International Journal of Computer Vision, 3, 261–292 (1989)

Reconstruction of Consistent Shape from Inconsistent Data: Optimization of 2½D Sketches

KENICHI KANATANI
Department of Computer Science, Gunma University Kiryu, Gunma 376, Japan

Abstract
Although the 3D orientations of edges and surfaces are theoretically sufficient for reconstructing the 3D object shape, this does not mean that the 3D object shape can actually be reconstructed. Specifying the edge and surface orientations is often overspecification, and inconsistency may result if image data contain errors. We propose a scheme of optimization to construct a consistent polyhedron shape from inconsistent data. Our optimization is achieved by solving a set of *linear* equations; no searchers and iterations are necessary. This technique is first applied to the problem of *shape-from-motion* and then to the 3D recovery based on the *rectangularity hypothesis* and the *parallelism hypothesis.* We also present a strategy of heuristic reasoning on rectangularity and parallelism.

1 What Does 3D Shape Recovery Mean?

In the past, various 3D shape recovery paradigms called "shape from..." have been proposed—shape-from-motion, shape-from-shading, and shape-from-texture, for example. Now, we must ask the following question: Do these techniques really enable us to recover the 3D object shape? They certainly provide us with *sufficient* information for the 3D shape recovery. However, "to provide sufficient information" does not necessarily mean "to recover the 3D shape." The purpose of this paper is to illustrate this fact and present an *optimization* strategy to fill this gap.

If we closely look into the "shape from..." paradigms, we find that the information we obtain is the *3D orientations* of edges and surfaces constituting the object. For example, the surface gradient can be recovered from optical flow (shape-from-motion), intensity of light reflectance (shape-from-shading), or texture density (shape-from-texture), and the 3D orientations of edges can be recovered from the 2D orientations of their projections. In other words, we are in general unable to recover the *absolute depth;* we can estimate only the 3D orientations of edges and surfaces.

Mathematically speaking, the object shape is uniquely determined if the 3D orientations of its constituent edges and surfaces are specified.[1] In other words, the 3D orientations of edges and surfaces are *sufficient* for 3D shape recovery. In spite of this trivial fact, however, there arises a serious difficulty in real situations: Specifying the 3D orientations of edges and surfaces is often *overspecification.*

For example, suppose the image is segmented into regions resulting from projection of a planar part of the object surface—let us call such regions *planar patches.* If the object is a polyhedron, regions representing its faces naturally constitute planar patches. If the object has a smooth surface, the image region is segmented into a collection of approximate planar patches; the object is approximated by an appropriate polyhedron.[2] Assume that the surface gradient is known for each planar patch. Can we reconstruct the 3D object shape? Theoretically, yes: We can reconstruct the shape by patching together appropriate planar surfaces one by one in the scene according to the prescribed orientations in such a way that each of them is projected exactly onto the corresponding region of the image plane.

However, we are working in the real world; the image analysis is based on the data supplied by low-level image processing stages, and these data necessarily contain errors. If the surface gradient estimates are not accurate, the reconstructed object shape depends on the order of the patching, and there may arise *incompatibility* of face adjacency; two faces may not meet at a common boundary (figure 1a).

[1]Here and in the following, we disregard the *absolute size* when we talk about the "3D shape." In other words, the "shape" is the same if it is multipled by an abritrary scale factor.

[2]We mean the *visible part* of the object surface. In the following, we ignore the invisible part occluded by the object itself.

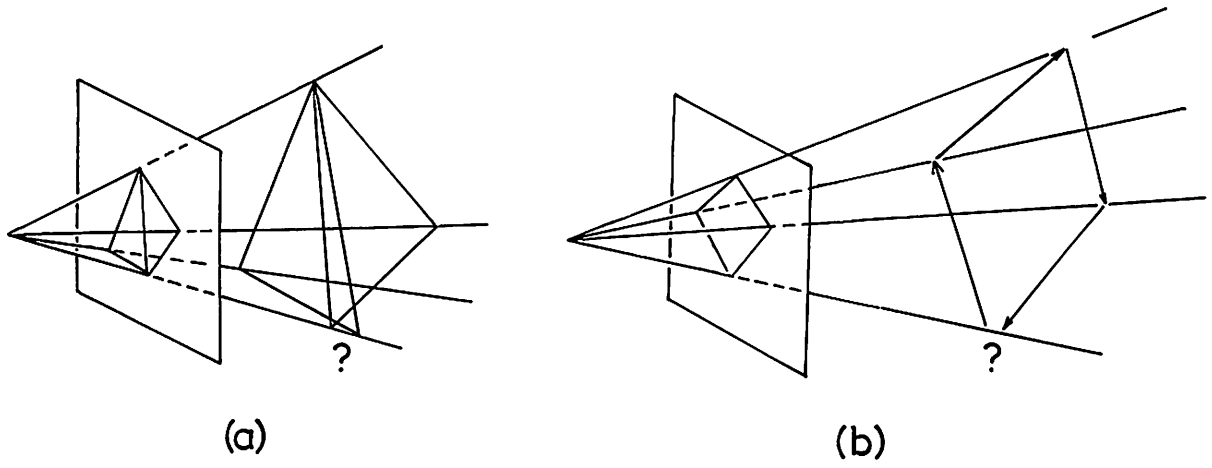


Fig. 1 (a) Incompatibility of face adjacency: If faces are placed in the scene according to their surface gradient estimates in such a way that they are projected onto the observed image, adjacent faces may not meet with common boundaries. (b) Incompatibility of edge adjacency: If edges are placed in the scene according to their 3D orientation estimates in such a way that they are projected onto the observed image, edges constituting the boundary of a face may not form a closed loop.

A similar trouble arises when 3D edge orientations are estimated. Theoretically, we can reconstruct the 3D object shape by placing its edges in the scene according to their prescribed 3D orientations in such a way that each edge is projected exactly onto the corresponding edge image. In the presence of noise, however, if we apply this process to edges constituting the boundary of a face, they do not necessarily form a closed loop; the end point may not coincide with the starting point (figure 1b). Even if they do coincide, the resulting face may not be planar.

It follows that there is a big gap between obtaining *sufficient* information for 3D recovery and actually reconstructing the 3D shape. This consideration leads to the following observation: As long as the object is approximated by a polyhedron, *the surface gradient values cannot be assigned arbitrarily.* This fact has been realized by many researchers, but it was Sugihara [35–38] that first studied this problem in explicit algebraic terms. In the next section, we study this strong constraint in detail.

Next, we present a scheme of *optimization* for computing a consistent polyhedron shape from inconsistent image data. A basic idea was already given by Sugihara [37]. In this paper, we show that by an ingenious change of variables the optimization reduces to solving a set of simultaneous *linear* equations. Then, our technique is applied to the following three typical types of shape clues.[3]

We first apply our technique to the shape-from-motion problem and demonstrate how a consistent object is reconstructed from inaccurate data. Then, we consider 3D shape recovery of polyhedra from a single image. First, we adopt the *rectangularity hypothesis,* assuming corners to be rectangular unless inconsistency results. The optimization technique can reconstruct a consistent 3D shape once rectangular corners are identified. We present a heuristic strategy for finding rectangular corners. Next, we adopt the *parallelism hypothesis,* assuming edges to be parallel unless inconsistency results. Again, the optimization technique can reconstruct a consistent 3D shape if parallel edges are identified. We also present a heuristic strategy for finding parallel edges.

The subject of this paper encompasses almost all aspects of 3D interpretation of images; here, they are viewed from the standpoint of optimizing inconsistent data. Hence, it is practically impossible to refer to all existing works related to the present work, since it would mean listing virtually all literature on computer vision. Thus, the references made in this paper are inevitably incomplete.

[3]Of course, there exist many other ways of 3D shape reasoning, e.g., direct methods such as stereo and range sensing, analytical methods such as shape-from-shading and shape-from-texture, and various heuristic types. Also see Pollard et al. [30], Porrill et al. [31], and Sugihara [37,38], for example, for the consistent use of such clues.

Since the subsequent discussions cover a very wide range of topics, only the mathematical framework of our optimization scheme and relevant principles of heuristic reasoning are discussed in the main body; related miscellaneous topics are discussed in appendixes, where many already published results are summarized, including the author's own, in the context of our optimization scheme. They are presented only for the sake of consistency; they are not meant to be original.

2 Constraints on 2½D Sketches

As we pointed out in the preceding section, the "shape from..." paradigms usually present us with object images equipped with the following types of 3D information:

1. The surface gradient (p, q), or equivalently the unit surface normal $\boldsymbol{n}$, is *densely* estimated (i.e., as a function of location) over the region corresponding to the object surface (figure 2a).
2. The region corresponding to the object surface is segmented into planar patches, and the surface gradient (p, q), or equivalently the unit surface normal $\boldsymbol{n}$, is estimated for each patch (figure 2b). In other words, the object surface is approximated by a polyhedron.
3. The region corresponding to the object surface is segmented into planar patches and approximated by a polyhedron with estimated 3D edge orientations (figure 2c).

Let us call an image equipped with such 3D information a *2½D sketch.*[4] Among these three, case (1) may be the one we most often encounter, because what we can estimate is often the *local surface gradient*—the surface gradient obtained *on the assumption that the object surface is locally flat.* If the object is itself a polyhedron, its faces themselves constitute planar patches. If the object has a smooth surface, it is approximated by a collection of planar patches.

First, consider case (2), and assume that the object surface is (or is approximated by) a polyhedron. Case (3) will be considered later. (See appendix A for case (1)). As pointed out in the preceding section, *being a polyhedron* is a very strong constraint, and the surface gradient cannot be assigned arbitrarily: If the surface gradient is assigned to one face, the surface gradients of the adjacent faces are partially constrained. Then, there arises a natural question: How much freedom do we have in assigning values? This question leads to the definition of the *degree of freedom* of a polyhedron image.

[4]This term was coined by David Marr [24], although he meant only case (1).

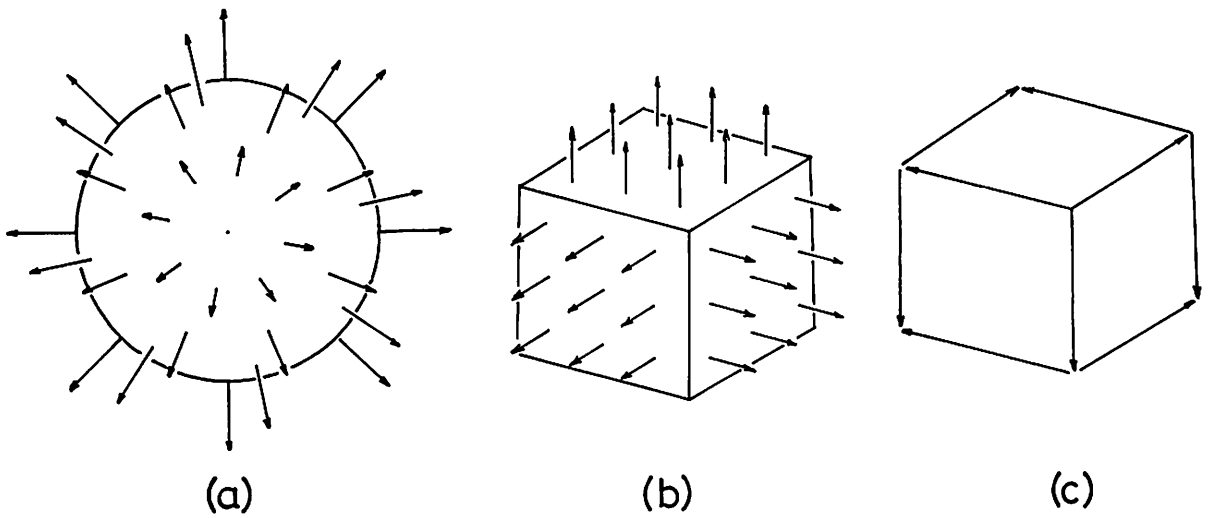


Fig. 2. 2½D sketch. (a) The surface gradient, or equivalently the surface normal, is estimated densely over the region corresponding to an object surface. (b) The image region corresponding to an object surface is segmented into planar patches, and the surface gradient, or equivalently the surface normal, is estimated for each patch. (c) The image region corresponding to an object surface is segmented into planar patches, and the 3D orientation is estimated for each edge.

This issue was first raised by Sugihara [35–38], who introduced the notion of the *incidence structure* of a polyhedron. Following him, we define the incidence structure as follows: We say that a vertex is *incident* to a face if the vertex is on the boundary of the face. Consider a polyhedron image. Let $V = \{V_1, \ldots, V_n\}$ be the set of its vertexes, and $F = \{F_1, \ldots, F_m\}$ be the set of its faces. The incidence structure is specified by a set R of *incidence pairs* (F_α, V_i) meaning that vertex V_i is incident to face F_α.[5] Let l be the number of such incidence pairs.

Let (X_i, Y_i, Z_i) be the scene coordinates of vertex V_i, $i = 1, \ldots, n$, and let $Z = p_\alpha X + q_\alpha Y + r_\alpha$ be the equation of face F_α, $\alpha = 1, \ldots, m$. The pair (p_α, q_α) indicates its surface gradient. Let us call $p_\alpha, q_\alpha, r_\alpha$ the *surface parameters* of face F_α. The incidence pair (F_α, V_i) states that vertex V_i is incident to face F_α:

$$Z_i = p_\alpha X_i + q_\alpha Y_i + r_\alpha \tag{1}$$

In this paper, we use a coordinate system fixed to the camera in such a way that the Z axis coincides with the optical axis and point $(0, 0, -f)$ (which we henceforth call the *viewpoint*) coincides with the center of the lens. Then, we can think of the XY-plane as the image plane (figure 3): A point in the scene is perspectively projected onto the image plane from the viewpoint, which is on the negative side of the Z-axis away from the image plane by distance f. (See Remark 2 for the reason why the coordinate system is chosen this way.) We call the constant f the *focal length*.[6]

From figure 3, it is easy to see that the image coordinates (x_i, y_i) of vertex V_i are related to its scene coordinates (X_i, Y_i, Z_i) by the projection equations

$$x_i = \frac{fX_i}{f + Z_i} \qquad y_i = \frac{fY_i}{f + Z_i} \tag{2}$$

Now, we introduce a new quantity

$$z \equiv \frac{fZ}{f + Z} \tag{3}$$

and call it the *reduced depth*. It follows that there exists a one-to-one correspondence between the scene coordinates (X_i, Y_i, Z_i) of vertex V_i and (x_i, y_i, z_i):

$$x_i = \frac{fX_i}{f + Z_i} \qquad y_i = \frac{fY_i}{f + Z_i} \qquad z_i = \frac{fZ_i}{f + Z_i} \tag{4}$$

$$X_i = \frac{fx_i}{f - z_i} \qquad Y_i = \frac{fy_i}{f - z_i} \qquad Z_i = \frac{fz_i}{f - z_i} \tag{5}$$

Remark 1. Equations (4) and (5) can be regarded as defining a one-to-one mapping between the XYZ space and the xyz space.

$$x = \frac{fX}{f + Z} \qquad y = \frac{fY}{f + Z} \qquad z = \frac{fZ}{f + Z} \tag{6}$$

$$X = \frac{fx}{f - z} \qquad Y = \frac{fy}{f - z} \qquad Z = \frac{fz}{f - z} \tag{7}$$

This is a *projective transformation* of the XYZ space, preserving *collinearity* and *coplanarity*: A line is mapped onto a line, and a plane is mapped onto a plane. From equations (6) and (7), we find that a plane $Z = \text{const.}$ is mapped onto a plane $z = \text{const.}$ In particular, the XY-plane ($Z = 0$) is mapped onto the xy-plane ($z = 0$), and the "plane" $Z = \infty$ at infinity is mapped onto the plane $z = f$, while the plane $Z = -f$ is mapped onto the "plane" $z = \infty$ at infinity. We can also prove easily that lines passing through the viewpoint $(0, 0, -f)$ in the XYZ-space are mapped onto lines parallel to the z-axis in the xyz-space. Thus, the transformation is like the one shown in figure 4. This

[5] To put it in algebraic terms, the incidence structure R is a *relation* over F and V, i.e., a subset of the Cartesian product $F \times V$.

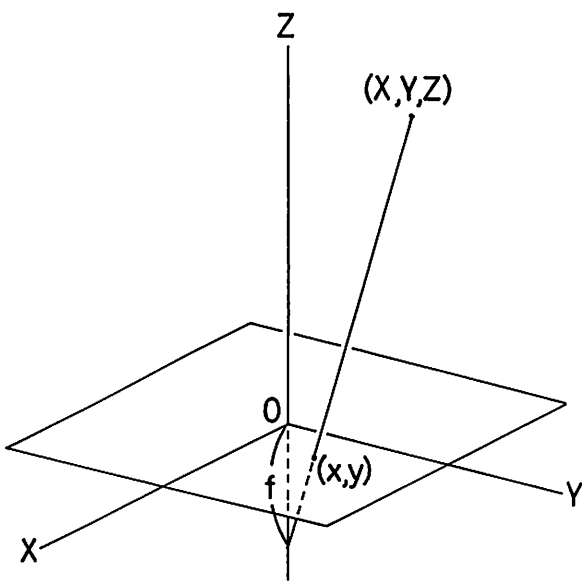


Fig. 3. Perspective projection: A point (X, Y, Z) is projected onto point (x, y) on the image plane $Z = 0$ by the lay passing through the viewpoint $(0, 0, -f)$.

[6] It does not necessarily coincide with the focal length of the lens itself.

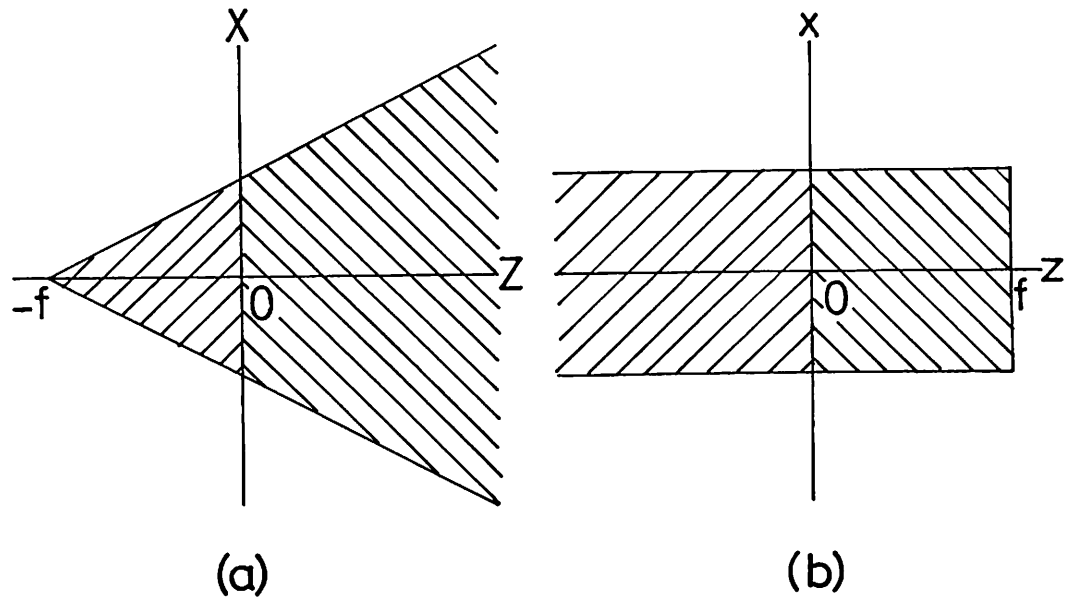


Fig. 4. Reduced depth z. The *XYZ*-space (a) is mapped onto the *xyz*-space (b) by a projective transformation which preserves collinearity and coplanarity. The infinite depth $Z = \infty$ is reduced to $z = f$. In the *xyz*-space, the projection can be regarded *as if* being orthographic.

transformation is frequently used in computer graphics, because it makes the removal of *hidden lines* very easy: Rays starting from the viewpoint become parallel to the z-axis, and visibility can be checked as if the projection were orthographic.

Substituting equations (5) into equation (1), we can express the reduced depth z_i of vertex V_i in terms of its image coordinates (x_i, y_i):

$$z_i = \frac{fp_\alpha}{f + r_\alpha} x_i + \frac{fq_\alpha}{f + r_\alpha} y_i + \frac{fr_\alpha}{f + r_\alpha} \quad (8)$$

Now, we define new parameters

$$P_\alpha \equiv \frac{fp_\alpha}{f + r_\alpha} \quad Q_\alpha \equiv \frac{fq_\alpha}{f + r_\alpha} \quad R_\alpha \equiv \frac{fr_\alpha}{f + r_\alpha} \quad (9)$$

and call these the *reduced surface parameters*. The inverse relationship to the original surface parameters p_α, q_α, r_α is given by

$$p_\alpha = \frac{fP_\alpha}{f - R_\alpha} \quad q_\alpha = \frac{fQ_\alpha}{f - R_\alpha} \quad r_\alpha = \frac{fR_\alpha}{f - R_\alpha} \quad (10)$$

In terms of the reduced surface parameters P_α, Q_α, R_α, the reduced depth z_i is written as

$$z_i = P_\alpha x_i + Q_\alpha y_i + R_\alpha \quad (11)$$

Since the image coordinates (x_i, y_i) of vertex V_i are known, the 3D position of vertex V_i is determined by equations (5) if its reduced depth z_i is known. Consequently, the reduced depths z_i, $i = 1, \ldots, n$, can be taken as unknowns to specify the 3D vertex positions instead of the original depths Z_i, $i = 1, \ldots, n$. Similarly, the reduced surface parameters P_α, Q_α, R_α, $\alpha = 1, \ldots, m$, can serve as unknowns for the surface shape instead of the original surface parameters p_α, q_α, r_α, $\alpha = 1, \ldots, m$. Thus, we obtain l (= the number of incidence pairs) equations

$$x_i P_\alpha + y_i Q_\alpha + R_\alpha - z_i = 0 \quad (F_\alpha, V_i) \in R \quad (12)$$

These equations are *linear* in the reduced depths z_i, $i = 1, \ldots, n$, and the reduced surface parameters P_α, Q_α, R_α, $\alpha = 1, \ldots, m$. Let us call these l equations simply the *constraints* of the polyhedron image.

Suppose the l constraints are linearly independent. The number of unknowns is $N = n + 3m$ (the number of vertexes plus three times the number of faces). This means that the solution is a point in an N-dimensional space. Since equation (12) imposes l linear constraints on the N variables, the solution is constrained to an $(N - l)$-dimensional linear subspace. The dimensionality of the solution space is called the *degree of freedom*. Thus, if the degree of freedom is r, the

solution contains r indeterminate parameters. This means that we can specify at most r values to the polyhedron image.[7]

If the l constraints are linearly independent, the degree of freedom is given by $r = n + 3m - l$ $(= N - l)$. If the l constraints are not independent, say only l' $(<l)$ of them are independent, the degree of freedom is $r = n + 3m - l'$ $(>n + 3m - l)$. Thus, the degree of freedom is at least $n + 3m - l$. However, the degree of freedom of a polyhedron image should be at least *four* if the polyhedron is not *coplanar*, that is, not "flattened out" (see appendix B). In regard to this problem, Sugihara [35–38] introduced the notion of *singularity* of the incidence structure, and gave a complete algebraic treatment to test if a given polyhedron image can be interpreted to be a projection of a real polyhedron (see appendix C).

Remark 2. The use of the image coordinates (x_i, y_i) and the reduced depths z_i as well as the reduced surface parameters P_α, Q_α, R_α enables us to forget the difference between perspective and orthographic projections. In fact, orthographic projection is attained in the limit of $f \to \infty$, and in this limit we can see from equations (4), (5), (9), and (10) that all the reduced parameters simply coincide with the original parameters: $x_i \to X_i$, $y_i \to Y_i$, $z_i \to Z_i$, $P_i \to p_i$, $Q_i \to q_i$, $R_i \to r_i$. Hence, all the subsequent discussions also hold for orthographic projection if this limit is taken. This is the reason why we adopt the coordinate system centered at the image plane instead of the viewpoint.[8]

3 Optimization of a 2½D Sketch

3.1 Surface Gradients Estimated

Suppose we are given a 2½D sketch. Let $V = \{V_1, \ldots, V_n\}$ and $F = \{F_1, \ldots, F_m\}$ be the sets of its vertexes and faces, respectively. Let $R = \{(F_\alpha, V_i)\}$ be its incidence structure. We assume that this incidence structure is nonsingular.[9] Let $(\hat{p}_\alpha, \hat{q}_\alpha)$ be the estimate

[7]However, this does not mean that we can assign r surface gradients arbitrarily. The interdependence of surface gradients depends on the incidence structure.

[8]It would certainly be possible to stick to either throughout, but then we would miss many critical observations buried in notational complexity.

[9]If it is singular, we can always make it nonsingular with a small modification as shown in appendix C.

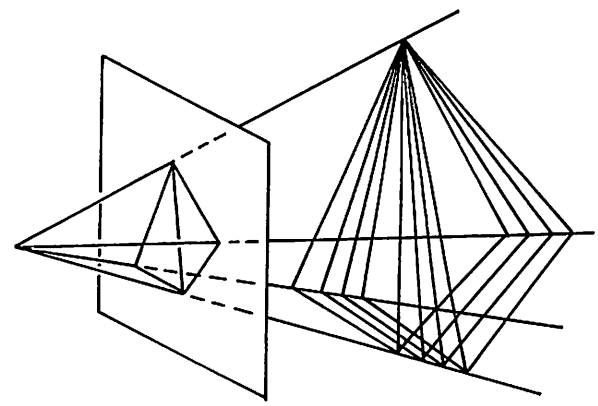


Fig. 5. Optimization: From among infinitely many inconsistent polyhedron solutions which are exactly projected onto the observed image, we seek the one whose surface gradients are the closest to the given estimates *on the average*.

of the surface gradient of face F_α. As we argued in the preceding section, there does not in general exist polyhedron that is compatible with the given projection image and yet exactly has the prescribed surface gradients.

Here, we assume that the surface gradient estimates $(\hat{p}_\alpha, \hat{q}_\alpha)$, $\alpha = 1, \ldots, m$, are not accurate, while the given image coordinates (x_i, y_i), $i = 1, \ldots, n$, are accurate.[10] Let (p_α, q_α) be the true surface gradient of face F_α. Here, we seek, from among the infinitely many consistent polyhedron solutions that are exactly projected onto the observed image, the one whose surface gradients are the closest to the given estimates *on the average* (figure 5). Specifically, let us consider the least-square method to minimize

$$J = \frac{1}{2} \sum_{\alpha=1}^{m} W_\alpha [(p_\alpha - \hat{p}_\alpha)^2 + (q_\alpha - \hat{q}_\alpha)^2] \quad (13)$$

where W_α is the weight for face F_α. If equations (12) are substituted, equation (13) is rewritten in terms of the reduced surface parameters as follows:

[10]The observed image coordinates (x_i, y_i), $i = 1, \ldots, n$, may also contain errors if the object is a "real" polyhedron. However, most surface gradient estimation processes involve these image coordinates. Consequently, the computed surface gradient estimates are *more indirect* than these coordinates. If the image is a polyhedral approximation of a smooth surface, the image coordinates are always "exact," since the tessellation is completely arbitrary.

$$J = \frac{1}{2}\sum_{\alpha=1}^{m} W_\alpha \left(\frac{f + r_\alpha}{f}\right)^2 \left[\left(P_\alpha + \frac{\hat{p}_\alpha}{f} R_\alpha - \hat{p}_\alpha\right)^2 + \left(Q_\alpha + \frac{\hat{q}_\alpha}{f} R_\alpha - \hat{q}_\alpha\right)^2\right] \tag{14}$$

In order to make our analysis easy, we now introduce a small trick: We replace r_α in the above equation by its estimate $\hat{r}_\alpha$, assuming that it is somehow available. Put

$$W_\alpha \left(\frac{f + \hat{r}_\alpha}{f}\right)^2 = \frac{1}{\zeta_\alpha} \tag{15}$$

Now, ζ_α is a constant assigned to face F_α whose value is yet to be determined.

Remark 3. One example of assigning the value of ζ_α is as follows. If face F_α has a large depth $\hat{r}_\alpha$, estimation of the surface gradient may not be very accurate, so a small weight W_α should be assigned. At the same time, if either $\hat{p}_\alpha$ or $\hat{q}_\alpha$ has a very large magnitude, the measurement may also be inaccurate, so a small weight W_α should be assigned, too. In view of these considerations, it may be appropriate to choose the following weight:[11]

$$W_\alpha = \frac{1}{\hat{p}_\alpha^2 + \hat{q}_\alpha^2 + 1}\left(\frac{f}{f + \hat{r}_\alpha}\right)^2 \tag{16}$$

For this weight, the constraint ζ_α becomes simply

$$\zeta_\alpha = \hat{p}_\alpha^2 + \hat{q}_\alpha^2 + 1 \tag{17}$$

Of course, the choice is not unique. Different choices may result in different solutions. The validity of the choice must be checked a posteriori.

The problem now reduces to minimization of J under the constraints (12) for all incidence pairs $(F_\alpha, V_i) \in R$. However, as long as we try to reconstruct the 3D object shape from a single image by using surface gradient cues, the absolute depth cannot be determined.[12] Hence, we must give the depth Z, or equivalently the reduced depth z, to one vertex. Let that vertex be V_n. Since J is quadratic and the constraints (12) are linear in the unknowns, the minimum is attained by solving a set of *linear* equations. If we introduce Lagrangian multipliers $\Lambda_{\alpha i}$ to all the incidence pairs $(F_\alpha, V_i) \in R$, the final result becomes as follows:

Proposition 1. The reduced depths z_i of vertexes V_i, $i = 1, \ldots, n-1$, are given by solving the following $n + 3m + l - 1$ linear equations in $n + 3m + l - 1$ unknowns z_i, $i = 1, \ldots, n-1$, P_α, Q_α, R_α, $\alpha = 1, \ldots, m$, and $\Lambda_{\alpha i}$ for $(F_\alpha, V_i) \in R$:

$$x_i P_\alpha + y_i Q_\alpha + R_\alpha - z_i = 0, \quad (F_\alpha, V_i) \in R \tag{18}$$

$$P_\alpha + \frac{\hat{p}_\alpha}{f} R_\alpha + \zeta_\alpha \sum_{i:(F_\alpha,V_i)\in R} x_i \Lambda_{\alpha i} = \hat{p}_\alpha, \quad \alpha = 1, \ldots, m \tag{19}$$

$$Q_\alpha + \frac{\hat{q}_\alpha}{f} R_\alpha + \zeta_\alpha \sum_{i:(F_\alpha,V_i)\in R} y_i \Lambda_{\alpha i} = \hat{q}_\alpha, \quad \alpha = 1, \ldots, m \tag{20}$$

$$\sum_{i:(F_\alpha,V_i)\in R} \left(\frac{\hat{p}_\alpha x_i + \hat{q}_\alpha y_i}{f} - 1\right) \Lambda_{\alpha i} = 0, \quad \alpha = , \ldots, m \tag{21}$$

$$\sum_{\alpha:(F_\alpha,V_i)\in R} \Lambda_{\alpha i} = 0, \quad i = 1, \ldots, n-1 \tag{22}$$

Remark 4. In Proposition 1, we assumed that surface gradient estimates $(\hat{p}_\alpha, \hat{q}_\alpha)$ are given for all faces. This is, however, not always necessary; if some faces are given no estimates, the corresponding terms are simply dropped from equation (13). As a result, if face F_α is given no surface gradient estimate, equations (19–21) are respectively replaced by the following equations:

$$\sum_{i:(F_\alpha,V_i)\in R} x_i \Lambda_{\alpha i} = 0$$

$$\sum_{i:(F_\alpha,V_i)\in R} y_i \Lambda_{\alpha i} = 0 \tag{23}$$

$$\sum_{i:(F_\alpha,V_i)\in R} \Lambda_{\alpha i} = 0$$

[11] In the denominator, +1 is added to avoid zero division in case $\hat{p}_\alpha = \hat{q}_\alpha = 0$.

[12] Note that the projection image and the surface gradient of each face is kept identical if the object is translated in the scene away from the viewer and at the same time its size is proportionally increased.

Remark 5. In the orthographic limit $f \to \infty$, equations (19–21) simply reduce to

$$P_\alpha + \zeta_\alpha \sum_{i:(F_\alpha,V_i)\in R} x_i \Lambda_{\alpha i} = \hat{p}_\alpha, \quad \alpha = 1, \ldots, m \tag{24}$$

$$Q_\alpha + \zeta_\alpha \sum_{i:(F_\alpha,V_i)\in R} y_i \Lambda_{\alpha i} = \hat{q}_\alpha, \quad \alpha = 1, \ldots, m \tag{25}$$

$$\sum_{i:(F_\alpha,V_i)\in R} \Lambda_{\alpha i} = 0, \quad \alpha = 1, \ldots, m \tag{26}$$

3.2 Edge Orientations Estimated

Next, consider case (3)—the 3D edge orientations estimated. Again, Let $V = \{V_1, \ldots, V_n\}$ and $F = \{F_1, \ldots, F_m\}$ be the sets of vertexes and faces, respectively. Let $E_\alpha = \{e_1, \ldots, e_{N\alpha}\}$ be the set of edges constituting the boundary of face F_α, and let $\hat{e}_k = (\hat{e}_{k(1)}, \hat{e}_{k(2)}, \hat{e}_{k(3)})$ be the unit vector indicating the estimated 3D orientation of edge e_k. If (p_α, q_α) is the surface gradient of face F_α, the unit surface normal $\boldsymbol{n}_\alpha = (n_{\alpha(1)}, n_{\alpha(2)}, n_{\alpha(3)})$, to face F_α is given by

$$\boldsymbol{n}_\alpha = \left(\frac{p_a}{\sqrt{p_\alpha^2 + a_\alpha^2 + 1}}, \frac{q_\alpha}{\sqrt{p_\alpha^2 + q_\alpha^2 + 1}}, \frac{-1}{\sqrt{p_\alpha^2 + q_\alpha^2 + 1}} \right) \tag{27}$$

The vectors $\hat{e}_k$ for edges $\hat{e}_k \in E_\alpha$ should be all orthogonal to $\boldsymbol{n}_\alpha$, but this is not necessarily guaranteed in the presence of noise. Hence, it is reasonable to estimate the surface normal $\boldsymbol{n}_\alpha$ by the least-square method which minimizes[13]

$$\frac{1}{2} \sum_{e_k\in E_\alpha} (\hat{e}_k, \boldsymbol{n}_\alpha)^2 = \frac{1}{2} \sum_{e_k\in E_\alpha} (\hat{e}_{k(1)} n_{\alpha(1)} + \hat{e}_{k(2)} n_{\alpha(2)} + \hat{e}_{k(3)} n_{\alpha(3)})^2 \tag{28}$$

However, this minimization need not be done for each face separately. The surface normals of all the faces are estimated by minimizing

$$J = \frac{1}{2} \sum_{\alpha=1}^{m} W_\alpha \sum_{e_k\in E_\alpha} (\hat{e}_{k(1)} n_{\alpha(1)} + \hat{e}_{k(2)} n_{\alpha(2)} + \hat{e}_{k(3)} n_{\alpha(3)})^2$$
$$= \frac{1}{2} \sum_{\alpha=1}^{m} \frac{W_\alpha}{p_\alpha^2 + q_\alpha^2 + 1} \sum_{e_k\in E_\alpha} (\hat{e}_{k(1)} p_\alpha + \hat{e}_{k(2)} q_\alpha - \hat{e}_{k(3)})^2 \tag{29}$$

where W_α is the weight for face F_α. If equations (10) are substituted, this expression is rewritten in terms of the reduced surface parameters P_α, Q_α, R_α, $\alpha = 1, \ldots, m$, as follows:

$$J = \frac{1}{2} \sum_{\alpha=1}^{m} \frac{W_\alpha}{p_\alpha^2 + q_\alpha^2 + 1} \left(\frac{f + r_\alpha}{f} \right)^2 \times \sum_{e_k\in E_\alpha} (\hat{e}_{k(1)} P_\alpha + \hat{e}_{k(2)} Q_\alpha + \frac{1}{f} \hat{e}_{k(3)} R_\alpha - \hat{e}_{k(3)})^2 \tag{30}$$

It follows that we can apply optimization as a single step by minimizing this J under the constraints (12).

Again, we resort to a trick to make the subsequent analysis easy. We replace r_α in the above equation by its estimate $\hat{r}_\alpha$ as before. Moreover, we also replace p_α and q_α in equation (30) by their estimates $\hat{p}_\alpha$, $\hat{q}_\alpha$, assuming that they are somehow available. Put

$$\frac{W_\alpha}{\hat{p}_\alpha^2 + \hat{q}_\alpha^2 + 1} \left(\frac{f + \hat{r}_\alpha}{f} \right)^2 = \frac{1}{\zeta_\alpha} \tag{31}$$

Then, ζ_α is a constant for face F_α whose value is yet to be determined.

Remark 6. For example, we can reason that if face F_α has a large depth value $\hat{r}_\alpha$, the orientation estimation of its boundary edges may not be very accurate, so a small weight W_α should be assigned. On the other hand, if $\hat{p}_\alpha$ and $\hat{q}_\alpha$ are close to zero, the variation of the 3D edge orientation in the Z-direction will cause only a small effect on its projection image, so it may be appropriate to assign a small weight W_α. In view of these considerations, one candidate is

$$W_\alpha = (\hat{p}_\alpha^2 + \hat{q}_\alpha^2 + 1) \left(\frac{f}{f + \hat{r}_\alpha} \right)^2 \tag{32}$$

For this weight, the constant ζ_α becomes simply

$$\zeta_\alpha = 1 \tag{33}$$

[13]In this paper, $(\boldsymbol{a}, \boldsymbol{b})$ $(= a_1b_1 + a_2b_2 + a_3b_3)$ denotes the inner product of vectors $\boldsymbol{a} = (a_1, a_2, a_3)$ and $\boldsymbol{b} = (b_1, b_2, b_3)$.

Of course, other factors—the projected areas of the faces, the projected lengths of the edges, etc.—can also be taken into account if considered to be relevant.

Thus, the problem reduces to minimization of J under the constraints (12) for all incidence pairs $(F_\alpha, V_i) \in R$. Since the absolute depth cannot be determined as long as only 3D edge orientation cues are used, we must give the depth Z, or equivalently the reduced depth z, to one vertex. Let that vertex be V_n. Since J is quadratic, and the constraints (12) are linear in the unknowns, the minimum is attained by solving a set of *linear* equations. The final result is given as follows.

Proposition 2. The reduced depths z_i of vertexes V_i, $i = 1, \ldots, n-1$, are given by solving the following $n + 3m + l - 1$ linear equations in $n + 3m + l - 1$ unknowns z_i, $i = 1, \ldots, n-1$, P_α, Q_α, R_α, $\alpha = 1, \ldots, m$, and $\Lambda_{\alpha i}$ for $(F_\alpha, V_i) \in R$:

$$x_i P_\alpha + y_i Q_\alpha + R_\alpha - z_i = 0, \quad (F_\alpha, V_i) \in R \tag{34}$$

$$A_{11}P_\alpha + A_{12}Q_\alpha + \frac{1}{f}A_{13}R_\alpha + \zeta_\alpha \times \sum_{i:(F_\alpha, V_i) \in R} x_i \Lambda_{\alpha i} = A_{13}, \quad \alpha = 1, \ldots, m \tag{35}$$

$$A_{21}P_\alpha + A_{22}Q_\alpha + \frac{1}{f}A_{23}R_\alpha + \zeta_\alpha \times \sum_{i:(F_\alpha, V_i) \in R} y_i \Lambda_{\alpha i} = A_{23}, \quad \alpha = 1, \ldots, m \tag{36}$$

$$A_{31}P_\alpha + A_{32}Q_\alpha + \frac{1}{f}A_{33}R_\alpha + \zeta_\alpha \times \sum_{i:(F_\alpha, V_i) \in R} f\Lambda_{\alpha i} = A_{33}, \quad \alpha = 1, \ldots, m \tag{37}$$

$$\sum_{\alpha:(F_\alpha, V_i) \in R} \Lambda_{\alpha i} = 0, \quad i = 1, \ldots, n-1 \tag{38}$$

where

$$A_{ab} \equiv \sum_{e_k \in E_\alpha} \hat{e}_{k(a)}\, \hat{e}_{k(b)}, \quad a, b = 1, 2, 3 \tag{39}$$

Remark 7. Again, 3D orientation estimates need not be given for all edges. If some edges are given no orientation estimates, the set E_α in the above equations is interpreted as *the set of edges for which the 3D orientation estimates are given.* If no boundary edges of face F_α are given their 3D orientation estimates, the corresponding terms are simply dropped from equation (30). This means that equations (35–37) are respectively replaced by equations (23).

Remark 8. In the orthographic limit $f \to \infty$, equations (35–37) are simply replaced by

$$A_{11}P_\alpha + A_{12}Q_\alpha + \zeta_\alpha \times \sum_{i:(F_\alpha, V_i) \in R} x_i \Lambda_{\alpha i} = A_{13}, \quad \alpha = 1, \ldots, m \tag{40}$$

$$A_{21}P_\alpha + A_{22}Q_\alpha + \zeta_\alpha \times \sum_{i:(F_\alpha, V_i) \in R} x_i \Lambda_{\alpha i} = A_{23}, \quad \alpha = 1, \ldots, m \tag{41}$$

$$A_{31}P_\alpha + A_{32}Q_\alpha + \zeta_\alpha \times \sum_{i:(F_\alpha, V_i) \in R} \Lambda_{\alpha i} = 0, \quad \alpha = 1, \ldots, m \tag{42}$$

4 Optimization of Shape from Motion

In this section, we apply our optimization technique to the shape-from-motion problem. Suppose we are given a sequence of images of a polyhedron moving in a scene. Let us assume that the point-to-point correspondence has already been detected, telling us which vertex corresponds to which between consecutive frames. Then, we can write down the equations that determine the 3D shape and motion from these observed finite displacements of the vertexes on the image plane (e.g., see Longuet-Higgins [19] and Tsai and Huang [39]). In the presence of noise, however, the computed 3D positions of vertexes incident to one face will not necessarily be coplanar.

One way to avoid this inconsistency is to choose, as unknowns, not the 3D positions of vertices but the *surface gradients* of the object faces. Then, computed solutions necessarily have planar faces, and we can obtain a 2½D sketch. Since noise is unavoidable, the computed surface gradients are in general inaccurate; the resulting 2½D sketch does not necessarily define a consistent polyhedron. This difficulty is, however, overcome by the optimization technique discussed in the preceding section.

Consider a face which has four or more corners. If the image velocities are observed at at least four vertexes, and if no three of them are collinear, the hypothetical *optical flow* is uniquely determined because optical flow resulting from planar surface motion contains only eight parameters, which we call *flow parameters*. Once these eight flow parameters are estimated, say by the least-square fitting, we can compute the surface gradient and the 3D motion of the face.[14] In particular, we can *analytically* compute the rotation velocity $(\omega_1, \omega_2, \omega_3)$ and the surface gradient (p, q) (appendix D), and no indeterminate parameters are involved. Although two sets of solutions exist, the spurious solutions can be discarded if two or more faces of the polyhedron are observed, since the true rotation velocity $(\omega_1, \omega_2, \omega_3)$ must be common to all faces. Thus, we can estimate the surface gradient (p, q) for the faces which have four or more corners. Then, the optimization technique is applied to the resulting 2½D sketch.

Remark 9. The theory of optical flow is based on the assumption that *instantaneous* velocities are observed on the image plane. If a sequence of images is given, the instantaneous velocities must be approximated by the *displacements* between consecutive frames by taking the time lapse between frames as unit time. This approximation introduces considerable error into the subsequent processes even if measurements on individual images are very accurate.

Remark 10. The rotation velocity $(\omega_1, \omega_2, \omega_3)$ should be common to all the faces. In the presence of noise, however, we may not be able to find a strictly common rotation velocity. It follows that we need some "clustering" technique in the three-dimensional $\omega_1\omega_2\omega_3$-space to find the one most likely to be the common rotation velocity.

Remark 11. If the surface gradients are computed for individual faces, we can obtain a 2½D sketch by assigning them to either the first image frame or the second, or we can assign them to the "middle" image obtained by connecting the midpoints of the vertex displacements.

[14]For example, see Kanatani [12], Longuet-Higgins [20], Negahdaripour and Horn [26], and Subbarao and Waxman [33]. Also see Buxton et al. [3,4], Kanatani [11], Koenderink et al. [17,18], Longuet-Higgins and Prazdney [21], Subbarao [34], Waxman et al. [40,41] for various aspects of optical flow analysis.

Remark 12. If the object has a triangular face, its surface gradient is not assigned. However, our optimization works even if some faces do not have assigned surface gradient estimates (cf. Remarks 4 and 7).

Example 1. Consider the two images of figures 6a,b. Let us label the vertexes and the faces as shown in figure 7a.[15] The displacements of the vertexes are shown in figure 7b. Regarding the displacements as instantaneous velocities, and applying the procedure described above, we can reconstruct the 3D shape uniquely up to a single scale factor. If surface gradient estimates are assigned to the first image, and equation (16) (i.e., ζ_α of equation (17) is used for the weight W_α, the 3D shape shown in figure 8 is obtained. Figure 8a shows the top view (orthographic projection onto the YZ-plane), while figure 8b shows the side view (orthographic projection onto the ZX-plane). In spite of the presence of noise and the inaccuracy of the estimated surface gradient values, the final result is fairly correct.

5 Optimization of Rectangularity Heuristics

In this section, we study the 3D reconstruction of a polyhedron from a single image by invoking the *rectangularity hypothesis*. The 3D orientation of a corner can be computed if its three edges are assumed to meet perpendicularly. Since many man-made objects—buildings, machine parts, furniture, etc.—have rectangular corners, the study of this assumption is very useful for practical purposes. Besides, this hypothesis is also very natural from the viewpoint of human perception psychology (cf. Barnard [2] and Mackworth [22]) If one corner is known to be rectangular and has three visible edges, we can compute the 3D orientations of the three edges up to the mirror image (cf. Kanatani [10] and Shakunaga and Kaneko [32]). Here, we use the formulation of Kanatani [15] (appendix E). Each edge orientation indicates the surface normal to the face defined by the other two edges. Hence, we can determine the surface gradient of the three faces.

For each rectangular corner, we obtain two mirror image solutions. However, it is usually easy to choose the true solution. For example, we can apply a scheme of line-drawing interpretation.[16] We may also use other

[15]The condition (C1) of appendix C is satisfied, and hence the incidence structure is nonsingular.

[16]For example, the Huffman-Clowes edge labeling (cf. Clowes [5] and Huffman [9]) and Kanatani's scheme (Kanatani [10]).

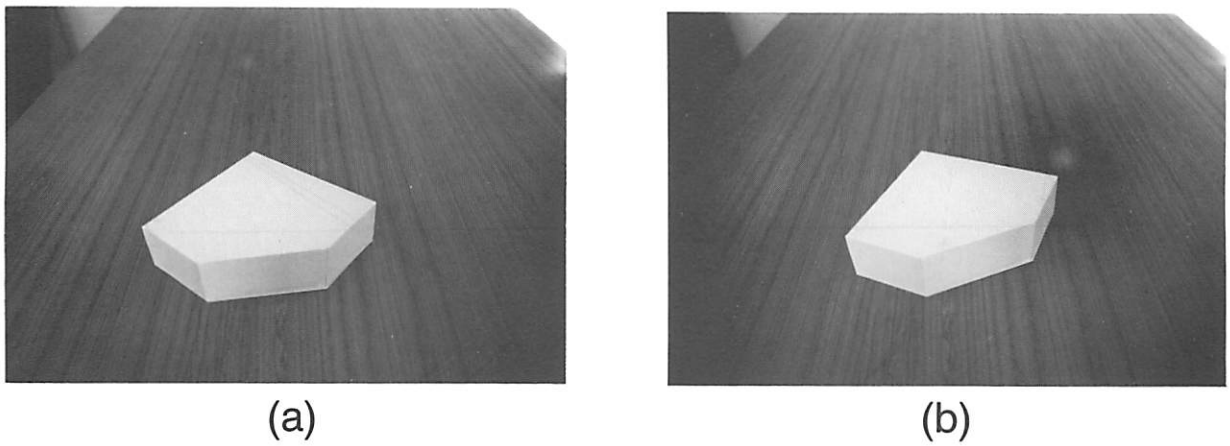


Fig. 6. Two object images.

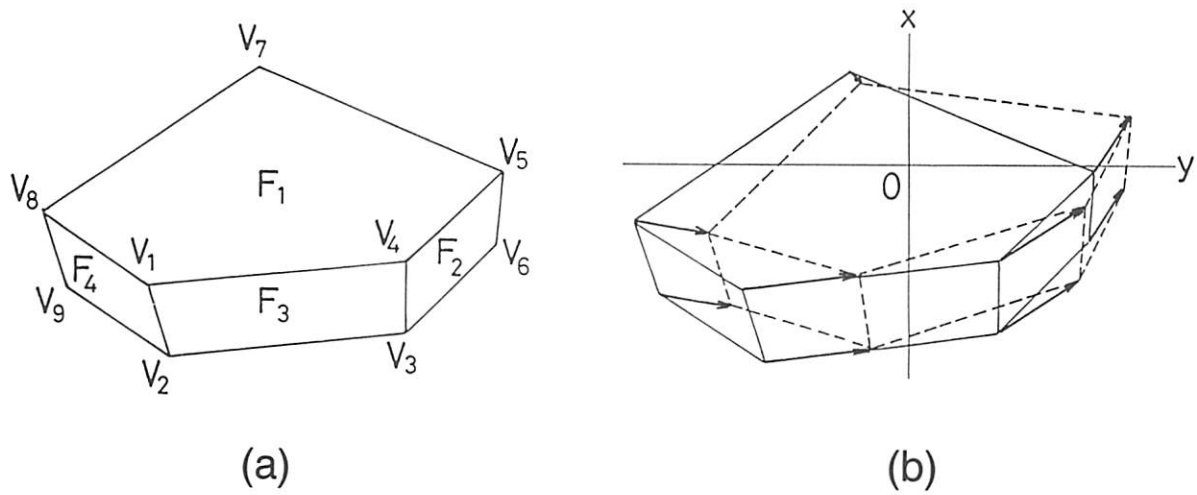


Fig. 7. (a) Labeled line drawing for the object image of figure 6a. (b) Displacements of vertexes obtained from the two images of figure 6.

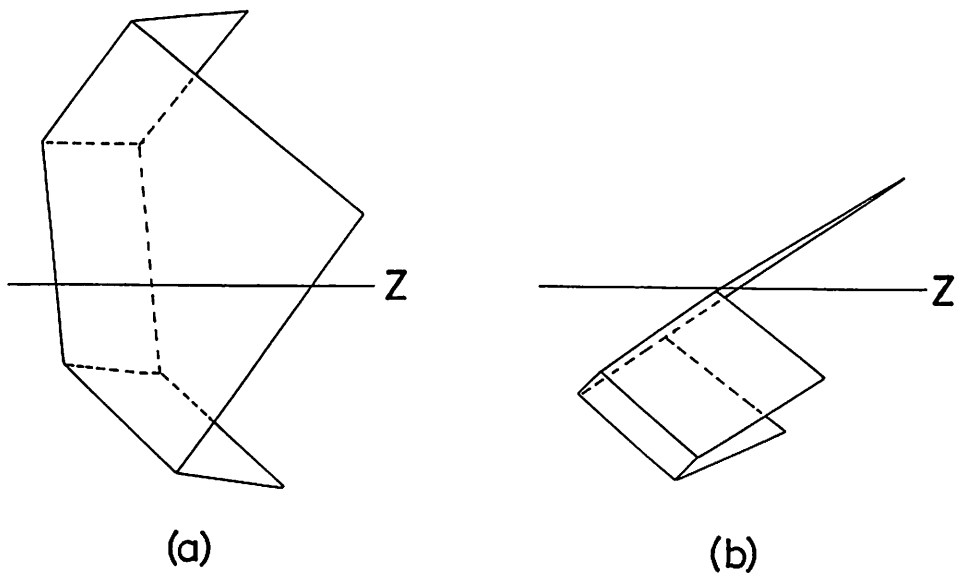


Fig. 8. The 3D shape reconstructed from figure 6: (a) The top view (orthographic projection onto the *YZ*-plane). (b) The side view (orthographic projection onto the *ZX*-plane).

sources of information if available—range sensing, shading information, etc. If neither of the two mirror image solutions can be eliminated, we can simply retain both and produce multiple solutions. In the following, let us assume for simplicity that the true solution can be distinguished from its mirror image.

If a face has two or more rectangular corners with three visible edges, and multiple surface gradients are obtained, their average is assigned. Then the optimization technique is applied to the resulting in a 2½D sketch. Alternatively, the optimization technique can be directly applied to the 2½D sketch with 3D edge orientation estimates. Recall that all faces and edges need not be given estimates; our optimization works if some faces or edges lack their orientation estimates (cf. Remarks 4 and 7).

Thus, the remaining question is how to find rectangular corners. Theoretically, this is an impossible problem, since all we have is a single image. Here, we resort to a heuristic: As many corners are assumed to be rectangular as possible *unless inconsistency results by assuming so.* This means that all we need to do is reject those corners that are definitely nonrectangular. The first criterion available is the following *rectangularity test.* For a corner image with three visible edges, consider its *canonical image,* (that is, the corner image defined by its *canonical angles,* (cf. appendix E).[17] The corner cannot be rectangular if the three angles made by the three edges satisfy one of the following conditions (see Kanatani [15] for the proof):

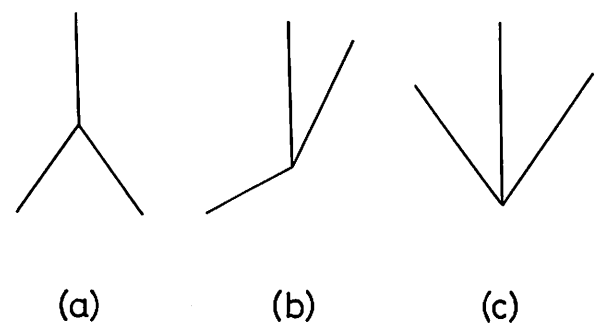


Fig. 9. Rectangularity test: If these configurations arise in the *canonical* position, none of these can be the projection of a rectangular corner.

1. One is an acute angle, and the remaining two are obtuse angles (figure 9a).
2. One is an acute angle, another is an obtuse angle, and the other is larger than π (figure 9b).
3. Two are acute angles, and the other is larger than $3\pi/2$ (figure 9c).

[17]This is the image of the corner we would obtain if the camera were rotated around the center of the lens so that the corner vertex coincides with the image origin (cf. Kanatani [15]).

The next critierion available is the *compatibility test* for two corners. We choose two corners which share at least one face, and compute the 3D edge orientations and the surface gradients at these two corners, assuming that both are rectangular corners. If this assumption is correct, the computation must predict an identical 3D orientation for the connecting edge (if the two corners are connected) and identical surface gradients for the common faces (within some fixed tolerance). If not, we say that they are *incompatible* as rectangular corners.

Then, we form *maximal compatible sets* of corners in such a way that as many corners are included as possible unless imcompatible pairs arise among them. Then, assuming that the corners belonging to each set are all rectangular, we end up with as many 2½D sketches as the number of these maximal compatible sets.

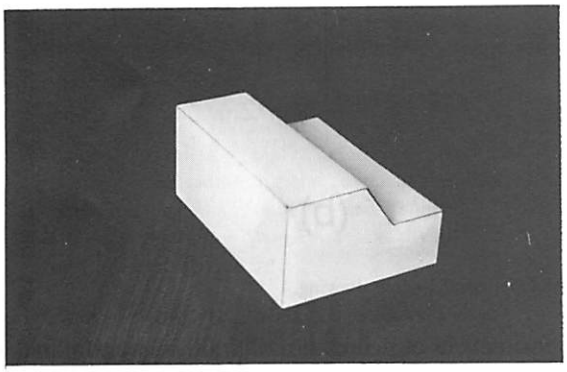
Example 2. Figure 10a is a real image of a polyhedron. Suppose the line drawing of figure 10b is obtained, and its vertexes and faces are labeled as indicated in the figure.[18] Vertexes V_1, V_3, V_4, V_6, V_{10} have three visible edges.[19] First the rectangularity test rejects vertexes V_3 as nonrectangular. The rest are combined into pairs sharing at least one common face, resulting in $\{V_1, V_4\}$, $\{V_1, V_{10}\}$, $\{V_4, V_6\}$, $\{V_4, V_{10}\}$, $\{V_6, V_{10}\}$. Then, we perform the compatibility test by computing the canonical angles and estimating the surface gradients by the procedure of Kanatani [15] (appendix E). All the above pairs pass this test. So, all of these corners are assumed to be rectangular. Then, surface gradient estimates are assigned to the faces to which these corners are incident (the average is taken if necessary), and the optimization technique is applied to the resulting 2½D sketch. If equation (16) (i.e., ζ_α of equation (17)) is used for the weight W_α, the 3D shape indicated in figure 11 is obtained. Figure 11a shows the top view (orthographic production onto the *YZ*-plane) while figure 11b shows the side view (orthographic projection onto the *ZX*-plane).

Example 3. Figure 12a is another real image of a polyhedron. Suppose the line drawing of figure 12b is obtained, and its vertexes and faces are labeled as indicated in the figure.[20] Vertexes V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_{10}, have three visible edges. This time, the rectangularity test cannot reject any of these vertexes as definitely nonrectangular. The compatibility test tells us that these vertexes are split into two compatible groups $\{V_2, V_7, V_8, V_{10}\}$, $\{V_3, V_4, V_5, V_6\}$. Thus, we obtain two solutions if we separately assume that the

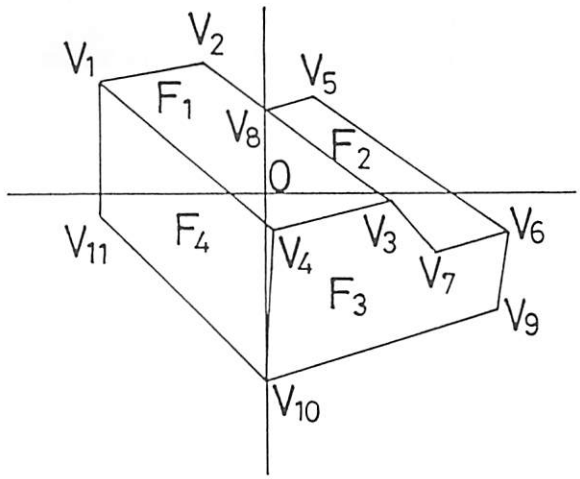
[18]The condition (C1) of appendix C is satisfied, and hence this incidence structure is nonsingular.

[19]Vertex V_8 is judged as a T-junction, and hence it is incident to F_2 alone.

[20]The incidence structure is also nonsingular.



(a)



(b)

Fig. 10. (a) A polyhedron image. (b) The labeling of its line drawing.

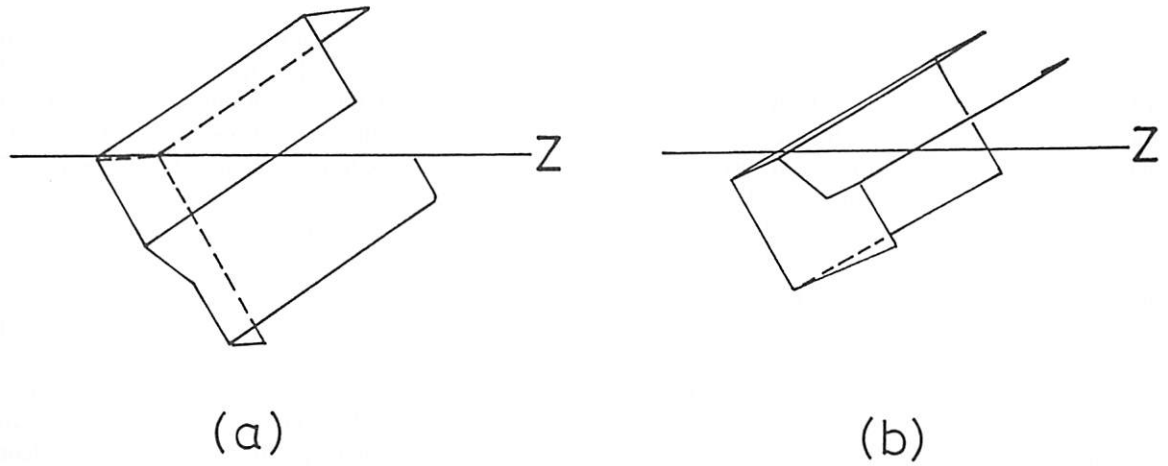


Fig. 11. The 3D shape reconstructed from figure 10: (a) The top view (orthographic projection onto the *YZ*-plane). (b) The side view (orthographic projection onto the *ZX*-plane).

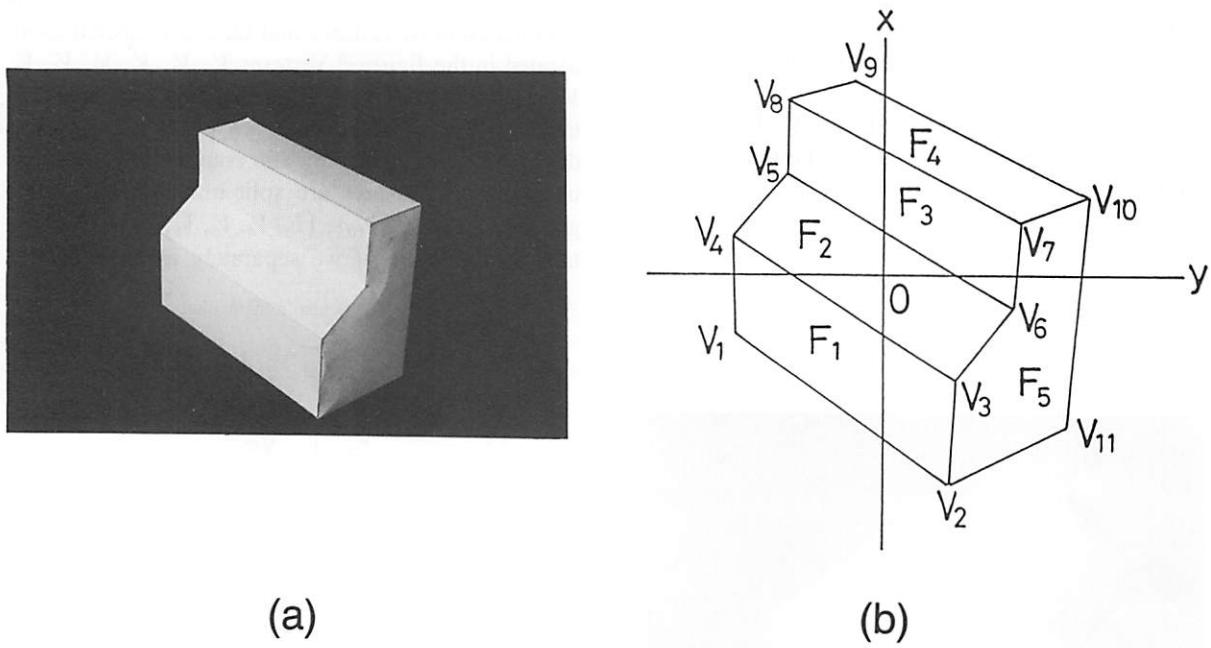


Fig. 12. (a) A polyhedron image. (b) The labeling of its line drawing.

corners of each group are rectangular. Applying the optimization technique, we can reconstruct the two 3D shapes shown in figures 13 and 14. In both of them, (a) shows the top view (orthographic projection onto the *YZ*-plane) while (b) shows the side view (orthographic projection onto the *ZX*-plane).

Remark 13. The objects shown in figures 10a and 12a are actually the same but placed differently, and the reconstructions of figures 11 and 13 are correct. From this observation, we notice that false solutions can be removed if two images of the same object are obtained from different angles, since the true solutions must have an identical shape. Alternatively, we can eliminate false solutions if some global 3D characteristic, say the *aspect ratio* (i.e., the maximum of the ratio of the "height" measured in one direction over the "width" measured perpendicularly to it), is given or estimated beforehand.

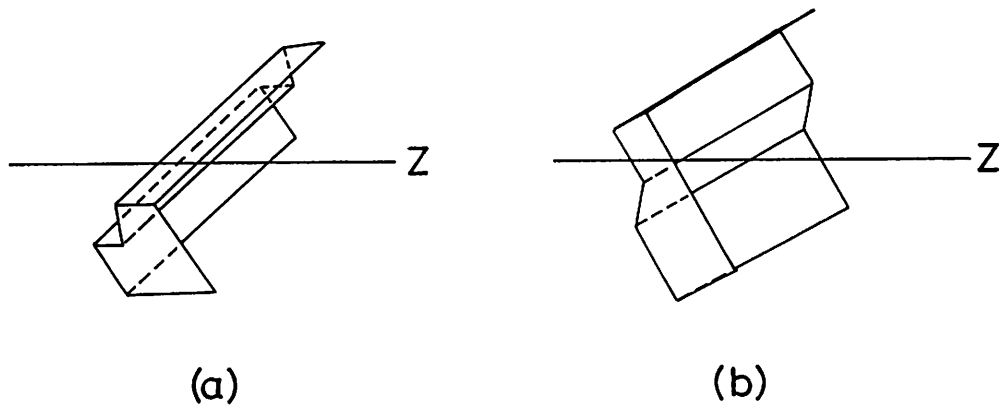


Fig. 13. One 3D shape reconstructed from figure 12: (a) The top view (orthographic projection onto the *YZ*-plane). (b) The side view (orthographic projection onto the *ZX*-plane).

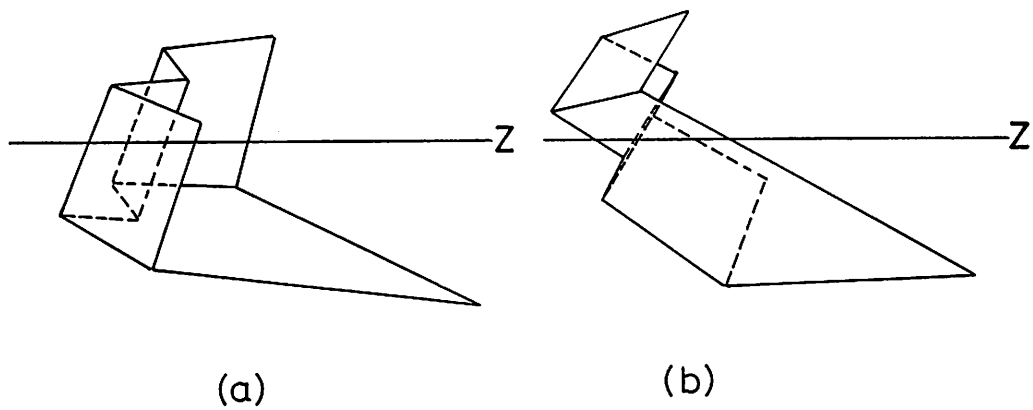


Fig. 14. Another 3D shape reconstructed from figure 12: (a) The top view (orthographic projection onto the *YZ*-plane). (b) The side view (orthographic projection onto the *ZX*-plane).

6 Optimization of Parallelism Heuristics

Now, let us study 3D shape recovery based on the *parallelism hypothesis* coupled with the optimization technique. If two lines in the image are interpreted to be projections of parallel lines in the scene, they define a *vanishing point* on the image plane, which, as is well known, determines the 3D orientation of these lines. Since foreshortening greatly helps humans perceive 3D depth, this reasoning is also very natural for humans (cf. Haralick [7] and Mulgaonkar et al. [25]).

Suppose a polyhedron image is observed. If we can find a set of edges that are parallel in the scene, their 3D orientation is computed from their vanishing points by the equations shown in appendix F. Hence, if a parallel edge-finding algorithm is available, we can obtain a 2½D sketch with estimated 3D edge orientations. The 3D edge orientations computed from vanishing points may not be consistent with each other in the presence of noise, but this inconsistency can be overcome by the optimization technique of section 3.

Thus, it remains to construct an algorithm for finding parallel edges. A naive heuristic is to group together those edges that are *nearly parallel* on the image plane. However, if two edges are far apart on the image plane, they can be parallel in the scene even if they make a large angle of the image plane. On the other hand, if two edges meet at a corner of the object, they cannot be parallel however small the angle between them is. In general, two parallel edges in the scene can be projected onto lines making any angle, depending on the surface gradient of the plane on which they lie.

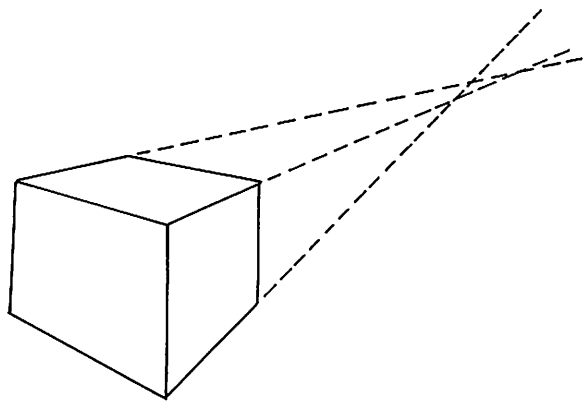


Fig. 15. Concurrency test: Three or more edges are judged as parallel if they are concurrent on the image plane when extended. In the presence of noise, however, they may not necessarily intersect at a single vanishing point. How can we tolerate the error to judge concurrency?

Again, we adopt the same heuristic strategy we used for the rectangularity hypothesis. Namely, as many edge sets are assumed to be parallel as possible *unless inconsistency results by assuming so.* First, there exists a strong constraint on parallelism: Projections of parallel lines in the scene must be concurrent on the image plane, defining a common vanishing point. In view of this, we apply the *concurrency test:* If three or more edges are concurrent on the image plane, they are judged to be parallel in the scene. This assumption is reasonable because it is very unlikely (though not entirely impossible) that three or more nonparallel edges happen to be concurrent when projected.

However, a serious problem arises. As we have already pointed out many times, the edges we observe on the image plane may not be accurate. As a result, parallel edges, when extended, may not necessarily intersect at a single point (figure 15). What criterion should we use to accept them as concurrent or reject them as nonconcurrent? One solution is to decide that lines are concurrent if the maximum separation among their mutual intersections is smaller than a threshold value ϵ.

The trouble is that the threshold value cannot be fixed. For instance, if two lines intersect at a point far apart from the image origin, slight displacements of them will cause very large displacements of the intersection. Hence, the threshold value ϵ must be taken to be very large, but then we cannot distinguish intersections near the image origin. Thus, the threshold value must depend on the distance of the intersection point from the image origin.

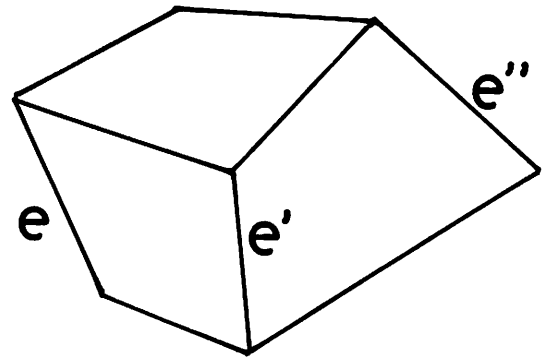


Fig. 16. Only those edge pairs that share common faces are checked, because it is highly unlikely that two edges belonging to different faces are parallel, yet no other edges are parallel to them.

We must also confine the computation into the *finite* domain so that computational overflow does not occur. One way to do so is write all relationships of points and lines in terms of unit vectors or *normalized homogeneous coordinates* (appendix F). This is easily done if we use, instead of the usual image plane, the *image sphere* (or the *Gaussian sphere*) of radius f centered at the viewpoint [23]. However, we must note that the use of the image sphere is only for *visualization.* All computation is done over the original image data; we need not *generate* a new image over the image sphere. One algorithm of the concurrency test is given in appendix G.

After the concurrency test has detected sets of three or more parallel edges, we sort out, from among the remaining edges, those pairs that share common faces. It is reasonable to check only these pairs, because it is highly unlikely that two edges belonging to different faces are parallel, yet no other edges are parallel to them (figure 16). To these candidate pairs, we apply the following two tests.

The first test is the *parallelogram test:* If two pairs of parallel lines lying on the same plane are projected onto half-lines starting from their respective vanishing points, they must intersect with each other at exactly *four* points on the image plane. This constraint comes from the fact that two pairs of parallel lines on a plane must define a parallelogram. For example, suppose edges e_1, e_2 are already judged to be parallel. In figure 17a,

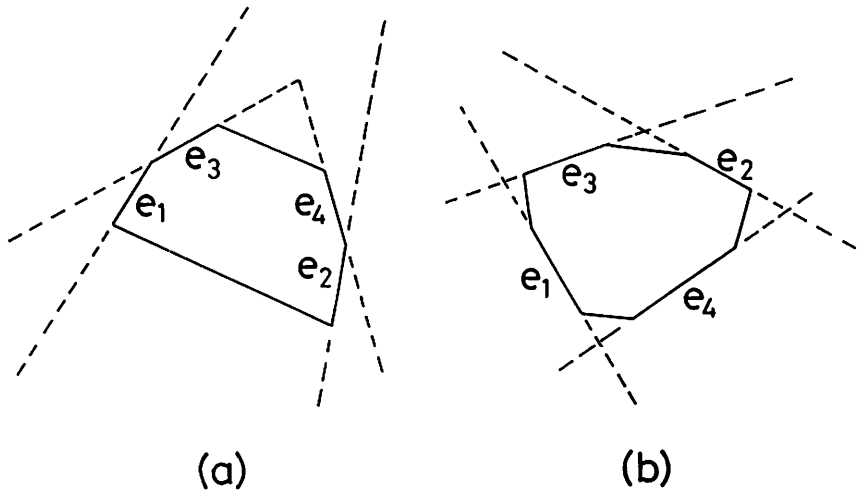


Fig. 17. Parallelogram test: Two pairs of parallel edges regarded as half-lines starting from their respective vanishing points must intersect each other at exactly *four* intersection points. If edges e_1, e_2 are parallel, edges e_3, e_4 cannot be parallel in (a) but can be parallel in (b).

edges e_3, e_4 cannot be parallel in the scene because they do not define a parallelogram with edges e_1, e_2, while edges e_3, e_4 of figure 17b pass this test.

The second test is the *collinearity test:* If three or more sets of parallel lines belong to the same face, their vanishing points must be collinear. Namely, a candidate pair of edges cannot be parallel if their intersection is not on the vanishing line already established from other vanishing points (within some tolerance) (figure 18).

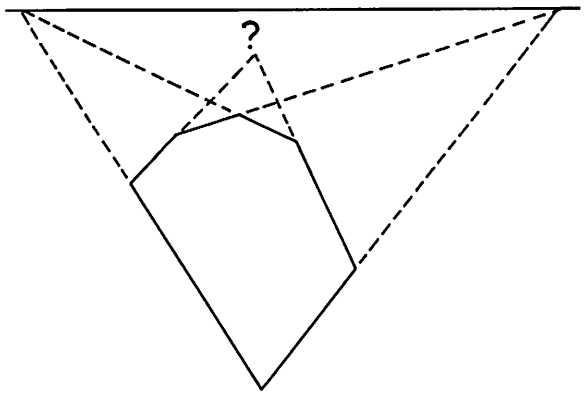


Fig. 18. Collinearity test: The vanishing points of parallel edges belonging to the same face must be collinear (within some tolerance).

Thus, those edge pairs that have passed these two tests are assumed to be parallel in the scene. However, if one edge appears in multiple pairs, we choose, by invoking the *vanishing point heuristic,* the pair whose intersection is located farthest away from the image origin O (figure 19).

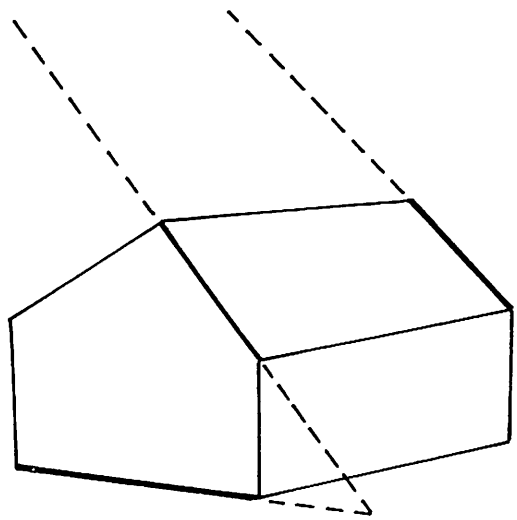


Fig. 19. Vanishing point heuristic: Two edges are more likely to be parallel in the scene if their intersection is farther away from the image origin O.

Example 4. Figure 20a is a real image of a polyhedron. Suppose the line drawing of figure 21b is obtained. Its vertexes, edges, and faces are labeled as indicated in this figure.[21] Applying the concurrency

[21]The condition (C1) of appendix C is satisfied and hence this incidence structure is nonsingular.

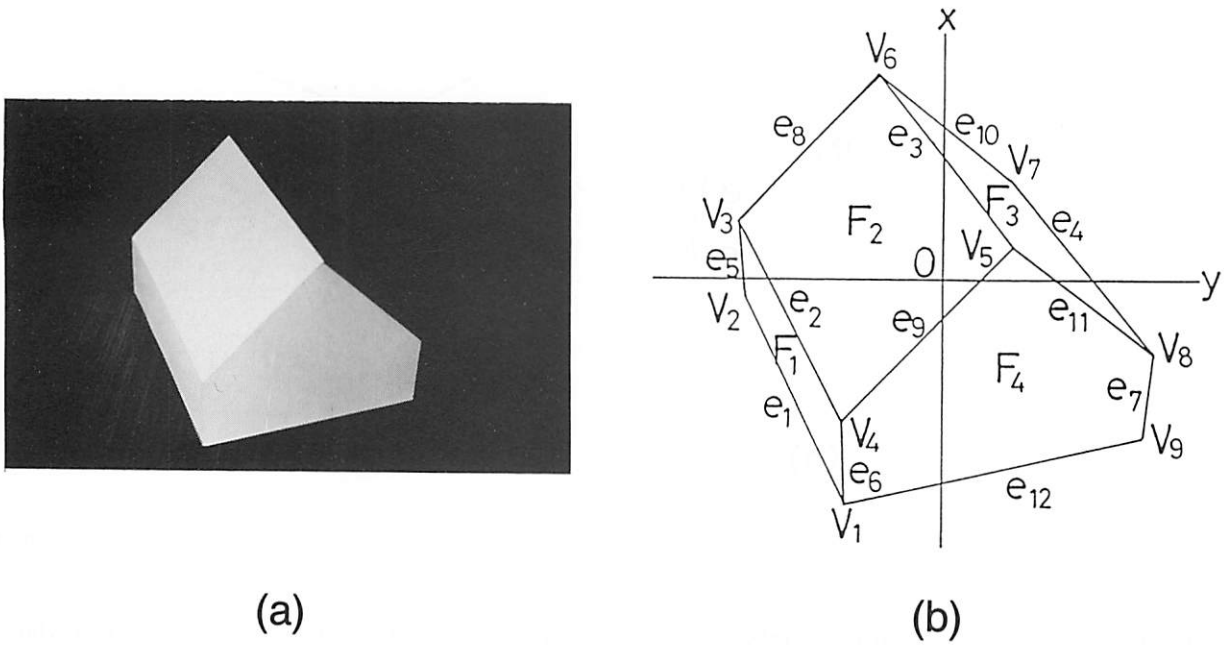


Fig. 20. (a) A polyhedron image. (b) The labeling of its line drawing.

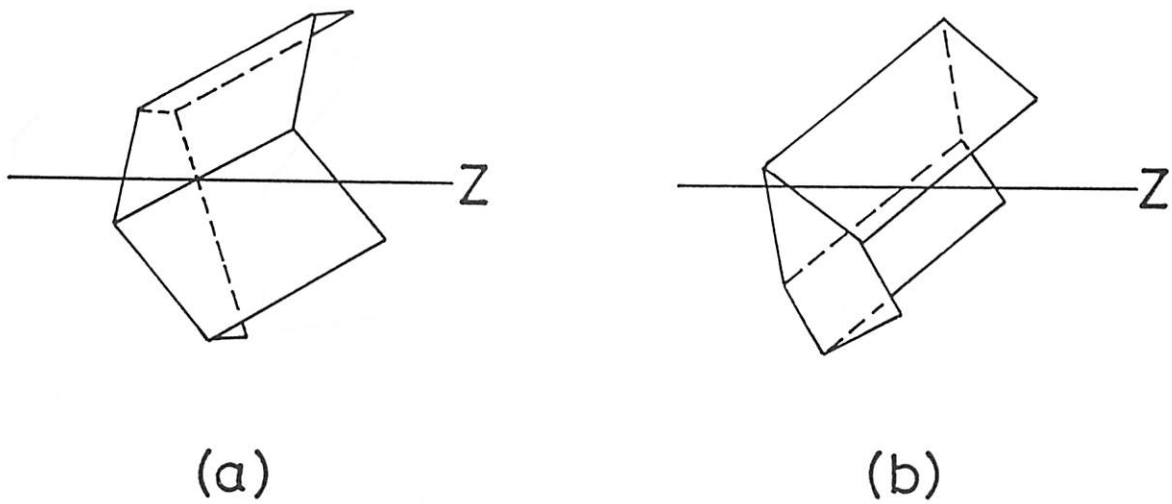


Fig. 21. The 3D shape reconstructed from figure 19: (a) The top view (orthographic projection onto the *YZ*-plane). (b) The side view (orthographic projection onto the *ZX*-plane).

test, we detect the following set of parallel edges: $\{e_1, e_2, e_3, e_4\}$, $\{e_5, e_6, e_7\}$. From among the remaining edges, those pairs that share common faces but no common vertexes are the next candidates for parallel pairs: $\{e_8, e_9\}$, $\{e_{10}, e_{11}\}$, $\{e_9, e_{12},\}$, $\{e_{11}, e_{12}\}$. All of these pairs pass both the parallelogram test and the collinearity test. Edge e_9 belongs to two pairs $\{e_8, e_9\}$, $\{e_9, e_{12}\}$. Invoking the vanishing point heuristic, we choose the former, because its intersection is located farther away than that of the latter. Similarly, edge e_{11} belongs to two pairs $\{e_{10}, e_{11}\}$ and $\{e_{11}, e_{12}\}$, but we choose the former. Thus, we conclude that the sets of parallel edges are $\{e_1, e_2, e_3, e_4\}$, $\{e_5, e_6, e_7\}$, $\{e_8, e_9\}$, $\{e_{10}, e_{11}\}$. Applying the optimization technique, we can recover the 3D shape up to a single scale factor. If equation (32) (i.e., ζ_α of equation (33), is used for the weight W_α, the 3D shape shown in figure 21 is obtained. Figure 21a shows the top view (orthographic projection onto the *YZ*-plane), while figure 21b shows the side view (orthographic projection onto the *ZX*-plane).

7 Concluding Remarks

In this paper, we have presented a scheme of *optimization* for computing a consistent polyhedron shape from inconsistent image data. Our formulation is based on the algebraic expressions of the *constraints* on the *incidence structure* of a polyhedron line drawing. This ideas was first proposed and studied in detail by Sugihara [35–38]. However, his formulation is based on the assumption that the projection is orthographic, although he pointed out that the same algebraic analysis applies to perspective projection by changing variables. Here, we have presented a fully unified approach. Namely, we introduced the *reduced depth* and *reduced surface* parameters which enable us to treat orthographic and perspective projections in a single framework.

Sugihara [37,38] also proposed an optimization scheme to construct a consistent polyhedral shape that fits observed image data on the average. His approach is first to construct the solution space defined by the constraints on the incidence structure (i.e., the subspace whose dimensionality equals the degree of freedom of the incidence structure) and then to perform an *unconstrained nonlinear* optimization over this solution space, thus requiring an appropriate initial guess and an iteractive search algorithm.

In our approach, we do not construct the solution space. Instead, we perform a *constrained quadratic* optimization. By introducing Lagrangian multipliers corresponding to the constraints on the incidence structure, the optimal solution is immediately obtained by solving a set of *linear* equations: no initial guesses or iterations are necessary.

We first applied our technique to the shape-from-motion problem and demonstrated how a consistent object can be reconstructed from inaccurate data. Then, we applied it to 3D shape recovery of polyhedra from a single image. First, we adoped the rectangularity hypothesis, assuming corners to be rectangular unless inconsistency results. We have presented a heuristic strategy for finding rectangular corners. Next, we adopted the parallelism hypothesis, assuming edges to be parallel unless inconsistency results. We also presented a heuristic strategy for finding parallel edges. For each problem, we showed examples based on real images.

As we have shown in these examples, our optimization scheme requires complete line drawings over which 2½D sketches are optimized. This may be a problem if the line drawings are supposed to be edge images of "real" polyhedra. However, this is not a problem when smooth surfaces are approximated by polyhedra, since surface tessellation is completely arbitrary: Our optimization equally works whether the line drawings are edge images or arbitrary tessellations.

We have also collected in appendixes various mathematical facts and techniques relevant to 3D interpretation of images in a concise form readily available for actual computation.

Acknowledgements

The author thanks Azriel Rosenfeld of the University of Maryland and Kokichi Sugihara of the University of Tokyo for helpful discussions. Part of this work was supported by Yazaki Memorial Foundation for Science and Technology and Inamori Foundation.

References

1. J. (Y.) Aloimonos and M. Swain, "Shape from patterns: Regularization," *Intern. J. Comput. Vision* 2(2): 171–187, 1988.
2. S.T. Barnard, "Choosing a basis for perceptual space," *Comput. Vision Graphics Image Process.* 29:87–99, 1985.

3. B.F. Buxton and H. Buxton, "Monocular depth perception from optical flow by space time signal processing," *Proc. Roy. Soc. London* B218:27–47, 1983.
4. B.F. Buxton, D.W. Murray, H. Buxton and N.S. Williams, "Structure-from-motion algorithms for computer vision on an SIMD architecture," *Comput. Phys. Comm.* 37:273–280, 1985.
5. M.B. Clowes, "On seeing things," *Artificial Intelligence* 2:79–116, 1971.
6. R.T. Frankot and R. Chellapa, "A method for enforcing integrability in shape from shading algorithms," *IEEE Trans. PAMI* 10(4): 439–451, 1988.
7. R.M. Haralick, "Using perspective transformations in scene analysis," *Comput. Graphics Image Process.* 13:191–221, 1980.
8. B.K.P. Horn and M.J. Brooks, "The variational approach to shape from shading," *Comput. Vision Graphics Image Process.* 33:174–208, 1986.
9. D.A. Huffman, "Impossible objects as nonsense sentences," *Machine Intelligence*, vol. 6 (eds. B. Meltzer and D. Michie). Edinburgh Univ. Press: Edinburgh, U.K., pp. 295–323, 1971.
10. K. Kanatani, "The constraints on images of rectangular polyhedra," *IEEE Trans. PAMI* 8(2):456–463, 1986.
11. K. Kanatani, "Structure and motion from optical flow under orthographic projection," *Comput. Vision Graphics Image Process.* 35:181–199, 1986.
12. K. Kanatani, "Structure and motion from optical flow under perspective projection," *Comput. Vision Graphics Image Process.* 38:122–146, 1987.
13. K. Kanatani, "Coordinate rotation invariance of image characteristics for 3D shape and motion recovery," *Proc. 1st Int. Conf. Comput. Vision*, London, pp. 55–64, June 1987.
14. K. Kanatani, "Camera rotation invariance of image characteristics," *Comput. Vision Graphics Image Process.* 39:328–354, 1987.
15. K. Kanatani, "Constraints on length and angle," *Comput. Vision Graphics Image Process.* 41:28–42, 1988.
16. K. Kanatani, "Transformation of optical flow by camera rotation," *IEEE Trans. PAMI* 10:131–143, 1988.
17. J.J. Koenderink and A.J. van Doorn, "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer," *Optica Acta* 22:773–791, 1975.
18. J.J.Koenderink and A.J. van Doorn, "Local structure of movement parallax of the plane," *J. Opt. Soc. Am.* 66:717–723, 1976.
19. H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature* 293:133–135, 1981.
20. H.C. Longuet-Higgins, "The visual ambiguity of a moving plane," *Proc. Roy. Soc. London* B 223:165–175, 1984.
21. H.C. Longuet-Higgins and K, Prazdney, "The interpretation of a moving retinal image," *Proc. Roy. Soc. London* B 208:385–397, 1980.
22. A.K. Mackworth, "Model-driven interpretation in intelligent vision system," *Perception* 5:349–370, 1976.
23. M.J. Magee and J.K. Aggarwal, "Determining vanishing points from perspective images," *Comput. Vision Graphics Image Process.* 26:256–267, 1984.
24. D. Marr, *Vision: A Computational Investigation into the Human Respresentation and Processing of Visual Information*, Freeman: San Francisco, 1982.
25. P.G. Mulgaonkar, L.G. Shapiro, and R.M. Haralick, "Shape from perspective: A rule-based approach," *Comput. Vision Graphics Image Process* 36:298–320, 1986.
26. S. Negahdaripour and B.K.P. Horn, "Direct passive navigation," *IEEE Trans. PAMI* 9(1):168–176, 1987.
27. T. Poggio and C. Koch, "Ill-posed problems in early vision: From computational theory to analogue networks," *Proc. Roy. Soc. London* B 226:303–323, 1985.
28. T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature* 317(26):314–319, 1983.
29. A.N. Tikhonov and V.Y. Arsenin, *Solutions of Ill-posed Problems*, Winston: Washington, DC, 1977.
30. S.B. Pollard, J. Porrill, J.E.W. Mayhew, and J.P. Frisby, "Matching geometrical descriptions in three-space," *Image Vision Computing* 5:73–78, 1987.
31. J. Porrill, S.B. Pollard, and J.E.W. Mayhew, "Optimal combination of multiple sensors including stereo vision," *Image Vision Computing* 5:174–180, 1987.
32. T. Shakunaga and H. Kaneko, "Perspective angle transformation and its application to 3-D configuration recovery," *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Miami Beach, FL, pp. 594–601, June 1986.
33. M. Subbarao and A.M. Waxman, "Closed form solution to image flow equations for planar surface in motion," *Comput. Vision Graphics Image Process.* 36:208–228, 1986.
34. M. Subbarao, "Interpretation of image flow: Rigid curved surfaces in motion," *Intern. J. Comput. Vision.* 2:77–96, 1988.
35. K. Sugihara, "Mathematical structures of line drawing of polyhedrons—toward man-machine communication by means of line drawings," *IEEE Trans. PAMI* 4:458–469, 1982.
36. K. Sugihara, "A necessary and sufficient condition for a picture to represent a polyhedral scene," *IEEE Trans. PAMI* 6:578–586, 1984.
37. K. Sugihara, "An algebraic approach to shape-from-image problems," *Artificial Intelligence* 23:59–95, 1984.
38. K. Sugihara, *Machine Interpretation of Line Drawings*, MIT Press: Cambridge, MA, 1986.
39. R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. PAMI* 6(1):13–27, 1984.
40. A.M. Waxman and S. Ullman, "Surface structure and three-dimensional motion from image flow kinematics," *Intern. J. Robotics Res.* 4:72–94, 1985.
41. A.M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation and global image flow: Planar surfaces in motion," *Intern. J. Robotics Res.* 4:95–108, 1985.

Appendix A: Integrability Condition of a Smooth Surface

Let us briefly consider case (1)—the 2½D sketch with densely estimated surface gradient. Suppose the surface gradient (p, q) is estimated as a smooth function over a region of the image plane. Then, we are questioning the *integrability condition:* Does a smooth surface that has the specified surface gradient—$p(x, y)$, $q(x, y)$—exist? Suppose the region corresponding to the surface is *simply connected* (i.e., with no *holes*).

We are seeking a solution in the form of $Z = Z(X, Y)$ such that $p(x, y)$ and $q(x, y)$ are respectively equal to the values of $\partial Z/\partial X$ and $\partial Z/\partial Y$ evaluated at the point (X, Y, Z) corresponding to point (x, y). This means that we are looking for a surface $Z = Z(X, Y)$ such that

$$dZ = p\left(\frac{fX}{f+Z}, \frac{fY}{f+Z}\right) dX + q\left(\frac{fX}{f+Z}, \frac{fY}{f+Z}\right) dY \quad \text{(A1)}$$

The following integrability condition is well known in the theory of differential equations: The necessary and sufficient condition that there exists a surface along which differentials dX, dY, dZ satisfy

$$A(X, Y, Z)\, dX + B(X, Y, Z)\, dY + C(X, Y, Z)\, dZ = 0 \quad \text{(A2)}$$

is

$$\left(\frac{\partial C}{\partial Y} - \frac{\partial B}{\partial Z}\right) A + \left(\frac{\partial A}{\partial Z} - \frac{\partial C}{\partial X}\right) B + \left(\frac{\partial B}{\partial X} \quad \frac{\partial A}{\partial Y}\right) C = 0 \quad \text{(A3)}$$

(*Frobenius' theorem*). From this theorem, we find that the desired surface exists if and only if the following condition is satisfied:

$$\frac{\partial q}{\partial x} - \frac{\partial p}{\partial y} = \frac{1}{f}\left[\left(p\frac{\partial q}{\partial x} - q\frac{\partial p}{\partial x}\right) x + \left(p\frac{\partial q}{\partial y} - q\frac{\partial p}{\partial y}\right) y\right] \quad \text{(A4)}$$

If we take the orthographic limit $f \to \infty$, we obtain

$$\frac{\partial q}{\partial x} - \frac{\partial p}{\partial y} = 0 \quad \text{(A5)}$$

which is a well known result. Thus, whether the projection is orthographic or perspective, the surface gradient (p, q) cannot be assigned arbitrarily to the 2½D sketch.

Violation of this integrability condition for real data has been realized by many researchers to be one of the major obstacles for reconstructing surface shapes from densely estimated surface gradient cues, and various attempts have been made to force the integrability conditions (cf. Frankot and Chellapa [6] and Horn and Brooks [8]). Such attempts can also be viewed as a kind of *regularization* (cf. Aloimonos and Swain [1], Poggio et al. [27,28], and Tikhonov and Arsenin [29]). However, we do not go into this problem here.

Appendix B: Degree of Freedom of a Polyhedron Line Drawing

The degree of freedom of a polyhedron image should be *at least four* if the polyhedron is not *coplanar* (i.e., not "flattened out"). This fact was first pointed out by Sugihara [35]. Suppose there exists a solution polyhedron that satisfies all the constraints for given data (x_i, y_i), $i = 1, \ldots, n$. Then, we can freely translate and deform the solution polyhedron into another while keeping the projection equations and the incidence structure preserved (figure B1). Such a "deformation" involves four parameters. Hence, there exist at least four degrees of freedom.

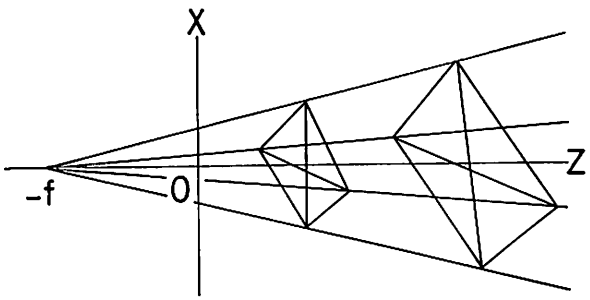


Fig. B1. Degree of freedom: The deformation of a *non*-flat polyhedron which preserves the projection relationship and the incidence structure involves at least four independent paramters. Hence, the degree of freedom is at least four.

Here, we give a formal proof which is slightly different from that of Sugihara [38]. First, we can assume that $n \geq 4$.[22] Let z_i, $i = 1, \ldots, n$, P_α, Q_α, R_α, $\alpha = 1, \ldots, m$, be a solution satisfying the constraints (12). It is easy to confirm that the following are also a solution:

$$z_i' = Ax_i + By_i + Cz_i + D \quad i = 1, \ldots, n \quad \text{(B1)}$$

[22] If $n \leq 3$, we can only define a point, a line segment, or a triangular plane.

$$P'_\alpha = CP_\alpha + A,$$
$$Q'_\alpha = CQ_\alpha + B,$$
$$R'_\alpha = CR_\alpha + D, \quad \alpha = 1, \ldots, m \qquad \text{(B2)}$$

Here, A, B, C, D are arbitrary constants such that $C > 0$. Our proof is complete if we can show that the four constants A, B, C, D are independent, that is, there is no redundancy among them. Assume the contrary, and suppose A, B, C, D, and A', B', C', D' define the same deformation. From equation (B1), this means that

$$Ax_i + By_i + Cz_i + D = Ax'_i + By'_i + Cz'_i + D', \quad i = 1, \ldots, n \qquad \text{(B3)}$$

or

$$(A - A')x_i + (B - B')y_i + (C - C')z_i + (D - D') = 0, \quad i = 1, \ldots, n \qquad \text{(B4)}$$

Thus, we obtain n (≥ 4) homogeneous linear equations in $A - A'$, $B - B'$, $C - C'$, $D - D'$. Among them, there should be at least four independent equations, and hence $A - A' = 0$, $B - B' = 0$, $C - C' = 0$, $D - D' = 0$, or $A = A'$, $B = B'$, $C = C'$, $D = D'$. In other words, any changes of these four parameters will produce different deformations. This means that the degree of freedom is at least four.

The fact that at least four of equations (B4) are independent can be seen easily: If not, any four of them are linearly dependent. Consequently, all minors of degree four are zero:

$$\begin{vmatrix} x_{i_1} & y_{i_1} & z_{i_1} & 1 \\ x_{i_2} & y_{i_2} & z_{i_2} & 1 \\ x_{i_3} & y_{i_3} & z_{i_3} & 1 \\ x_{i_4} & y_{i_4} & z_{i_4} & 1 \end{vmatrix} = 0, \quad \{i_1, \ldots, i_4\} \subset \{1, \ldots, n\} \qquad \text{(B5)}$$

But this is exactly the condition that all the vertexes are coplanar in the xyz-space (hence in the XYZ-space as well, cf. Remark 1). This contradicts our assumption that the polyhedron is not coplanar.

Appendix C: Singularity of the Incidence Structure

Consider the polyhedron image of figure Cla. The number of vertexes is $n = 6$, the number of faces is $m = 4$, and the incidence stucture consists of the following 15 incidence pairs:

$$R = \{(F_1, V_1), (F_1, V_2), (F_1, V_3), (F_2, V_1)$$
$$(F_2, V_2), (F_2, V_4), (F_2, V_5)$$
$$(F_3, V_2), (F_3, V_3), (F_3, V_5)$$
$$(F_3, V_6), (F_4, V_3), (F_4 V_1)$$
$$(F_4, V_6), (F_4, V_4)\}$$

We omit the numerical data, but it can be shown that the 15 constraints are linearly independent. It follows that the degree of freedom is $r = 6 + 3 \times 4 - 15 = 3$. According to appendix B, however, it must be at least 4 if the polyhedron is not to be coplanar. This means that the polyhedron must necessarily be "flat"; all the six vertexes and the four faces are coplanar. This can easily be understood: If any of the vertexes V_1, V_2, V_3 is out of the plane defined by the vertexes V_4, V_5, V_6, the faces F_2, F_3, F_4 cannot all be planar. This is because edges e_1, e_2, e_3 are not concurrent, while they should be if faces F_2, F_3, F_4 are all to be planar.

Consider figure Clb. This polyhedron has exactly the same incidence structure as that of figure Cla. This time, the image coordinates of vertexes V_1, V_2, V_3 are such that edges e_1, e_2, e_3 are concurrent, and hence a "nonflat" interpretation is possible. In analytical terms, the 15 constraints happen to be linearly dependent; only 14 of them are independent, and hence the degree of freedom is $6 + 3 \times 4 - 14 = 4$.

In a real problem, however, *this situation is highly unlikely to occur,* because only a small disturbance to the image coordinates will make the 15 constraints linearly independent, thus reducing the degree of freedom to 3. In other words, the situation of figure Clb is practically impossible.[23] Hence, we must regard this polyhedron as having an inherently "bad" structure. To put it differently, we must assume in general that the l constraints are always linearly independent; we cannot expect the very small possibility of degeneracy to occur.

In appendix B, we showed that the degree of freedom of a polyhedron which has n vertexes, m faces, and l incidence pairs is at least $n + 3m - l$. Hence, if $n + 3m - l > 4$, it is expected that a "nonflat" interpretation can exist. However, a *part* of it may have the "bad" structure, allowing only the "flat" interpretation for that part (except in the case of a very unlikely coincidence). We say, according to Sugihara [35-38], that a polyhedron (or, to be precise, its incidence structure) is

[23]We may say that the occurrence of this event is of *measure 0* by introducing an appropriate measure.

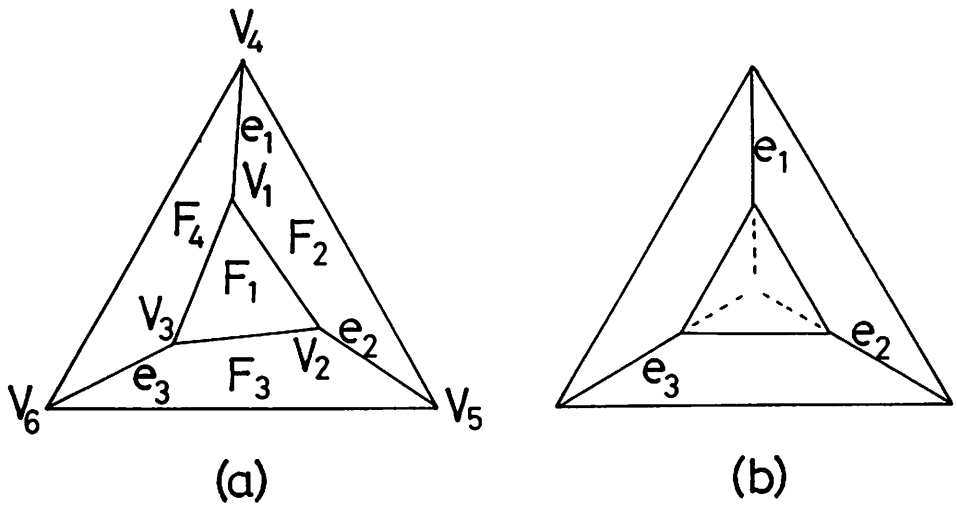


Fig. C1. Singularity: (a) The only possible interpretation is a "flat" polyhedron, all its faces lying on a plane. (b) A nonflat interpretation is possible if and only if the vertex coordinates take special values, which is highly unlikely in real situations.

singular if some part of it has this "bad" structure. Requiring the above inequality for all "subpolyhedra," we can expect that a polyhedron is nonsingular if and only if

$$|V(F')| + 3\,|F'| - |R(F')| \geq 4 \qquad \text{(C1)}$$

for any subset $F' \subset F = \{F_1, \ldots, F_m\}$ such that $|F'| \geq 2$. Here, $V(F')$ is the set of the vertexes incident to at least one of the faces in subset F', and $F(F')$ is the set of the incidence pairs involving the faces in subset F'. The vertical bars $|\cdot|$ denote the number of elements.

Sugihara [36,38] proved that the above condition is indeed the necessary and sufficient condition for an incidence structure to be nonsingular. This result is very important because the criterion of equation (C1) is easy to check: The process is *combinatorial*, and no numerical computation is involved.[24]

If the incidence structure is singular, we can make it nonsingular by partitioning some of its faces. For example, the incidence structure of figure C1a becomes nonsingular if face F_2 is partitioned into two faces F_2', F_2''. As a result, a "nonflat" interpretation becomes possible (figure C2a). In general, the incidence structure always becomes nonsingular if all the faces are partitioned into triangular surfaces. In particular, if the 2½D sketch is obtained as a polyhedral approximation of a smooth surface, we can always do this because the partitioning is completely arbitrary.

On the other hand, if we want to preserve the original partitioning, we can make its incidence structure nonsingular by removing some of the incidence pairs, as pointed out by Sugihara [35–38]. Then, a "nonflat" interpretation becomes possible, but some faces will meet in different places. Hence, the vertex positions must also be displaced in the 3D reconstruction. For example, the incidence structure of figure C1a becomes nonsingular if the incidence pair (F_2, V_1) is removed, but vertex V_1 must be displaced to a new position V_1' in the resulting nonflat interpretation (figure C2b).

As we noted in Remark 2, none of the above argument depends on whether the projection is perspective or orthographic.

[24] If we are to exhaust all subpolyhedra, $O(2^m)$ steps of computation are required, where m is the number of faces. However, there exists an efficient algorithm, due to Sugihara [38], which requires only $O(l^2)$ steps, where l is the number of incidence pairs.

Appendix D: 3D Recovery of a Planar Surface from Optical Flow

Consider a planar face with four corners moving in the scene. If image velocities are observed at those four corners, the surface gradient (p, q) and the rotation velocity $(\omega_1, \omega_2, \omega_3)$ are computed by the following procedure (see Kanatani [12] for the proof).

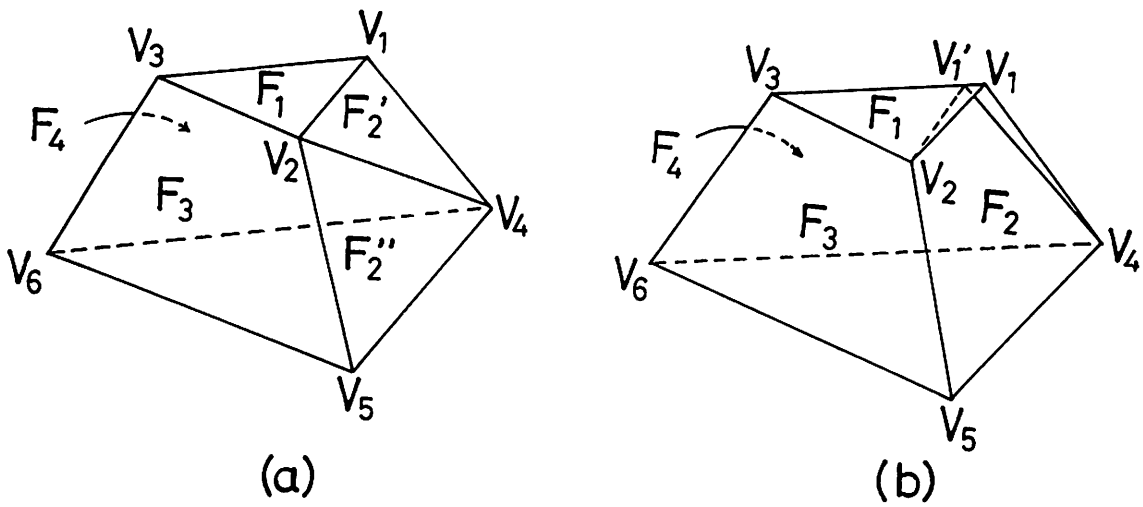


Fig. C2. A singular incidence structure can be made nonsingular by (a) partitioning some faces, or (b) removing some incidence pairs and displacing some vertexes. These two figures show possible 3D interpretations of the line drawing of figure C1a after such modifications.

Step 1. If velocities (u_i, v_i) are observed at four points (x_i, y_i), i = 1,...,4, compute the *flow parameters* u_0, v_0, A, B, C, D, E, F by solving the following simultaneous linear equations:

$$\begin{bmatrix} 1 & x_1 & y_1 & & & x_1^2 & x_1y_1 & \\ 1 & x_2 & y_2 & & & x_2^2 & x_2y_2 & \\ 1 & x_3 & y_3 & & & x_3^2 & x_3y_3 & \\ 1 & x_4 & y_4 & & & x_4^2 & x_4y_4 & \\ & 1 & & x_1 & y_1 & x_1y_1 & y_1^2 & \\ & 1 & & x_2 & y_2 & x_2y_2 & y_2^2 & \\ & 1 & & x_3 & y_3 & x_3y_3 & y_3^2 & \\ & 1 & & x_4 & y_4 & x_4y_4 & y_4^2 & \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \\ A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad \text{(D1)}$$

It can be proved that the determinant of the above matrix does not vanish unless three among the four points are collinear. Hence, if no three points are collinear, the flow parameters are uniquely determined from the velocities at the four points.

Step 2. Compute the following *invariants:*

$$\begin{aligned} T &= A + D, \\ R &= C - B, \\ S &= (A - D) + i(B + C), \\ L &= (E - \frac{u_0}{f}) + i(F - \frac{v_0}{f}) \end{aligned} \quad \text{(D2)}$$

Here, i is the imaginary unit, so U_0, S, and L are complex numbers in general.[25]

Step 3. Solve the following cubic equation:

$$X^3 + TX^2 + 1/4(T^2 - |S|^2 - |L|^2)X + \frac{1}{8}(\mathrm{Re}[L^2S] - T|L|^2) = 0 \quad \text{(D3)}$$

Here, $|\cdot|$ denotes the absolute value of a complex number, and Re [·], Im [·] the real part and the imaginary part, respectively. It can be proved that equation (D3) has three real roots (see Kanatani [12] for the proof). Let α be the middle of the three real roots.

Step 4. The surface gradient (p, q) and the rotation velocity $(\omega_1, \omega_2, \omega_3)$ are given as follows:

$$\begin{aligned} p &= \frac{1}{2\alpha} \mathrm{Re}\,[L \pm \sqrt{L^2 - 4\alpha S}] \\ q &= \frac{1}{2\alpha} \mathrm{Im}\,[L \pm \sqrt{L^2 - 4\alpha S}] \\ \omega_1 &= \frac{1}{2} \mathrm{Im}\,[L \mp \sqrt{L^2 - 4\alpha S}] - \frac{v_0}{f} \\ \omega_2 &= \frac{1}{2} \mathrm{Re}\,[L \mp \sqrt{L^2 - 4\alpha S}] + \frac{u_0}{f} \\ \omega_3 &= \frac{1}{2} R \pm \mathrm{Im}\,[L^* \sqrt{L^2 - 4\alpha S}] \end{aligned}$$

[25]These quantities define *irreducible representations* of the 2D rotation group $SO(2)$ corresponding to rotations of the image coordinate system, see Kanatani [13] for details.

Here, $*$ denotes complex conjugate, and one particular branch is chosen for all the complex square roots.[26] Equations (D4) show that there exist two sets of solutions. However, since the rotation velocities $\omega_1, \omega_2, \omega_3$ are common to all the faces, we can pick up the true solutions if two or more faces are observed. Hence, the surface gradient of each face is uniquely determined. See Kanatani [12] for numerical examples.

Appendix E: 3D Orientation of a Rectangular Corner in Perspective

Suppose we observe a rectangular corner on the image plane. Let (a, b) be the image coordinate of the vertex. Let ϕ_1, ϕ_2, ϕ_3 be the angles of the three edges (call them 1-, 2-, and 3-edges) measured from the image x-axis (figure E1). We want to compute the three unit vectors $\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3$ indicating the 3D orientations of these edges in the scene. They are computed by the following procedure (see Kanatani [15] for the proof).

Step 1. Compute

$$l_1 = \frac{a}{\sqrt{a^2 + b^2 + f^2}},$$

$$l_2 = \frac{b}{\sqrt{a^2 + b^2 f^2}}, \qquad \text{(E1)}$$

$$l_3 = \frac{f}{\sqrt{a^2 + b^2 f^2}}$$

$$E = \frac{a^2 l_3 + b^2}{a^2 + b^2},$$

$$F = \frac{ab(l_3 - 1)}{a^2 + b^2}, \qquad \text{(E2)}$$

$$G = \frac{b^2 l_3 + a^2}{a^2 + b^2}$$

Step 2. Compute the *canonical angles* $\bar{\phi}_1, \bar{\phi}_2, \bar{\phi}_3$ by

$$\bar{\phi}_i = -\tan^{-1} \frac{(fE + al_1)\tan\phi_i - (fF + bl_1)}{(fF + al_2)\tan\phi_i - (fG + bl_2)}, \quad i = 1, 2, 3 \qquad \text{(E3)}$$

[26]We assume that α is not zero. If it happens to be zero, another set of formulas must be used. See Kanatani [12] for details.

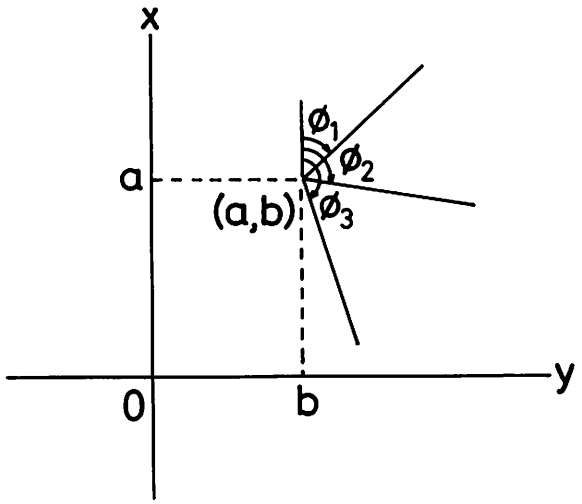


Fig. E1. The 3D orientation of a rectangular corner under perspective projection is computed *anayltically* in terms of the vertex position (a, b) and the three angles ϕ_1, ϕ_2, ϕ_3 of the edge images measured from the $x$axis.

where from among the two values resulting from $\tan^{-1}$ the one closer to the observed ϕ_i is chosen.

Step 3. Compute three angles $\bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_3$ by

$$\bar{\theta}_i = \tan^{-1} \sqrt{\frac{-\cos(\bar{\phi}_j - \bar{\phi}_k)}{-\cos(\bar{\phi}_i - \bar{\phi}_j)\cos(\bar{\phi}_i - \bar{\phi}_k)}}, \quad i \neq j \neq k \neq i \qquad \text{(E4)}$$

where $i, j, k \in \{1, 2, 3\}$.

Step 4. Compute three angles $\theta_1, \theta_2, \theta_3$ as follows. Put $\Delta\phi_{ij} \equiv |\phi_i - \phi_j| \pmod{2\pi}$. There are two possibilities.[27] There exists two solutions for each case.

1. If $\pi/2 < \Delta\phi_{ij} < \pi$ for all $(i, j) \in \{(2, 3), (3, 1), (1, 2)\}$, then $\theta_i = \bar{\theta}_i$ for $i = 1, 2, 3$ or $\theta_i = \pi - \bar{\theta}_i$ for $i = 1, 2, 3$.
2. If $0 < \Delta\phi_{ij} < \pi/2$, $0 < \Delta\phi_{jk} < \pi/2$, $\pi/2 < \Delta\phi_{jk} < 3\pi/2$ for some $i, j, k \in \{1, 2, 3\}$, $i \neq j \neq k \neq i$, then $\theta_i = \bar{\theta}_i$, $\theta_j = \pi - \bar{\theta}_j$, $\theta_k = \bar{\theta}_k$ or $\theta_i = \pi - \bar{\theta}_i$, $\theta_j = \bar{\theta}_j$, $\theta_k = \pi - \bar{\theta}_k$.

[27]We are checking the configuration of the corner image defined by the canonical angles $\bar{\phi}_1, \bar{\phi}_2, \bar{\phi}_3$: Case (1) corresponds to a "fork," and case (2) corresponds to an "arrow." If some $\Delta\phi_{ij}$ happen to be exactly 0, $\pi/2$, or π (i.e., for an L or T), the problem is degenerate, and infinitely many solutions exist. In all other cases, the image cannot be a perspective projection of a rectangular corner, which leads to the *rectangularity test* introduced in section 5 (figure 9).

Step 5. The 3D orientations of the three edges are specified by unit vectors $\boldsymbol{n}_1$, $\boldsymbol{n}_2$, $\boldsymbol{n}_3$ given by

$$\boldsymbol{n}_i = \begin{bmatrix} E & F & l_1 \\ F & G & l_2 \\ -l_1 & -l_2 & l_3 \end{bmatrix} \begin{bmatrix} \sin\theta_i \cos\bar{\phi}_i \\ \sin\theta_k \sin\bar{\phi}_i \\ \cos\theta_i \end{bmatrix}, \quad i = 1, 2, 3 \tag{E5}$$

where l_1, l_2, l_3, E, F, and G are given by equations (E1) and (E2).

Thus, there exist two solutions; one configuration is the mirror image of the other with respect to a mirror perpendicularly to the line of sight passing through the viewpoint and the corner vertex in the scene. See Kanatani [15] for numerical examples.

Appendix F: Computation of Vanishing Points

Projections of parallel lines in the scene must be concurrent on the image plane. Namely, they must intersect a common point—called the *vanishing point.* In order to prove this well-known fact, we only need to show that the vanishing point of a line is determined by its 3D orientation alone, irrespective of its location in the scene:

PROPOSITION F.1. *The vanishing point of a line whose 3D orientation is* $\boldsymbol{m} = (m_1, m_2, m_3)$ *is given by* $(fm_1/m_3, fm_2/m_3)$.

Proof. A line passing through point (X_0, Y_0, Z_0) and extending in the direction of vector $\boldsymbol{m} = (m_1, m_2, m_3)$ is given by

$$X = X_0 + tm_1 \quad Y = Y_0 + tm_2 \quad Z = Z_0 + tm_3 \tag{F1}$$

where t is a real number. The perspective projection of this line is given by

$$x = \frac{fX}{f+Z} = f\frac{X_0 + tm_1}{f + Z_0 + tm_3}$$
$$y = \frac{fY}{f+Z} = f\frac{Y_0 + tm_2}{f + Z_0 + tm_3} \tag{F2}$$

The vanishing point (a, b) of this line is obtained by taking the limit $t \to \pm\infty$: $a = fm_1/m_3$, $b = fm_2/m_3$. This result holds irrespective of the position (X_0, Y_0, Z_0).

COROLLARY. *The 3D orientation of a line whose vanishing point is* (a, b) *is given by the unit vector*

$$\boldsymbol{m} = \pm\left(\frac{a}{\sqrt{a^2+b^2+f^2}}, \frac{b}{\sqrt{a^2+b^2+f^2}}, \frac{f}{\sqrt{a^2+b^2+f^2}}\right) \tag{F3}$$

Thus, once we detect parallel edges, their vanishing point determines their 3D orientation. There is, however, one issue to be solved. Since the edges observed on the image plane are detected by image processing techniques, they may not be accurate. As a result, parallel lines may not necessarily meet at a single point when projected onto the image plane (figure 15). How can we estimate their common vanishing point? Evidently, some kind of average must be taken. At the same time, the computation must be done in the *finite* domain, because the vanishing point can be infinitely far apart from the image origin. This is done if we prepresent points and lines on the image plane by unit vectors or in terms of projective geometry the use of *(normalized) homogeneous coordinates.*

Consider a point $p : (a, b)$ on the image plane. The unit vector $\boldsymbol{m}$ starting from the viewpoint and pointing toward the point P on the image plane is given by

$$\boldsymbol{m} = \left(\frac{a}{\sqrt{a^2+b^2+f^2}}, \frac{b}{\sqrt{a^2+b^2+f^2}}, \frac{f}{\sqrt{a^2+b^2+f^2}}\right) \tag{F4}$$

(figure Fla). We call this $\boldsymbol{m}$ the unit vector *associated with* point P. Consider a line $l : Ax + By + C = 0$ on the image plane. The unit surface normal to the plane passing through the viewpoint and this line on the image plane is given by

$$\boldsymbol{n} = \left(\frac{A}{\sqrt{A^2+B^2+C^2/f^2}}, \frac{B}{\sqrt{A^2+B^2+C^2/f^2}}, \frac{C/f}{\sqrt{A^2+B^2+C^2/f^2}}\right) \tag{F5}$$

(figure 15b). We call this $\boldsymbol{n}$ the unit vector *associated with* the line l.

Here are three algorithms to identify a common intersection of "almost" concurrent lines.

ALGORITHM 1. Let $A_i x + B_i y + C_i = 0$, $i = 1, \ldots, n$, be almost concurrent lines on the image plane. Since these equations can be multiplied by an arbitrary nonzero constant, we can assume that the

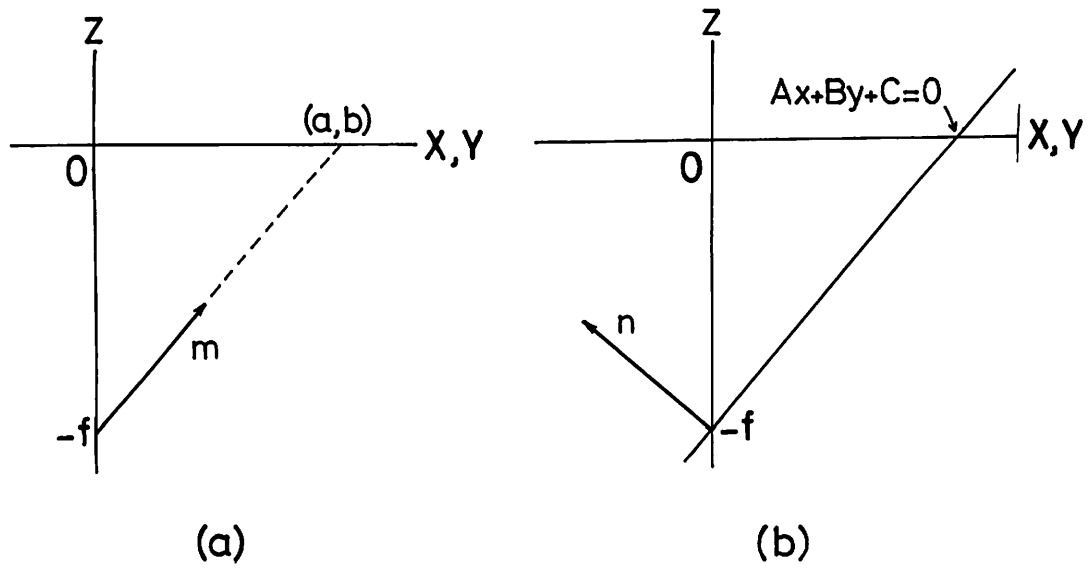


Fig. F1. (a) The unit vector m starting from the viewpoint and pointing toward point (a, b) on the image plane $Z = f$ is associated with that point. (b) The unit vector n starting from the viewpoint and normal to the plane passing through the viewpoint and line $Ax + By + C = 0$ on the image plane $Z = f$ is associated with that line.

coefficients A_i, B_i, C_i are so chosen that $A_i^2 + B_i^2 = 1$, $i = 1, \ldots, m$. Then, as shown in figure F2, the distance of line $A_ix + B_iy + C_i = 0$ from point (a, b) is given by $|A_ia + B_ib + C_i|$. Hence, we can determine the common vanishing point (a, b) by minimizing

$$\sum_{i=1}^{N} (A_ia + B_ib + C_i)^2 \qquad \text{(F6)}$$

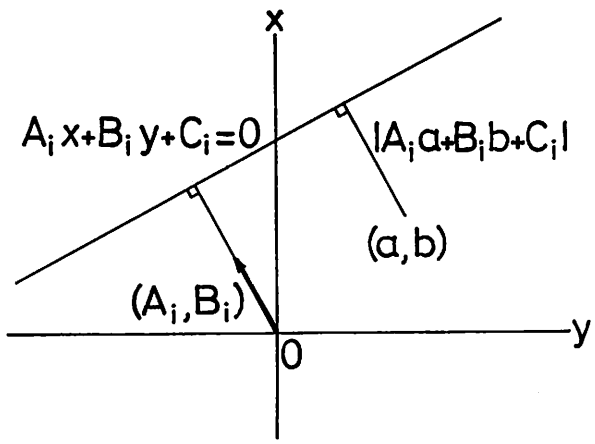


Fig. F2. The distance of line $A_ix + B_iy + C_i = 0$ from point (a,b) is given by $|A_ia + B_ib + C_i|$ if the equation is normalized so that $A_i^2 + B_i^2 = 1$.

The common intersection (a, b) is obtained by taking derivatives with respect to a and b and setting the respective results to be zero. The final result is as follows. If we put

$$m_1' = \sum_{i=1}^{N} A_iC_i \sum_{i=1}^{N} B_i^2 - \sum_{i=1}^{N} B_iC_i \sum_{i=1}^{N} A_iB_i$$

$$m_2' = \sum_{i=1}^{N} A_i^2 \sum_{i=1}^{N} B_iC_i - \sum_{i=1}^{N} A_iB_i \sum_{i=1}^{N} A_iC_i \qquad \text{(F7)}$$

$$m_3' = -f \left[\sum_{i=1}^{N} A_i^2 \sum_{i=1}^{N} B_i^2 - \left(\sum_{i=1}^{N} A_iB_i \right)^2 \right]$$

the vector $m' = (m_1', m_2', m_3')$ indicates the 3D orientation of these parallel lines. Since no division is involved, the computation can be confined within the *finite* doman. The unit vector m associated with the common intersection is obtained by normalizing m' into $m = m'/\|m'\|$.

Algorithm 1 may break down or result in poor accuracy if the original lines are obtained by *computation* and are not necessarily located in the actual image; if these lines are computed to be lying very far apart from the image origin, the coefficients can be very large. This difficulty is overcome if we use, instead of the equations of the lines, the unit vectors associated with them.

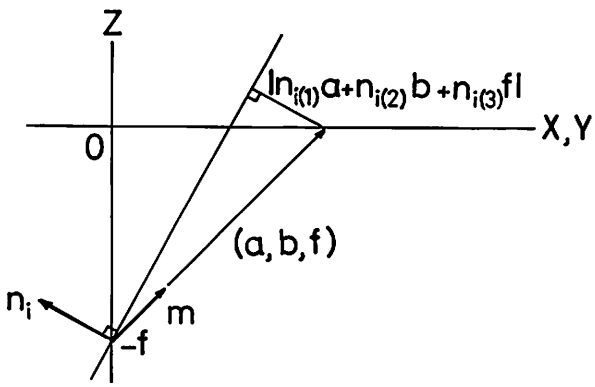


Fig. F3. The distance from point (a,b) on the image plane $Z = f$ to the plane passing through the viewpoint and line $A_ix + B_iy + C_i = 0$ on the image plane is given by $|n_{i(1)}a + n_{i(2)}b + n_{i(3)}f|^2$, where $n_i = (n_{i(1)}, n_{i(2)}, n_{i(3)})$ is the unit vector associated with that line.

ALGORITHM 2. Let $n_i = (n_{i(1)}, n_{i(2)}, n_{i(3)})$ be the unit vectors associated with lines $A_ix + B_iy + C_i = 0$, $i = 1, \ldots, N$. As shown in figure F3, the distance from point (a, b) on the image plane $Z = f$ to the plane passing through the viewpoint and line $A_ix + B_iy + C_i = 0$ on the image plane is given by $|n_{i(1)}a + n_{i(2)}b + n_{i(3)}f|$. It follows that we can determine the common vanishing point (a, b) by minimizing

$$\sum_{i=1}^{N} (n_{i(1)}a + n_{i(2)}b + n_{i(3)}f)^2 \qquad \text{(F8)}$$

The final result is as follows. If we put

$$m_1' = \sum_{i=1}^{N} n_{i(1)}n_{i(3)} \sum_{i=1}^{N} n_{i(2)}^2 - \sum_{i=1}^{N} n_{i(2)}n_{i(3)} \sum_{i=1}^{N} n_{i(1)}n_{i(2)}$$

$$m_2' = \sum_{i=1}^{N} n_{i(1)}^2 \sum_{i=1}^{N} n_{i(2)}n_{i(3)} - \sum_{i=1}^{N} n_{i(1)}n_{i(2)} \sum_{i=1}^{N} n_{i(1)}n_{i(3)} \qquad \text{(F9)}$$

$$m_3' = -f \left[\sum_{i=1}^{N} n_{i(1)}^2 \sum_{i=1}^{N} n_{i(2)}^2 - \left(\sum_{i=1}^{N} n_{i(1)}n_{i(2)} \right)^2 \right]$$

the vector $m' = (m_1', m_2', m_3')$ indicates the 3D orientation of these parallel lines. Since no division is involved, the computation is done within the *finite* domain. The unit vector m is obtained by normalizing m' into $m = m'/||m'||$.

ALGORITHM 3. Since the unit vector m associated with the common intersection is supposed to be orthogonal to all the unit vectors n_i, $i = 1, \ldots, N$, associated with the concurrent lines, we can estimate the vector m by minimizing

$$\sum_{i=1}^{N} (n_i, m)^2 \qquad \text{(F10)}$$

on the condition that m is a unit vector. It is easy to see that expression (F10) can be rewritten as the quadratic form

$$\sum_{j,k=1}^{3} N_{jk}\, m_j m_k \qquad \text{(F11)}$$

in m_1, m_2, m_3, where $N = (N_{jk})$, $j, k = 1, 2, 3$, is a symmetric matrix given by

$$N_{jk} = \sum_{i=1}^{N} n_{i(j)}n_{i(k)}, \qquad j, k = 1, 2, 3 \text{ (F12)}$$

The minimum of the quadratic form (F11) for unit vector m is attained by choosing the *eigenvector* of the matrix N corresponding to the minimum eigenvalue.

Algorithm 3 has the advantage that it is *invariant to camera rotation transformation* (cf. Kanatani [14,16]). However, since computation of eigenvalues and eigenvectors is involved, this method is computationally more costly than algorithms 1 and 2, by which the solutions are given as explicit expressions.

Appendix G: Concurrency Test of Edges

We first consider the *concurrency test.* We must check if three or more edges are concurrent on the image plane. In order to find an appropriate threshold value to make allowance for deviation of the intersection of two lines, consider a right *spherical triangle* ΔOPQ drawn on the *image sphere* (or the *Gaussian sphere*) [23] of radius f centered at the viewpoint (figure G1). Here, corner O is right-angle and located at the intersection of the image sphere with the Z-axis (figure G2).

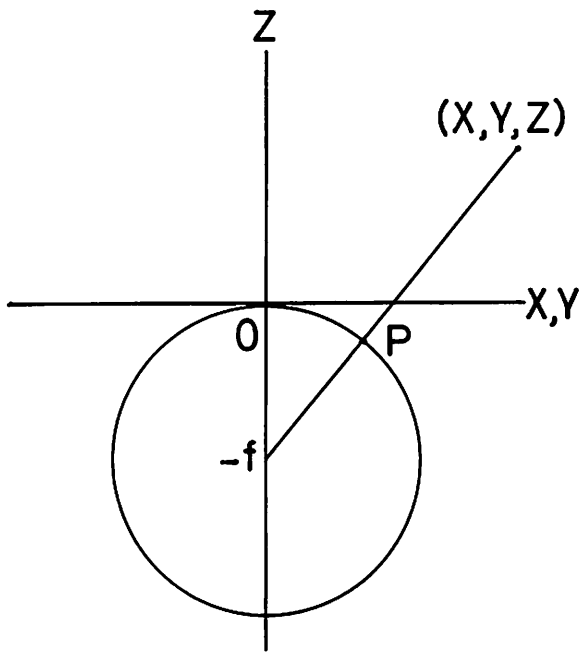


Fig. G1. Image sphere (Gaussian sphere): A point (X, Y, Z) in the scene is projected onto the intersection P of the sphere of radius f centered at the viewpoint $(0, 0, -f)$ with the ray connecting the point (X, Y, Z) and the viewpoint.

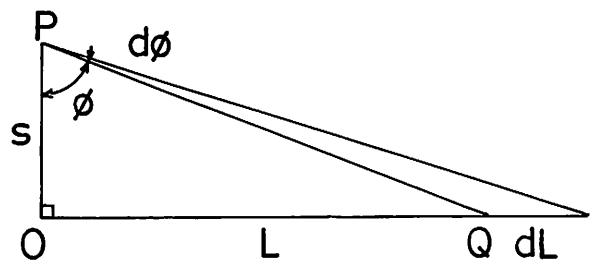


Fig. G2. A right spherical triangle drawn on the "image sphere."

Let $s = \rho(O, P)$, $L = \rho(O, Q)$, and $\phi = \theta(PO, PQ)$, where $\rho(\cdot, \cdot)$ and $\theta(\cdot, \cdot)$ are respectively the arc length and the angle measured on the image sphere.

LEMMA G.1.

$$\frac{dL}{d\phi} = f\left[\frac{1}{\sin(s/f)} - \left(\frac{1}{\sin(s/f)} - \sin\frac{s}{f}\right)\cos^2\frac{L}{f}\right] \tag{G1}$$

Proof. Invoking spherical trigonometry, we obtain

$$\tan\phi = \frac{\tan(L/f)}{\sin(s/f)} \tag{G2}$$

Keeping s fixed, and differentiating both sides, we obtain

$$\frac{d\phi}{\cos^2\phi} = \frac{dL/f}{\sin(s/f)\cos^2(L/f)} \tag{G3}$$

Equation (G2) is obtained if ϕ is eliminated from this by using equation (G3) and rearrange the result.

If $s/f \ll 1$, we obtain

$$\frac{dL}{d\phi} \approx \frac{f^2}{s}\left[1 - \cos^2\frac{L}{f}\right] \tag{G4}$$

We can interpret this as expressing the *sensitivity* of the location of the intersection of two lines, separated by distance s, to possible inaccuracy of edge orientation ϕ. Motivated by this interpretation, we regard L as *the arc length measured on the image sphere* of the intersection of two lines from the image origin O.

LEMMA G.2. *The unit vector* $\boldsymbol{m}$ *associated with the intersection of two lines* l, l' *is*

$$\boldsymbol{m} = \pm\frac{\boldsymbol{n} \times \boldsymbol{n}'}{||\boldsymbol{n} \times \boldsymbol{n}'||} \tag{G5}$$

where $\boldsymbol{n}$ *and* $\boldsymbol{n}'$ *are the unit vectors with the two lines* l, l'.

Proof. The unit vector $\boldsymbol{m}$ associated with the intersection must be orthogonal to both of the unit vectors $\boldsymbol{n}$, $\boldsymbol{n}'$ associated with these lines (figure G3). (See appendix F for the definition of vectors associated with points and lines.)

PROPOSITION G.3. The arc length L of the intersection of edges e, e' from the image origin O is given by

$$L = f\cos^{-1}\eta_{ee'} \tag{G6}$$

where[28]

$$\eta_{ee} \equiv \left|\frac{|\boldsymbol{nn'k}|}{||\boldsymbol{n} \times \boldsymbol{n}'||}\right| \tag{G7}$$

Here, $\boldsymbol{n}$ and $\boldsymbol{n}'$ are the unit vectors associated with the two lines, and $\boldsymbol{k} = (0, 0, 1)$.

[28] $|\boldsymbol{abc}|$ $[= (\boldsymbol{a} \times \boldsymbol{b}, \boldsymbol{c}) = (\boldsymbol{b} \times \boldsymbol{c}, \boldsymbol{a}) = (\boldsymbol{c} \times \boldsymbol{a}, \boldsymbol{b})]$ is the scalar triple product of three vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, which is equal to the determinant of the matrix whose columns are $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ in this order.

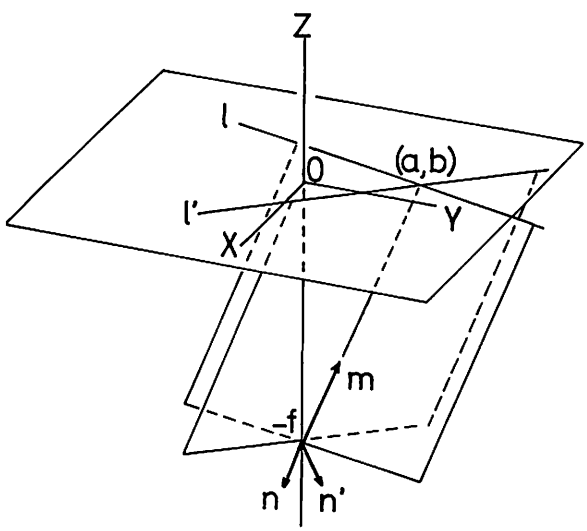


Fig. G3. The unit vector $\boldsymbol{m}$ associated with the intersection (a, b) of two lines l, l' must be orthogonal to both the unit vectors $\boldsymbol{n}$, $\boldsymbol{n}'$ associated with these lines.

Proof. The unit vector $\boldsymbol{m}$ associated with the intersection of the two lines l, l' is given by equation (G6). Since the angle of $\boldsymbol{m}$ from the Z-axis is L/f or $\pi - L/f$, we see that

$$\cos\frac{L}{f} = |(\boldsymbol{m}, \boldsymbol{k})| = \left|\frac{(\boldsymbol{n} \times \boldsymbol{n}', \boldsymbol{k})}{||\boldsymbol{n} \times \boldsymbol{n}'||}\right| = \left|\frac{|\boldsymbol{n}\boldsymbol{n}'\boldsymbol{k}|}{||\boldsymbol{n} \times \boldsymbol{n}'||}\right| \quad \text{(G8)}$$

For measuring the distance of the intersection from the image origin O, the quantity $\eta_{ee'}$ of equation (G7) is more convenient than the arc length L of equation (G8), since it is a nondimensional quantity whose value is always in the range $0 \leq \eta_{ee'} \leq 1$. The value 0 corresponds to an infinitely distant intersection, while the value 1 corresponds to the image origin O. The computation directly involves the unit vectors $\boldsymbol{n}$, $\boldsymbol{n}'$ associated with the two lines alone; the coordinates of the intersection need not be computed.

Now, we adopt

$$\epsilon_{ee'} = \frac{f^2 \Delta\phi_{ee'}}{s_{ee'}}(1 - \eta^2_{ee'}) \quad \text{(G9)}$$

as the threshold value $\epsilon_{ee'}$ associated with two edges e, e', where $s_{ee'}$ is taken to be an appropriately defined *average separation* of edges e, e', and $\Delta\phi_{ee'}$ is the probable error for edge orientation.

First, consider $s_{ee'}$. Let us proceed as follows. Let (x_0, y_0) and (x_1, y_1) be the two endpoints of edge e, and let (x_0', y_0') and (x_1', y_1') be those of edge e'. Put

$$e = (x_1 - x_0, y_1 - y_0), \quad e' = (x_1' - x_0', y_1' - y_0') \quad \text{(G10)}$$

We define their *average orientation* $\boldsymbol{l}$ weighted by their lengths as follows:

$$\boldsymbol{l} = \begin{cases} (e + e')/||e + e'|| & (e, e') \geq 0 \\ (e - e')/||e - e'|| & (e, e') < 0 \end{cases} \quad \text{(G11)}$$

The average separation $s_{ee'}$ of edges e, e' is defined to be the projection of the distance between their midpoints onto the axis perpendicular to the average orientation $\boldsymbol{l} = (l_1, l_2)$ (figure G4):

$$s_{ee'} \equiv \left(\frac{x_0 + x_1}{2} - \frac{x_0' + x_1'}{2}\right) l_2 - \left(\frac{y_0 + y_1}{2} - \frac{y_0' + y_1'}{2}\right) l_2 \quad \text{(G12)}$$

Note that $s_{ee'}$ becomes exactly zero if the two edges e, e' share a common endpoint.

Finally, the probable orientation error $\Delta\phi_{ee'}$ must be determined. Let $|e|$ and $|e'|$ be the lengths of edges e, e' respectively. Since it is reasonable to assume that the orientation of an edge becomes less accurate as its lengths becomes shorter, we put

$$\Delta\phi_{ee'} = \frac{\text{const.}}{\min(|e|, |e'|)} \quad \text{(G13)}$$

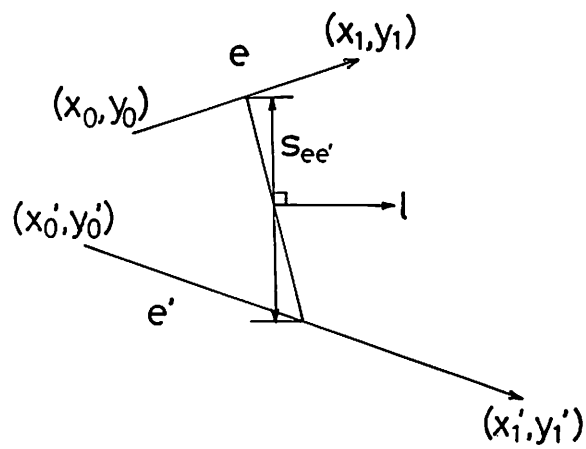


Fig. G4. The average separation $s_{ee'}$ of edges e, e' is defined to be the projection of the distance between their midpoints onto the axis perpendicular to the average orientation $\boldsymbol{l}$.

Now, we obtain the following procedure for the concurrency test. First, we choose two edges e, e' which do not meet at a corner of the object, and regard them as candidate parallel edges. Let $\boldsymbol{m}''$ be the unit vector associated with their intersection P'' (when they are extended) computed by equation (G5). Let us adopt the convention that the one with $m_3 \geq 0$ is chosen whenever we compute such a vector.

Consider the third edge e'', and let $\boldsymbol{m}$ and $\boldsymbol{m}'$ be the unit vectors respectively associated with the intersections P, P' of edge e'' with edges e, e' (when these edges are extended). We judge the three edges to be concurrent if

$$\begin{aligned} \rho(P, P') &< \epsilon_{ee''} + \epsilon_{e'e''} \\ \rho(P, P'') &< \epsilon_{ee'} + \epsilon_{e''e''} \\ \rho(P', P'') &< \epsilon_{e'e} + \epsilon_{e''e} \end{aligned} \tag{G14}$$

(figure G5). In terms of the vectors $\boldsymbol{m}$, $\boldsymbol{m}'$, $\boldsymbol{m}''$, these are rewritten as

$$\begin{aligned} |(\boldsymbol{m}, \boldsymbol{m}')| &> \cos \frac{\epsilon_{ee''} + \epsilon_{e'e''}}{f} \\ |(\boldsymbol{m}, \boldsymbol{m}'')| &> \cos \frac{\epsilon_{ee'} + \epsilon_{e''e'}}{f} \\ |(\boldsymbol{m}', \boldsymbol{m}'')| &> \cos \frac{\epsilon_{e'e} + \epsilon_{e''e}}{f} \end{aligned} \tag{G15}$$

Those edges that satisfy this condition are judged to be parallel to the candidate pair e, e', and their parallelism is established. However, if two edges that share a common endpoint are judged to be parallel to e and e', the one that passes this test for a lower threshold is chosen (with respect to an appropriately defined criterion). If no edge is judged to be parallel to the candidate pair e, e', edges e, e' are regarded as nonparallel.

In the above process, the order of choosing candidate edges can be arbitrary, but it is desirable to start from edges that are *most likely* to be parallel. One reasonable way is to start from the pair e, e' which has the smallest value of $\eta_{ee'}$ defined by equation (G7). This is equivalent to assuming that *two edges are more likely to be parallel in the scene if their intersection is located farther away from the image origin O.* Let us call this assumption the *vanishing point* heuristic. At the same time, it is reasonable to reject as nonparallel those edges intersecting at points within an appropriately set distance L_0 from the image origin O (i.e., $\eta_{ee'} > \cos(L_0/f)$). For example, if (x_i, y_i), $i = 1 \ldots, n$, are the image coordinates of the vertices, we can set $L_0 = \max_{1 \geq i \geq n} \sqrt{x_i^2 + y_i^2}$, assuming that the polyhedron is drawn near the center of the image.

Appendix H: Parallelogram Test

Let us construct an algorithm for the parallelogram test. This test is decomposed into repeated tests for intersection of two half-lines represented by their endpoints and points on them. Let l and l' be two half-lines starting respectively from points (a, b) and (a', b'), and let (x, y) and (x', y') be points lying respectively on half-lines l

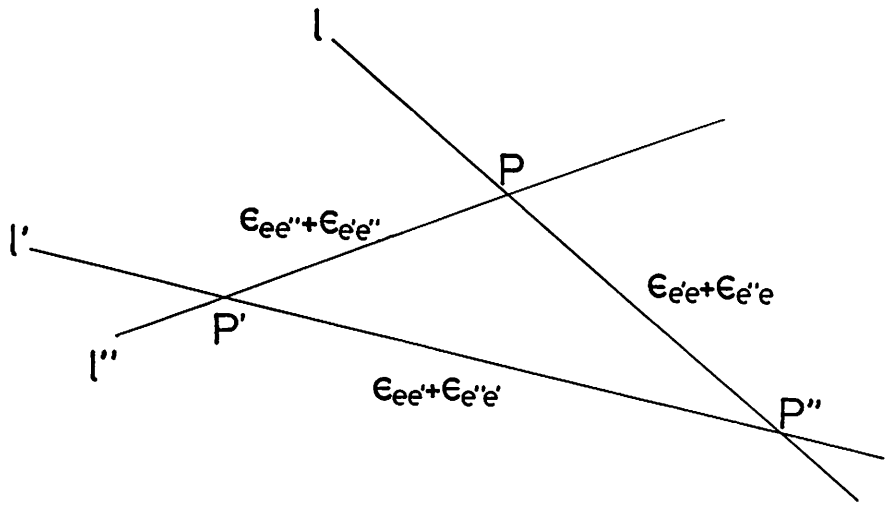


Fig. G5. Concurrency of three lines is judged by the pairwise arc lengths of the three intersections measured on the image sphere.

and l' (figure H1). We want to check whether or not half-lines l, l' intersect each other when (a,b), (x,y), (x',y') and (a',b') are given as input data. The intersection (ξ,ζ) exists and is given by

$$\begin{bmatrix} \xi \\ \zeta \end{bmatrix} = \begin{bmatrix} a + t(x - a) \\ b + t(y - b) \end{bmatrix} = \begin{bmatrix} a' + t'(x' - a') \\ b' + t'(y' - b') \end{bmatrix} \tag{H1}$$

if and only if such positive t and t' exist. From this relation, the constants t and t' are obtained in the form

$$\begin{aligned} t &= \frac{1}{\Delta}[(x' - a)(b' - b) - (y' - b)(a' - a)] \\ t' &= \frac{1}{\Delta}[(x - a)(b' - b) - (y - b)(a' - a)] \\ \Delta &= (x - a)(y' - b) - (y - b)(x' - a) \end{aligned} \tag{H2}$$

Hence, the test is performed by checking whether the three expressions

$$\begin{aligned} &(x' - a)(b' - b) - (y' - b)(a' - a), \\ &(x - a)(b' - b) - (y - b)(a' - a), \\ &(x - a)(y' - b') - (y - b)(x' - a') \end{aligned} \tag{H3}$$

have the same sign.

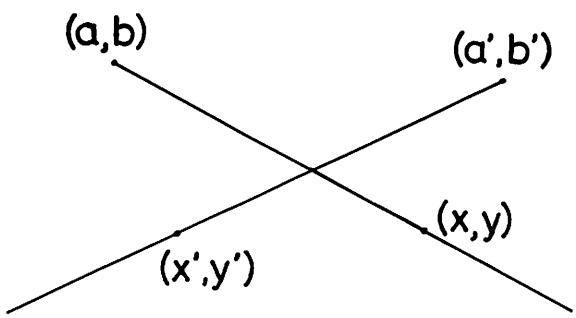


Fig. H1. Test for intersection of two half-lines starting from and passing through given points.

However, if (a,b) and (a',b') are the vanishing points of parallel lines, they may be located on the image plane very far apart from the image origin. As pointed out in appendix F, the computation can be done in the finite domain if we introduce (not necessarily unit) vectors $\boldsymbol{n} = (n_1,n_2,n_3)$ and $\boldsymbol{n}' = (n_1',n_2',n_3')$ associated with points (a,b) and (a',b'). Since $\boldsymbol{n}$ and $-\boldsymbol{n}$ represents the same point, let us choose the one with $n_3 \geq 0$. Similarly, we choose $\boldsymbol{n}'$ with $n_3' \geq 0$. If we substitute

$$a = f\frac{n_1}{n_3}, \quad b = f\frac{n_2}{n_3}, \quad a' = f\frac{n_1'}{n_3'}, \quad b' = \frac{n_2'}{n_3'} \tag{H4}$$

into expressions (H3) and multiply them by n_3 and n_3' appropriately, we see that the test is reduced to checking whether the following three expressions have the same sign:

$$\begin{aligned} &(n_3x - fn_1)(n_2'n_3 - n_2n_3') \\ &\quad -(n_3y - fn_2)(n_1'n_3 - n_1n_3'), \\ &(n_3'x' - fn_1')(n_2'n_3 - n_2n_3') \\ &\quad -(n_3'y' - fn_2')(n_1'n_3 - n_1n_3'), \\ &(n_3x - fn_1)(n_3'y' - fn_2') \\ &\quad -(n_3y - fn_2)(n_3'x' - fn_1') \end{aligned}$$

Thus, all computations are done in the finite domain.