'93 KACC (1993.10.20~22)

Estimating 3-D Surface Geometrical Features on the Basis of Surface Curvature Consistency

H.B. Zha, S. Muramatsu and T. Nagata
Faculty of Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812, JAPAN

Abstract

This paper presents a method of estimating 3-D surface geometrical features that are necessary for 3-D object recognition and image interpretation. The features, such as surface needle maps and curvatures, are computed from range or intensity images. In general, the range and intensity images are prone to noises, and hence the features computed by differentiation calculi on such a noisy image are hardly applicable to industrial recognition tasks. In our approach, we try to obtain a more accurate estimate of the features by using a least-squares minimization procedure subject to local curvature consistency constraints. The algorithm is robust with respect to noises and is completely independent of the viewpoint at which the image is taken. The performance of the algorithm is evaluated using both synthetic data and real intensity images.

1 Introduction

Surface characterization based on higher-level 3-D geometrical features is a primary step to the goal of 3-D object recognition and image interpretation [1, 2, 3]. Some of the features, such as *surface needle maps and curvatures*, can be computed from range images which are acquired either by using active sensors or from stereoscopic cues and shapes recovered from shading analysis [3, 4, 5]. In general, the range and intensity images are noise-contaminated because of low accuracy of available range finders and uncertainty in image formation. Consequently, the higher-level geometrical features computed by the differentiation calculi on such a noisy image are usually too inaccurate to find applications in real recognition tasks.

In this paper, we propose a *shape refinement algorithm* that uses constraints on the surface curvature consistency to stabilize and correct real 3-D shape measurements. Given any kinds of 3-D image data, the algorithm computes the *augmented Darboux frame (ADF)*.which is a second-order differential description of 3-D surface, for each image point by utilizing a local *quadric surface model* [3, 4]. The resultant ADFs within a small neighbor of each point are then examined to find local distortion in the local surface shapes. A *least-squares minimization* procedure is formed to minimize the local shape distortions over the whole image in an iterative manner. The use of ADFs in the local shape refinements is the greatest feature of our approach. It makes the algorithm completely independent of the viewpoint at which the image is taken.

In our shape refinement algorithm, a needle map is used as the initial estimation of surface shapes over the whole image. Many researchers have proposed a lot of methods to estimate needle maps from range and/or intensity images [6, 7]. Our algorithm can be applied to any kind of needle maps no matter whether they are extracted from range or intensity images. In our experiments, the algorithm shows significant robustness with respect to noisy needle maps derived from single intensity images. It also demonstrates a good convergency for images including either simple or complex curved object surfaces.

2 Estimating Augmented Darboux Frames(ADFs) from Needle Maps

The section describes a method of estimating ADFs of 3-D surfaces from a needle map by utilizing a local quadric surface model. It begins with a brief description of the local surface model and then give a simple introduction of the method for computing ADFs.

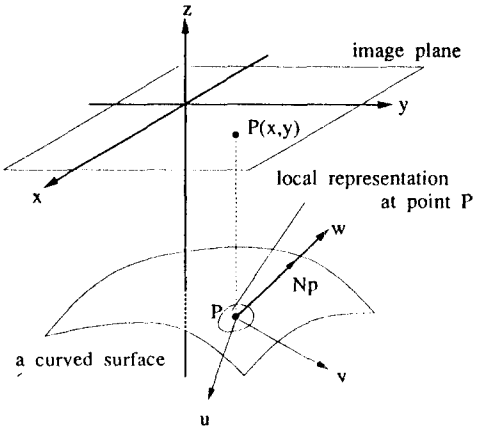


Fig. 1: A coordinate system for local surface representation.

2.1 A Local Quadric Surface Model

For a curved surface, we assume that a patch around any point P can be locally represented by a quadric of the form

$$w = au^2 + buv + cv^2, \tag{1}$$

with origin at P, $u-$ and $v-$axes on the tangent plane at P, and the $w-$axis aligned with the surface normal

$$N_P = (N_P^u, N_P^v, N_P^w), \tag{2}$$

as shown in Fig.1. In general, this assumption is sufficiently accurate for any smooth surface within a small domain in the (u,v) space.

With such assumptions, we have

$$\frac{N_P^u}{N_P^w} = -2au - bv, \tag{3}$$

$$\frac{N_P^v}{N_P^w} = -2cv - bu. \tag{4}$$

Given a needle map, which represents the surface normals on the sampling grids corresponding to pixels of a general image, the coefficients of the quadric form at P can be determined by standard estimation methods. In our approach, the estimation is carried out by a least squares algorithm that makes use of normal vectors at at P itself and at its 8 neighboring points.

2.2 ADF and Its Computation from Local Surface Model

3-D surface geometrical features that can be computed by utilizing the local quadric surface model at a point P are as follows: the *principal curvatures* κ_{M_P} and κ_{m_P}, which are maximum and minimum values of the normal curvatures at the point P, the *principal direction vectors* M_P and $\mathcal{M}_P$, which are the directions in which the principal curvatures are computed respectively, and the normal vector, N_P. Following the convention of [3], we refer to

$$D(P) = \{P, M_P, \mathcal{M}_P, N_P, \kappa_{M_P}, \kappa_{m_P}\} \tag{5}$$

collectively as the augmented Darboux frame (ADF) at point P. By well-known orthogonality between any two of M_P, $\mathcal{M}_P$ and N_P, the ADF forms a local coordinate system on any surface point. The local surface model at point P, therefore, can be represented by a quadric of the form

$$w = \frac{1}{2}(\kappa_{M_P}u^2 + \kappa_{m_P}v^2), \tag{6}$$

with origin at P and the $w-$, $u-$ and $v-$axes parallel to N_P, M_P and $\mathcal{M}_P$ respectively.

By representing a local surface patch with P $(u,v,h(u,v))$, where $h(u,v)$ is a quadric given in (1), the mathematical formulas to estimate the ADF at any point (u,v) are given as follows.

Given the partial derivatives,

$$P_u = (1, 0, h_u), P_v = (0, 1, h_v),$$

$$P_{uu} = (0, 0, h_{uu}), P_{uv} = (0, 0, h_{uv}), P_{vv} = (0, 0, h_{vv}), \tag{7}$$

the normal vector is computed as

$$N_P = \frac{P_u \times P_v}{|P_u \times P_v|} = \frac{(-h_u, -h_v, 1)}{\sqrt{1 + h_u^2 + h_v^2}}. \tag{8}$$

Now let

$$E = (P_u \cdot P_u) = 1 + h_u^2,$$

$$F = (P_u \cdot P_v) = (P_v \cdot P_u) = h_u h_v,$$

$$G = (P_v \cdot P_v) = 1 + h_v^2,$$

$$e = (P_{uu} \cdot N) = \frac{h_{uu}}{\sqrt{1 + h_u^2 + h_v^2}}, \tag{9}$$

$$f = (P_{uv} \cdot N) = \frac{h_{uv}}{\sqrt{1 + h_u^2 + h_v^2}},$$

$$g = (P_{vv} \cdot N) = \frac{h_{vv}}{\sqrt{1 + h_u^2 + h_v^2}},$$

where $(a \cdot b)$ denotes the inner product of a and b, and

$$\mathcal{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

where,

$$a_{11} = \frac{fF - eG}{EG - F^2}, a_{12} = \frac{gF - fG}{EG - F^2}, \tag{10}$$

$$a_{21} = \frac{eF - fE}{EG - F^2}, a_{22} = \frac{fF - gE}{EG - F^2}.$$

The eigenvalues of $\mathcal{A}$ are the principal curvatures κ_{M_P} and κ_{m_P}, and the eigenvectors of $\mathcal{A}$ are the principal direction vectors M_P and $\mathcal{M}_P$. They can be computed, respectively, as

$$-\kappa_{M_P} = \frac{1}{2}(a_{11} + a_{22} - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}), \tag{11}$$

$$-\kappa_{m_P} = \frac{1}{2}(a_{11} + a_{22} + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}), \tag{12}$$

and, if $a_{11} \geq a_{22}$,

$$M_P = (a_{12}, -\frac{1}{2}(a_{11} - a_{22} + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}})),$$

$$\mathcal{M}_P = (\frac{1}{2}(a_{11} - a_{22} + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}); \tag{13}$$

if $a_{11} < a_{22}$,

$$M_P = (\frac{1}{2}(a_{11} - a_{22} - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}), a_{21}),$$

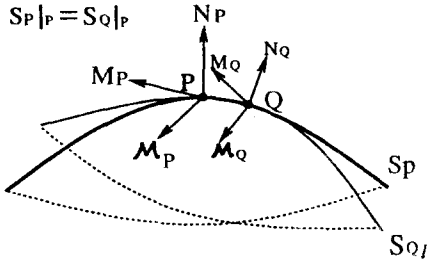
$$\mathcal{M}_P = (-a_{12}, \frac{1}{2}(a_{11} - a_{22} - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}})). \tag{14}$$

3 Shape Refinement Algorithm

3.1 The Refinement Algorithm of ADFs

Although it can approximately represent the local surface shapes, the ADF at any surface point is also subject to errors arising from the usual effects of noises and quantization errors. In particular, the ADFs computed by the differentiation calculi, as shown in the earlier discussion, on noisy image data will be very inaccurate. To eliminate the harmful effects of noises, it is necessary to refine the obtained ADFs on the basis of some smoothness constraints on general surfaces.

Given densely sampled 3-D data, we can show that the local shape at the surface point P has a tight relationship with shapes of its near-neighbors. In other words, the ADF at P should have the same value no matter whether it is



Sp : Local representation at point P.

SoI : Local representation at point Q.

Fig. 2: Continuity principle on a smoothly curved surface.

computed from the local quadric representation determined at $\boldsymbol{P}$ itself or at any other point within a near-neighborhood of $\boldsymbol{P}$. This fundamental principle of surface continuity is schematically shown in Fig.2, and it will provide a mechanism for updating ADF at the point $\boldsymbol{P}$ from ADFs at its surrounding neighborhood.

The steps of the iterative refinement algorithm for estimating $D(\boldsymbol{P})$ at point $\boldsymbol{P}$ are as follows.

Given any needle map that is represented in a global, sensor-centered coordinate system,

1. compute $D(\boldsymbol{P})$ in the local coordinate system determined at $\boldsymbol{P}$ and transform it into the global coordinate system;
2. compute ADFs $D(\boldsymbol{P}_l)$ from the surrounding neighborhood in each local coordinate system determined at each near-surrounding point by utilizing the local surface model in (6);
3. transform $D(\boldsymbol{P}_l)$ into the global coordinate system in which the original needle map is provided;
4. compare transformed $D(\boldsymbol{P})$ and $D(\boldsymbol{P}_l)$ and update $D(\boldsymbol{P})$ by using a least-squares minimization procedure.

These steps are carried out over the whole image, and the updating is repeated so that the shape distortions accumulated over the whole image are eliminated in the viewpoint of least-squares optimization. The shape distortions are evaluated by

$$\begin{aligned} E_P &= \sum_{l=1}^{n}[(|\boldsymbol{M}_P - \boldsymbol{M}_{P_l}|)^2 + (|\boldsymbol{N}_P - \boldsymbol{N}_{P_l}|)^2 \\ &\quad +(\kappa_{M_P} - \kappa_{M_{P_l}})^2 + (\kappa_{m_P} - \kappa_{m_{P_l}})^2] \\ &\quad +\lambda((|\boldsymbol{M}_P|)^2 + (|\boldsymbol{N}_P|)^2 - 2), \end{aligned} \tag{15}$$

where $\{\boldsymbol{P},\ \boldsymbol{M}_P,\ \boldsymbol{\mathcal{M}}_P,\ \boldsymbol{N}_P,\ \kappa_{M_P},\ \kappa_{m_P}\}$ is the ADF determined at point $\boldsymbol{P}$; $\{\boldsymbol{P}_l,\ \boldsymbol{M}_{P_l},\ \boldsymbol{\mathcal{M}}_{P_l},\ \boldsymbol{N}_{P_l},\ \kappa_{M_{P_l}},\ \kappa_{m_{P_l}}\}$ the ADF determined at the surrounding point $\mathbf{P}_l$, and n is the number of all near-surrounding points. Using standard least-squares methods, the updating functionals for $\boldsymbol{M}_P$, $\boldsymbol{N}_P$,κ_{M_P} and κ_{m_P} are

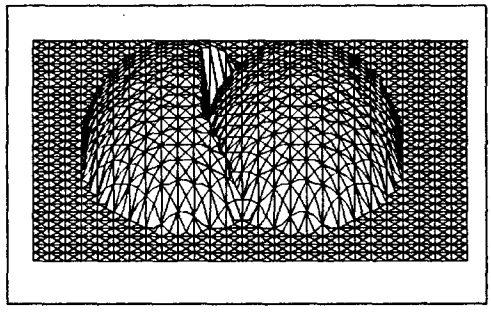


Fig. 3: Two overlapping ellipsoids used in our experiment for determining C (They are illustrated by using surfaces recovered from refined ADFs).

$$\boldsymbol{N}_P^{(i+1)} = \frac{(\sum_{l=1}^{n} N_{P_l}^{x(i)}, \sum_{l=1}^{n} N_{P_l}^{y(i)}, \sum_{l=1}^{n} N_{P_l}^{z(i)})}{\sqrt{(\sum_{l=1}^{n} N_{P_l}^{x(i)})^2 + (\sum_{l=1}^{n} N_{P_l}^{y(i)})^2 + (\sum_{l=1}^{n} N_{P_l}^{z(i)})^2}}, \tag{16}$$

$$\boldsymbol{M}_P^{(i+1)} = \frac{(\sum_{l=1}^{n} M_{P_l}^{x(i)}, \sum_{l=1}^{n} M_{P_l}^{y(i)}, \sum_{l=1}^{n} M_{P_l}^{z(i)})}{\sqrt{(\sum_{l=1}^{n} M_{P_l}^{x(i)})^2 + (\sum_{l=1}^{n} M_{P_l}^{y(i)})^2 + (\sum_{l=1}^{n} M_{P_l}^{z(i)})^2}} \tag{17}$$

$$\kappa_{M_{P_l}}^{(i+1)} = \sum_{l=1}^{n} \frac{\kappa_{M_{P_l}}}{n} \tag{18}$$

and

$$\kappa_{m_{P_l}}^{(i+1)} = \sum_{l=1}^{n} \frac{\kappa_{m_{P_l}}}{n}, \tag{19}$$

where the superscript i refers to the current iteration step.

To investigate effectiveness of the algorithm, we have summation of $E_P^{(i)}$ over the whole image as

$$R^{(i)} = \sum_{P \in \mathcal{I}} E_P^{(i)}, \tag{20}$$

where $\mathcal{I}$ represents the set of all image points.

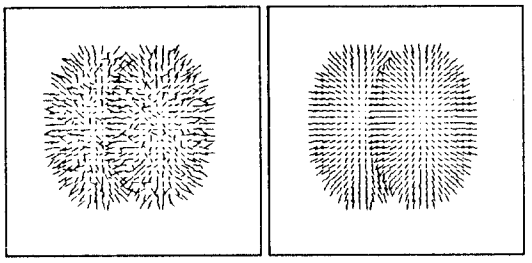
The algorithm is allowed to iterate until satisfying

$$|R^{(i-1)} - R^{(i)}| < CR^{(i-1)}, \tag{21}$$

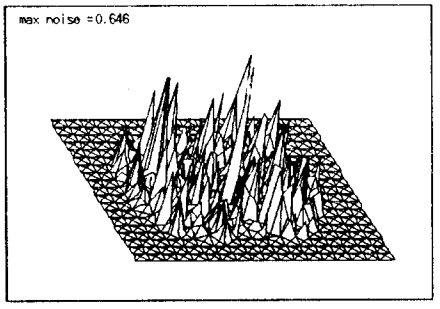
where C is a thresholding parameter. In the next section, we will show how the parameter is determined empirically in our approach.

3.2 Determination of Parameter C

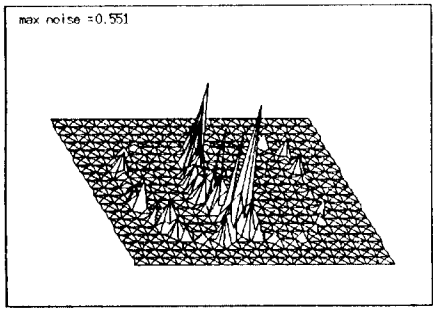
In general, the smaller $R^{(i)}$ in (20) is, the more accurate the recovered surface according to the surface continuity principle. That is to say that a very small C is preferable in our algorithm. On the other hand, any image of complex objects usually includes a large number of discontinuous points at edges of the object surfaces, and too small C will, against our expectation, make the recovered object surfaces extremely erroneous due to the blurred discontinuity. Therefore, it is important to choose an appropriate



(a) (b)



(c)



(d)

Fig. 4: Results of the experiment for determining the parameter C.

C for dealing with problems of recovering complex object surfaces. In our approach, some simulation experiments are performed for the purpose of the parameter determination and one of them is described in the following.

In the experiment, two overlapping ellipsoids with moderate eccentricities (axis ratios of 25:40:25) are used as object surfaces to be recovered using our shape refinement algorithm. They are illustrated in Fig.3. Fig.4(a) shows the needle map that is derived by adding white noises (σ^2=0.05) to the real map given by theoretical computation. The difference between the real map and each updated map based on the noisy data is

$$D^{(i)} = \sum_{p \in \mathcal{I}} d_p^{(i)} = \sum_{p \in \mathcal{I}} |1 - (\boldsymbol{N}_p \cdot \boldsymbol{N}_p^{(i)})|, \qquad (22)$$

where $\boldsymbol{N}_p$ is the real normal vector. The distribution of $d_p^{(0)}$ is shown in Fig.4(c). After the application of the refinement

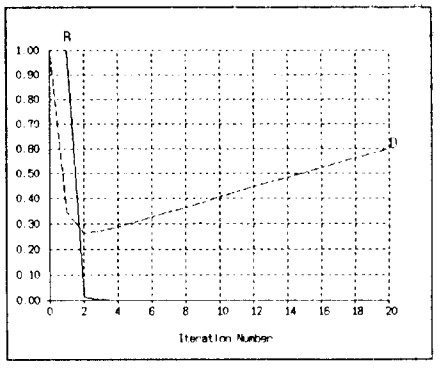


Fig. 5: Values of $R^{(i)}/R^{(1)}$ and $D^{(i)}/D^{(0)}$.

algorithm in 5 iterations, we have a needle map as shown in Fig.4(b) and the distribution of $d_p^{(5)}$ is shown in Fig.4(d). Fig.5 shows the values of $R^{(i)}/R^{(1)}$ and $D^{(i)}/D^{(0)}$, and we define a new function

$$F^{(i)} = \frac{R^{(i)}}{R^{(2)}} + \frac{D^{(i)}}{D^{(1)}} \qquad (23)$$

for choosing an appropriate C. $R^{(2)}$ and $D^{(1)}$, instead of $R^{(1)}$ and $D^{(0)}$, are used here because $R^{(1)}$ and $D^{(0)}$ resulting from the initial estimation are usually too large. Table 1 shows the values of $F^{(i)}$ with respect to i in the range {2,8}.

i	2	3	4	5	6	7	8
$F^{(i)}$	2.000	1.026	0.963	0.976	1.007	1.051	1.099

Table 1: Values of $F^{(i)}$ with respect to i.

On the basis of the results, we choose $C = 0.35$ since $F^{(i)}$ reaches its smallest value after the 4th iteration and at the point

$$\frac{|R^{(3)} - R^{(4)}|}{R^{(3)}} \cong 0.35. \qquad (24)$$

4 Results of Experiments

4.1 Initial Estimation of Needle Maps

The shape refinement algorithm proposed here can be applied to any kind of needle maps as the initial estimation. In our experiments, this initial estimation is obtained by using Pentland's *local shading analysis method* [6]. There are some reasons for using the method as shown in the following. 1) Pentland's method is an algorithm which uses only some simple computations on a single intensity image but can estimate the normal vector at every image point. 2) Needle maps estimated by using Pentland's method include a lot of noises, which is suitable for us to show robustness of our algorithm. The Pentland'd method is briefly described in the following.

There are a number of assumptions about the needle map estimation method: 1) Position of the light source is known or can be estimated. 2) The reflectance of surface is Lambertian. 3) The image is formed in orthographic projection

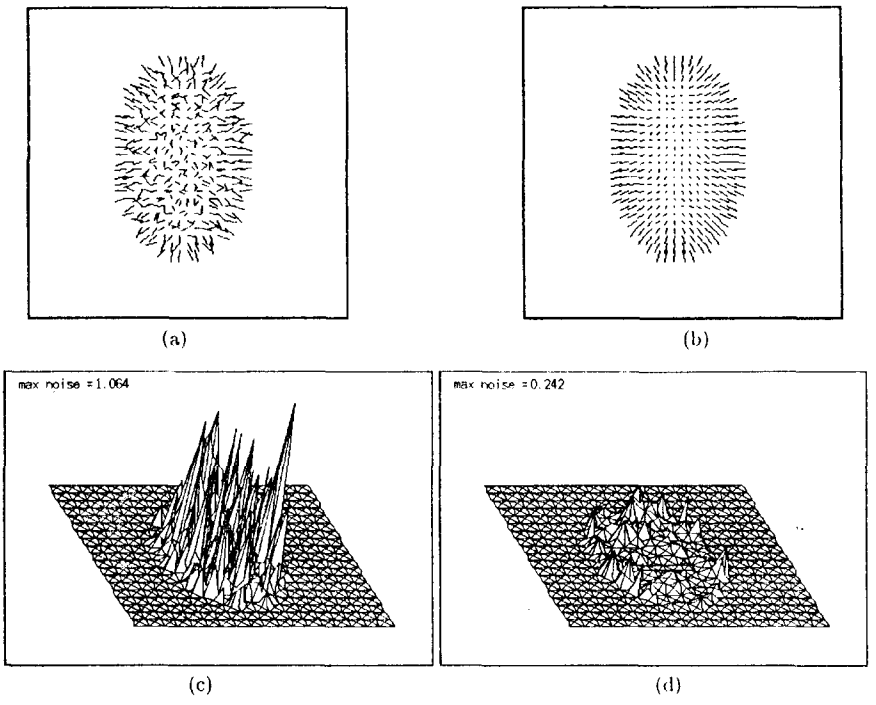


Fig. 6: Results of an experiment using synthetic intensity image.

onto the image plane. For this simplified model an approximation of the image irradiance equation is given by

$$I = \rho\lambda(\boldsymbol{N} \cdot \boldsymbol{L}), \tag{25}$$

where I is intensity value of the image point, ρ corresponds to the surface albedo, λ the incident light flux, and $\boldsymbol{N}$ and $\boldsymbol{L}$ the surface normal vector and the light source direction vector respectively. Without any loss in generality, consider the case where the light source is the viewer position. In this case, it can be shown that an estimate of the unit surface normal $\boldsymbol{N}_P$ at P is given by

$$N^x = \cos(\tau)\sin(\sigma), N^y = \sin(\tau)\sin(\sigma), N^z = \frac{I}{\rho\lambda}, \tag{26}$$

where τ is equal to the orientation of the intensity gradient at the image of P and σ is $\cos^{-1}(N^z)$ [8]. In this way, we can estimate a needle map from a single intensity image.

4.2 Results of Experiments Using Synthetic and Real Intensity Images

We have implemented the algorithm in a workstation-centered vision system and made a large number of experiments on synthetic and real images. Some results are given and discussed as follows.

Fig.6(a) shows the needle map of a ellipsoid with an eccentricity of 25:40:25. It is estimated from a noisy intensity image (added with white noises of a variance 0.02) with 8bits/pixel by Pentland's method. Application of the refinement algorithm (5 iterations) produces a needle map shown in Fig.6(b). Fig.6(c),(d) show the errors D in (22) for the initial estimated needle map and the map produced after the shape refinement. The results give an intuitive demonstration of the robustness of our algorithm with respect to noises.

Next experiment is performed by using a real intensity image of a complex surface object. Fig.7(a) shows the intensity image of a bust of Miro (256 × 240 and 8 bits/pixel). Fig.7(b) is a needle map estimated from Fig.7(a) by using Pentland's method. The needle map, produced by the refinement algorithm (5 iterations), is shown in Fig.7(c). Fig.7(d),(e) show the reconstructed surfaces from two needle maps, respectively. It is illustrated that the refinement algorithm makes the initial estimation of needle map considerably smooth while preserving the topological properties (concave, convex and so on) of its surfaces.

We have found over a large number of experiments that the algorithm generaly produces stable results within 6 iterations, even for images including complex object surfaces.

5 Conclusions

We present a shape refinement algorithm based on curvature consistency constraints. The use of ADF makes the algorithm completely independent of the view point, and it is very helpful for recognize 3-D curved surface objects.

The performance of the algorithm shows a significant ro-

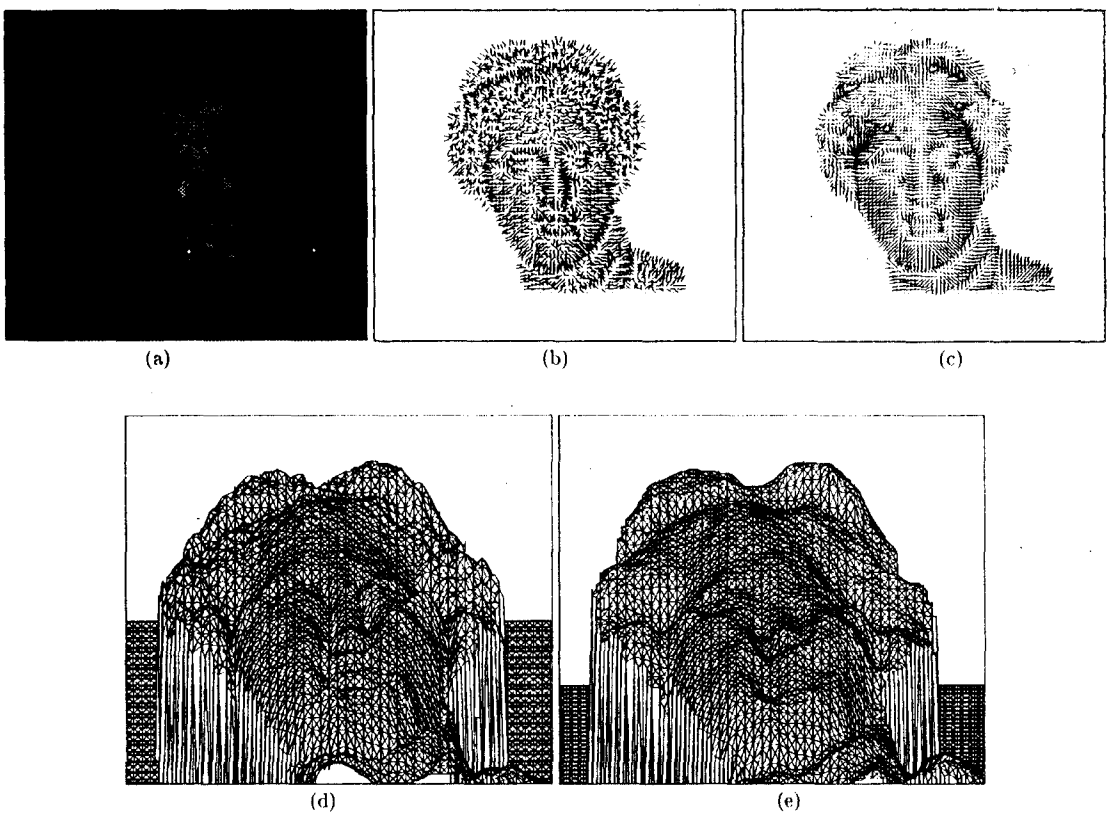


Fig. 7: Results of an experiment using the real intensity image of a complex surface object.

bustness in the computation of higher-level geometrical features with a good convergence property. However, because the algorithm assumes that the curved surface is always continuous, it will make surface discontinuity blurred. In our future work, discontinuity-preserving mechanism should be added to the algorithm to improve its robustness with respect to complexity of the observed object surfaces in the images.

References

[1] T.Nagata and H.B.Zha, "Determining Orientation, Local and Size of Primitive Surfaces by a modified Hough Transformation", *Pattern Recognition*, Vol.21, No.5, pp.481-491 (1988)

[2] T.-J.Fan, *Describing and Recognizing 3-D Objects Using Surface Properties*, Springer-Verlag New York Inc., pp.27-54 (1990)

[3] P.T.Sander and S.W.Zucker, "Inferring Surface Trace and Differential Structure from 3-D Images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.12, pp.833-854 (1990)

[4] F.P.Ferrie and J.Lagarde, "Curvature Consistency Improves Local Shading Analysis", *CVGIP: Image Understanding*, Vol.55, No.1, pp.95-105 (1992)

[5] I. Cohen, L.D. Cohen and N. Ayache, "Using Deformable Surfaces to Segment 3-D Images and Infer Differential Structures", *CVGIP: Image Understanding*, Vol.56, No.2, pp.242-263 (1992)

[6] A.P.Pentland, "Local Shading Analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.6, pp.170-187 (1984)

[7] Q.Zheng and R.Chellappa, "Estimation of Illuminant Direction, Albedo, and Shape from Shading", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.13, pp.680-702 (1991)

[8] F.P.Ferrie and M.D.Levine, "Where and why local shading analysis works", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.11, pp.198-205 (1989)