Chapter 12

3D Content Creation by Passive Optical Methods

Luca Ballan, Nicola Brusco and Guido Maria Cortelazzo
Department of Information Engineering
University of Padova
Via Gradenigo 6/B, 35131 Padova, Italy

12.1 Introduction

The possibility of obtaining 3D models, i.e., mathematical descriptions of real objects or scenes has paved the way to a wide range of new and exciting applications in fields such as virtual simulation, human-computer interaction (HCI), scientific visualization, cultural heritage documentation, medicine, industrial prototyping, reverse engineering, entertainment (movies and video games), web-based commerce and marketing, just to name a few.

The construction of the 3D model of a real object or scene by optical sensors, also referred to as *3D modeling pipeline*, essentially consists of four steps: 1) data acquisition, 2) calibration, 3) reconstruction, and 4) model editing. Any optical sensing device used to collect data can only capture the surface front side and not what is occluded by it. Therefore, a full model must be built from a number of images covering the entire object (data acquisition). In order to perform 3D reconstruction, the camera's parameters must be estimated by a procedure called calibration. Such information can also be obtained from the acquired images if they represent some common regions (by a procedure which is typically called self-calibration). Reconstruction is then performed and the resulting model is stored in an efficient description such as polygonal meshes, implicit surfaces, depth maps or volumetric descriptions. In practical situations, reconstruction may lead to

models with some imperfections; thus, a further repairing step is recommend (model editing) [Davis *et al.* (2002); Levoy *et al.* (2000)].

Optical 3D reconstruction methods can be classified into passive or active methods based on the type of sensors used in the acquisition process. Passive sensing refers to the measurement of the visible radiation which is already present in the scene; active sensing refers instead, to the projection of structured light patterns into the scene to scan. Active sensing facilitates the computation of 3D structure by intrinsically solving the correspondence problem which is one of the major issues with some passive techniques. For a detailed description of the operations of the 3D modeling pipeline by active sensing see [Rushmeier and Bernardini (2002); Rioux *et al.* (2000)]. In general, active techniques such as those based on laser scanning tend to be more expensive and slower than their passive counterparts. However, the best active methods generally produce more accurate 3D reconstructions than those obtained by any passive technique.

Passive optical methods, as previously mentioned, do not need auxiliary light sources. In this case, the light reflected by the surface of the object comes from natural sources, that is, sources whose characteristics are generally unknown and in most cases, not controllable by the acquisition process. Furthermore, passive methods do not interact in any way with the observed object, not even with irradiation. Passive reconstruction, in principle, can use any kind of pictures, i.e., it does not need pictures taken for 3D reconstruction purposes (even holiday photographs could be used). Passive methods are more robust than their active counterparts, can capture a wider range of objects, can be obtained by inexpensive hardware (such as a simple digital camera) and are characterized by fast acquisition times. Such features are the reason for the attention they received and they continue to receive. Their drawbacks concern accuracy and computational costs. Indeed, passive reconstruction algorithms are complex and time-consuming. Moreover, since their acquisition scenarios are often far from the ideal conditions, noise level is typically much higher than that of active methods which tend to guarantee optimal conditions.

Passive optical methods are classified by the types of visual cues used for 3D reconstruction. For this reasons, they are also called *"Shape from X"* (SfX) techniques, where X stands for the cue or the cues used to infer shape. Methods which deal with one single type of visual cue are called monomodal whereas methods jointly exploiting information of different types are called multimodal. The simultaneous use of different cues, in principle, is clearly more powerful than the use of a single one; however, this poses

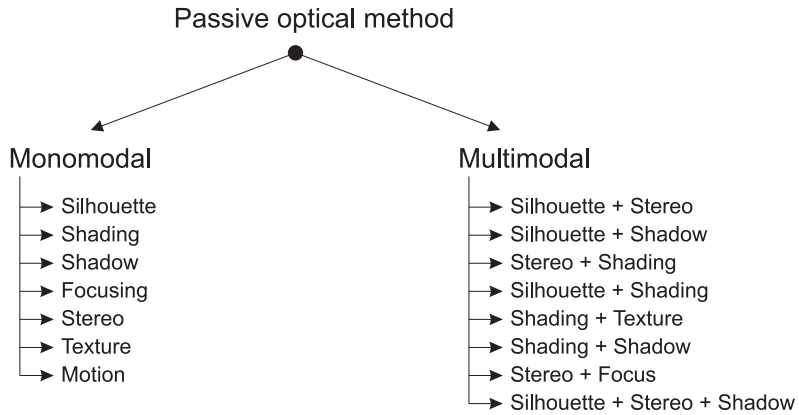

Fig. 12.1 Overview of existing passive optical methods.

the challenge of how to synergistically fuse different information avoiding mutual conflicts.

Figure 12.1 proposes a taxonomy of the passive optical methods. Typical information used for reconstruction are:

- Silhouette or apparent contour;
- Shading;
- Shadow;
- Focusing;
- Pictures differences, i.e., stereo information;
- Texture;
- Motion;

This chapter reviews 3D reconstruction by passive optical methods. This is not an easy task in light of the broad scope of the topic. The spirit we adopted is to give a conceptual outline of the field, referring the reader to the literature for details. We also reserve special attention to recent methods. Section 12.2 introduces basic concepts and terminology about the image formation process. Section 12.3 reviews major monomodal methods: shape from silhouette, shape from shading, shape from shadows, shape from focus/defocus and shape from stereo. In Section 12.4 we introduce a framework for multimodal methods, focusing on the deformable model technique. Finally, Section 12.5 draws the conclusions.

12.2 Basic notation and calibrated images

A calibrated image is an image for which all the parameters of the camera used to take it are known. Formally, a calibrated image is an ordered pair (I, ζ) where I is an image and ζ is an image formation function $\Re^3 \to \Re^2$ that maps the points from the physical 3D world to the image plane. An image is a function from a rectangular subset of $\Re^2$ representing the image space coordinates to an interval of $\Re$ representing the image intensity values. In this section, the image formation process is approximated using the ideal pinhole camera model (see Fig. 12.2) with lens distortion. This means that neither the effects due to finite aperture nor other types of lens aberrations are considered. In this case, ζ can be expressed as the combination of two functions, namely

$$\zeta = \phi \circ V \tag{12.1}$$

where $\phi : \Re^2 \to \Re^2$ is a nonlinear bijection representing the camera lens distortion and V is a function $\Re^3 \to \Re^2$, called *view*, incorporating both the pinhole model and the camera point of view information. Function V is a combination of two further functions, i.e., $V = \pi \circ T$. Function π is the pinhole perspective projection[1] simply defined as $\pi(x, y, z) = \left(\frac{x}{z}, \frac{y}{z}\right)$ for all point $P = (x, y, z)$ in $\Re^3$ with $z > 0$. Function $T : \Re^3 \to \Re^3$ is an affine bijective transformation which performs 3D coordinates transformation from the world space to the camera space. Let us note that, given a calibrated image (I, ζ), one can always find its non-distorted version $(I \circ \phi, V)$ by estimating camera lens distortion parameters ϕ from image I. This is a classical inverse problem for which a vast literature is available. Popular methods are due to [Tsai (1987)] which estimates ϕ using known calibration patterns and to [Prescott and McLean (1997)] which use the knowledge that the image represents straight lines of the scene.

Projective Geometry is a powerful framework for describing the image formation process, not adopted in this chapter for reasons of simplicity. Interested readers are referred to [Hartley and Zisserman (2000)] for an excellent introduction.

By definition, transformation T can be written as

$$T(P) = M \cdot P + O \tag{12.2}$$

where M is an invertible $3x3$ matrix and $O, P \in \Re^3$. Furthermore, M and O form to the so-called projection matrix K, a $3x4$ matrix defined as

[1] This model was first proposed by Brunelleschi at the beginning of the 15th century.

follows

$$K = \begin{bmatrix} M & O \end{bmatrix} \tag{12.3}$$

Projection matrix K is related to the physical model of the ideal pinhole camera and can be decomposed according to the following scheme

$$K = I \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times E \tag{12.4}$$

where I is the intrinsic matrix, depending on the so-called intrinsic parameters only due to the camera internal characteristics, and E is the extrinsic matrix, depending on the so-called extrinsic parameters only due to the camera position and orientation in the space. Namely, matrix I is defined as follows

$$I = \begin{bmatrix} \frac{f}{p_x} & \frac{(\tan \alpha) f}{p_y} & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{12.5}$$

where f (expressed in millimeters) is the camera focal length, that is the distance between the sensor surface (also known as retinal plane or image plane) and pinhole C (also known as center of projection); p_x, p_y respectively are the width and the height in millimeters of a single pixel on the retinal plane; α is the skew angle, measuring how much the image axes x and y are away from orthogonality; $c = (c_x, c_y, 1)$ is the principal point of the camera, i.e., the point at which the optical axis intersects the retinal plane. We recall that the optical axis is the line, orthogonal to the retinal plane, passing through the center of projection C. Another useful parameter is the camera field-of-view along the y axis defined as

$$FOV_y = 2arctan\left(\frac{p_y \ N_y}{2f}\right) \tag{12.6}$$

where N_y is the vertical resolution of the sensor. Figure 12.2(above) shows the ideal pinhole camera. Rays of light pass through the pinhole and form an inverted image of the object on the sensor plane. Figure 12.2(below) shows an equivalent pinhole model where the image plane is placed in front of the center of projection obtaining a non-inverted image.

Matrix E is defined as

$$E = \begin{bmatrix} R & t^T \\ 0 & 1 \end{bmatrix} \tag{12.7}$$

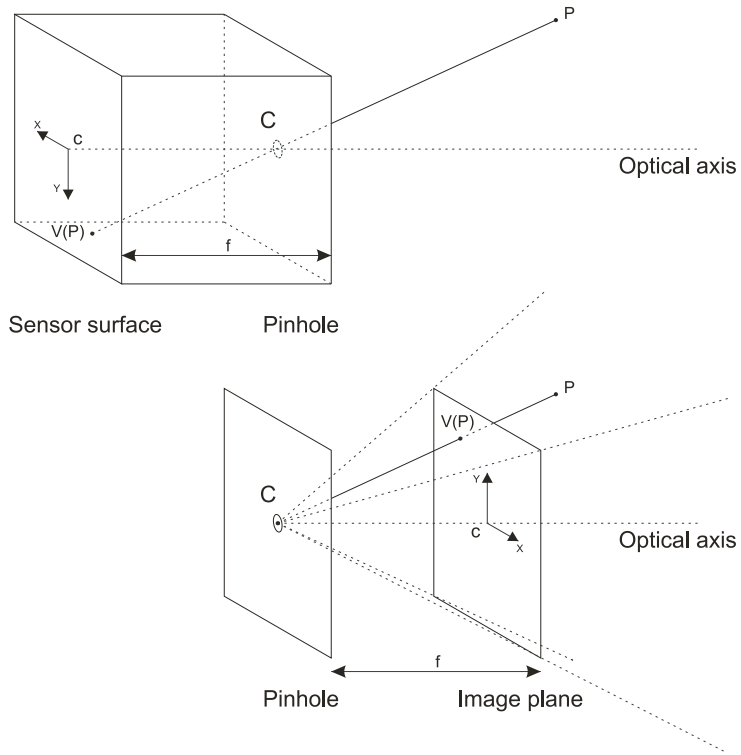

Fig. 12.2 Ideal pinhole camera (above) and its equivalent model (below) where the image plane is placed in front of the center of projection.

where R is $3x3$ rotation matrix and t is a vector belonging to $\Re^3$. For example, given a camera with center of projection C, optical axis D and *up-vector* U (that is the y axis of the camera reference system), the relative extrinsic matrix is:

$$E = \begin{bmatrix} B^{-1} & -B^{-1}C^T \\ 0 & 1 \end{bmatrix} \tag{12.8}$$

where:

$$B = \begin{bmatrix} (U \times D)^T & U^T & D^T \end{bmatrix} \tag{12.9}$$

The center of projection $C = (X_c, Y_c, Z_c)$ can be obtained from the

columns of projection matrix $K = \begin{bmatrix} k1 & k2 & k3 & k4 \end{bmatrix}$ as follows

$$X_c = \frac{\det\left(\begin{bmatrix} k2 & k3 & k4 \end{bmatrix}\right)}{Q} \tag{12.10}$$

$$Y_c = -\frac{\det\left(\begin{bmatrix} k1 & k3 & k4 \end{bmatrix}\right)}{Q} \tag{12.11}$$

$$Z_c = \frac{\det\left(\begin{bmatrix} k1 & k2 & k4 \end{bmatrix}\right)}{Q} \tag{12.12}$$

where:

$$Q = -\det\left(\begin{bmatrix} k1 & k2 & k3 \end{bmatrix}\right) \tag{12.13}$$

In order to extract 3D information from a set of images, the related view functions must be estimated for each image of this set. This can be done in two ways: conventional calibration or self-calibration. The first approach uses pictures imaging a known target object such as a planar checkerboard. In this case, function V can be estimated by solving an over-constrained linear system [Hartley and Zisserman (2000)]. Self-calibration instead, computes the view functions associated to a set of un-calibrated images without any information about the scene or any object in it. These methods, see for instance [Mendoca and Cipolla (1999)], essentially extract relevant features from two or more images then, find the matching between them and finally, proceed like conventional calibration methods.

12.3 Monomodal methods

This section introduces the most common monomodal methods namely, the methods using silhouette, shading, shadow, focus and stereo as 3D reconstruction information. Texture and motion are excluded from this analysis, however the interested reader is referred to [Forsyth (2002)] and [Jebara *et al.* (1999)] for an example of these two techniques.

12.3.1 *Silhouette*

Algorithms which reconstruct 3D objects using only silhouette information extracted from a set of images are called *"Shape from Silhouette"* methods. They were first proposed in [Baumgart (1974)] and afterwards formalized in [Laurentini (1994)], where the concept of *visual hull* was first introduced.

All these methods must face the problem of extracting silhouette information from the set of images. In other words, in each picture, they must

identify the points belonging to the object to be acquired with respect to the background. This problem does not have a general solution as it strongly depends on the scene characteristics. The most common approaches to this task are chroma keying (e.g., blue screen matting [Chuang *et al.* (2001)]), background subtraction [Piccardi (2004)], clustering [Potmesil (1990)] and many other segmentation techniques. For a comprehensive account see [Lucchese and Mitra (2001)]. However, silhouette information is affected by two types of error. The first one is the quantization error due to image resolution and it is directly proportional to the camera-object distance z as

$$\varepsilon_x = \frac{p_x}{2f} z, \quad \varepsilon_y = \frac{p_y}{2f} z \tag{12.14}$$

The second one depends on the specific silhouette extraction method and its amount is usually confined within ± 1 pixel.

In order to recall the concept of visual hull, some definitions related to the notion of contour may be useful. Given a view V and a closed surface M in $\Re^3$, let us denote by $V(M)$ the projection (or the *silhouette*) of M on the image plane of V, i.e., the shape of M viewed by V, and by

$$\gamma_M^V = \partial V(M) \tag{12.15}$$

the *apparent contour* of M viewed by V, and by

$$\Gamma_M^V = V^{-1}\left(\gamma_M^V\right) \tag{12.16}$$

the *3D contour* of M viewed by V.
By definition $V(M)$ is a set of points in $\Re^2$ and its boundary γ_M^V is a set of curves in $\Re^2$ which do not intersect each other. As we can easily see with the aid of Figure 12.3, neither $V(M)$ nor γ_M^V are generally regular. Indeed, it is likely that they have some singularities. On the contrary, Γ_M^V is a set of not necessarily closed curves in $\Re^3$, with points belonging to M.

Shape from silhouette methods use $V(M)$ as source of information. However, there exists a class of methods, called *"Shape from Contour"* [Cipolla and P.Giblin (2000)], that use the apparent contour γ_M^V instead of $V(M)$ in order to infer shape.

The concept of *visual hull* [Laurentini (1994)] is a fundamental definition for the shape from silhouette methods.

Definition 12.1. Given a set of views $R = (V_1, \ldots, V_n)$ and a closed surface M in $\Re^3$, the visual hull of M with respect to R, denoted as $vh(M, R)$, is the set of points of $\Re^3$ such that $P \in vh(M, R)$ if and only if for every view $V_i \in R$, the half-line starting from the center of projection of V_i and passing through P, contains at least one point of M.

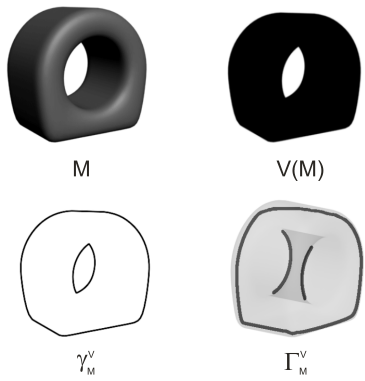

Fig. 12.3 M is a 3D object. $V(M)$ represents the silhouette of M, γ_M^V its apparent contour and Γ_M^V its 3D contour. In the figure, Γ_M^V is slightly rotated with respect to the point of view of the other three pictures in order to evidence that Γ_M^V is a set of not necessarily closed 3D curves.

In other words, the visual hull of a surface M related to a set of views R is the set of all points in the 3D space which are classified as belonging to the object for every view $V_i \in R$. Laurentini proved that the boundary of the visual hull $\partial vh(M, R)$ is the best approximation of M that can be obtained using only silhouette information coming from the projections of M in each view of R. Some implications of this result follow:

- the visual hull always includes the original surface, i.e., $M \subseteq vh(M, R)$, or in other words, the visual hull is an upper-bound of the original object;
- $\partial vh(M, R)$ and M have the same projections in R, in other words for every $V \in R$, we have:

$$V(M) = V(\partial vh(M, R)) \tag{12.17}$$

- If $R_1 \subseteq R_2$ then $vh(M, R_2) \subseteq vh(M, R_1)$
- $vh(M, R) = \bigcap_{V \in R} vh(M, \{V\})$

The last property suggests a method for computing the visual hull as the intersection of the visual cones $vh(M, \{V\})$ generated by M for every view $V \in R$ (see Fig. 12.4). A visual cone $vh(M, \{V\})$ is formed by all the half-lines starting from the center of projection of V and intersecting the projection of M on the image plane of V.

All shape from silhouette methods are based on the above principle. They can be classified by the way the visual hull is internally represented,

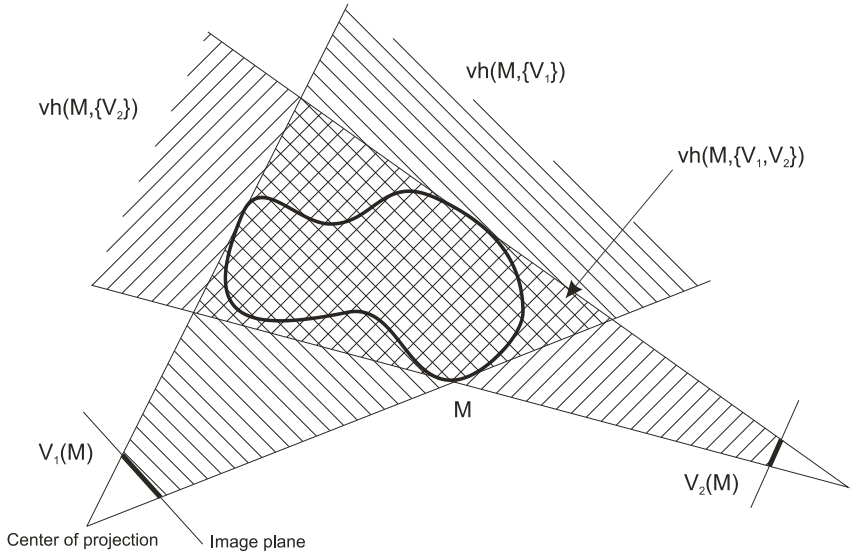

Fig. 12.4 Computation of the visual hull as intersection of the visual cones generated by V_1 and V_2.

namely by voxels or by polyhedra. The former class, called *"Volume Carving"* algorithms [Potmesil (1987)], was the first to be proposed. The idea behind it is to divide space into cubic elements of various sizes, called *voxels*, in order to store volume information of the reconstructed object. The latter class of algorithms, recently formulated in [Matusik *et al.* (2001)], represents the boundary of the reconstructed visual hull by polygonal meshes. They are proposed for real-time applications aimed at acquiring, transmitting and rendering dynamic geometry. Indeed, polyhedral visual hull can be rapidly computed and rendered using the projective texture mapping feature of modern graphics cards [Li *et al.* (2003)]. Besides, polyhedral representations give exact estimations of the visual hulls avoiding the quantization and the aliasing artifacts typical of the voxel approach. However, voxel representations are preferred when the result of shape from silhouette is used as first approximation to be refined by other reconstruction algorithms such as shadow carving (see Section 12.3.3) and some multimodal methods (see Section 12.4).

For this reason the remaining of this section focuses on the volume carving algorithms. In this case, the 3D space is divided into voxels which can bear three types of relationship with respect to the visual hull: "belong",

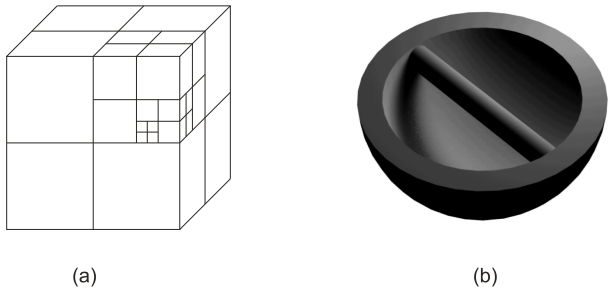

Fig. 12.5 (a) Space subdivision by an octree. (b) Example of surface M for which its external visual hull has genus lower than the genus of M. The visual hull cannot completely describe the topology of this surface.

"partially belong" or "not belong". In order to verify such characteristics one must check if a voxel completely belongs to every visual cone[2]. In this case the voxel belongs to the visual hull of M. Otherwise, if the voxel is completely outside at least one visual cone, then it does not belong to the visual hull. In any other case, the voxel partially belongs and one must further subdivide it and repeat the check with respect to its sub-voxels until the desired resolution is reached.

Data structures like *octrees* [de Berg *et al.* (1999)] allow for a fast space subdivision and reduce the memory requirements. An octree is a tree where each internal node has 8 children. Every node j is associated with a cube B such that the set of the cubes associated to each child of j is an equipartition of B. The root of the tree represents the whole space under analysis, which is divided into 8 cubes of equal size as shown in Fig. 12.5(a). Each of these cubes can be again subdivided into 8 further cubes or alternatively be a leaf of the tree. The possibility of arbitrarily stopping the subdivision is the key characteristic of octrees. In fact, octrees can optimize memory requirements since they allow to describe volumes by a multi-resolution grid where detailed regions are described at resolutions higher than those in uniform regions.

Finally, in order to find a polygonal mesh representation of the boundary of the estimated volume, one may resort to the *"marching cubes"* algorithm [Cline and Lorensen (1987)]. Figure 12.6 shows an example of a model obtained by a volume carving algorithm.

Let us observe that given the set of all possible views whose centers of

[2]Observe that, since voxels are cubes, one can determine whether all their points belong to a visual cone only by checking the eight vertices of the cube.

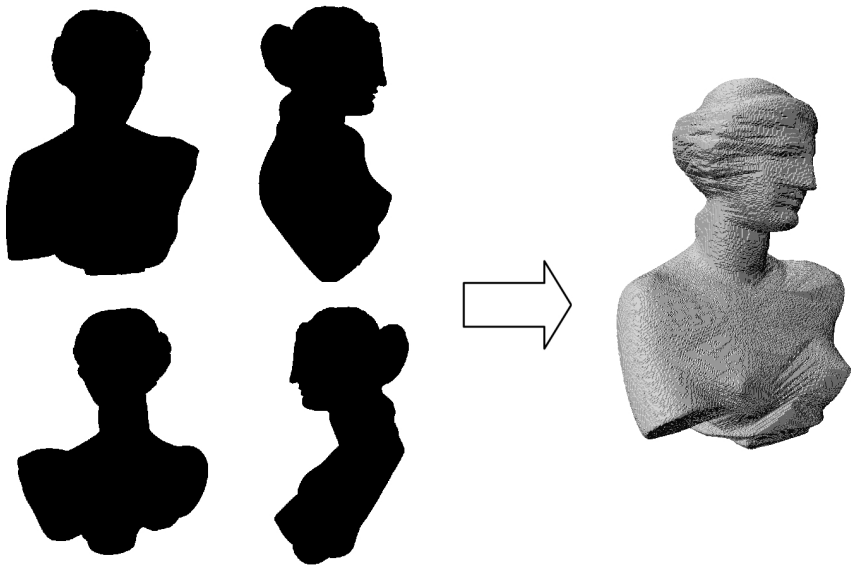

Fig. 12.6 Model obtained by volume carving algorithm.

projection are outside the convex hull of M, the relative visual hull, called $vh_{\infty}(M)$, is in general not equal to M. In fact, $vh_{\infty}(M)$ cannot describe the concave regions of M which are not visible from viewpoints outside the convex hull of M. As a consequence, in general, the visual hull cannot completely capture the topology of a surface. $vh_{\infty}(M)$ is called the external visual hull and it is a subset of the convex hull of M. Figure 12.5(b) shows an object for which its external visual hull has genus lower than that of the original surface.

In conclusion, shape from silhouette algorithms are fast and robust but can only reconstruct a small set of objects, i.e., those objects the visual hulls of which, related to the available views, are similar to their surfaces.

12.3.2 *Shading*

Shading information is used in both photometric stereo and shape from shading algorithms. The former operates with a series of pictures of the object taken under different lighting conditions. The latter instead, recovers the surface shape from the brightness of a single picture.

Both methods rest on approximations of the reflectance characteristics of the object to be reconstructed, that are the relationships between incoming illumination to a point on the surface and the light reflected by it. For this reason, it may be useful to recall some radiometric definitions.

Light power is the amount of light energy per unit time, measured in Watt $[W]$. The *outgoing radiance* at surface point P in the direction $\omega_o = (\theta_o, \phi_o)$ (where θ_o and ϕ_o are the two angles defining direction ω_o) is the light power per unit area perpendicular to ω_o emitted at P in the unit solid angle of direction ω_o. Such a radiance is denoted as $L_o(P, \omega_o)$ where the subscript o denotes that it is an outgoing radiance. It is measured in $[W][m]^{-2}[sr]^{-1}$, where *Steradian* [sr] is the unit of solid angle. On the other hand, the *incoming radiance* $L_i(P, \omega_i)$ at surface point P in direction $\omega_i = (\theta_i, \phi_i)$ is the incident light power at P per unit area perpendicular to ω_i in the unit solid angle of direction ω_i. Note that, if the surface normal at P forms an angle β with respect to direction ω_i, the infinitesimal area dA centered at P seen from the direction ω_i is $dA\cos(\beta)$. Therefore, the incoming light power per unit area contributed to P by the light sources through the infinitesimal solid angle $d\omega$ of direction ω_i, is $L_i(P, \omega_i)\cos(\beta)\, d\omega$. This quantity is called *incident irradiance* at surface point P in the direction ω_i and it is measured in $[W][m]^{-2}$.

The *bidirectional reflectance distribution function* (BRDF) was introduced in [Nicodemus (1970)] as a unified notation of reflectance in terms of incident and reflected beam geometry. It is defined as the ratio between the outgoing radiance at surface point P in the direction ω_o and the incident irradiance at P in the direction ω_i, i.e.,

$$f_r(P, \omega_o, \omega_i) = \frac{L_o(P, \omega_o)}{L_i(P, \omega_i)\cos(\beta)\, d\omega} \tag{12.18}$$

and it is measured in $[sr]^{-1}$.

The actual BRDF of an object is usually a very complex function and it is difficult to estimate in practical situations, therefore a number of approximations are used instead. For example, *Lambertian* (or ideal diffuse) surfaces, i.e., surfaces that reflect light equally in all directions, lead to a strong simplification namely, they have a constant BRDF

$$f_r(P, \omega_o, \omega_i) = \rho(P) \tag{12.19}$$

where ρ is called the *albedo* or the diffuse reflectance of the object. Models for partially specular surfaces were developed by Torrance-Sparrow [Torrance and Sparrow (1967)], Phong [Phong (1975)] and Blinn [Blinn (1977)]. The last two models are widely used in computer graphics.

The algorithms described in this section consider only Lambertian surfaces and local shading models; thus, neither specularity nor interreflections are considered. However, state of the art of photometric stereo and shape from shading algorithms make use of more general BRDF models such as the simplified Torrance-Sparrow model used in [Healey and Binford (1988)].

Some definitions used in both types of algorithms are in order. Let M be the unknown surface in $\Re^3$ and let $I(x, y)$ be the image intensity seen by a view V. If the surface point $P \in M$ is visible from the viewpoint V then $I(V(P))$ is its brightness. Clearly, $I(V(P))$ is proportional to the outgoing radiance leaving P in direction of the center of projection of V. Therefore, for Lambertian objects illuminated by a single point light source, one can write

$$L_o(P, \omega_o) = \rho(P) L_i(P, \omega_i) \cos(\beta) \tag{12.20}$$

thus,

$$I(V(P)) = \rho(P) l(P) L(P) \cdot N(P) \tag{12.21}$$

where $l(P)$ and $L(P)$ are respectively, intensity and direction of the incident light at P, $\rho(P)$ is the surface albedo at P and $N(P)$ is the surface normal.

12.3.2.1 *Photometric stereo*

Photometric stereo was first introduced in [Woodham (1980)]. Given a set of calibrated images $(I_1, V), \ldots, (I_n, V)$ taken from the same point of view V but under different lightings $L_1, \ldots, L_n$, one can estimate surface normal $N(P)$ for every visible point of M. Let

$$\mathbf{I}(x, y) = [I_1(x, y), \ldots, I_n(x, y)] \tag{12.22}$$

be the vector of all measured brightness at image point $(x, y) = V(P)$, for any visible point P of M, and let

$$\mathbf{L}(x, y) = \begin{bmatrix} l_1(P) L_1(P) \\ \vdots \\ l_n(P) L_n(P) \end{bmatrix} \tag{12.23}$$

be the matrix of all light directions and intensities incident at P. From Eq. (12.21), one may write

$$\mathbf{I}^T(x, y) = \rho(P) \mathbf{L}(x, y) \times N^T(P) \tag{12.24}$$

which is a linear system of n equations in the three unknowns $\rho(P)N(P)$ [3]. Eq. (12.24) has a unique solution when $n > 3$ and it can be solved using least square methods.

Once the values of $\rho(P)N(P)$ are available for each visible point P, one can extract the surface albedo and the normal at P using $\rho(P) = \|\rho(P)N(P)\|$ and $N(P) = \rho(P)N(P) / \|\rho(P)N(P)\|$ respectively. Retrieving shape from normals is trivial under the assumption that the view V performs an orthographic projection. Indeed, let us represent M by a Monge patch description, i.e.,

$$M = \{(x, y, z(x, y)) \mid \forall (x, y)\} \tag{12.25}$$

where $z(x, y)$ is the surface depth at (x, y). Consequently, the surface normal at $P = V^{-1}(x, y) = (x, y, z(x, y))$ is

$$N(P) = \frac{(\partial z_x, \partial z_y, -1)}{\sqrt{1 + \partial z_x^2 + \partial z_y^2}} \tag{12.26}$$

where $(\partial z_x, \partial z_y)$ are the partial derivatives of $z(x, y)$ with respect to x and y. $(\partial z_x, \partial z_y)$ can be recovered from $N(P) = (N_x(P), N_y(P), N_z(P))$ using the following

$$(\partial z_x, \partial z_y)(P) = \left(-\frac{N_x(P)}{N_z(P)}, -\frac{N_y(P)}{N_z(P)}\right) \tag{12.27}$$

Surface M can be finally reconstructed by integrating a one-form:

$$z(x, y) = z(x_0, y_0) + \int_\gamma (\partial z_x dx + \partial z_y dy) \tag{12.28}$$

where γ is a planar curve starting at (x_0, y_0) and ending at (x, y). $(x_0, y_0, z(x_0, y_0))$ is a generic surface point of known height $z(x_0, y_0)$. Clearly, if $z(x_0, y_0)$ is unknown, the result will be the actual surface up to some constant depth error.

Unfortunately, errors in surface normal measurements can propagate along the curve γ generating unreliable solutions. For this reason, [Vega (1991)] suggests an alternative height recovery method based on local information only. The more general case where V performs a perspective projection is treated in [Tankus and Kiryati (2005)].

[3] $\rho(P)N(P)$ has only three degrees of freedom because $N(P)$ is assumed to be normalized.

12.3.2.2 *Shape from shading*

Shape from shading algorithm operates only on a single image I, therefore for each image point $(x, y) = V(P)$, we have one equation in three unknowns

$$I(x, y) = \rho(P)\, l(P)\, L(P) \cdot N(P) \tag{12.29}$$

which cannot be solved without imposing additional constraints.

The first attempt to solve Eq. (12.29) was done by Horn in his PhD thesis [Horn (1970)]. Since then, many other solution approaches were developed typically classified into: minimization approaches, propagation approaches, local approaches and linear approaches. For an extensive description of all these methods the reader is referred to [Zhang *et al.* (1999)].

In this chapter we only introduce the minimization approach suggested in [Ikeuchi and Horn (1981)]. Ikeuchi and Horn reformulated the solution of Eq. (12.29) as the minimization of a cost functional ξ defined as

$$\xi(M) = Bc(M) + \lambda \cdot Sc(M) \tag{12.30}$$

where $Bc(M)$ is the brightness constraint and $Sc(M)$ is the smoothness constraint. The former measures the the total brightness error of the reconstructed image compared with the input image, namely

$$Bc(M) = \int\int \left(I(x, y) - \overline{I}(x, y)\right)^2 dxdy \tag{12.31}$$

where $I(x, y)$ is the input image and $\overline{I}(x, y)$ is the image related to the estimated surface M.

Cost functional $Sc(M)$ penalizes non-smooth surfaces, reducing the degrees of freedom of Eq. (12.29). It is defined as

$$Sc(M) = \int\int \left(\left\| \frac{\partial N}{\partial x}(x, y) \right\|^2 + \left\| \frac{\partial N}{\partial y}(x, y) \right\|^2 \right) dxdy \tag{12.32}$$

Constant λ controls surface smoothness.

In this formulation, $\rho(P)$ is assumed to be known for all $P \in M$ thus, one can add another constraint which imposes normals to be unit. This is what Brooks and Horn did in 1985. The new term was named unit normal constraint and it was defined as follows

$$\int\int \left(1 - \|N(x, y)\|^2\right) dxdy \tag{12.33}$$

The numerical solution is typical achieved using gradient descent algorithms on the Euler-Lagrange equation related to Eq. (12.30) (see Section 12.4.1 for additional information).

Fig. 12.7 An example of the concave/convex ambiguity: it seems that this two images represent two different objects, a concave and a convex one. Nevertheless, the first image is a rotated version of the second one.

12.3.2.3 *Estimating the light source properties*

It can be proven that both photometric stereo and shape from shading become ill-posed problems if light direction, intensity and surface albedo are unknown. This means that a solution may not be unique and it strongly depends on these three parameters[4]. The so-called concave/convex ambiguity, occurring when light orientation is unknown, is a clear example of this ill-posed characteristic. The concave/convex ambiguity refers to the fact that, the same image seems to describe two different objects, one concave and the other convex as shown in Fig. 12.7.

More generally, [Belhumeur *et al.* (1997)] showed that a surface $(x, y, z(x, y))$ is indistinguishable from its *"generalized bas-relief"* (GBR) transformation, defined as

$$\overline{z}(x, y) = \lambda z(x, y) + \mu x + \nu y \tag{12.34}$$

if its albedo and the light properties are unknown. More precisely for all possible values of λ, μ and ν there exists an albedo $\overline{\rho}(x, y)$ and a light $\overline{L}$ such that the brightness image related to the depth map $\overline{z}$ is equal to the one related to z. Moreover, Belhumeur *et al.* showed that even if self-shadow information is used in addition to shading, the two surfaces $\overline{z}$ and z remain indistinguishable.

Two interesting methods to estimate light direction are due to [Koenderink and Pont (2003)] and [Vogiatzis *et al.* (2005a)]. The former recovers the azimuthal angle of the light sources from a single image using texture information. The limit of this approach is the assumption that the textured

[4]If we suppose Lambertian surfaces, $\rho(P)$ an $l(P)$ can be grouped together thus, we have only three degrees of freedom.

surface has to be an isotropic gaussian random rough surface with constant albedo.

Instead, [Vogiatzis *et al.* (2005a)] use the brightness values of the contour points of the imaged object in order to estimate light direction by equation Eq. (12.21). Indeed in such points, surface normals can be retrieved knowing that they are perpendicular to the viewing ray connecting these points to the center of projection of the camera.

12.3.3 *Shadows*

Scene shadows bear a lot of information about the shape of the existing objects. They can give information when no other sources do, indeed shadow regions represent the absence of any other type of information. Methods which exploit this particular visual cue are called either *"Shape form Shadows"* or *"Shape from Darkness"*. They first appear in [Shafer and Kanade (1983)] where shadows were used to relate the orientations of two surfaces. Subsequent works on shadows generally used either the shape of the object casting the shadow in order to constrain the shape of the object being shadowed or vice versa. Indeed, one can infer the shape of an unknown object from the shadows casted on it by a known one. This is the same principle used in structured light projectors with the only difference that the methods based on shadow information use shadow patterns instead of light patterns. The interested reader is sent to [Bouguet and Perona (1999)] for the description of a low cost scanner based on this principle. On the contrary, if an unknown object casts shadows on a known one, for simplicity, let it be a plane, one can use an appropriately modified shape from silhouette method in order to reconstruct its shape. This approach is proposed in [Leibe *et al.* (2000)] in order to avoid segmentation problems implicit in shape from silhouettes methods.

However, in general, shape from darkness methods deal only with the shadow that an object casts on itself, the so-called *self-shadow*. In this case, both the object that casts the shadow and the object being shadowed are unknown as they are all parts of the same unknown surface. Nevertheless, self-shadow can reveal a lot of information. Indeed, let us observe the situation depicted in Figure 12.8(a) where p_1 is first shadow boundary points and p_2 is the last one. Knowing their coordinates, one can obtain an upper bound for the shadowed region, i.e., the line η. In other words, a point in such a region cannot be above line η, otherwise it would be a lighted point. Furthermore, all lighted points at the right of p_1 must be above η

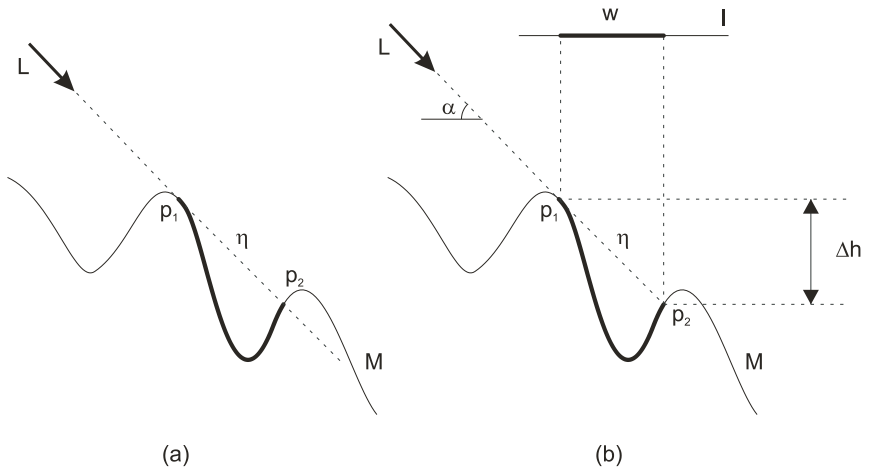

Fig. 12.8 Shadowed surfaces: (a) the coordinates of p_1 and p_2 are assumed to be known; (b) α and w are known.

otherwise they would be shadowed. Thus, η is also a lower bound for the lighted points. Obviously, same results can be obtained if one knows the coordinates of the light source and the coordinates of one of the points p_1 or p_2.

Figure 12.8(b) shows a situation similar to the previous one but, in this case, it is assumed that the coordinates of p_1 and p_2 are unknown. Moreover, it is supposed that the camera performs an orthographic projection of the scene and that the light source casts parallel rays of known direction α. This can be obtained by placing both the camera and the light source far away from the scene. The measured shadow width w can be used to estimate the relative height between p_1 and p_2 using the following

$$\Delta h = w \tan(\alpha) \tag{12.35}$$

Moreover, if one assumes that the unknown surface M admits a tangent plane in p_1, such a plane must be parallel to η.

¿From the above considerations, using multiple images taken with different light source positions, one can estimate the unknown surface by constraining a model (e.g. a spline) to fit all the extracted information about relative heights and tangent planes (see [Hatzitheodour and Kender (1988)]).

Furthermore, combining equations of type (12.35) together with the linear inequality constraints related to the various η, one can obtain a set of

upper/lower bounds and equations which can be solved by *Linear Programming* algorithms as in [Yang (1996)] or by iterative relaxation methods like in [Daum and Dudek (1998)].

[Smith and Kender (1986)] introduced the concept of *shadowgram*. Let us suppose the situation depicted in Fig. 12.9(a) where θ is the angle between the x-axis and the light rays. A shadowgram is a binary function $f(x, \theta)$ recording, for each value of θ, a 0 (black) value at the x coordinates of the shadow points and a 1 (white) value at the x coordinates of the lighted points. Therefore, a shadowgram typically looks like two irregular black stripes of variable thickness. Smith and Kender demonstrate that the real surface can be reconstructed from the curves representing the discontinuities of the shadowgram $f(x, \theta)$, i.e., the edges of the dark stripes.

The definition of self-shadow consistency follows. Let us assume first that the scene is only illuminated by a point light source positioned at ℓ. Given an object M, the self-shadow generated on M by the light ℓ is the set of all the points on its surface not visible from ℓ. Let $\Theta(M, \ell)$ denote this set. In other words, a generic point P belongs to $\Theta(M, \ell)$ if and only if the segment joining P and ℓ intersects M in at least one point different from P. Therefore, given a calibrated image (I, V), the shadow region generated by ℓ on M and viewed by V is the set of the V-projections of all the points of $\Theta(M, \ell)$ visible from V. Let $\Omega(M, \ell, V)$ denotes this set; then, formally it is

$$\Omega(M, \ell, V) = V(\Theta(M, \ell) \cap \Pi(M, V)) \tag{12.36}$$

where $\Pi(M, V)$ is the set of all the points of M visible from V. Now, given the image I and the estimated shadow regions on I, call them $\omega(I)$, one can say that M is self-shadow consistent with image I if and only if $\omega(I) \subseteq \Omega(M, \ell, V)$. In other words, it is consistent if V does not see shadow points which are not theoretically generated by M. The contrary is not required, since, as we will describe below in this section, in practical situations, only subsets of $\omega(I)$ can be accurately estimated. In this way, the consistent condition is relaxed making consistent surfaces which are not. However, for correctness, one could also estimate $\overline{\omega(I)}$, i.e. the complement of $\omega(I)$, and define that consistency holds when also $\overline{\omega(I)} \subseteq \overline{\Omega(M, \ell, V)}$ holds. Extension to multiple lights is trivial; since, given the set of active lights $(\ell_1, \ldots, \ell_k)$ one can define

$$\Omega(M, L, V) = \bigcup_{\forall \ell_j} \Omega(M, \ell_j, V) \tag{12.37}$$

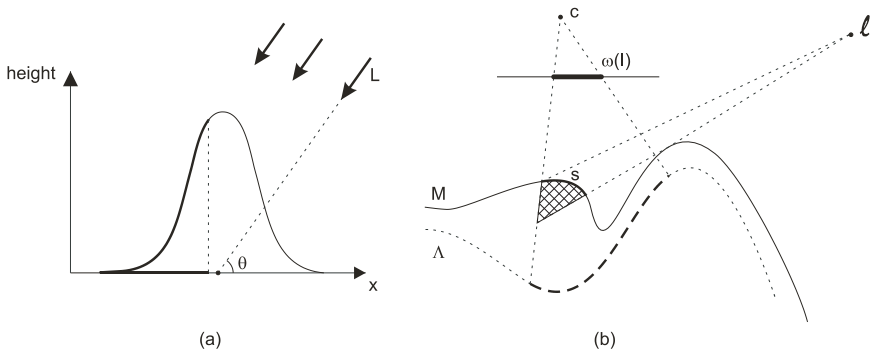

Fig. 12.9 (a) Surface to be reconstructed using the shadowgram technique. (b) Conservative shadow carving.

Besides, consistency for multiple views holds if only if it holds for each singular view. Finally, given an unknown surface Λ and a set of images taken under different lighting conditions, one can build the maximal surface[5] consistent with the extracted shadow information. Let $\Psi(\Lambda)$ denotes this surface, then it is obvious that it contains the actual surface Λ, since Λ is consistent with shadow information.

In [Savarese *et al.* (2007)] a carving approach is proposed to the problem of finding $\Psi(\Lambda)$. The algorithm, called *"Shadow Carving"*, computes first a coarse estimate of the surface using volume carving then it incrementally carves the model removing inconsistencies with self-shadow information. It is known, from Section 12.3.1 that volume carving computes a volume which certainly contains the original object. The subsequent carving based on shadow cue is performed in a conservative way, i.e., in such a way that the carved model will always contain the actual surface Λ.

Given the situation shown in Fig. 12.9(b) where Λ is the actual surface and M is its current estimates. Let (I, V) be a calibrated image, c the center of projection of V and $\omega(I)$ the shadow region on I generated by the light source ℓ. Let us call inconsistent shadow region s, the set of all surface points which are visible from both c and ℓ and such that they project in $\omega(I)$. Savarese *et al.* proved that the cross-hatched area in Fig. 12.9(b) can be removed from M in a conservative way, i.e., obtaining a new estimate that still contains the actual surface Λ.

[5] A maximal surface for a property Q is the surface which satisfied Q and contains every other surfaces that satisfied Q. In order to avoid degeneration, the maximal surface is typically upper bounded.

The major problem of all these algorithms is how to decide whether a surface point P lies on a shadow region or not. This is not a trivial task since it is difficult to distinguish low reflectance points from points in actual shadow regions. The camera only measures radiance coming from some point of the scene. Thus, low radiance measured in a particular direction can be due to a low reflectance (dark textured) point as well as to insufficient illumination. Moreover, insufficient illumination may be due to light sources too far from the object or to an actual shadow region. In the latter case, one must ensure that the shadow is generated by the object itself and not by other objects in the scene. Using only a single image, there is no way to distinguish between these four cases. Furthermore, even if a shadow region is detected, it difficult to accurately extract its boundaries, because shadows, in general, vanish gradually on the surface. Unfortunately, shadow detection plays an important role in reconstruction since small errors can lead to a totally incorrect reconstruction.

[Savarese *et al.* (2001)] propose a conservative shadow detection method, i.e., a technique which classifies a point as shadow only when it is certain that it is a shadow. The inverse condition is not required so that there can be shadow points classified as non-shadow. Obviously, the more shadow points are detected the more accurate is the reconstruction result. First of all, one must fix a threshold δ which separates light points from dark points. A point P of the surface is *"detectable"* if and only if in at least one picture it appears lighter than δ, otherwise it is *"undetectable"*. This provision ensures that P is not a low reflectance point, but unfortunately, it excludes many points not lighted by the actual light sources. For every image, a point is a shadow point if and only if it is *"detectable"* and it is darker than the threshold δ.

It is finally worth observing that, like shading information, also shadow is subject to the rules of the GBR [Kriegman and Belhumeur (2001)]. Therefore, even if the exact position of the light source is not known, one can reconstruct the observed surface up to a GBR transformation.

12.3.4 *Focus/Defocus*

There are two techniques to infer depth from a set of defocused images, called *"Shape from Focus"* (SfF) and *"Shape from Defocus"* (SfD). The first one, SfF, acquires a large number of images with small focal settings differences. On the other hand, the second one, SfD, needs only few differently focused images, typically two or three, in order to estimate depth

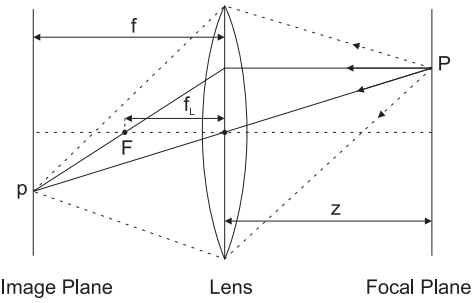

Fig. 12.10 Camera with lens: all the light rays coming from a point P in the focal plane are projected into a single point p in the image plane.

information. In both cases, defocused images are obtained by varying settings like the camera or the lens focal length, the aperture radius or the distance between the object to be acquired and the camera. Afterwards, depth is estimated by comparing the blurriness of different regions in the acquired images.

Both methods are based on the assumption that a defocused image is obtained by convolving the focused one with a kernel h^s_ϕ, called *point spread function* (PSF), that depends on the camera optic ϕ as well as on the scene shape s. Such an assumption comes from the observation that, since pinhole cameras with an infinitesimal aperture are not feasible, each point of the image plane is not illuminated by a single light ray but by a cone of light rays subtending a finite solid angle. Consequently, these points appear blurry. This effect can be reduced by a proper use of lenses. Indeed, it is well known that in this case, there exists a plane Π, called the *focal plane*, parallel to the retinal plane, the points of which are all in focus, or in other words, each point of Π projects into a single point of the image plane. The situation is shown in Figure 12.10, where z is the distance between Π and the center of the lens (the equivalent of the center of projection), f_L is the focal length of the lens and f is the camera focal length defined in Section 12.2. These quantities are related by the *thin lens equation*

$$\frac{1}{z} + \frac{1}{f} = \frac{1}{f_L} \tag{12.38}$$

Figure 12.10 shows that all light rays coming from a point P in the focal plane are projected into a single point p in the image plane. Consequently, an object is perfectly imaged only if it lies exactly on Π, otherwise, it

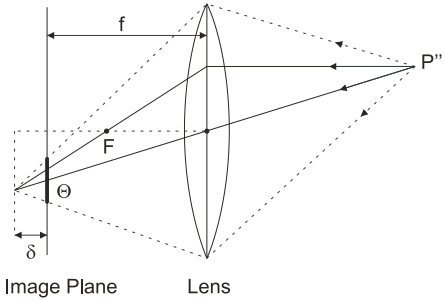

Fig. 12.11 All the light rays coming from a point P'' outside the focal plane are projected to a circular region Θ on the image plane.

appears blurred. As shown in Fig. 12.11, all the rays coming from a point P'' outside the focal plane are projected to a circular region Θ on the image plane.

The image of P'' can be modeled as the integral of the ideal image, where P'' is correctly imaged, weighted by a function (the PSF) which generates the blur effect. Therefore, the relationship between the actual image $\overline{I}$ and the ideal image where all the scene points are correctly imaged I is given by

$$\overline{I}(p) = \int h_{\phi}^{s}(p,q)\, I(q)\, dq \tag{12.39}$$

If the surface to be acquired is parallel to the focal plane then the PSF can be assumed to be shift invariant, i.e., $h_{\phi}^{s}(p,q) = h_{\phi}^{s}(p-q)$ and Eq. (12.39) can be rewritten as a convolution

$$\overline{I}(p) = \int h_{\phi}^{s}(p-q)\, I(q)\, dq = \left(h_{\phi}^{s} * I\right)(p) \tag{12.40}$$

As a first approximation, the blur intensity depends on the radius r of Θ, also known as the blurring radius, which is proportional to the distance δ between the actual image plane and an ideal one where P would be correctly imaged (see Fig. 12.11). More precisely,

$$r = \frac{\delta R}{f} \tag{12.41}$$

where R is the radius of the lens.

As mentioned above, both SfF and SfD estimate depth from Eq. (12.40). Namely, SfF identifies the regions of the input images where h_{ϕ}^{s} has not been

applied, i.e., the in-focus regions. Since h_ϕ^s is a low pass filter, a defocused region appears poor of high spatial frequency. Furthermore, if the surface to be acquired has high spatial frequency content, i.e., for instance it is a rough surface, a focused region can be recognized by analyzing its local Fourier transform.
The typical approach is to filter each input image $\overline{I}$ with a high pass FIR with impulse response ω and to evaluate the level of blur $v(p)$ of each point p as

$$v(p) = \int_{A_\varepsilon(p)} \left(\omega * \overline{I}\right)(q)\, dq \tag{12.42}$$

where $A_\varepsilon(p)$ is a neighborhood of p. Once these values are computed for a set of images $\left(\overline{I_1}, \ldots, \overline{I_n}\right)$, shape can be inferred by the following algorithm:

- Let $v_i(p)$ be the level of blur of the point p of image $\overline{I_i}$
- Let z_i be the depth of the focus plane related to $\overline{I_i}$
- For each point p, find j such that $j = \arg\max\{v_j(p)\}$ (i.e., find the image $\overline{I_j}$ with the sharpest representation of p)
- assign to p depth z_j

For a more precise reconstruction using gaussian interpolation the reader is referred to [Nayar and Nakagawa (1994)].

SfD methods instead, try to invert directly Eq. (12.40). The difficulty lies in the fact that neither h_ϕ^s nor I are known. Thus, *blind deconvolution* techniques are used in this task. Given a set of blurred images $\left(\overline{I_1}, \ldots, \overline{I_n}\right)$, from Eq. (12.40), one can write

$$\begin{aligned} \overline{I_1} &= h_{\phi_1}^s * I \\ &\vdots \\ \overline{I_n} &= h_{\phi_n}^s * I \end{aligned} \tag{12.43}$$

where ϕ_i is the optical setting used for image $\overline{I_i}$. Many strategies were developed to solve the above ill-posed problem. Classical approaches can be found in [Chaudhuri and Rajagopalan (1999)]. Effective variational and optimization approaches are due to [Jin and Favaro (2002)] and [Favaro and Soatto (2005)] respectively. In particular, in [Favaro *et al.* (2003)] shape is estimated by inferring the diffusion coefficient of a heat equation.

These methods are widely used in optical microscopy because microscopes have narrow depth of field; therefore, it is easy to obtain pictures containing both blurry and sharp regions.

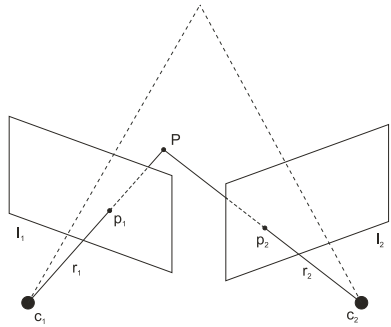

Fig. 12.12 From a pair of matched points p_1 and p_2, the 3D coordinates of point P can be computed by triangulation.

12.3.5 *Picture differences: stereo methods*

Algorithms which exploit the differences between two or more pictures of a scene are called *"stereo-matching algorithms"* [Marr and Poggio (1979)]. They are based on the same process used by human vision system to perceive depth, called *stereopsis*. For this reason, this particular depth cue is typically called stereo information.

In stereo methods, 3D reconstruction is accomplished in two main steps, the first addressing the so-called correspondences (or matching) problem and the second addressing the so-called reconstruction problem. The former recognizes if two or more points belonging to different images are the projection of the same point P of the 3D scene. The latter uses these correspondences in order to estimate the exact position of P in the 3D space.

Reconstruction task is achieved by triangulation. For example, let p_1 and p_2 be a pair of matched points in two different images I_1 and I_2 respectively. Thus, the real point P which they refer to, belongs to both the optical rays r_1 and r_2 related to p_1 and p_2 respectively. The situation is schematically depicted in Fig. 12.12. Therefore, P must lie at the intersection of r_1 and r_2. In practice, r_1 and r_2 may not intersect due to a imperfect camera calibration or to image discretization errors. The associated depth estimation problem in projective geometry is a linear overconstrained system with three unknowns and four independent equations which can be solved by least squared methods. Details can be found in any computer vision book, for instance in [Hartley and Zisserman (2000)].

Stereo methods were widely used in many applications; hence, various versions of these algorithms were developed in order to cope with different

types of practical challenges. Recent comparisons of existing stereo matching techniques can be found in [Scharstein and Szeliski (2002)] and in [Seitz *et al.* (2006)]. A classification of these techniques is not easy because of the number of characteristics to take into account. In the following, stereo methods will be presented according to a basic taxonomy distinguishing them with respect to baselines lengths, number of input images and type of correspondences used. A baseline is a segment connecting the centers of projection of a pair of cameras. Stereo methods which operate with long baselines are called *wide baseline stereo* methods, otherwise they are called *small baseline stereo* methods. Matching problem is different in these two situations. For example, perspective deformations effects can be ignored in the small baseline case but not in the wide baselines case.

Algorithms which use two, three and $n > 3$ images as input are called respectively *binocular stereo*, *trifocal stereo* and *multi-view stereo*. The use of multiple cameras simplifies the reconstruction task reducing errors in the 3D coordinates estimation; moreover, in many situations it eliminates matching ambiguities. In fact, one can use a third camera to check if an hypothetical match is correct or not.

Binocular stereo stores matching information in a map, called *disparity map*, which associates each pixel of the first input image with the matched pixel of the second input image as follows

$$p_2 = p_1 + d\,(p_1) \tag{12.44}$$

where p_1 and p_2 are the coordinates of the two matched pixels and d is the disparity map.

In a multi-view stereo algorithm the matching and reconstruction tasks are mixed together. Therefore, a disparity map is typically replaced by a complex internal scene representation, such as, a volumetric or a level-sets [Faugeras and Keriven (1998)] representation. In particular, using a volumetric representation, reconstruction is achieved by techniques like voxel coloring [Seitz and Dyer (2000)], space carving [Kutulakos and Seitz (2000)] and max-flow [Roy and Cox (1998); Vogiatzis *et al.* (2005b)]. Space carving applies the above mentioned carving paradigm. In this case, voxels are carved out if they do not project consistently into the set of input images. Therefore, starting from an initial estimate of the surface which includes the actual one, the algorithm finds the maximal surface, called *Photo Hull*, photo-consistent with all the input images. Instead, voxel coloring operates in a single pass through the entire volume of the scene, computing for each voxel a likelihood ratio used to determine whether this voxel belongs to the scene or not.

With respect to the type of correspondences used, an important family of algorithms, called *features based stereo* (FBS), concerns the methods which use image features as stereo information. A feature is a high level data structure that captures some information locally stored in an image. The most used features are edges and corners, but in the literature one can find many other higher order primitives such as regions [Cohen *et al.* (1989)] or topological fingerprints [Fleck (1992)]. It is important to note that a feature in the image space does not always correspond to a real feature in the 3D scene.
Restricting matching problem to a small set of a priori fixed features has two big advantages. First of all, features are not affected by photometric variations because they are simple geometrical primitives. Furthermore, since the correspondence search space is highly reduced, the matching task is speeded up. Unfortunately, any feature space gives a discrete description of the scene; thus, reconstruction results in a sparse set of 3D points.

Methods which perform matching between two or more points comparing the regions around them are called *area based stereo* (ABS) methods. These techniques are based on the assumption that given two or more views of the same scene, the image regions surrounding corresponding points look similar. This can be justified by the fact that since corresponding points are the projection of the same point P, their surrounding regions are the projection of the same piece of surface around point P. Therefore, what ABS methods do is to perform matching using only the local reflectance properties of the objects to be acquired.

A formal explanation requires some definitions. Let P be a point of surface M and denote by $A_\epsilon(P) \subset M$ the surface neighborhood of P with radius ϵ. Let (I_1, V_1) and (I_2, V_2) be two calibrated images, assume that P is visible on both images and let $(p_1, p_2) = (V_1(P), V_2(P))$ be a valid correspondence. Therefore, $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ are the projection of $A_\epsilon(P)$ on the image space of I_1 and I_2 respectively. Suppose that the cameras are placed in such a way that the shapes of the image regions $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ look similar, i.e., they are subject to a limited projective distortion. This can be achieved by a pair of parallel cameras with equal up-vectors (see Section 12.2) and small baseline/depth ratio. In other words, the surface to be acquired has to be far away from the point of views or/and the camera baseline has to be small. Assume that surface M, in $A_\epsilon(P)$, behaves as a pure Lambertian surface. Therefore, the radiance leaving $A_\epsilon(P)$ is independent of the viewpoint. Consequently, the image intensities acquired by the viewpoints V_1 and V_2 in $V_1(A_\epsilon(P))$ and

$V_2(A_\epsilon(P))$ must be equal, up to different camera optical settings (such as focusing, exposure or white balance). More formally, let

$$\begin{aligned} n_1(p_1) &= I_1|_{V_1(A_\epsilon(P))} \\ n_2(p_2) &= I_2|_{V_2(A_\epsilon(P))} \end{aligned} \tag{12.45}$$

be the image intensities around the corresponding points p_1 and p_2, i.e., the restrictions of the images I_1 and I_2 to respectively $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$. Since $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ can be supposed to be equal, images $n_1(p_1)$ and $n_2(p_2)$ are defined in the same domain up to different discretization of the image space. Therefore, one can make a one to one intensities comparison between $n_1(p_1)$ and $n_2(p_2)$ using simple similarity measures such as for example, *Sum of Squared Differences* (SSD), *Sum of Absolute Differences* (SAD) or *Intensity Correlation Distance* (ICD), respectively defined as:

$$\begin{aligned} SSD(p_1,p_2) &= \|n_1(p_1) - n_2(p_2)\|_2 \\ SAD(p_1,p_2) &= \|n_1(p_1) - n_2(p_2)\|_1 \\ ICD(p_1,p_2) &= \langle n_1(p_1), n_2(p_2)\rangle \end{aligned} \tag{12.46}$$

where $\|\cdot\|_1$, $\|\cdot\|_2$ and $\langle\cdot,\cdot\rangle$ are respectively the one-norm, the two-norm and the dot-product in function space. In order to make the above measures invariant to camera settings such as white balance and exposure, $n_1(p_1)$ and $n_2(p_2)$ should be replaced by their normalized versions $\overline{n_1}(p_1)$ and $\overline{n_2}(p_2)$, where

$$\overline{n}(p) = \frac{n(p) - \mu}{\sigma} \tag{12.47}$$

with μ the sample mean of $n(p)$ and σ^2 its sample variance.

If the above assumptions were satisfied, one could choose an arbitrary shape for image region $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ and compare them by one of the "metrics"of Eq. (12.46). Usually, square or rectangular shaped windows are preferred since they simplify the computation. Window size plays a crucial role in matching problem. Indeed, small windows are unable to solve matching ambiguities, while large windows make no longer valid the assumption of limited perspective distortion.

In synthesis, given a metric $D(\cdot,\cdot)$, the matching problem is reduced to finding all correspondences (p_1,p_2) such that $D(p_1,p_2)$ is less than a given threshold. Matching task is time expensive since it has to compare each pixel of each image with all the pixels of the other images. However, the knowledge of the calibration parameters can help to restrict the correspondence search space. Indeed, given a scene point P and its projection

Fig. 12.13 Left and right images of a stereo pair: ℓ is the epipolar line associated to p_1.

p_1 on the image I_1, then P certainly belongs to the optical ray r_1 related to p_1 as depicted in Fig. 12.12. Ray r_1 starts from the center of projection c_1 of the image I_1 and passes through p_1 in the image plane of I_1. Therefore, if p_2 is the projection of P on the second image I_2, then p_2 must belong to the projection of ray r_1 on I_2, i.e., it must belong to the half-line $\ell = V_2(r_1)$ called the epipolar line associated to p_1 (see Fig. 12.13). As a consequence, the correspondence search space related to point p_1 is reduced from a two-dimensional search domain to a one-dimensional one.

In order to improve speed in binocular stereo one may replace the two input images I_1 and I_2 with their rectified versions, i.e., the two equivalent pictures obtained with cameras positioned in such a way to have a common image plane parallel to the baseline and equal up-vectors. Such a process, known as *rectification*, is achieved by projecting the original images I_1 and I_2 into the new image plane. For a comprehensive tutorial on image rectification the reader is referred to [Fusiello *et al.* (1997)]. The main characteristic of a rectified image is that its epipolar lines are either all horizontal or all vertical, thus, the search for the correspondences can be performed only along rows or columns. In this case, disparity in Eq. (12.44) can be rewritten as

$$x_2 = x_1 + d(x_1, y_1), \qquad y_2 = y_1 \tag{12.48}$$

where $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. Consequently, the matching problem is reduced to the following maximization problem

$$d(x_1, y_1) = -x_1 + \arg\max\{D((x_1, y_1), (x_2, y_1)) \mid \forall x_2 \in [1, N_X]\} \tag{12.49}$$

where $D(\cdot, \cdot)$ is a generic similarity metric and N_X is the image width. Sometimes rectification is used also in multi-view stereo systems with appropriate adjustments.

The physics of the image formation process imposes that each image point has at most one corresponding point in each other image. Therefore, an ambiguity occurs when the solution of the maximization Problem (12.49) is not unique. Such an ambiguity can be solved by adding constraints to the problem, such as surface continuity, disparity bounds or disparity ordering constraint which the scene to be acquired may respect or not. The first type of constraints is obvious while the second says that $d(x_1, y_1)$ must be less than a given threshold for all possible values of (x_1, y_1). The third imposes that the ordering along the epipolar lines must be preserved. This last one allows one to use *dynamic programming* approaches to the matching problem as in [Meerbergen *et al.* (2002)].

Computation can be further speeded up if it is organized in a pyramidal structure. In this case, each image is partitioned into different resolution layers (e.g. a Gaussian or a Laplacian pyramid) and the 3D reconstruction is performed at each resolution. At the first iteration, the algorithm runs at the lowest resolution layer creating a first coarse estimate of the surface. At the subsequent stages, the correspondence search interval is restricted using information extracted at the previous layer so that the search is considerably simplified. A detailed account of this method can be found in [Menard and Brandle (1995)].

Unfortunately, the pure Lambertian assumption for the surface reflectance is too strict for general purpose, indeed objects with constant BRDF are rather rare while surfaces with some kind of specularity are much more common. Therefore, since the radiance reflected by a surface point P changes as a function of the point of view, image intensities $n_1(p_1)$ and $n_2(p_2)$ can be quite different. A typical example is the highlight on a specular surface which moves as the point of view moves. In order to face this problem, one can estimate the object radiance together with its shape as in [Jin *et al.* (2003)]. Another solution is proposed in [Yang *et al.* (2003)] which describes a similarity measure invariant with respect to the specularity effects.

Another difficulty in the matching task is due to the fact that it is not always possible to have $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ within limited projective distortions. Indeed, in general, they are only related by a projective transformation; thus, their shapes can differ in scale, orientation and so on. Sometimes rectification may help to reduce projective distortions. Several techniques were developed to avoid this problem. It is worth recalling the level set method proposed in [Faugeras and Keriven (1998)] which uses the current geometry estimate to infer shape and size of the matching windows

$V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$. This method iteratively refines the model geometry and performs the match with the estimated windows.

12.4 Multimodal methods

As previously mentioned, multimodal methods reconstruct the shape of an object from more than just one type of information. Since some methods work well in some situations but fail in others, the basic idea of multimodal methods is to integrate information not supplied by one method with that provided by the others. These methods hold the promise of reconstructing a wide range of objects, avoiding the restrictions characterizing individual monomodal methods. Furthermore, the possibility of measuring the same information in different ways allows us to reduce errors typical of specific methods. In short, the characteristics that make these methods superior to monomodal methods, are their robustness and the possibility of acquiring wider ranges of objects.

Unfortunately, the use of more types of information increases algorithmic and time complexity. Indeed, multimodal methods often need a computationally expensive final stage that fuses together all the data extracted and processed in the previous stages. In the literature there exist several ways to combine these data and the specific algorithms depend on the type of data to be fused. For example, [Vogiatzis *et al.* (2006)] proposes a method that combines silhouette and shading information. In particular silhouettes are employed to recover camera motion and to construct the visual hull. This is then used to recover the light source position and finally, the surface is estimated by a photometric stereo algorithm. In [White and Forsyth (2006)] a method is described that combines texture and shading cues. More precisely, this latter information is used to solve surface estimation ambiguities of the shape from texture algorithm.

However, most techniques combine multiple cues by classical paradigms like carving or optimization. In particular, as we mentioned before, the carving approach leads to a maximal surface consistent with all the extracted information and certainly including the actual surface. The idea behind multimodal methods based on carving, is to carve all voxels inconsistent with at least one type of information. *"Shadow carving"* and *"Space carving"* are examples of this approach combining respectively shadow and silhouette information and stereo and silhouette information.

On the other hand, the optimization paradigm minimizes a cost func-

tional that takes into account of all the various types of information, delivering as solution a surface fitting the extracted data as much as possible. More formally:

Problem 12.1. *Given Ω the set of all closed surfaces in $\Re^3$, i.e., the set of all the possible surfaces that can be reconstructed, and $(\alpha_1, \alpha_2, \ldots, \alpha_j)$ a j-tuple, where α_i is information of type i extracted from the input images, the multimodal fusion problem consists in finding M such that*

$$M = \arg\min \{\xi(M) \mid \forall M \in \Omega\} \tag{12.50}$$

where $\xi : \Omega \to \Re$ is the cost functional

$$\xi(M) = \kappa_{int} \cdot \xi_{int}(M) + \sum_i \kappa_i \cdot \xi_i(M, \alpha_i) \tag{12.51}$$

with ξ_{int} a cost functional that penalizes non-smooth surfaces and $\xi_i(\cdot, \alpha_i)$ functionals that penalize surfaces inconsistent with information α_i; κ_{int} and $\kappa_1, \ldots, \kappa_j$ are constants a priori fixed.

Consequently, the solution surface M will be as smooth as possible and consistent with as many data as possible. Constants κ_{int} and $\kappa_1, \ldots, \kappa_j$ balance the impact of the various types of information and the smoothness requirement.

Typically ξ_{int} is related to the mean or to the Gaussian curvature of the surface. For example, it can be defined as

$$\xi_{int} = \int_M \overline{\kappa} ds \tag{12.52}$$

where $\overline{\kappa}$ is the mean curvature.

Functionals $\xi_i(\cdot, \alpha_i)$ instead, depend on the type of information to which they are related. The literature reports many of such functionals accounting for a great variety of visual cues. An interesting functional which penalizes surfaces far from a generic cloud of points Σ is defined as

$$\xi_{cloud}(M) = \left(\int_M d_\Sigma(P)^k ds \right)^{\frac{1}{k}} \tag{12.53}$$

where $d_\Sigma(P)$ is the minimum distance between point $P \in M$ and the points of set Σ (see Fig. 12.14). Therefore, Eq. (12.53) can be used as one of the ξ_i, in order to penalize surfaces inconsistent with information extracted, for example, by the stereo-matching algorithm.

Let us observe that Eq. (12.53) accounts for the contribution $d_\Sigma(P)$ of the distance between each $P \in M$ and Σ. Therefore, a surface through

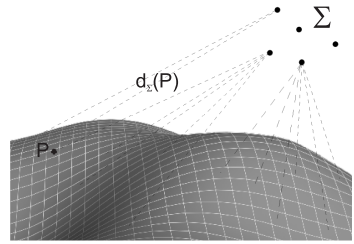

Fig. 12.14 In order to evaluate Eq. (12.53), one must measure the distance between Σ and each infinitesimal part of the surface.

empty regions of Σ is bound to have a high value of ξ_{cloud}. Consequently, the solution will be a surface that avoids those regions. This is not always desirable because the empty regions of Σ may be due to actual holes in the object or to failures of the stereo matching algorithm (e.g. in case of dark or poor texture areas).

Several works in the literature address the multimodal fusion problem by an optimization approach. [Wohler (2004)] uses both shading and shadow information to reconstruct the lunar surface. [Fua and Leclerc (1995)] fuse together stereo and shading. [Gheta *et al.* (2006)] use stereo and focus information. [Esteban and Schmitt (2004); Matsuyama *et al.* (2004); Sinha and Pollefeys (2005)] fuse stereo and silhouette. [Ballan and Cortelazzo (2006)] combine silhouette, stereo and shadow information.

Problem (12.1) can be solved in several ways, but, the current trends are the *max-flow/min-cut* and the *deformable models* techniques. Max-flow/min-cut techniques transform the fusion problem into a graph problem where the optimal surface is obtained as the minimum cut solution of a weighted graph. For a recent account see [Sinha and Pollefeys (2005)]. Instead, deformable models techniques [Kass *et al.* (1987); Osher and Fedkiw (2003)] solve Problem (12.1) by a gradient descent algorithm on the Euler-Lagrange equation obtained from functional ξ as described in the next section.

12.4.1 *Deformable models*

A *deformable model* is a manifold deforming itself under forces of various nature. Typically, but not always, these forces make the surface minimize an a priori fixed functional. These forces are classified as internals or ex-

ternals. The former are generated by the model itself and usually have an elastic nature while the latter depend on the specific problem to solve.

Deformable models appeared for the first time in [Kass *et al.* (1987)] within the definition of *snake* or *active contour*. A snake is a parametric curve $x(s)$ in the two-dimensional image space that deforms itself maintaining its smoothness and converging to the boundary of a represented object in the image. It is associated to a functional similar to the one of Eq. (12.51) with

$$\xi_{int}(x) = \frac{1}{2}\int_0^1 \left[\alpha \left|x'(s)\right|^2 + \beta \left|x''(s)\right|^2\right] ds \tag{12.54}$$

$$\xi_1(x) = -\int_0^1 \left|\bigtriangledown[G_\sigma * I](x(s))\right|^2 ds \tag{12.55}$$

where $I(x,y)$ is the image intensity function and $G_\sigma(x,y)$ the zero mean bi-dimensional gaussian function with standard deviation σ. Note that, in this case, the manifold M of Eq. (12.51) is replaced by the snake, $x(s)$, which is a specific parameterization of M.

Since their introduction, deformable models were used in many computer vision tasks, such as: edge-detection, shape modeling [Terzopoulos and Fleischer (1988); McInerney and Terzopoulos (1995)], segmentation [Leymarie and Levine (1993); Durikovic *et al.* (1995)] and motion tracking [Leymarie and Levine (1993); Terzopoulos and Szeliski (1992)]. Actually, in the literature there exist two types of deformable models: the *parametric* (or classical) one [Kass *et al.* (1987); Terzopoulos and Fleischer (1988); Cohen (1991)] and the *geometric* one [Caselles *et al.* (1993, 1995); Osher and Fedkiw (2003)]. The former are the direct evolution of snakes, while the latter are characterized by the fact that their surface evolution only depends on the geometrical properties of the model.

Geometrical framework is based on the level set methods. In this case, the model M is a surface in $\Re^3$, for which there exists a regular function $\psi : \Re^3 \to \Re$ and a constant $c \in \Re$ such that

$$M = \left\{x \in \Re^3 \mid \psi(x) = c\right\} = LevelSet^{\psi}(c) \tag{12.56}$$

In other words, M is the section of level c of a function $\Re^3 \to \Re$ (see Fig. 12.15). Besides, the forces are applied to ψ and not directly to M and only when convergence is reached, M is computed. Thus, both ψ and M evolve over time according to the partial differential equation

$$\begin{cases} \psi(0) = \psi_0 \\ \frac{\partial \psi}{\partial t}(t) = F(\psi(t)) \end{cases} \tag{12.57}$$

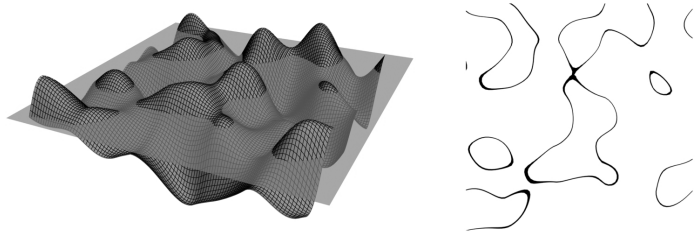

Fig. 12.15 Left: representation of $LevelSet^{\psi}(c)$ where $\psi : \Re^2 \to \Re$. Right: the result of the operation.

where $\psi(t)$ is the function ψ at time t, ψ_0 is its initial state and $F(\psi(t))$ is the force applied to ψ at time t. Hence, since this method operates only on ψ, surface M can dynamically change its topology. Roughly speaking, M can change its number of holes. For example, the reader could imagine to move upwards and downwards the plane of Figure 12.15, as the plane moves one obtains sections of ψ with a different number of connected components. Dynamic topology is the key feature that makes the geometrical framework a more powerful tool than the parametric one. The interested reader is sent to [Osher and Sethian (1988)] for further details.

The remainder of this section is focused on classical deformable model techniques. In this case, in order to solve the minimum problem researchers propose a standard variational approach based on the use of a gradient descent on the Euler-Lagrange equation obtained from functional ξ, which we explain by way of the following example.
Let s be a specific parameterization of M, i.e., s is a function from an open subset $A \subset \Re^2$ to $\Re^3$, and consider the functional

$$\xi(s) = \kappa_{int} \cdot \xi_{int} + \kappa_{cloud} \cdot \int_A d_\Sigma(s(u,v))dudv \tag{12.58}$$

where d_Σ is the same as in Eq. (12.53) and

$$\xi_{int} = \int_A \left\| \frac{\partial s}{\partial u} \right\|^2 + \left\| \frac{\partial s}{\partial v} \right\|^2 dudv + \int_A \left\| \frac{\partial^2 s}{\partial u^2} \right\|^2 + \left\| \frac{\partial^2 s}{\partial v^2} \right\|^2 + 2 \left\| \frac{\partial^2 s}{\partial v \partial u} \right\|^2 dudv \tag{12.59}$$

where the first term penalizes non-isometric parameterizations of M and the second term is equal to the total curvature of M if s is an isometry, thus penalizing non-smooth surfaces.

The related Euler-Lagrange equation [Fox (1987)] is:

$$-\nabla^2 s(u,v) + \nabla^4 s(u,v) - F_{cloud}(s(u,v)) = 0 \tag{12.60}$$

where $\nabla^2 s$, $\nabla^4 s$ are respectively the laplacian and the bi-laplacian of s and $F_{cloud} : \Re^3 \rightarrow \Re^3$ is a field that associates to each point P in the space a unit vector pointing to the point of Σ nearest to P.

The problem of finding M which minimizes ξ has been turned into the problem of finding s, a parameterization of M, which satisfies Eq. (12.60). Therefore, the solution can be computed by a gradient descent algorithm on the following problem

$$\arg\min \left\{ \left\| -\nabla^2 s\left(u,v\right) + \nabla^4 s\left(u,v\right) - F_{cloud}\left(s\left(u,v\right)\right) \right\|, \forall s \right\} \tag{12.61}$$

Consequently, this algorithm can be interpreted as the deformation of a parametric surface s subject to two forces defined as follows

$$F_{int} = \nabla^2 s - \nabla^4 s \tag{12.62}$$

$$F_{ext} = F_{cloud} \tag{12.63}$$

Let $s\left(t\right)$ be the model s at time t, therefore the evolution is described by the following partial differential equation

$$\begin{cases} s\left(0\right) = s_0 \\ \frac{\partial s}{\partial t}\left(t\right) = \beta \cdot \left(F_{int} + F_{ext}\right) \end{cases} \tag{12.64}$$

where s_0 is the initial surface and β determines the evolution speed. In order to find a numerical solution of Eq. (12.64), one can use forward Euler and apply the forces to all the vertices of the mesh. The discrete versions of ∇^2 and ∇^4 on a triangular mesh can be computed using the umbrella $\widetilde{\Delta}$ and the squared umbrella $\widetilde{\Delta}^2$ operators respectively [Esteban and Schmitt (2004)].

The advantages and the drawbacks of geometric and parametric deformable models can be summarized as follows. Geometric models have dynamic topology but are not easy to control. Their computation is typically slower than that of parametric models. On the other hand, parametric models have a fixed topology and suffer local minima problems in proximity of concavities. Their computation is faster and by a suitable parameters choice one can also control the parametric characteristics of the final mesh.

12.4.2 *Application examples*

Multimodal methods, in principle, can use any combination of the visual cues previously seen. Clearly, some combinations can be more effective and manageable than others. This section reviews two multimodal techniques recently proposed in the literature.

A method that combines silhouette and stereo information using classical deformable models is described in [Matsuyama *et al.* (2004); Esteban and Schmitt (2004)]. A first estimate s_0 of M is found by volume carving. Starting from s_0, the model evolves subject to three types of forces:

$$\frac{\partial s}{\partial t}(t) = \beta \cdot (F_{int} + F_{stereo} + F_{sil}) \tag{12.65}$$

where F_{int} is defined as above, F_{stereo} enforces stereo consistency and F_{sil} silhouette information.

In order to avoid local minima problems, [Esteban and Schmitt (2004)] define F_{stereo} as the *gradient vector flow* (GVF) [Xu and Prince (1998)] of Σ, that is a vector field solution of a diffusion equation.

Let $P_1, \ldots, P_m$ be the projections (silhouettes) of the real surface Λ viewed by $V_1, \ldots, V_m$ respectively and M be the mesh that currently approximates Λ. Let v be a vertex of mesh M, $F_{sil}(v)$ in [Esteban and Schmitt (2004)] is defined as

$$F_{sil}(v) = \alpha(v) \cdot d_{vh}(v) \cdot N(v) \tag{12.66}$$

where $N(v)$ is the surface normal in v, d_{vh} is the signed distance between the visual hull and the projection of vertex v, defined as

$$d_{vh}(v) = \min_j d(V_j(v), P_j) \tag{12.67}$$

where $d(V_j(v), P_j)$ is the signed distance between $V_j(v)$ and P_j, i.e., it is positive if v belongs to the visual hull and negative otherwise. $\alpha(v)$ is defined as

$$\alpha(v) = \begin{cases} 1 & if\ d_{vh}(v) \leq 0 \\ \frac{1}{(1+d(V_c(v),V_c(M)))^k} & if\ d_{vh}(v) > 0 \end{cases} \tag{12.68}$$

where $c = \arg\min_j d(V_j(v), P_j)$ and $V_c(M)$ is the projection of M viewed by V_c. This means that if v is outside the visual hull, $F_{sil}(v)$ is equal to $d_{vh}(v) \cdot N(v)$. Instead, if v is inside the visual hull, $\alpha(v)$ controls the transition of v from a contour point where $d(V_c(v), V_c(M)) = 0$ to a concave point where $d(V_c(v), V_c(M)) > 0$. In this way, F_{sil} reduces its intensity as much as v is inside the visual hull and parameter k controls the decreasing factor. Figure 12.16 exemplifies the situation.

As we can see in Figure 12.17(a) and in Fig. 12.17(b), silhouette information cannot describe model concavities which cannot be seen from the acquisition viewpoints, while stereo based methods fail in low variance regions and contours. Silhouette and stereo fusion Fig. 12.17(c) makes a

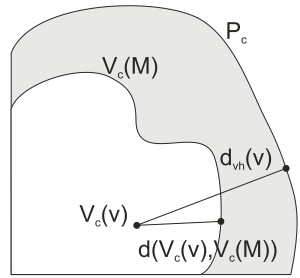

Fig. 12.16 Distances involved in F_{sil} computation.

better reconstruction of the original surface correcting errors and integrating information missing in each monomodal reconstruction. The final mesh turns out to be smooth and rather uniformly sampled.

An algorithm which combines stereo, silhouette and shadow information using the deformable model framework is proposed in [Ballan and Cortelazzo (2006)]. In particular F_{shadow}, i.e., the force related to shadow information, is defined in a way that minimizes the inconsistency with shadow information. In fact, like in the carving approach, the inconsistent surface portions (for example, portion s in Figure 12.9(b)) are pushed inside the surface. More formally,

$$F_{shadow}(v) = -i(v) \cdot N(v) \tag{12.69}$$

where $N(v)$ is the outer normal to the surface in v and $i(v)$ is a scalar function equal to 1 if the vertex v is inconsistent with shadow information, and equal to 0 otherwise.
Shadow information can improve the reconstruction obtained from just stereo and silhouette information; indeed, it can describe the shape of the concavities where stereo and silhouette information are missing.

12.5 Summary

This chapter presented an overview of passive optical 3D reconstruction methods as a tool for creating content. The main reason for the great attention that passive methods received in the literature probably rest in their acquisition speed (without intruding into the scene in any way, not even by just radiating energy) and in their inexpensive acquisition equipment (as simple as a digital camera). The counterpart of such a simplicity

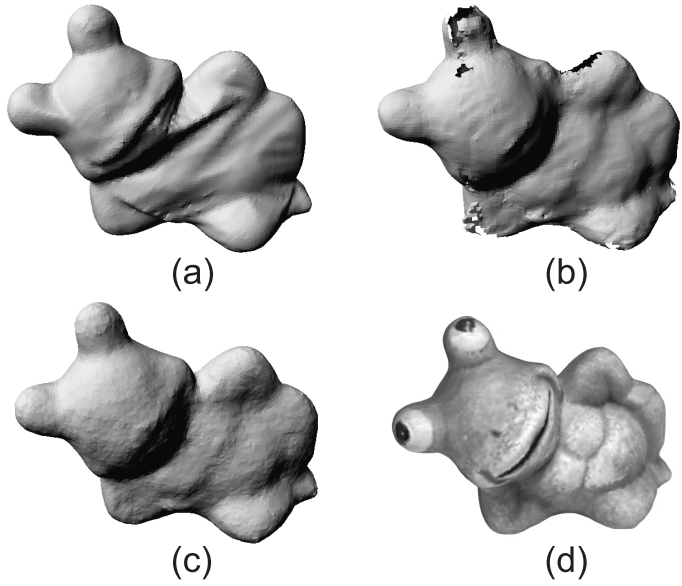

Fig. 12.17 Multimodal vs monomodal methods: (a) smoothed model obtained by volume carving; (b) model obtained by fusing together different partial models obtained by stereo matching; (c) model obtained by silhouette and stereo fusion; (d) model obtained by texturing model c.

is the complication that the reconstruction algorithms may have. However, algorithmic complexity, when needed, can be effectively handled by parallel computation or special hardware such as GPUs. In this context it may be worth recalling that passive methods are also employed for real time reconstructions of dynamic scenes, in applications such as interactive television, *3DTV* and *free-viewpoint video* [Matsuyama *et al.* (2004); Hilton and Starck (2004); Magnor (2005)].

While early research about passive methods concentrated in the discovery and exploration of the various types of visual cues available from images, the current research trend, instead, aims to blend together several types of information in order to overcome the limitations of monomodal reconstruction. This operation at present uses carving or optimization approaches for synergistically fusing the various information sources. Detailed examples of how to combine different visual cues were presented in Section 12.4. Multimodal passive methods, as expected, are more robust than monomodal methods with respect to errors and scene characteristics.

Appendix A

Appendix

Bibliography

Ballan, L. and Cortelazzo, G. M. (2006). Multimodal 3d shape recovery from texture, silhouette and shadow information, *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* .

Baumgart, B. G. (1974). Geometric modelling for computer vision, *PhD thesis, Standford University* .

Belhumeur, P., Kriegman, D. and Yuille, A. (1997). The bas-relief ambiguity, *Proceedings of IEEE International Conference on Computer Vision* , pp. 1060–1066.

Blinn, J. (1977). Models of light reflection for computer synthesized pictures, *SIGGRAPH* , pp. 192–198.

Bouguet, J.-Y. and Perona, P. (1999). 3D photography using shadows in dual-space geometry, *International Journal of Computer Vision* **35**, 2, pp. 129–149.

Brooks, M. and Horn, B. (1985). Shape and source from shading, *Proceedings of the International Joint Conference on Artificial Intelligence, Los Angeles* , pp. 932–936.

Caselles, V., Catte, F., Coll, T. and Dibos, F. (1993). A geometric model for active contours, *Numerische Mathematik* **66**, 1, pp. 1–31.

Caselles, V., Kimmel, R. and Sapiro, G. (1995). Geodesic active contours, *Proceedings 5th Internationa Conference Computer Vision* , pp. 694–699.

Chaudhuri, S. and Rajagopalan, A. N. (1999). *Depth from Defocus: a real aperture imaging approach* (Springer verlag).

Chuang, Y., Curless, B., Salesin, D. and Szeliski, R. (2001). A bayesian approach to digital matting, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* **2**, pp. 264–271.

Cipolla, R. and P.Giblin (2000). *Visual Motion of Curves and Surfaces* (Cambridge university press).

Cline, H. and Lorensen, W. (1987). Marching cubes: a high resolution 3D surface construction algorithm, *Computer Graphics* **21**, 4, pp. 163–168.

Cohen, L., Vinet, L., Sander, P. and Gagalowicz, A. (1989). Hierarchical region based stereo matching, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* , pp. 416–421.

Cohen, L. D. (1991). On active contour models and balloons, *CVGIP: Image Understand* **53**, pp. 211–218.

Daum, M. and Dudek, G. (1998). Out of the dark: Using shadows to reconstruct 3D surfaces, *Proceedings Asian Conference on Computer Vision, Hong Kong, China* , pp. 72–79.

Davis, J., Marschner, S., Garr, M. and Levoy, M. (2002). Filling holes in complex surfaces using volumetric diffusion, *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* , pp. 428–438.

de Berg, M., Kreveld, M. V., Overmars, M. and Shwarzkopf, O. (1999). *Computational Geometry* (Springer).

Durikovic, R., Kaneda, K. and Yamashita, H. (1995). Dynamic contour: A texture approach and contour operations, *Visual Computing* **11**, pp. 277–289.

Esteban, C. H. and Schmitt, F. (2004). Silhouette and stereo fusion for 3D object modeling, *Computer Vision and Image Understanding* **96**, 3, pp. 367–392.

Faugeras, O. and Keriven, R. (1998). Variational principles, surface evolution, PDE's, level set methods and the stereo problem, *IEEE Transactions on Image Processing* **7**, 3, pp. 336–344.

Favaro, P., Osher, S., Soatto, S. and Vese, L. (2003). 3d shape from anisotropic diffusion, *Conference on Computer Vision and Pattern Recognition* **1**, pp. 179–186.

Favaro, P. and Soatto, S. (2005). A geometric approach to shape from defocus, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 3, pp. 406–417.

Fleck, M. M. (1992). A topological stereo matcher, *International Journal of Computer Vision* **6**, 3, pp. 197–226.

Forsyth, D. A. (2002). Shape from texture without boundaries, *Proceedings of European Conference on Computer Vision* , pp. 225–239.

Fox, C. (1987). *An Introduction to the Calculus of Variations* (Dover Publications).

Fua, P. and Leclerc, Y. G. (1995). Object-centered surface reconstruction: combining multi-image stereo shading, *The International Journal of Computer Vision* **16**, 1, pp. 35–56.

Fusiello, A., Trucco, E. and Verri, A. (1997). Rectification with unconstrained stereo geometry, *Proceedings of the British Machine Vision Conference* , pp. 400–409.

Gheta, I., Frese, C. and Heizmann, M. (2006). Fusion of combined stereo and focus series for depth estimation, *Workshop Multiple Sensor Data Fusion, Dresden* .

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision* (Cambridge university press).

Hatzitheodour, M. and Kender, M. (1988). An optimal algorithm for the derivation of shape from shadows, *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition* , pp. 486–491.

Healey, G. and Binford, T. O. (1988). Local shape from specularity, *Computer Vision, Graphics and Image Processing* , pp. 62–86.

Hilton, A. and Starck, J. (2004). Multiple view reconstruction of people, *Interna-*

tional Symposium on 3D Data Processing, Visualization and Transmission (3DPVT) , pp. 357–364.

Horn, B. K. P. (1970). Shape from shading: A method for obtaining the shape of a smooth opaque object from one view, *PhD thesis, MIT* .

Ikeuchi, K. and Horn, B. (1981). Numerical shape from shading and occluding boundaries, *Artificial Intelligence* **17**, pp. 141–184.

Jebara, T., Azarbayejani, A. and Pentland, A. (1999). 3D structure from 2D motion, *IEEE Signal Processing Magazine* **16**, 3, pp. 66–84.

Jin, H. and Favaro, P. (2002). A variational approach to shape from defocus, *Proceedings of the European Conference on Computer Vision, Part II* , pp. 18–30.

Jin, H., Soatto, S. and Yezzi, A. (2003). Multi-view stereo beyond lambert, *Proceedings of IEEE conference on Computer Vision and Pattern recognition* , pp. 171–178.

Kass, M., Witkin, A. and Terzopoulos, D. (1987). Snakes: Active contour models, *International Journal of Computer Vision* **1**, pp. 321–331.

Koenderink, J. J. and Pont, S. C. (2003). Irradiation direction from texture, *Journal of the Optical Society of America* **20**, 10, pp. 1875–1882.

Kriegman, D. J. and Belhumeur, P. N. (2001). What shadows reveal about object structure, *Journal of the Optical Society of America* **18**, 8, pp. 1804–1813.

Kutulakos, K. N. and Seitz, S. M. (2000). A theory of shape by space carving, *International Journal of Computer Vision* **38**, 3, pp. 197–216.

Laurentini, A. (1994). The visual hull concept for silhouette based image understanding, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, 2, pp. 150–162.

Leibe, B., Starner, T., Ribarsky, W., Wartell, Z., Krum, D., Weeks, J., Singletary, B. and Hodges, L. (2000). Toward spontaneous interaction with the perceptive workbench, *IEEE Computer Graphics and Applications* **20**, 6, pp. 54–65.

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, R., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J. and Fulk, D. (2000). The digital michelangelo project: 3D scanning of large statues, *Proceedings of SIGGRAPH Computer Graphics* , pp. 131–144.

Leymarie, F. and Levine, M. D. (1993). Tracking deformable objects in the plane using an active contour model, *IEEE Transactions Pattern Analysis and Machine Intelligence* **15**, pp. 617–634.

Li, M., Magnor, M. and Seidel, H. (2003). Hardware-accelerated visual hull reconstruction and rendering, *Proceedings of Graphics Interface* , pp. 65–72.

Lucchese, L. and Mitra, S. K. (2001). Color image segmentation: A state of the art survey, *Proceedings of the Indian National Science Academy* **67**, 2, pp. 207–221.

Magnor, M. A. (2005). *Video-Based Rendering* (AK Peters, Ltd.).

Marr, D. and Poggio, T. (1979). A computational theory of human stereo vision, *Proceedings Royal Society of London* **204**, pp. 301–328.

Matsuyama, T., Wu, X., Takai, T. and Nobuhara, S. (2004). Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualizia-

tion for 3D video, *Computer Vision and Image Understanding* **96**, 3, pp. 393–434.

Matusik, W., Buehler, C. and McMillan, L. (2001). Polyhedral visual hulls for real-time rendering, *Proceedings of 12th Eurographics Workshop on Rendering* , pp. 116–126.

McInerney, T. and Terzopoulos, D. (1995). A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis, *Comput. Med. Imag. Graph.* **19**, pp. 69–83.

Meerbergen, G. V., Vergauwen, M., Pollefeys, M. and Van Gool, L. (2002). A hierarchical symmetric stereo algorithm using dynamic programming, *International Journal of Computer Vision* **47**, pp. 275–285.

Menard, C. and Brandle, N. (1995). Hierarchical area-based stereo algorithm for 3D acquisition, *Proceedings International Workshop on Stereoscopic and Three Dimensional Imaging, Greece* , pp. 195–201.

Mendoca, P. and Cipolla, R. (1999). A simple technique for self-calibration, *Proceedings of IEEE Conference on Computer Vision and Pattern recognition* **1**, pp. 500–505.

Nayar, S. K. and Nakagawa, Y. (1994). Shape from focus, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, 8, pp. 824–831.

Nicodemus, F. (1970). Reflectance nomenclature and directional reflectance and emissivity, *Applied Optics* **9**, pp. 1474–1475.

Osher, S. and Fedkiw, R. (2003). *Level Set Methods and Dynamic Implicit Surfaces*, *Applied Mathematical Sciences*, Vol. 153 (Springer).

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics* **79**, pp. 12–49.

Phong, B. T. (1975). Illumination for computer generated images, *Communications of the ACM* **18**, 6, pp. 311–317.

Piccardi, M. (2004). Background subtraction techniques: a review, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* **4**, pp. 3099–3104.

Potmesil, M. (1987). Generating octree models of 3D objects from their silhouettes in a sequence of images, *Computer Vision, Graphics and Image Processing* **40**, pp. 1–29.

Potmesil, M. (1990). *Introduction to statistical pattern recognition* (Academic press).

Prescott, B. and McLean, G. (1997). Line-based correction of radial lens distortion, *Graphical Models and Image Processing* **59**, 1, pp. 39–47.

Rioux, M., Blais, F., Beraldin, A., Godin, G., Blulanger, P. and Greenspan, M. (2000). Beyond range sensing: Xyz-rgb digitazing and modeling, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA* , pp. 111–115.

Roy, S. and Cox, I. J. (1998). A maximum-flow formulation of the n-camera stereo correspondence problem, *Proceedings of IEEE International Conference on Computer Vision* , pp. 492–502.

Rushmeier, H. and Bernardini, F. (2002). The 3D model acquisition pipeline, *Computer Graphics Forum* **2**, 2, pp. 149–172.

Savarese, S., Andreetto, M., Rushmeier, H., Bernardini, F. and Perona, P. (2007). 3D reconstruction by shadow carving: Theory and practical evaluation, *International Journal of Computer Vision* **71**, 3, pp. 305–336.

Savarese, S., Rushmeier, H. E., Bernardini, F. and Perona, P. (2001). Shadow carving, *ICCV* , pp. 190–197.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *International Journal of Computer Vision* **47**, 1, pp. 7–42.

Seitz, S., Curless, B., Diebel, J., Scharstein, D. and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* **1**, pp. 519–528.

Seitz, S. and Dyer, C. (2000). Photorealistic scene reconstruction by voxel coloring, *International Journal of Computer Vision* **38**, 3, pp. 197–216.

Shafer, S. and Kanade, T. (1983). Using shadows in finding surface orientations, *Computer Vision, Graphics and Image Processing* **22**, 1, pp. 145–176.

Sinha, S. and Pollefeys, M. (2005). Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation, *Proceedings of IEEE International Conference on Computer Vision* **1**, pp. 349–356.

Smith, E. and Kender, J. (1986). Shape from darkness: Deriving surface information from dynamic shadows, *In AIII* , pp. 664–669.

Tankus, A. and Kiryati, N. (2005). Photometric stereo under perspective projection, *Proceedings of IEEE International Conference of Computer Vision* , pp. 611–616.

Terzopoulos, D. and Fleischer, K. (1988). Deformable models, *Visual Computing* **4**, pp. 306–331.

Terzopoulos, D. and Szeliski, R. (1992). Tracking with kalman snakes, in A. Blake and A. Yuille (eds.), *Active Vision* (Eds. Cambridge, MA, MIT Press), pp. 3–20.

Torrance, K. E. and Sparrow, E. M. (1967). Theory for off-specular reflection from roughened surfaces, *Journal Of Optical Society Of America* **57**, pp. 1105–1114.

Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses, *IEEE Journal of Robotics and Automation* **3**, 5, pp. 323–344.

Vega, O. (1991). Default shape theory: with applications to the recovery of shape and light source from shading, *Master's thesis, University of Saskatchewan, Computational Sciecne Department* , pp. 1474–1475.

Vogiatzis, G., Favaro, P. and Cipolla, R. (2005a). Using frontier points to recover shape, reflectance and illumination, *Proceedings of IEEE International Conference on Computer Vision* , pp. 228–235.

Vogiatzis, G., Hernández, C. and Cipolla, R. (2006). Reconstruction in the round using photometric normals and silhouettes, *Proceedings IEEE Conference*

on Computer Vision and Pattern Recognition , pp. 1847–1854.

Vogiatzis, G., Torr, P. and Cipolla, R. (2005b). Multi-view stereo via volumetric graph-cuts, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* , pp. 391–398.

White, R. and Forsyth, D. A. (2006). Combining cues: Shape from shading and texture, *Conference on Computer Vision and Pattern Recognition* **2**, pp. 1809–1816.

Wohler, C. (2004). 3d surface reconstruction by self-consistent fusion of shading and shadow features, *Proceedings of 17th International Conference on Pattern Recognition* **2**, pp. 204–207.

Woodham, R. J. (1980). Photometric method for determining surface orientation from multiple images, *Optical Engineering* **19**, 1, pp. 139–144.

Xu, C. and Prince, J. L. (1998). Snakes, shapes, and gradient vector flow, *IEEE Transactions on Image Processing* , pp. 359–369.

Yang, D. K.-M. (1996). Shape from darkness under error, *Ph.D. thesis, Columbia University* .

Yang, R., Pollefeys, M. and Welch, G. (2003). Dealing with textureless regions and specular highlights - a progressive space carving scheme using a novel photo-consistency measure, *Proceedings of the 9th International Conference on Computer Vision* , pp. 576–584.

Zhang, R., Tsai, P., Cryer, J. and Shah, M. (1999). Shape from shading: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**, 8, pp. 690–706.