[27] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[28] R. Shapira. A note on Sugihara's claim. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(1):122–123, January 1984.

[29] R. Siegal and J. R. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Publishing, New York, 1992.

[30] K. Sugihara. An algebraic approach to the shape-from-image-problem. *Artificial Intelligence*, 23:59–95, 1984.

[31] K. Sugihara. A necessary and sufficient condition for a picture to represent a polyhedral scene. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(5):578–586, September 1984.

[32] H. D. Tagare and Rui J. P. deFigueriedo. Simultaneous estimation of shape and reflectance maps from photometric stereo. In *Proc. Int. Conf. Comp. Vision*, pages 223–230, Boston, Massachusetts, December 1988.

[33] K.E. Torrance and E.M. Sparrow. Theory for off-specular reflections from roughened surfaces. *Journal of the Optical Society of America*, 57:1105–1114, September 1967.

[34] D. L. Waltz. Understanding line drawings of scenes with shadows. In P.H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, New York, 1975.

[35] R. J. Woodham. Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. *In Proc. SPIE*, 155:136–143, 1978.

[17] T. Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence*, 17:409–460, 1981.

[18] R. Kimmel and A. Bruckstein. Global shape from shading. *Comp. Vis. Graph. Im. Understanding*, 1995. to appear.

[19] D.G. Luenberger. *Linear and nonlinear programming*. Addison-Wesley, 1984. Second edition.

[20] A.K. Mackworth. Interpreting pictures of polyhedral scenes. *Artificial Intelligence*, 4(2):121–137, June 1973.

[21] G. W. Meyer, H. E. Rushmeier, M. F. Cohen, D. P. Greenberg, and K. E. Torrance. An experimental evaluation of computer graphics imagery. *ACM Transactions on Graphics*, 5(1):30–50, 1986.

[22] J.J. Moré, B.S. Garbow, and K.E. Hillstrom. User guide for MINPACK-1. ANL-80-74, Argonne National Laboratories, 1980.

[23] S. K. Nayar, K. Ikeuchi, and T. Kanade. Shape from interreflections. *Int. J. of Comp. Vision*, 6(3):173–195, August 1991.

[24] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: Physical and geometrical perspectives. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(7):611–634, July 1991.

[25] S.K. Nayar and M. Oren. Diffuse reflectance from rough surfaces. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 763–764, 1993.

[26] V.D. Nguyen. Labeling polyhedral scenes. In *Proc. DARPA Image Understanding Workshop*, pages 1160–1165, April 1988.

[6] S.W. Draper. The use of gradient and dual space in line-drawing interpretation. *Artificial Intelligence*, 17:461–508, 1981.

[7] G. Falk. Interpretation of imperfect line data as a three-dimensional scene. *Artificial Intelligence*, 3:101–144, 1972.

[8] D. Forsyth and A. Zisserman. Reflections on shading. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(7):671–679, July 1991.

[9] A. Guzman. Computer recognition of three-dimensional objects in a visual scene. Technical Report MAC-TR-59, MIT, 1968.

[10] G. Healey and T.O. Binford. Local shape from specularity. In *Proc. DARPA Image Understanding Workshop*, pages 874–877, 1987.

[11] B.K.P Horn. Obtaining shape from shading information. In P.H. Winston, editor, *The Psychology of Computer Vision*, pages 115–155. McGraw-Hill, New York, 1975.

[12] B.K.P. Horn. *Computer Vision.* MIT Press, Cambridge, Mass., 1986.

[13] D.A. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 6:295–323, 1971.

[14] D.A. Huffman. Realizable configurations of lines in pictures of polyhedra. *Machine Intelligence*, 8:493–509, 1977.

[15] K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.

[16] T. Kanade. A theory of the Origami world. *Artificial Intelligence*, 13:279–311, 1980.

represented by a set of linear equalities and inequalities, and the uncertainty in vertex position is explicitly taken into account.

We have shown that this scheme can be used to recover the 3D structure of objects which have complex reflectance models. It can actually be used for any polyhedral object whose reflectance can be reliably modelled. This includes images with shadows, metallic or dielectric objects, rough or smooth objects etc... Reflectance parameters (such as albedo or roughness) and light source parameters can be recovered as part of the process.

The main limitation of this approach is that it assumes that the line-drawing has been correctly extracted by the edge detector. Future research should be directed to combining line-drawing analysis and shading information to correctly extract the line-drawing, fully automating the shape recovery process.

References

[1] D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[2] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. Pergamon, New York, 1963.

[3] M.B. Clowes. On seeing things. *Artificial Intelligence*, 2(1):79–116, 1971.

[4] M. F. Cohen and D. P. Greenberg. The hemi-cube: A radiosity solution for complex environments. In *Proc. SIGGRAPH 85*, pages 31–40, San Francisco, July 1985.

[5] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Comput. Graphics*, 15(3):307–316, 1981.

these results to be quite good taking into account that the reflectance model does not model exactly the reflectance of the object.

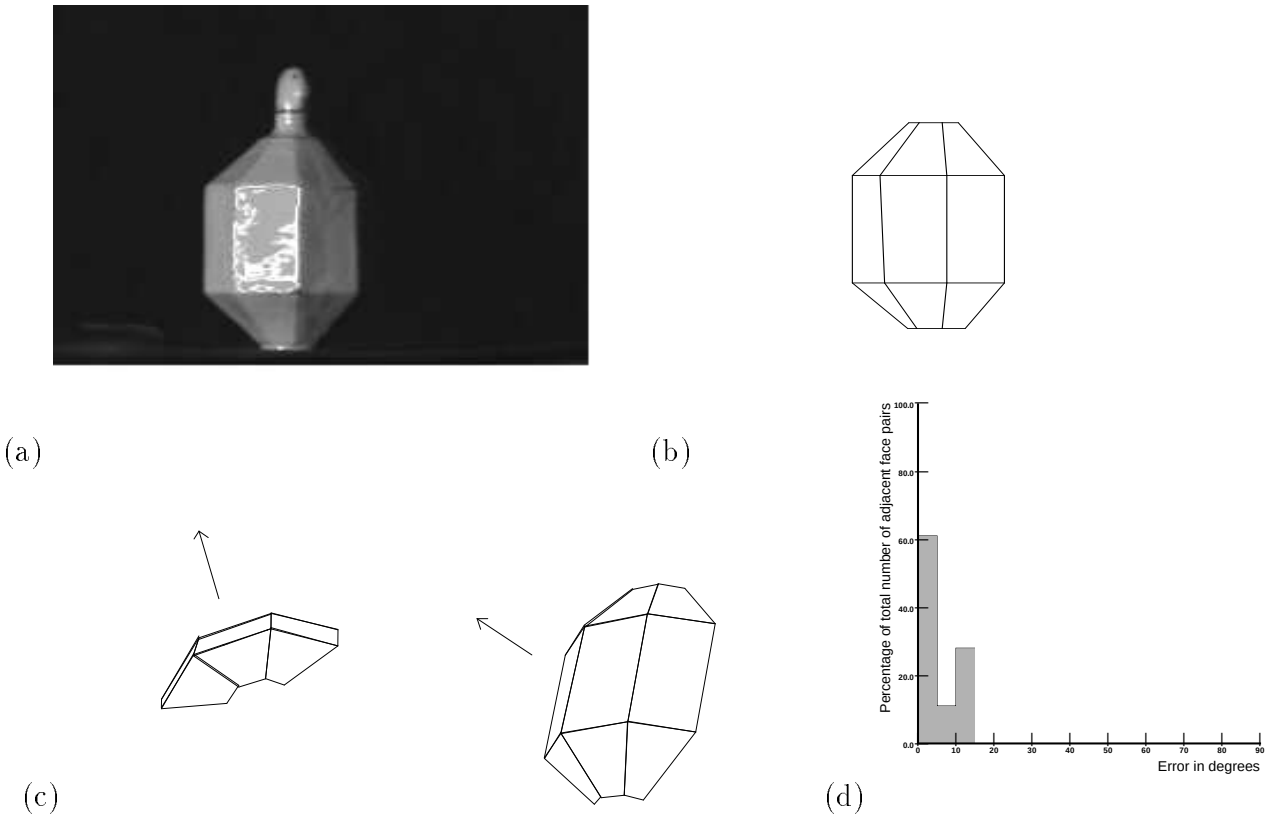

Figure 21: Results for the model for specular objects: (a) the input image; (b) the line-drawing; (c) two views of the recovered shape; (d) the errors in the angles between adjacent faces presented in a histogram.

7 Discussion and Future Work

We have presented a new approach to 3D shape recovery from a 2D image using the geometric information contained in the line-drawing and the shading information contained in the image. The geometric constraints imposed by the line-drawing of a polyhedron are

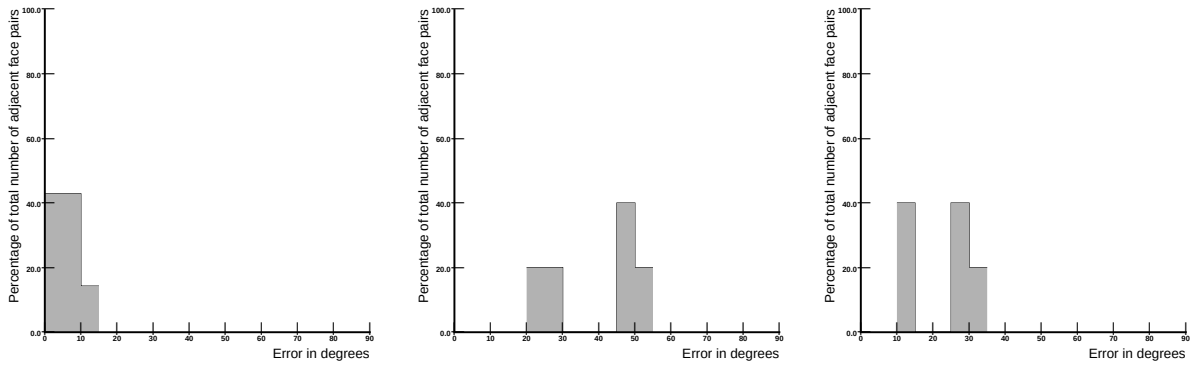

Figure 20: Error histograms comparing angles between adjacent faces measured in the object and measured in the recovered shape for the following objects: (a) the object in Figure 17; (b) the object in Figure 18; (c) the object in Figure 19.

6.4 Specular Objects

We have also tested our shape recovery algorithm on specular objects. Figure 21(a) shows the image of an object painted with glossy white paint. Notice that one of the faces displays a specularity. The algorithm was able to take advantage of the fact that one of the faces was specular by calculating directly an estimate for the direction of the light source when given an estimate for the parameters of the specular face, thus reducing the number of unknowns and speeding up the shape recovery process. Figure 16(b) shows the line-drawing extracted by hand from the edge detection results. Figure 16(c) shows two views of the reconstructed object. The arrows in Figure 16(c) show the direction from which the image was taken. In order to obtain a quantitative evaluation of our results we compared the angles between adjacent faces in the original objects and in the recovered object. The error measured was between 0.1° and 15° and the average was 6°. The distribution of the errors is shown in the histogram in Figure 16(d). We consider

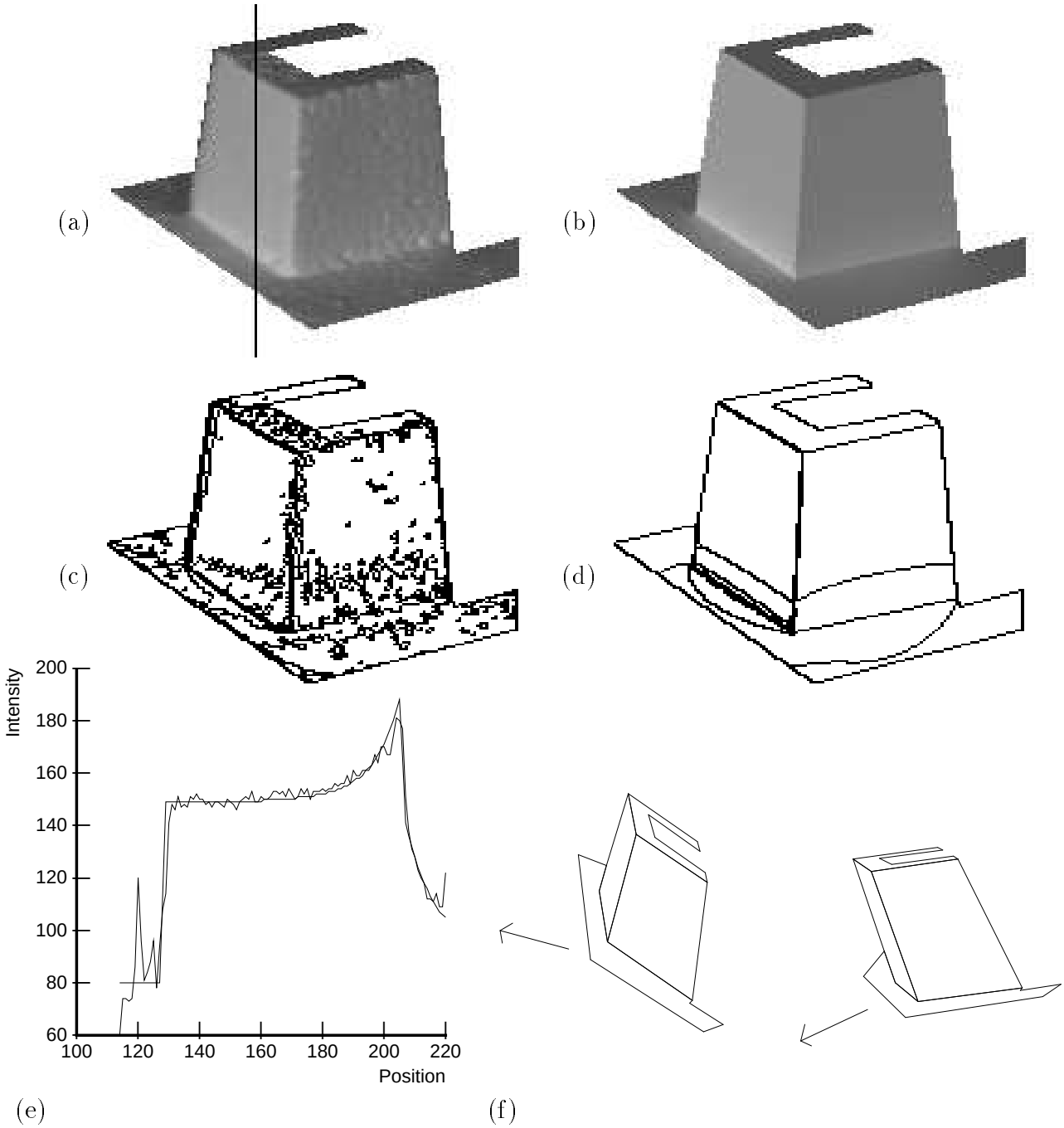

Figure 19: Second image of the styrofoam object: (a) the image of the object; (b) rendering of the recovered object; (c) iso-intensity lines in the image; (d) iso-intensity lines in the rendered image; (e) intensity values of the cross-section depicted by the line in (a) of the image and the rendered image; (f) views of the recovered object.

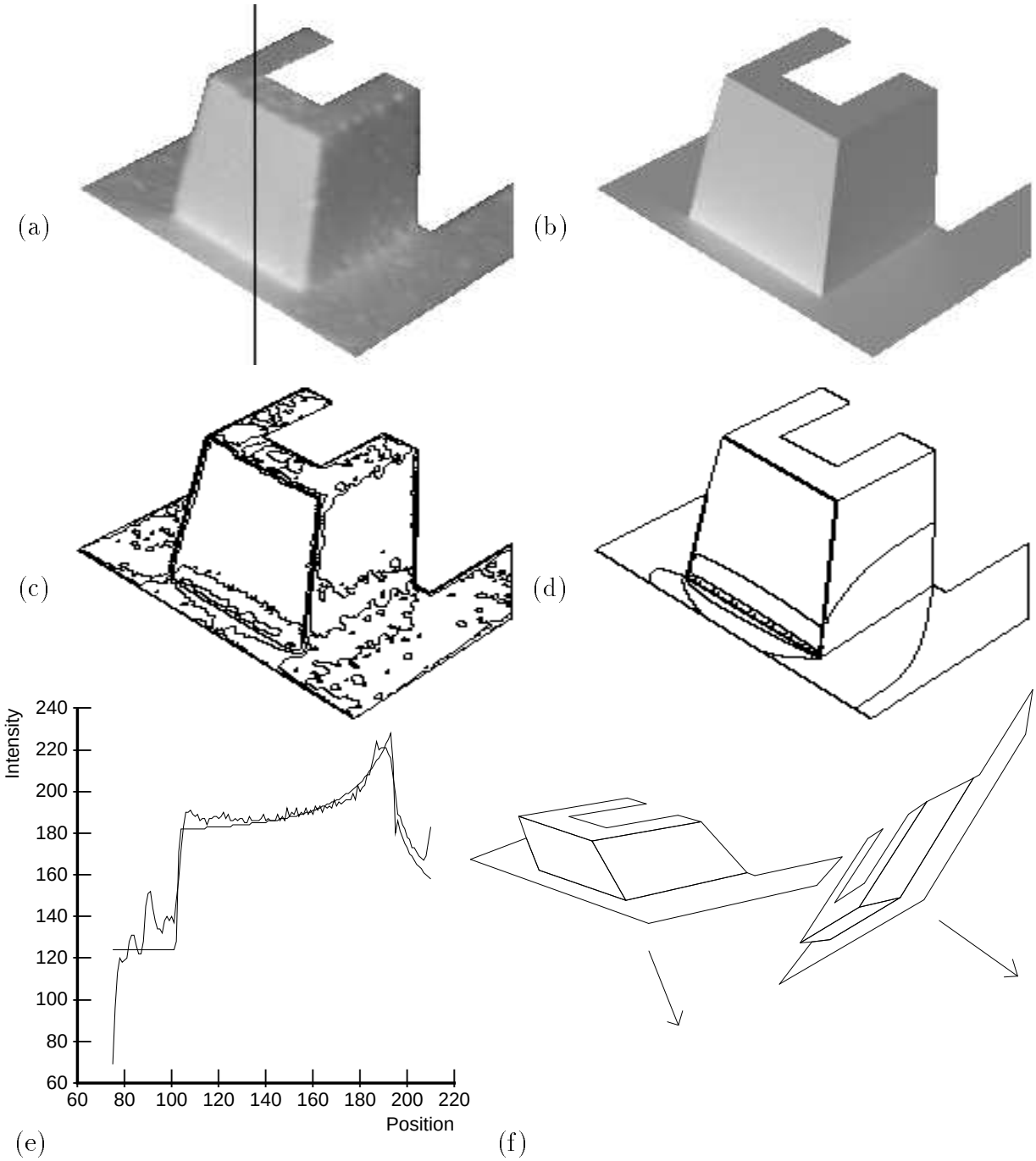

Figure 18: An image of a styrofoam object: (a) the image of the object; (b) rendering of the recovered object; (c) iso-intensity lines in the image; (d) iso-intensity lines in the rendered image; (e) intensity values of the cross-section depicted by the line in (a) of the image and the rendered image; (f) views of the recovered object.

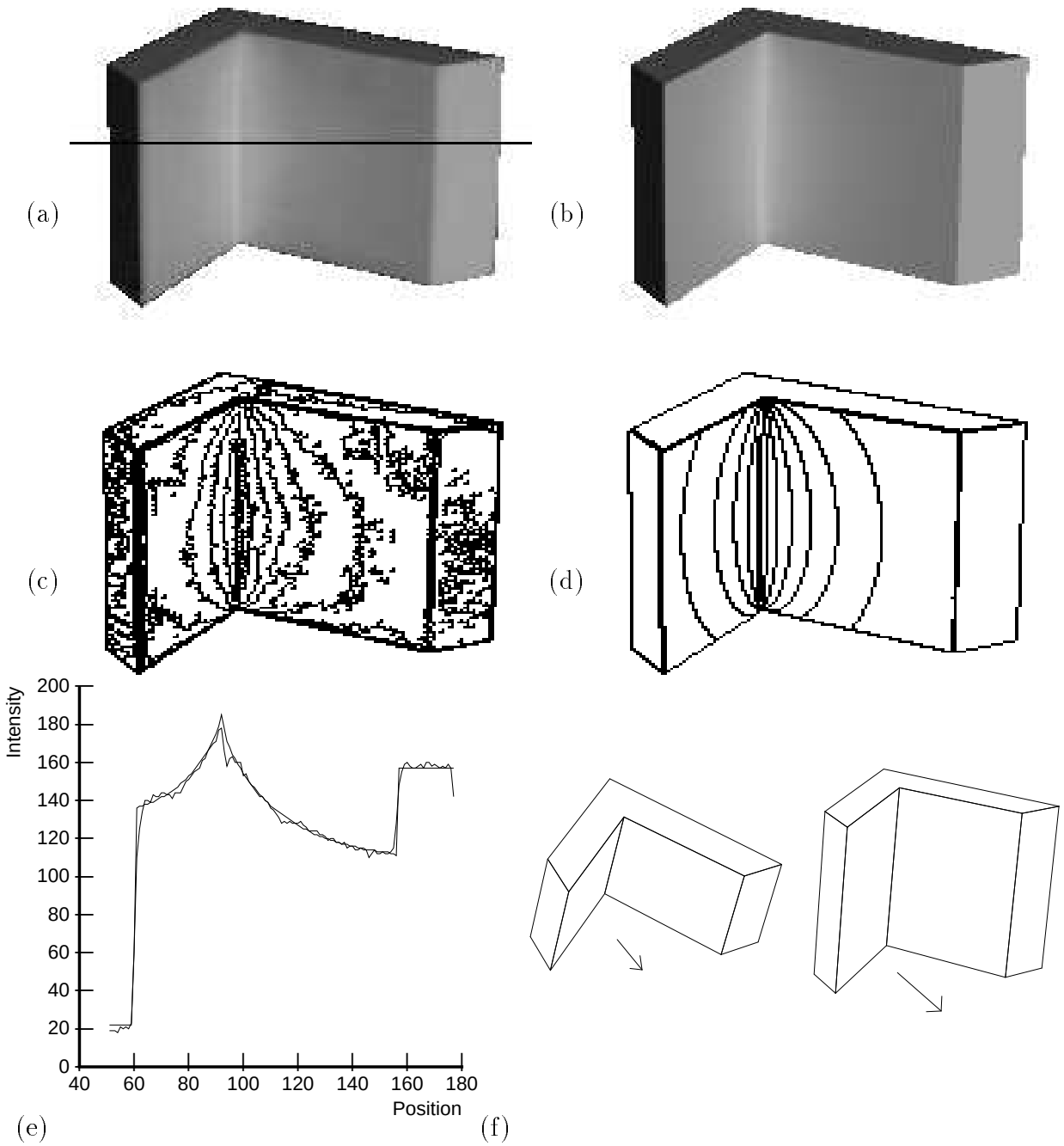

Figure 17: An L-shaped object: (a) the input image; (b) rendering of the recovered object; (c) iso-intensity lines in the image; (d) iso-intensity lines in the rendered image; (e) intensity values of the cross-section depicted by the line in (a) of the image and the rendered image; (f) views of the recovered object.

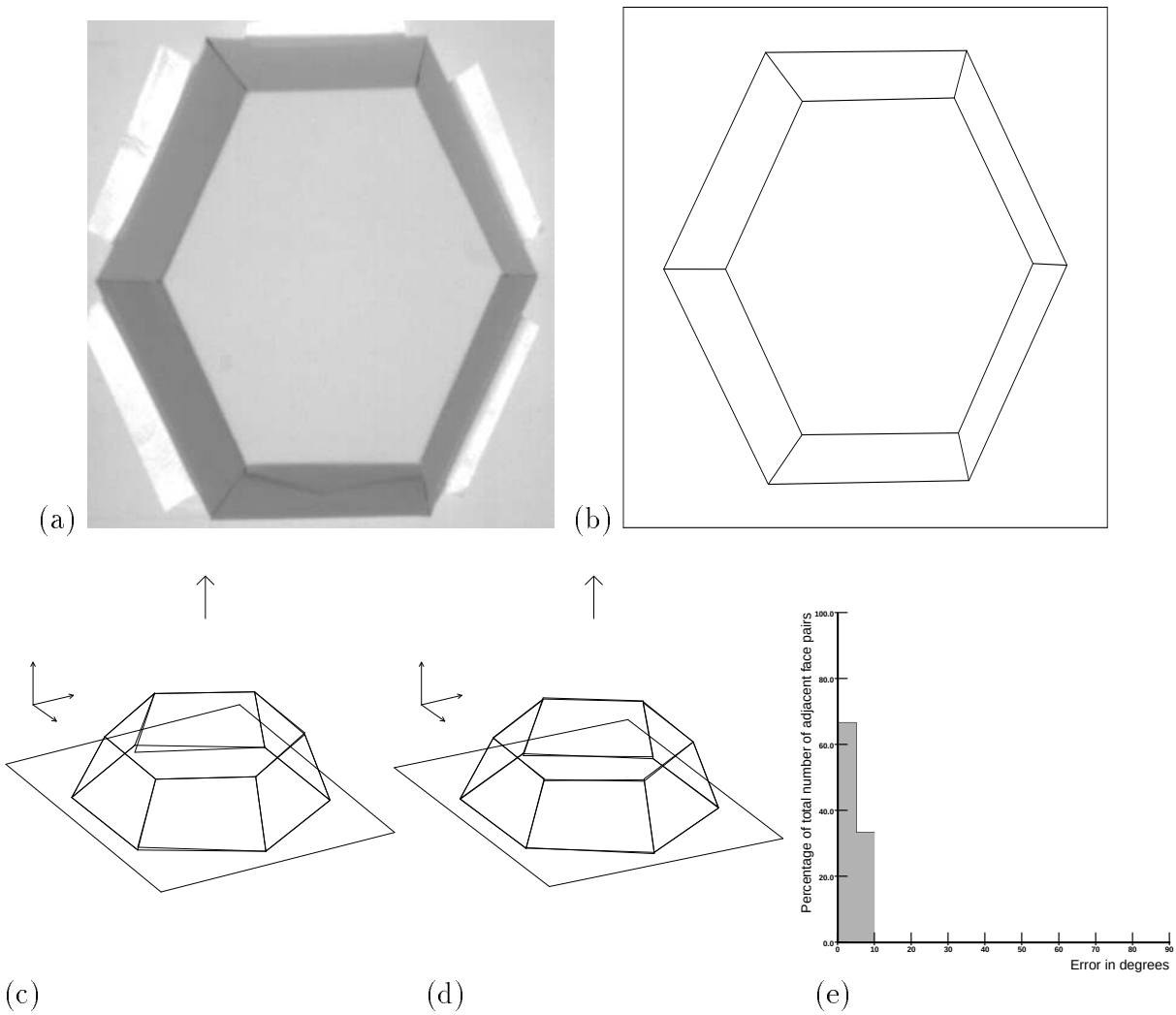

Figure 16: Results of the Lambertian model algorithm: (a) the input image; (b) the line-drawing; (c) first stage; (d) final stage of shape recovery; (e) the errors in the angles between adjacent faces presented in a histogram.

between adjacent faces in the original objects (which we measured by hand) and in the recovered objects. For the object in Figure 17 the maximum error is 11°. For the object in Figure 19 the error is between 12° and 31°. For the object in Figure 18 the recovered object is not as close to the original object, and the maximum error is 51° even though the intensity reconstruction is good (Figure 18(e)). The distribution of the errors for these three objects is shown in the histograms in Figure 20.

a quantitative evaluation of our results we compared the angles between adjacent faces in the original object (which we measured by hand) and in the recovered object. Figure 16(e) shows the distribution of the errors by angles.

6.3 Lambertian Objects with Interreflections

We have also run the algorithm on a number of objects which have concave edges causing interreflections between adjacent faces. To recover the shape of an object we take a subset of the pixels by uniformly sampling the image. In deciding on the size of the subset there is a tradeoff between the cost of running the algorithm which is proportional to the size of the subset, and the accuracy of the shape recovery which improves as the size of the selected subset grows. When computing $J_k(\mathbf{w}, \mathbf{l})$ for a certain pixel P, we assume that the intensity $L(P')$ of every other pixel P' in the image is as was measured in the image.

Figure 17(a) shows an image of an L-shaped wooden object painted with flat white paint. Figure 17(b) shows the intensity reconstruction of the image using the shape and scene parameters recovered by the optimization algorithm. Figure 17(c) shows the iso-intensity lines in the original image, and Figure 17(d) shows those lines in the rendered image. In Figure 17(e) the intensity of a cross-section of the two images is plotted, and Figure 17(f) shows two views of the recovered object. The arrows in Figure 17(f) show the direction from which the image was taken.

We have also run the algorithm on two images of a styrofoam object (once again painted with flat white paint) which is used for packing computers. The results for the first image (Figure 18) are not quite as good as the results for the second image (Figure 19), even though for both images the results of the intensity reconstruction are good despite much texture due to the grainy appearance of styrofoam.

In order to obtain a quantitative evaluation of our results we compared the angles

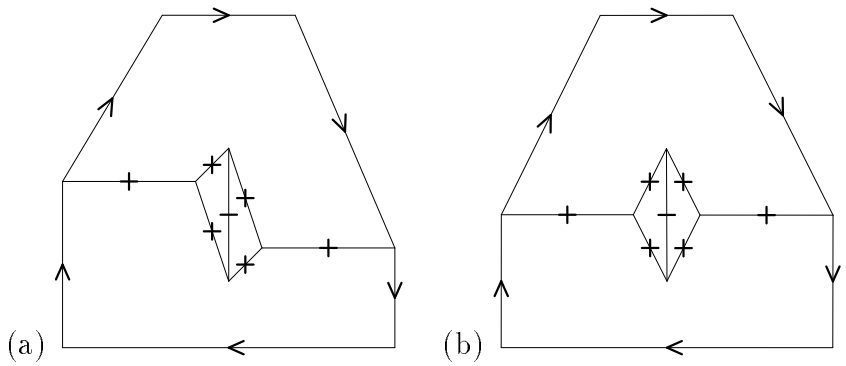

Figure 15: Results for two combinatorially equivalent line-drawings: (a) an incorrect line-drawing; (b) a correct line-drawing.

program. In this case, the corresponding linear programming problem has 44 variables and 80 constraints, with a total running time of 4.0s. Figure 15(b) shows a correct version of the same line-drawing, with a random perturbation of range 0.01 in vertex position. Note that this perturbation is not visible in this picture where the longest edge has length 10. The line-drawing is correctly classified by our program with $\epsilon = 0.01$. The number of variables and constraints is the same as before, and the running time is 3.9s.

6.2 Lambertian Objects

We have tested our shape recovery algorithm on several real images. Figure 16(a) shows the image of an object made of flat white paper. Edge detection was run on the image and a "clean" line-drawing shown in Figure 16(b) was extracted by hand from the results. Figure 16(c) shows the reconstruction obtained using the first stage of the algorithm. Notice that some adjacent faces do not coincide. This is due to removal of constraints due to superstrictness. Figure 16(d) shows the final result when all constraints have been taken into account and uncertainty in vertex position is reduced to below ϵ. The arrows in Figures 16(c,d) show the direction from which the image was taken. In order to obtain

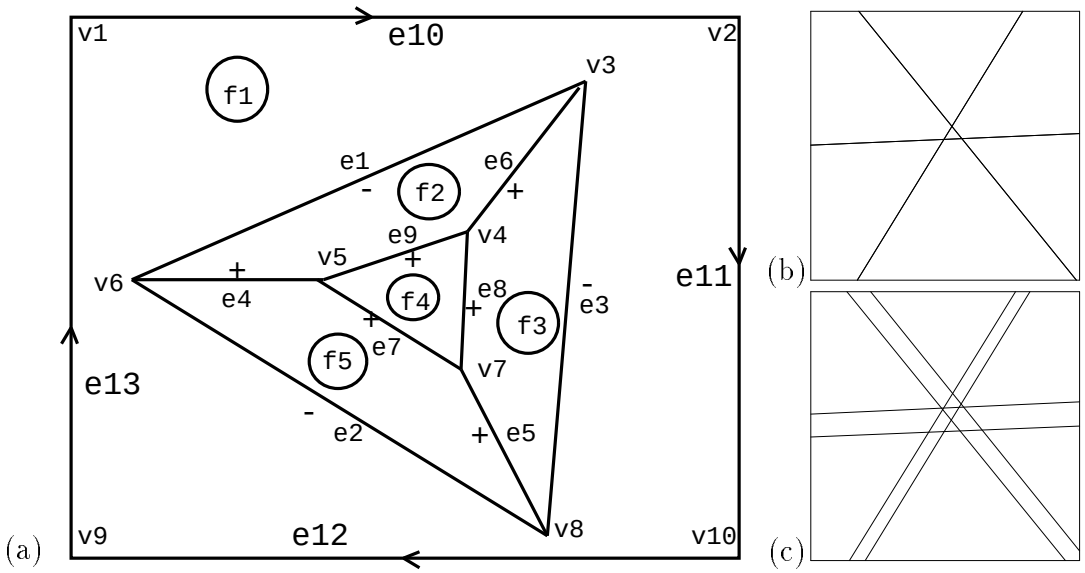

Figure 14: (a) A line-drawing from [30]. Close-ups of the region where the support lines of e_4, e_5, e_6 should intersect. (b) Incorrect line-drawing with $\epsilon = 0$. (c) Correct line-drawing with $\epsilon = 0.002$.

the nominal values given in [30], and an $\epsilon = 0$ uncertainty bound, this "correct looking" line-drawing is classified as incorrect by our program (Figure 14(b)). The corresponding linear programming problem has 51 variables and 84 constraints. The running time of the program is 3.6s. Next, we set the uncertainty bound to $\epsilon = 0.002$. This time, the line-drawing is classified as correct by our program, as confirmed by Figure 14(c), where the double cones corresponding to the three edges have been drawn and are shown to intersect. The number of variables is the same as before, but there are 120 constraints, and the running time of the program is 8.2s. The difference in the number of constraints comes from the fact that, for $\epsilon = 0$, each pair of inequalities corresponding to (16-18) can be replaced by a single equality.

Figure 15(a) is reproduced from [1]. It is an incorrect line-drawing, as verified by our

when such faces appear in the image is easier. When there is no such face the algorithm is reduced to the algorithm described for the Lambertian model. For this reflectance model we use the same two stages of optimization as for the Lambertian model.

6 Implementation and Results

We have implemented the approach proposed in Section 4 to test the correctness of a line-drawing in the presence of uncertainty using linear programming. In this case, linear programming is not used for optimization, but rather for deciding whether there exists a point satisfying these constraints. We present results classifying line-drawings as legal or illegal for different uncertainty bounds. We also present our implementation of 3D shape recovery from intensity data using the method described in Section 5. We have applied an active set optimization method to the solution space obtained through performing Gaussian elimination on the linear equality constraints. However, no special techniques were used exploiting the linearity of the inequalities. This did not degrade the performance of the algorithm because usually a point was found in the early stages of the algorithm that satisfied those constraints, and the main effort was put into computing and minimizing the objective function for points that satisfied all of the inequalities. The algorithm has been implemented in C using the simplex algorithm from [27] and the Levenberg-Marquardt procedure of the MINPACK library [22]. Experiments were run on a SUN SPARC Station.

6.1 Line-Drawing Analysis

Figure 14(a) shows a sample line-drawing, reproduced from [30]. Of particular interest are the three edges e_4, e_5, e_6 converging toward the center of this truncated pyramid. With

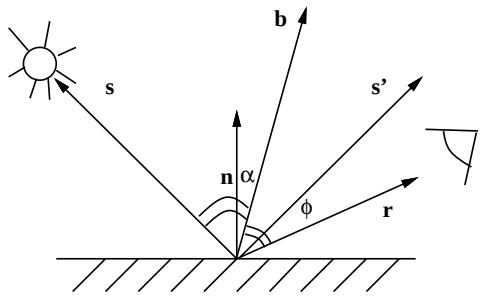

Figure 12: The general reflectance model; $\mathbf{n}$ is the normal to the surface, $\mathbf{s}$ is the light source direction, $\mathbf{s}'$ is the mirror direction, $\mathbf{r}$ is the viewing direction and $\mathbf{b}$ is the bisector between $\mathbf{s}$ and $\mathbf{r}$; α is the angle between $\mathbf{n}$ and $\mathbf{b}$, and ϕ is the angle between $\mathbf{r}$ and $\mathbf{s}'$.

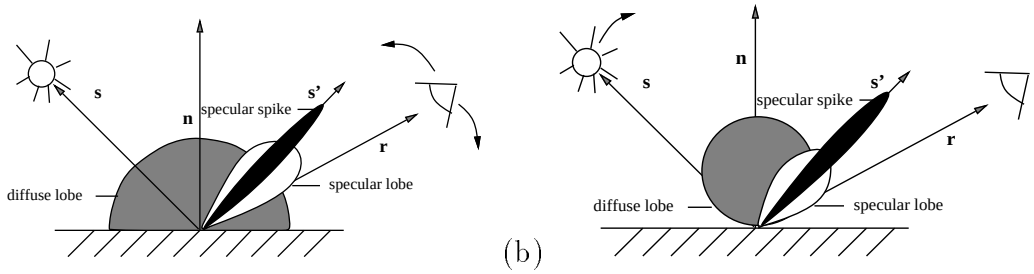

Figure 13: Polar plots of the three reflection components (adapted from [24]): (a) a function of the sensor direction for a fixed light source direction; (b) a function of the light source direction for a fixed sensor direction.

are easily distinguished from the others because of the high intensity of light reflected from them. For faces without a specular component the reflectance model reduces to the Lambertian model. For faces with a specular component we know that the camera is approximately in the mirror direction of that face. Thus given an estimate for the geometric parameters of a specular face, the light source direction $\mathbf{s}$ can be computed directly. Therefore when a specular face appears in the image, the number of independent unknowns is reduced. The number of unknowns is further reduced when there are several specular faces because that implies that those faces are parallel to each other. So although this reflectance model is more complicated then the Lambertian one, recovering the shape

graphics [5] and has also been used computer vision [10, 32]). The relative strengths of the specular lobe and the specular spike vary with the roughness of the object. As the object is rougher the magnitude of the specular lobe gets stronger and the magnitude of the specular spike decreases. In most instances one of the two specular components is significant while the other is negligible. In addition to these two components, the reflectance of most materials has a diffuse lobe which can be approximated by the Lambertian model.

In [24], Nayar et. al. analyzed the above reflectance models and proposed a simpler model to approximate them, and make them easier to use in computer vision algorithms. According to this model, the irradiance in the $\mathbf{r}$ direction E_{im} is written as:

$$E_{im} = K_{dl}(\mathbf{n} \cdot \mathbf{s}) + \frac{K_{sl}}{\mathbf{n} \cdot \mathbf{r}} \exp(-\frac{\alpha^2}{2\sigma^2}) + K_{ss}\delta(\phi),$$

where (as shown in Figure 12) ϕ denotes the angle between $\mathbf{r}$ and the mirror direction s' for source direction $\mathbf{s}$ reflected off the surface whose normal is $\mathbf{n}$, α is the angle between the bisector of ($\mathbf{s}$ and $\mathbf{r}$) and $\mathbf{n}$, σ denotes the slope roughness of the surface, and K_{dl}, K_{sl} and K_{ss} denote the strengths of the diffuse lobe, specular lobe, and specular spike components respectively. The specular spike and the specular lobe components peak when $\mathbf{r} = \mathbf{s}'$ as α and ϕ vanish. As the viewing direction changes the specular spike component decays rapidly to zero. The decay of the specular lobe component is more gradual and depends on the roughness of the surface. The rougher the surface, the larger σ gets, and the slower the decay. In Figure 13, polar plots of the three reflection components are shown as a function of the sensor direction for a fixed light direction (Figure 13(a)) and as a function of the light source direction for a fixed sensor direction (Figure 13(b)).

We have concentrated on relatively smooth objects. In this case most of the specular component of the reflected light is concentrated in directions close to the mirror direction. Faces which have a specular component in their reflection in the direction of the camera

where $L(P)$ is the intensity of light radiated from pixel P and $\mathbf{n}_P$ is the normal to the surface at P.

When the distance between P and the pixels it can "see" is large compared with the size of P, (27) can be approximated by:

$$L(P) = s\frac{\rho}{\pi}\mathbf{n}_P \cdot \mathbf{s} + \frac{\rho}{\pi}\sum_{P' \neq P} \frac{F(\mathbf{x}_P, \mathbf{x}_{P'})D\ L(P')}{\mathbf{v} \cdot \mathbf{n}_{P'}} \tag{28}$$

where $\mathbf{x}_P$ is the center of pixel P. For the contribution of each pixel to $L(P)$ we decide which of (27) or (28) to use by the distance between the pixels. Therefore (28) is mainly used for pairs of pixels which lie on both sides of a concave edge, close to it.

For this reflectance model we have three stages of optimization, using the result of each stage as the starting position of the next stage. In the first stage of the optimization we disregard interreflections and the uncertainty in vertex position. For each face f_k we choose the intensity I_k as the minimum intensity value of a pixel of that face. As interreflections can only add to the intensity of a pixel the minimum value should be as close as possible to the light reflected only due to the light source. In the second stage we take into account interreflections. We finally perform a third optimization stage which takes into account vertex uncertainty.

5.4 Reflectance Model for Specular Objects

The Lambertian model described above is limited to objects with matte surfaces. To deal with other types of materials, more complete reflectance models have been derived. Beckmann and Spizzichino [2] developed a reflectance model for rough specular objects. In this model the specular reflectance of an object was divided into two components: the specular lobe, and the specular spike. Torrance and Sparrow [33] developed a simpler model which captures only the specular lobe component (this model is widely used in

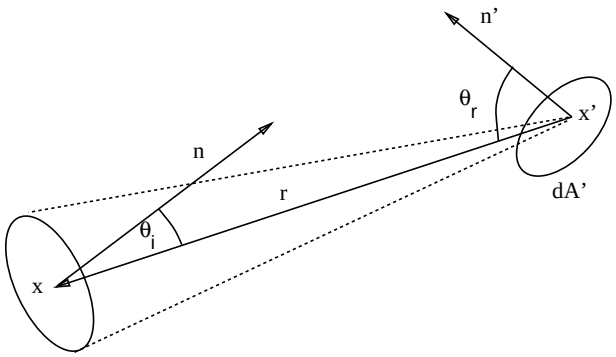

Figure 11: The contribution of light reflected from a surface patch dA' around $\mathbf{x}'$ to the irradiance of point $\mathbf{x}$, where $\mathbf{r} = \mathbf{x} - \mathbf{x}'$, and $\mathbf{n}$ and $\mathbf{n}'$ are the surface normals at $\mathbf{x}$ and $\mathbf{x}'$ respectively.

Here, View$(\mathbf{x}, \mathbf{x}')$ equals one when $\mathbf{x}$ and $\mathbf{x}'$ can see each other, and therefore can illuminate each other, and zero otherwise. Thus the irradiance $E(\mathbf{x})$ of the surface element dA' due to the radiance of the point $\mathbf{x}'$ is:

$$E(\mathbf{x}) = F(\mathbf{x}, \mathbf{x}')L(\mathbf{x}')dA'. \tag{25}$$

From equations (23) and (25), we obtain:

$$L(\mathbf{x}) = L_s(\mathbf{x}) + \frac{\rho}{\pi} \int F(\mathbf{x}, \mathbf{x}')L(\mathbf{x}')dA'. \tag{26}$$

We use (26) to compute the intensity of the light reflected by the pixels in the image. Each pixel which is a rectangle in the image whose area is D is actually a parallelogram in space whose area is $D/(\mathbf{v} \cdot \mathbf{n})$ where $\mathbf{v}$ is the viewing direction. We assume that each pixel has uniform intensity. Applying (26) at a pixel P we obtain:

$$L(P) = s\frac{\rho}{\pi}\mathbf{n}_P \cdot \mathbf{s} + \frac{\rho[\mathbf{v} \cdot \mathbf{n}_P]}{D\pi} \int_{\mathbf{x} \in P} \sum_{P' \neq P} L(P') \int_{\mathbf{x}' \in P'} F(\mathbf{x}, \mathbf{x}')dA'dA \tag{27}$$

first stage as the starting position of the second stage. The simplified model we use in the first stage disregards the uncertainty in vertex position. This reduces the number of unknowns considerably. In the case of superstrictness, like Sugihara [30], we delete the corresponding constraints. In the second stage we take into account all the constraints and the uncertainty in vertex position.

5.3 Lambertian Model With Interreflections

Radiosity methods are used in computer graphics to compute the interreflections of visible light and to render physically-accurate images of objects [4, 29]. They have also been used in computer vision to recover the shape of objects from images with interreflections [8, 24]. In [24] photometric stereo is used in the initial stages of the algorithm requiring several images and knowing the light source parameters. On the other hand the albedo is unknown and does not have to be constant, and the shape of the recovered object does not have to be polyhedral.

Under the interreflection model the total irradiance $E(\mathbf{x})$ at point $\mathbf{x}$ is expressed as the sum of the irradiance $E_s(\mathbf{x})$ due to the light source and the irradiance due to all other points on the surface which can see $\mathbf{x}$. The radiance L at the point $\mathbf{x}$ is related to its irradiance E by:

$$L(\mathbf{x}) = \frac{\rho(\mathbf{x})}{\pi} E(\mathbf{x}) \tag{23}$$

where $\rho(\mathbf{x})/\pi$ is the bi-directional reflectance distribution function for a Lambertian surface. The geometric factor (the form factor) which relates the irradiance $E(\mathbf{x})$ due to the radiance of the point $\mathbf{x}'$ to the radiance of $\mathbf{x}'$, whose geometry is illustrated in Figure 11 is:

$$F(\mathbf{x}, \mathbf{x}') = \text{View}(\mathbf{x}, \mathbf{x}') \frac{[\mathbf{n} \cdot (\mathbf{x}' - \mathbf{x})][\mathbf{n}' \cdot (\mathbf{x} - \mathbf{x}')]}{|\mathbf{x}' - \mathbf{x}|^4}. \tag{24}$$

5.2 Lambertian Model

Under the Lambertian model we assume that there is a distant point light source whose direction $\mathbf{s}$ and intensity s are unknown and suppose that the observed object has a constant (but unknown) albedo ρ (see Figure 10). Our method differs from classical shape-from-shading techniques [15] in which these parameters are assumed to be known (on the other hand those methods do not assume that the object is polyhedral). Photometric stereo [35] does not require the albedo to be constant or that the object be polyhedral but requires the direction and intensity of the light sources and more than one image.

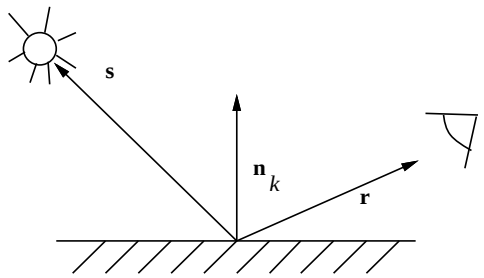

Figure 10: The Lambertian reflectance model; $\mathbf{n}_k$ is the normal to the surface, $\mathbf{s}$ is the light source direction, and $\mathbf{r}$ is the viewing direction.

The k^{th} cue I_k is the intensity measured on the k^{th} face. We compute J_k as:

$$J_k(\mathbf{w}, \mathbf{l}) = \frac{s\rho}{\pi}\mathbf{n}_k \cdot \mathbf{s} \tag{22}$$

where $\mathbf{n}_k = \frac{1}{\sqrt{(p_k^2+q_k^2+1)}}(-p_k, -q_k, 1)^T$ is the normal to face number k, and the scene parameters $\mathbf{l}$ denote the lighting direction $\mathbf{s}$ the intensity of the light s, and the albedo ρ. In this case we cannot decouple the albedo ρ from the intensity s of the light source, therefore we recover their product.

For this reflectance model we have two stages of optimization, using the result of the

reduces the original problem to the *unconstrained* minimization of:

$$\sum_k (I_k - J_k(\mathbf{w}, \mathbf{l}))^2 + \sum_j (min\{0, B_j \mathbf{w}\})^2, \tag{21}$$

where B_j denotes the j^{th} row of B. From the way the problem is stated, it is clear that any model J_k which can recover I_k accurately when given the scene parameters $\mathbf{w}$ and $\mathbf{l}$ can be used to recover the shape information and the scene parameters. Therefore models like the ones used in computer graphics to render physically accurate images can be substituted for J_k in the formalization of the problem.

Because optimization algorithms may fall into local minima a method must be devised to give the algorithm a starting position as close as possible to the optimal one. The method we have chosen is to divide the algorithm into two stages. In the first one we run the optimization algorithm using a simplified model (e.g., disregarding the effects of interreflections on the image). Such a model will have fewer unknown parameters than the full model or make it easier to compute $J_k(\mathbf{w}, \mathbf{l})$, but it may give less accurate results. We run the optimization using the simple model with various starting positions. In the second stage we use the best result of the first stage as the starting position for optimization using the elaborate model. For complex reflectance models we generalized this method for more than two stages, where the result of each stage is the starting position of the next stage.

In order to demonstrate the applicability of our method we have implemented shape recovery from shading information using three reflectance models: the pure Lambertian model, the Lambertian model with interreflections [8, 24], and a reflectance model for smooth specular objects [2, 24, 33].

5 3D Shape Recovery

5.1 Principle

Like Sugihara [30], we formulate 3D shape recovery as an optimization problem. Consider a set of visual cues, such as intensity or texture. Let $\mathbf{w}$ denote the unknown geometric parameters, $\mathbf{l}$ denote the other scene parameters (e.g., light source direction and intensity, surface albedo), I_k denote the observed value of the k^{th} cue (e.g., the intensity of face f_k), and $J_k(\mathbf{w},\mathbf{l})$ denote the value of this cue that would be observed if the actual scene parameters were $\mathbf{w}$ and $\mathbf{l}$. Recovering $\mathbf{w}$ and $\mathbf{l}$ amounts to minimizing:

$$\sum_k (I_k - J_k(\mathbf{w},\mathbf{l}))^2 \tag{19}$$

under the linear constraints derived in Section 4:

$$\begin{cases} A\mathbf{w} = 0, \\ B\mathbf{w} \geq 0. \end{cases} \tag{20}$$

This is what Sugihara calls a quadratic error minimization, a term we find a bit misleading since the error term is in general *not* quadratic in the unknown parameters. This is unfortunate since quadratic minimization under linear constraints can be solved exactly.

To solve the constrained minimization problem, we first use Gaussian elimination to delete all linear equalities and determine the values of a corresponding number of variables as a linear function of the remaining ones. Among the possible choices, we choose the subset of variables which yields the best numerical stability. This is done by finding which n-tuple of standard basis vectors produce a square matrix with the highest possible condition number when added to matrix A.

We solve our constrained optimization problem using an active set method [19], which

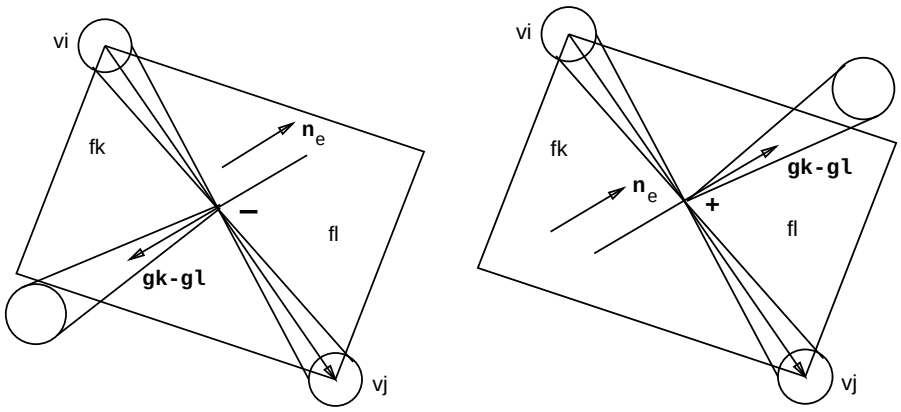

Figure 9: Constraints on the direction of $\mathbf{g}_k - \mathbf{g}_l$ imposed by concave (left) and convex (right) edges.

4.3 Remarks

We have shown that, even in the presence of uncertainty, a labelled line-drawing can be characterized by a set of linear constraints. Let us close this section with a few remarks. First, one of the variables r_k can be set to some arbitrary value since absolute depth cannot be recovered under orthographic projection. The variables z_i can be recovered from the other variables. Second, (11) and (15) contain strict inequalities, which is normally a difficulty for linear programming. As remarked by Sugihara [31], all the constraints are homogeneous, and strict inequalities with a zero constant term can be replaced by non-strict inequalities with an arbitrary positive constant term. Third, these constraints only provide a necessary condition for correctness. In particular, the signs of $p_k - p_l$ and $q_k - q_l$ cannot always be determined.

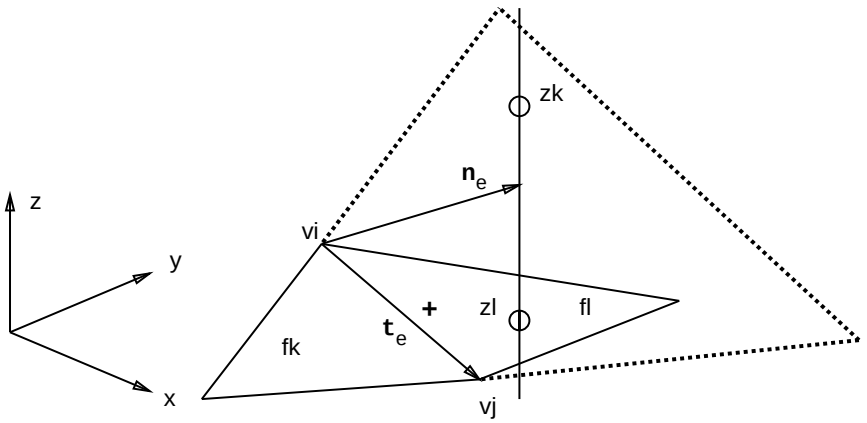

Figure 8: The geometric constraints imposed by a convex edge.

Combining (13) and (15) yields a *linear* constraint:

$$|(\mathbf{g}_k - \mathbf{g}_l) \cdot \tilde{\mathbf{t}}_e| \leq ((\mathbf{g}_k - \mathbf{g}_l) \cdot \tilde{\mathbf{n}}_e) \tan \alpha. \tag{16}$$

The analysis applies to concave edges by inverting the orientation of $\mathbf{g}_k - \mathbf{g}_l$ (Figure 9).

When the double cone centered in $\tilde{\mathbf{n}}_e$ does not contain the x axis, (16) determines the sign $s = \mp 1$ of $p_k - p_l$, yielding the *linear* constraint:

$$|a_{ki} - a_{li}| \leq \epsilon s (p_k - p_l). \tag{17}$$

Similarly, when the double cone does not contain the y axis, (16) determines the sign s of $q_k - q_l$, yielding:

$$|b_{ki} - b_{li}| \leq \epsilon s (q_k - q_l). \tag{18}$$

In general, a convex or concave edge provides at least one of these two constraints for each of its extremities.

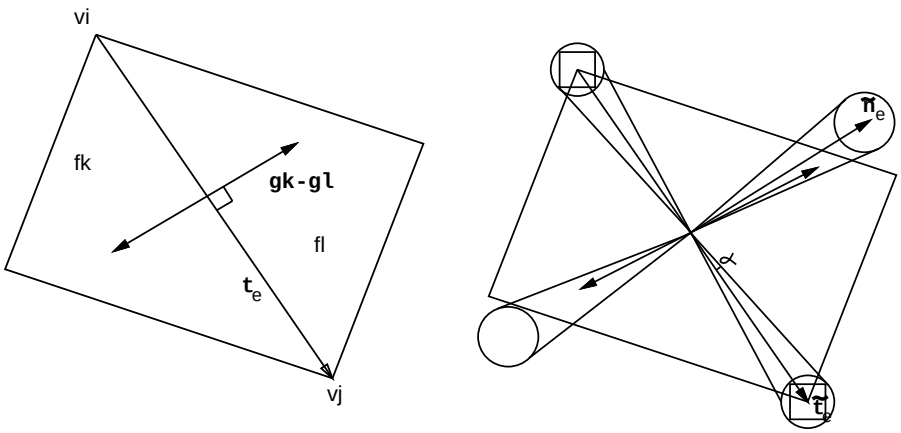

Figure 7: Edge constraints. The double arrow indicates that the orientation of $\mathbf{g}_k - \mathbf{g}_l$ is unknown.

This condition is not linear. Next, we show that the sign of $(\mathbf{g}_k - \mathbf{g}_l) \cdot \tilde{\mathbf{n}}_e$ is actually known, which brings us back to the realm of linear algebra.

We have not used the convexity (or concavity) of the edge yet. Suppose e is convex, let $\mathbf{n}_e$ be the real direction of its normal, and z_l and z_k be the points where the line parallel to the z axis in $\mathbf{x}_i + \mathbf{n}_e$ intersects the planes of f_l and f_k (Figure 8).

Since e is convex, we know that $z_k > z_l$, yielding, after some algebraic manipulation:

$$(\mathbf{g}_k - \mathbf{g}_l) \cdot \mathbf{n}_e > 0, \tag{14}$$

which determines the side of the double-cone where $\mathbf{g}_k - \mathbf{g}_l$ lies (Figure 9). This is again a well known constraint [17]. We already know that $\mathbf{g}_k - \mathbf{g}_l$ and $\mathbf{n}_e$ are aligned, so:

$$(\mathbf{g}_k - \mathbf{g}_l) \cdot \tilde{\mathbf{n}}_e > 0, \tag{15}$$

except in the degenerate case where the two vertices are so close to each other as to make $\alpha = \pi/2$.

By (3) a_{ki}, b_{ki} satisfy non-linear constraints:

$$\begin{cases} |a_{ki}| \leq \epsilon |p_k|, \\ |b_{ki}| \leq \epsilon |q_k|. \end{cases} \tag{10}$$

A similar derivation gives the following constraint for occluding edges:

$$(p_k - p_l)\tilde{x}_i + (q_k - q_l)\tilde{y}_i + (r_k - r_l) + (a_{ki} - a_{li}) + (b_{ki} - b_{li}) > 0. \tag{11}$$

The next subsection shows how to use gradient space constraints to eliminate the four quadratic constraints.

4.2 Edge Constraints

Consider an edge e with extremities v_i, v_j, and two adjacent faces f_k, f_l. Assume that e is either convex or concave. Writing that v_i and v_j belong to f_k and f_l, and eliminating the unknown depths z_i, z_j yields:

$$(\mathbf{g}_k - \mathbf{g}_l) \cdot \mathbf{t}_e = 0, \tag{12}$$

where $\mathbf{t}_e$ is the vector joining the projections of the vertices v_i and v_j into the image plane. This is the well known constraint imposed by an edge on the gradients of the adjacent faces [17].

Observe that $\mathbf{t}_e$ is necessarily contained in the double cone joining the uncertainty regions centered in $\tilde{\mathbf{x}}_i = (\tilde{x}_i, \tilde{y}_i)$ and $\tilde{\mathbf{x}}_j = (\tilde{x}_j, \tilde{y}_j)$ (Figure 7).

Let $\tilde{\mathbf{t}}_e = \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i$ be the axis of this cone, α be its half-angle, and let $\tilde{\mathbf{n}}_e$ be the vector obtained by rotating $\tilde{\mathbf{t}}_e$ 90° counterclockwise. Since $\mathbf{g}_k - \mathbf{g}_l$ and $\mathbf{t}_e$ are orthogonal, it follows that $\mathbf{g}_k - \mathbf{g}_l$ lies in the double-cone centered on $\tilde{\mathbf{n}}_e$ with half-angle α, i.e.,

$$|(g_k - \mathbf{g}_l) \cdot \tilde{\mathbf{t}}_e| \leq |(\mathbf{g}_k - \mathbf{g}_l) \cdot \tilde{\mathbf{n}}_e| \tan \alpha. \tag{13}$$

The above constraints are linear in the unknowns p_i, q_i, r_i, z_j. Sugihara gathered all such constraints into (1), where A and B are matrices whose coefficients are functions of the vertex coordinates x_j, y_j, and $\mathbf{w}$ is obtained by concatenating the unknowns for all faces and vertices. He proved that a necessary and sufficient condition for a line-drawing to correctly represent the projection of a polyhedron is that these equations admit a solution.

In the next section, we will derive similar constraints in the case where the x, y coordinates of the vertices are not known exactly.

4 Constraints under Uncertainty

We first examine the constraints imposed by incident vertices and faces, and then derive gradient space constraints associated with convex and concave edges.

4.1 Vertex Constraints

Writing that a vertex v_i lies on two faces f_k and f_l and eliminating the unknown depth z_i yields:

$$(p_k - p_l)x_i + (q_k - q_l)y_i + (r_k - r_l) = 0, \tag{7}$$

which can be rewritten as:

$$(p_k - p_l)\tilde{x}_i + (q_k - q_l)\tilde{y}_i + (r_k - r_l) + (p_k - p_l)\mu_i + (q_k - q_l)\nu_i = 0. \tag{8}$$

The quadratic terms can be eliminated by introducing the new variables $a_{ki} = p_k\mu_i, b_{ki} = q_k\nu_i$:

$$(p_k - p_l)\tilde{x}_i + (q_k - q_l)\tilde{y}_i + (r_k - r_l) + (a_{ki} - a_{li}) + (b_{ki} - b_{li}) = 0. \tag{9}$$

$z = p_i x + q_i y + r_i$ and the coordinates of v_j be (x_j, y_j, z_j), we have the following equality constraints:

$$\begin{cases} z_1 = p_1 x_1 + q_1 y_1 + r_1, \\ z_1 = p_2 x_1 + q_2 y_1 + r_2, \\ z_2 = p_2 x_2 + q_2 y_2 + r_2. \end{cases} \tag{4}$$

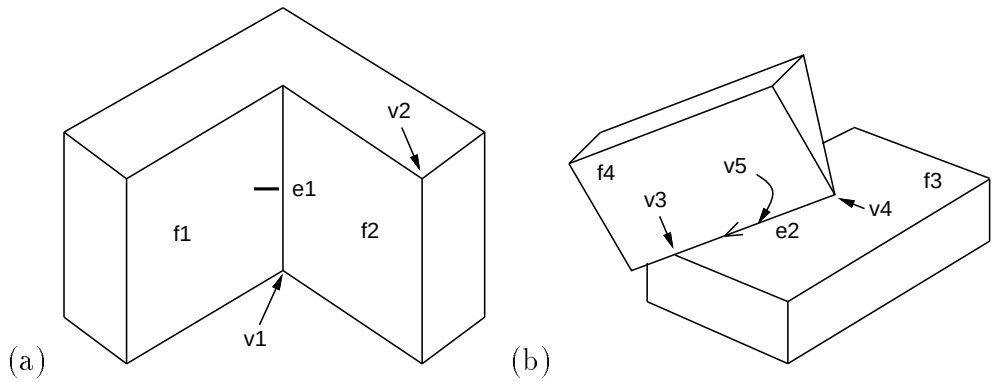

Figure 6: Line-drawings and constraints.

Since e_1 is concave, v_2 must lie *above* the plane of f_1, yielding the inequality constraint:

$$z_2 > p_1 x_2 + q_1 y_2 + r_1. \tag{5}$$

Consider now the line-drawing in Figure 6(b). Face f_4 occludes f_3 along e_2. Let v_3 and v_4 be two consecutive vertices along e_2, and v_5 be their mid-point, we have the following inequalities:

$$\begin{cases} z_3 \geq p_3 x_3 + q_3 y_3 + r_3, \\ z_4 \geq p_3 x_4 + q_3 y_4 + r_3, \\ z_5 > p_3 x_5 + q_3 y_5 + r_3. \end{cases} \tag{6}$$

Equality is allowed in the first two inequalities, but not in the third one. This is because the occluding edge e_2 can touch f_3 at some point, but not at every point (otherwise it would be a concave edge) [30]. Three equality constraints should be added, corresponding to the fact that v_3, v_4, v_5 belong to f_4.

3 Approach

3.1 Notation

We consider a labelled line-drawing, represented by a set of faces, vertices, labelled edges, and their incidence relations. Orthographic projection is assumed. The image is in the x, y plane, and the projection direction is along the z axis.

A face f is represented by its equation:

$$z = px + qy + r, \tag{2}$$

and the gradient vector (p, q) is denoted by $\mathbf{g}$.

A vertex v is represented by its image position (x, y) and its (unknown) depth z. In the presence of uncertainty, the actual position (x, y) is not known exactly. Instead, it is related to the measured image position $(\tilde{x}, \tilde{y})$ through a perturbation (μ, ν), with the constraints:

$$\begin{cases} x = \tilde{x} + \mu, & |\mu| \leq \epsilon, \\ y = \tilde{y} + \nu, & |\nu| \leq \epsilon. \end{cases} \tag{3}$$

In other words, the actual position is within some small rectangle of side 2ϵ from the measured position.

An edge e is represented by its endpoints v_i, v_j and its adjacent faces f_k, f_l. It is oriented from v_i to v_j, and f_k (resp. f_l) lies on its right (resp. its left).

3.2 Sugihara's Fundamental Equations

Assume for the moment that the image coordinates of the vertices are known perfectly. Consider the line-drawing in Figure 6(a). Faces f_1 and f_2 meet along concave edge e_1. Vertex v_1 lies on e_1, and thus on f_1, f_2, while v_2 lies on f_2 only. Let the equation of f_i be

The above applications have shown that the Lambertian model does reasonably well in describing pure diffuse reflection. There are many cases however where this model is not adequate. Forsyth and Zisserman [8] study the effects of interreflections between Lambertian surfaces on the intensity of light reflected from them. Nayar et. al. [23] present a shape-from-shading algorithm which specifically deals with interreflections.

One of the important factors which determines the reflectance is the roughness of the surface. Nayar and Oren [25] present a reflectance model for rough matte surfaces. For very smooth (mirror-like) specular surfaces, most of the reflected light is concentrated in the specular direction. Beckmann and Spizzichino [2] and Torrance and Sparrow [33] present reflectance models which deal with rough specular surfaces. Nayar et. al. [24] compare the two models and present a general model which simplifies these models and combines them with the Lambertian reflectance model. These reflectance models are used in several graphics and shape-from-shading algorithms. Cook and Torrance [5] present a modified version of the Torrance-Sparrow model for rendering images of objects; Healey and Binford [10] use the Torrance-Sparrow model to determine local shape from specular reflections; and Tagare and deFigueiredo [32] use it to recover the shape and reflectance of surfaces.

In our case we are not able to use the shape-from-shading methods described above because the smoothness assumption is violated at the edges of the object, and the shading over faces of the object is constant which does not enable us to recover the surface orientation. We therefore develop special shape recovery techniques for polyhedral objects which use the line-drawing and the shading information.

shading is needed to recover the 3D shape. The problem of recovering shape from shading has been extensively studied [11, 12, 15, 18]. Most approaches assume a Lambertian reflectance which models matte surfaces. According to that model the intensity of light reflected from a surface is proportional to the cosine of the angle between the direction of the light source and the normal to the surface and does not depend on the viewer position. Assuming that the surface of object is Lambertian, the main problem is that the shading information at a point in the image provides only one constraint on surface orientation, while surface orientation has two degrees of freedom. The shape can be recovered by adding the assumption that the object is smooth and therefore the orientation of the surface is continuous. Several methods have been proposed to solve this problem. The traditional method for solving problems mathematically similar to this one depends on growing characteristic strips [12]. In [15], Ikeuchi and Horn present a relaxation method on a grid. In [35], Woodham presents the photometric stereo method which uses several images taken with different lighting directions. The constraints obtained from each image are combined to yield the surface orientation.

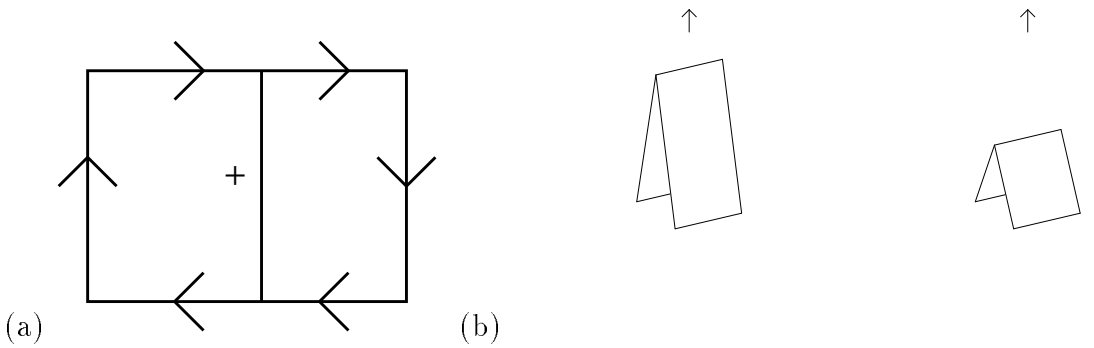

Figure 5: An infinite number of shapes which yield the same line-drawing: (a) a line-drawing; (b) two of the infinite number of shapes which yield this line-drawing.

a problem with his method is that condition (1) is too strict: minute perturbations of vertex positions can make a line-drawing incorrect. For example, the labelled line-drawing in Figure 4(a) appears to correctly represent a truncated pyramid seen from above and floating in space. Closer inspection reveals that, due to error in vertex position, the three edges that (once extended) should intersect at the pyramid's apex actually do not meet (Figure 4(b)).

This "superstrictness" problem is in fact common to most quantitative approaches to line-drawing analysis, including those based on gradient space [6]. To avoid it, Sugihara proposed to detect and delete the constraints that lead to superstrictness by using the purely combinatorial notion of *position-free incidence structures* [30]. This, unfortunately, leads to a new difficulty: The remaining conditions may not be strict enough, and incorrect line-drawings such as the one in Figure 4(c) may be classified as correct. In addition, the missing constraints introduce gaps in the recovered surfaces [30].

In this paper we will avoid superstrictness by explicitly accounting for uncertainty in vertex position. Unlike Sugihara, we do not eliminate constraints that lead to a superstrict set of equations. Instead, we explicitly introduce uncertainty in these constraints. A linear form is obtained by examining the constraints imposed by edges in gradient space [6, 14, 16, 17, 20], yielding a system of equalities and inequalities similar to (1). A necessary condition for a line-drawing to be the correct projection of a polyhedron is that this system admits a solution. It can be tested, as before, through linear programming.

2.2 Shape From Shading and Reflectance Models

As mentioned earlier, the line-drawing itself does not contain enough information for 3D shape recovery, and for each legal line-drawing there are an infinite number of shapes which could yield that line-drawing (Figure 5). Therefore more information such as

projection of some polyhedron if and only if the linear constraints

$$\begin{cases} A\mathbf{w} = 0, \\ B\mathbf{w} \geq 0, \end{cases} \tag{1}$$

admit a solution. These are the *fundamental equations* associated with the line-drawing. The matrices A and B are derived from the positions of the vertices, the incidence relations between vertices, edges, and faces, and the edge labels. The vector $\mathbf{w}$ denotes the unknown face parameters of the observed polygon.

Thus, linear programming can be used to determine whether a line-drawing is "correct", i.e., whether there exists some polyhedron projecting onto it [31]. Due to the loss of depth information in the imaging process, a correct line-drawing admits an infinite number of interpretations. In other words, if there exists a solution $\mathbf{w}$ to (1), then there exists an infinite number of distinct solutions. However, when additional cues such as intensity or texture are available, it is possible to select a unique 3D interpretation $\mathbf{w}$ through non-linear optimization under the linear constraints (1) [30].

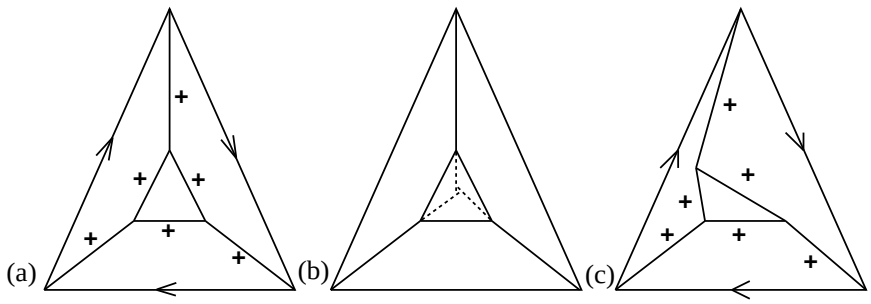

Figure 4: (a) A "correct looking" labelled line-drawing. (b) Superstrictness. (c) An incorrect line-drawing.

Sugihara's approach is rigorous and elegant. It has been called "the final breakthrough in quantitative (line-drawing) analysis" [12]. However, as remarked by Sugihara himself,

The main problem that arises with line-drawing labelling is that even line-drawings that have consistent labellings are not guaranteed to be a picture of a real polyhedron. For example, Figure 3(a) shows a line-drawing with a consistent labelling, but admits no correct interpretations with planar faces (Figure 3(b)), because any edges between two planar faces must be collinear, which is not the case for faces A and B in the figure. Quantitative approaches using vertex position information are used to address this problem. Each face appearing in the image is parameterized by $z = px + qy + r$, where the pair (p, q) is the gradient of the face, a point in gradient space. The legality of the line-drawing is shown by proving that a set of consistent parameters can be found for all the faces in the image. These methods use equality and inequality constraints on the gradient space, known as gradient space constraints. Mackworth [20], Huffman [14], Kanade [16, 17], and Draper [6] use these constraints to determine whether a line-drawing could be a picture of a polyhedral object.

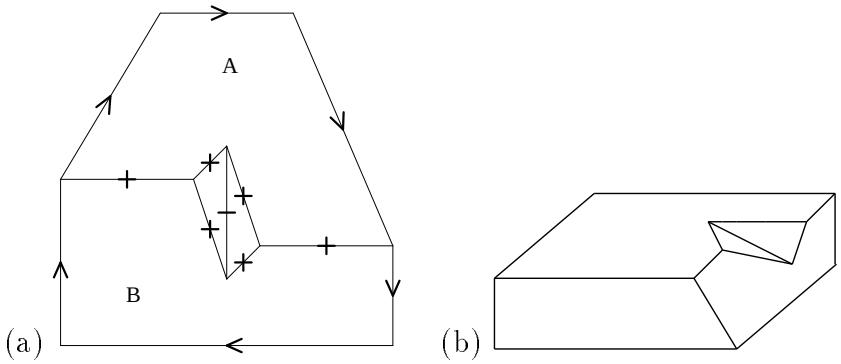

Figure 3: A line-drawing with a consistent labelling that only non-polyhedral objects yield: (a) a line-drawing with a consistent labelling; (b) a non-polyhedral object which yields this line-drawing.

In [30, 31], Sugihara proved that a labelled line-drawing correctly represents the

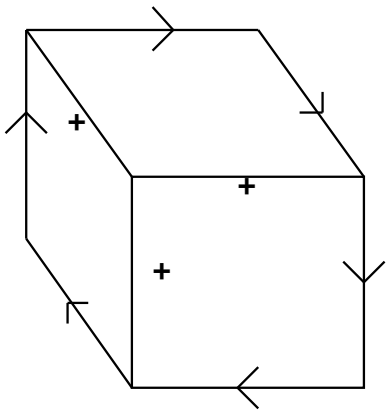

Figure 1: Labelling of a line-drawing of a cube.

edge in the line-drawing as convex (+ label), concave (−), or occluding (← or →). For occluding edges, the occluding face lies on the right when one looks in the direction of the arrow label. For example, Figure 1 shows the labelled line-drawing of a cube floating in space. The outer edges are occluding ones, while the inner ones are convex. Several algorithms have been proposed to solve this problem. For example Huffman [13] and Clowes [3] study objects with trihedral junctions (Figure 2), and show that only a small number of the possible labellings for each type of junction actually occurs in polyhedral objects. In [34], Waltz presents a constraint propagation algorithm which exploits these constraints to efficiently find all the consistent labellings of a line-drawing.

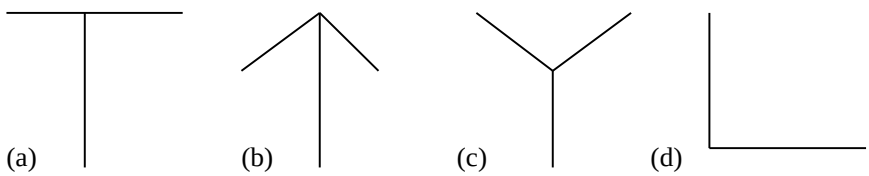

Figure 2: Types of trihedral junctions: (a) T junction, (b) arrow junction, (c) fork junction (d) L junction.

1 Introduction

We address the problem of deciding whether a labelled line-drawing [3, 7, 9, 13, 26, 34] represents a picture of some polyhedron, and, in this case, reconstructing a polyhedron that projects onto the line-drawing [6, 14, 16, 17, 20, 30, 31] using shading information. We propose an approach that combines Sugihara's algebraic characterization of line-drawings of polyhedra [28, 30, 31] with gradient space constraints [6, 14, 16, 17, 20], shape-from-shading techniques [8, 24], and radiosity methods [4, 21]. Its novelty is that it explicitly takes into account uncertainty in vertex position and phenomena such as interreflections. Its main advantages are its efficiency in using linear programming to reject incorrect scene interpretations and its ability to perform shape recovery using real images of objects with complex reflectance properties [2, 8, 24, 23, 33].

The rest of the paper is organized as follows. Section 2 discusses the state of the art in line-drawing analysis and 3D shape recovery. We introduce our approach in Section 3. We present an algorithm for line-drawing analysis which takes into account uncertainty in vertex position in Section 4, and present our 3D shape recovery algorithm in Section 5. Implementation and results are presented in Section 6. Finally, a number of issues raised by our algorithm and its implementation are discussed in Section 7.

2 Literature Review

2.1 Line-Drawing Analysis

The problem of labelling a line-drawing and deciding whether a labelled line-drawing represents a correct picture of some polyhedron has been studied extensively [3, 7, 9, 13]. Given a line-drawing of a polyhedron, the labelling problem is to correctly label each

Recovering The Shape of Polyhedra Using Line-Drawing Analysis and Complex Reflectance Models

Ilan Shimshoni and Jean Ponce

Abstract

Following Sugihara [30], we represent the geometric constraints imposed by the line-drawing of a polyhedron as a set of linear equalities and inequalities. Unlike him, we explicitly take into account the uncertainty in vertex position. This allows us to circumvent the superstrictness of the constraints without deleting any of them. For a given error bound, deciding whether a line-drawing is the correct projection of a polyhedron is reduced to linear programming, and 3D shape recovery is reduced to optimization under linear constraints. Our method can be used for recovering the shape of any polyhedral object whose reflectance can be modelled accurately. We have implemented it for the following reflectance models: the Lambertian model, a the Lambertian model with interreflections, and a reflectance model for specular objects. We present results obtained using real images.

Recovering The Shape of Polyhedra Using Line-Drawing Analysis and Complex Reflectance Models

Ilan Shimshoni and Jean Ponce
Department of Computer Science and Beckman Institute
University of Illinois, Urbana, IL 61801

Ilan Shimshoni is currently with the Computer Science Department, Technion, Haifa, Israel, so please send all communications to:
Ilan Shimshoni Computer Science Department, Technion, Haifa 32000, Israel
Tel. 972-4-294-528 Fax. 972-4-294353
e-mail: ilan@cs.technion.ac.il