9-1-1993

An Efficient Method for Computing Volume and Surface Area Over Grid Sampled Surfaces

June Ho Yi
Purdue University School of Electrical Engineering

David M. Chelberg
Purdue University School of Electrical Engineering

AN EFFICIENT METHOD FOR COMPUTING VOLUME AND SURFACE AREA OVER GRID SAMPLED SURFACES

JUNE HO YI
DAVID M. CHELBERG

TR-EE 93-31
SEPTEMBER 1993

SCHOOL OF ELECTRICAL ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

An Efficient Method for Computing Volume and Surface Area Over Grid Sampled Surfaces*

June Ho Yi
David M. Chelberg

Geometric Modeling and Perceptual Processing Laboratory
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907-1285

'This work was supported by a Digital Equipment Corporation Faculty Incentives for Excellence Grant, and by NSF Grant number IRI-9011421.

Abstract

In this paper, we propose the volume between two surfaces normalized by the surface area as an invariant quantitative measure for comparing surface reconstruction results. The invariant property of the volume quantity provides the same measure with respect to an arbitrary coordinate system. By normalizing the volume by the surface area, the values of the measure can be compared for different size of images. We also present a computationally simple and efficient way of computing the volume between two surfaces and the surface area using a least-square-error fit plane approximation of a surface patch defined over a rectangular grid. Experiments indicate that the method using a least-square-error fit plane approximation gives equivalent performance as other more complicated and computationally expensive methods. The advantage of this method is that computation is extremely simple and efficient. Similarly, we propose the area between two curves normalized by the arc length as an invariant measure for comparing plane curve reconstruction results.

Contents

List of Figures

List of Tables

1 Introduction

In computer vision, reconstruction of a curve or surface is necessary to derive a complete representation of a curve or surface from sparse noisy sets of geometric information, such as depth and orientation or other sources of information. A reconstructed curve or surface is an intermediate representation to bridge the gap between the sensor data and the symbolic descriptions.

Considerable research has been devoted to the problem of the reconstruction of visible surfaces [1-10] [13-15] [17-24]. Reconstruction results from different reconstruction methods have been compared in order to decide which method performs better than the others by showing their visual differences using three dimensional displays or displays of a slice of reconstructed surfaces. However, use of only visual displays makes it difficult to see small differences between reconstruction results when they are visually similar. It is also difficult to determine how much better one method performs than others. A quantitative measure is necessary for a precise and more informative comparison of reconstruction results from different reconstruction methods in order to decide how much better one method performs than others. A quantitative measure alone can be used as a concise, precise, and informative comparison method of reconstruction results.

A good quantitative measure for comparing reconstruction results should have several properties. The measure must be invariant with respect to an arbitrary coordinate system. If a measure is not invariant with respect to a coordinate system, then it is possible to obtain inconsistent performance measures. For example, two reconstruction methods A and B are to be compared. Suppose that a surface is reconstructed by methods A and B in two different coordinate systems. A noninvariant performance measure may lead to a conclusion that method A is better than method B in one coordinate system but is worse in the other coordinate system. It is also desirable to use a measure that produces consistent results for different sizes of images. Finally, the value of the measure should be able to be computed simply and efficiently with reasonable accuracy. In this work, we propose the volume between two surfaces normalized by the surface area as an invariant measure for comparing reconstruction results. We also present a computationally simple and efficient method of computing the volume between two surfaces and the surface area. The time complexity of our method to compute the volume and the area is $O(\text{number of pixels})$.

The metrics induced by the L^1, L^2 or L^∞ norms have been popularly used in order to give

a quantitative measure for comparison of reconstruction results. The use of these metrics has often naturally resulted because some reconstruction methods employ minimization techniques of L^1, L^2 or L^w error between the true or target data set and the measurement data set (constraints) to compute the reconstruction results. In Chu and Bovik's work [6], the reconstruction results were computed by minimizing the L^w error (maximum absolute error) and the normalized L^1 (average absolute error) and L^w error were used as performance measures. On the other hand, the L^2 metric has mostly been used as a reasonable performance measure irrespective of reconstruction methods [19] [14] [17]. L^2 and L^w measures were computed in [19]. Sinha and Schunck [17] employed a RMS error measure which is the normalized L^2 metric for comparison of two different reconstructions. The following is the discrete form of L^1 and L^2 metrics normalized by the number of the sampled points and L^∞ metric for two functions, f and g in the two dimensional case.

$$
\begin{aligned}
L_d^1 metric &= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} |f(i,j) - g(i,j)| \\
L_d^2 metric &= \frac{1}{nm} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} \{f(i,j) - g(i,j)\}^2 \right]^{\frac{1}{2}} \\
L_d^\infty metric &= \sup_{i,j} |f(i,j) - g(i,j)|
\end{aligned}
$$

where $f(i,j)$ and $g(i,j)$, $i = 1,2,...,n$, $j = 1,2,...,m$ are sampled points of two functions, $f(x,y)$ and $g(x,y)$ respectively. The L^1, L^2 and L^w metrics, however, are not invariant with respect to a coordinate system, resulting in different measures in different coordinate systems. If the L^2 metric is used as a difference measure, where the slope is high, the difference value is emphasized more than in flat regions.

Not much interest has been raised in the computer vision area regarding the problem of computing volume and surface area except for the problem of estimating the volume or the surface area of biological organs from cardiographic data or tomographic data. On the other hand, in the areas of CAD/CAM and robotics, the automatic computation of integral properties such as surface area, volume, centroid, and moments for geometrically complex solids has been an important problem. These integral properties of solids are defined by triple (volumetric) integrals over subsets of three dimensional Euclidean space. A survey of algorithms for computing volume and other integral properties of solids is given in [12]. Most known methods for calculating integral properties may be associated naturally with the representation methods which may be organized as primitive instancing, quasidisjoint decomposition, simple sweeping, boundary representation, and constructive solid geometry (CSG) [16]. However, these methods can not be directly applied to the problem of estimating the volume or the surface

area in computer vision. In computer vision, most initial data to start with is sensor data. This sensor data is then processed to provide intermediate representations similar to those listed in the above. Considering that regions between sampled pixels of the original sensed image are ambiguous, estimating the volume or the surface area from the intermediate representations computed from the original data is not advantageous and may lead to accumulated errors. In this paper, we present a simple and efficient method of computing the volume and surface area given data $z(x_i, y_j)$ where $z(x_i, y_j)$ denote a sampled point of a surface $z(x, y)$ at (x_i, y_j). In order to compute the volume and the surface area, our approach approximates each surface patch (Figure 1 (b)) defined by $z(x_i, y_j)$, $z(x_{i+1}, y_j)$, $z(x_i, y_{j+1})$, and $z(x_{i+1}, y_{j+1})$ on a rectangular grid by the least-square-error fit plane (Figure 1 (c)) obtained from these four points. The four points $z(x_i, y_j)$, $z(x_{i+1}, y_j)$, $z(x_i, y_{j+1})$, and $z(x_{i+1}, y_{j+1})$ are not necessarily coplanar. In the results section, four methods of computing the volume between two surfaces and the surface area are compared. Let us denote these four computing methods as follows.

- LSE-PLANE (Figure 1 (c)): computes the volume in Figure 1 (b) by approximating the surface patch using the least-square-error fit plane obtained from four surface points $z(x_i, y_j)$, $z(x_{i+1}, y_j)$, $z(x_i, y_{j+1})$, and $z(x_{i+1}, y_{j+1})$. Repeat for the second surface and sum the difference of volume over the image domain.

- TWO-TRIANGLES (Figure 1 (d)): computes the volume in Figure 1 (c) by approximating the surface patch using two triangles. Repeat for the second surface and sum the difference of volume over the image domain.

- LSE-PLANE-I: same as LSE-PLANE except that, where two surface patches intersect, the volume between them is computed by decomposing the volume between them into tetrahedra.

- TWO-TRIANGLES-I: same as TWO-TRIANGLES except that, where two surface patches intersect, the volume between them is computed by decomposing the volume between them into tetrahedra.

LSE-PLANE-I and TWO-TRIANGLES-I use the same approximations of a surface patch as in LSE-PLANE-I and TWO-TRIANGLES respectively. However, in the regions where two surface patches intersect, they compute the volume between two surface patches in the same way by decomposing it into tetrahedra. When the surface area is computed, only LSE-PLANE and TWO-TRIANGLES are compared because LSE-PLANE-I and TWO-TRIANGLES-I are

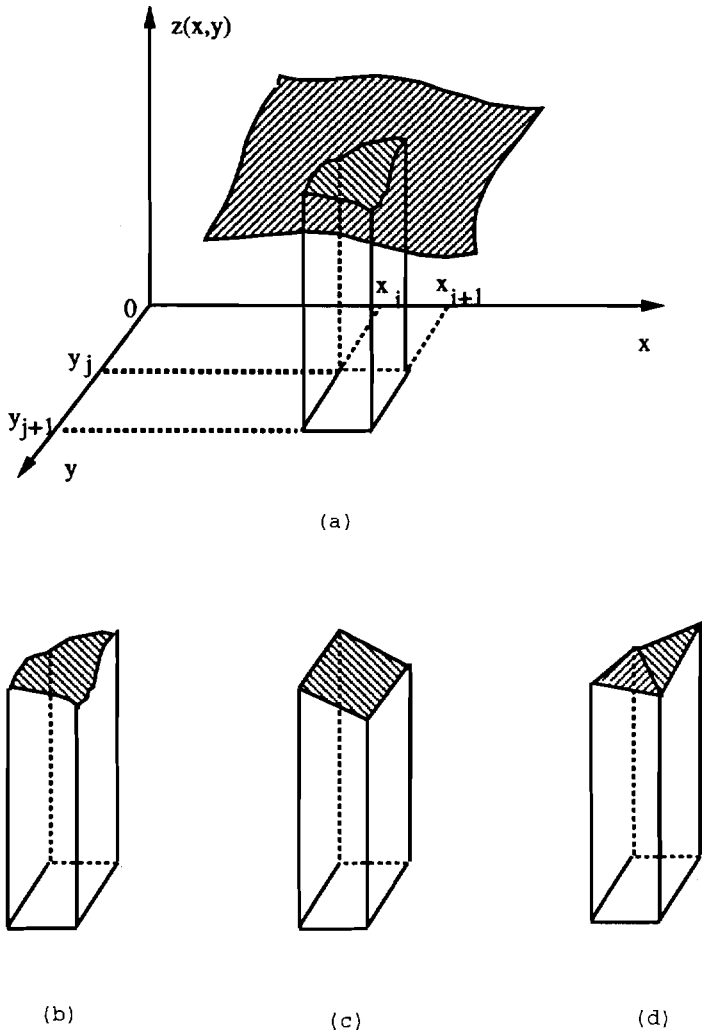


Figure 1: (a) **A** reconstructed surface, $h_x = x_{i+1} - x$; and $h_y = y_{j+1} - y_j$, (b) the volume under the surface patch marked in **(a),** the volume displayed in (b) can be estimated by approximating the surface patch in (b) using (c) the least-square-error fit plane computed from $z(x_i, y_j)$, $z(x_{i+1}, y_j)$, $z(x_i, y_{j+1})$, and $z(x_{i+1}, y_{j+1})$, or (d) two triangles.

not different from LSE-PLANE and TWO-TRIANGLES respectively in computing the surface area. It will be shown in section 7 that LSE-PLANE gives equivalent performance as TWO-TRIANGLES, LSE-PLANE-I, and TWO-TRIANGLES-I even if two surfaces intersect. We recommend LSE-PLANE for computing the volume between two surfaces and the surface area because of its computational simplicity, efficiency over other methods and good accuracy. As expected, our experimental results also verify that the computational advantage becomes greater as the problem size gets large. Knowing that a reconstructed surface is an approximated surface and the surface shape is ambiguous in regions between pixels, LSE-PLANE gives good estimates of the volume between two reconstructed surfaces and the area of a surface. In addition, the area between two curves normalized by the arc length is proposed as an invariant measure for comparison of plane curve reconstruction results. The measure uses linear approximation of curve segments. This is described in the Appendix.

The paper is organized as follows. In the following section, we briefly describe some mathematical lemmas which will be used in later sections. In section 3, we present how the volume between two surfaces and the surface area are computed using the LSE-PLANE method. Section 4 describes the TWO-TRIANGLE method. In section 5, we present how the LSE-PLANE-I and TWO-TRIANGLES-I algorithms compute the volume between two surface patches in regions where two surface patches intersect. The computational cost for the four methods are analyzed in section 6. The experiment results of four computing methods are reported in section 7.

2 Mathematical preliminaries

In this section, simple mathematical facts which will be used in section **3,** 4, and 5 are described.

Lemma 1 : Let $z(x,y) = ax + by + c$ be the plane defined by three points $P_1(0,0,z_{i,j})$, $P_2(h,,\ 0, z_{i,j+1})$, and $P_3(0, h,,\ z_{i+1,j})$ in the rectangular coordinate system. See Figure **2** (a). Then the plane equation is $z(x,y) = \frac{z_{i,j+1}-z_{i,j}}{h_x}x + \frac{z_{i+1,j}-z_{i,j}}{h_y}y + z_{i,j}$ and the volume, V, of the prism defined by P_1, P_2, P_3 and $(0,0,0)$, $(h,,\ 0,0)$, $(0,h,,\ 0)$, is computed as $\frac{h_x h_y(z_{i,j}+z_{i,j+1}+z_{i+1,j})}{6}$

Proof :

Given three points $P_1(0,0,z_{i,j})$, $P_2(h_x,0,z_{i,j+1})$, $P_3(0,h,,\ z_{i+1,j})$ and the plane equation $z(x,y) = ax + by + c$, we get

$$\begin{bmatrix} 0 & 0 & 1 \\ h_x & 0 & 1 \\ 0 & h_y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} z_{i,j} \\ z_{i,j+1} \\ z_{i+1,j} \end{bmatrix}$$

Thus

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -\frac{1}{h_x} & \frac{1}{h_x} & 0 \\ -\frac{1}{h_y} & 0 & \frac{1}{h_y} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{i,j} \\ z_{i,j+1} \\ z_{i+1,j} \end{bmatrix} = \begin{bmatrix} \frac{z_{i,j+1}-z_{i,j}}{h_x} \\ \frac{z_{i+1,j}-z_{i,j}}{h_y} \\ z_{i,j} \end{bmatrix}$$

And the volume V is,

$$\begin{aligned} V &= \int_0^{h_x}\int_0^{-\frac{h_y}{h_x}x+h_y} z(x,y)dxdy \\ &= \int_0^{h_x}\int_0^{-\frac{h_y}{h_x}x+h_y} \left(\frac{z_{i,j+1}-z_{i,j}}{h_x}x + \frac{z_{i+1,j}-z_{i,j}}{h_y}y + z_{i,j}\right)dxdy \\ &= \frac{h_x h_y(z_{i,j}+z_{i,j+1}+z_{i+1,j})}{6} \end{aligned} \quad (1)$$

Lemma 2 : Let four points $P_1(0,0,z_{i,j})$, $P_2(h_x,0,z_{i,j+1})$, $P_3(0,h_y,z_{i+1,j})$, and $P_4(h_x,h_y,z_{i+1,j+1})$ *be* coplanar in the rectangular coordinate system. See Figure *2 (b)*. Then the volume, V, defined by these four points and $(0,0,0)$, $(h_x,0,0)$, $(0,h_y,0)$, $(h,,\ h_y,0)$ is computed as, $V = \frac{1}{4}h_x h_y(z_{i,j}+z_{i,j+1}+z_{i+1,j}+z_{i+1,j+1})$.

Proof :

Let $z(x, y) = ax + by + c$ be the equation of the plane passing through four coplanar points, P_1, P_2, P_3, and P_4. Then $z_{i,j} = c$, $z_{i,j+1} = ah + c$, $z_{i+1,j} = bh_y + c$, and $z_{i+1,j+1} = ah + bh_y + c$. The volume, V is :

$$\begin{aligned} V &= \int_0^{h_x} \int_0^{h_y} z(x,y)dxdy \\ &= \frac{1}{2}h_x h_y(ah_x + bh_y + 2c) \\ &= h_x h_y \frac{(ah_x + c) + (bh_y + c) + (ah + bh_y + c) + c}{4} \\ &= \frac{1}{4}h_x h_y(z_{i,j} + z_{i,j+1} + z_{i+1,j} + z_{i+1,j+1}) \end{aligned} \tag{2}$$

Volume of tetrahedron: The volume, **V**, of a tetrahedron defined by four points $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_3(x_3, y_3, z_3)$, and $P_4(x_4, y_4, z_4)$ (Figure 3) is computed as

$$\begin{aligned} V &= \frac{1}{6}\left|\text{Determinant}\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{bmatrix}\right| \\ &- \frac{1}{6}|-x_3y_2z_1 + x_4y_2z_1 + x_2y_3z_1 - x_4y_3z_1 - x_2y_4z_1 + x_3y_4z_1 + x_3y_1z_2 \\ &\quad -x_4y_1z_2 - x_1y_3z_2 + x_4y_3z_2 + x_1y_4z_2| \end{aligned} \tag{3}$$

The volume of a tetrahedron can be viewed as $\frac{1}{3}\cdot$ (area of one of four faces as a base) $\cdot$ (perpendicular height of the fourth point to this base). This results in the same expression as the equation (3) which is just a reformulation by elementary vector analysis. A tetrahedron is a 3-D simplex. See [11] for a volume computing formula and its proof for the general n-D simplex. Based on these simple mathematical facts, we present in the next section how the volume between two surfaces and the surface area can be efficiently computed using the least-square-error fit plane (LSE-PLANE) approximation of a surface patch.

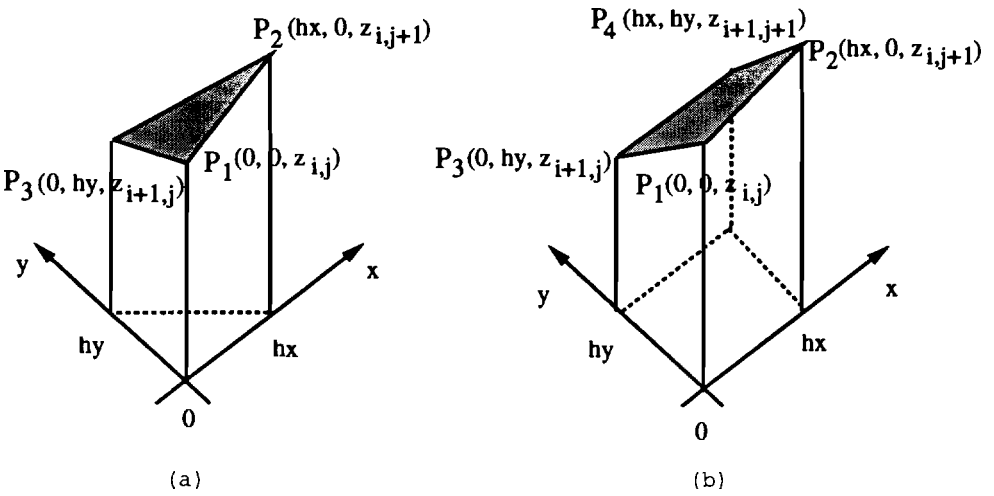


Figure 2: (a) Lemma 1 (b) Lemma 2

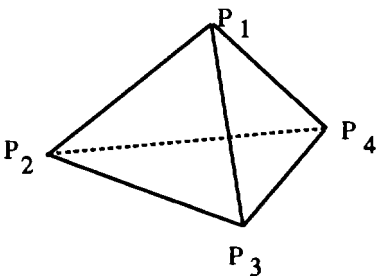


Figure 3: tetrahedron $P_1P_2P_3P_4$

3 Volume by least-square-error fit plane approximation

In order to compute the volume and the surface area, LSE-PLANE method approximates each surface patch defined by $z(x_i, y_j)$, $z(x_{i+1}, y_j)$, $z(x_i, y_{j+1})$, and $z(x_{i+1}, y_{j+1})$ on a rectangular grid by the least-square-fit plane (Figure 1 (c)) computed from these four points. Recall that the four points $z(x_i, y_j)$, $z(x_{i+1}, y_j)$, $z(x_i, y_{j+1})$, and $z(x_{i+1}, y_{j+1})$ are not necessarily coplanar. Let us denote a rectangular domain defined by (x;, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}), and (x_{i+1}, y_{j+1}) by i j th grid domain. We use the following lemma with the lemma 1 and 2 in section 2.

Lemma 3 : Let $z(x, \mathrm{y}) = \mathrm{ax} + \mathrm{by} + c$ be the least-square-error fit plane computed from four points, $P_1(0,0,z_{i,j})$, $P_2(h_x,0,z_{i,j+1})$, $P_3(0,h_y,z_{i+1,j})$, and $P_4(h_x,h_y,z_{i+1,j+1})$. See Figure 4. The volume, V, under the plane $z(x, \mathrm{y})$ on the rectangular domain [h,, 0]x$[0, h_y]$ is computed as $\mathrm{V} = \frac{1}{4}h_x h_y(z_{i,j} + z_{i,j+1} + z_{i+1,j} + z_{i+1,j+1})$. This is the volume of **a** rectangular box with the height $\frac{1}{4}(z_{i,j} + z_{i,j+1} + z_{i+1,j} + z_{i+1,j+1})$ in the same domain.

Proof :

Given four points, P_1, P_2, P_3, and P_4 and a plane equation, $z(x, \mathrm{y}) = \mathrm{ax} + by + c$, we get a system equation,

$$A\vec{u} = \vec{b}$$

where

$$A = \begin{bmatrix} 0 & 0 & 1 \\ h_x & 0 & 1 \\ 0 & \mathrm{h}, & 1 \\ h_x & h_y & 1 \end{bmatrix}, \quad \vec{u} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} z_{i,j} \\ z_{i,j+1} \\ z_{i+1,j} \\ z_{i+1,j+1} \end{bmatrix}.$$

Then the least-square-error solution is

$$\vec{u} = (A^t A)^{-1} A^t b = \begin{bmatrix} \frac{1}{2h_x}(-z_{i,j} + z_{i,j+1} - z_{i+1,j} + z_{i+1,j+1}) \\ \frac{1}{2h_y}(-z_{i,j} - z_{i,j+1} + z_{i+1,j} + z_{i+1,j+1}) \\ \frac{1}{4}(3z_{i,j} + z_{i,j+1} + z_{i+1,j} - z_{i+1,j+1}) \end{bmatrix}$$

Then volume of V is computed by **Lemma** 2,

$$\begin{aligned} V &= \frac{z(0,0) + z(h_x,0) + z(0,h_y) + z(h_x,h_y)}{4} h_x h_y \\ &= \frac{r + (h_x a + r) + (h_y b + c) + (h_x a + h_y b + c)}{4} h_x h_y \end{aligned}$$

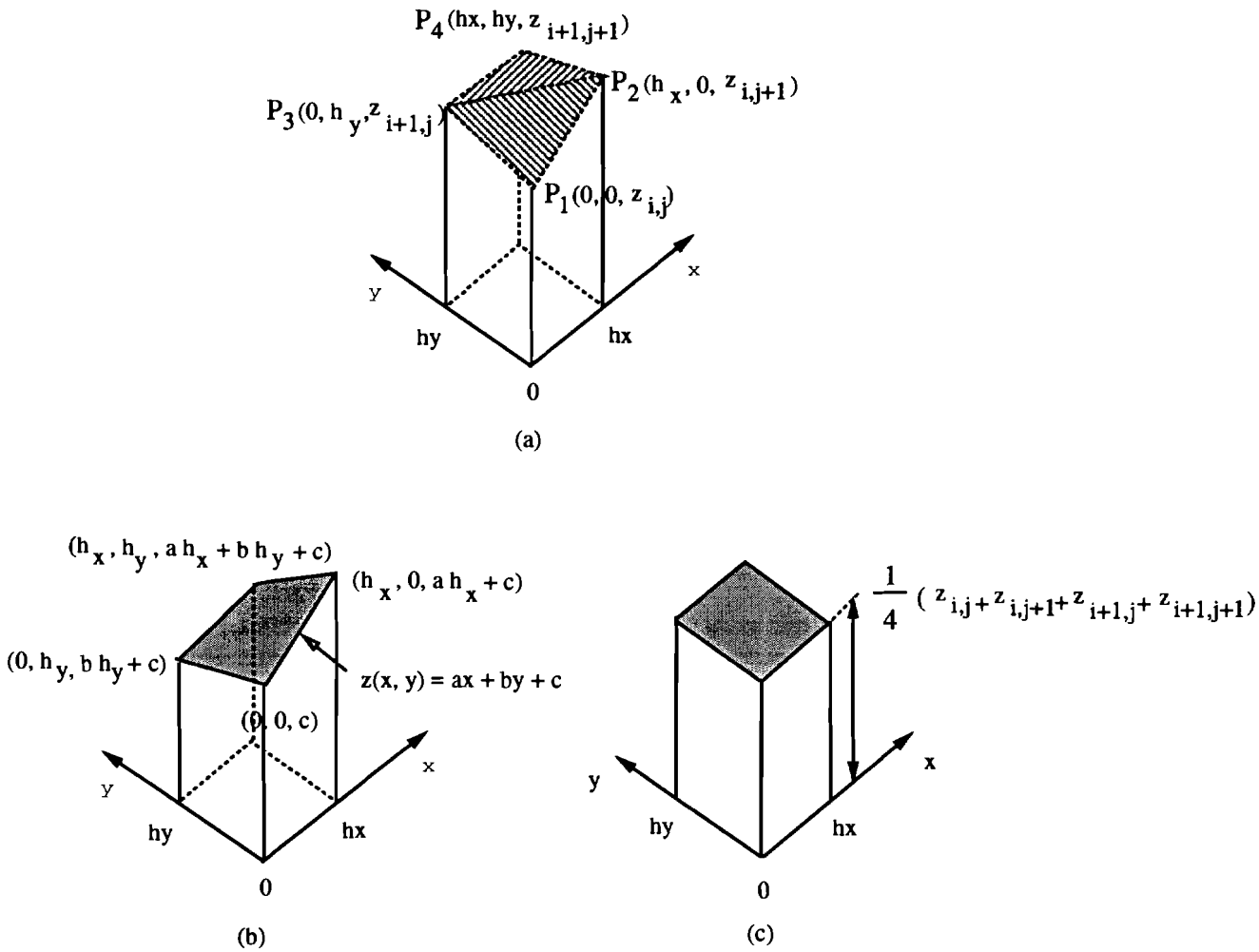


Figure 4: (a) a surface patch in which four vertices P_1, P_2, P_3, and P_4 are not necessarily coplanar, (b) its least-square-error fit plane computed from the four points in (a), (c) shows an equivalent volume of (b).

$$= \frac{1}{2}h_x h_y(ah_x + bh_y + 2c)$$
$$= \frac{1}{4}h_x h_y(z_{i,j} + z_{i,j+1} + z_{i+1,j} + z_{i+1,j+1}) \quad (4)$$

The volume between two surfaces on the ij th domain, $\Delta V_{i,j}$, is simply computed as the absolute difference of volumes under two surfaces on the same domain.

$$\Delta V_{i,j} = \frac{h_x h_y}{4}\left|z^1_{i,j} + z^1_{i+1,j} + z^1_{i,j+1} + z^1_{i+1,j+1} - \left(z^2_{i,j} + z^2_{i+1,j} + z^2_{i,j+1} + z^2_{i+1,j+1}\right)\right| \quad (5)$$

where $z^k_{i,j}$, k = 1,2 represents two surfaces. Therefore the volume, V, between two surfaces is obtained by summing $\Delta V_{i,j}$ over the image domain. Assuming a (n,m) rectangular grid,

$$V = \frac{h_x h_y}{4}\sum_{i=1}^{n-1}\sum_{j=1}^{m-1}\left|z^1_{i,j} + z^1_{i+1,j} + z^1_{i,j+1} + z^1_{i+1,j+1} - \left(z^2_{i,j} + z^2_{i+1,j} + z^2_{i,j+1} + z^2_{i+1,j+1}\right)\right| \quad (6)$$

Note that this equation does not explicitly take into account regions where two surface patches intersect while LSE-PLANE-I does. The surface area, A is calculated as

$$\begin{aligned} A &= \sum_{i=1}^{n-1}\sum_{j=1}^{m-1}\Delta A_{i,j} \\ &= h_x h_y \sum_{i=1}^{n-1}\sum_{j=1}^{m-1}\sec\phi_{i,j} \\ &= h_x h_y \sum_{i=1}^{n-1}\sum_{j=1}^{m-1}\sqrt{E_{i,j}G_{i,j} - F^2_{i,j}} \\ &= h_x h_y \sum_{i=1}^{n-1}\sum_{j=1}^{m-1}\sqrt{1 + a^2_{i,j} + b^2_{i,j}} \\ &= \frac{1}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{m-1}\sqrt{4h^2_x h^2_y + h^2_y(-z_{i,j} + z_{i+1,j} - z_{i,j+1} + z_{i+1,j+1})^2} \\ &\qquad \overline{+h^2_x(-z_{i,j} - z_{i+1,j} + z_{i,j+1} + z_{i+1,j+1})^2} \end{aligned} \quad (7)$$

where $\Delta A_{i,j}$ denotes the surface area on the rectangular domain [h,, 0]x[0, h_y] and $z_{i,j}(x,y) = a_{i,j}x + b_{i,j}y + c_{i,j}$ is the least-square-error fit plane computed from the four points $z_{i,j}$, $z_{i+1,j}$, $z_{i,j+1}$, and $z_{i+1,j+1}$. $\phi_{i,j}$ represents slant angle of $z_{i,j}(x,\mathrm{y})$ and $E_{i,j}$, $F_{i,j}$, and $G_{i,j}$ are the first fundamental forms of $z_{i,j}(x,\mathrm{y})$. Hence, the volume between a known surface and its reconstructed surface normalized by the known surface area becomes V/A.

The following section discusses how the TWO-TRIANGLES method computes volume and surface area.

4 Volume by two-triangle approximation

TWO-TRIANGLES method approximates each surface patch by two triangles (Figure 1 (d)) instead of the least-square-error fit plane described in the previous section. Each surface patch is approximated by two triangles in a consistent direction over an entire image. We triangulate each surface patch so that one triangle is defined by three points $z_{i,j}$, $z_{i+1,j}$, $z_{i,j+1}$ (we will call this ***left*** triangle) and the other triangle by three points $z_{i+1,j}$, $z_{i,j+1}$, $z_{i+1,j+1}$ (we will call this ***right*** triangle) as shown Figure 5. However, two triangles can be defined in the other consistent direction over the entire image, i.e, one triangle by $z_{i,j}$, $z_{i+1,j}$, $z_{i+1,j+1}$ and the other by $z_{i,j}$, $z_{i,j+1}$, $z_{i+1,j+1}$. Without loss of generality, we describe only for the triangulation shown in Figure 5. The volume of a prism defined on a triangle domain is obtained using **Lemma 1.** The TWO-TRIANGLES method computes the volume between two surfaces on the ***left*** triangle domain of the ***ij*** th gird as follows.

$$\Delta V^L_{i,j} = \frac{1}{6} h_x h_y \left| z^1_{i,j} + z^1_{i+1,j} + z^1_{i,j+1} - (z^2_{i,j} + z^2_{i+1,j} + z^2_{i,j+1}) \right| \tag{8}$$

where $z^k_{i,j}$, k = 1,2 represents two surfaces. This equation does not explicitly take into account regions where two surface patches intersect while the TWO-TRIANGLES-I algorithm does. The volume bewteen two surfaces, V, is obtained as follows assuming a (n, m) retangular grid.

$$\begin{aligned} \mathrm{V} &= \textstyle\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left(\Delta V^L_{i,j} + \Delta V^R_{i,j} \right) \\ &= \tfrac{1}{6} h_x h_y \textstyle\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \Big(\left| z^1_{i,j} + z^1_{i+1,j} + z^1_{i,j+1} - (z^2_{i,j} + z^2_{i+1,j} + z^2_{i,j+1}) \right| \\ &\quad + \left| z^1_{i+1,j} + z^1_{i,j+1} + z^1_{i+1,j+1} - (z^2_{i+1,j} + z^2_{i,j+1} + z^2_{i+1,j+1}) \right| \Big) \end{aligned} \tag{9}$$

where $\Delta V^L_{i,j}$ an $\Delta V^R_{i,j}$ denote the volume between two surfaces on the ***left*** and ***right*** triangle domain, respectively. The surface area, A is calculated as sum of areas of all ***left*** and ***right*** triangles over the entire surface. If we let the plane equations of the ***left*** and ***right*** triangles on the ***ij*** the grid be $z^1_{i,j}(x, \mathrm{y}) = a^1_{i,j} x + b^1_{i,j} y + c^1_{i,j}$ and $z^2_{i,j}(x, \mathrm{y}) = a^2_{i,j} x + b^2_{i,j} y + c^2_{i,j}$ respectively,

$$\begin{aligned} A &= \frac{1}{2} h_x h_y \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} (\sec \phi^1_{i,j} + \sec \phi^2_{i,j}) \\ &= \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \frac{1}{2} h_x h_y (\sqrt{E^1_{i,j} G^1_{i,j} - (F^1_{i,j})^2} + \sqrt{E^2_{i,j} G^2_{i,j} - (F^2_{i,j})^2}) \\ &= \frac{1}{2} h_x h_y \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} (\sqrt{1 + (a^1_{i,j})^2 + (b^1_{i,j})^2} + \sqrt{1 + (a^2_{i,j})^2 + (b^2_{i,j})^2}) \\ &= \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left(\sqrt{h_x^2 h_y^2 + h_x^2 (z_{i+1,j} - z_{i,j})^2 + h_y^2 (z_{i,j+1} - z_{i,j})^2} \right. \end{aligned}$$

$$+\sqrt{h_x^2 h_y^2 + h_x^2(z_{i,j+1} - z_{i,j})^2 + h_y^2(z_{i+1,j} - z_{i,j})^2}\Big) \tag{10}$$

where $E^1_{i,j}$, $F^1_{i,j}$, $G^1_{i,j}$ and $E^2_{i,j}$, $F^2_{i,j}$, $G^2_{i,j}$ are the first fundamental forms of $z^1_{i,j}(x,y)$ and $z^2_{i,j}(x,y)$ respectively.

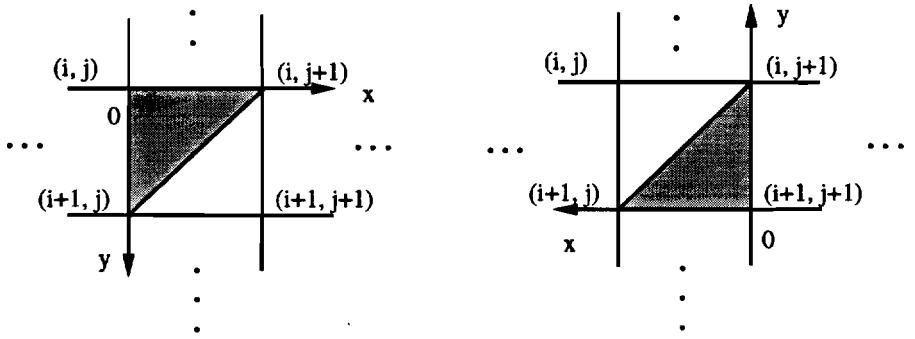


Figure 5: (a) left triangle domain (b) **right** triangle domain

5 Volume between two surfaces where intersection occurs

As mentioned in section 1, LSE-PLANE-I and TWO-TRIANGLES-I use the same approximations of a surface patch as in LSE-PLANE and TWO-TRIANGLES respectively. However, when LSE-PLANE-I or TWO-TRIANGLES-I are used to compute the volurne between two surfaces, regions where two surface patches intersect are treated in a different manner than the non-intersecting regions. This section describes how the volume between two surface patches is computed by decomposing the volume into tetrahedra in these regions.

For each grid domain, we test whether two surfaces intersect each other. We define an intersection test as follows. On the i j th grid domain,

$$(z^1_{i,j} \geq z^2_{i,j}, z^1_{i,j+1} \geq z^2_{i,j+1}, z^1_{i+1,j} \geq z^2_{i+1,j}, z^1_{i+1,j+1} \geq z^2_{i+1,j+1})$$
$$\text{OR}$$
$$(z^1_{i,j} \leq z^2_{i,j}, z^1_{i,j+1} \leq z^2_{i,j+1}, z^1_{i+1,j} \leq z^2_{i+1,j}, z^1_{i+1,j+1} \leq z^2_{i+1,j+1})$$

then the two surfaces do not intersect on the current domain.
Otherwise, the two surfaces are said to intersect on the current domain. $z^k_{i,j}$, k = 1,2 represents two surfaces.

When two surfaces are known to intersect on a grid domain according to this test, special consideration can be given to compute the volume between two surface patches on this grid domain. Each surface patch is approximated by two triangles as in the previous section and the left and ***right*** triangle domain are considered separately. If all three vertices of one triangle (surface) are above or below three vertices of the other triangle (surface) on a triangle domain, two surfaces are said not to intersect on this triangle domain. The intersection occurs in the other triangle domain. In this triangle domain, the volume between two surface patches is decomposed into tetrahedra. We have to consider three cases on each left and ***right*** triangle domain when the above intersection test is true.

- case 1 : Two triangles do not intersect.
- case 2 : Two triangles intersect with one vertex shared. See Figure 6 (a) (b) (c).
- case 3 : Two triangles intersect without sharing vertices. See Figure 6 (d) (e) (f)

Without loss of generality, we describe only for the left triangle domain (Figure 6). In case 1, the volume between two prisms can be simply computed as the absolute value of the difference of each prism's volume as in TWO-TRIANGLES method (equation (8)). In case 2, there are three subcases (a), (b), and (c) depending on the location of the shared vertex. Each subcase is determined by the location of the shared vertex P_3. The point P_4 is easily computed as the intersection point of two straight lines. In each subcase, the volume between two prisms is computed as sum of two tetrahedra. Case 3 also has three subcases depending on the location of the points Q_1 and Q_2. Three subcases are identified by the signs of $z^1_{i,j} - z^2_{i,j}$, $z^1_{i,j+1} - z^2_{i,j+1}$, and $z^1_{i+1,j} - z^2_{i+1,j}$ where $z^k_{i,j}$, k = 1,2 denotes two triangles (or surfaces). The points P_3 and P_4 are computed as intersection points of two straight lines. These three subcases produce the volume between two prisms as sum of one tetrahedron and one irregular prism which can be always decomposed into three tetrahedra. Figure 7 (a) shows a prism, $P_1P_2P_3P_4P_5P_6$ resulting from the first subcase of case 3 (Figure 6 (d)). Figure 7 (b) is one example of possible decompositions of the prism $P_1P_2P_3P_4P_5P_6$ into three tetrahedra. The volume of the prism, V, shown in Figure 7 (a) is computed as:

$$V = tetrahedron(P_1P_2P_3P_4) + tetrahedron(P_1P_2P_4P_5) + tetrahedron(P_2P_4P_5P_6) \qquad (11)$$

Thus the volume between two prisms, V, in the cases of Figure 6 (a), (b), and (c) is computed as

$$V = tetrahedron(P_1P_2P_3P_4) + tetrahedron(P_3P_4P_5P_6) \qquad (12)$$

In the cases of Figure 6 (d), (e), and (f),

$$\begin{aligned} V \;=\; & tetrahedron(P_3P_4Q_1Q_2) + tetrahedron(P_1P_2P_3P_4) \\ & + tetrahedron(P_1P_2P_4P_5) + tetrahedron(P_2P_4P_5P_6) \end{aligned} \qquad (13)$$

The points, P_3 and P_4, are computed as follows.

(a) : $P_3(0, 0, z^1_{i,j})$
$P_4\left(\frac{h_x(z^1_{i+1,j}-z^2_{i+1,j})}{z^1_{i+1,j}-z^2_{i+1,j}-z^1_{i,j+1}+z^2_{i,j+1}}, \frac{h_y(-z^1_{i,j+1}+z^2_{i,j+1})}{z^1_{i+1,j}-z^2_{i+1,j}-z^1_{i,j+1}+z^2_{i,j+1}}, \frac{-z^2_{i+1,j}z^1_{i,j+1}+z^1_{i+1,j}-z^2_{i,j+1}}{z^1_{i+1,j}-z^2_{i+1,j}-z^1_{i,j+1}+z^2_{i,j+1}}\right)$

(b) : $P_3(h_x, 0, z^1_{i,j+1})$
$P_4\left(0, \frac{h_y(-z^2_{i,j}+z^1_{i,j})}{z^2_{i+1,j}-z^1_{i+1,j}-z^2_{i,j}+z^1_{i,j}}, \frac{-z^2_{i+1,j}z^1_{i,j+1}+z^1_{i+1,j}-z^2_{i,j+1}}{z^2_{i+1,j}-z^1_{i+1,j}-z^2_{i,j}+z^1_{i,j}}, \right)$

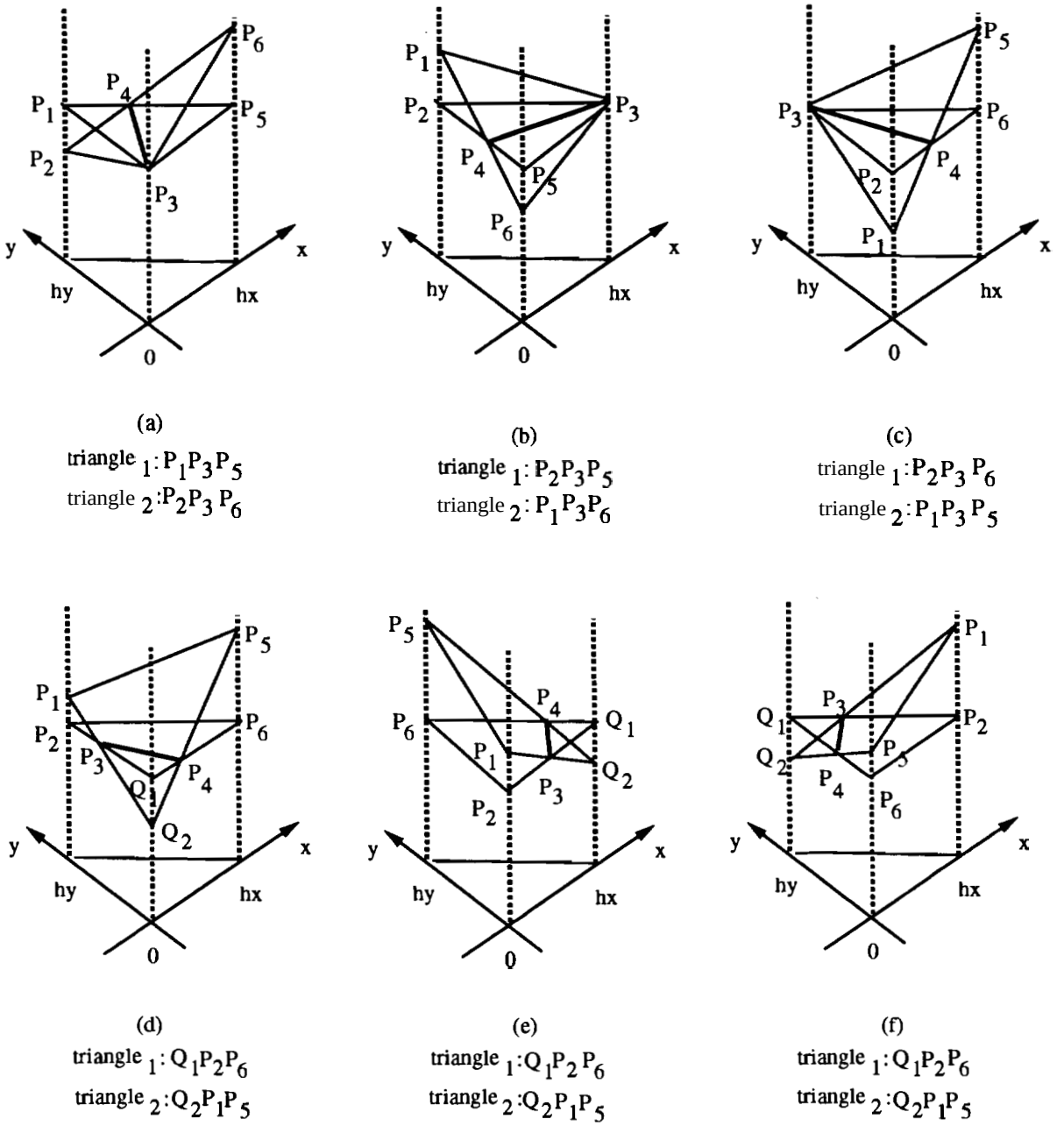
(c) : $P_3(0, h_y, z^1_{i+1,j})$
$P_4\left(\frac{h_x(z^2_{i,j}-z^1_{i,j})}{z^2_{i,j}-z^1_{i,j}-z^2_{i,j+1}+z^1_{i,j+1}}, 0, \frac{-z^2_{i+1,j}z^1_{i,j+1}+z^1_{i+1,j}-z^2_{i,j+1}}{z^2_{i,j}-z^1_{i,j}-z^2_{i,j+1}+z^1_{i,j+1}}, \right)$

(d) : $P_3(\frac{c_2-c_1}{a_1-a_2}, 0, \frac{a_1c_2-a_2c_1}{a_1-a_2})$
$P_4(0, \frac{c_2-c_1}{b_1-b_2}, \frac{b_1c_2-b_2c_1}{b_1-b_2})$

(e) : $P_3(\frac{c_2-c_1}{a_1-a_2}, 0, \frac{a_1c_2-a_2c_1}{a_1-a_2})$
$P_4\left(\frac{(c_2-c_1)h_x+(b_2-b_1)h_xh_y}{(a_1-a_2)h_x+(b_2-b_1)h_y}, \frac{(c_1-c_1)h_y+(a_1-a_2)h_xh_y}{(a_1-a_2)h_x+(b_2-b_1)h_y}, \frac{(a_1c_2-a_2c_1)h_x+(b_2c_1-b_1c_2)h_y+(a_1b_2-a_2b_1)h_xh_y}{(a_1-a_2)h_x+(b_2-b_1)h_y}\right)$

(f) : $P_3\left(\frac{(c_2-c_1)h_x+(b_2-b_1)h_xh_y}{(a_1-a_2)h_x+(b_2-b_1)h_y}, \frac{(c_1-c_2)h_y+(a_1-a_2)h_xh_y}{(a_1-a_2)h_x+(b_2-b_1)h_y}, \frac{(a_1c_2-a_2c_1)h_x+(b_2c_1-b_1c_2)h_y+(a_1b_2-a_2b_1)h_xh_y}{(a_1-a_2)h_x+(b_2-b_1)h_y}\right)$
$P_4(0, \frac{c_2-c_1}{b_1-b_2}, \frac{b_1c_2-b_2c_1}{b_1-b_2})$

where $a_k = \frac{z^k_{i,j+1}-z^k_{i,j}}{h_x}$, $b_k = \frac{z^k_{i+1}-z^k_{i,j}}{,}$, and $c_k = z^k_{i,j}$, k $= 1, 2$.



(a)
triangle $_1$: $P_1P_3P_5$
triangle $_2$: $P_2P_3P_6$

(b)
triangle $_1$: $P_2P_3P_5$
triangle $_2$: $P_1P_3P_6$

(c)
triangle $_1$: $P_2P_3P_6$
triangle $_2$: $P_1P_3P_5$

(d)
triangle $_1$: $Q_1P_2P_6$
triangle $_2$: $Q_2P_1P_5$

(e)
triangle $_1$: $Q_1P_2P_6$
triangle $_2$: $Q_2P_1P_5$

(f)
triangle $_1$: $Q_1P_2P_6$
triangle $_2$: $Q_2P_1P_5$

Figure 6: (a) (b) (c) Three subcases of case 2: one vertex is shared. (d) (e) (f) Three subcases of case 3: no vertex is shared. triangle$_1$ and triangle$_2$ are represented by $z^1(x,y) = a_1x + b_1y + c_1$ and $z^2(x,y) = a_2x + b_2y + c_2$ respectively.

6 Computational efficiency

In this section, we analyze the computational cost of the four methods of computing the volume between two surfaces and the surface area. For simplicity, it is assumed that the cost of taking the absolute value or that of one logical operation("OR" operation in the intersection test) is equivalent to that of one addition. We also assume that the constants such as h_x^2, h_y^2, $h_x^2 h_y^2$ and $4h_x^2 h_y^2$ in the equations (4), (7), (9), and (10) are computed in advance. The cost of computation is for images of size (n x m). ε represents the percentage of intersecting regions. The following is the number of operations for the four methods.

1. LSE-PLANE method
   - volume: (n – 1)(6m – 4) additions + 1 multiplication
   - area: (n – 1)(9m – 12) additions +(n – 1)(m – 1) square root operations + {4(n – 1)(m – 1) + 1) multiplication
2. TWO-TRIANGLES method
   - volume: 10(n – 1)(m – 1) additions + 1 multiplication
   - area: 7(n – 1)(m – 1) additions + 2(n – 1)(m – 1) square root operations + {8(n – 1)(m – 1) + 1) multiplications
3. LSE-PLANE-I method: In the worst case,
   - volume: (n – 1)(m – 1){(1 – ε)(8 additions + 1 multiplication) + ε(71 additions + 108 multiplications + 7 divisions))
   - area: same as in the method 1.
4. TWO-TRIANGLES-I method: In the worst case,
   - volume: (n – 1)(m – 1){(1 – ε) (10 additions + 1 multiplication) + ε(71 additions + 108 multiplications + 7 divisions) }
   - area: same as in the method 2.

In order to speed up the computation, a simple caching scheme is used. In equation (6) of the LSE-PLANE method, the partial sums $z_{i,j+1}^k + z_{i+1,j+1}^k$, k = 1,2 computed in the previous region is cached and used in the next region without recomputing them. In equation (9) of

the TWO-TRIANGLES method, the partial sums, $z^k_{i,j+1} + z^k_{i+1,j}$, k = 1,2 in the left triangle domain are cached for reuse in the right triangle domain. In computing the surface area, $-z_{i,j+1} + z_{i+1,j+1}$ and $z_{i,j+1} + z_{i+1,j+1}$ are cached in equation (7) of the LSE-PLANE method for reuse in the next region. In computing the volume using the methods LSE-PLANE-I and TWO-TRIANGLES-I, the intersection test cost 5 additions and the test results can be reused for identifying three cases described in the previous section. The worst case comes from the second and third subcases (Figure 6 (e) and (f) respectively) of case 3. These cases add the cost of identifying the points P_3 and P_4 to the cost of computing the volume of four tetrahedra. For simplicity, the cost of identifying subcases is ignored. From the above analysis, we can see that the method using the least-square-error plane approximation (LSE-PLANE method) is the most efficient in computing the volume between two surfaces normalized by the surface area. Experimental results in the following section verify this analysis.

7 Experimental results

In this section, we report on the accuracy and the computational efficiency of the four methods listed in the previous section. For an accuracy test, the volume between two surfaces and the surface area computed from the four methods are compared with exact values of known volume and surface area.

The following synthetic graph surfaces of size 128 x 128 were used to test the accuracy of the above four methods.

$$\begin{aligned} z_1(i,j) &= \begin{cases} 5.0\sin(2\pi\frac{1}{T}r) + 10.0 & \text{if } r \le 52.0 \\ 10.0 & \text{otherwise} \end{cases} \\ z_2(i,j) &= \begin{cases} -5.0\sin(2\pi\frac{1}{T}r) + 10.0 & \text{if } r \le 52.0 \\ 10.0 & \text{otherwise} \end{cases} \end{aligned} \tag{14}$$

where $r = \sqrt{(i-63)^2 + (j-63)^2}$.

Two surfaces, $z_1(i,j)$ and $z_2(i,j)$ are defined on a disc domain of which radius is 52.0 centered at (63, 63) in arrays of size 128 x 128. A surface plot of $z_1(i,j)$ is shown in Figure 8 when T = 13.0 and 26.0. As the period T increases from a small value, the percentage of intersecting regions decreases and the sampled surfaces appear smoother. A small T, a high frequency surface, represents a very rough sampling of an image. See Figure 9. These images are general in that surfaces are curved. The performance of LSE-PLANE and TWO-TRIANGLES varies as the percentage of intersecting regions changes compared to that of LSE-PLANE-I and TWO-TRIANGLES-I. Recall that LSE-PLANE and TWO-TRIANGLES do not consider intersecting regions separately. We can see in Figure 10 that the volume between two surfaces computed from the four methods quickly approaches the real volume between two surfaces as T increases. It is hard to visually distinguish between LSE-PLANE-I and TWO-TRIANGLES-I in this Figure because they provide almost same volume for this pair of images. Figure 11 and 12 show the surface area computed by LSE-PLANE and TWO-TRIANGLES and their computing errors. Figure 12 shows that LSE-PLANE certainly gives a better approximation of the image surface than TWO-TRIANGLES. The computation time shown in Figure 13 indicates that LSE-PLANE is computationally more efficient than the other three methods. An experiment was carried out to verify that LSE-PLANE is computationally more advantageous as problems get larger. Images of six different (n × n) sizes are tested where n = 32, 64, 128, 256, 512, 1024. The same form of surfaces (14) is used except that T is fixed to 10.0 and the radius of the disc domain is defined as $0.8125\frac{n}{2}$ so that two surfaces can maintain an approximately constant rate of intersecting regions for different size n. Figun: 14 shows that

the percentage of intersecting regions is maintained approximately constant for various sizes of surfaces. The actual and theoretical computation time is illustrated in Figure 15 and 16 respectively. The theoretical computation cost was computed from the analysis made in the previous section. For simplicity, the computational cost of addition, multiplication/division, and square root operations were assumed to be equal although in reality the square root operation is much more expensive than the other operations. Computation of only surface area involves the square root operations and TWO-TRIANGLES has more square root operations than LSE-PLANE. Therefore, this assumption does not change the order of computational efficiency of the four methods in Figure 16. As can be expected, computation time increases proportionally to n^2 for all four methods. The computational cost for LSE-PLANE-I and TWO-TRIANGLES-I (two upper curves in both figures) becomes much more expensive than LSE-PLANE and TWO-TRIANGLES as the problem size gets larger. The theoretical computation cost for LSE-PLANE and TWO-TRIANGLES is displayed again in Figure 17 for a clear comparison. Shapes of plots for the actual (Figure 15) and theoretical (Figures 16 and 17) computation time strongly resemble each other. The use of method LSE-PLANE is computationally more advantageous for a large image size.

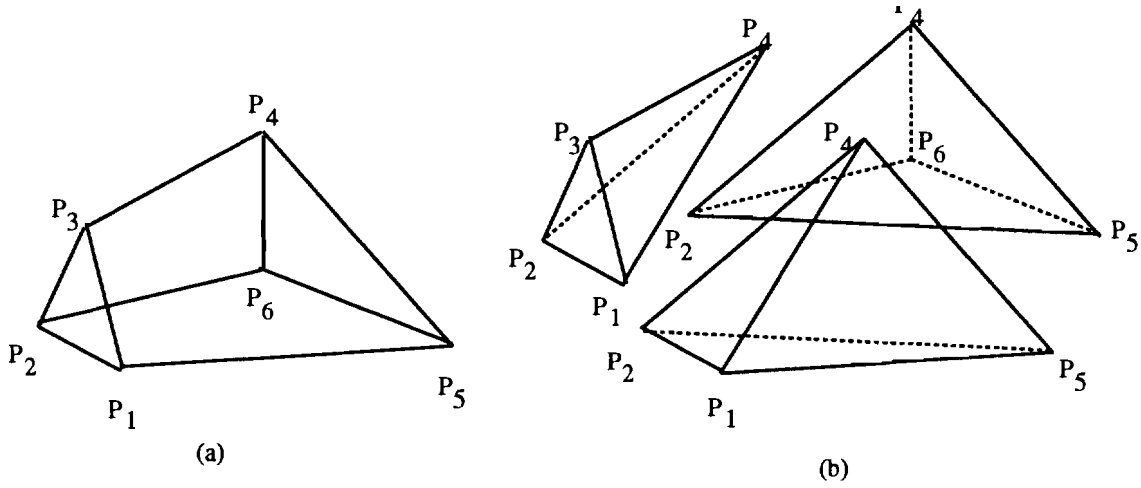
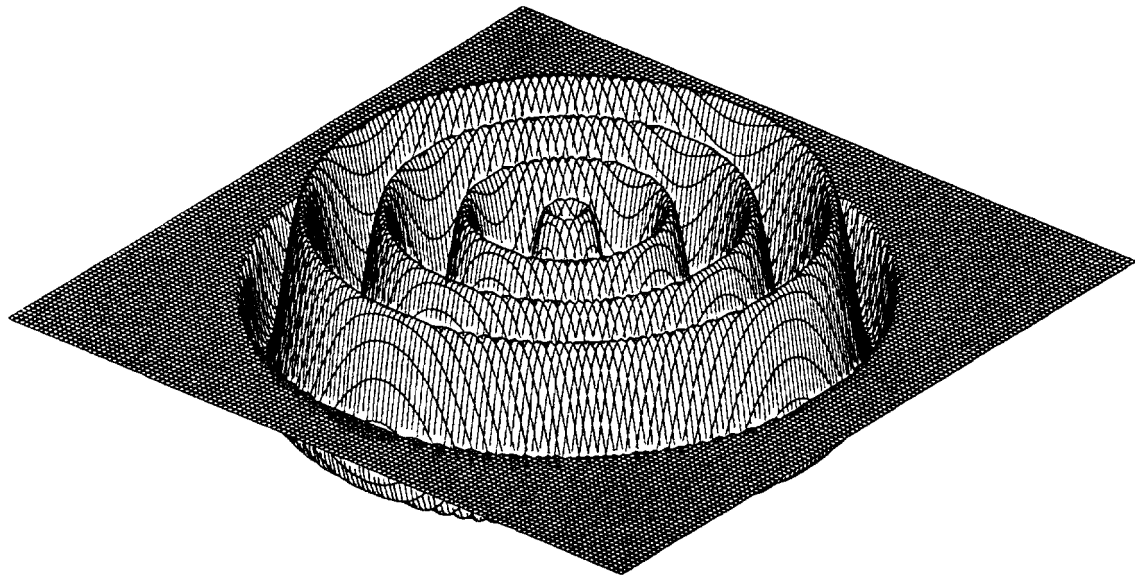
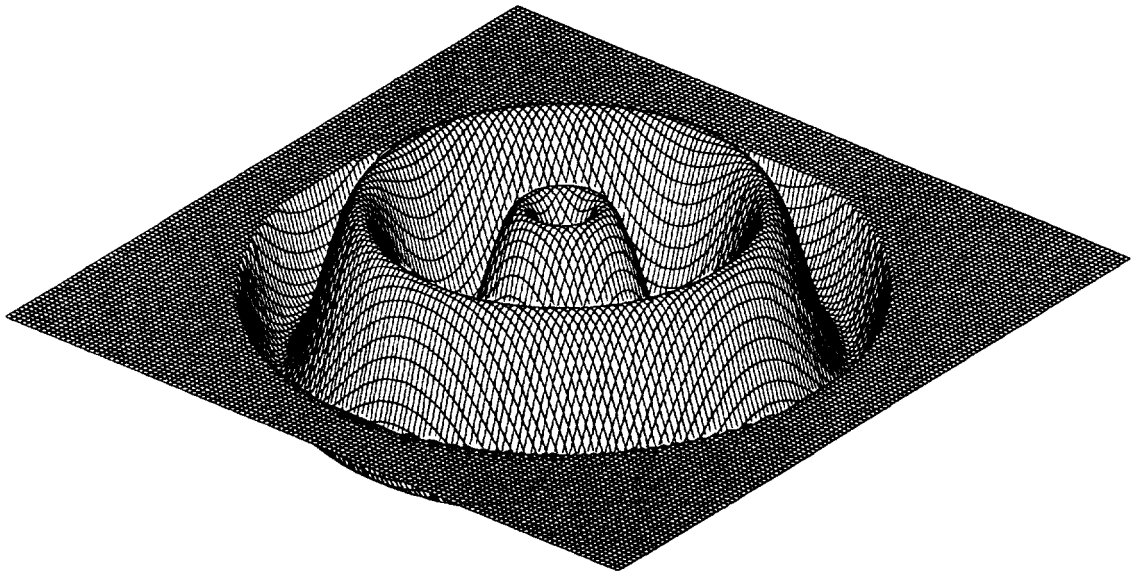


Figure 7: (a) a prism resulting from the first subcases of case 3, (b) a decomposition of (a) into three tetrahedron



(a)



(b)

Figure 8: Surface **plot** of $z_1(i, \mathbf{j})$ when (a) $\mathrm{T} = 13.0$ and (b) $\mathrm{T} = 26.0$

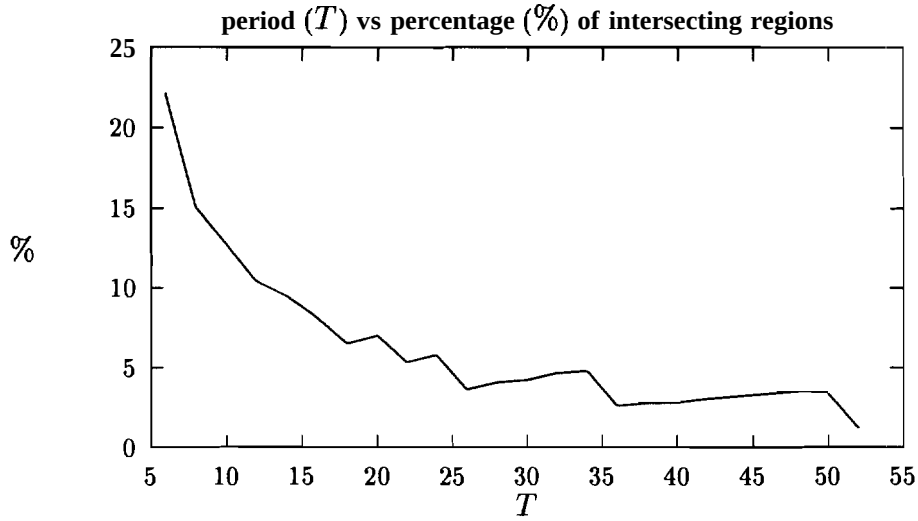


Figure 9: period (T)vs percentage of intersecting regions

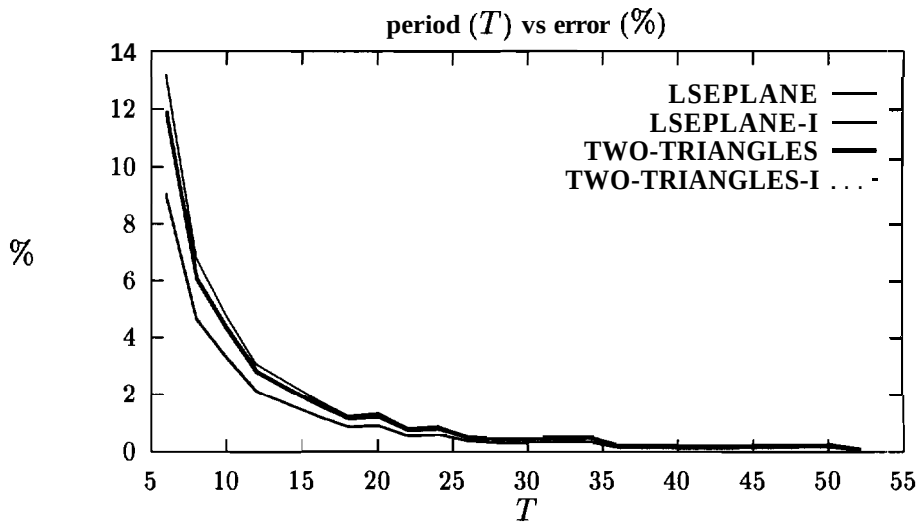


Figure 10: period (T)vs error in computing the volume between two surfaces

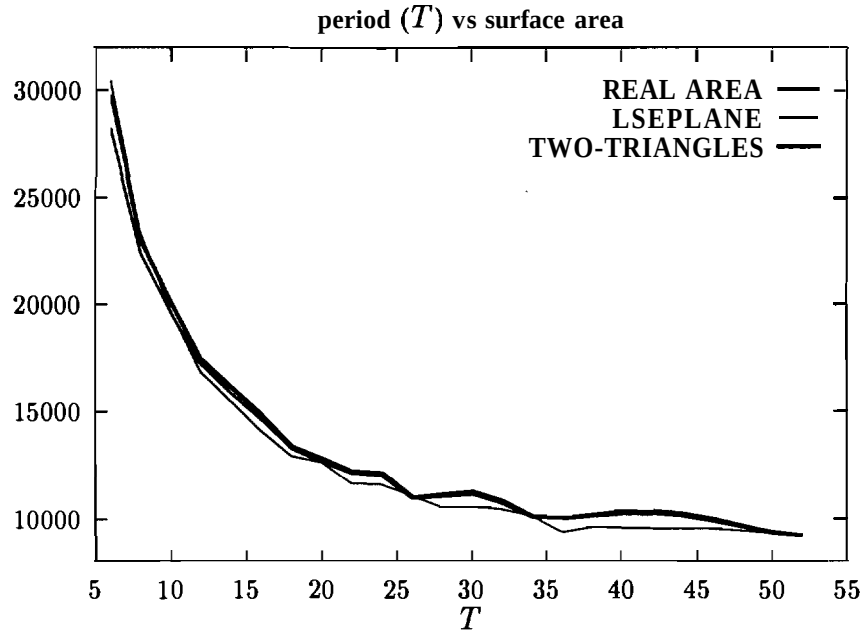


Figure 11: Comparison of real area computed by the methods LSE-PLANE and TWO-TRIANGLES

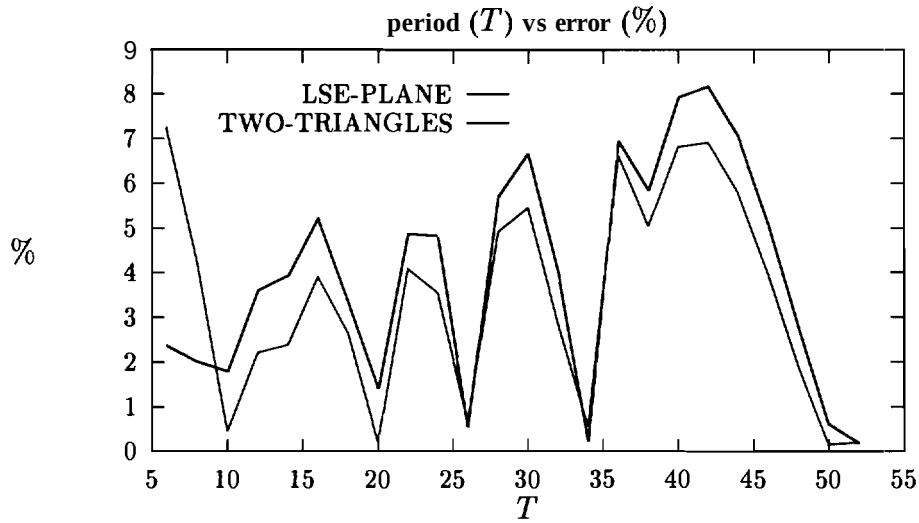


Figure 12: period (T)vs error in computing the surface area

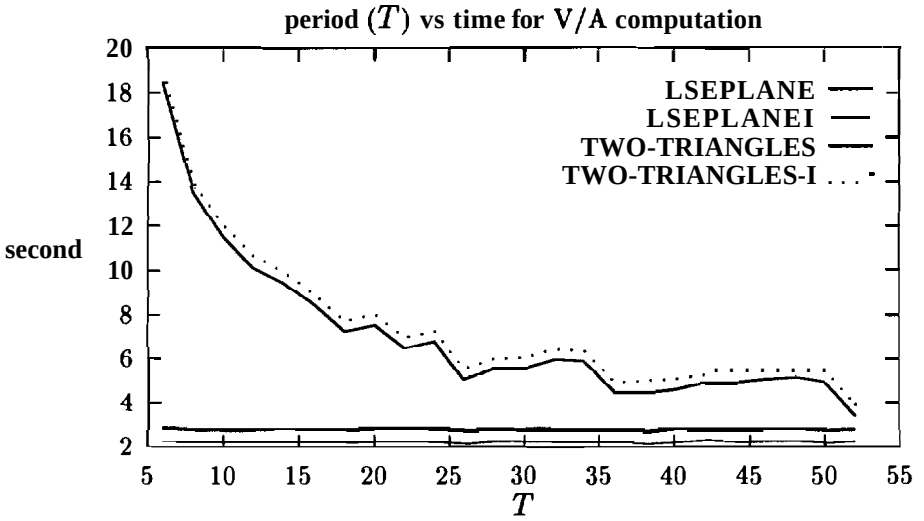


Figure 13: Computation time of four methods for two (128x128) surfaces

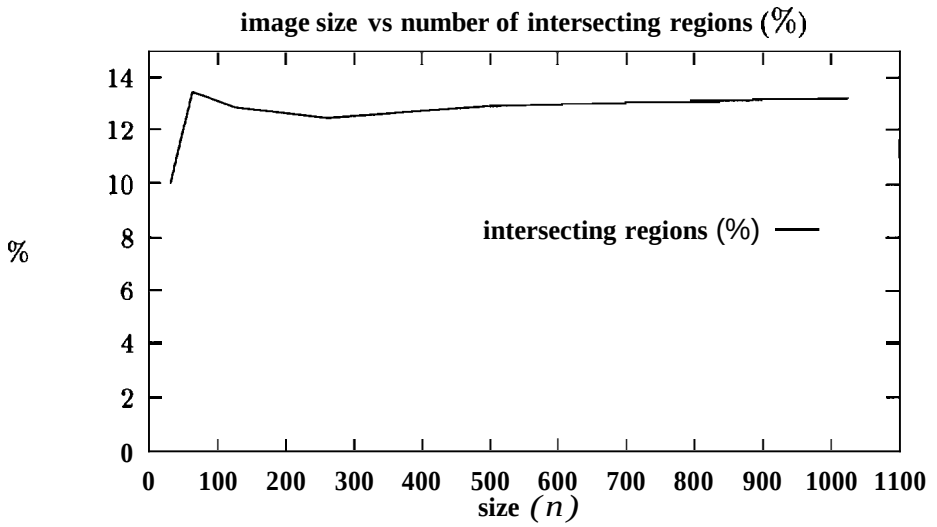


Figure 14: The percentage of intersecting regions is almost same for various sizes of surfaces.

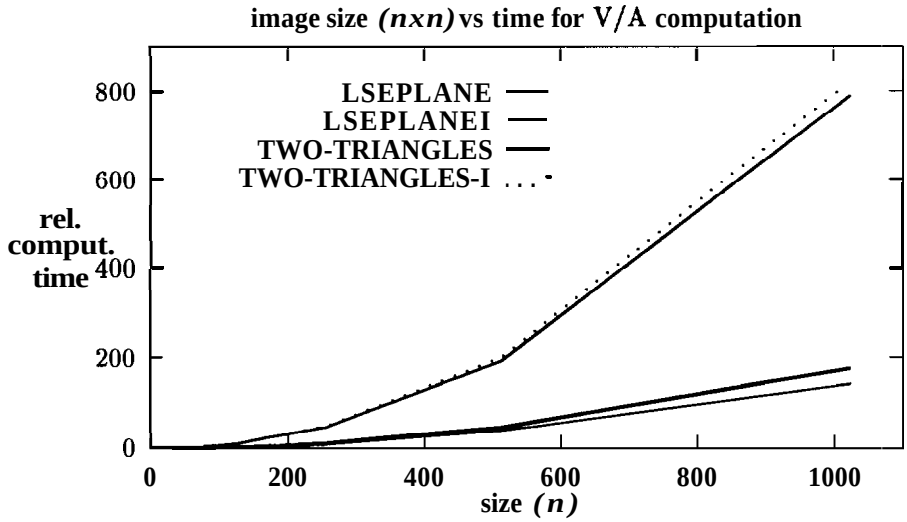


Figure 15: problem size versus relative computation times

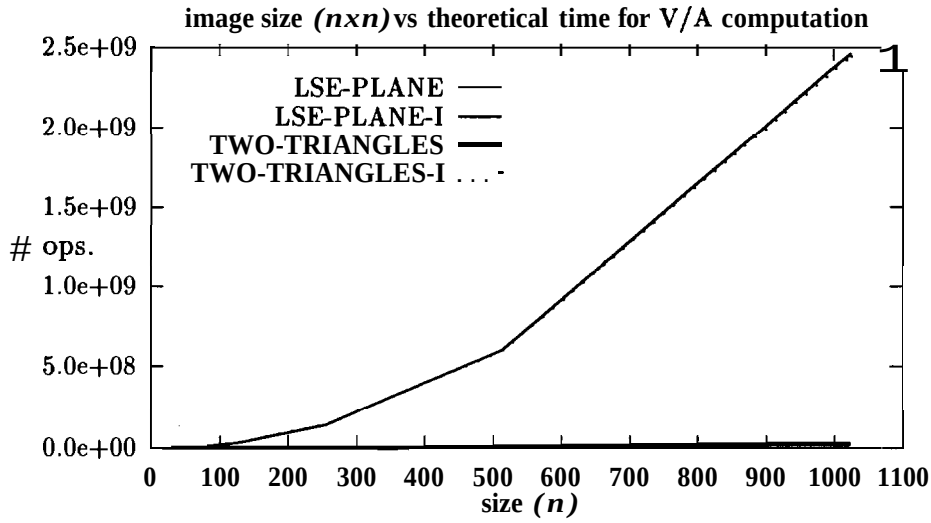


Figure 16: problem size versus relative computation times (theoretical), # ops. denotes number of operations.

We now present an example where this proposed algorithm is used for the intended application: comparison of different surface reconstruction techniques. Table **1** shows the volume between two surfaces normalized by the surface area using the four methods for reconstruction results of a sparse image reported in [24]. This ***curved-inclined*** image has three flat, two inclined (slope **1** and $\frac{1}{2}$) and two curved surfaces (curvature $\frac{1}{20}$ and $\frac{1}{30}$) and 50% of pixels of the image are randomly deleted. Figure 18 shows a section display of the original noiseless and noisy images. The four methods of computing **V/A** give almost the same performance ratio (in the last column of Table 1) for two reconstruction results although **V/A** values are slightly different for each method. The **V/A** values allow us to quantify the improved reconstruction as being three times better than the method it is compared to. **A** section display of the reconstructed surfaces from the two different methods is shown in Figure 19.

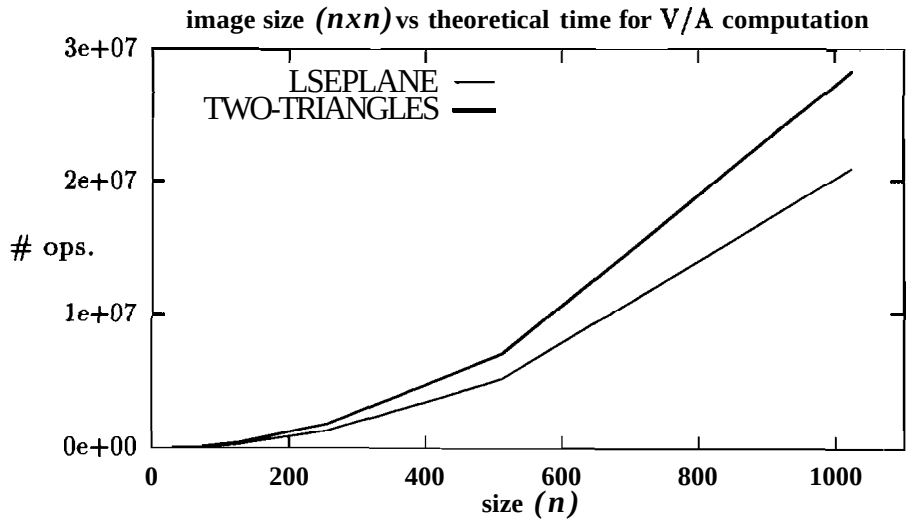


Figure **17:** problem size versus relative computation times (theoretical) for LSE-PLANE and TWO-TRIANGLES. This is a close up of lower two curves of Figure **16.** # ops. denotes number of operations.

	ordinary method	Yi & Chelberg's new method	performance ratio
LSE-PLANE	**0.6770**	**0.2176**	3.11
LSE-PLANE-I	**0.7198**	**0.2383**	**3.02**
TWO-TRIANGLES	**0.6966**	**0.2254**	**3.09**
TWO-TRIANGLES-I	**0.7198**	**0.2383**	**3.02**

Table **1:** V/A measure for the reconstruction results from ordinary method and Yi and Chelberg's new method

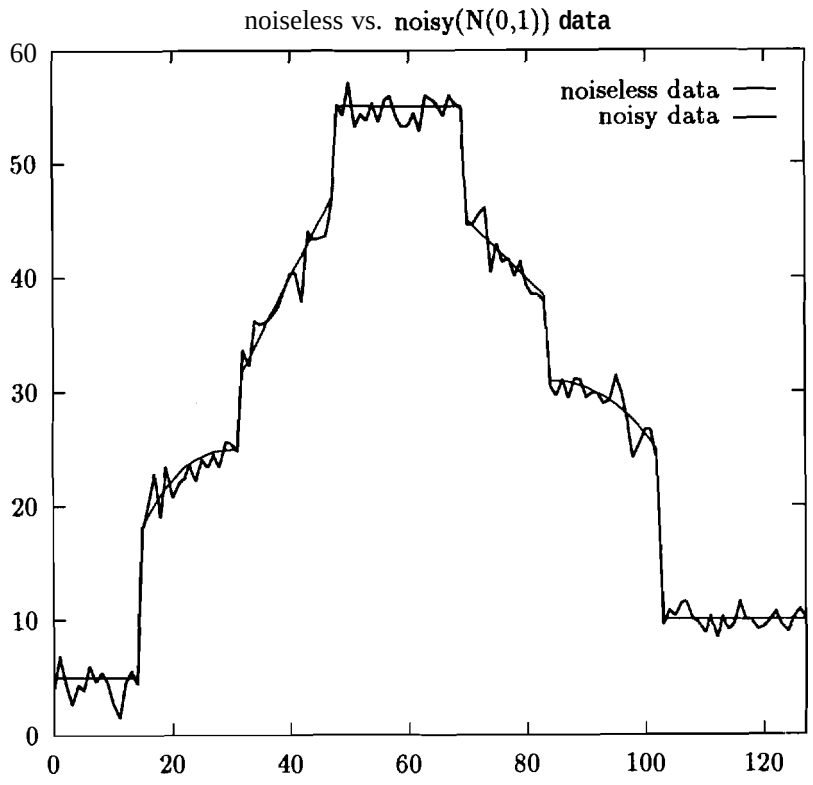


Figure 18: A section display of noiseless vs noisy image

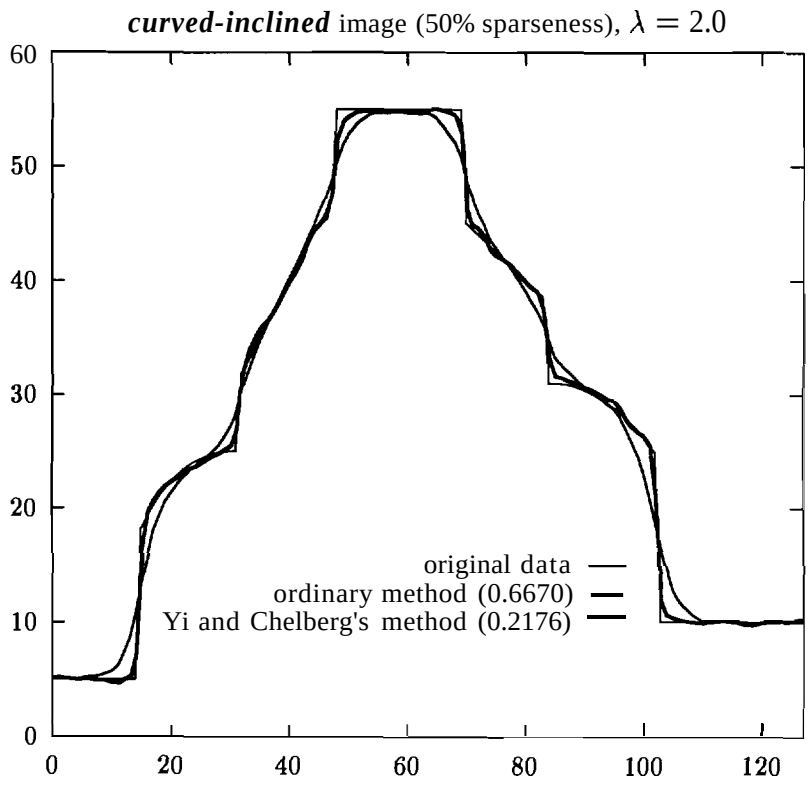


Figure 19: Reconstruction results with V/A measure using the LSE-PLANE method for a sparse image of which sparseness is 50%

8 Conclusion

We propose the volume between two surfaces normalized by the surface area as an invariant measure for comparing reconstruction results. We present a computationally simple and efficient method of computing the volume between two surfaces and the surface area by the least-square-fit plane approximation of a surface patch. The new methods computing the volume and the surface area were successfully applied to compute the proposed invariant performance measure for surface reconstruction. Knowing that a reconstructed surface is an approximated surface and the surface shape is ambiguous in regions between pixels, the described method of the least-square-error fit plane approximation gives good estimates of the volume between two surfaces and the area of a surface. The advantage of our method is that computation is extremely simple and efficient. A standard quantitative measure for the comparison of different reconstruction techniques allows analysis of different reconstruction algorithms when applied to the same input data. This ability to objectively compare different algorithms should facilitate further research in the area of surface reconstruction.

References

[1] J. Aloimonos and D. Schulman. Learning early-vision computations. J. Optical Society of America, 6(6), June 1989.

[2] A. Blake. Reconstructing a visible surface. In Proc. A A A I Conference, pages 23–26, 1984.

[3] A. Blake and A. Zisserman. Invariant surface reconstruction using weak continuity constraints. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pages 22–26, Miami, FL, June 1986.

[4] T. E. Boult and J. R. Kender. Visual surface reconstruction using sparse depth data. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pages 68–76, Miami, FL, June 1986.

[5] P. B. Chou and C. M. Brown. Multimodal reconstruction and segmentation with Markov random fields and HCF optimization. In Proc. DARPA Image Understanding Workshop, 1988.

[6] C. Chu and Alan C. Bovik. Visible surface reconstruction via local Minimax approximation. Pattern Recognition, 21(4):303–312, 1988.

[7] D. Geiger and F. Federico. Parallel and deterministic algorithms from MRF's: Surface reconstruction. IEEE Trans. *Patt.* Anal. Machine *Intell.*, 13(5):401–412, May 1991.

[8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Patt. Anal. Machine *Intell.*, 6(6):721–741, November 1984.

[9] W. E. Grimson and T. Pavlidis. Discontinuity detection for surface reconstruction. Computer Vision, Graphics, and Image Processing, 30:316–330, 1985.

[10] V. E. Johnson, W. H. Wong, X. Hu, and C. Chen. Image restoration using Gibbs priors: Boundary modeling, treatment of blurring, and selection of hyperparameter. IEEE Trans. *Patt.* Anal. Machine *Intell.*, 13(5):413–425, May 1991.

[11] S. Lang. Analysis I. Addison-Wesley, Reading, MA, 1969.

[12] Y. T. Lee and A. A. G. Requicha. Algorithms for computing the volume and other integral properties of solids. i. known methods and open issues. Communications of ACM, 25(9):635–641, September 1982.

[13] S. Z. Li. Invariant surface segmentation through energy minimization with discontinuities. Int. J. Comput. Vision, 5(2):161–194, 1990.

[14] J. Marroquin, S. Mitter, and T. Poggio. Probablistic solution of ill-posed problems in computational vision. J. American Statistical Association, 82(397):76–89, March 1987.

[15] P. Perona and J. Malik. Scale-space edge detection using anisotropic cliffusion. IEEE Trans. Patt. Anal. Machine *Intell.*, 12(7):629–639, July 1990.

[16] A. A. G. Requicha. Representations for rigid solids: Theory, methods, and systems. Computing Surveys, 12(4):437–464, December 1980.

[17] S. S. Sinha and B. G. Schunck. A two stage algorithm for discontinuity-preserving surface reconstruction. IEEE Trans. Patt. Anal. Machine *Intell.*, 14(1):36–55, November 1988.

[18] S. S. Sinha and B. G. Schunck. Discontinuity preserving surface reconstruction. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pages 229–234, San Diego, CA, June 1989.

[19] R. L. Stevenson and E. J. Delp. Viewpoint invariant recovery of visual surfaces from sparse data. IEEE Trans. Patt. Anal. Machine *Intell.*, 14(9):897–909, September 1992.

[20] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. Int. J. *Comput.* Vision, 5(3):271–301, 1990.

[21] R. Szeliski. Fast surface interpolation using hierarchical basis functions. IEEE Trans. Patt. Anal. Machine *Intell.*, 12(6):513–520, June 1990.

[22] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. IEEE Trans. Patt. Anal. Machine *Intell.*, 8(4):413–423, July 1986.

[23] D. Terzopoulos. The computation of visible-surface representations. IEEE Trans. Patt. Anal. Machine *Intell.*, 10(4):417–438, July 1988.

[24] J. Yi and D. M. Chelberg. Discontinuity-preserving and viewpoint invariant reconstruction of visible surfaces from sparse data using a first order regularization. In Proceedings of the IEEE International Conference on Automation, Robotics, and Computer Vision, volume 1, pages cv5.1.1–cv5.1.5, Singapore, September 1992.

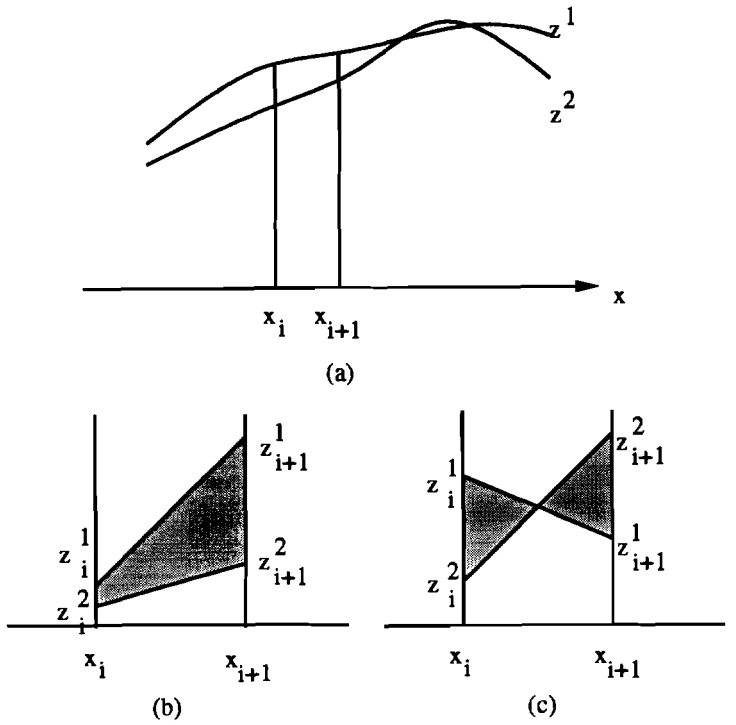


Figure 20: (a) Two reconstructed curves, z^1 and z^1. The area between two curve segments (b) when they do not intersect and (c) when they intersect.

A Area between two curves

A curve segment on each interval h is approximated by a linear segment as in Figure 20. The equations for two line segments in Figure 20 are $z^1 = \frac{z^1_{i+1}-z^1_i}{h}x + z^1_i$ and $z^2 = \frac{z^2_{i+1}-z^2_i}{h}\mathrm{x} + z^2_i$ where z^k, k $= 1, 2$ denotes two curves.

In case of Figure 20 (a),

$$\Delta A_i = \frac{h}{2}|z^2_{i+1} + z^2_i - z^1_{i+1} - z^1_i|$$

In case of Figure 20 (b), the intersection point (x_I, z_I) is $\left(\frac{h(t^l - z^2_i)}{-z^1_{i+1} - z^2_i - z^2_{i+1}}, \frac{-z^2_{i+1} z^1_i + z^1_i z^2_{i+1}}{-z^1_{i+1} - z^2_i - z^2_{i+1}} \right)$.

$$\begin{aligned}
\Delta A_i^{intersection} &= \frac{1}{2}\left| determinant \begin{pmatrix} 0 & z^1_i & 1 \\ 0 & z^2_i & 1 \\ x_I & z_I & 1 \end{pmatrix} \right| + \frac{1}{2}\left| determinant \begin{pmatrix} h & z^1_{i+1} & 1 \\ h & z^2_{i+1} & 1 \\ x_I & z_I & 1 \end{pmatrix} \right| \\
&= \frac{h}{2}\left(\left| \frac{(z^1_i)^2 - 2z^1_i z^2_i + (z^2_i)^2}{z^1_i - z^1_{i+1} - z^2_i - z^2_{i+1}} \right| + \left| \frac{(z^1_{i+1})^2 - 2z^1_{i+1} z^2_{i+1} + (z^2_{i+1})^2}{z^1_i - z^1_{i+1} - z^2_i - z^2_{i+1}} \right| \right) \qquad (15)
\end{aligned}$$

If $(z_i^1 - z_i^2)(z_{i+1}^1 - z_{i+1}^2) < 0$, use $\Delta A_i^{intersection}$. Otherwise use ΔA_i.

The arc length is computed as follows assuming n points for each curve.

$$l = \sum_{i=1}^{n-1} \sqrt{(z_{i+1} - z_i)^2 + h^2} \tag{16}$$