# Shape from Texture: General Principle

**Ken-ichi Kanatani**
*Department of Computer Science, Gunma University, Kiryu, Gunma 376, Japan*

**Tsai-Chia Chou**
*Center for Automation Research, University of Maryland, College Park, MD 20742, U.S.A.*

Recommended by H.-H. Nagel

ABSTRACT

*The 3D shape of a textured surface is recovered from its projected image on the assumption that the texture is homogeneously distributed. Our method does not require recognition of the "structure"—regularity, periodicity, parallelism, orthogonality, etc.—of the texture distribution. First, the "homogeneity" of a discrete texture is precisely defined in terms of the "theory of distributions". Next, distortion of the observed texture due to perspective projection is described in terms of the "first fundamental form" expressed as a function of the image coordinates. Based on this result, the 3D recovery equations for determining the surface shape are derived for both planar and curved surfaces. Some numerical schemes for solving these equations are proposed. Ambiguity in the interpretation of curved surfaces is also analyzed. Finally, some numerical examples for synthetic data are presented, and our method is compared with other existing methods.*

## 1. Introduction

Seeing a pattern with some kind of regularity, or *texture*, converging on a receding plane, humans can easily perceive the 3D depth of the scene. This fact has long since interested many researchers, and efforts have been made to simulate, by a computer, this seemingly "highly intelligent" human preception. This problem is now widely known as 3D recovery of *shape from texture*.

In general, 3D recovery from texture is possible if we have some "prior knowledge" about the true texture; if the observed texture has properties different from those of the true texture, the 3D shape is computed in such a way that the discrepancy is accounted for. For example, if the true texture is known to be an array of elements with a known shape, say circular, the surface

*Artificial Intelligence* **38** (1989) 1–48
0004-3702/89/$3.50 

gradient can be inferred from the observed distorted shape, say elliptical, of the elements. If the true texture elements are known to be distributed periodically at intervals of the same length, the surface gradient can be computed from the ratio of the converging interval lengths. If the true texture elements are aligned on parallel lines, or if individual texture elements have parallel line segments, the surface gradient is inferred from the "vanishing points" defined by pairs of such lines. Similar reasoning is possible if the true texture elements or their alignments are known to possess orthogonality or symmetry of some kind.

One major issue about these approaches is that we must first recognize the "structure" of the "true" texture—regularity, periodicity, collinearity, parallelism, orthogonality, symmetry, and so on. This is in general very difficult to automate by computer, since the "observed" texture does not exhibit the expected regularity, periodicity, etc. (That is why the 3D shape can be recovered!) Despite this difficulty, these "structure-based approaches" have been attempted by many researchers (e.g., [9, 10, 13–15, 20]). Perhaps this is because humans apparently seem to employ this type of inference; recognition of such texture structures is very easy for humans.

Then, a new approach which does not require the recognition of texture structure appeared. It is based on *statistical assumptions* about the true texture distribution. For example, if the true texture is distributed *isotropically*, namely, the line segments constituting the texture have no preferred orientations, the 3D surface shape can be computed from observed "preferred orientations". This approach was first proposed by Witkin [21], and the algorithm was improved by Davis et al. [4]. Kanatani [11] gave a rigorous mathematical description of the problem and explicit analytical formulae by invoking tensor calculus and *stereology*.

Another possible statistical assumption is *homogeneity*. When observed, the texture looks dense on the surface part away from the observer and sparse on the part near the observer. This phenomenon has also been considered to play a central role in human preception of the outside world (cf. Gibson [6–8], Sedwick [19]) and many attempts have been made to simulate this effect by a computer. However, most of the arguments were based on naive intuition or heuristics (e.g., Bajcsy and Lieberman [2], Rosenfeld [16], Zucker et al. [22]). It was not until Aloimonos and Swain [1] and Dunn [5] that the problem was treated in analytical terms based on the imaging geometry of perspective projection. However, their formations involve many ad hoc approximations and assumptions.

In this paper, we give a mathematically consistent treatment based on *differential geometry* and the *theory of distributions*—in particular, we show how to relate the "discreteness" of texture correctly with the description of "smooth surfaces" in "differential terms". As it turns out, the problem of shape from texture does not require heuristics or "high intelligence"; we need

not recognize the structure of the texture at all. We will demonstrate that the problem is easily solved by a simple "computational" principle.

We first give a precise definition of the *homogeneity* of a texture. If a texture consists of dots or line segments, the *texture density*, in its literal sense, is a singular function taking the value $\infty$ at the texture dots or line segments and 0 elsewhere. How can we say that the density is uniform? How can we tell that a given texture is homogeneously distributed? As we will see, how to define texture homogeneity is the core of the theory; all subsequent procedures depend on it. We define the texture density in a formal abstract way based on the *theory of distributions* (Schwarz [17, 18]).

We next give a mathematical analysis of the texture distortion due to perspective projection. We describe the perspective distortion in terms of the *first fundamental form* expressed as a function of the *image coordinates*. Our formulation consists of two stages. First, we present the 3D recovery equations for determining the surface shape for both planar and curved surfaces. Although these equations are difficult to solve directly, various theoretical consequences can be inferred, among which is the "ambiguity" in the interpretation of the surface shape.

Then, we propose numerical schemes for solving these equations, and also give some numerical examples for synthetic images. Various aspects related to applications and implementation of our method are also discussed.

In Appendices C and D, our formulation is compared with those of Aloimonos and Swain [1] and Dunn [5]. By doing this, we can better understand the essential mathematical structure of our formulation. At the same time, we can see that their methods are both explained in our mathematical framework.

## 2. Texture Density and Homogeneity

We study, in this section, how to define the *density* of a discrete texture. Consider textures composed of dots and line segments. If we are to seek a function $\rho(x, y)$ describing the amount of texture divided by the area it occupies, we are forced to consider delta-function-like singularities, the value of $\rho(x, y)$ being $\infty$ at the texture elements and 0 elsewhere, since the area of a dot or a line segment is 0. These types of singularities make analytical treatment very difficult. One way to avoid singularities is to define the texture density in a formal abstract way as a *functional*.

### 2.1. Dirac delta function

Let us recall the definition of the (*Dirac*) *delta function* $\delta(x)$. Mathematically, it is not a function; if a function takes the value 0 except at one point, its integral must be 0, since one point is of Lebesgue measure 0. It must be regarded as a *linear functional* $T$ mapping a smooth (say $C^\infty$) *test function* $m(x)$ having a finite

support (i.e., taking nonzero values only in a finite interval) to the value $m(0)$: $T[m(x)] = m(0)$. This functional is a well-defined mathematical entity.

However, it usually appeals more to our intuition if we *denote* it by $\int \delta(x)(\cdot)\,dx$ rather than $T[\cdot]$. As a result, the above definition is written as $\int \delta(x)m(x)\,dx = m(0)$. Hence, $\delta(x)$ is merely a symbolic notation. In fact, usually we do not use the delta function $\delta(x)$ itself: It is useful only when it is multiplied by some function and integrated. Hence, it suffices to define only the *rule of integration*; we need not worry about its singularity. This is the view developed in detail by Schwartz in his *theory of distributions* [17, 18].

### 2.2. Texture density as a functional

We fix a *window* $W$ on the textured image, and define the *texture density* $\rho(x, y)$ of a dot texture over the window $W$ formally as follows:

**Definition 2.1** (*Dot density*). The texture density $\rho(x, y)$ of a dot texture over window $W$ is a linear functional over a set $\mathcal{M}$, yet to be specified, of test functions $m(x, y)$ defined by

$$\int_W \rho(x, y)m(x, y)\,dx\,dy = \sum_{P_i \in W} m(x_i, y_i)\,, \tag{2.1}$$

where $P_i(x_i, y_i)$ are the positions of the dot texture elements on the image plane.

Since the texture density is defined formally as a functional, we need not worry about the singularities of $\rho(x, y)$. We can just "imagine" that $\rho(x, y)$ takes the value $\infty$ at texture elements and 0 elsewhere. All we need is the "rule of integration". The texture density of a line segment texture is similarly defined as follows.

**Definition 2.2** (*Line segment density*). The texture density $\rho(x, y)$ of a line segment texture in window $W$ is a linear functional over a set $\mathcal{M}$, yet to be specified, of test functions $m(x, y)$ defined by

$$\int_W \rho(x, y)m(x, y)\,dx\,dy = \sum_{L_i \subset W} \int_{L_i} m(x(l), y(l))\,dl\,, \tag{2.2}$$

where $L_i$ are the line segments on the image plane, and the right-hand side is the sum of curvilinear integrals along all line segments parameterized by arc length $l$.

Here again, we can "imagine" that $\rho(x, y)$ takes the value $\infty$ along texture line segments and 0 elsewhere.

### 2.3. Homogeneity

Now, we are in a position to define *homogeneity* of the texture density. Let $\rho(x, y)$ be the texture density formally defined above. We would like to say that the texture is homogeneous if $\rho(x, y) \approx c$, where $c$ is a constant. However, since the texture density is defined as a functional, this requirement must be interpreted *in the weak sense* or *in the sense of a distribution*. Namely

**Definition 2.3** (*Homogeneity*). A texture density $\rho(x, y)$ is homogeneous if

$$\int_W \rho(x, y) m(x, y)\, dx\, dy \approx c \int_W m(x, y)\, dx\, dy\,, \tag{2.3}$$

for test functions $m(x, y)$ of the set $\mathcal{M}$, yet to be specified, where $c$ is a constant independent of the test functions $m(x, y)$.

The constant $c$ can be interpreted as the *texture density* in the intuitive sense, i.e., the "total number of dots per unit area" or the "total length of line segments per unit area". Combined with the definitions of (2.1) and (2.2), this definition is restated as follows.

**Lemma 2.4.** *If the texture density is homogeneous, the following approximation holds*:

$$\int_W m(x, y)\, dx\, dy \approx \begin{cases} \dfrac{1}{c} \displaystyle\sum_{P_i \in W} m(x_i, y_i) & \textit{for dot textures}\,, \\ \dfrac{1}{c} \displaystyle\sum_{L_i \subset W} \int_{L_i} m(x(l), y(l))\, dl & \textit{for line segment textures}\,. \end{cases} \tag{2.4}$$

Equation (2.4) can be viewed as the *Monte Carlo method* to integrate a test function $m(x, y)$, where $1/c$ is the "area per dot" or the "area per unit length line segment". This type of Monte Carlo method is known to give a good approximation only when the distribution of the sampling points is "homogeneous". Here, we use this very fact to "define" the homogeneity. Namely, our definition is equivalent to saying that a texture is homogeneous *if* the Monte Carlo method of integration over the texture elements yields a good approximation.

**Remark 2.5.** If we choose, as a test function $m(x, y)$, the *characteristic function*,

$$\chi_S(x, y) = \begin{cases} 1, & (x, y) \in S, \\ 0, & \text{otherwise}, \end{cases} \tag{2.5}$$

of a region $S$, equation (2.4) states that the number of dots or the length of the region $S$ is approximately proportional to the area of $S$, the constant $c$ being the number or the length of texture elements in the region $S$ divided by its area. This is the interpretation which most people informally think of as the definition of homogeneity (cf. the method of Aloimonos and Swain [1] recapitulated in Appendix C).

**Remark 2.6.** As can be seen from Definitions 2.1 and 2.2, the definition of homogeneity depends on the choice of the set $\mathcal{M}$ of test functions $m(x, y)$. Even if the texture is sparse, it can be homogeneous for very smooth test functions $m(x, y)$ (i.e., viewed "coarsely" or viewed "with low resolution"), while it may not be homogeneous for rapidly varying test functions $m(x, y)$ (i.e., viewed "finely" or viewed "with high resolution"). Figuratively, we are looking at a discrete texture through "filters" $m(x, y)$, and the homogeneity is affected by the "coarseness" of the filter through which we are looking. If, for example, we take $\mathcal{M} = \{\exp i\pi(kx/a + ly/b)\}$, assuming that the window $W$ is a rectangle of size $2a \times 2b$, and set a certain threshold for the approximation of (2.3), we can define the *degree of homogeneity* by those $(k, l)$ satisfying the approximation. However, we do not go into the details, since what we have described so far is sufficient for the subsequent discussions.

### 2.4. Change of variables

Since the integration over the texture is defined as a functional by (2.1) and (2.2), we must be careful when we change the variables of integration: The rule for usual integration does not apply here. Consider two smooth functions $x'(x, y)$, $y'(x, y)$ such that the correspondence between $(x, y)$ and $(x', y')$ is one-to-one. Let $x(x', y')$, $y(x', y')$ express the inverse relationship. Suppose we use $x'$, $y'$ as new coordinates. Let $W'$ be the window in the $x'y'$-domain corresponding to the window $W$ in the $xy$-domain. Define the transformed texture density $\rho'(x', y')$ also as a functional by

$$\int_{W'} \rho'(x', y')m'(x', y')\,dx'\,dy' = \int_{W} \rho(x, y)m(x, y)\,dx\,dy\,, \tag{2.6}$$

where function $m'(x', y')$ is defined by $m'(x', y') \equiv m(x(x', y'), y(x', y'))$. Then, the action, as a functional, of the new density $\rho'(x', y')$ on a given function $m'(x', y')$ is given as follows.

**Proposition 2.7** (Density transformation).

$$\rho'(x', y') = \begin{cases} \rho(x(x', y'), y(x', y')) & \text{for dot textures,} \\ \rho(x(x', y'), y(x', y'))\Gamma(x', y', t') & \text{for line segment textures,} \end{cases} \tag{2.7}$$

*where*

$$\Gamma(x', y', t') \equiv \left\{ \left( \left( \frac{\partial x}{\partial x'} \right)^2 + \left( \frac{\partial y}{\partial x'} \right)^2 \right) t_1'^2 + 2 \left( \frac{\partial x}{\partial x'} \frac{\partial x}{\partial y'} + \frac{\partial y}{\partial x'} \frac{\partial y}{\partial y'} \right) t_1' t_2' + \left( \left( \frac{\partial x}{\partial y'} \right)^2 + \left( \frac{\partial y}{\partial y'} \right)^2 \right) t_2'^2 \right\}^{1/2} \tag{2.8}$$

*and $t' = (t_1', t_2')$ is the unit tangent vector to the line segment at point $(x', y')$ in the $x'y'$-domain.*

**Proof.** First, consider a dot texture. Let $P_i'(x', y')$ be the point in the $x'y'$-domain corresponding to point $P_i(x, y)$ in the $xy$-domain. Then, we see that

$$\begin{aligned} &\int_W \rho(x, y) m(x, y) \, dx \, dy \\ &= \sum_{P_i \in W} m(x_i, y_i) \\ &= \sum_{P_i' \in W'} m(x(x_i', y_i'), y(x_i', y_i')) = \sum_{P_i' \in W'} m'(x_i', y_i') . \end{aligned} \tag{2.9}$$

Since this relation defines the action of density $\rho'(x', y')$, as a functional, on the test function $m'(x', y')$ in the $x'y'$-domain, we obtain (2.7) for dot textures.

Next, consider a line segment texture. Let $L_i'$ be the line segment (parameterized by arc length $l'$) in the $x'y'$-domain corresponding to the line segment $L_i$ (parameterized by arc length $l$) in the $xy$-domain. Then, we see that

$$\begin{aligned} &\int_W \rho(x, y) m(x, y) \, dx \, dy \\ &= \sum_{L_i \subset W} \int_{L_i} m(x(l), y(l)) \, dl \end{aligned}$$

$$= \sum_{L_i' \subset W'} \int_{L_i'} m'(x'(l'), y'(l')) \left\{ \left( \left( \frac{\partial x}{\partial x'} \right)^2 + \left( \frac{\partial y}{\partial x'} \right)^2 \right) t_1'^2 \right.$$

$$\left. + 2 \left( \frac{\partial x}{\partial x'} \frac{\partial x}{\partial y'} + \frac{\partial y}{\partial x'} \frac{\partial y}{\partial y'} \right) t_1' t_2' + \left( \left( \frac{\partial x}{\partial y'} \right)^2 + \left( \frac{\partial y}{\partial y'} \right)^2 \right) t_2'^2 \right\}^{1/2} \mathrm{d}l'$$

$$= \sum_{L_i' \subset W'} \int_{L_i'} m'(x'(l'), y'(l')) \Gamma(x'(l'), y'(l'), t'(l')) \, \mathrm{d}l' . \tag{2.10}$$

Since this defines the action, as a functional, of the density $\rho'(x', y')$ on the test function $m'(x', y')$ in the $x'y'$-domain, we obtain (2.7) for line segment textures. □

**Remark 2.8.** For the usual integration of a continuous density $\rho(x, y)$, we would have

$$\rho'(x', y') = \rho(x(x', y'), y(x', y'))|\Delta(x', y')| , \tag{2.11}$$

where $\Delta(x', y')$ is the *Jacobian* defined by

$$\Delta(x', y') \equiv \begin{vmatrix} \partial x/\partial x' & \partial x/\partial y' \\ \partial y/\partial x' & \partial y/\partial y' \end{vmatrix} . \tag{2.12}$$

In summary, a continuous density is multiplied by the Jacobian $\Delta$, which is the magnification ratio of "area", while a line segment density is multiplied by $\Gamma$, which is the elongation ratio of "length", and a dot density is multiplied by 1, which is the increase ratio of "number". Note that the number of dots is preserved by a continuous mapping. Also note that the elongation ratio $\Gamma$ depends on the orientations of individual line segments as well as their positions.

## 3. First Fundamental Form

In this section, we describe the 3D shape of a surface in terms of the image coordinates.

### 3.1. Perspective projection

Take a Cartesian $XYZ$-coordinate system in the scene in such a way that the $Z$-axis corresponds to the camera optical axis. We take the point $(0, 0, -f)$ on the negative side of the $Z$-axis at distance $f$ from the $XY$-plane as the *viewpoint* corresponding to the center of the lens. The constant $f$ is often referred to as the *focal length*, though it does not necessarily coincide with the exact focal length of the lens.

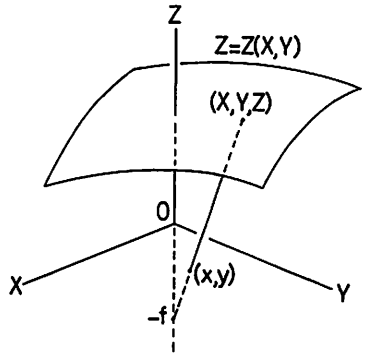



Fig. 1. Point $(X, Y, Z)$ on the surface $Z = Z(X, Y)$ is projected onto point $(x, y)$ on the image plane by perspective projection, point $(0, 0, -f)$ being the viewpoint.

We regard the $XY$-plane as the image plane, but when describing positions on the image plane, we use lower-case letters $x$, $y$. Namely, we fix an image $xy$-coordinate system in such a way that the $x$- and $y$-axes respectively coincide with the $X$- and $Y$-axes on the image plane. A point $(X, Y, Z)$ in the scene is mapped by perspective projection onto the intersection $(x, y)$ of the $XY$-plane with the ray connecting the point and the viewpoint (Fig. 1). It is easily seen from Fig. 1 that the relationship between the space coordinates $X$, $Y$, $Z$ and the image coordinates $x$, $y$ is given by

$$x = \frac{fX}{f+Z}, \qquad y = \frac{fY}{f+Z}. \tag{3.1}$$

### 3.2. Surface differential

Consider a smooth surface in the scene, and assume that it is described by an equation of the form of $Z = Z(X, Y)$. If this equation is given, equations (3.1) establish a one-to-one correspondence between the image plane and the surface part visible from the viewpoint. We first study how the space coordinates $X$, $Y$, $Z$ change on the surface. This is seen by taking *differentials* $\mathrm{d}X\,\mathrm{d}Y$, $\mathrm{d}Z$ along the surface. From (3.1), we obtain

$$f\,\mathrm{d}X - x\,\mathrm{d}Z = (f+Z)\,\mathrm{d}x, \qquad f\,\mathrm{d}Y - y\,\mathrm{d}Z = (f+Z)\,\mathrm{d}y. \tag{3.2}$$

Taking the differential of the surface equation $Z = Z(X, Y)$, we obtain

$$\mathrm{d}Z = P\,\mathrm{d}X + Q\,\mathrm{d}Y, \qquad P \equiv \partial Z/\partial X, \quad Q \equiv \partial Z/\partial Y. \tag{3.3}$$

Equations (3.2) and (3.3) can be viewed as a set of simultaneous linear equations in $\mathrm{d}X$, $\mathrm{d}Y$, $\mathrm{d}Z$. The solution is obtained in the form

$$dX = \frac{f+Z}{f(f-Px-Qy)}\,[(f-Qy)\,dx + Qx\,dy]\,,$$

$$dY = \frac{f+Z}{f(f-Px-Qy)}\,[Py\,dx + (f-Px)\,dy]\,, \tag{3.4}$$

$$dZ = \frac{f+Z}{f-Px-Qy}\,[P\,dx + Q\,dy]\,.$$

### 3.3. First fundamental form

Consider two points $(x, y)$, $(x + dx, y + dy)$ infinitesimally far apart on the image plane. Let $ds$ be the "3D distance" between the corresponding points on the surface (Fig. 2). If we substitute (3.4) into $ds^2 = dX^2 + dY^2 + dZ^2$, we obtain:

**Proposition 3.1** (First fundamental form).

$$ds^2 = \sum_{i,j=1}^{2} g_{ij}\,dx_i\,dx_j\,, \tag{3.5}$$

*where* $x_1 = x$, $x_2 = y$, *and*

$$g_{11} = \frac{(1+Z/f)^2}{(1-(Px+Qy)/f)^2}\left[(1+P^2) - 2Q\frac{y}{f} + (P^2+Q^2)\frac{y^2}{f^2}\right],$$

$$g_{12} = \frac{(1+Z/f)^2}{(1-(Px+Qy)/f)^2}\left[PQ + Q\frac{x}{f} + P\frac{y}{f} - (P^2+Q^2)\frac{xy}{f^2}\right] = g_{21}\,, \tag{3.6}$$

$$g_{22} = \frac{(1+Z/f)^2}{(1-(Px+Qy)/f)^2}\left[(1+Q^2) - 2P\frac{x}{f} + (P^2+Q^2)\frac{x^2}{f^2}\right].$$

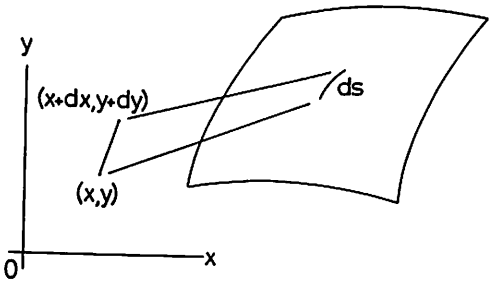



Fig. 2. The infinitesimal line segment connecting $(x, y)$ and $(x + dx, y + dy)$ on the image plane corresponds to an infinitesimal line segment on the surface whose length is $ds$.

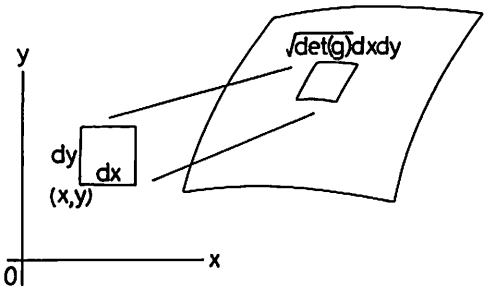



Fig. 3. The ratio of the area of an infinitesimal region on the surface to the area of the corresponding region on the image plane is given by $\sqrt{\det(g)}$.

Equation (3.5) is called the *first fundamental form*, and $g = (g_{ij})$, $i, j = 1, 2$, the *first fundamental metric tensor*. The first fundamental form indeed plays a fundamental role in computing 3D quantities of the surface in terms of the image coordinates. For example, consider a smooth curve $L$ on the image plane. The true arc length of the corresponding curve on the surface is given by the integration

$$\int_L \mathrm{d}s = \int_L \left( \sum_{i,j=1}^{2} g_{ij}\, \mathrm{d}x_i\, \mathrm{d}x_j \right)^{1/2}$$

on the image plane.

Consider an infinitesimally small square on the image plane defined by four points $(x, y)$, $(x + \mathrm{d}x, y)$, $(x, y + \mathrm{d}y)$, $(x + \mathrm{d}x, y + \mathrm{d}y)$ (Fig. 3). The area of this square on the image plane is $\mathrm{d}x\, \mathrm{d}y$, but the true area of the corresponding region on the surface is, as is well known, $\sqrt{\det(g)}\, \mathrm{d}x\, \mathrm{d}y$. From equations (3.6), we obtain

$$\sqrt{\det(g)} = \frac{\sqrt{1 + P^2 + Q^2}(1 + Z/f)^2}{1 - (Px + Qy)/f} . \tag{3.7}$$

Hence, the true area of the region on the surface corresponding to a region $S$ on the image plane is given by integration

$$\int_S \sqrt{\det(g)}\, \mathrm{d}x\, \mathrm{d}y$$

on the image plane.

### 3.4. Planar surfaces

If the surface is a plane whose equation is $Z = pX + qY + r$, where $p$, $q$, $r$ are constants, (3.1) can be solved for $X$, $Y$ in the form

$$X = \frac{(f + r)x}{f - px - qy} , \quad Y = \frac{(f + r)y}{f - px - qy} , \quad Z = \frac{f(px + qy + r)}{f - px - qy} . \tag{3.8}$$

Then, (3.6) and (3.7) become:

$$g_{11} = \frac{(1+r/f)^2}{(1-(px+qy)/f)^4}\left[1+p^2-2q\frac{y}{f}+(p^2+q^2)\frac{y^2}{f^2}\right],$$
$$g_{12} = \frac{(1+r/f)^2}{(1-(px+qy)/f)^4}\left[pq+q\frac{x}{f}+p\frac{y}{f}-(p^2+q^2)\frac{xy}{f^2}\right] = g_{21}, \tag{3.9}$$
$$g_{22} = \frac{(1+r/f)^2}{(1-(px+qy)/f)^4}\left[1+q^2-2p\frac{x}{f}+(p^2+q^2)\frac{x^2}{f^2}\right];$$

$$\sqrt{\det(g)} = \frac{\sqrt{1+p^2+q^2}(1+r/f)^2}{(1-(px+qy)/f)^3}. \tag{3.10}$$

## 4. Surface Shape Recovery from Texture

In this section, we propose a mathematical principle for reconstructing the surface shape for observation of inhomogeneous texture density $\rho(x, y)$ based on the assumption that the true surface is homogeneously textured. For this purpose, we first study how the texture density $\rho(x, y)$ is distorted by perspective projection. Since the texture density is defined abstractly as a functional, what we need to know is how the observed density $\rho(x, y)$ acts on test functions $m(x, y)$. From this, we derive the 3D recovery equations for determining the surface shape in terms of *observables* computed on the image plane.

### 4.1. Distortion of homogeneous texture

Suppose the equation $Z = Z(X, Y)$ of the surface under consideration is known. Consider temporarily a curvilinear $uv$-coordinate system on the surface. (We will soon do away with any surface coordinate systems.) Since the surface equation is known there is a one-to-one correspondence between the image plane and the visible part of the surface, so that we can express the correspondence in the form of $u = u(x, y)$, $v = v(x, y)$, or $x = x(u, v)$, $y = y(u, v)$. Let $W_0$ be the region of the surface corresponding to the window $W$ on the image plane. Let $\rho_0(u, v)$ be the homogeneous texture density on the surface. It is a functional, and the homogeneity is expressed in the form

$$\int_{W_0} \rho_0(u, v) m_0(u, v)\, dS_0 \approx c \int_{W_0} m_0(u, v)\, dS_0, \tag{4.1}$$

where $m_0(u, v)$ is a test function, and $dS_0$ is the area element of the surface.

Since the right-hand side is an ordinary integration, it can be rewritten in

terms of the image coordinates if we note the relation $dS_0 = \sqrt{\det(g)}\, dx\, dy$; namely, it becomes

$$c \int_W m(x, y)\sqrt{\det(g)}\, dx\, dy\,, \tag{4.2}$$

where we put $m(x, y) \equiv m_0(u(x, y), v(x, y))$.

On the other hand, if we want to rewrite the left-hand side of (4.1) in terms of the image coordinates, the expression depends on whether the texture consists of dots or line segments. For a dot texture, the left-hand side of (4.1) is rewritten, from (2.6) and Proposition 2.7, as

$$\sum_{P_i \in W} m(x_i, y_i)\,, \tag{4.3}$$

which is a quantity that can be computed on the image plane (i.e., an "observable").

For a line segment texture, the left-hand side of (4.1) is rewritten, from Proposition 2.7, as

$$\sum_{L_i \subset W} \int_{L_i} m(x(l), y(l))\Gamma(x(l), y(l), t(l))\, dl\,, \tag{4.4}$$

where $l$ is the arc length, $t$ is the unit tangent vector along the line segments on the image plane, and

$$\Gamma(x, y, t) \equiv \sqrt{g_{11}(x, y)t_1^2 + 2g_{12}(x, y)t_1t_2 + g_{22}(x, y)t_2^2}\,. \tag{4.5}$$

Function $\Gamma(x, y, t)$ describes the elongation ratio of the line segments on the surface compared with their projections on the image plane. The fact that it depends on the orientation of the line segment passing through the point where $\Gamma(x, y, t)$ is evaluated makes the subsequent analysis very difficult. Here, we adopt the approximation

$$\Gamma(x, y, t) \approx (\sqrt{\det(g)})^{1/2}\,. \tag{4.6}$$

Then, as we will show now, dot textures and line segment textures can be treated in the same formulation.

The interpretation of approximation (4.6) is that the line segments are distributed nearly isotropically, and hence if the area is enlarged $\sqrt{\det(g)}$ times, the individual line segments become roughly $(\sqrt{\det(g)})^{1/2}$ times as long. As a result, if we regard $m(x, y)\Gamma(x, y) \approx m(x, y)(\sqrt{\det(g)})^{1/2}$ as a new test function $m(x, y)$ (ignoring the dependence on orientation $t$), we can express

the left-hand side of (4.1) in the form of

$$\int_W \rho(x, y)m(x, y)\,dx\,dy\,, \tag{4.7}$$

which is a quantity that can be computed on the image plane (i.e., an "observable"). The right-hand side of (4.1) then becomes

$$c\int_W m(x, y)(\sqrt{\det(g)})^{1/2}\,dx\,dy\,. \tag{4.8}$$

**Remark 4.1.** Equations (4.2) and (4.8) are intuitively interpreted as follows. Consider a small region $S$ on the image plane, and let $S_0$ be the corresponding region on the surface. For a dot texture, the number of dots in $S$ is equal to the number of dots in $S_0$, while the area of $S$ is $1/\sqrt{\det(g)}$ times that of $S_0$. Hence, the texture density in $S$ is $\sqrt{\det(g)}$ times that in $S_0$. For a line segment texture, if the texture is nearly isotropic, the total length of the line segments in $S$ is approximately $1/(\sqrt{\det(g)})^{1/2}$ times that of $S_0$, while the area of $S$ is $1/\sqrt{\det(g)}$ times that of $S_0$. Hence, the texture density in $S$ is $(\sqrt{\det(g)})^{1/2}$ times that in $S_0$.

### 4.2. Principle of surface recovery

Our principle of surface recovery is as follows.

(1) Let the object surface be modeled by a parameterized equation in the form

$$Z = Z(X, Y; \alpha, \beta, \gamma, \ldots)\,, \tag{4.9}$$

where $\alpha$, $\beta$, $\gamma, \ldots$ are the surface shape parameters to be determined.

(2) From the above assumed surface model and the geometry of perspective projection (equations (3.1)), compute the first fundamental metric tensor

$$g(x, y; \alpha, \beta, \gamma, \ldots) = (g_{ij}(x, y; \alpha, \beta, \gamma, \ldots))\,, \quad i, j = 1, 2\,, \tag{4.10}$$

which is a function of the surface shape parameters $\alpha$, $\beta$ $\gamma, \ldots$.

(3) Provide appropriately test functions $m_0(x, y)$, $m_1(x, y)$, $m_2(x, y), \ldots$, and compute the corresponding *observables*

$$J_i = \int_W \rho(x, y)m_i(x, y)\,dx\,dy\,, \quad i = 0, 1, 2, \ldots, \tag{4.11}$$

which can be obtained by evaluating the test functions $m_i(x, y)$ at the texture

points or integrating them along the line segments on the image plane (equations (2.1) and (2.2)).

(4) Replacing the approximation symbol in (4.1) with equality, and taking the ratio $J_i/J_0$ to eliminate the unknown true texture density value $c$, we obtain the *3D recovery equations* to determine the surface shape parameters in the form

$$\frac{J_i}{J_0} = \frac{\int_W m_i(x, y)(\sqrt{\det(g(x, y; \alpha, \beta, \gamma, \ldots))})^\kappa \, dx\, dy}{\int_W m_0(x, y)(\sqrt{\det(g(x, y; \alpha, \beta, \gamma, \ldots))})^\kappa \, dx\, dy}, \quad i = 1, 2, \ldots . \tag{4.12}$$

Here, $\kappa = 1$ for a dot texture (equation (4.2)) and $\kappa = 1/2$ for a line segment texture (equation (4.8)). Equation (4.12) provides a set of nonlinear equations in unknowns $\alpha, \beta, \gamma, \ldots$. The values of these surface shape parameters are determined by solving these equations.

**Remark 4.2.** It is intuitively easy to understand that this is the most "natural" formulation derivable from the assumption of texture homogeneity. Since the change of the texture density is observed through the change of the "surface area" under perspective projection, the equation *must* be expressed in terms of $\sqrt{\det(g)}$. However, since the texture elements are discretely distributed, we *cannot* observe $\sqrt{\det(g)}$ point-wise: We need some "smoothing" over a finite domain. The test functions $m_i(x, y)$, yet to be specified, do the required smoothing.

## 5. Recovery of Planar Surfaces

### 5.1. 3D recovery equations

The simplest case is when the surface is a plane. If we model the planar surface by equation $Z = pX + qY + r$, we can obtain $\sqrt{\det(g)}$ in the form of (3.10). Hence, we obtain

$$c \int_W m(x, y)(\sqrt{\det(g)})^\kappa \, dx\, dy = c(\sqrt{1 + p^2 + q^2})^\kappa \left(1 + \frac{r}{f}\right)^2 \int_W \frac{m(x, y)\, dx\, dy}{(1 - (px + qy)/f)^3} . \tag{5.1}$$

From this, we obtain the following equations.

**Proposition 5.1** (3D recovery equations). *The surface gradient* $(p, q)$ *is determined by solving*

$$\int_W \frac{m_i(x, y) - (J_i/J_0)m_0(x, y)}{(1 - (px + qy)/f)^{3\kappa}} \, dx \, dy = 0, \quad i = 1, 2. \tag{5.2}$$

**Remark 5.2.** Equation (5.2) provides two equations in the unknowns $p$, $q$, and can be solved in principle, say, by iterative search in the $pq$-plane (the *gradient space*). Evidently, three test functions $m_0(x, y)$, $m_1(x, y)$, $m_2(x, y)$ are sufficient for determination of the two unknowns $p$, $q$. However, we can also use many more test functions and determine $p$, $q$ by some fitting scheme, say the least square method (cf. the method of Dunn [5] recast in Appendix D).

**Remark 5.3.** Note that the denominator becomes zero (i.e., $\sqrt{\det(g)}$ becomes $\infty$) along the line $px + qy = f$ on the image plane. This line is known as the *vanishing line* or *horizon* of the planar surface. Evidently, we must avoid this line by appropriately confining the size and location of the window $W$. Then, we can assume that no vanishing line exists inside the window $W$. In fact, if the vanishing line is observed, there is no need to use the texture to estimate the surface gradient; the equation $px + qy = f$ the vanishing line immediately tells us the gradient $(p, q)$.

### 5.2. Small gradient approximation

Suppose the surface gradient $(p, q)$ is close to zero compared with the focal length $f$: $px + qy \ll f$. Then, in the integrand of the basic equations (5.2), Taylor expansion around the image origin $O$ yields

$$\frac{1}{(1 - (px + qy)/f)^{3\kappa}} = 1 + \frac{3\kappa}{f}(px + qy) + \cdots. \tag{5.3}$$

If we put

$$\begin{aligned} L_i &= \int_W m_i(x, y) \, dx \, dy, \\ M_i &= \frac{3\kappa}{f} \int_W x m_i(x, y) \, dx \, dy, \\ N_i &= \frac{3\kappa}{f} \int_W y m_i(x, y) \, dx \, dy, \end{aligned} \tag{5.4}$$

for $i = 0, 1, 2$, and neglect higher-order terms, the 3D recovery equations (5.2) reduce to the following linear equations in $p$, $q$:

$$\begin{bmatrix} M_1 - (J_1/J_0)M_0 & N_1 - (J_1/J_0)N_0 \\ M_2 - (J_2/J_0)M_0 & N_2 - (J_2/J_0)N_0 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = -\begin{bmatrix} L_1 - (J_1/J_0)L_0 \\ L_2 - (J_2/J_0)L_0 \end{bmatrix}. \tag{5.5}$$

A simple choice of the test functions $m_i(x, y)$, $i = 0, 1, 2$, is

$$m_0(x, y) = 1\,, \qquad m_1(x, y) = x\,, \qquad m_2(x, y) = y\,. \tag{5.6}$$

This means that we compute the following observables:

$$\begin{aligned} J_0 &= \int_W \rho(x, y)\,dx\,dy\,, \\ J_1 &= \int_W x\rho(x, y)\,dx\,dy\,, \qquad J_2 = \int_W y\rho(x, y)\,dx\,dy\,. \end{aligned} \tag{5.7}$$

Note that $(J_1/J_0, J_2/J_0)$ is the *center of gravity* of the texture inside the window $W$. In other words, if each dot has unit mass or each line segment has unit mass per length, the center of gravity $(\bar{x}, \bar{y})$ is given by $(J_1/J_0, J_2/J_0)$. Thus, what we need is

$$\bar{x} = \frac{J_1}{J_0} = \frac{\int_W x\rho(x, y)\,dx\,dy}{\int_W \rho(x, y)\,dx\,dy}\,, \qquad \bar{y} = \frac{J_2}{J_0} = \frac{\int_W y\rho(x, y)\,dx\,dy}{\int_W \rho(x, y)\,dx\,dy}\,. \tag{5.8}$$

In particular, if the window $W$ is a rectangle defined by $-a \leqslant x \leqslant a$, $-b \leqslant y \leqslant b$, equations (5.4) reduce to

$$\begin{aligned} &L_0 = 4ab\,, \qquad M_1 = 4\kappa a^3 b/f\,, \qquad N_2 = 4\kappa ab^3/f\,, \\ &L_1 = L_2 = M_0 = M_2 = N_0 = N_1 = 0\,, \end{aligned}$$

and hence the solution of (5.5) is given by

$$p = \frac{f}{\kappa a^2}\,\bar{x}\,, \qquad q = \frac{f}{\kappa b^2}\,\bar{y}\,. \tag{5.9}$$

This result is intuitively interpreted as follows: Suppose $(p, q) = (0, 0)$, i.e., the surface is viewed orthogonally. Then, the center of gravity should coincide with the image origin $O$ if the texture is truly homogeneous. Otherwise, the orientation and the "shift" of the center of gravity give the surface gradient $(p, q)$ in the form of (5.9).

**Remark 5.4.** The accuracy of the result depends on both the number or length of the texture elements in the window $W$ and their distribution patterns. Let $N$ be the number or the length of the texture elements in the window $W$. The rule of thumb is that the error is approximately proportional to $1/\sqrt{N}$ when the texture is completely random, and is approximately proportional to $1/N$ when the texture is very regular and periodic (Appendix A). Textures we often

encounter in natural scenes and man-made objects are usually regular and periodic tessellations. In such cases, high accuracy is expected. However, as we discussed in Section 1, *the computer need not recognize the regularity or periodicity, if they exist.*

## 6. Numerical Scheme of Planar Surface Recovery

As we showed in Section 5, the solution of the 3D recovery equation is immediately obtained if the gradient $(p, q)$ is close to zero (equation (5.5)). If it is not close to zero, we need some iterative scheme.

### 6.1. Principle of camera rotation simulation

Suppose the camera is rotated by a certain angle around the center of the lens (i.e., the viewpoint) relative to a stationary scene. As a result, a different image is seen on the image plane. However, the essential information contained in the image is the same if we ignore the existence of the boundary of the image plane: A point on the image plane corresponds to a "ray" passing through the viewpoint, and the same rays are still observed after the camera is rotated. In particular, the state of occlusion is not affected by the camera rotation. In fact, we can obtain an explicit expression of the image transformation due to the camera rotation. The transformation *does not require any knowledge about the 3D scene.*

Suppose the camera is rotated by an orthogonal matrix $\boldsymbol{R} = (r_{ij})$, $i, j = 1, 2, 3$, around the viewpoint. The rotation of the camera by $\boldsymbol{R}$ is equivalent to the rotation of the scene by $\boldsymbol{R}^{-1}$ $(=\boldsymbol{R}^{\mathrm{T}})$. By rotation $\boldsymbol{R}^{\mathrm{T}}$, a point $(X, Y, Z)$ in the scene moves to a point $(X', Y', Z')$ given by

$$\begin{bmatrix} X' \\ Y' \\ f + Z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ f + Z \end{bmatrix} \tag{6.1}$$

(cf. Fig. 1). This point is projected onto $(x', y')$ in the image plane given by $x' = fX'/(f + Z')$, $y' = fY'/(f + Z')$. Combining this with (3.1), we obtain the following transformation rule:

$$x' = f\,\frac{r_{11}x + r_{21}y + r_{31}f}{r_{13}x + r_{23}y + r_{33}f}, \qquad y' = f\,\frac{r_{12}x + r_{22}y + r_{32}f}{r_{13}x + r_{23}y + r_{33}f}. \tag{6.2}$$

For a more general discussion of this camera rotation transformation and its invariance, see [12].

Suppose the surface gradient is not small. We first apply the method of Section 5. Let $(p_0, q_0)$ be the computed gradient. This estimation may not be accurate. Suppose the camera is rotated in such a way that the estimated surface becomes parallel to the new image plane. If we regard this new image

as the input, the surface gradient should be small. Hence the method of the preceding section can be applied. Let $(p_0', q_0')$ be the computed surface gradient. If this newly estimated surface is rotated back into the original camera orientation, we obtain a better estimation $(p_1, q_1)$. This process can be applied repeatedly, producing a sequence of estimations $(p_i, q_i)$, $i = 0, 1, 2, \ldots$, until no further improvement is made. This is the basic concept of the scheme of camera rotation simulation.

We should emphasize the fact that *the camera need not actually be rotated*, because the image transformation is given in the analytical expression (6.2). However, we should also note that *the transformed image need not be generated*. This is because *all we need is the values of the observables* $J_i$, $i = 0, 1, 2$. As we will now show, we can derive the "transformation rule of observables"; we only need to compute the "transformed observables" $J_i'$, $i = 0, 1, 2$. To be specific, if we want to compute observable $J_i$ for test function $m'(x', y')$ over the "transformed image" $\rho'(x', y')$ inside the "transformed window" $W'$, we can compute it by integrating a "modified test function" $\tilde{m}(x, y)$ applied to the "original image" $\rho(x, y)$ inside the "original window" $W$:

$$\int_{W'} m'(x', y')\rho'(x', y')\,dx'\,dy' = \int_{W} \tilde{m}(x, y)\rho(x, y)\,dx\,dy\,. \tag{6.3}$$

### 6.2. Camera rotation simulation

Let $(p_0, q_0)$ be the initial estimate of the gradient. The corresponding surface unit normal vector $\boldsymbol{n} = (n_1, n_2, n_3)$ is given by

$$n_1 = \frac{-p_0}{\sqrt{1 + p_0^2 + q_0^2}}, \qquad n_2 = \frac{-q_0}{\sqrt{1 + p_0^2 + q_0^2}}, \qquad n_3 = \frac{1}{\sqrt{1 + p_0^2 + q_0^2}}. \tag{6.4}$$

This vector makes angle

$$\Omega = \cos^{-1} n_3 \tag{6.5}$$

with the unit vector $\boldsymbol{k} = (0, 0, 1)$ along the $Z$-axis. The unit vector which is normal to both $\boldsymbol{n}$ and $\boldsymbol{k}$ is given by

$$\boldsymbol{l} = \frac{\boldsymbol{k} \times \boldsymbol{n}}{\|\boldsymbol{n} \times \boldsymbol{k}\|} = \frac{(q_0, -p_0, 0)}{\sqrt{p_0^2 + q_0^2}}\,. \tag{6.6}$$

If the camera is rotated around this vector $\boldsymbol{l}$ by angle $\Omega$ screw-wise, the estimated surface becomes parallel to the new image plane. The corresponding

rotation matrix is given by

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & n_1 \\ r_{12} & r_{22} & n_2 \\ -n_1 & -n_2 & n_3 \end{bmatrix}, \tag{6.7}$$

where

$$r_{11} = \frac{p_0^2 n_3 + q_0^2}{p_0^2 + q_0^2}, \qquad r_{12} = \frac{p_0 q_0 (n_3 - 1)}{p_0^2 + q_0^2}, \qquad r_{22} = \frac{q_0^2 n_3 + p_0^2}{p_0^2 + q_0^2}. \tag{6.8}$$

Hence, the image transformation is given by

$$x'(x, y) = f \frac{r_{11}x + r_{12}y - n_1 f}{n_1 x + n_2 y + n_3 f}, \qquad y'(x, y) = f \frac{r_{12}x + r_{22}y - n_2 f}{n_1 x + n_2 y + n_3 f}. \tag{6.9}$$

Its Jacobian is

$$\begin{aligned} \Delta(x, y) &\equiv \begin{vmatrix} \partial x'/\partial x & \partial x'/\partial y \\ \partial y'/\partial x & \partial y'/\partial y \end{vmatrix} \\ &= -f \frac{n_1 x'(x, y) + n_2 y'(x, y) - n_3 f}{(n_1 x + n_2 + n_3 f)^2}. \end{aligned} \tag{6.10}$$

We can see that (6.9) maps the image origin $O$ onto point $(fp_0, fq_0)$. If the original window $W$ is placed near the image origin $O$, the transformed window $W'$ is located near point $(fp_0, fq_0)$. Let $(p', q')$ be the true surface gradient relative to this new $x'y'$-image plane. Our next aim is to estimate this gradient $(p', q')$. Taylor expansion around point $(fp_0, fq_0)$ yields

$$\begin{aligned} &\frac{1}{(1 - (p'x' + q'y')/f)^{3\kappa}} \\ &\quad = \frac{1}{(1 - p_0 p' - q_0 q')^{3\kappa}} (1 + A(x' - fp_0) + B(y' - fq_0) + \cdots), \end{aligned} \tag{6.11}$$

where

$$A = \frac{3\kappa p'}{f(1 - p_0 p' - q_0 q')}, \qquad B = \frac{3\kappa q'}{f(1 - p_0 p' - q_0 q')}. \tag{6.12}$$

If we use $m_i'(x', y')$, $i = 0, 1, 2$, as the test functions, and substitute (6.11) into the 3D recovery equations (5.2) in the $x'y'$-domain, we obtain a set of linear equations which has the form (5.5), namely

$$\begin{bmatrix} M_1' - (J_1'/J_0')M_0' & N_1' - (J_1'/J_0')N_0' \\ M_2' - (J_2'/J_0')M_0' & N_2' - (J_2'/J_0')N_0' \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = -\begin{bmatrix} L_1' - (J_1'/J_0')L_0' \\ L_2' - (J_2'/J_0')L_0' \end{bmatrix}, \tag{6.13}$$

where

$$J_i' = \int_{W'} m_i'(x', y')\rho'(x', y')\,dx'\,dy', \tag{6.14}$$

$$L_i' = \int_{W'} m_i'(x', y')\,dx'\,dy',$$

$$M_i' = \int_{W'} (x' - fp_0)m_i'(x', y')\,dx'\,dy', \tag{6.15}$$

$$N_i' = \int_{W'} (y' - fq_0)m_i'(x', y')\,dx'\,dy'$$

for $i = 0, 1, 2$, and $\rho'(x', y')$ is the transformed texture density.

Once $A$, $B$ are determined by solving (6.13), the gradient $(p', q')$ is determined by solving (6.12), which are rewritten as

$$\begin{bmatrix} Ap_0 + 3\kappa/f & Aq_0 \\ Bp_0 & Bq_0 + 3\kappa/f \end{bmatrix}\begin{bmatrix} p' \\ q' \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix}. \tag{6.16}$$

If the computed gradient $(p', q')$ is sufficiently close to zero, the initial estimate is correct. Otherwise, the camera is rotated back into the original orientation, and the surface gradient is transformed into

$$p_1 = \frac{r_{11}p' + r_{12}q' - n_1}{n_1p' + n_2q' + n_3}, \qquad q_1 = \frac{r_{12}p' + r_{22}q' - n_2}{n_1p' + n_2q' + n_3}. \tag{6.17}$$

(Note that the "camera" was rotated by $\boldsymbol{R}$. This means that the surface was rotated by $\boldsymbol{R}^{-1}$ $(=\boldsymbol{R}^{\mathrm{T}})$ relative to the camera. Hence, the surface is rotated back into its original position by rotation $(\boldsymbol{R}^{-1})^{-1} = \boldsymbol{R}$.) This process can be iterated to compute $(p_2, q_2)$, $(p_3, q_3)$, ... until convergence.

### 6.3. Computation on the original image

In the above procedure, the only quantities which must be computed for the transformed density $\rho'(x', y')$ over the transformed window $W'$ are equations (6.14) and (6.15). We now show that *they can be computed for the original density* $\rho(x, y)$ *over the original window* $W$ *by changing variables*. For a dot texture, we see from Proposition 2.7 that if we put

$$\tilde{m}_i(x, y) \equiv m_i'(x'(x, y), y'(x, y)), \quad i = 0, 1, 2, \tag{6.18}$$

(6.14) is written as

$$J_i' = \sum_{P_i \in W} \tilde{m}_i(x_i, y_i), \quad i = 0, 1, 2, \tag{6.19}$$

which can be computed over the original image.

For a line segment texture, we obtain, from Proposition 2.7, the following rule. (Note that variables $x$, $y$ in Proposition 2.7 correspond to $x'$, $y'$ here, and variables $x'$, $y'$ there correspond to $x$, $y$ here.)

$$J_i' = \sum_{L_i \subset W} \int_{L_i} \tilde{m}_i(x(l), y(l)) \sqrt{Et_1^2 + 2Ft_1t_2 + Gt_2^2}\, dl, \quad i = 0, 1, 2. \tag{6.20}$$

Here, $l$ is the arc length, $t = (t_1, t_2)$ is the unit tangent vector along the individual line segments of the "original" image, while $E$, $F$, $G$ are functions of $x$, $y$ defined by

$$\begin{aligned} E(x, y) &\equiv \left(\frac{\partial x'}{\partial x}\right)^2 + \left(\frac{\partial y'}{\partial x}\right)^2 \\ &= f^2 \frac{1 + n_1^2(x'^2 + y'^2 - 1) - 2n_1(r_{11}x' + r_{12}y')}{(n_1x + n_2y + n_3f)^2}, \\ F(x, y) &\equiv \frac{\partial x'}{\partial x}\frac{\partial x'}{\partial y} + \frac{\partial y'}{\partial x}\frac{\partial y'}{\partial y} \\ &= f^2 \frac{n_1n_2(x'^2 + y'^2 - 1) - n_1(r_{12}x' + r_{22}y') - n_2(r_{11}x' + r_{12}y')}{(n_1x + n_2y + n_3f)^2}, \\ G(x, y) &\equiv \left(\frac{\partial x'}{\partial y}\right)^2 + \left(\frac{\partial y'}{\partial y}\right)^2 \\ &= f^2 \frac{1 + n_2^2(x'^2 + y'^2 - 1) - 2n_2(r_{12}x' + r_{22}y')}{(n_1x + n_2y + n_3f)^2}. \end{aligned} \tag{6.21}$$

If we use the relations of (6.9), these are all expressed in terms of the original coordinates $x$, $y$. Hence, (6.20) is computed over the original image.

Equations (6.15) are, on the other hand, integrations of continuous functions. Hence, we immediately rewrite them in terms of the original coordinates $x$, $y$ as

$$\begin{aligned} L_i' &= \int_W \tilde{m}_i(x, y)\Delta(x, y)\, dx\, dy, \\ M_i' &= \int_W (x'(x, y) - fp_0)\tilde{m}_i(x, y)\Delta(x, y)\, dx\, dy, \end{aligned} \tag{6.22}$$

$$N_i' = \int_W (y'(x, y) - fq_0)\tilde{m}_i(x, y)\Delta(x, y)\,dx\,dy\,,$$

for $i = 1, 2, 3$. They are easily computed by a numerical integration scheme.

**Remark 6.1.** Equation (6.20) is a rigorous relation. Since the Jacobian $\Delta(x, y)$ plays the role of $\sqrt{\det(g)}$ in the discussion in Section 4, a simple approximation corresponding to (4.6) is

$$\sqrt{Et_1^2 + 2Ft_1t_2 + Gt_2^2}\,dl \approx \sqrt{\Delta(x, y)}\,, \tag{6.23}$$

and hence we could use the approximation

$$J_i' \approx \sum_{L_i \subset W} \int_{L_i} \tilde{m}_i(x(l), y(l))\sqrt{\Delta(x(l), y(l))}\,dl\,. \tag{6.24}$$

However, this approximation is not used here. See Remark 6.2 below.

### 6.4. Alternative method by Taylor expansion

The method described above is somewhat complicated. There exists an alternative scheme. Suppose $(p_k, q_k)$ is the estimate at the $k$th step. If we expand the left-hand side of (5.3) at $(p_k, q_k)$ rather than at $(0, 0)$, we obtain

$$\frac{1}{(1 - (px + qy)/f)^{3\kappa}} = L(x, y) + M(x, y)\delta p_k + N(x, y)\delta q_k + \cdots\,, \tag{6.25}$$

where $\delta p_k = p - p_k$, $\delta q_k = q - q_k$, and

$$\begin{aligned} L(x, y) &= \frac{1}{(1 - (p_kx + q_ky)/f)^{3\kappa}}\,, \\ M(x, y) &= \frac{3\kappa x}{f(1 - (p_kx + q_ky)/f)^{3\kappa+1}}\,, \\ N(x, y) &= \frac{3\kappa y}{f(1 - (p_kx + q_ky)/f)^{3\kappa+1}}\,. \end{aligned} \tag{6.26}$$

Then, the 3D recovery equations (5.2) reduce to a set of linear equations which has the form of (5.5), namely

$$\begin{bmatrix} M_1 - (J_1/J_0)M_0 & N_1 - (J_1/J_0)N_0 \\ M_2 - (J_2/J_0)M_0 & N_2 - (J_2/J_0)N_0 \end{bmatrix} \begin{bmatrix} \delta p_k \\ \delta q_k \end{bmatrix} = -\begin{bmatrix} L_1 - (J_1/J_0)L_0 \\ L_2 - (J_2/J_0)L_0 \end{bmatrix}, \tag{6.27}$$

where

$$L_i = \int_W m_i(x, y)L(x, y)\,dx\,dy\,,$$

$$M_i = \int_W m_i(x, y)M(x, y)\,dx\,dy\,, \tag{6.28}$$

$$N_i = \int_W m_i(x, y)N(x, y)\,dx\,dy\,,$$

for $i = 0, 1, 2$. If $\delta p_k$, $\delta q_k$ are sufficiently close to zero, the $k$th estimate is sufficiently accurate. Otherwise, $p_{k+1} = p_k + \delta p_k$, $q_{k+1} = q_k + \delta q_k$, are better approximations, and the process can be iterated until convergence.

**Remark 6.2.** This method is exactly the *Newton–Raphson iteration* of the 3D recovery equations over two variables $p$, $q$. It is known that if we start from an initial guess which is sufficiently close to the true solution, the iterations converge to it "quadratically" (the error is roughly proportional to the square of the error at the preceding step). However, it is difficult to tell if an estimate is sufficiently close to the true solution, and if we start from an inaccurate guess, the behavior of the iterations is unpredictable. Also, the geometrical meaning of functions $M(x, y)$, $N(x, y)$ is not clear.

On the other hand, the geometrical interpretation of the camera rotation simulation method is very clear: Essentially, we are incrementally "undoing" perspective distortion so that the center of gravity of the texture elements coincides with the center of the window, though it seems difficult to give a mathematical proof of convergence. Moreover, while the Taylor expansion method still involves the approximation (4.6), the camera rotation simulation method is not much affected by this approximation; the texture image is *exactly* transformed (cf. (6.19), (6.20)) repeatedly so that the true surface becomes more and more parallel to the image plane, and in the limit of $p \to 0$, and $q \to 0$, the approximation (4.6) reduces to a trivial identity.

## 7. Curved Surface Recovery from Texture

Equation (5.5) can be applied even if the surface is not planar; it can be applied if we choose windows of an appropriate size so that the surface is approximately planar in each window. Then, (5.5) gives the gradient $(p, q)$ for each window, and hence the global 3D shape of an arbitrary smooth surface can be recovered in principle. Another approach is to use a global surface model. Assume, for example, that the surface is modeled by a quadric surface

$$Z = r + pX + qY + \alpha X^2 + 2\beta XY + \gamma Y^2\,. \tag{7.1}$$

### 7.1. Basic equations for quadric surfaces

If the surface is planar, the observed texture inhomogeneity is solely due to the perspective distortion, while if the surface is curved, the inhomogeneity is due to two separate sources—the perspective distortion and the varying gradient. Inhomogeneity is not observed for planar surface if the projection is orthographic (i.e., the limit of $f \to 0$), while for curved surfaces inhomogeneity results even if the projection is orthographic. For this reason, we consider here the case of orthographic projection, which not only simplifies the problem but also reveals the ambiguity of interpretation inherent to it.

If we let $f \to \infty$, equations (3.1) become $x = X$, $y = Y$. If we substitute (7.1) into (3.7) and take the limit $f \to \infty$, we obtain

$$\sqrt{\det(g)} = \sqrt{1+p^2+q^2}\sqrt{1+A_1x+A_2y+A_3x^2+2A_4xy+A_5y^2}\,, \tag{7.2}$$

where

$$A_1 = \frac{4(\alpha p+\beta q)}{1+p^2+q^2}\,, \quad A_2 = \frac{4(\beta p+\gamma q)}{1+p^2+q^2}\,,$$
$$A_3 = \frac{4(\alpha^2+\beta^2)}{1+p^2+q^2}\,, \quad A_4 = \frac{4\beta(\alpha+\gamma)}{1+p^2+q^2}\,, \quad A_5 = \frac{4(\beta^2+\gamma^2)}{1+p^2+q^2}\,. \tag{7.3}$$

Since $\sqrt{1+p^2+q^2}$ drops off when the unknown texture density $c$ is eliminated, the parameters we can determine are $A_i$, $i = 1, \ldots, 5$. Let us call these the *texture density parameters*; they describe the inhomogeneity of the observed texture density. As we will discuss later, the issue of the ambiguity of interpretation arises here: *We cannot distinguish surfaces which have the same values of the texture density parameters* $A_i$, $i = 1, \ldots, 5$.

If we provide six test functions $m_0(x, y), \ldots, m_5(x, y)$ and compute as "observables"

$$J_i = \int_W \rho(x, y) m_i(x, y)\, dx\, dy\,, \quad i = 0, 1, \ldots, 5\,, \tag{7.4}$$

the 3D recovery equations (4.12) becomes as follows:

**Proposition 7.1** (3D recovery equations). *The texture density parameters are determined by solving*

$$\int_W \left(m_i(x, y) - \left(\frac{J_i}{J_0}\right) m_0(x, y)\right) \times (\sqrt{1+A_1x+A_2y+A_3x^2+2A_4xy+A_5y^2})^{\kappa}\, dx\, dy = 0\,, \tag{7.5}$$

*for* $i = 1, \ldots, 5$.

Thus, six test functions are enough. Of course, we can use many more test functions and determine the parameters, say, by the least squares method.

### 7.2. Surface interpretation

Once the texture density parameters $A_i$, $i = 1, \ldots, 5$, are obtained, the surface shape parameters $p$, $q$, $\alpha$, $\beta$, $\gamma$ are determined by solving (7.3). The solution is given as follows (see Appendix B for proof).

**Proposition 7.2** (Surface interpretation). *Let*

$$\tau \equiv \pm \sqrt[4]{A_3 + A_5 \pm 2\sqrt{A_3 A_5 - A_4^2}}\,. \tag{7.6}$$

*Four values result depending on the choice of the two ± signs. For each value, compute*

$$\alpha' = \tau + \frac{A_3 - A_5}{16\tau}\,, \qquad \beta' = \frac{A_4}{8\tau}\,, \qquad \gamma' = \tau - \frac{A_3 - A_5}{16\tau}\,. \tag{7.7}$$

*Also compute*

$$\begin{aligned} p' &= \frac{\gamma' A_1 - \beta' A_2}{4(\alpha'\gamma' - \beta'^2)}\,, \qquad q' = \frac{-\beta' A_1 + \alpha' A_2}{4(\alpha'\gamma' - \beta'^2)}\,, \\ k &= \frac{1}{\sqrt{1 - p'^2 - q'^2}}\,. \end{aligned} \tag{7.8}$$

*Then, the surface shape parameters $p$, $q$, $\alpha$, $\beta$, $\gamma$ are given by*

$$p = kp'\,, \quad q = kq'\,, \quad \alpha = k\alpha'\,, \quad \beta = k\beta'\,, \quad \gamma = k\gamma'\,. \tag{7.9}$$

### 7.3. Ambiguity of interpretation

From Proposition 7.2 given above, we find some important observations. First, as is obvious from the fact that the projection is orthographic, we can immediately see that:

**Observation 1.** *The distance r along the Z-axis of the surface from the image plane cannot be determined.*

This is because the texture density parameters $A_i$, $i = 1, \ldots, 5$, do not contain parameter $r$. Next, we find that Proposition 7.2 gives four solutions. The first ambiguity occurs due to the following obvious fact (Fig. 4).

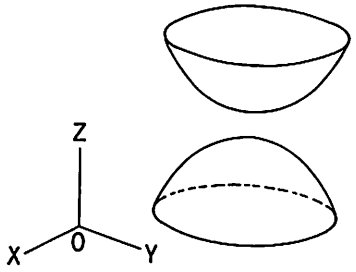



Fig. 4. Two surfaces which are "mirror images" of each other with respect to a "mirror" perpendicular to the $Z$-axis cannot be distinguished under orthographic projection.

**Observation 2.** *Two surfaces which are "mirror images" with respect to a "mirror" perpendicular to the Z-axis cannot be distinguished.*

This is because the texture density parameters $A_i$, $i = 1, \ldots, 5$, have numerators and denominators which are quadratic in the surface shape parameters $p$, $q$, $\alpha$, $\beta$, $\gamma$. Hence, the texture density parameters $A_i$, $i = 1, \ldots, 5$, are the same if $p$, $q$, $\alpha$, $\beta$, $\gamma$ are respectively replaced by $-p$, $-q$, $-\alpha$, $-\beta$, $-\gamma$. The other ambiguity is described as follows (Fig. 5).

**Observation 3.** *An elliptic surface cannot be distinguished for a hyperbolic surface having principal curvatures of the same absolute values.*

**Proof.** Since the projection is orthographic, the relationship between the surface and the image plane is essentially the same if the surface is translated in space parallel to the image plane or rotated around the axis perpendicular to the image plane (or equivalently the $XYZ$-coordinate system is translated or rotated). Suppose that the surface is not a plane, i.e., $\alpha$, $\beta$, $\gamma$ are not zero at the same time. Then, as is well known in linear algebra, if we appropriately

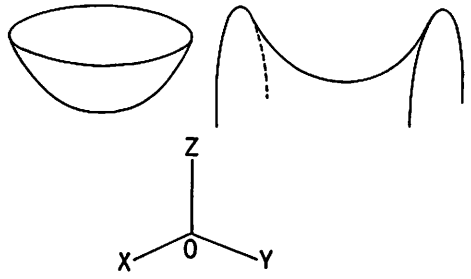



Fig. 5. An elliptic surface cannot be distinguished from a hyperbolic surface having principal curvatures of the same absolute value.

translate and rotate the surface, we can make the equation in either of the following forms:

$$Z = r + \alpha X^2 + \gamma Y^2, \qquad \alpha, \gamma \neq 0, \tag{7.10}$$

for which equations (7.3) become

$$A_1 = 0, \quad A_2 = 0, \quad A_3 = 4\alpha^2, \quad A_4 = 0, \quad A_5 = 4\gamma^2, \tag{7.11}$$

or

$$Z = r + pX + \gamma Y^2, \qquad \gamma \neq 0, \tag{7.12}$$

for which equations (7.3) become

$$\begin{aligned} &A_1 = 0, \quad A_2 = 0, \quad A_3 = 0, \quad A_4 = 0, \\ &A_5 = 4\gamma^2/(1 + p^2). \end{aligned} \tag{7.13}$$

Hence, changing sign of $\alpha$ or $\gamma$ in (7.11) does not affect the texture density parameters $A_i$, $i = 1, \ldots, 5$. □

An intuitive interpretation of this fact is to say that the texture density tells only about the "angle" of the surface orientation (i.e., the *slant*) and nothing about the "orientation" of inclination (i.e, the *tilt*).

Thus, we can understand why there exist in general four interpretations. However, there are two exceptional cases where infinitely many interpretation are possible. These two cases correspond to the "singularities" of the solution given in Proposition 7.2. First, note that in Proposition 7.2, $\tau$ appears in denominator. If $\tau$ happens to be zero, the following ambiguity occurs.

**Observation 4.** *If the surface is hyperbolic with mean curvature zero, the principal directions of the surface are indeterminate.*

**Proof.** As discussed above, we can assume without losing generality that the equation of the surface is given by either (7.10) or (7.12). In the former case, we have $\tau = \pm\frac{1}{2}(\alpha \pm \gamma)$, while in the latter case, we have $\tau = \pm\gamma/2(1 + p^2)^{1/2}$ ($\neq 0$). Hence, $\tau = 0$ occurs only when $\alpha = \pm\gamma$ in the former case. In other words, the equation of the surface is either $Z = r + \alpha(X^2 + Y^2)$ or $Z = r + \alpha(X^2 - Y^2)$. According to Observation 3, these two surfaces "look" exactly the same. The first one is a *circular paraboloid* and is symmetric around the $Z$-axis. Consequently, the same applies to the second one, and the surface "looks" the same even if rotated by an arbitrary angle around the $Z$-axis, resulting in infinitely many interpretations. □

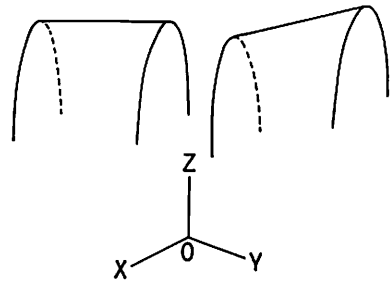



Fig. 6. If the surface is parabolic, the depth gradient along the asymptotic direction is indeterminate.

In (7.8), $\alpha'\gamma' - \beta'^2$ appears in denominator. If $\alpha'\gamma' - \beta'^2$ happens to be zero, the following ambiguity occurs (Fig. 6).

**Observation 5.** *If the surface is parabolic, the depth gradient of the asymptotic direction is indeterminate.*

**Proof.** If $\alpha'\gamma' - \beta'^2 = 0$ the values of $p'$ and $q'$ are indeterminate. However, the ratio $\alpha : \beta : \gamma$ is equal to the ratio $\alpha' : \beta' : \gamma'$, and hence $\alpha\gamma - \beta^2 = 0$. This means that the *Hessian* of the surface is zero, so that one of the principal curvatures is zero. Hence, after appropriate translation and rotation, the equation of the surface is given by (7.12). From (7.13), we see that there exist infinitely many solutions which share the same *asymptotic direction* or "ridge"; the depth gradient $p$ along it is indeterminate, and infinitely many interpretations are possible. □

This result is easily understood if we recall the fact that the gradient of a planar surface, for which the depth changes linearly in all directions, is indeterminate under orthographic projection.

## 8. Numerical Scheme of Curved Surface Recovery

Now, we present a numerical scheme to solve the 3D recovery equations (7.5). Since they are nonlinear equations, iterations are necessary.

### 8.1. First-order approximation

If the window $W$ is small and located at the center of the image, Taylor expansion in $x$, $y$ yields

$$(\sqrt{1 + A_1 x + A_2 y + A_3 x^2 + 2A_4 xy + A_5 y^2})^{\kappa}$$
$$= 1 + Ax + By + Cx^2 + Dxy + Ey^2 + \cdots, \tag{8.1}$$

where

$$A = \tfrac{1}{2}\kappa A_1\,, \qquad B = \tfrac{1}{2}\kappa A_2\,,$$
$$C = \tfrac{1}{2}\kappa(A_3 + \tfrac{1}{4}(k-2)\,A_1^2)\,, \quad D = \kappa(A_4 + \tfrac{1}{4}(\kappa-2)\,A_1A_2)\,, \tag{8.2}$$
$$E = \tfrac{1}{2}\kappa(A_5 + \tfrac{1}{4}(\kappa-2)\,A_2^2)\,.$$

If we put

$$L_{i0} = \int_W m_i(x, y)\,dx\,dy\,, \qquad L_{i1} = \int_W xm_i(x, y)\,dx\,dy\,,$$
$$L_{i2} = \int_W ym_i(x, y)\,dx\,dy\,, \qquad L_{i3} = \int_W x^2m_i(x, y)\,dx\,dy\,, \tag{8.3}$$
$$L_{i4} = \int_W xym_i(x, y)\,dx\,dy\,, \qquad L_{i5} = \int_W y^2m_i(x, y)\,dx\,dy\,,$$

for $i = 0, 1, \ldots, 5$, and neglect higher-order terms, the 3D recovery equations (7.5) become

$$[L_{ij} - (J_i/J_0)L_{0j}]\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = -[L_{i0} - (J_i/J_0)L_{00}]\,. \tag{8.4}$$

Once $A$, $B$, $C$, $D$, $E$ are obtained by solving this equation, the texture density parameters $A_i$, $i = 1, \ldots, 5$, are determined from (8.2) as follows:

$$A_1 = \frac{2}{\kappa}\,A\,, \qquad A_2 = \frac{2}{\kappa}\,B\,,$$
$$A_3 = \frac{2}{\kappa}\,C - \frac{\kappa-2}{\kappa^2}\,A^2\,, \quad A_4 = \frac{1}{\kappa}\,D - \frac{\kappa-2}{\kappa^2}\,AB\,, \tag{8.5}$$
$$A_5 = \frac{2}{\kappa}\,E - \frac{\kappa-2}{\kappa^2}\,B^2\,.$$

### 8.2. Iterative scheme by Taylor expansion

The iterative scheme by Taylor expansion described in Section 6.4 can be applied to this case as well. Let $A_i^{(k)}$, $i = 1, \ldots, 5$, be the $k$th estimates of the texture density parameters. (The initial guess is given, say, by the first-order approximation described above.) The Taylor expansion at these estimate values yields

$$(\sqrt{1 + A_1 x + A_2 y + A_3 x^2 + 2A_4 xy + A_5 y^2})^\kappa$$

$$= M_0(x, y) + \sum_{i=1}^{5} M_i(x, y)\delta A_i^{(k)} + \cdots , \tag{8.6}$$

where $\delta A_i^{(k)} = A_i - A_i^{(k)}$, $i = 1, \ldots, 5$, and

$$\begin{aligned}
M_0 &= (\sqrt{1 + A_1^{(k)} x + A_2^{(k)} y + A_3^{(k)} + 2A_4^{(k)} xy + A_5^{(k)} y^2})^\kappa , \\
M_1 &= \tfrac{1}{2}\kappa \frac{x}{(\sqrt{1 + A_1^{(k)} x + A_2^{(k)} y + A_3^{(k)} x^2 + 2A_4^{(k)} xy + A_5^{(k)} y^2})^{2-\kappa}} , \\
M_2 &= \tfrac{1}{2}\kappa \frac{y}{(\sqrt{1 + A_1^{(k)} x + A_2^{(k)} y + A_3^{(k)} x^2 + 2A_4^{(k)} xy + A_5^{(k)} y^2})^{2-\kappa}} , \\
M_3 &= \tfrac{1}{2}\kappa \frac{x^2}{(\sqrt{1 - A_1^{(k)} x + A_2^{(k)} y + A_3^{(k)} x^2 + 2A_4^{(k)} xy + A_5^{(k)} y^2})^{2-\kappa}} , \\
M_4 &= \tfrac{1}{2}\kappa \frac{2xy}{(\sqrt{1 + A_1^{(k)} x + A_2^{(k)} y + A_3^{(k)} x^2 + 2A_4^{(k)} xy + A_5^{(k)} y^2})^{2-\kappa}} , \\
M_5 &= \tfrac{1}{2}\kappa \frac{y^2}{(\sqrt{1 + A_1^{(k)} x + A_2^{(k)} y + A_3^{(k)} x^2 + 2A_4^{(k)} xy + A_5^{(k)} y^2})^{2-\kappa}} .
\end{aligned} \tag{8.7}$$

If we use as the test functions these $M_i(x, y)$ themselves and put

$$J_i = \int_W \rho(x, y) M_i(x, y)\, dx\, dy , \qquad i = 0, 1, \ldots, 5 , \tag{8.8}$$

$$M_{ij} = \int_W M_i(x, y) M_j(x, y)\, dx\, dy , \qquad i = 0, 1, \ldots, 5 , \tag{8.9}$$

the 3D recovery equations (7.5) become

$$[M_{ij} - (J_i/J_0)M_{0j}][\delta A_i^{(k)}] = -[M_{i0} - (J_i/J_0)M_{00}] . \tag{8.10}$$

If all $A_i^{(k)}$, $i = 1, \ldots, 5$, are sufficiently close to zero, the $k$th estimates are sufficiently accurate. Otherwise, $A_i^{(k+1)} = A_i^{(k)} + \delta A_i^{(k)}$ are better approximations, and the process is repeated until convergence.

## 9. Examples for Synthetic Images

Figures 7–10 are synthetic images of textured planar surfaces. The focal length $f$ is taken as the unit of length, and the window size is $a = b = f \tan 10° = 0.176$.

(The actual window is a rectangle of size $2a \times 2b$.) The $Z$-axis, i.e., the camera optical axis, is assumed to pass through the center of the window. The true gradient is $(p, q) = (1.500, 0.866)$ for all the figures.

### 9.1. Regularly aligned dot textures

Figures 7(a)–(c) show projected images of regularly aligned dot textures on a planar surface. Using $m_0(x, y) = 1$, $m_1(x, y) = x$, $m_2(x, y) = y$ as the test functions, we first obtain the initial estimate by computing the center of gravity. Then, we iteratively apply the camera rotation simulation method. The successively computed values of $(p, q)$ are as follows:

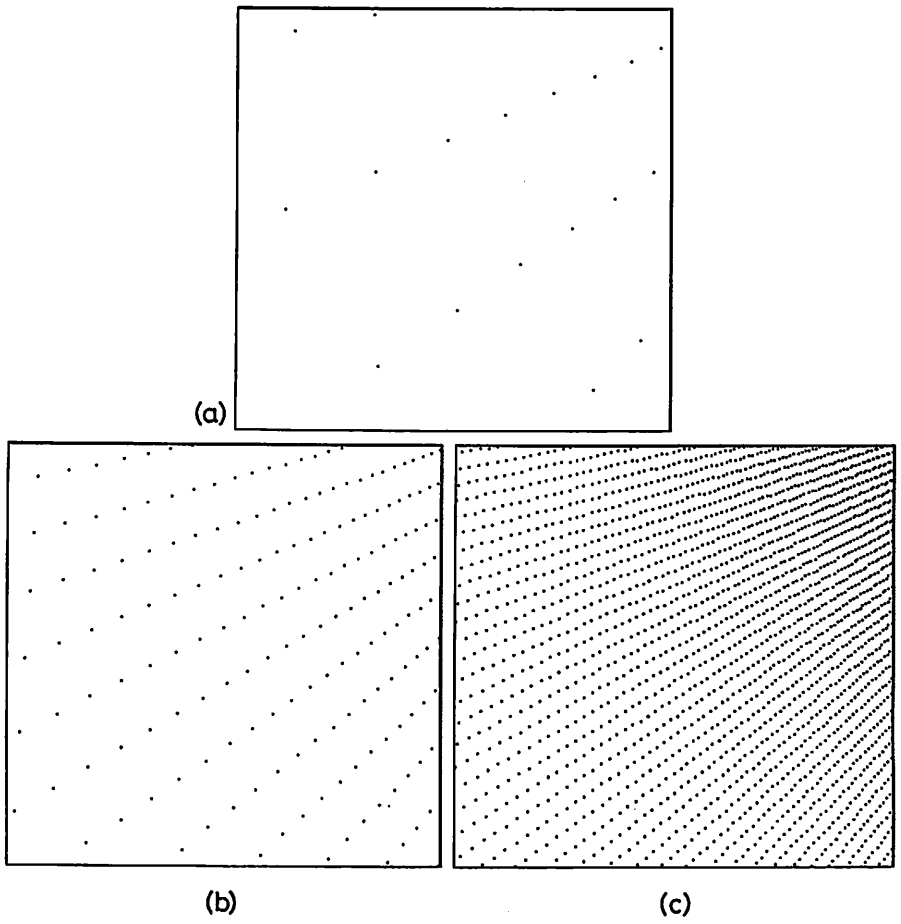



Fig. 7. Regularly aligned dot textures on a planar surface.

| | Fig. 7(a) | Fig. 7(b) | Fig. 7(c) |
|---|---|---|---|
| 1 | (1.459, 1.088) | (1.610, 0.965) | (1.548, 0.958) |
| 2 | (1.402, 0.939) | (1.552, 0.875) | (1.499, 0.871) |
| 3 | (1.397, 0.937) | (1.548, 0.871) | (1.496, 0.868) |
| 4 | (1.397, 0.937) | (1.548, 0.871) | (1.496, 0.867) |

If we apply the Taylor expansion method to the same initial estimates with $m_0(x, y) = L(x, y)$, $m_1(x, y) = M(x, y)$, $m_2(x, y) = N(x, y)$, we obtain:

| | Fig. 7(a) | Fig. 7(b) | Fig. 7(c) |
|---|---|---|---|
| 1 | (1.459, 1.088) | (1.610, 0.965) | (1.548, 0.958) |
| 2 | (1.549, 0.767) | (1.549, 0.897) | (1.499, 0.869) |
| 3 | (1.527, 0.816) | (1.542, 0.887) | (1.496, 0.864) |
| 4 | (1.529, 0.807) | (1.541, 0.887) | (1.496, 0.864) |

### 9.2. Random dot textures

Figures 8(a)–(c) show random dot textures on a planar surface. In the following, we apply the procedures described earlier. For the camera rotation simulation method, we obtain:

| | Fig. 8(a) | Fig. 8(b) | Fig. 8(c) |
|---|---|---|---|
| 1 | (1.558, 0.798) | (1.662, 0.945) | (1.535, 0.974) |
| 2 | (1.505, 0.695) | (1.617, 0.843) | (1.493, 0.891) |
| 3 | (1.504, 0.694) | (1.613, 0.840) | (1.491, 0.888) |
| 4 | (1.504, 0.694) | (1.613, 0.840) | (1.491, 0.888) |

For the Taylor expansion method, we obtain:

| | Fig. 8(a) | Fig. 8(b) | Fig. 8(c) |
|---|---|---|---|
| 1 | (1.558, 0.798) | (1.662, 0.945) | (1.535, 0.974) |
| 2 | (1.422, 0.560) | (1.674, 0.832) | (1.527, 0.899) |
| 3 | (1.434, 0.583) | (1.667, 0.834) | (1.525, 0.897) |
| 4 | (1.430, 0.578) | (1.667, 0.834) | (1.525, 0.897) |

### 9.3. Regularly aligned line segment textures

Figures 9(a)–(c) show regularly aligned line segment textures on a planar surface. For the camera rotation simulation method, we obtain:

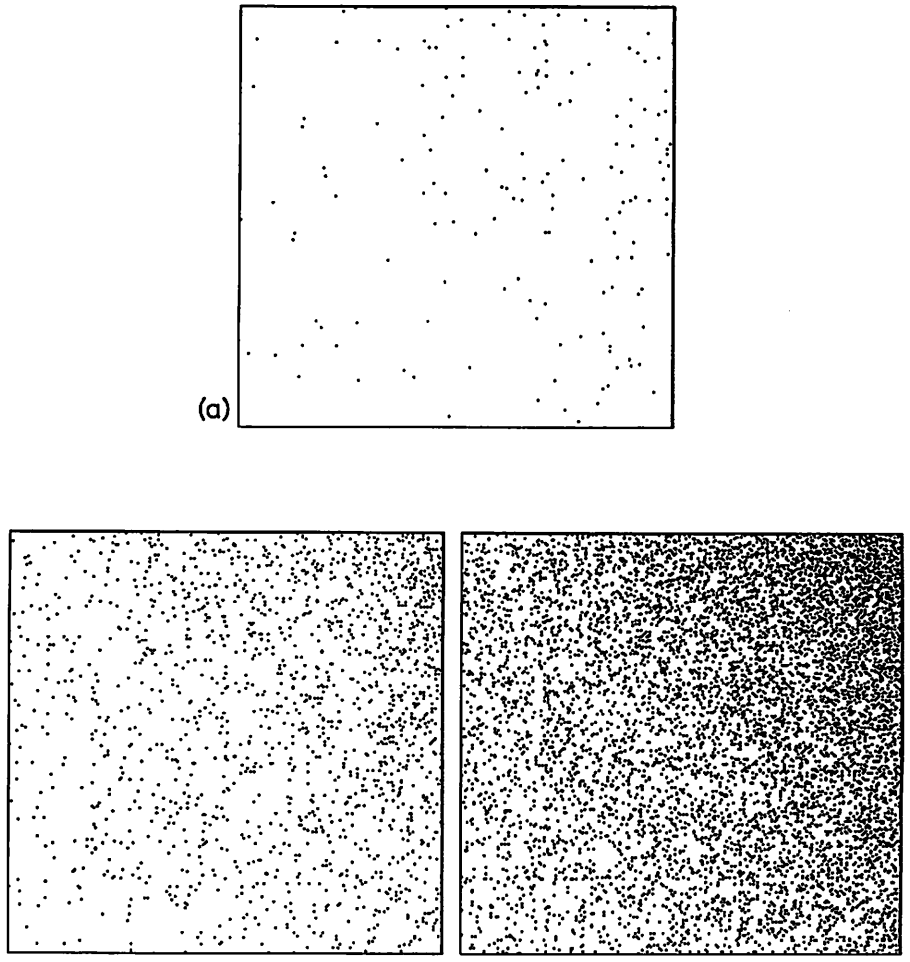



Fig. 8. Random dot textures on a planar surface.

| | Fig. 9(a) | Fig. 9(b) | Fig. 9(c) |
|---|---|---|---|
| 1 | (1.603, 0.994) | (1.762, 1.048) | (1.751, 1.045) |
| 2 | (1.379, 0.747) | (1.534, 0.899) | (1.527, 0.893) |
| 3 | (1.352, 0.756) | (1.505, 0.875) | (1.500, 0.869) |
| 4 | (1.351, 0.758) | (1.504, 0.874) | (1.499, 0.869) |

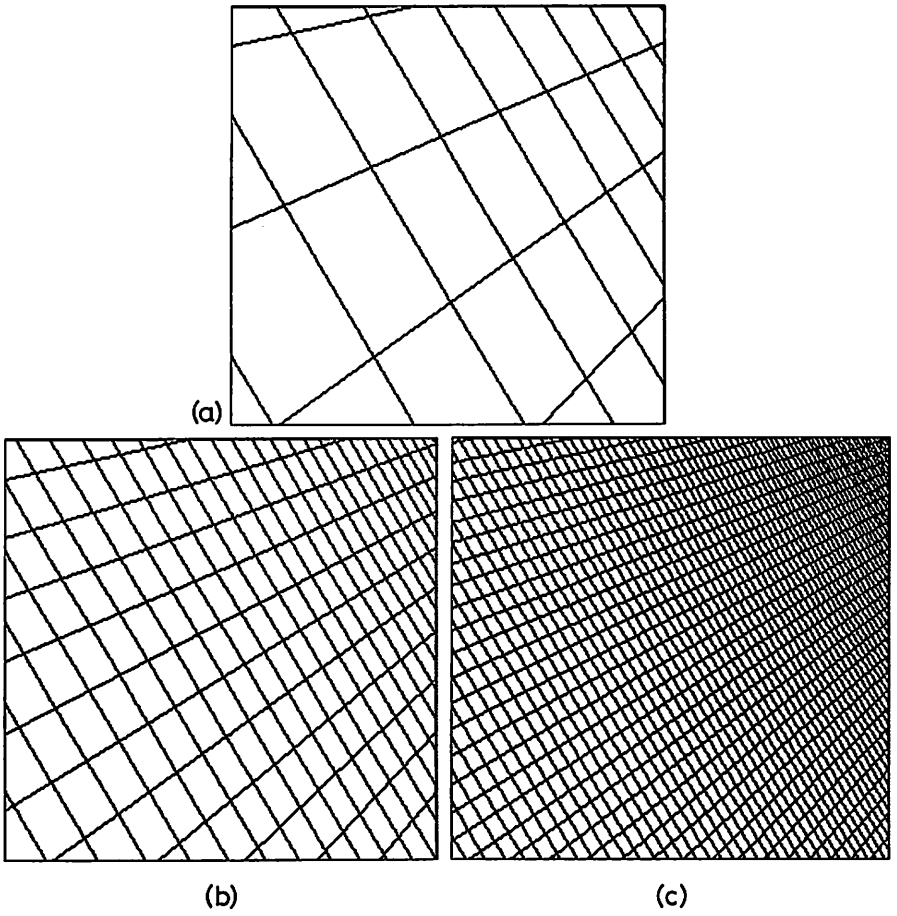


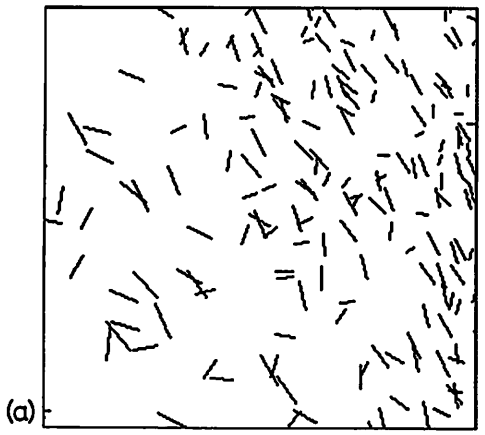

Fig. 9. Regularly aligned line segment textures on a planar surface.

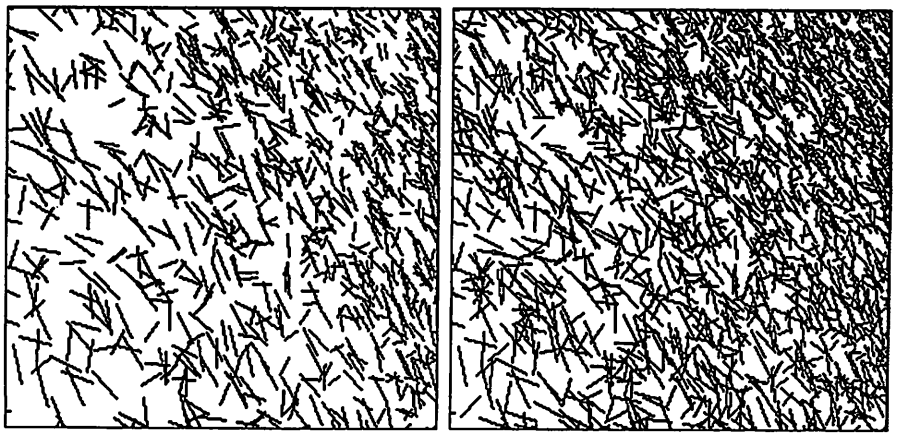
For the Taylor expansion method, we obtain:

| | Fig. 9(a) | Fig. 9(b) | Fig. 9(c) |
|---|---|---|---|
| 1 | (1.603, 0.994) | (1.762, 1.048) | (1.751, 1.045) |
| 2 | (1.569, 0.893) | (1.672, 0.960) | (1.666, 0.958) |
| 3 | (1.568, 0.893) | (1.669, 0.956) | (1.663, 0.954) |
| 4 | (1.568, 0.893) | (1.669, 0.956) | (1.663, 0.954) |

### 9.4. Random line segment textures

Figures 10(a)–(c) show random line segment textures on a planar surface. For the camera rotation simulation method, we obtain:



(b) (c)

Fig. 10. Random line segment textures on a planar surface.

| | Fig. 10(a) | Fig. 10(b) | Fig. 10(c) |
|---|---|---|---|
| 1 | (2.821, 0.602) | (2.196, 0.768) | (2.006, 0.910) |
| 2 | (2.275, 0.372) | (1.906, 0.590) | (1.710, 0.722) |
| 3 | (2.088, 0.271) | (1.856, 0.559) | (1.660, 0.684) |
| 4 | (2.111, 0.275) | (1.856, 0.559) | (1.661, 0.684) |

For the Taylor expansion method, we obtain:

| | Fig. 10(a) | Fig. 10(b) | Fig. 10(c) |
|---|---|---|---|
| 1 | (2.821, 0.602) | (2.196, 0.768) | (2.006, 0.910) |
| 2 | (2.477, 0.238) | (2.033, 0.561) | (1.825, 0.787) |
| 3 | (2.447, 0.257) | (2.022, 0.567) | (1.820, 0.781) |
| 5 | (2.445, 0.258) | (2.022, 0.567) | (1.820, 0.781) |

### 9.5. Texture on a curved surface

Figure 11 is an orthographic view of a regularly aligned dot texture on a quadric surface. The window size $a$ $(=b)$ is taken as the unit of length. The true surface shape parameters are $(\alpha, \beta, \gamma) = (2, 0, 0)$. As discussed in Section 7, parameters $p$, $q$ are indeterminate in this case. However, parameters $\alpha$, $\beta$, $\gamma$ are uniquely determined except for sign. If we use the method of Section 8 with 1, $x$, $y$, $x^2$, $xy$, $y^2$ as the initial test functions, the successive estimates of $(\alpha, \beta, \gamma)$ become as follows

| | Fig. 11 |
|---|---|
| 1 | (0.494, 0.000, −0.007) |
| 2 | (1.066, −0.000, −0.009) |
| 3 | (1.632, −0.001, −0.008) |
| 4 | (1.929, −0.001, −0.006) |
| 5 | (1.991, −0.001, −0.006) |

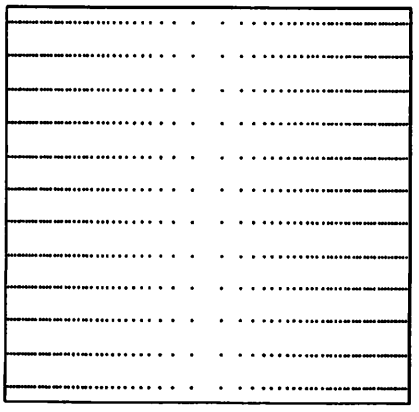


Fig. 11. An orthographic view of a regularly aligned dot texture on a curved surface.

### 9.6. Observations

We see that our method can produce fairly good results. In particular, our method can be applied to very sparsely distributed textures. For very sparse textures, our results show that the Taylor expansion method gives more accurate values than the camera rotation simulation method. Otherwise, both of them yield almost the same results for dot textures, but for line segment textures, the camera rotation simulation method predicts more accurate values, as is expected from Remark 6.2. The convergence is very rapid for both methods; only two or three iterations are necessary to determine the gradient values up to two or three decimal places.

On the whole, the results are better for dot textures than for line segment textures. This is easily understood because a line segment is a restricted coalescence of constituent points, and hence the degree of homogeneity is lower for line segment textures. On the other hand, the results are far better for regular textures than random ones, as is also expected from Remark 5.4. This is not a drawback; natural or man-made textures are often "tessellated" to a high degree of regularity.

## 10. Concluding Remarks

### 10.1. Texture density as a functional

Our formulation makes use of the *exact* texture density $\rho(x, y)$ describing the discrete distribution of texture elements, so that no *ad hoc smoothing is necessary*. This is possible because we do not use particular *values* of the texture density $\rho(x, y)$; all we need is the *rule of integation*. Hence, the texture density $\rho(x, y)$ is defined as a *functional*, or a *distribution* in the sense of Schwartz [17, 18]. This is one of the most important differences from all existing approaches.

Many works on shape from texture assume existence of a smooth texture density $\rho(x, y)$ obtained by some kind of local averaging, directly using the values of $\rho(x, y)$ (cf. the method of Aloimonos and Swain [1] recast in Appendix C). There are some methods which even require the *derivatives* of the texture density $\rho(x, y)$ computed by numerical differentiation. The use of the values or derivatives of the texture density $\rho(x, y)$ does not seem appropriate in view of the discrete nature of the texture.

### 10.2. Differential geometry in terms of image coordinates

We derived the exact relationship between the surface texture density and the observed texture density by *differential geometry*. The existing methods seem to have failed to obtain this exact relationship. One reason, among others, seems to be that many authors employed some "surface coordinate system"

placed on the surface, and tried to obtain the rule of transformation from the image coordinates to the surface coordinates. The Jacobian of this transformation should determine the observed texture density (cf. Aloimonos and Swain [1], Dunn [5]). However, this is usually a tedious process. The key to success here is the fact that *we do away with the surface coordinates*; all surface characteristics are described in terms of the image coordinates through the *first fundamental form*.

### 10.3. Integrations as observables

One big advantage resulting from the use of integrations as *observables* is that the method works for very "sparse" textures. In the extreme, the texture can consist of only a *single dot*; if the dot is at the center of the image plane, the gradient is predicted to be zero, and otherwise the plane is predicted to be slanted in the direction of the displacement of the dot from the image origin. None of the existing methods have this property, because these methods make use of "local clues", rather than "global clues".

A typical approach is to choose several "small but finite regions"—let us call these *test regions*—on the image plane, and compute the number or length of texture elements in those test regions. The test regions must not be too large, since variations of the texture density could not be detected, but at the same time the test regions must not be too small, since the result would be unreliable due to fluctuations. (There may exist no texture elements in a test region if the texture is very sparse.) For those methods, the texture must be *locally homogeneous* in the sense that the texture is dense enough everywhere so that the homogeneity condition is satisfied in each of the test regions.

While there is no limit on the texture sparsity, the estimate of our method approaches the numerically *exact* value as the texture density increases. No methods so far known seem to have this property.

### 10.4. Computational efficiency

Since the necessary data, or *observables*, are obtained by integration of functions over the image, and since integration is essentially summation, the time complexity is simple $O(N)$, where $N$ is the number of texture elements. Although iterations are used, the convergence is very rapid; two or three iterations seem sufficient.

The computational process of our method is completely *non-intelligent*; the computer need not recognize the "structures" of the texture such as the shapes of the texture elements (circles, ellipses, etc.), the regularity and periodicity of their alignment, and the ratio of convergence of the sizes of the texture elements or the intervals between them. Such structure-based approaches require "intelligent" heuristic programming for detecting structures in the texture.

Still, the computational burden may increase as the number of texture elements increases. Since most of the computation time is consumed in numerical evaluation of integrands, high-speed numerical processors are necessary. On the other hand, methods based on a small number of test regions such as that of Aloimonos and Swain [1] may be faster, since these methods use only partial information. Thus, there is a trade-off between efficiency and accuracy.

Since the access to each texture element is an independent process, high speed performance is expected by parallel architecture; the image can be divided in any way, and the computation can be performed independently and simultaneously.

### 10.5. Preprocessing

We should not forget the fact that appropriate preprocessing is necessary for our method. In our method, a texture is assumed to be composed of dots "without area" and line segments "without width". If the dots have area, their centroids can be used as their positions, or their boundaries can be regarded as texture elements. If the line segments have width, their center lines ("skeletons" or "medial axes") or boundaries can be regarded as texture elements. Our method is essentially (weighted) *number counting* of dots and (weighted) *length measuring* of line segments. For natural texture images containing gray levels, a simple way may be just to apply edge detection. The detected edges would serve as line segment texture elements.

### 10.6. Resolution threshold and subtexture

One of the difficulties for *any* type of shape from texture analysis is the effect of *resolution threshold*: The part of the texture far away from the viewer becomes dense on the image plane, and hence its spatial frequency becomes high. As a result, variations cannot be detected above a certain spatial frequency. In contrast, the part near the viewer is likely to be processed with excessive resolution, resulting in *subtexture*—variations of gray levels within individual texture elements. Hence, all texture elements may not be detected on the dense part, and subtexture may be counted as texture elements, resulting in underestimation of the surface gradient.

Many ways of avoiding this effect are conceivable, although none may be complete. For example, the window may be confined so that a small depth range is observed, or only clearly visible clues could be focused on by lowering the resolution. However, restricting input information in this way may reduce the estimation accuracy. Since appearance of subtexture may be greatly affected by the level of resolution, while the true texture can be stable, the true texture can be picked out by changing the resolution level and choosing only stable patterns. This approach was attempted by Blostein and Ahuja [3].

In any case, whichever approach is used to cope with this effect, our method has the advantage that it does not require local homogeneity and works well even for very sparse textures.

### 10.7. Dimensionality of texture elements

One of the important findings is that dot textures and line segment textures cannot be treated in the same manner: *Pixels constituting line segments cannot be identified with pixels of a dot texture.* The necessity of this distinction does not seem to have been widely recognized. The reason for this can be easily understood if we note the fact that pixels constituting a line segment are necessarily *uniformly sampled* as long as we use a uniform array of pixels for discrete images. If pixels on a line segment could be sampled with converging intervals in such a way that the effect of perspective projection is precisely reflected, no distinction would be needed between dots and line segments. That is why we must introduce into our theory the exponent $\kappa$ to distinguish these two types of texture.

### 10.8. Generality of the principle

The main emphasis in this paper is the generality of our formulation, from which various modifications and applications become possible. We show in Appendices C and D that the methods of Aloimonos and Swain [1] and Dunn [5] can be regarded as special variants of our general principle. While their particular methods have merits in some respects, our theory can make explicit the assumptions and approximations underlying their methods. This wide range of flexibility stems from a mathematically *correct understanding* of the geometry of perspective projection. Particular heuristics or ad hoc assumptions and approximations may result in particular algorithms which may be useful sometimes. Lacking generality, however, these algorithms usually do not reveal the underlying essential nature of the problem and may not be extended to other problems.

## Appendix A. Error Due to Randomness of the Texture

Consider a dot texture for simplicity. Let $(x_1, y_1), \ldots, (x_N, y_N)$ be the coordinates of the texture points in the window $-a < x \leqslant a$, $-b < y \leqslant b$. The center of gravity $(\bar{x}, \bar{y})$ is given by

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i , \qquad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i . \tag{A.1}$$

First, consider the case where the distribution is completely random: $x_i$, $i = 1, \ldots, N$, are random variables chosen from the uniform distribution over $-a < x \leqslant a$ independently of each other, and likewise $y_i$, $i = 1, \ldots, N$, are also

independently chosen from the uniform distribution over $-b < y \leqslant b$. This means

$$E[x_i] = 0\,, \qquad V[x_i] = \tfrac{1}{3}a^2\,, \qquad E[y_i] = 0\,, \qquad V[y_i] = \tfrac{1}{3}b^2\,, \tag{A.2}$$

$i = 1, \ldots, N$, where $E[\cdot]$ and $V[\cdot]$ designate the expectation value and the variance respectively. If follows then that the expectation values and variances of $\bar{f}$, $\bar{y}$ are given by

$$E[\bar{x}] = 0\,, \qquad V[\bar{x}] = \tfrac{1}{3}a^2/N\,, \qquad E[\bar{y}] = 0\,, \qquad V[\bar{y}] = \tfrac{1}{3}b^2/N\,. \tag{A.3}$$

Hence, the center of gravity is at the origin on the average. Since the magnitude of error is estimated by the standard deviation, we expect errors of about $a/\sqrt{3N}$, $b/\sqrt{3N}$ for $\bar{x}$, $\bar{y}$, respectively.

On the other hand, consider the other extreme where $x_i$, $i = 1, \ldots, N$, are distributed at equal intervals of distance $2a/N$, and $y_i$, $i = 1, \ldots, N$, at equal intervals of distance $2b/N$. Then, the center of gravity must be located within the range of

$$-a/N < \bar{x} \leqslant a/N\,, \qquad -b/N < \bar{y} \leqslant b/N\,. \tag{A.4}$$

From (A.3) and (A.4), we can roughly say that the errors $\delta\bar{x}$, $\delta\bar{y}$ of $\bar{x}$, $\bar{y}$, respectively, are

$$\delta\bar{x} = \mathrm{O}(1/N^{\varepsilon})\,, \qquad \delta\bar{y} = \mathrm{O}(1/N^{\varepsilon})\,, \tag{A.5}$$

where $\frac{1}{2} < \varepsilon < 1$. The exponent $\varepsilon$ approaches $\frac{1}{2}$ as the distribution becomes more random, while it approaches 1 as the distribution becomes more regular.

## Appendix B. Interpretation of Curved Surfaces

Define $k$, $p'$, $q'$, $\alpha'$, $\beta'$, $\gamma'$ as follows:

$$k = \sqrt{1 + p^2 + q^2}\,, \qquad p' = p/k\,, \qquad q' = q/k\,, \qquad a' = \alpha/k\,, \qquad \beta' = \beta/k\,, \qquad \gamma' = \gamma/k\,. \tag{B.1}$$

Equations (7.3) are now rewritten as follows:

$$\alpha' p' + \beta' q' = \tfrac{1}{4}A_1\,, \quad \beta' p' + \gamma' q' = \tfrac{1}{4}A_2\,, \quad a'^2 + \beta'^2 = \tfrac{1}{4}A_3\,, \quad \beta'(\alpha' + \gamma') = \tfrac{1}{4}A_4\,, \quad \beta'^2 + \gamma'^2 = \tfrac{1}{4}A_5\,. \tag{B.2}$$

Let us define new variables

$$\tau = \tfrac{1}{2}(\alpha' + \gamma'), \quad \sigma = \tfrac{1}{2}(\alpha' - \gamma') + \mathrm{i}\beta', \tag{B.3}$$

where i is the imaginary unit. Hence, $\sigma$ is a complex number. Next, put

$$T = \tfrac{1}{8}(A_3 + A_5), \qquad S = \tfrac{1}{8}(A_3 - A_5) + \tfrac{1}{4}\mathrm{i}A_4. \tag{B.4}$$

Hence, $S$ is also a complex number. In terms of these new quantities, the last three of (B.2) are equivalent to

$$T = \tau^2 + \sigma\sigma^*, \qquad S = 2\tau\sigma, \tag{B.5a, b}$$

where the asterisk denotes the complex conjugate. From (B.5b), we obtain $\sigma = \frac{1}{2}S\tau$. Substituting this in (B.5a), we find that $\tau$ is the solution of

$$\tau^4 - T\tau^2 + \tfrac{1}{4}SS^* = 0, \tag{B.6}$$

and consequently

$$\tau = \tfrac{1}{2}(T \pm \sqrt{T^2 - SS^*}), \tag{B.7}$$

or in terms of $A_i$, $i = 1, \ldots, 5$,

$$\tau^2 = \tfrac{1}{16}(A_3 + A_5 \pm 2\sqrt{A_3A_5 - A_4^2}). \tag{B.8}$$

Hence, (7.6) is obtained. From (B.3), $\alpha'$, $\beta'$, $\gamma'$ are given by

$$\alpha' = \tau + \mathrm{Re}[\sigma], \qquad \beta' = \mathrm{Im}[\sigma], \qquad \gamma' = \tau - \mathrm{Re}[\sigma], \tag{B.9}$$

where $\mathrm{Re}[\cdot]$ and $\mathrm{Im}[\cdot]$ designate the real and imaginary parts. Substituting $\sigma = \frac{1}{2}S/\tau$ into these, we obtain (7.7). Then, $p'$, $q'$ are determined from the first two of (B.2) in the form of the first two of (7.8). The last of (7.8) is obtained from the relation

$$1 - p'^2 - q'^2 = 1 - \frac{p^2}{1 + p^2 + q^2} - \frac{q^2}{1 + p^2 + q^2} = \frac{1}{1 + p^2 + q^2} = \frac{1}{k^2}. \tag{B.10}$$

Finally, equations (7.9) are obtained from (B.1).

The derivation here is based on the theory of coordinate rotation invariance; variables $\tau$, $\sigma$, $T$, $S$ are *invariants* corresponding to *irreducible representations* of the two-dimensional rotation group SO(2) (cf. Kanatani [12]).

## Appendix C. Method of Aloimonos and Swain

Here, we compare our method with that of Aloimonos and Swain [1]. Since direct comparison is difficult because their derivation is based on different concepts and assumptions, we now newly derive, in our mathematical setting, a scheme which is essentially equivalent to theirs.

Consider three circular regions $S_0$, $S_1$, $S_2$ on the image plane with respective centers $(x_0, y_0)$, $(x_1, y_1)$, $(x_2, y_2)$, and call these regions *test regions*. Assume that these test regions are sufficiently small compared with the size of the window $W$, yet the texture is sufficiently dense, so that each test region contains a sufficiently large number of texture elements. We use, as observables, the integrals over the test regions:

$$J_i = \int_{S_i} \rho(x, y)\, dx\, dy\,, \quad i = 0, 1, 2\,. \tag{C.1}$$

By assumption, each test region $S_i$, $i = 0, 1, 2$, contains a large number of texture elements, and the homogeneity condition is satisfied within each $S_i$. Hence, the above integral can be approximated as follows:

$$J_i \approx c \int_{S_i} (\sqrt{\det(g)})^{\kappa}\, dx\, dy\,, \quad i = 0, 1, 2\,. \tag{C.2}$$

This is equivalent to choosing, as the test functions $m_i(x, y)$, $i = 0, 1, 2$, the characteristic functions $\chi_{S_i}$, $i = 0, 1, 2$ (cf. (2.5)).

Also by assumption, each test region $S_i$, $i = 0, 1, 2$, is very small, so that the integral may be replaced by the area $S_i$ times the value at the center $(x_i, y_i)$ of the test region:

$$\int_{S_i} (\sqrt{\det(g)})^{\kappa}\, dx\, dy \approx S_i (\sqrt{\det(g)})^{\kappa}\big|_{x=x_i, y=y_i}\,. \tag{C.3}$$

Hence, for a planar surface, the observables $J_i$ are approximated by

$$J_i \approx cS_i \frac{(\sqrt{1+p^2+q^2})^{\kappa}(1+r/f)^{2\kappa}}{(1-(px_i+qy_i)/f)^{3\kappa}}\,, \quad i = 0, 1, 2\,. \tag{C.4}$$

From this, we obtain the 3D recovery equations

$$\left(\left(\frac{S_0 J_1}{S_1 J_0}\right)^{1/3\kappa} x_1 - x_0\right) p + \left(\left(\frac{S_0 J_1}{S_1 J_0}\right)^{1/3\kappa} y_1 - y_0\right) q = f\left(\left(\frac{S_0 J_1}{S_1 J_0}\right)^{1/3\kappa} - 1\right), \tag{C.5}$$

$$\left(\left(\frac{S_0 J_2}{S_2 J_0}\right)^{1/3\kappa} x_2 - x_0\right) p + \left(\left(\frac{S_0 J_2}{S_2 J_0}\right)^{1/3\kappa} y_2 - y_0\right) q = f\left(\left(\frac{S_0 J_2}{S_2 J_0}\right)^{1/3\kappa} - 1\right),$$

from which the gradient $(p, q)$ is determined. This is the essential idea of the method of Aloimonos and Swain [1]. They also tried recovery of curved surfaces by similar ideas.

In this method, the assumption of *local homogeneity*—the texture is sufficientyl dense and homogeneous within each test region—plays a fundamental role. Furthermore, a crude approximation like (C.3) is made. This relation holds only when each test region is very small. In our formulation, global *test functions* $m_i(x, y)$ are used instead of local *test regions* $S_i$. Hence, the texture need not be locally homogeneous; it only need be globally homogeneous.

Also, in our method, the integration is exactly performed over all the texture elements. Hence, our method can be applied even to a texture consisting of a single dot, for which the method of Aloimonos and Swain [1] fails. However, it should be pointed out that their method has the advantage that it requires less computation time, because only textures within small test regions are observed.

## Appendix D. Method of Dunn

Dunn [5] proposed another approach. Again, direct comparison is difficult, and we derive its equivalent in our own mathematical framework. Consider a narrow strip $S(h, \theta)$ of a fixed width $\delta$ and a fixed length $l$ along line $x \cos\theta + y \sin\theta = h$. Let us call it a *test strip*. Use the following observables:

$$J(h, \theta) = \int_{S(h,\theta)} \rho(x, y)\, dx\, dy\,. \tag{D.1}$$

This is equivalent to choosing, as the test function, the characteristic function of the test strip.

Take a new $x'y'$-coordinate system by rotating the original $xy$-coordinate system counterclockwise by angle $\theta$ (Fig. D.1). Line $x \cos\theta + y \sin\theta = h$ now becomes $x' = h$ in the new coordinate system. Hence, the observable of (D.1) is rewritten as

$$J(h, \theta) = \int_{-1/2}^{1/2} \int_{h-\delta/2}^{h+\delta/2} \rho(x', y')\, dx'\, dy'\,. \tag{D.2}$$

If there exist a sufficiently large number of texture elements in the test strip $S(h, \theta)$ and the homogeneity condition is satisfied within $S(h, \theta)$, this observable may be approximated by

$$J(h, \theta) \approx c \int_{-l/2}^{l/2} \int_{h-\delta/2}^{h+\delta/2} (\sqrt{\det(g)})^{\kappa}\, dx'\, dy'\,. \tag{D.3}$$

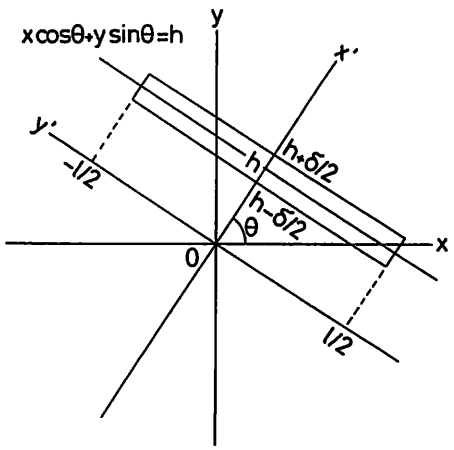



Fig. D.1. A test strip $S(h, \theta)$ with width $\delta$ and length $l$ placed along line $x \cos\theta + y \sin\theta = h$.

If the approximation (5.3) is used in (3.10) for a planar surface, this equation becomes

$$J(h, \theta) \approx c\delta l(\sqrt{1+p^2+q^2})^{\kappa}\left(1+\frac{r}{f}\right)^{2\kappa}\left(1+\frac{3\kappa}{f}(p\cos\theta+q\sin\theta)h\right). \tag{D.4}$$

If, as in the method of Aloimonos and Swain [1], observables $J_i = J(h_i, \theta_i)$, $i = 0, 1, 2$, are computed for three pairs $(h_i, \theta_i)$, $i = 0, 1, 2$, we obtain the 3D recovery equations

$$\begin{aligned}
&\left(h_1\cos\theta_1 - \frac{J_1}{J_0}h_0\cos\theta_0\right)p + \left(h_1\sin\theta_1 - \frac{J_1}{J_0}h_0\sin\theta_0\right)q \\
&\quad = -\frac{f}{3\kappa}\left(1-\frac{J_1}{J_0}\right), \\
&\left(h_0\cos\theta_2 - \frac{J_2}{J_0}h_0\cos\theta_0\right)p + \left(h_2\sin\theta_2 - \frac{J_2}{J_0}h_0\sin\theta_0\right)q \\
&\quad = -\frac{f}{3\kappa}\left(1-\frac{J_2}{J_0}\right),
\end{aligned} \tag{D.5}$$

from which the gradient $(p, q)$ is determined.

Dunn [5], however, took another approach. He searched for $\theta$ such that $J(h, \theta)$ does not depend on $h$. If $\theta_1$ is the one, we see from (D.4) that $\theta_1$ must satisfy

$$p\cos\theta_1 + q\sin\theta_1 = 0, \tag{D.6}$$

and hence

$$J(h, \theta_1) \approx c\delta l(\sqrt{1+p^2+q^2})^{\kappa}\left(1+\frac{r}{f}\right)^{2\kappa} \tag{D.7}$$

is a constant. Let this constant be $J_1$. Next, search for $\theta$ such that $J(h, \theta)$ has the steepest ascent with respect to $h$. If $\theta_2$ is the one, we see from (D.4) that $\theta_2 = \theta_1 \pm \frac{1}{2}\pi$ and that

$$p \cos \theta_2 + q \sin \theta_2 = \sqrt{p^2+q^2}\,, \tag{D.8}$$

and hence

$$J(h, \theta_2) \approx J_1\left(1+\frac{3\kappa}{f}\sqrt{p^2+q^2}h\right). \tag{D.9}$$

The orientation of the gradient $(p, q)$ is given by (D.6), and its magnitude is obtained by computing the (average) gradient of $J(h, \theta_2)/J_1$ with respect to $h$.

Thus, Dunn's method uses (theoretically infinitely) many test strips rather than only three, and the gradient $(p, q)$ is obtained by parametric fitting. In this sense, his method accounts for all the texture elements in contrast to the method of Aloimonos and Swain [1]. For this reason, this method is expected to be more robust at the expense of more computation time. Dunn [5] also tried recovery of curved surfaces by similar ideas.

However, the underlying approximation is essentially the small gradient approximation. Although the integration is performed exactly up to the first order (equation (D.4)), the texture is assumed to be locally homogeneous (though mildly compared with the method of Aloimonos and Swain [1]), and the homogeneity condition is assumed to be satisfied in each test strip.

## ACKNOWLEDGMENT

This work was done while Ken-ichi Kanatani was staying at the Center for Automation Research, University of Maryland in 1985–1986, in collaboration with Tsai-Chia Chou, and supported in part by the Defense Advanced Research Projects Agency and the U.S. Army Night Vision and Electro-Optics Laboratory under Contract DAAB07-86-K-F073. The authors thank Azriel Rosenfeld and Larry Davis for many comments. They also thank Thomas Seidman, Stanley Dunn, John Aloimonos, and Narendra Ahuja for helpful discussions.

## REFERENCES

1. Aloimonos, J. and Swain, M.J., Shape from texture, in: *Proceedings IJCAI-85*, Los Angeles, CA (1985) 926–931.
2. Bajcsy, R., and Lieberman, L., Texture gradient as a depth cue, *Comput. Graph. Image Process.* **5** (1976) 52–67.
3. Blostein, D. and Ahuja, N., Representation and three-dimensional interpretation of image texture: An integrated approach, in: *Proceedings International Conference on Computer Vision*, London (1987) 444–449.

4. Davis, L.S., Janos, L. and Dunn, S.M., Efficient recovery of shape from texture, *IEEE Trans. Pattern Anal. Mach. Intell.* **5** (1983) 485–492.
5. Dunn, S.M., Recovering the orientation of textured surfaces, Ph.D. Thesis, Center for Automation Research, University of Maryland, College Park, MD (1986).
6. Gibson, J.J., *The Perception of the Visual World* (Houghton Mifflin, Boston, MA, 1950).
7. Gibson, J.J., *The Scenes Considered as Perceptual Systems* (Houghton Mifflin, Boston, MA, 1966).
8. Gibson, J.J., *The Ecological Approach to Visual Perception* (Houghton Mifflin, Boston, MA, 1979).
9. Ikeuchi, K., Shape from regular patterns: An example of constraint propagation in vision, MIT AI Memo 567, Cambridge, MA (1980).
10. Kanade, T., Recovery of the three-dimensional shape of an object from a single view, *Artificial Intelligence* **17** (1981) 409–460.
11. Kanatani, K., Detection of surface orientation and motion from texture by a stereological technique, *Artificial Intelligence* **23** (1984) 213–237.
12. Kanatani, K., Coordinate rotation invariance of image characteristics for 3D shape and motion recovery, in: *Proceedings International Conference on Computer Vision*, London (1987) 55–64.
13. Kender, J.R., Surface constraints from linear extents, in: *Proceedings AAAI-83*, Washington, DC (1983) 187–190.
14. Nakatani, H., Kimura, S., Saito, O. and Kitahashi, T., Extraction of vanishing point and its application to scene analysis based on image sequence, in: *Proceedings International Conference on Pattern Recognition*, Miami Beach, FL (1980) 370–372.
15. Ohta, Y., Maenobu, K. and Sakai, T., Obtaining surface orientation from texels under perspective projection, in: *Proceedings IJCAI-81*, Vancouver, BC (1981) 746–751.
16. Rosenfeld, A., A note on automatic detection of texture gradients, *IEEE Trans. Comput.* **24** (1975) 988–991.
17. Schwartz, L., *Théorie des Distributions* **1, 2** (Hermann, Paris, 1950, 1951).
18. Schwartz, L., *Méthodes Mathématiques pour les Sciences Physiques* (Hermann, Paris, 1961).
19. Sedgwick, H.A., Environment-centered representation of spatial layout: Available visual information from texture and perspective, in: J. Beck, B. Hope and A. Rosenfeld (Eds.), *Human and Machine Vision* (Academic Press, New York, 1983) 425–458.
20. Stevens, K.A., Slant-tilt: The visual encoding of surface orientation, *Biol. Cybern.* **46** (1983) 183–195.
21. Witkin, A.P., Recovering surface shape and orientation from texture, *Artificial Intelligence* **17** (1981) 17–45.
22. Zucker, S.W., Rosenfeld, A. and Davis, L.S., Picture segmentation by texture discrimination, *IEEE Trans. Comput.* **24** (1975) 1228–1233.

*Received May 1986; revised version received August 1987*