{02}^2 + R_{12}^2) + d_{12}^2(R_{01}^2 + R_{02}^2 - R_{12}^2)
\end{aligned}
$$

$$
\begin{aligned}
c &= 4d_{01}^2 d_{02}^2 - (d_{01}^2 + d_{02}^2 - d_{12}^2)^2 \\
&= (d_{01} + d_{02} + d_{12})(-d_{01} + d_{02} + d_{12})(d_{01} - d_{02} + d_{12})(d_{01} + d_{02} - d_{12})
\end{aligned}
$$

This biquadratic is equivalent to the one originally derived by Ullman. But Ullman made no attempt to interpret or decide among its solutions, which will be done here.

To prove existence and uniqueness, the biquadratic's solutions must be examined. We are interested only in positive, real solutions for s, the scale factor. In general, the positive solutions of the biquadratic are given by

$$
s = \sqrt{\frac{b \pm \sqrt{b^2 - ac}}{a}} \tag{23}
$$

Depending on the radicands, there will be zero, one, or two real solutions. Particularly, we are interested in whether each number of solutions can arise, and, if so, to what the solutions correspond geometrically.

To begin, let us determine the signs of a, b, and c. In Fig. 2, let ϕ denote the angle between $\vec{m}_1 - \vec{m}_0$ and $\vec{m}_2 - \vec{m}_0$, and let ψ be the angle between $\vec{i}_1 - \vec{i}_0$ and $\vec{i}_2 - \vec{i}_0$. Notice by the law of cosines that

$$
\begin{aligned}
a &= 4R_{01}^2 R_{02}^2 - (2R_{01} R_{02} \cos\phi)^2 \\
&= 4(R_{01} R_{02} \sin\phi)^2 \tag{24}
\end{aligned}
$$

$$
\begin{aligned}
b &= 2R_{01}^2 d_{02}^2 + 2R_{02}^2 d_{01}^2 - (2R_{01}R_{02}\cos\phi)(2d_{01}d_{02}\cos\psi) \\
&= 2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi)
\end{aligned}
\tag{25}
$$

$$
\begin{aligned}
c &= 4d_{01}^2 d_{02}^2 - (2d_{01}d_{02}\cos\psi)^2 \\
&= 4(d_{01}d_{02}\sin\psi)^2
\end{aligned}
\tag{26}
$$

Further, $\frac{1}{2}R_{01}R_{02}\sin\phi$ equals the area of the model triangle, so that a measures sixteen times the square of the area of the model triangle. Analogously, c measures sixteen times the square of the area of the image triangle.

In what follows, I assume that the model triangle is not degenerate, that is, not simply a line or a point. This situation is the only time the solution is not guaranteed to exist. (For a discussion of this case see Section 4.5.) Note that this assumption implies that $a \neq 0$ and $\phi \neq 0$.

From Equations 24 and 26, clearly $a > 0$ and $c \geq 0$. From Equation 25, it is straightforward to see that $b > 0$:

$$
\begin{aligned}
b &= 2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi) \\
&> 2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}), \quad \text{since } \cos\phi < 1, \cos\psi \leq 1 \\
&= 2(R_{01}d_{02} - R_{02}d_{01})^2 \\
&\geq 0
\end{aligned}
$$

Returning to Equation 23, Appendix E shows that $b^2 - ac \geq 0$. From this fact and that $a > 0$, $b > 0$, and $c \geq 0$, we can derive that there are in general two solutions for s with a single special case when $b^2 - ac = 0$, which can be seen as follows:

$$
\begin{aligned}
b^2 - ac \geq 0 &\implies b \pm \sqrt{b^2 - ac} \geq 0, \text{ since } b > 0 \text{ and } ac \geq 0 \\
&\implies \frac{b \pm \sqrt{b^2 - ac}}{a} \geq 0, \text{ since } a > 0
\end{aligned}
$$

Hence

$$
s = \sqrt{\frac{b \pm \sqrt{b^2 - ac}}{a}},
$$

which gives one or two solutions for the biquadratic, depending on whether $b^2 - ac$ is positive or equal to zero.

Next, I show that of the two solutions for the scale, exactly one of them is valid, that is, corresponds to an orthographic projection of the model points onto the image points. Furthermore, the other solution arises from inverting the model and image distances in Fig 2. In addition, there being one solution for scale corresponds to the special case in which the model triangle is parallel to the image plane. The following proposition, which is proved in Appendix C, will be useful in establishing these claims.

Proposition 1: Let

$$
s_1 = \sqrt{\frac{b - \sqrt{b^2 - ac}}{a}} \qquad s_2 = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}} .
\tag{27}
$$

Then

$$s_1 \leq \frac{d_{01}}{R_{01}}, \frac{d_{02}}{R_{02}} \leq s_2. \tag{28}$$

4.1 The true solution for scale

Here it is shown that exactly one of the two solutions for scale can satisfy the geometry shown in Fig. 2, and it is always the same one. If the two solutions are the same, then both solutions can satisfy the geometry (this case is discussed in Section 4.3). As will be seen, the valid solution is

$$s = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}}.$$

Note that proving this statement establishes the existence and uniqueness of the solution given in Section 3.

In Fig. 2, $(sR_{01})^2 - d_{01}^2 = h_1^2 \geq 0$ and $(sR_{02})^2 - d_{02}^2 = h_2^2 \geq 0$, which implies that any solution s for scale satisfies

$$\frac{d_{01}}{R_{01}} \leq s \qquad \text{and} \qquad \frac{d_{02}}{R_{02}} \leq s.$$

Consequently, Proposition 1 implies that s_2 is the only possible solution. Still, the question remains whether s_2 is itself a solution; the fact that it satisfies the biquadratic (Equation 22) is not sufficient since the steps used to derive the biquadratic from Equations 18-20 are not always reversible due to the squaring used to obtain Equation 21.

Next, I show that s_2 is indeed a solution by giving an assignment to the remaining variables that satisfies the constraints in Equations 18-20. Since $(sR_{01})^2 - d_{01}^2 \geq 0$ and $(sR_{02})^2 - d_{02}^2 \geq 0$, we can set $h_1^2 = (sR_{01})^2 - d_{01}^2$ and $h_2^2 = (sR_{02})^2 - d_{02}^2$, which immediately give Equations 18 and 19. Furthermore, we know s satisfies Equation 22, or, equivalently, Equation 21. Substitute h_1^2 and h_2^2 into the left-hand side of Equation 21:

$$4h_1^2 h_2^2 = \left(s^2 (R_{01}^2 + R_{02}^2 - R_{12}^2) - (d_{01}^2 + d_{02}^2 - d_{12}^2) \right)^2.$$

which is the same as

$$\pm 2 h_1 h_2 = s^2 (R_{01}^2 + R_{02}^2 - R_{12}^2) - (d_{01}^2 + d_{02}^2 - d_{12}^2).$$

At this point, we are free to choose the signs of h_1 and h_2. In particular, let the sign of h_1 watch the sign on the left-hand side so that

$$2 h_1 h_2 = s^2 (R_{01}^2 + R_{02}^2 - R_{12}^2) - (d_{01}^2 + d_{02}^2 - d_{12}^2). \tag{29}$$

Once this choice is made, we are forced to choose the sign of h_2 to make the sign of the left-hand side consistent with the right-hand side. In particular, let σ be the sign of h_2. Then unless the right-hand side is 0,

$$\sigma = \begin{cases} 1 & \text{if } s^2 (d_{01}^2 + d_{02}^2 - d_{12}^2) < R_{01}^2 + R_{02}^2 - R_{12}^2, \\ -1 & \text{if } s^2 (d_{01}^2 + d_{02}^2 - d_{12}^2) > R_{01}^2 + R_{02}^2 - R_{12}^2. \end{cases}$$

10
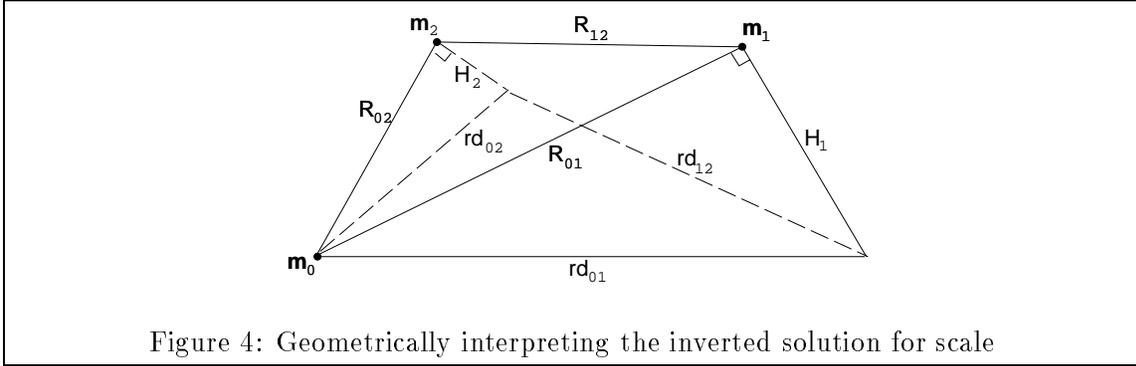
Figure 4: Geometrically interpreting the inverted solution for scale

On the other hand, if $s^2(R_{12}^2 - R_{01}^2 - R_{02}^2) = d_{12}^2 - d_{01}^2 - d_{02}^2$, then Equation 29 implies h_1 or h_2 is 0, so that the sign of h_2 is not forced and so is arbitrary. Having chosen the sign of h_2, substituting h_1^2 and h_2^2 into the right-hand side of Equation 29 gives

$$2h_1h_2 = h_1^2 + h_2^2 - (s^2 R_{12}^2 - d_{12}^2),$$

or

$$(h_1 - h_2)^2 = s^2 R_{12}^2 - d_{12}^2,$$

which is Equation 20.

Returning to the signs of h_1 and h_2, there is two-way ambiguity in the sign of h_1 which imposes the same two-way ambiguity on the pairs (h_1, h_2) and (H_1, H_2). As can be seen in Fig. 2, the ambiguity geometrically corresponds to a flip of the plane containing the space points $\vec{m}_0$, $\vec{m}_1$, and $\vec{m}_2$. The flip is about a plane in space that is parallel to the image plane, but which plane it is cannot be determined since the problem gives no information about offsets of the model in the z direction. Due to the reflection, for planar objects the two solutions are equivalent, in that they give the same image points when projected. On the other hand, for non-planar objects the two solutions project to different sets of image points.

There is a special case, as mentioned above, when the sign of h_2 is arbitrary relative to the sign of h_1. In this case, the right-hand side of Equation 29 is zero, and this implies that h_1 or h_2 is zero also. Looking at Fig. 2, geometrically what is occurring is that one of the sides of the model triangle that emanates from $\vec{m}_0$ lies parallel to the image plane, so that the reflective ambiguity is obtained by freely changing the sign of the non-zero altitude.

4.2   The inverted solution for scale

Of the two solutions for scale that satisfy the biquadratic, we know that one of them corresponds to the geometry in Fig. 2, but what about the other? Using a similar argument to that used to prove s_2 is a solution for the weak-perspective geometry, we can infer a geometric interpretation for s_1. Consider, then, $s = s_1$. The interpretation I will derive satisfies the equations,

$$H_1^2 + R_{01}^2 \quad = \quad (r d_{01})^2 \tag{30}$$

11
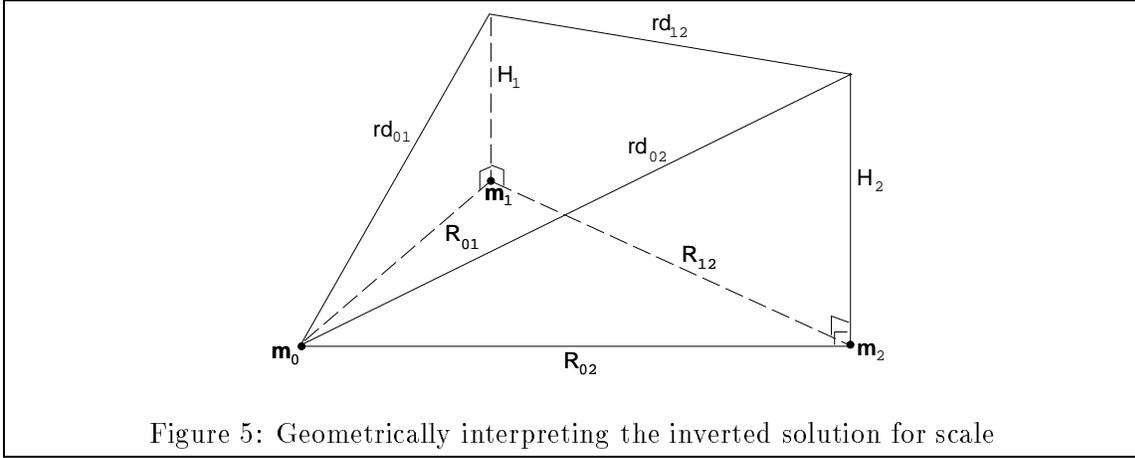
Figure 5: Geometrically interpreting the inverted solution for scale

$$H_2^2 + R_{02}^2 = (rd_{02})^2 \tag{31}$$
$$(H_1 - H_2)^2 + R_{12}^2 = (rd_{12})^2, \tag{32}$$

where $r = \frac{1}{s}$. Observe that $r = \frac{1}{s_1}$ and s_2 have similar forms (see Equation 27):

$$
\begin{aligned}
r &= \sqrt{\frac{a}{b - \sqrt{b^2 - ac}}} \\
&= \sqrt{\frac{b + \sqrt{b^2 - ac}}{c}} \ . \tag{33}
\end{aligned}
$$

To begin the derivation, Proposition 1 gives that $d_{01}^2 - (sR_{01})^2 \geq 0$ and $d_{02}^2 - (sR_{02})^2 \geq 0$, which implies we can set $h_1^2 = d_{01}^2 - (sR_{01})^2$ and $h_2^2 = d_{02}^2 - (sR_{02})^2$. Dividing through by s^2 gives Equations 30 and 31. As before, since s satisfies Equation 22 and, equivalently, Equation 21, we can substitute into Equation 21 with h_1^2 and h_2^2 to obtain

$$(h_1 - h_2)^2 = d_{12}^2 - s^2 R_{12}^2,$$

where the sign of h_2 relative to h_1 is 1 if $d_{01}^2 + d_{02}^2 - d_{12}^2 \geq s^2(R_{01}^2 + R_{02}^2 - R_{12}^2)$, and -1 otherwise. Dividing through by s^2 gives Equation 32, and so the derivation is completed.

Geometrically, Equation 30 forms a right triangle with sides H_1 and R_{01}, and hypotenuse rd_{01}. Analogously, Equations 31 and 32 imply right triangles as well. The interpretation is displayed in Fig. 4. Another way to see what is occurring geometrically is to note that the roles of image and model distances from Equations 18-20 are inverted in Equations 30-32. In effect, what is happening is that instead of scaling down the model triangle and projecting it orthographically onto the image triangle, the image triangle is being scaled up and projected orthographically onto the model triangle, that is, projected along parallel rays that are perpendicular to the model triangle. This interpretation is shown in Fig. 5 as a rotated version of Fig. 4.

4.3 Model triangle is parallel to the image plane

The two solutions for the scale factor are the same when $b^2 - ac = 0$, and here I demonstrate that geometrically this corresponds to the plane containing the three model points being parallel to the image plane. Before proving this, let us establish the existence of the solution for scale in this special case of $b^2 - ac = 0$. Looking at Equation 23,

$$b^2 - ac = 0 \implies b \pm \sqrt{b^2 - ac} = b$$
$$\implies s = \sqrt{\frac{b}{a}}$$

is a solution to the biquadratic since $a > 0$ and $b > 0$.

Appendix D shows that $b^2 - ac = 0$ exactly when $\phi = \pm\psi$ or $\phi = \pm\psi + \pi$ and $\frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}$. Using this result and Equations 24 and 26,

$$s = \sqrt{\frac{b}{a}} = \sqrt{\frac{\sqrt{c}}{\sqrt{a}}} = \sqrt{\frac{|d_{01} d_{02} \sin \phi|}{|R_{01} R_{02} \sin \psi|}} = \frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}} \tag{34}$$

$$\implies \quad h_1 = \sqrt{(sR_{01})^2 - d_{01}^2} = 0$$
$$h_2 = \sqrt{(sR_{02})^2 - d_{02}^2} = 0.$$

Thus $b^2 - ac = 0$ only if the model triangle is parallel to the image plane.

Conversely, if the model triangle is parallel to the image plane, it must be that $\phi = \psi$. Further, in this case $h_1 = h_2 = 0$, so that

$$s = \frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}, \tag{35}$$

which from Appendix D implies that $b^2 - ac = 0$.

Since the two solutions are the same, we know that $s_1 = s_2 = \frac{1}{r}$. Notice in Figs. 3 and 4 that the geometric interpretations for the two solutions for scale collapse to the same solution when $h_1 = h_2 = H_1 = H_2 = 0$ and $s = \frac{1}{r}$. As a result, when there is one solution for scale, there is also one solution for (h_1, h_2) and (H_1, H_2), albeit $(0, 0)$.

4.4 Model triangle is perpendicular to the image plane

The situation where the model triangle is perpendicular to the image plane is of interest since the projection is a line. Note, however, that the solution given earlier makes no exception for this case as long as the model triangle is not degenerate. As for what happens in this case, since the image triangle is a line, we know $\psi = 0 \implies c = 0 \implies$ Equation 23 becomes

$$s = \sqrt{\frac{b \pm \sqrt{b^2}}{a}} = 0, \sqrt{\frac{2b}{a}}. \tag{36}$$
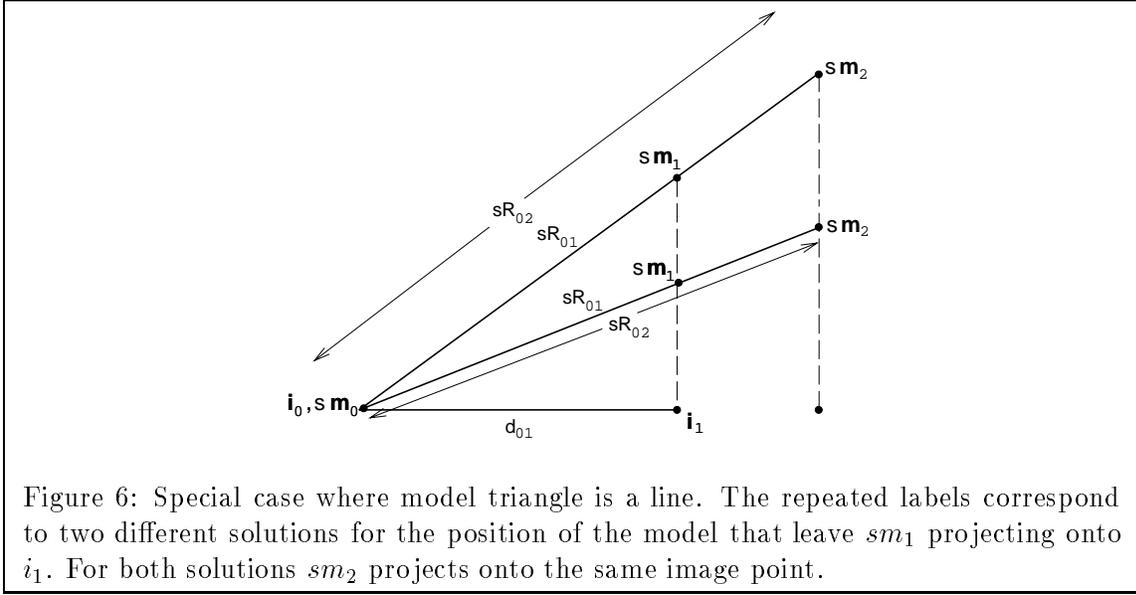
Figure 6: Special case where model triangle is a line. The repeated labels correspond to two different solutions for the position of the model that leave sm_1 projecting onto i_1. For both solutions sm_2 projects onto the same image point.

As shown above, of the two solutions for scale, the true one is $\sqrt{\frac{2b}{a}}$ and the inverted one is 0.

To see why the inverted solution is zero, recall that the solution can be viewed as scaling and projecting the image triangle onto the model triangle, using for scale $r = \frac{1}{s}$, which in this case does not exist. Since the image triangle is a line, graphically this amounts to trying to scale a line so that it can project as a triangle, which is not possible.

4.5 Model triangle is a line

This is the one case where the solution for the scale fails, and it fails because a, which is a measure of the area of the model triangle, is zero. Despite this fact, we can determine when a solution exists. First, we know that the image triangle must be a line as well. To see if this condition is enough, consider looking for a 3D rotation and scale that leaves sm_1 orthographically projecting onto i_1 as in Fig. 6. Observe that every such rotation and scale leaves sm_2 projecting onto the same point in the image. This means is that for a solution to exist, it must be that

$$\frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}.$$

Even when the image triangle is a line, this in general is not true. When it is true, there is an infinity of solutions corresponding to every scaled rotation that leaves sm_1 projecting onto i_1.

Another way to look at this situation is to notice that the model triangle being a line when using the true solution is analogous to the image triangle being a line when using the inverted solution, where the roles of the model and image triangles are reversed. As discussed in the previous section, the image triangle is a line when the model triangle is perpendicular to the

image plane. The analysis there reveals that for the inverted solution the scale factor r is undefined, which means that here the true solution for the scale factor s is undefined as well.

4.6 Summary

Our goal was to determine the three unknown parameters of the geometry displayed in Fig. 3, namely s, h_1, and h_2. The figure gave three constraints (Equations 18-20), from which a biquadratic in the scale factor s was derived. The biquadratic always has two positive solutions, and its coefficients, a, b, and c, are all non-negative. Of the two solutions, Section 4.1 showed that one and only one can satisfy the three constraints, and that solution is $s = s_2$ from Proposition 1 (see Equation 27). Given s, there are two pairs of valid assignments for h_1 and h_2. They correspond to reflecting the plane of the three matched model points about any plane parallel to the image; all planes parallel to image plane are equally-good. This proved that the solution for 3D pose exists and is unique up the reflective ambiguity.

In Section 4.2, Proposition 1 was used to infer the geometry that gives rise to the other solution to the biquadratic, namely $s = s_1$ (Equation 27). This solution, which is illustrated in Fig. 5, is obtained by inverting the roles of the model and image points in Fig. 3. The difference with the inverted solution is that the image points are being scaled and then orthographically projected onto the model points, instead of the reverse. The inverted geometry satisfies three constraints, Equations 30-32, that parallel the true constraints in function and form. Similarly, the expression for the scale factor of the inverted solution, $r = \frac{1}{s_1}$ (Equation 33), parallels the expression for the true scale factor, $s = s_2$.

Three special cases were discussed next, one in which the plane of the matched model points is parallel to the image plane (Section 4.3), one in which it is perpendicular to the image plane, or, equivalently, in which the matched image points are collinear (Section 4.4), and one in which the matched model points are collinear (Section 4.5). The first case is the one and only situation in which the two solutions collapse to the same one, and in this case $h_1 = h_2 = 0$. In addition, this situation is exactly where the two solutions to the biquadratic are the same; this is seen geometrically by looking at Figs. 3 and 4 with h_1, h_2, H_1, and H_2 all zero and $s = s_1 = s_2 = \frac{1}{r}$.

In the case where the matched image points are collinear, the solution for 3D pose is still valid. It is interesting to note, however, that the inverted solution for the scale factor does not exist. Yet the inverted solution for the scale does exist when the model points are collinear, but, in this case, the true solution does not. Section 4.5 determined that when the model points are collinear a solution for 3D pose may still exist, but if and only if a further constraint is satisfied. The section concludes by giving the constraint and describing how it arises geometrically.

5 Image Position of a Fourth Model Point

To compute the position in the image of a fourth model point, I first use the solution from the previous section to compute its 3D position in camera-centered coordinates. By so doing, I can project the camera-centered model point under weak-perspective and obtain the image position without having to calculate a model-to-image transformation. Let the image points be

15

$\vec{i}_0 = (x_0, y_0)$, $\vec{i}_1 = (x_1, y_1)$, and $\vec{i}_2 = (x_2, y_2)$. Given s, h_1, h_2, we can invert the projection to get the three model points:

$$
\begin{aligned}
\vec{m}_0 &= \frac{1}{s}(x_0, y_0, w) \\
\vec{m}_1 &= \frac{1}{s}(x_1, y_1, h_1 + w) \\
\vec{m}_2 &= \frac{1}{s}(x_2, y_2, h_2 + w),
\end{aligned}
$$

where w is an unknown offset in a direction normal to the image plane.

Given three 2D points, $\vec{q}_0$, $\vec{q}_1$, and $\vec{q}_2$, a fourth 2D point $\vec{q}_3$ can be uniquely represented by its "affine coordinates," (α, β), which are given by the equation

$$
\vec{q}_3 = \alpha(\vec{q}_1 - \vec{q}_0) + \beta(\vec{q}_2 - \vec{q}_0) + \vec{q}_0.
$$

Given three 3D points, $\vec{p}_0$, $\vec{p}_1$, and $\vec{p}_2$, this representation can be extended to uniquely represent any other 3D point $\vec{p}_3$ in terms of what I shall call its "extended affine coordinates," (α, β, γ), as follows:

$$
\vec{p}_3 = \alpha(\vec{p}_1 - \vec{p}_0) + \beta(\vec{p}_2 - \vec{p}_0) + \gamma(\vec{p}_1 - \vec{p}_0) \times (\vec{p}_2 - \vec{p}_0) + \vec{p}_0 \tag{37}
$$

Let

$$
\begin{aligned}
x_{01} &= x_1 - x_0, & y_{01} &= y_1 - y_0, \\
x_{02} &= x_2 - x_0, & y_{02} &= y_2 - y_0.
\end{aligned}
$$

Then, using the three model points with $\vec{p}_0 = \vec{m}_0$, $\vec{p}_1 = \vec{m}_1$, and $\vec{p}_2 = \vec{m}_2$,

$$
\vec{p}_1 - \vec{p}_0 = \frac{1}{s}(x_{01}, y_{01}, h_1) \tag{38}
$$

$$
\vec{p}_2 - \vec{p}_0 = \frac{1}{s}(x_{02}, y_{02}, h_2) \tag{39}
$$

$$
(\vec{p}_1 - \vec{p}_0) \times (\vec{p}_2 - \vec{p}_0) = \frac{1}{s^2}(y_{01}h_2 - y_{02}h_1, x_{02}h_1 - x_{01}h_2, x_{01}y_{02} - x_{02}y_{01}). \tag{40}
$$

Next, substitute Equations 38-40 into Equation 37 to get the three-space location of the fourth point:

$$
\begin{aligned}
\vec{m}_3 &= \frac{1}{s}\alpha(x_{01}, y_{01}, h_1) + \frac{1}{s}\beta(x_{02}, y_{02}, h_2) \\
&\quad + \gamma\frac{1}{s^2}(y_{01}h_2 - y_{02}h_1, -x_{01}h_2 + x_{02}h_1, x_{01}y_{02} - x_{02}y_{01}) + \frac{1}{s}(x_0, y_0, w) \\
&= \frac{1}{s}(\alpha x_{01} + \beta x_{02} + \gamma\frac{y_{01}h_2 - y_{02}h_1}{s} + x_0, \\
&\qquad \alpha y_{01} + \beta y_{02} + \gamma\frac{-x_{01}h_2 + x_{02}h_1}{s} + y_0, \\
&\qquad \alpha h_1 + \beta h_2 + \gamma\frac{x_{01}y_{02} - x_{02}y_{01}}{s} + w)
\end{aligned} \tag{41}
$$

To project, first apply the scale factor s:

$$\begin{aligned}
s\vec{m_3} \quad = \quad & (\alpha x_{01} + \beta x_{02} + \gamma\frac{y_{01}h_2 - y_{02}h_1}{s} + x_0, \\
& \alpha y_{01} + \beta y_{02} + \gamma\frac{-x_{01}h_2 + x_{02}h_1}{s} + y_0, \\
& \alpha h_1 + \beta h_2 + \gamma\frac{x_{01}y_{02} - x_{02}y_{01}}{s} + w)
\end{aligned} \tag{42}$$

Let Π represent an orthogonal projection along the z axis. Then project orthographically to get the image location of the fourth point:

$$\begin{aligned}
\Pi(s\vec{m_3}) \quad = \quad & (\alpha x_{01} + \beta x_{02} + \gamma(y_{01}H_2 - y_{02}H_1) + x_0, \\
& \alpha y_{01} + \beta y_{02} + \gamma(-x_{01}H_2 + x_{02}H_1) + y_0)
\end{aligned} \tag{43}$$

Notice that the unknown offset w has dropped out. This expression computes the image position of $\vec{p_3}$ from its extended affine coordinates, from the image points, and from H_1 and H_2, the altitudes in the weak-perspective geometry. There are no intermediate results about the actual 3D pose stored along the way, and as a result, this computation should be very efficient. Nonetheless, it should be kept in mind that H_1 and H_2 depend on the specific imaging geometry; that is, they depend on the pose of the model.

It may be worthwhile to observe that Equation 43, the expression for the fourth point, can be rewritten as a weighted sum of the three image points:

$$\begin{aligned}
\Pi(s\vec{m_3}) \quad = \quad & (\alpha x_{01} + \beta x_{02} + \gamma(y_{01}H_2 - y_{02}H_1) + x_0, \\
& \alpha y_{01} + \beta y_{02} + \gamma(-x_{01}H_2 + x_{02}H_1) + y_0) \\
\\
= \quad & (\alpha x_1 + \gamma H_2 y_1, \alpha y_1 - \gamma H_2 x_1) - (\alpha x_0 + \gamma H_2 y_0, \alpha y_0 - \gamma H_2 x_0) + \\
& (\beta x_2 - \gamma H_1 y_2, \beta y_2 + \gamma H_1 x_2) - (\beta x_0 - \gamma H_1 y_0, \beta y_0 + \gamma H_1 x_0) + \\
& (x_0, y_0) \\
\\
= \quad & \begin{bmatrix} 1-\alpha-\beta & \gamma(H_1-H_2) \\ -\gamma(H_1-H_2) & 1-\alpha-\beta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \\
& \begin{bmatrix} \alpha & \gamma H_2 \\ -\gamma H_2 & \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} \beta & -\gamma H_1 \\ \gamma H_1 & \beta \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}
\end{aligned}$$

Let $\mathbf{R}_\theta$ represent a 2D rotation matrix that rotates by an angle θ. Then

$$\Pi(s\vec{m_3}) = \delta_0 \mathbf{R}_{\theta_0}\vec{i_0} + \delta_1 \mathbf{R}_{\theta_1}\vec{i_1} + \delta_2 \mathbf{R}_{\theta_2}\vec{i_2}, \tag{44}$$

where

$$\delta_0 \quad = \quad \sqrt{(1-\alpha-\beta)^2 + (\gamma(H_1-H_2))^2} \tag{45}$$

$$\delta_1 \quad = \quad \sqrt{\alpha^2 + (\gamma H_2)^2} \tag{46}$$

$$\delta_2 \quad = \quad \sqrt{\beta^2 + (\gamma H_1)^2} \tag{47}$$

$$\cos\theta_0 = \frac{1-\alpha-\beta}{\delta_0} \quad \sin\theta_0 = \frac{-\gamma(H_1-H_2)}{\delta_0}$$
$$\cos\theta_1 = \frac{\alpha}{\delta_1} \quad \sin\theta_1 = \frac{-\gamma H_2}{\delta_1} \tag{48}$$
$$\cos\theta_2 = \frac{\beta}{\delta_2} \quad \sin\theta_2 = \frac{\gamma H_1}{\delta_2}$$

Thus, we can view the computation as a 2D rotation and scale of each image point separately followed by a sum of the three. It is important to keep in mind, however, that the rotations and scales themselves depend on the image points, because of H_1 and H_2.

When the model is planar, the form of Equation 44 facilitates understanding the effects of error in the image points. Error in the locations of the matched image points leads to uncertainty in the image location of the fourth model point. Suppose that the true locations of the matched image points are known to be within a few, say ϵ_i, pixels of their nominal locations, for $i = 0, 1, 2$. Let $\vec{i}_i$ and $\vec{c}_i$ be the true and nominal locations of an image point, for $i = 0, 1, 2$. Then, for some $\vec{e}_0$, $\vec{i}_0 = \vec{c}_0 + \vec{e}_0$, where $\parallel \vec{e}_0 \parallel = \epsilon_0$, and similarly for $\vec{i}_1$ and $\vec{i}_2$. Then

$$\begin{aligned}
\Pi(s\vec{m}_3) &= \delta_0\mathbf{R}_{\theta_0}\vec{i}_0 + \delta_1\mathbf{R}_{\theta_1}\vec{i}_1 + \delta_2\mathbf{R}_{\theta_2}\vec{i}_2 \\
&= (\delta_0\mathbf{R}_{\theta_0}\vec{c}_0 + \delta_1\mathbf{R}_{\theta_1}\vec{c}_1 + \delta_2\mathbf{R}_{\theta_2}\vec{c}_2) + (\delta_0\mathbf{R}_{\theta_0}\vec{e}_0 + \delta_1\mathbf{R}_{\theta_1}\vec{e}_1 + \delta_2\mathbf{R}_{\theta_2}\vec{e}_2)
\end{aligned}$$

When the fourth point is in the plane of the first three, $\gamma = 0$, so that the scales, δ_0, δ_1, and δ_2, and 2D rotations, $\mathbf{R}_{\theta_0}$, $\mathbf{R}_{\theta_1}$, and $\mathbf{R}_{\theta_2}$, are all constant (see Equations 45-48). This means that the first term in parentheses is just the nominal image location of the fourth model point. Since $\vec{e}_0$, $\vec{e}_1$, and $\vec{e}_2$ move around circles, the 2D rotations in the second term can be ignored. Further, since these error vectors move independently around their error circles, their radii simply sum together. Therefore, the region of possible locations of the fourth model point is bounded by a circle of radius $\delta_0\epsilon_0 + \delta_1\epsilon_1 + \delta_2\epsilon_2$ that is centered at the nominal point. By plugging $\gamma = 0$ into Equations 45-47, we get that

$$\delta_0 = |1-\alpha-\beta|, \qquad \delta_1 = |\alpha|, \qquad \delta_2 = |\beta|,$$

Assuming $\epsilon_0 = \epsilon_1 = \epsilon_2 = \epsilon$, this implies that the uncertainty in the image location of a fourth point is bounded by a circle with radius $(|1-\alpha-\beta| + |\alpha| + |\beta|)\epsilon$ and with its center at the nominal point, which repeats the result given earlier by Jacobs [19].

Although the non-planar case clearly is more complicated, since the scales and 2D rotations are no longer constant, Equation 44 may prove useful for obtaining bounds on the effects of error in this situation as well.

6 Stability of the 3D Pose Solution

In numerical computations, it is well-advised to determine whether a computation is *stable*, since, if not, it could produce inaccurate results. A computation is *unstable* if any roundoff error can propagate and magnify such that the true answer is significantly altered. The most common source of roundoff error is known as *catastrophic cancellation*, where two numbers of nearly equal magnitudes and opposite signs are summed. In fact, catastrophic cancellation is the only way a sudden loss of precision can occur [31]. Otherwise, in general precision can be lost by an accumulation of small errors over several operations.

In the 3D pose solution, there are a few subtractions of positive numbers to be wary of. In computing h_1 and h_2 from s (Equation 7), the values of h_1 and h_2 may have little precision if cancellation occurs in the radicands, in which case h_1 or h_2 will be small relative to its range of values. As discussed at the end of Section 4.1, h_1 or h_2 is zero when one of the sides of the model triangle that emanates from $\vec{m}_0$ lies parallel to the image plane.

The calculation of h_1 and h_2 can also be unstable if s is inaccurate. Looking at Equation 6 and recalling that a, b, and c are non-negative, catastrophic cancellation can only occur in the inner radicand. Even if it does, this is not a problem, since the result of the square root would be negligible when added to b.

Another way for s to become inaccurate is if the value of a, b, or c in Equation 6 is obtained with little precision. For a and c, Equations 9 and 11 show in parentheses one of the sides of a triangle being subtracted from the sum of the other two; therefore, catastrophic cancellation may occur when the triangle is nearly a line. Equation 10 shows that cancellation may occur in computing b if either the terms in parentheses or the total sum approaches zero relative to their ranges of values. From the law of cosines, the terms in parentheses are near zero when some angle of the model triangle is small. From Equation 25, the total sum, i.e., b, is small only if certain angles in the model and image triangles are small also. This says we should be careful of b in the same circumstances in which we are careful of a and c, namely, when the model or image points are nearly collinear.

To conclude, the parameters s, h_1 and h_2 (or s, H_1, and H_2) are prone to instability when the matched model or image points are almost collinear, and, additionally, H_1 or H_2 can be unstable when one of the vectors from $\vec{m}_0$ to $\vec{m}_1$ or $\vec{m}_2$ is nearly parallel to the image. In the latter case, the unstable H_1 or H_2 is close to zero. If only one of H_1 and H_2 is close to zero, then the instability can be avoided by re-ordering the matched points to make both H_1 and H_2 large. However, if this is done, the difference $H_1 - H_2$ will be close to zero and may be imprecise. If both H_1 and H_2 are almost zero, which means the model triangle is nearly parallel to the image, then re-ordering the matched points will not help.

Finally, it is worth observing that much of the instability in the pose solution occurs at places in which the problem is *ill conditioned*, that is, places where instability is inherent in the geometry. For instance, H_1 was said to be unstable when the vector from $\vec{m}_0$ to $\vec{m}_1$ is nearly parallel to the image. Geometrically, in this situation a small change in the position of $\vec{i}_1$ can cause a large change in the altitude H_1 (Fig. 2). For the same reason, recovering the altitude H_2 is unstable when the vector from $\vec{m}_0$ to $\vec{m}_2$ is nearly parallel to the image. This situation would be worse if both vectors emanating from $\vec{m}_0$ were parallel to the image. By a similar argument, it is intrinsically unstable to recover the pose when the model points are nearly collinear, due to there being an infinity of solutions when the model points are exactly collinear (Section 4.5).

This suggests that recognition systems like alignment and pose clustering should give special attention to situations where the model triangle is almost a line and where the model triangle being viewed straight on. These cases could be avoided by checking if the model points are nearly collinear or if the corresponding angles between the model and image points are very close. For the latter case, the suggestion does not apply if alignment is being used to recognize

planar models. This is because, if Equation 14 is used, error in H_1 or H_2 has no effect on the image locations of points in the plane, since for these points $\gamma = 0$.

7   Review of Previous Solutions

There have been several earlier solutions to the weak-perspective three-point problem, notably by Kanade and Kender [20], Cyganski and Orr ([7], [8]), Ullman ([28], [17]), Huttenlocher and Ullman ([16], [18], [29]), and Grimson, Huttenlocher, and Alter [12]. All the previous solutions compute the 3D pose by going through a 3D rigid transformation or a 2D affine transformation relating the model to the image. A 2D affine transform is a linear transform plus a translation, and it can be applied to any object lying in the plane. All but Ullman's and Grimson, Huttenlocher, and Alter's solutions compute an affine transformation between the three model and image points. Also, all but Kanade and Kender's solution compute a model-to-image rigid transformation, either via a rotation matrix or via Euler angles.

Not all of the solutions directly solve the weak-perspective three-point problem. The earliest solution, which was given by Kanade and Kender in 1983, applies Kanade's skewed-symmetry constraint to recover the 3D orientation of a symmetric, planar pattern [20]. More precisely, Kanade and Kender showed how to compute the 3D orientation of the plane containing a symmetric, planar pattern from a 2D affine transform between an image of the pattern and the pattern itself. To apply this result to the weak-perspective three-point problem, the three points can be used to construct a symmetric, planar pattern, and a 2D affine transform can be computed from two sets of three corresponding points. The solution was shown to exist and to give two solutions related by a reflective ambiguity, assuming that the determinant of the affine transform is positive.

The remaining methods all concentrate on computing the 3D rigid transform from the model to the image. In 1985, while presenting a system for recognizing planar objects, Cyganksi and Orr showed how to use higher-order moments to compute a 2D affine transform between planar regions ([7], [8]). Given the affine transform, they listed expressions for computing the 3D Euler angles from the 2D affine transform[1]. They did not, however, discuss how they derived the expressions.

The next method is the solution given by Ullman in 1986 [28], which appeared again in [17]. The paper included a proof that the solution for the scale factor is unique and the solution for the rotation matrix is unique up to an inherent two-way ambiguity. (This corresponds to the ambiguity in H_1 and H_2.) But Ullman did not show the solution exists. When it does exist, Ullman described a method for obtaining the rotation matrix and scale factor.

In 1988, Huttenlocher and Ullman gave another solution, and, in the process, gave the first complete proof that the solution both exists and is unique (up to the two-way ambiguity) ([16], [18], [29]). Like Kanade and Kender, and Cyganski and Orr, Huttenlocher and Ullman's solution relies on a 2D affine transform. The solution itself is based on algebraic constraints derived from rigidity, which are used to recover the elements of the scaled rotation matrix.

---

[1]The expressions that appear in [7] contain typesetting errors, but are listed correctly in [8].

The last solution, which was published this year, was developed by Grimson, Huttenlocher, and Alter for the purpose of analyzing the effects of image noise on error in transformation space [12]. Towards this end, the method facilitates computing how a small perturbation in each transformation parameter propagates to uncertainty ranges in the other parameters.

8 Presentation of Three Previous Solutions

The solutions discussed in the previous section differ significantly in how they compute the transformation, and, as a result, each one can provide different insights into solving related problems, such as error analysis in alignment-based recognition and pose clustering. It seems useful, then, to present the previous solutions in detail, so they conveniently can be referred to and compared.

The first method presented is Ullman's solution, which the first part of this paper extended. After that, I give Huttenlocher and Ullman's solution. Lastly, I present the method of Grimson, Huttenlocher, and Alter. I do not present Kanade and Kender's method nor Cyganski and Orr's, because Kanade and Kender did not directly solve the weak-perspective three-point problem, and Cyganski and Orr did not detail their solution.

It should be pointed out that the presentations here differ somewhat from the ones given by the original authors, but the ideas are the same. Basically, the presentations emphasize the steps that recover the 3D pose while being complete and concise. For more details, the reader is referred to the original versions in the references.

In the following presentations, we are looking for a rigid transform plus scale that aligns the model points to the image points. In all methods, we are free to move rigidly the three image points or the three model points wherever we wish, since this amounts to tacking on an additional transform before or after the aligning one. For example, this justifies the assumption made below that the plane of the model points is parallel to the image plane.

For consistency, the same notation as in Sections 3 and 4 is used in the proofs that follow: Let the model points be $\vec{m}_0$, $\vec{m}_1$, $\vec{m}_2$ and the image points be $\vec{i}_0$, $\vec{i}_1$, $\vec{i}_2$, with the respective distances between the points being R_{01}, R_{02}, and R_{12} for the model points, and d_{01}, d_{02}, and d_{12} for the image points.

8.1 Overview

This section provides an overview of the three methods.

Initially, all three methods compute a transformation that brings the model into image coordinates, such that the plane of the three matched model points is parallel to the image plane and such that $\vec{m}_0$ projects onto $\vec{i}_0$, which has been translated to the origin. The three methods then compute the out-of-plane rotation and scale that align the matched model and image points. In so doing, the methods all end up solving a biquadratic equation.

In Ullman's method, the model and image points are further transformed via rotations around the z axis to align $\vec{m}_1$ and $\vec{i}_1$ along the x axis. Then the 3D rotation matrix for rotating successively around the x and y axes is expressed in terms of Euler angles. This leads to a

series of three equations in three unknowns, which are solved to get a biquadratic in the scale factor. To get the elements of the rotation matrix, the solution for scale factor is substituted back into the original three equations.

Instead of further rotating the model and image points, Huttenlocher and Ullman compute an affine transform between them, which immediately gives the top-left sub-matrix of the scaled rotation matrix. Then by studying what happens to two equal-length vectors in the plane, a biquadratic is obtained. The scale factor and the remaining elements of the scaled rotation matrix are found using the algebraic constraints on the columns of a scaled rotation matrix.

Like Ullman did, Grimson, Huttenlocher, and Alter rotate the model further to align $\vec{m}_1$ and $\vec{i}_1$. The desired out-of-plane rotation is expressed in terms of two angles that give the rotation about two perpendicular axes in the plane. Next, Rodrigues' formula, which computes the 3D rotation of a point about some axis, is used to eliminate the scale factor and obtain two constraints on the two rotation angles. The two constraints are solved to get a biquadratic in the cosine of one of the angles. Its solution is substituted back to get the other angle and the scale factor, which can be used directly by Rodrigues' formula to transform any other model point.

As mentioned in the introduction, Ullman's solution is incomplete because it does not show which of the two solutions for the scale factor is correct; actually, the solution is completed by the result given in Section 4.1 of this paper. Similar to Ullman's method, Grimson, Huttenlocher, and Alter's solution has the same drawback of not showing which solution to its biquadratic is correct. Huttenlocher and Ullman, on the other hand, have no such problem because it turns out that one of the two solutions to their biquadratic is obviously not real, and so it immediately is discarded.

8.2   Ullman's method

This section gives Ullman's solution to the weak-perspective three-point problem. The main idea is first to transform the three model points to the image plane and then solve for the scale and out-of-plane rotation that align the transformed points.

Specifically, the model points first are rigidly transformed to put the three model points in the image plane with $\vec{m}_0$ at the origin of the image coordinate system and $\vec{m}_1 - \vec{m}_0$ aligned with the x axis. After rigidly transforming the model points, the resulting points can be represented by $(0, 0, 0)$, $(\bar{x}_1, 0, 0)$, and $(\bar{x}_2, \bar{y}_2, 0)$. Similarly, let the image points be rigid transformed to put $\vec{i}_0$ at the origin and $\vec{i}_1 - \vec{i}_0$ along the x axis, and let the resulting image points be $(0, 0, 0)$, $(x_1, 0, 0)$, and $(x_2, y_2, 0)$.

Next, we break the out-of-plane rotation into a rotation around the x axis by an angle θ followed by a rotation around the y axis by an angle ϕ, as pictured in Fig. 7. The corresponding rotation matrix is

$$
\mathbf{R} \;=\; \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}
$$
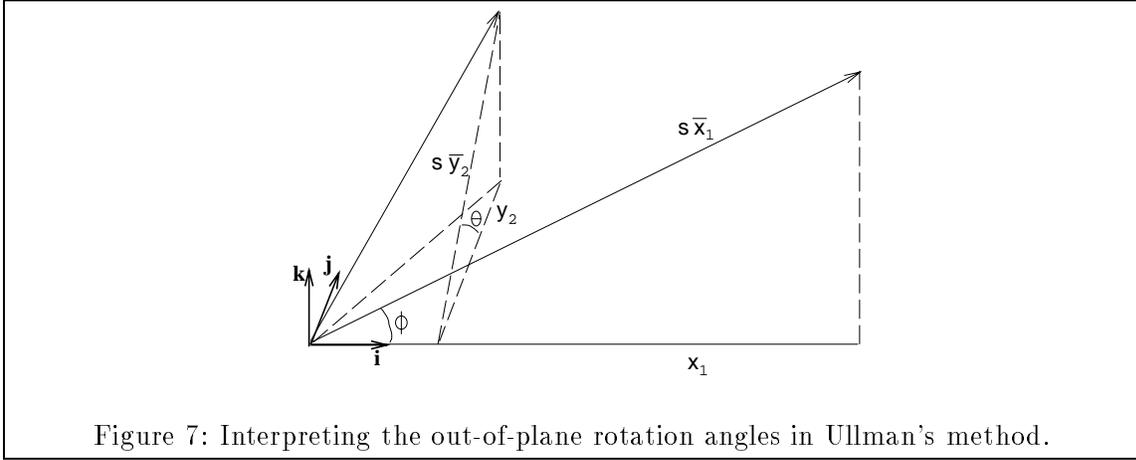
Figure 7: Interpreting the out-of-plane rotation angles in Ullman's method.

$$= \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & \sin\phi\cos\theta \\ 0 & \cos\theta & -\sin\theta \\ -\sin\phi & \cos\phi\sin\theta & \cos\phi\cos\theta \end{bmatrix} \tag{49}$$

After rotation and scale, $(0,0,0)$, $(\bar{x_1},0,0)$, and $(\bar{x_2},\bar{y_2},0)$ become $(0,0,0)$, $(x_1,0,z_1)$, and (x_2, y_2, z_2), respectively, where z_1 and z_2 are unknown. Thus, we need to find θ, ϕ, and s such that

$$s\mathbf{R}(\bar{x_1},0,0) = (x_1,0,z_1)$$
$$s\mathbf{R}(\bar{x_2},\bar{y_2},0) = (x_2,y_2,z_2)$$

Expanding the first two rows of $\mathbf{R}$ yields three equations in three unknowns:

$$s\bar{x_1}\cos\phi = x_1 \tag{50}$$
$$s\bar{y_2}\cos\theta = y_2 \tag{51}$$
$$s\bar{x_2}\cos\phi - s\bar{y_2}\sin\phi\sin\theta = x_2 \tag{52}$$

Fig. 7 gives a graphical interpretation of the first two equations. Substituting Equations 50 and 51 along with expressions for $\sin\phi$ and $\sin\theta$ into Equation 52 yields a biquadratic in the scale factor s:

$$as^4 - bs^2 + c = 0, \tag{53}$$

where

$$a = \bar{x_1}^2\bar{y_2}^2 \tag{54}$$
$$b = x_1^2(\bar{x_2}^2 + \bar{y_2}^2) + \bar{x_1}^2(x_2^2 + y_2^2) - 2x_1 x_2 \bar{x_1}\bar{x_2} \tag{55}$$
$$c = x_1^2 y_2^2 \tag{56}$$

The positive solutions for s are given by

$$s = \sqrt{\frac{b \pm \sqrt{b^2 - 4ac}}{2a}} \tag{57}$$

In general there can be one, two, or no solutions for s. Ullman makes no further attempt to determine when or if each solution arises, except to refer to a uniqueness proof he gives earlier in the paper. The uniqueness proof implies there can be at most one solution for s, but does not say which solution it is or whether it can be either one at different times.

Given s, the rotation matrix $\mathbf{R}$ is obtained using $\cos\phi = \frac{x_1}{s\bar{x_1}}$ and $\cos\theta = \frac{y_2}{s\bar{y_2}}$ in Equation 49. One difficulty with this is that we do not know the signs of $\sin\theta$ and $\sin\phi$; this leaves four possibilities for the pair $(\sin\theta, \sin\phi)$. In his uniqueness proof, Ullman points out that the inherent reflective ambiguity corresponds to multiplying simultaneously the elements r_{13}, r_{23}, r_{31}, and r_{32} of $\mathbf{R}$ by -1. In Equation 49, the signs of those elements also are inverted when both $\sin\theta$ and $\sin\phi$ are multiplied by -1, which, visually, corresponds to reflecting the model points about the image plane (Fig. 7). Still, we have no way to know which of the two pairs of solutions is correct. One way to proceed is to try both and see which solution pair aligns the points.

8.3   Huttenlocher and Ullman's method

First, assume the plane containing the model points is parallel to the image plane. Then subtract out $\vec{m}_0$ and $\vec{i}_0$ from the model and image points, respectively, to align them at the origin. Let the resulting model points be $(0,0,0)$, $(\bar{x_1}, \bar{y_1}, 0)$, and $(\bar{x_2}, \bar{y_2}, 0)$, and the resulting image points be $(0,0)$, (x_1, y_1), and (x_2, y_2). At this point, what is left is to compute the scaled rotation matrix that brings $(\bar{x_1}, \bar{y_1}, 0)$ and $(\bar{x_2}, \bar{y_2}, 0)$ to (x_1, y_1, z_1) and (x_2, y_2, z_2), respectively, where z_1 and z_2 are unknown. That is, we need

$$
\begin{aligned}
s\mathbf{R}(\bar{x_1}, \bar{y_1}, 0) &= (x_1, y_1, z_1) \\
s\mathbf{R}(\bar{x_2}, \bar{y_2}, 0) &= (x_2, y_2, z_2).
\end{aligned}
$$

Letting $l_{11} = sr_{11}$, $l_{12} = sr_{12}$, etc., and focusing on the first two rows of the rotation matrix, we get two sets of equations:

$$
\begin{aligned}
l_{11}\bar{x_1} + l_{12}\bar{y_1} &= x_1 \qquad\qquad &(58) \\
l_{11}\bar{x_2} + l_{12}\bar{y_2} &= x_2 \qquad\qquad &(59)
\end{aligned}
$$

$$
\begin{aligned}
l_{21}\bar{x_1} + l_{22}\bar{y_1} &= y_1 \qquad\qquad &(60) \\
l_{21}\bar{x_2} + l_{22}\bar{y_2} &= y_2, \qquad\qquad &(61)
\end{aligned}
$$

which give $\begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix}$, the top-left sub-matrix of the scaled rotation matrix. Note that this step fails if the determinent, $\bar{x_1}\bar{y_2} - \bar{x_2}\bar{y_1}$, equals zero.

Next, we make a digression to consider what happens to two orthogonal, equal-length vectors in the plane, $\vec{e_1}$ and $\vec{e_2}$. Since $\vec{e_1}$ and $\vec{e_2}$ are in the plane, we can apply the sub-matrix just computed to obtain the resulting vectors, $\vec{e_1}'$ and $\vec{e_2}'$:

$$
\vec{e_1}' = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} \vec{e_1}, \qquad \vec{e_2}' = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} \vec{e_2} \qquad\qquad (62)
$$
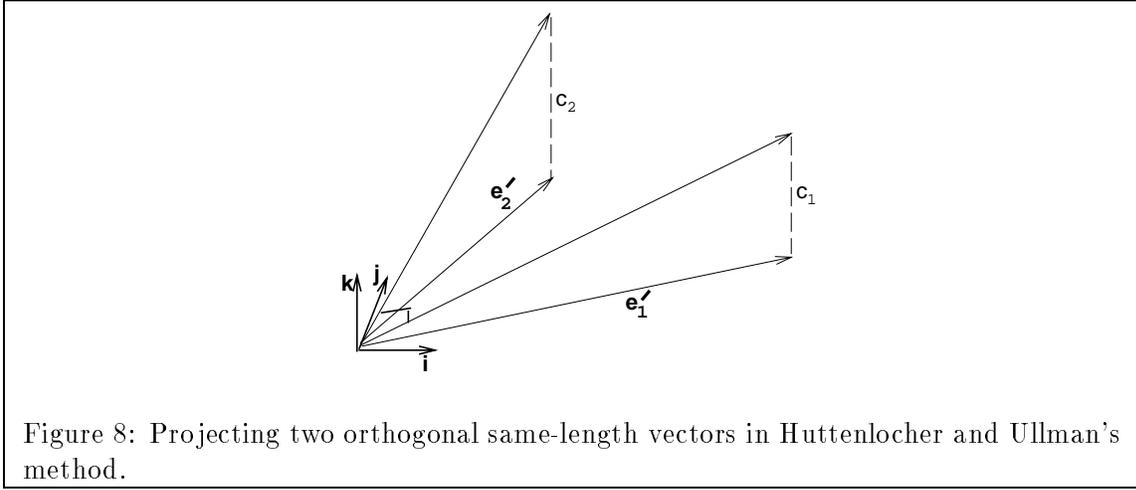
Figure 8: Projecting two orthogonal same-length vectors in Huttenlocher and Ullman's method.

When a model is transformed, $\vec{e_1}$ and $\vec{e_2}$ undergo a rigid transformation plus scale before projection. As shown in Fig. 8, after transformation these vectors become $\vec{e_1}' + c_1\hat{z}$ and $\vec{e_2}' + c_2\hat{z}$. Since a scaled, rigid transform preserves angles and ratios of lengths between vectors, and since $\vec{e_1} \cdot \vec{e_2} = 0$ and $\parallel \vec{e_1} \parallel = \parallel \vec{e_2} \parallel$, it must be that

$$(\vec{e_1}' + c_1\hat{z}) \cdot (\vec{e_2}' + c_2\hat{z}) = 0$$

$$\parallel \vec{e_1}' \parallel + c_1^2 = \parallel \vec{e_2}' \parallel + c_2^2.$$

These two equations simplify to

$$c_1 c_2 = k_1$$
$$c_1^2 - c_2^2 = k_2$$

where

$$k_1 = -\vec{e_1}' \cdot \vec{e_2}' \tag{63}$$

$$k_2 = \parallel \vec{e_2}' \parallel - \parallel \vec{e_1}' \parallel \tag{64}$$

Substituting for $c_2 = \frac{k_1}{c_1}$ in the second equation leads to a biquadratic in c_1:

$$c_1^4 - k_2 c_1^2 - k_1^2 = 0 \tag{65}$$

The general solution is

$$c_1 = \pm \sqrt{\frac{1}{2}\left(k_2 \pm \sqrt{k_2^2 + 4k_1^2}\right)} \, .$$

Conveniently, the inner discriminant always is greater than or equal to zero. Furthermore, since $4k_1^2 \geq 0$, the real solutions are given by

$$c_1 = \pm \sqrt{\frac{1}{2}\left(k_2 + \sqrt{k_2^2 + 4k_1^2}\right)} \, , \tag{66}$$

25

since otherwise the outer discriminant is less than zero.

These two solutions for c_1 give two corresponding solutions for c_2, which from Fig. 8 can be seen to correspond to a reflection about the image plane.

The solution for c_2 does not work when $c_1 = 0$. In this case,

$$c_2 = \pm\sqrt{-k_2} = \pm\sqrt{\| \vec{e_1}' \| - \| \vec{e_2}' \|} . \tag{67}$$

This gives two solutions for c_2, if it exists, which can be seen as follows. Since $c_1 = 0$, $\vec{e_1}$ ends up in the plane, so that that the length of $\vec{e_1}$ is just scaled down by s, whereas the length of $\vec{e_2}$ reduces both by being scaled down and by projection. Consequently, $\| \vec{e_2}' \| \leq \| \vec{e_1}' \|$, and, therefore, c_2 exists.

Given c_1 and c_2, we can recover two more elements of the scaled rotation matrix. Since $\vec{e_1}$ and $\vec{e_2}$ are in the plane, we know that $s\mathbf{R}\vec{e_1} = \vec{e_1}' + c_1\hat{z}$ and $s\mathbf{R}\vec{e_2} = \vec{e_2}' + c_2\hat{z}$. Focusing on the last row of the scaled rotation matrix, we get the two equations $l_{31} = c_1$ and $l_{32} = c_2$.

At this point, we have the first two columns of $s\mathbf{R}$, and, from the constraints on the columns of a rotation matrix, we can get the last column from the cross product of the first two. In total, this gives

$$s\mathbf{R} = \begin{bmatrix} l_{11} & l_{12} & \frac{1}{s}(c_2 l_{21} - c_1 l_{22}) \\ l_{21} & l_{22} & \frac{1}{s}(c_1 l_{12} - c_2 l_{11}) \\ c_1 & c_2 & \frac{1}{s}(l_{11} l_{22} - l_{12} l_{21}) \end{bmatrix} \tag{68}$$

Since the columns of a rotation matrix have unit length, we know

$$s = \sqrt{l_{11}^2 + l_{21}^2 + c_1^2} = \sqrt{l_{12}^2 + l_{22}^2 + c_2^2} . \tag{69}$$

Notice that the ambiguity in c_1 and c_2 inverts the signs of the appropriate elements of the rotation matrix as discussed in Section 8.2.

8.4 Grimson, Huttenlocher, and Alter's method

Grimson et al. gave another solution to the weak-perspective three point problem in order to get a handle on how small perturbations affect the individual transformation parameters.

To start, assume the plane containing the model points is parallel to the image plane. Next, rigidly transform the model points so that $\vec{m}_0$ projects to $\vec{i}_0$ and $\vec{m}_1 - \vec{m}_0$ projects along $\vec{i}_1 - \vec{i}_0$. Let Π represent an orthogonal projection along the z axis, and in general let $\vec{v}^\perp$ be the 2D vector rotated ninety degrees clockwise from the 2D vector $\vec{v}$. Then the translation is $\vec{i}_0 - \Pi\vec{m}_0$, and the rotation is about $\hat{z}$ by an angle ψ given by

$$\cos\psi = \hat{m}_{01} \cdot \hat{i}_{01}, \qquad \sin\psi = -\hat{m}_{01} \cdot \hat{i}_{01}^\perp.$$

(see Fig. 9).

At this point, assign $\vec{m}_{01} = \vec{m}_1 - \vec{m}_0$, $\vec{m}_{02} = \vec{m}_2 - \vec{m}_0$, $\vec{i}_{01} = \vec{i}_1 - \vec{i}_0$, and $\vec{i}_{02} = \vec{m}_2 - \vec{m}_0$. Also, consider the out-of-plane rotation to be a rotation about $\hat{i}_{01}$ by some angle θ followed by a rotation about $\hat{i}_{01}^\perp$ by some angle ϕ. Let us compute where the vectors $\hat{i}_{01}$ and $\hat{i}_{01}^\perp$ project to
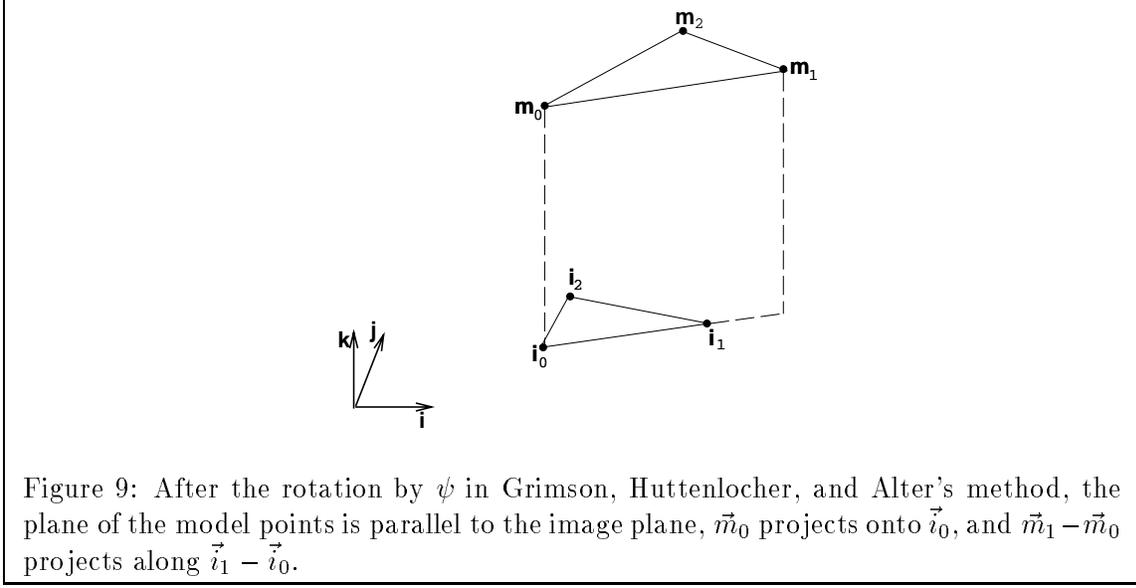
Figure 9: After the rotation by ψ in Grimson, Huttenlocher, and Alter's method, the plane of the model points is parallel to the image plane, $\vec{m}_0$ projects onto $\vec{i}_0$, and $\vec{m}_1 - \vec{m}_0$ projects along $\vec{i}_1 - \vec{i}_0$.

after the two rotations and scale. To do this, we use Rodrigues' formula: Let $\mathbf{R}_{\hat{v},\tau}\vec{p}$ represent a rotation of a point $\vec{p}$ about a direction $\hat{v}$ by an angle τ. Rodrigues' formula is

$$\mathbf{R}_{\hat{v},\tau}\vec{p} = \cos\tau\vec{p} + (1 - \cos\tau)(\hat{v}\cdot\vec{p})\hat{v} + \sin\tau(\hat{v}\times\vec{p}). \tag{70}$$

Using the formula, we can compute

$$\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}\hat{i}_{01} = \cos\phi\hat{i}_{01} - \sin\phi\hat{z} \tag{71}$$

$$\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}\hat{i}_{01}^{\perp} = \sin\theta\sin\phi\hat{i}_{01} + \cos\theta\hat{i}_{01}^{\perp} + \sin\theta\cos\phi\hat{z}.$$

Initially, $\vec{m}_{01}$ was rotated about $\hat{z}$ to align it with $\vec{i}_{01}$. In order for the scaled orthographic projection of $\vec{m}_{01}$ to align with $\vec{i}_{01}$, Equation 71 implies that

$$s = \frac{\|\vec{i}_{01}\|}{\|\vec{m}_{01}\|}\frac{1}{\cos\phi}$$

$$= \frac{d_{01}}{R_{01}}\frac{1}{\cos\phi}. \tag{72}$$

Then

$$s\Pi\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}\hat{i}_{01} = \frac{d_{01}}{R_{01}}\hat{i}_{01} \tag{73}$$

$$s\Pi\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}\hat{i}_{01}^{\perp} = \frac{d_{01}}{R_{01}}\frac{1}{\cos\phi}(\sin\theta\sin\phi\hat{i}_{01} + \cos\theta\hat{i}_{01}^{\perp}) \tag{74}$$

Next, we use the expressions in Equations 73 and 74 to constrain θ and ϕ such that $\vec{m}_{02}$ projects along $\vec{i}_{02}$. When we aligned $\vec{m}_{01}$ and $\vec{i}_{01}$, $\vec{m}_{02}$ rotated to $\mathbf{R}_{\hat{z},\psi}\vec{m}_{02}$. Since $\vec{m}_{02}$ has no $\hat{z}$ component (by assumption), we can represent $\mathbf{R}_{\hat{z},\psi}\vec{m}_{02}$ by

$$R_{02}\cos\xi\hat{i}_{01} + R_{02}\sin\xi\hat{i}_{01}^{\perp},$$

where ξ is a known angle. Consequently, the transformed, projected, and scaled $\vec{m}_{02}$, which must equal $\vec{i}_{02}$, is

$$s\Pi\mathbf{R}_{\widehat{i}_{01}^{\perp},\phi}\mathbf{R}_{\widehat{i}_{01},\theta}(R_{02}\cos\xi\widehat{i}_{01} + R_{02}\sin\xi\widehat{i}_{01}^{\perp})$$

$$= R_{02}\cos\xi(s\Pi\mathbf{R}_{\widehat{i}_{01}^{\perp},\phi}\mathbf{R}_{\widehat{i}_{01},\theta}\widehat{i}_{01}) + R_{02}\sin\xi(s\Pi\mathbf{R}_{\widehat{i}_{01}^{\perp},\phi}\mathbf{R}_{\widehat{i}_{01},\theta}\widehat{i}_{01}^{\perp})$$

$$= R_{02}\cos\xi\left(\frac{d_{01}}{R_{01}}\widehat{i}_{01}\right) + R_{02}\sin\xi\left(\frac{d_{01}}{R_{01}}\frac{1}{\cos\phi}(\sin\theta\sin\phi\widehat{i}_{01} + \cos\theta\widehat{i}_{01}^{\perp})\right)$$

$$= \frac{d_{01}}{\cos\phi}\frac{R_{02}}{R_{01}}(\cos\xi\cos\phi + \sin\xi\sin\phi\sin\theta)\widehat{i}_{01} + \frac{d_{01}}{\cos\phi}\frac{R_{02}}{R_{01}}(\sin\xi\cos\theta)\widehat{i}_{01}^{\perp}.$$

Similar to $\mathbf{R}_{\widehat{z},\psi}\vec{m}_{02}$, we can represent $\vec{i}_{02}$ as

$$\vec{i}_{02} = d_{02}\cos\omega\widehat{i}_{01} + d_{02}\sin\omega\widehat{i}_{01}^{\perp},$$

where ω is known. By equating terms we get

$$\frac{d_{01}}{d_{02}}\frac{R_{02}}{R_{01}}(\cos\xi\cos\phi + \sin\xi\sin\phi\sin\theta) = \cos\phi\cos\omega \tag{75}$$

$$\frac{d_{01}}{d_{02}}\frac{R_{02}}{R_{01}}(\sin\xi\cos\theta) = \cos\phi\sin\omega. \tag{76}$$

These two equations can be solved to get a biquadratic in $\cos\phi$:

$$\sin^2\omega\cos^4\phi - (t^2 + 1 - 2t\cos\omega\cos\xi)\cos^2\phi + t^2\sin^2\xi = 0, \tag{77}$$

where

$$t = \frac{R_{02}d_{01}}{R_{01}d_{02}}. \tag{78}$$

Since $\mathbf{R}_{\widehat{z},\psi}\vec{m}_{01}$ is aligned with $\vec{i}_{01}$, we need $\cos\phi$ to be positive so that $\vec{m}_{01}$ projects in the same direction as $\vec{i}_{01}$. The positive solutions are given by

$$\cos\phi = \frac{1}{|\sin\omega|}\sqrt{\nu \pm \sqrt{\nu^2 - t^2\sin^2\omega\sin^2\xi}} \tag{79}$$

with

$$\nu = \frac{1}{2}(1 + t^2 - 2t\cos\omega\cos\xi).$$

This equation gives up to two solutions, but Grimson et al. make no further attempt to show which solutions exists when, except to say the equation gives real solutions only if $\nu \geq 0$ or

$$\cos\omega\cos\xi \leq \frac{1 + t^2}{2t}. \tag{80}$$

Given ϕ, Equations 75 and 76 provide θ:

$$\cos\theta \quad = \quad \frac{\sin\omega\cos\phi}{t\sin\xi} \tag{81}$$

$$\sin\theta \quad = \quad \frac{\cos\phi(\cos\omega - t\cos\xi)}{t\sin\xi\sin\phi} \tag{82}$$

Given any model point $\vec{m}$, we can use the computed angles along with Rodrigues' formula to find its image location. In particular, once $\vec{m}_0$ and $\vec{i}_0$ have been subtracted out, only the scale and 3D rotation are left. The scale is given by Equation 72, and, as shown above, the rotation is

$$\mathbf{R}_{\widehat{i_{01}^\perp},\phi}\mathbf{R}_{\widehat{i_{01}},\theta}\mathbf{R}_{\widehat{z},\psi}. \tag{83}$$

As with Ullman's method (Section 8.2), we do not know the signs of $\sin\theta$ and $\sin\phi$, but only that inverting both signs simultaneously corresponds to the reflective ambiguity.

8.5 Summary of the three computations

Here I summarize how each method can be used to compute 3D pose from three corresponding points. To begin, transform the model and image points so that (1) the model points lie in the image plane, (2) $\vec{m}_0$ and $\vec{i}_0$ are at the origin of the image coordinate system, and (3) $\vec{m}_1 - \vec{m}_0$ and $\vec{i}_1 - \vec{i}_0$ lie along the x axis. Then use one of the three methods to compute the scale factor and out-of-plane rotation, as follows:

- Ullman's method

  1. Use Equations 54-56 to get a, b, and c.
  2. Substitute a, b, and c into Equation 57 to get s.
  3. Calculate $\cos\phi = \frac{x_1}{sx_1}$ and $\cos\theta = \frac{y_2}{s\bar{y}_2}$.
  4. Calculate $\sin\phi = \sqrt{1 - \cos^2\phi}$ and $\sin\theta = \sqrt{1 - \cos^2\theta}$.
  5. Construct the rotation matrix $\mathbf{R}$ using Equation 49.

- Huttenlocher and Ullman's method

  1. Solve Equations 58 and 59 for l_{11} and l_{12}, and Equations 60 and 61 for l_{21} and l_{22}.
  2. Let $\vec{e_1} = (0,1)$ and $\vec{e_2} = (1,0)$. (Any orthogonal, equal-length vectors can be used.)
  3. Use Equation 62 to get $\vec{e_1}'$ and $\vec{e_2}'$.
  4. Substitute $\vec{e_1}'$ and $\vec{e_2}'$ into Equations 63 and 64 to get k_1 and k_2.
  5. Substitute k_1 and k_2 into Equation 66 to get c_1.
  6. If $c_1 \neq 0$, calculate $c_2 = \frac{k_1}{c_1}$. Otherwise get c_2 from Equation 67.
  7. Use Equation 69 to get s.

8. Use Equation 68 to get $s\mathbf{R}$. Divide through by s if $\mathbf{R}$ is desired instead of $s\mathbf{R}$.

- Grimson, Huttenlocher, and Alter's method

  1. From the model points, compute R_{01}, R_{02} and ξ, and, from the image points, compute d_{01}, d_{02}, and ω.
  2. Use Equation 78 to get t.
  3. Use Equation 79 to get $\cos\phi$.
  4. Use Equation 72 to get s.
  5. Calculate $\sin\phi = \sqrt{1 - \cos^2\phi}$.
  6. Use Equations 81 and 82 to get $\cos\theta$ and $\sin\theta$.
  7. To transform any point $\vec{p}$, substitute $\cos\phi$, $\sin\phi$, $\cos\theta$, $\sin\theta$, and $\vec{p}$ into Rodrigues' formula, Equation 70, to get $\mathbf{R}\vec{p} = \mathbf{R}_{\widehat{i_{01}^{\perp}},\phi}\mathbf{R}_{\widehat{i_{01}},\theta}\vec{p}$.

9   Stability and Comparison of Three Previous Solutions

For computing 3D pose, it is desirable to know how the solutions compare in terms of stability. To address this issue, let us examine how susceptible the solutions are to catastrophic cancellation [31]. For ease of reference, I will indicate which steps in the pose computation summaries of Section 8.5 may be unstable.

Ullman's solution computes s in the same way as this paper does, and, as a result, is unstable at the same places (see Section 6). For instance, precision may be lost if the model or image points are nearly collinear when computing the coefficients, a, b, and c, of the biquadratic. (In Section 8.5, this is step 1 of Ullman's solution.) Looking for a moment at Ullman's computation of a and c, it may appear that the computation is stable since there is no addition in Equation 54 or 56. In actuality, instability is hidden in the initial transformation that aligns the model and image with the x axis.

Given s, Ullman computes the cosines of the angles θ and ϕ and then implicitly uses $\sqrt{1 - \cos^2\theta}$ and $\sqrt{1 - \cos^2\phi}$ to get their sines. (This is step 4 of Ullman's solution.) Either sine could be inaccurate, however, if $\cos\theta$ or $\cos\phi$ is very close to one. Fig. 7 shows that when this happens one of the vectors emanating from $\vec{m}_0$ is nearly parallel to the image plane. When the rotation matrix $\mathbf{R}$ is computed, inaccuracy in the sines affects the elements r_{12}, r_{13}, and r_{23} (see Equation 49). Since r_{12} is affected, when the solution is used to transform an unmatched model point, the instability can propagate to points that lie in the plane containing the three matched model points, which is not true for the solution in this paper (Section 6).

For Huttenlocher and Ullman's method in Section 8.5, catastrophic cancellations may occur in step 1 when l_{11}, l_{12}, l_{21}, and l_{22} are computed, in step 4 when k_1 and k_2 are computed, and in step 8 when $s\mathbf{R}$ is computed. Instability in step 4 can affect c_1 and c_2 in Equation 49: If c_1 is near zero, then k_1 and k_2 in Equation 66 also must be near zero. From Equations 63 and 64, k_1 and k_2 are computed with additions, and so cancellation can occur if they are small. Similarly, if c_2 is near zero, k_1 must be small as well, and so again cancellation can occur. From Fig. 8, c_1

or c_2 is nearly zero when one of the vectors emanating from the origin ($\vec{m}_0$) is nearly parallel to the image plane.

When l_{11}, l_{12}, l_{21}, and l_{22} are computed (Equations 58-61), the results will be inaccurate if the determinant, $\bar{x_1}\bar{y_2} - \bar{x_2}\bar{y_1}$, is close to zero, which happens exactly when the model points are almost collinear. In addition, if any of l_{11}, l_{12}, l_{21}, or l_{22} is almost zero, then cancellation can occur in computing it. There are many pairs of model and image triples that can make one or more of l_{11}, l_{12}, l_{21}, l_{22} close to zero (e.g., $l_{12} \approx 0$ whenever $x_1 \approx \bar{x_1}$ and $x_2 \approx \bar{x_2}$, independent of y_1, y_2, $\bar{y_1}$, and $\bar{y_2}$). Furthermore, in step 8, the additions in computing sr_{13} and sr_{23} can also contribute to instability (see Equation 68). Note, however, that the image triangle being nearly collinear does not necessarily make the computation unstable.

In Grimson et al.'s solution, instability may arise in step 5 if $\cos\phi$ is almost 1, in step 6 if t is 1 and ω is close to ξ, and in step 7 if $\cos\phi$ is near 1 or $\cos\theta$ is near 1. As with the solution in this paper, these situations occur when the model or image points are nearly collinear or when one of the sides of the model triangle that emanates from $\vec{m}_0$ is nearly parallel to the image plane. Like Ullman's method and Huttenlocher and Ullman's methods, however, instability can propagate to points inside the plane of the matched model points (in step 7).

In summary, each of the three previous solutions spreads instability in the pose solution to points in the plane of the three matched model points; however, the solution in this paper does not. Furthermore, the situations in which instability can arise are the same for Ullman's method, the method of Grimson et al., and the solution in this paper. Specifically, these situations are when one of the vectors from $\vec{m}_0$ is parallel to the image, when the model points are nearly collinear, and when the image points are nearly collinear. Huttenlocher and Ullman's method is unstable in the first two situations as well, which is expected since in these situations the problem is ill conditioned (Section 6). In addition, Huttenlocher and Ullman's method can be unstable in many cases where the other methods are not, but may be more stable in the case that the image points are nearly collinear.

10 Conclusion

The weak-perspective three-point problem is fundamental to many approaches to model-based recognition. In this paper, I illustrated the underlying geometry, and then used it to derive a new solution to the problem and to explain the various special cases that can arise. In particular, the times when there are zero, one, and two solutions are described graphically.

The new solution is based on the distances between the matched model and image points and is used to recover the three-space locations of the model points in image coordinates. From the recovered locations, a direct expression for the image location of a fourth model point is obtained. In contrast, earlier solutions computed an initial transformation that brought the model into image coordinates, and then computed an additional transformation to align the matched model points to their corresponding image points. As a result, the solution given here should be easier to use, and, for recognition systems that repeat the computation of the model pose many times, should be more efficient.

Another difference with the method presented here is that it makes evident the symmetry

of the solution with respect to the ordering of the model and image points. Previous methods that are based on the coordinates of the points after some initial transformations make this symmetry unclear.

Furthermore, this paper provides stability analyses for both the new and past solutions, none of which had been analyzed for stability previously. Each computation is examined for places where precision may be lost. From these places, the geometries that give rise to instability are inferred. These geometries are used to distinguish instabilities that arise in situations where the problem is ill conditioned, that is, situations where instability is inherent, from ones that are due to the particular computation.

In giving another solution, this paper revisits Ullman's original biquadratic equation for the scale factor, but, in addition, goes on to interpret both solutions to the equation, and to prove which one is correct. The false solution is shown to correspond to inverting the roles of the model and image points.

Lastly, the new solution is accompanied by a proof that the solution exists and is unique. Of the previous methods, only Huttenlocher and Ullman's demonstrates this as well, and was the first to do so. Such proofs may be useful for gaining insights into related problems as well as the problem itself. Even so, since existence and uniqueness have been established, all the solutions are valid, and should all be considered when a related problem needs to be solved.

Acknowledgments

A   Rigid Transform between 3 Corresponding 3D Points

This appendix computes a rigid transform between two sets of three corresponding points using right-handed coordinate systems built separately on each set of three points. A right-handed system is determined by an origin point, $\vec{o}$, and three perpendicular unit vectors, $(\hat{u}, \hat{v}, \hat{w})$. Given three points in space, $\vec{p}_0$, $\vec{p}_1$, $\vec{p}_2$, we can construct a right-handed system as follows: Let $\vec{p}_{01} = \vec{p}_1 - \vec{p}_0$ and $\vec{p}_{02} = \vec{p}_2 - \vec{p}_0$. Then let

$$
\begin{aligned}
\vec{o} &= \vec{p}_0 \\
\vec{u} &= \vec{p}_{01} \\
\vec{v} &= \vec{p}_{02} - (\vec{p}_{02} \cdot \hat{p}_{01})\hat{p}_{01} \\
\vec{w} &= \vec{u} \times \vec{v}
\end{aligned}
$$

Let $(\vec{o}_1; \hat{u}_1, \hat{v}_1, \hat{w}_1)$ and $(\vec{o}_2; \hat{u}_2, \hat{v}_2, \hat{w}_2)$ be the coordinate systems so defined for the original and camera-centered points, respectively.

Given a coordinate system $(\vec{o}; \hat{u}, \hat{v}, \hat{w})$, a rigid transformation that takes a point in world coordinates to a point in that coordinate system is given by $(\mathbf{R}, \vec{t})$, where

$$
\mathbf{R} = [\hat{u} \quad \hat{v} \quad \hat{w}], \qquad \vec{t} = \vec{o}
$$

32

(see for example [6]); the transformed $\vec{p}$ is $\mathbf{R}\vec{p} + \vec{t}$. Then we can bring a point $\vec{p}$ from the original system to the world and then to the camera-centered system using

$$\mathbf{R_2}\left(\mathbf{R_1}^T(\vec{p} - \vec{t_1})\right) + \vec{t_2} = \mathbf{R_2}\mathbf{R_1}^T\vec{p} + \vec{t_2} - \mathbf{R_2}\mathbf{R_1}^T\vec{t_1}$$

where

$$\mathbf{R_1} = [\widehat{u}_1 \quad \widehat{v}_1 \quad \widehat{w}_1], \qquad \vec{t_1} = \vec{o}_1$$
$$\mathbf{R_2} = [\widehat{u}_2 \quad \widehat{v}_2 \quad \widehat{w}_2], \qquad \vec{t_2} = \vec{o}_2.$$

Consequently a rigid transformation $(\mathbf{R}, \vec{t})$ that aligns the two coordinate systems is

$$\mathbf{R} = \mathbf{R_2}\mathbf{R_1}^T, \qquad \vec{t} = \vec{t_2} - \mathbf{R_2}\mathbf{R_1}^T\vec{t_1}. \tag{84}$$

B Biquadratic for the Scale Factor

This appendix shows

$$4(s^2 R_{01}^2 - d_{01}^2)(s^2 R_{02}^2 - d_{02}^2) = \left(s^2(R_{12}^2 - R_{01}^2 - R_{02}^2) - (d_{12}^2 - d_{01}^2 - d_{02}^2)\right)^2 \tag{85}$$

is equivalent to a biquadratic in s.

Expanding Equation 85,

$$4\left(s^4 R_{01}^2 R_{02}^2 - s^2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2) + d_{01}^2 d_{02}^2\right) =$$
$$s^4(R_{01}^2 + R_{02}^2 - R_{12}^2)^2 - 2s^2(R_{01}^2 + R_{02}^2 - R_{12}^2)(d_{01}^2 + d_{02}^2 - d_{12}^2)$$
$$+ (d_{01}^2 + d_{02}^2 - d_{12}^2)^2$$

$$s^4\left(4R_{01}^2 R_{02}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)^2\right)$$
$$-2s^2\left(2R_{01}^2 d_{02}^2 + 2R_{02}^2 d_{01}^2 - (R_{01}^2 + R_{02}^2 - R_{12})(d_{01}^2 + d_{02}^2 - d_{12}^2)\right)$$
$$+ \left(4d_{01}^2 d_{02}^2 - (d_{01}^2 + d_{02}^2 - d_{12}^2)^2\right) = 0$$

$$as^4 - 2bs^2 + c = 0,$$

where

$$a = 4R_{01}^2 R_{02}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)^2$$
$$b = 2R_{01}^2 d_{02}^2 + 2R_{02}^2 d_{01}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)(d_{01}^2 + d_{02}^2 - d_{12}^2)$$
$$c = 4d_{01}^2 d_{02}^2 - (d_{01}^2 + d_{02}^2 - d_{12}^2)^2.$$

C Two Solutions for Scale

This appendix proves Proposition 1. The proof uses the following lemma:

Lemma :   Let f be either $\left(\frac{d_{01}}{R_{01}}\right)^2$ or $\left(\frac{d_{02}}{R_{02}}\right)^2$. Then

$$af^2 - 2bf + c \leq 0. \tag{86}$$

Proof:

$$af^2 - 2bf + c$$

$$
\begin{aligned}
= \quad & 4(R_{01}R_{02}\sin\phi)^2 f^2 - \\
& 2\left(2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi)\right)f + \\
& 4(d_{01}d_{02}\sin\psi)^2, \qquad \text{from Equations 24, 25, and 26}
\end{aligned}
$$

$$
\begin{aligned}
= \quad & 4\left(R_{01}^2 R_{02}^2(1 - \cos^2\phi)f^2 - \right.\\
& (R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi)f + \\
& \left. d_{01}^2 d_{02}^2(1 - \cos^2\psi)\right)
\end{aligned} \tag{87}
$$

Suppose that $f = \left(\frac{d_{01}}{R_{01}}\right)^2$. Then 87 becomes

$$4\left(-\frac{R_{02}^2 d_{01}^4}{R_{01}^2}\cos^2\phi + 2\frac{R_{02}d_{01}^3 d_{02}}{R_{01}}\cos\phi\cos\psi - d_{01}^2 d_{02}^2\cos^2\psi\right)$$

$$= \quad -4R_{02}^2 d_{01}^2\left(\frac{d_{01}}{R_{01}}\cos\phi - \frac{d_{02}}{R_{02}}\cos\psi\right)^2$$

Suppose instead that $f = \left(\frac{d_{02}}{R_{02}}\right)^2$. Then 87 becomes

$$4\left(-\frac{R_{01}^2 d_{02}^4}{R_{02}^2}\cos^2\phi + 2\frac{R_{01}d_{02}^3 d_{01}}{R_{02}}\cos\phi\cos\psi - d_{01}^2 d_{02}^2\cos^2\psi\right)$$

$$= \quad -4R_{01}^2 d_{02}^2\left(\frac{d_{02}}{R_{02}}\cos\phi - \frac{d_{01}}{R_{01}}\cos\psi\right)^2$$

Either way, $af^2 - 2bf + c \leq 0$.
□

Proposition 1:   Let

$$s_1 = \sqrt{\frac{b - \sqrt{b^2 - ac}}{a}}, \qquad s_2 = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}}.$$

Then

$$s_1 \leq \frac{d_{01}}{R_{01}}, \frac{d_{02}}{R_{02}} \leq s_2.$$

Proof: Starting from the result of the lemma,

$$af^2 - 2bf + c \le 0$$

$$\frac{1}{a}\left((af - b)^2 - (b^2 - ac)\right) \le 0$$

$$(af - b)^2 \le b^2 - ac, \quad \text{since } a > 0$$

$$|af - b| \le \sqrt{b^2 - ac}$$

$$-(af - b) \le \sqrt{b^2 - ac} \qquad \text{and} \qquad af - b \le \sqrt{b^2 - ac}$$

$$f \ge \frac{b - \sqrt{b^2 - ac}}{a} \qquad \text{and} \qquad f \le \frac{b + \sqrt{b^2 - ac}}{a}$$

$$s_1^2 \le f \le s_2^2$$

$$s_1 \le \frac{d_{01}}{R_{01}}, \frac{d_{02}}{R_{02}} \le s_2$$

□

D   One Solution for Scale

In the "one solution" case, we wish to know when and if $b^2 - ac = 0$ holds. Using the result of Appendix F, this means that

$$4(R_{01}d_{02})^4 \left(t^2 - 2\cos(\phi + \psi)t + 1\right) \left(t^2 - 2\cos(\phi - \psi)t + 1\right) = 0.$$

For this to hold, either

$$t^2 - 2\cos(\phi + \psi)t + 1 = 0 \qquad \text{or} \qquad t^2 - 2\cos(\phi - \psi)t + 1 = 0.$$

Solving for t gives

$$t = \cos(\phi + \psi) \pm i \sin(\phi + \psi) \qquad \text{or} \qquad t = \cos(\phi - \psi) \pm i \sin(\phi - \psi), \tag{88}$$

where $i = \sqrt{-1}$. Consequently, there are real values of t that make $b^2 - ac = 0$ only if $\sin(\phi + \psi) = 0$ or $\sin(\phi - \psi) = 0$. These situations occur when $\phi = \pm\psi$ and $\phi = \pm\psi + \pi$. Substituting into Equation 88 gives that $b^2 - ac = 0$ iff both $\phi = \pm\psi$ or $\phi = \pm\psi + \pi$ and $t = 1$, where $t = 1$ is the same as $\frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}$.

E   No Solutions for Scale

This appendix shows that there always exists a solution to the biquadratic by showing that $b^2 - ac \geq 0$. From Appendix F,

$$
\begin{aligned}
b^2 - ac &= 4(R_{01}d_{02})^4 \left(t^2 - 2\cos(\phi + \psi)t + 1\right)\left(t^2 - 2\cos(\phi - \psi)t + 1\right) \\
&\geq 4(R_{01}d_{02})^4 \left(t^2 - 2t + 1\right)\left(t^2 - 2t + 1\right) \\
&= 4(R_{01}d_{02})^4(t-1)^4 \\
&\geq 0
\end{aligned}
$$

F   Simplifying $b^2 - ac$

In this appendix, I derive that

$$
b^2 - ac = 4(R_{01}d_{02})^4 \left(t^2 - 2\cos(\phi + \psi)t + 1\right)\left(t^2 - 2\cos(\phi - \psi)t + 1\right), \tag{89}
$$

where

$$
t = \frac{R_{02}d_{01}}{R_{01}d_{02}}.
$$

From Equations 24, 25, and 26,

$$
\begin{aligned}
a &= 4(R_{01}R_{02}\sin\phi)^2 \\
b &= 2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi) \\
c &= 4(d_{01}d_{02}\sin\psi)^2
\end{aligned}
$$

Then

$$
\begin{aligned}
b^2 &= 4(R_{02}^4 d_{01}^4 - 4R_{02}^3 d_{01}^3 R_{01}d_{02}\cos\phi\cos\psi + 2R_{01}^2 R_{02}^2 d_{01}^2 d_{02}^2 + \\
&\quad 4R_{01}^2 R_{02}^2 d_{01}^2 d_{02}^2 \cos^2\phi\cos^2\psi - 4R_{01}^3 d_{02}^3 R_{02}d_{01}\cos\phi\cos\psi + R_{01}^4 d_{02}^4)
\end{aligned}
$$

$$
ac = 16 R_{01}^2 R_{02}^2 d_{01}^2 d_{02}^2 \sin^2\phi\sin^2\psi
$$

$$
\begin{aligned}
b^2 - ac &= 4\left(R_{02}^4 d_{01}^4 - 4R_{02}^3 d_{01}^3 R_{01}d_{02}\cos\phi\cos\psi + \right. \\
&\quad (2 + 4\cos^2\phi\cos^2\psi - 4\sin^2\phi\sin^2\psi)R_{01}^2 R_{02}^2 d_{01}^2 d_{02}^2 - \\
&\quad \left. 4R_{01}^3 d_{02}^3 R_{02}d_{01}\cos\phi\cos\psi + R_{01}^4 d_{02}^4\right) \\
&= 4(R_{01}d_{02})^4 \left(t^4 - 4\cos\phi\cos\psi t^3 + (2 + 4\cos^2\phi\cos^2\psi - 4\sin^2\phi\sin^2\psi)t^2 - \right. \\
&\quad \left. 4\cos\phi\cos\psi t + 1\right), \qquad \text{where} \qquad t = \frac{R_{02}d_{01}}{R_{01}d_{02}}
\end{aligned}
$$

$$
\begin{aligned}
= \quad & 4(R_{01}d_{02})^4 \left(t^4 - 2\left(\cos(\phi+\psi) + \cos(\phi-\psi)\right)t^3 + \right. \\
& \left. \left(2 + 4\cos(\phi+\psi)\cos(\phi-\psi)\right)t^2 - 2\left(\cos(\phi+\psi) + \cos(\phi-\psi)\right)t + 1 \right) \\
= \quad & 4(R_{01}d_{02})^4 \left(t^2 - 2\cos(\phi+\psi)t + 1 \right)\left(t^2 - 2\cos(\phi-\psi)t + 1 \right)
\end{aligned}
$$

References

[1] Ayache, N., and O. D. Faugeras, "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 1, pp. 44-54, January 1986.

[2] Ballard, D. H., "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.

[3] Basri, R., and S. Ullman, "The Alignment of Objects with Smooth Surfaces," in *Proc. Second Inter. Conf. Computer Vision*, pp. 482-488, 1988.

[4] Cass, T. A., "Feature Matching for Object Localization in the Presence of Uncertainty," in *Proc. Third Inter. Conf. Computer Vision*, pp. 360-364, 1990.

[5] Clark, C. S., W. O. Eckhardt, C. A. McNary, R. Nevatia, K. E. Olin, and E. M. VanOrden, "High-accuracy Model Matching for Scenes Containing Man-Made Structures," in *Proc. Symp. Digital Processing of Aerial Images*, SPIE, vol. 186, pp. 54-62, 1979.

[6] Craig, J. J., *Introduction to Robotics*, Reading, MA: Addison-Wesley, 1955.

[7] Cyganski, D., and J. Orr, "Applications of Tensor Theory to Object Recognition and Orientation Determination," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 7, no. 6, November 1985.

[8] Cyganski, D., and J. Orr, "Object Recognition and Orientation Determination by Tensor Methods," Advances in Computer Vision and Image Processing, vol. 3, Thomas Huang, Ed., Jai Press, Inc., 1988.

[9] Faugeras, O. D., and M. Hebert, "The Representation, Recognition, and Locating of 3-D Objects," *Inter. J. Rob. Res.*, vol. 5, no. 3, pp. 27-52, 1986.

[10] Fischler, M. A., and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, pp. 381-396, June 1981.

[11] Grimson, W. E. L., and D. P. Huttenlocher, "On the Sensitivity of the Hough Transform for Object Recognition," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 12, no. 3, March 1990.

[12] Grimson, W. E. L., D. P. Huttenlocher, and T. D. Alter, "Recognizing 3D Objects from 2D Images: An Error Analysis," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, 1992.

[13] Haralick, R. M., C. Lee, K. Ottenberg, and M. Nolle, "Analysis and Solutions of the Three Point Perspective Pose Estimation Problem," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 592-598, 1991.

[14] Horn, B. K. P., "Closed-Form Solution of Absolute Orientation Using Unit Quaternions," *J. Opt. Soc. Am.*, vol. 4, no. 4, April 1987.

[15] Horaud, R., "New Methods for Matching 3-D Objects with Single Perspective Views," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 3, pp. 401-412, May 1987.

[16] Huttenlocher, D. P., "Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image," MIT TR 1045, 1988.

[17] Huttenlocher, D. P., and S. Ullman, "Object Recognition Using Alignment," in *Proc. First Inter. Conf. Computer Vision*, pp. 102-111, June 1987.

[18] Huttenlocher, D. P., and S. Ullman, "Recognizing Solid Objects by Alignment with an Image," *Inter. J. Computer Vision*, vol. 5, no. 2, pp. 195-212, 1990.

[19] Jacobs, D. W., "Optimal Matching of Planar Models in 3D Scenes," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, 1991.

[20] Kanade, T., and J. R. Kender, "Mapping Image Properties into Shape Constraints: Skew Symmetry, Affine-Transformable Patterns, and the Shape-from-Texture Paradigm," Human and Machine Vision, Beck, Hope, and Rosenfeld, Eds., Academic Press, 1983.

[21] Lamdan, Y., J. T. Schwartz, and H. J. Wolfson, "Object Recognition by Affine Invariant Matching" in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 335-344, 1988.

[22] Lamdan, Y., and H. J. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 238-249, 1988.

[23] Linnainmaa, S., D. Harwood, and L. S. Davis, "Pose Determination of a Three-Dimensional Object Using Triangle Pairs," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 5, pp. 634-647, Sept. 1988.

[24] Roberts, L. G., "Machine Perception of Three-Dimensional Solids," Optical and Electro-Optical Information Processing, J. T. Tippett et al., Eds., MIT Press, Cambridge, MA, 1965.

[25] Thompson, D. W., and J. L. Mundy, "Three-Dimensional Model Matching from an Unconstrained Viewpoint," in *Proc. IEEE Conf. Rob. Aut.*, pp. 208-220, 1987.

[26] Turney, J. L., T. N. Mudge, and R. A. Voltz, "Recognizing Partially Occluded Parts," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 7, no. 4, July 1985.

[27] Ullman, S., "Computational Studies in the Interpretation of Structure and Motion: Summary and Extension," Human and Machine Vision, Beck and Rosenfeld, Eds., Academic Press, 1983.

[28] Ullman, S., "An Approach to Object Recognition: Aligning Pictorial Descriptions," MIT A.I. Memo 931, December 1985.

[29] Ullman, S., "Aligning Pictorial Descriptions: An Approach to Object Recognition," *Cognition*, vol. 32, no. 3, pp. 193-254, August 1989.

[30] Ullman, S., and R. Basri, "Recognition by Linear Combinations of Models," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 13, no. 10, pp. 992-1006, October 1991.

[31] Watkins, D., *Fundamentals of Matrix Computations*, New York: John Wiley & Sons, Inc., 1991.