Research Assignment

Converting 2D to 3D: A Survey

Supervisors: Assoc. Prof. Dr. Ir. E. A. Hendriks
Dr. Ir. P. A. Redert

Information and Communication Theory Group (ICT)
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, the Netherlands

Qingqing Wei
Student Nr: 9936241
Email: weiqingqing@yahoo.com

December 2005

Title: Converting 2D to 3D: A Survey

Author: Q. Wei
Reviewers: E. A. Hendriks (TU Delft)
P. A. Redert

Project: Research Assignment for Master Program Media and Knowledge Engineering of Delft University of Technology

Customer: The Digital Signal Processing Group of Philips Research

Keywords: 2D to 3D conversion, depth cue, depth map, survey, comparison, 3D TV

Abstract: The survey investigates the existing 2D to 3D conversion algorithms developed in the past 30 years by various computer vision research communities across the world. According to the depth cues on which the algorithms reply, the algorithms are classified into the following 12 categories: binocular disparity, motion, defocus, focus, silhouette, atmosphere scattering, shading, linear perspective, patterned texture, symmetric patterns, occlusion (curvature, simple transform) and statistical patterns. The survey describes and analyzes algorithms that use a single depth cue and several promising approaches using multiple cues, establishing an overview and evaluating its relative position in the field of conversion algorithms.

.

Conclusion: The results of some 2D to 3D conversion algorithms are 3D coordinates of a small set of points in the images. This group of algorithms is less suitable for the 3D television application.

The depth cues based on multiple images yield in general more accurate results, while the depth cues based on single still image are more versatile.

A single solution to convert the entire class of 2D images to 3D models does not exist. Combing depth cues enhances the accuracy of the results. It has been observed that machine learning is a new and promising research direction in 2D to 3D conversion. And it is also helpful to explore the alternatives than to confine ourselves only in the conventional methods based on depth maps.

Content

- The picture on the cover is taken from http://www.ddd.com.

1 Introduction

Three-dimensional television (3D-TV) is nowadays often seen as the next major milestone in the ultimate visual experience of media. Although the concept of stereoscopy has existed for a long time, the breakthrough from conventional 2D broadcasting to real-time 3D broadcasting is still pending. However, in recent years, there has been rapid progress in the fields image capture, coding and display [1], which brings the realm of 3D closer to reality than ever before.

The world of 3D incorporates the third dimension of depth, which can be perceived by the human vision in the form of binocular disparity. Human eyes are located at slightly different positions, and these perceive different views of the real world. The brain is then able to reconstruct the depth information from these different views. A 3D display takes advantage of this phenomenon, creating two slightly different images of every scene and then presenting them to the individual eyes. With an appropriate disparity and calibration of parameters, a correct 3D perception can be realized.

An important step in any 3D system is the 3D content generation. Several special cameras have been designed to generate 3D model directly. For example, a stereoscopic dual-camera makes use of a co-planar configuration of two separate, monoscopic cameras, each capturing one eye's view, and depth information is computed using binocular disparity. A depth-range camera is another example. It is a conventional video camera enhanced with an add-on laser element, which captures a normal two-dimensional RGB image and a corresponding depth map. A depth map is a 2D function that gives the depth (with respect to the viewpoint) of an object point as a function of the image coordinates. Usually, it is represented as a gray level image with the intensity of each pixel registering its depth. The laser element emits a light wall towards the real world scene, which hits the objects in the scene and reflected back. This is subsequently registered and used for the construction of a depth map.

[1] Figure source: http://www.extra.research.philips.com/euprojects/attest/

current and past media data is in 2D format and should be possible to be viewed with a stereoscopic effect. This is where the 2D to 3D conversion method comes to rescue. This method recovers the depth information by analyzing and processing the 2D image structures. Figure 1 shows the typical product of 2D to 3D conversion algorithm – the corresponding depth map of a conventional 2D image. A diversity of 2D to 3D conversion algorithms has been developed by the computer vision community. Each algorithm has its own strengths and weaknesses. Most conversion algorithms make use of certain depth cues to generate depth maps. An example of depth cues is the defocus or the motion that could be present in the images.

This survey describes and analyzes algorithms that use a single depth cue and several promising approaches using multiple cues, establishing an overview and evaluating its relative position in the field of conversion algorithms. This may therefore contribute to the development of novel depth cues and help to build better algorithms using combined depth cues.

The structure of the survey is as follows. In Chapter 2, one or multiple representative algorithms for every individual depth cue are selected and their working principles are briefly reviewed. Chapter 3 gives a comparison of these algorithms in several aspects. Taking this evaluation into consideration, one relatively promising algorithm using certain investigated depth cues is chosen and described in more detail in Chapter 4. At the end, Chapter 5 presents the conclusion of the survey.

2 2D to 3D Conversion Algorithms

Depending on the number of input images, we can categorize the existing conversion algorithms into two groups: algorithms based on two or more images and algorithms based on a single still image. In the first case, the two or more input images could be taken either by multiple fixed cameras located at different viewing angles or by a single camera with moving objects in the scenes. We call the depth cues used by the first group the multi-ocular depth cues. The second group of depth cues operates on a single still image, and they are referred to as the monocular depth cues. The Table 1 summarizes the depth cues used in 2D to 3D conversion algorithms and their representative works. A review of algorithms using specific depth cue is given below.

Table 1: Depth Cues and Their Representative Algorithms

The Number of Input Images	***Depth Cues***	***Representative Works***
Two or More Images (binocular or multi-ocular)	Binocular disparity	Correlation-based, feature-based correspondence; triangulation [2][3]
	Motion	Optical flow [2]; Factorization [10]; Kalman filter [11]
	Defocus	Local image decomposition using the Hermite polynomial basis [4]; Inverse filtering [12]; S-Transform [13]
	Focus	A set of images of different focus level and sharpness estimation [5]
	Silhouette	Voxel-based and deformable mesh model [6]
One single image (monocular)	Defocus	Second Gaussian derivative [7]
	Linear perspective	Vanishing line detection and gradient plane assignment [8]
	Atmosphere Scattering	Light scattering model [15]
	Shading	Energy minimization [17]
	Patterned texture (Incorporates relative size)	Frontal texel [19]
	Symmetric patterns	Combination of photometric and geometric constraints [21]
	Occlusion	
	- Curvature	Smoothing curvature and isophote [22]
	- Single Transform	Shortest path [23]
	Statistical patterns	Color-based heuristics [8], Statistical estimators [25]

2.1 Binocular disparity

With two images of the same scene captured from slightly different view points, the binocular disparity can be utilized to recover the depth of an object. This is the main mechanism for depth perception. First, a set of corresponding points in the image pair are found. Then, by means of the triangulation method, the depth information can be retrieved with a high degree of accuracy (see Figure 2) when all the parameters of the stereo system are known. When only intrinsic camera parameters are available, the depth can be recovered correctly up to a scale factor. In the case when no camera parameters are known, the resulting depth is correct up to a projective transformation [2].

Assume p_l and p_r are the projections of the 3D point P on the left image and right image; O_l and O_r are the origin of camera coordinate systems of the left and right cameras. Based on the relationship between similar triangles (P, p_l, p_r) and (P, O_l, O_r) shown in Figure 2, the depth value Z of the point P can be obtained:

$$Z = f\frac{T}{d} \tag{2.1}$$

where $d = x_r - x_l$, which measures the difference in retinal position between corresponding image points. The disparity value of a point is often interpreted as the inversed distances to the observed objects. Therefore, finding the disparity map is essential for the construction of the depth map.

The most time-consuming aspect of depth estimation algorithms based on binocular disparity is the stereo correspondence problem. Stereo correspondence, also known as stereo matching, is one of the most active research areas in computer vision. Given an image point on the left image, how can one find the matching image point in the right image? Due to the inherent ambiguities of the image pairs such as occlusion, general stereo matching problem is hard to solve. Several constraints have been introduced to make the problem solvable. Epipolar geometry and camera calibration are the two most frequently used constraints. With these two constraints, image pairs can be rectified. Another widely accepted assumption is the photometric constraint, which states that the intensities of the corresponding pixels are similar to each other. The ordering constraint states that the order of points in the image pair is usually the same. The uniqueness constraint claims that each feature can have one match at most, and the smoothness constraint (also known as the continuity constraint) says that disparity changes smoothly almost everywhere. Some of these constraints are hard, like for example, the epipolar geometry, while others such as the smoothness constraints are soft. The taxonomy [3] of Scharstein and Szeliski together with their website "Middlebury stereo vision page' [9] have investigated the performance of approximately 40 stereo correspondence algorithms running on a pair of rectified images. Different algorithms impose various sets of constraints.

The current stereo correspondence algorithms are based on the correlation of local windows, on the matching of a sparse set of image features, or on global optimization. When comparing the correlation between windows in the two images, the corresponding element is given by the window where the correlation is maximized. A traditional similarity measure is the sum-of squared-differences (SSD). The local algorithms generate a dense disparity map. Feature-based methods are conceptually very similar to correlation-based methods, but they only search for correspondences of a sparse set of image features. The similarity measure must be adapted to the type of feature used. Nowadays global optimization methods are becoming popular because of their good performance. They make explicit use of the smoothness constraints and try to find a disparity assignment that minimizes a global energy function. The global energy is typically a combination of the matching cost and the smoothness term, where the latter usually measures the differences between the disparities of neighboring pixels. It is the different minimization step used in these algorithms which differentiates them from each other, e.g. dynamic programming or graph cuts.

2.2 Motion

The relative motion between the viewing camera and the observed scene provides an important cue to depth perception: near objects move faster across the retina than far objects do. The extraction of 3D structures and the camera motion from image sequences is termed as structure from motion. The motion may be seen as a form of “disparity over time”, represented by the concept of motion field. The motion field is the 2D velocity vectors of the image points, induced by the relative motion between the viewing camera and the observed scene. The basic assumptions for structure-from-motion are that the objects do not deform and their movements are linear. Suppose that there is only one rigid relative motion, denoted by$V$, between the camera and scenes. Let$P = [X, Y, Z]^T$ be a 3D point in the conventional camera reference frame. The relative motion$V$ between$P$ and the camera can be described as [2]:

$$V = -T - \omega \times P \tag{2.2}$$

where T andω are the translational velocity vector and the angular velocity of the camera respectively. The connection between the depth of 3D points and its 2D motion field is incorporated in the basic equations of the motion field, which combines equation (2.2) and the knowledge of perspective projection:

$$v_x = \frac{T_z x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f} \tag{2.3}$$

$$v_y = \frac{T_z x - T_y f}{Z} + \omega_x f - \omega_z y - \frac{\omega_y xy}{f} + \frac{\omega_x y^2}{f} \tag{2.4}$$

Where v_x and v_y are the components of motion field in x and y direction respectively; Z is the depth of the corresponding 3D point; and the subscripts x , y and z indicate the component of the x-axis, y-axis and z-axis directions. In order to solve this basic equation for depth values, various constraints and simplifications have been developed to lower the degree of freedom of the equation, which leads to the different algorithms for depth estimation, each suitable for solving problem in a specific domain. Some of them compute the motion field explicitly before recovering the depth information; others estimate the 3D structure directly with motion field integrated in the estimation process. An example of the latter is the factorization algorithm [10], where the registered measurement matrix, containing entries of the normalized image point coordinates over several video frames, is converted into a product of a shape matrix and motion matrix. The shape matrix registers the coordinates of the 3D object, and the motion matrix describes the rotation of a set of 3D points with respect to the camera. An introduction to explicit motion estimation methods is given below.

Dominant algorithms of motion field estimation are either optical flow based or feature based. Optical flow, also known as apparent motion of the image brightness pattern, is considered to be an approximation of the motion field. Optical flow subjects to the constraint that apparent brightness of moving objects remains constant, described by the image brightness constancy equation:

$$(\nabla E)^T v + E_t = 0 \tag{2.5}$$

where it is assumed that the image brightness is a function of image coordinates and the time. ∇E is the spatial gradients and E_t denotes the partial differentiation with respect to time. After computing the spatial and temporal derivatives of image brightness for a small $N \times N$ patch, we can solve (2.5) to obtain the motion field for that patch. This method is notorious for its noise sensitivity, which requires extra treatments such as tracking the motion across a long image sequence or imposing more constraints. In general, current optical flow methods yield dense but less accurate depth maps.

Another group of motion estimation algorithms is based on tracking separate features in the image sequence, generating sparse depth maps. Kalman filter [11] is for example a frequently used technique. It is a recursive algorithm that estimates the position and uncertainty of moving feature points in the subsequent frame.

It is worth to note that the sufficiently small average spatial disparity of corresponding points in consecutive frames is beneficial to the stability and robustness for the 3D reconstruction from the time integration of long sequences of frames. On the other hand, when the average disparity between frames is large, the depth reconstruction can be done in a way as that of binocular disparity (stereo). The motion field becomes equal to the stereo disparity map only if the spatial and temporal variances between frames are sufficiently small.

2.3 Defocus using more than two images

Depth-from-defocus methods generate a depth map from the degree of blurring present in the images. In a thin lens system, objects that are in-focus are clearly pictured whilst objects at other distances are defocused, i.e. blurred. Figure 3 shows a thin lens model of an out-of-focus real world point P projected onto the image plane. Its corresponding projection is a circular blur patch with constant brightness, centered at P'' with a blur radius of σ. The blur is caused by the convolution of the ideal projected image and the camera point spread function (PSF) $g(x, y, \sigma(x, y))$ where (x, y) are the coordinates of the image point P''. It is usually assumed that $\sigma(x, y) = \sigma$, where σ is a constant for a given window, to simplify the system and Gaussian function is used to simulate the PSF: $g_\sigma(x, y) = \frac{1}{\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}}$. In order to estimate the depth u, we need the following two equations. The fundamental equation of thin lenses describes the relation between u, v and f as:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \tag{2.6}$$

Pentland [12] has derived a relationship between the distance u (Figure 3) and the blur σ in equation (2.7):

$$u = \begin{cases} \dfrac{fs}{s - f - kf\sigma} & if\ u > v \\ \dfrac{fs}{s - f + kf\sigma} & if\ u < v \end{cases} \tag{2.7}$$

where u is the depth, v is the distance between the lens and the position of the perfect focus, s is the distance between the lens and the image plane, f is the focal length of the lens, and k is a constant determined by the lens system. Of these, s, f and k are camera parameters, which can be determined by camera calibration. Please note that the second case $u < v$ is possible to happen, for example, when $f < u < 2f$, based on the fundamental equation of thin lenses (2.6), we can obtain $v > 2f$, which yields thus $u < v$. With equation (2.7), the problem of computing depth u is converted into a task of estimating camera parameters and the blur parameter σ. When camera parameters can be obtained from camera calibration, the depth u can be computed from equation (2.7) once the blur parameter σ is known. The depth-from-focus algorithms focus thus on the blur radius estimation techniques.

Equation (2.7) indicates also when blur radius σ and other camera parameters except the focal length f are known, the depth u cannot be exactly determined. With 2 unknowns – u and f , equation (2.7) is under-constrained. In this case, the output signal can be a projection of an out-of-focus step edge, an in-focus smooth transition (e.g. a smooth texture) or infinite situations in between these two extremes [7]. This causes ambiguity when estimating the blur parameter. To tackle the problem, most of the depth- from-defocus algorithms reply on two or more images of the same scene taken from the same position with different camera focal settings to determine the blue radius. Once the blur radius is estimated and camera parameters are obtained from camera calibration, the depth can be computed by Equation (2.7).

The blur radius estimation techniques are based on, for example, inverse filtering [12], where the blur is estimated by solving a linear regression problem, or on S-Transform [13], which involves spatial domain convolution/de-convolution transform. Another example is the approach proposed by Ziou, Wang and Vaillancourt. It relies on a local image decomposition technique using the Hermite polynomial basis [4]. It is based on the fact that the depth can be computed once the camera parameters are available and the blur difference between two images, taken with different focal lengths, is known. The blur difference is retrieved by solving a set of equations, derived from the observation that the coefficients of the Hermite polynomial estimated from the more blurred image is a function of the partial derivation of the less blurred image and the blur difference.

[2] Figure source: reference [7]

2.4 Focus

The depth-from-focus approach is closely related to the family of algorithms using depth from defocus. The main difference is that the depth-from-focus requires a series of images of the scene with different focus levels by varying and registering the distance between the camera and the scene, while depth-from-defocus only needs 2 or more images with fixed object and camera positions and use different camera focal settings. Figure 4 illustrates the principle of the depth-from-focus approach [5]. An object with an arbitrary surface is placed at the translational stage, which moves towards the camera (optics) starting from the reference plane. The focused plane is defined by the optics. It is located at the position where all points on it are focused on the camera sensor plane. Let ‘s’ be a surface point on the object. When moving the stage towards the focused plane, the images of ‘s’ become more and more focused and will obtain its maximum sharpness when ‘s’ reaches the focused plane. After this, moving ‘s’ furthermore makes its image defocused again. During this process, the displacements of the translational stage are registered. If we assume that the displacement is d_{foused} when ‘s’ is maximally focused and the distance between the ‘focused plane’ and the reference plane is d_f, then the depth value of ‘s’ relative to the stage will be determined as $d_s = d_f - d_{focused}$. Applying this same procedure for all surface elements and interpolating the focus measures, a dense depth map can be constructed.

2.5 Silhouette

A silhouette of an object in an image refers to the contour separating the object from the background. Shape-from-silhouette methods require multiple views of the scene taken by cameras from different viewpoints. Such a process together with correct texturing

[3] Figure source: reference [5]

generates a full 3D model of the objects in the scene, allowing viewers to observe a live scene from an arbitrary viewpoint.

Shape-from-silhouette requires accurate camera calibration. For each image, the silhouette of the target objects is segmented using background subtraction. The retrieved silhouettes are back projected to a common 3D space (see Figure 5) with projection centers equal to the camera locations. Back-projecting a silhouette produces a cone-like volume. The intersection of all the cones forms the visual hull of the target 3D object, which is often processed in the voxel representation. This 3D reconstruction procedure is referred to as shape-from-silhouette.

Matsuyama [6] proposed an approach using parallel computing via a PC cluster system. Instead of computing the intersection of 3D cones directly, the 3D voxel space is partitioned into a group of parallel planes. Each PC is assigned a task to compute the cross section of the 3D object volume on one specific plane. By stacking up such cross sections, the voxel representation of the 3D object shape is reconstructed. In this way, the 3D volume intersection problem is decomposed into 2D intersection computation sub-problems which are concurrently carried out by all PCs. This leads to a promising speed gain. Furthermore, in order to capture the 3D object accurately, Matsuyama introduced a deformable mesh model, converting the 3D voxel volume into a surface mesh composed of triangular patches. According to a set of constraints, the surface mesh is deformed to fit the object surface. An example of the constraints is the 3D motion flow constraint, which requests that the mesh be adapted dynamically in conformity with object actions.

A shape-from-silhouette algorithm is often followed by a texturing algorithm. The visual hull is a geometry that encloses the captured object, but it does not capture the concave portion of the object that is not visible on the silhouette. Moreover, the number of views is often limited to make the processing time reasonable. This leads to a coarse geometry of the visual hull. Texturing assigns colors to the voxels on the surface of the visual hull and is therefore an indispensable step in creating realistic renderings.

[4] Figure source: reference [6]

2.6 Defocus using a single image

In section 2.3, depth-from-defocus algorithms based on two or more images are introduced. The reason for using more images is to eliminate the ambiguity in blur radius estimation when the focal setting of the camera is unknown. The images, with which this group of algorithms works, are required to be taken from a fixed camera position and object position but using different focal settings. However, only a small number of 2D video materials satisfy this condition. For example, the focus settings are changed when it is necessary to redirect the audience's attention from foreground to background or vice verse. To make defocus as a depth cue suitable for conventional video contents, where we do not have control of the focal settings of the camera, Wong and Ernst [7] have proposed a blur estimation technique using a single image based on the second derivative of a Gaussian filter [14]. When filtering an edge of blur radius σ with a second derivative of a Gaussian filter of certain variance s, the response has a positive and a negative peak. Denote the distance between the peaks as d, which can be measured directly from the filtered image. The blur radius is computed according to the formula $\sigma^2 = (\frac{d}{2})^2 - s^2$ (see Figure 6). With the estimated blur radius and the camera parameters obtained from camera calibration, a depth map can be generated that is based on equation (2.7). When the camera parameters are unknown, we can still estimate the relative depth level of each pixel based on its estimated blur radius by mapping a large blur value into a higher depth level and a smaller blur value to a lower depth level.

2.7 Linear perspective

Linear perspective refers to the fact that parallel lines, such as railroad tracks, appear to converge with distance, eventually reaching a vanishing point at the horizon. The more the lines converge, the farther away they appear to be. A recent representative work is the gradient plane assignment approach proposed by Battiato, Curti et al.. [8]. Their method performs well for single images containing sufficient objects of a rigid and geometric appearance. First, edge detection is employed to locate the predominant lines in the image. Then, the intersection points of these lines are determined. The intersection with

[5] Figure source: reference [7]

the most intersection points in the neighborhood is considered to be the vanishing point. The major lines close to the vanishing point are marked as the vanishing lines. Between each pair of neighboring vanishing lines, a set of gradient planes is assigned, each corresponding to a single depth level. The pixels closer to the vanishing points are assigned a larger depth value and the density of the gradient planes is also higher. Figure 7 illustrates the process and the resulting depth map where a darker grey level indicates a large depth value.

2.8 Atmosphere scattering

The earth is enveloped by a vast amount of air known as atmosphere. The propagation of light through the atmosphere is affected in the sense that its direction and power is altered through a diffusion of radiation by small particles in the atmosphere. This leads to the phenomenon called atmosphere scattering, also known as haze, which causes various visual effects: distant objects appear less distinct and more bluish than objects nearby; a flash light beam is diffused in a foggy environment.

Although atmosphere scattering is a classical topic of physics often referred as one of the major cues in human depth perception in psychology, little literature can be found in the field of computer vision on the matter directly converting the atmosphere scattering to depth information directly. Cozman and Krotkov [15] presented the first analysis of this conversion in 1997. It was based on Lord Rayleigh's 1871 physical scattering model. Their algorithm is suitable for estimating the depth of outdoor images containing a portion of sky. After simplifying the complex physics model, the following relationship is derived between the radiance of an image and the distance between the object and the viewer:

$$\tilde{C} = C_0 e^{-\beta z} + S(1 - e^{-\beta z}) \tag{2.8}$$

[6] Figure source: reference [8]

where $\tilde{C}$ is the measured intensity of an object, C_0 is the intensity of the object in the absence of scattering, β is the extinction coefficients, z is the depth of the object, and S is the sky intensity which is the intensity of an area in which objects are indistinguishable. Equation (2.8) contains two parts combined by addition. The first part describes the fact that the light power is attenuated when a light beam is projected onto an object through a scattering medium. The second part reflects the opposite phenomenon: an actual gain in intensity due to scattering. When a light ray is reflected from a scattering medium to the viewer, scattering events take place at each point of the light ray, and divert the light from its original path. As light reaches the viewer from all points of the light ray, the viewer actually perceives a new source of light.

In most of the cases, β and C_0 are unknown, S can be measured from any images that contain a sky region. For indoor scenes, the estimation of S needs experimental water vapor generation setting up, which cannot be realized automatically. This is one of the limitations of this algorithm. The algorithm results in a ratio of depth difference between different objects, from which a sparse depth map can be derived.

2.9 Shading

The gradual variation of surface shading in the image encodes the shape information of the objects in the image. Shape-from-shading (SFS) refers to the technique used to reconstruct 3D shapes from intensity images using the relationship between surface geometry and image brightness. SFS is a well-known ill-posed problem just like structure-from-motion, in the sense that the resolution may not exist, the solution is not unique or it does not depend continuously on the data [2]. In general, SFS algorithms make use of one of the following four reflectance models: pure Lambertian, pure specular, hybrid and more complex surfaces [16], of which Lambertian surface is the most frequently applied model because of its simplicity. A uniformly illuminated Lambertian surface appears equally bright from all viewpoints. Besides the Lambertian model, the light source is also assumed to be known and orthographic projection is usually used. Assume $p=[x,y]^T$ is the image of the 3D point $P=[X,Y,Z]$, where the Z-axis of the camera is the optical axis, and the depth Z of point P can be described as a function of its image coordinates $Z=Z(x,y)$. Then the surface slopes p and q can be computed by taking the x and y partial derivatives of the vector $[x,y,Z(x,y)]$, that is, $p=[1,0,\frac{\partial Z}{\partial x}]^T$ and $q=[0,1,\frac{\partial Z}{\partial y}]^T$. The relationship between the estimated reflectance map $R(p,q)$ and the surface slopes offers the starting point to many SFS algorithms:

$$R(p,q)=\frac{\rho}{\sqrt{1+p^2+q^2}}i^T[-p,-q,1] \tag{2.9}$$

where ρ is the surface albedo, a parameter of the surface's material, and i denotes the direction and the amount of incident light. After estimating the value of ρ and i, solving

(2.9) will yield the desired depth map $Z = Z(x, y)$ and surface slopes. However, the direct inversion of (2.9) appears to be a difficult ill-posed task because it is a nonlinear partial differential equation with uncertain boundary conditions. Extra constraints, such as the smoothness constraint, are therefore necessary. Within the framework of variational calculus, the problem is converted into a well-formulated global minimization task by looking for the minimum of energy function ε , where

$$\varepsilon = \iint_{\Omega} (E(x, y) - R(p, q))^2 + \lambda({p_x}^2 + {p_y}^2 + {q_x}^2 + {q_y}^2))dxdy \qquad (2.10)$$

where $E(x, y)$ denotes the image brightness of pixel (x, y) and p_j represents $\partial p/\partial j$. Note that the first term in the integral (2.10) reflects the brightness constraint and the second term controls the smoothness of the surface.

A variety of methods have been developed for solving this minimization problem. A straightforward example is the Euler-Lagrange equations [2]. A recent study carried out by Kang [17] is based on finite elements. The image is divided into small triangular patches. The reflectance map $R(p, q)$ is then approximated by a linear function. The depth map is obtained by converting the energy minimization model into a simpler problem of the form of solving a linear equation iteratively until a specified error threshold is reached.

2.10 Patterned texture

Patterned texture offers a good 3D impression because of the two key ingredients: the distortion of individual texels and the rate of change of texel distortion across the texture region. The latter is also known as texture gradient. The shape reconstruction exploits distortions such as perspective distortion, which makes texels far from the camera appear smaller, and/or foreshortening distortion, which makes texels that are not parallel to the image plane shorter.

In general, the output of shape-from-texture algorithms is a dense map of surface normals. This is feasible for recovering the 3D shape under the assumption of a smooth textured surface. As a rule, the shape of a surface at any point is completely specified by surface's orientation and curvatures. Since curvature estimation turns out to be

[7] Figure source: reference [19]

complicated, the shape-from-texture algorithms focus on the determination of textured surface's orientations in terms of surface normals. It is also worth noting that a lot of real-life images contain differently textured texture regions or textured areas surrounded by non-textured ones. These different textured regions need to be segmented before most shape-from-texture algorithms can be applied. Figure 8 shows a typical shape reconstruction process from textures. This group of methods, which require texture segmentation, belongs to the feature-based approach. In more recent years, there has been a shift toward Shape-from-Texture methods that utilize spectral information and avoids prior feature detection. These methods compare the spectral representation of windowed image patches to recover orientation. Commonly used spectral representations are the Fourier transform, wavelet decomposition and the Gabor transform.

Shape-from-texture is again an under-constrained problem. Most algorithms are designed to tackle specific group of textures [18]. The frequently used assumption is a combination of the following simplifying characters of textures: homogeneity, meaning that the texels are uniformly distributed; isotropy, indicating that the texels have a constant inertia about each axis; and stationary, denoting that the texels differ from each other only by a translation on the surface and no rotation is involved.

The conventional shape-from-texture algorithms appear to be fairly restrictive due to all these simplifying assumptions. Some researchers have been working on more versatile alternatives. Loh and Hartley [19] recently proposed a method suitable for perspective views, which is claimed to be the first algorithm not subjected to the above-mentioned three constraints. Their algorithm is based on establishing the frontal texel as the reference point (see Figure 9). It is the single texel viewed frontally in an image. The frontal texel is unique. Any incorrect hypothesis of the frontal texel leads to inconsistent estimates of the surface orientation, which cannot be realized by a reconstructed surface. Therefore, a search through all possible frontal texels with the surface consistency constraint yields a unique fontal texel estimate. Next to the frontal texel, other texels in a texture are considered to undergo an affine transformation with respect to the frontal texel. By estimating the affine transformation of each texel, the surface orientation at the texel location can be recovered.

In fact, as a single depth cue texture incorporates another important dept cue - relative size. A human's perception of depth is based on his or her experience and familiarity with the similar objects. As the boat comes closer, the retinal image becomes larger and larger. We interpret this as the boat getting closer (see Figure 10). Although it is not completely good metaphor, we might think of the boat as a texel, and the rate of change in the size of the boat as the texture gradient.

2.11 Bilateral symmetric pattern

Symmetric patterns often appear in natural or man-made scenes. Faces, animals or various man-made objects are all examples of this. The idea behind 3D reconstruction based on symmetric patterns is that a single non-frontal image of a bilaterally symmetric object can be viewed as two images of this object from different view angles. Francois et al. introduced in [20] a method to extract the corresponding stereo pair from a single perspective non-frontal view of a symmetric scene. Then traditional 2-view stereo theory based on binocular disparity is applied to retrieve the depth information. Some algorithms also take the different illumination directions of these two images into consideration, such as the approach proposed by Shimshoni, Moses and Lindenbaumlpr [21]. The following section expands on the principle of their algorithm.

Two local constraints are applied: symmetry-induced geometric constraints (stereo) and photometric constraints (shading). The connection between these two constraints is the postulation that the surface normal of each point found by using geometric constraints should be equal to the one based on photometric constraints. Combing these two constraints, the key role in the algorithm – the continuous correspondence function, which computes the coordinates of the corresponding symmetric point given those of the original point, can be induced. 3D reconstruction is achieved by propagating such correspondence functions to all symmetric points in the image.

The geometric constraints reflect the inherent geometry in symmetric patterns. The world coordinate system is chosen so that the symmetry plane, the plane which separates the object into two identical parts, is the $y \times z$ plane, and all lines which connect pairs of

[8] Figure source: “Sensation and Perception Tutorials”, http://psych.hanover.edu/Krantz/art/rel_size.html

symmetric points are parallel to the x axis (see Figure 11). Assume that the object points $p^r = (x, y, z)^T$ and $p^l = (-x, y, z)^T$ are a pair of symmetric points on a bilaterally symmetric object; $\tilde{p}^l = (\tilde{x}, \tilde{y})$ and $\tilde{p}^r = (C(\tilde{x}, \tilde{y}), \tilde{y})$ are their 2D projections on the images, where $C(\tilde{x}, \tilde{y})$ is the correspondence function of the point $(\tilde{x}, \tilde{y})$. Once $C(\tilde{x}, \tilde{y})$ is obtained, the correspondence functions of its neighboring points can be deduced by applying the Taylor expansion of $C(\tilde{x}, \tilde{y})$:

$$C(\tilde{x} + h, \tilde{y} + g) = C(\tilde{x}, \tilde{y}) + h\tfrac{\partial C}{\partial x} + g\tfrac{\partial C}{y} \tag{2.11}$$

Under the assumption of the orthographic projection, the relation between the 3D points and their image counterparts can be formulated by:

$$\tilde{p} = sRp + t \tag{2.12}$$

where s represents a scale factor, R is a rotation matrix with parameter θ, which denotes the viewing direction of the camera, and t is the translation matrix. By assuming $s = 1$ and $t = 0$, expanding (2.12) yields the result (2.13) with an acceptable ambiguity of the reconstructed object up to an affine transformation. Equation (2.13) serves as the geometric constraints.

$$\begin{aligned} x &= \tfrac{1}{2\cos\theta}(\tilde{x}^r - \tilde{x}^l) \\ y &= \tilde{y}^r \\ z &= \tfrac{1}{2\sin\theta}(\tilde{x}^r + \tilde{x}^l) \end{aligned} \tag{2.13}$$

The photometric constraints focus on the shading information in the image. As in shape-from-shading approaches, a typical illumination model is assumed: a Lambertian surface and a distant light source. Let $\hat{n} = (n_x, n_y, n_z)^T$ be the normal at the surface point of the object, $\hat{l} = (l_x, l_y, l_z)^T$ be the illumination direction, and ρ be the product of surface albedo and the intensity of the light source. The normals of the two symmetric points can be denoted as $\hat{n}^r = (n_x, n_y, n_z)^T$ and $\hat{n}^l = (-n_x, n_y, n_z)^T$. The photometric constraints are:

$$\begin{pmatrix} I_{\Delta} \\ I_{\Sigma} \end{pmatrix} = \begin{pmatrix} I^{r} - I^{l} \\ I^{r} + I^{l} \end{pmatrix} = 2\rho \begin{pmatrix} l_x & 0 & 0 \\ 0 & l_y & l_z \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = R_l \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \qquad (2.14)$$

The correspondence function can be computed based on (2.13) and(2.14). Starting from location p_0, making a circular tour, which returns to p_0, the correspondence function computed at the end point p_0 should be equal to that at the starting point p_0. The parameter values, which minimize the change in the correspondence function in such a circular tour, are then used for the 3D reconstruction process.

2.12 Occlusions

The principle of depth-from-occlusion algorithms has its roots in the phenomenon that an object which overlaps or partly obscures our view of another object is considered to be closer. Occlusion is also known as interposition and offers rich information in relative depth ordering of the objects. Curvature [22] and single transform [23] are two depth cues which might be grouped under the header of occlusion due to their inherent characteristics related to occlusion.

Curvature

Curvature is a depth cue based on the geometry and topology of the objects in an image. The majority of objects in 2D images have a sphere topology in the sense that they contain no holes, such as closed grounds, humans, telephones etc. It is observed that the curvature of object outline is proportional to the depth derivative and can thus be used to retrieve the depth information. Figure 13 shows the process of depth-from-curvature algorithm. The curvature of points on a curve can be computed from the segmentation of the image. A circle has a constant curvature and thus a constant depth derivative along its boundary, which indicates that it has a uniform depth value. A non-circle curve such as a square does not have a constant curvature. A smoothening procedure is needed in order to obtain a uniform curvature/depth profile. After the smoothing process, each object with an outline of uniform curvature is assigned one depth value.

An efficient way to apply the depth-from-curvature approach is to appeal to isophotes, which are used to represent the outlines of objects in an image. An isophote is a closed curve of constant luminance and always perpendicular to the image derivatives. An isophote with isophote value T can be obtained by thresholding an image with the threshold value that is equal to T. The isophotes then appear as the edges in the resulting binary image. Figure 14 shows how to derive the depth map for isophotes of value T. The topological ordering can be computed during the process of scanning the isophote image and flood-filling all 4-connected regions. First the image border pixels are visited. Each object found by flood-filling is assigned depth 0. Any other object found during the scan is then directly assigned a depth value equal to one plus the depth from the previous scanned pixel. In the end, a complete depth map of isophotes with a value of T is obtained. Repeating this procedure for a representative set of values of T, e.g. [0, 255], for an image, the final depth map is computed by adding or averaging all the T depth maps.

Note that isophotes usually occur in grey-scale images. In the case of color images, images of one or more color components are first extracted and each undergoes the entire operation separately. The results are then combined by averaging the depth maps.

Simple transform

The simple transform method creates a dense depth map by means of a straightforward transform from the pixel intensities. Given the intensity image of the original image, the intensities of pixels directly above the domain of interest are assigned value 0 and all other pixels outside the domain are assigned the intensity value ∞ (see Figure 15 to the left).

For each pixel, paths are constructed from the pixel to an arbitrary pixel at the top of the image domain. From pixel q to its neighboring pixel r , the step cost is the absolute value of intensity difference between the two pixels. A path from pixel p to a top pixel has a path cost equal to the sum of all the step costs it makes between these two ends. Various paths from a certain pixel p to the image top boundary exist. The depth of p is assigned the smallest path cost from itself to the top of the image.

2.13 Statistical patterns

Statistical patterns are the elements which occur repeatedly in images. When the number or the dimension of the input data is large, machine learning techniques can be an effective way to solve the problems. In recent years, as a tool to estimate depth maps, machine learning has been receiving increasing interest. Especially supervised learning making use of training data with the ground truth appears highly advantageous to the field of 2D to 3D conversion. As well as a set of representative and sufficient training data, good features and suitable classifiers are all essential ingredients for satisfactory results.

With statistical pattern recognition techniques, it is even possible to estimate the absolute average depth given a single image, even thought absolute depth estimation was claimed to be the plus point of multi-ocular cues such as binocular disparity or motion. Torralba and Oliva [24] looked into the Fourier domain of images and discovered that certain features like the global spectral signature of an image, which denotes the mean amplitude spectrum, are closely related to the average absolute depth. They applied the EM algorithm (Estimation-Maximization) to find the conditional probability function (PDF) of the mean depth, given the measured image features. The mean depth is then estimated by plugging the resulting PDF into a mixture model of linear regressions or simply by searching the depth value with the maximum likelihood.

If the mean depth of an image can be estimated by machine learning, then following the same way of thinking, it must be possible to estimate the depth value per pixel. A recent work of Saxena et al. proves this postulation [25]. Firstly, a set of global and local features, such as texture variations, texture gradients, haze (atmosphere scattering), image features extracted at multiple image scales etc., is collected. Secondly, the Markov Random Field (MRF) is chosen to model the conditional PDF - $P(\text{depth}|\text{features, model parameters})$. MRF is chosen because of its superior ability in modeling correlated data: the depth of a particular patch is close related to depth of its neighboring patches and also that of some of the non-immediate neighbors. The latter is tackled by incorporating the depths at multiple scales into the conditional PDF. Thirdly, the model parameters are learned using the maximum likelihood estimator and linear regression based on the training data. Finally, the estimated depth $\hat{d}$ is obtained as the maximum a posteriori estimate (MAP) of the depths $d's$, shown in equation(2.15):

$$\hat{d} = \arg\max_{d} \left\{ P(d \mid \text{features}, \text{model parameters} \right\} \qquad (2.15)$$

Machine learning is also helpful in understanding the image semantics. Knowledge of image content helps to enhance the accuracy of estimated depth. In the work of Battiato et al. [8], a group of color-based rules is derived using a large set of training images to categorize the color regions in the image to specific groups: Sky, Farthest Mountain, Far Mountain, Near Mountain, Land and Other. Then each category is assigned a depth level so that a coarse qualitative depth map is formed. Based on this information of categorization, the input image goes further through the image classification procedure, where the entire image is classified as one of the three types: Outdoor/Landscape, Outdoor with geometric appearance or Indoor. Applying a depth estimation technique based on linear perspective, tailored to each of these three classes, a dense depth map can be reconstructed. The algorithm will be explained in further detail in Chapter 4.

2.14 Other depth cues

Apart from the depth cues described in this chapter, which are fairly dominant in the current computer vision field, a number of other depth cues with different principles exist and have also been successfully translated into algorithms, for example, shadow, dynamic occlusion, static occlusion based on T-junction, and so on. Due to reasons like intellectual property rights and the limited scope of the survey, these depth cues are not investigated here.

3 Comparison

Fair and effective performance evaluation of 2D to 3D conversion algorithms requires careful design of criteria and data sets. However, there is a lack of uniformity in the framework based on which methods are evaluated. Furthermore, a lot of papers do not provide an explicit quantitative performance analysis, which complicates the evaluation process. It is thus imprudent to make explicit claims such as which methods indeed have the lowest error rates or which methods are the fastest. Implementing these algorithms and evaluating them based on common data sets and the same performance standards is beyond the scope of this survey. Another complication is that for each individual depth cue, there are a vast number of algorithms, each having different characteristics and performances. In order to grasp the general principle of each depth cue, efforts have been made in Chapter 2 to choose a number of representative algorithms for each depth cue. Therefore, when discussing issues such as accuracy or operating speed of the algorithms of each depth cue, we resort, if available, to the experimental results present in the papers of these representative algorithms. Only in the case when no performance data is available in these representative works, attempts have been made to find qualitative or quantitative data from other algorithms that belong to the same depth cue.

The comparison is based on 9 qualitative aspects. Some of them are correlated with each other. The results are presented in Table 2 and Table 3. This chapter is thus dedicated to evaluating and clarifying the results in the various aspects in the comparison table.

1. Image acquisition
This aspect describes the purposive modification of the image acquisition system’s parameter, that is, whether the method is active or passive. It is observed that almost all multi-ocular depth cues require a special camera set-up, and most monocular depth cues do not.

2. Image content
The image content aspect involves what kind of image characteristics is needed by the algorithms in order for them to work reliably. Some of them have been assigned the term “All” in the table, which indicates that no special requirement of the image content is needed.

3. Motion presence
The motion presence aspect concerns the presence of disparity of the same feature point in the input images. It is only applicable for multi-ocular depth cues. Since monocular depth cues operate on a single image, no motion is needed.

4. Real-time processing
Some of the investigated papers provide explicit running time and environment parameters, others just claim that the algorithm is suitable for real-time application or do not mention the speed at all. This is reflected in the comparison table. In order to make

the speed comparable, a simple conversion rule is applied when quantitative performance data is available. It is considered that the speed of 25 frame/second (fps) is the speed of real-time processing, running on one regular PC of current standards with a frame size of 640x480 pixels. If an algorithm runs on a normal PC with a speed of 25 frame/second and a frame size 256x256, this speed is then converted to 5.3 fps ($\frac{(256\times 256)\times 25}{(640\times 480)}\approx 5.3$) since the total number of processed pixels within the same period remains the same. If more PCs are used then this result needs also to be divided by the number of PCs. In the case when no explicit data are available, a general description is given.

Another remark is that the speed of algorithms is highly related to the accuracy. Higher accuracy requires more processing time. Therefore, a frequently used practice for achieving real-time speed is to simply reduce the accuracy to an acceptable limit. In the comparison table, we present the speed and accuracy for algorithms of each depth cue, but the data filled in the table do not necessarily belong to the same single algorithm.

The traditional depth-from-focus methods [5] are fast but less accurate. Several researchers have published different techniques to improve the accuracy, but they require higher computation costs. A recent example of more accurate algorithms is the one proposed by Ahmad and Choi [26] using dynamic programming optimization technique. They tested several representative depth-from-focus methods including the traditional one. The speeds for constructing a depth map vary from 4 seconds to 4 minutes for a sequence of 97 images of size 256x256, running on a 2.8 GHz P-IV PC. With this fastest result ($\frac{97\times(256\times 256)}{4\times(640\times 480)}=5.2\,\text{fps}$), it can be concluded that the current depth-from-focus methods are not yet suitable for real-time application.

Shape-from-Silhouette is normally computationally and memory intensive. Thanks to various techniques such as parallel PC processing or 2D intersection, it is possible to meet the real-time criterion. The hardware accelerated visual hulls algorithm developed by Li et al. [27] relies on several consumer PCs following a server-client architecture and renders arbitrary views of the visual hull directirely from the input silhouette. For input images of 320x240, it runs at a speed of 84 fps using 1 PC as server and 4 PCs as clients. Converted to our standard, it has a speed of $\frac{84\times(320\times 240)}{5\times(640\times 480)}=4.2\,\text{fps}$, which seems slow. But considering the strength of this algorithm lies exactly in parallel processing, it is labeled as real-time algorithm.

5. Accuracy

As the aspect of real-time processing, accuracy comparison also lacks experimental data for certain depth cues. And if it exists, the test data and environment are not based on a uniform foundation. We therefore only present the available error measurements here and do not make any normalization.

Binocular disparity (stereo) suffers from occlusion, i.e. points that cannot be seen by one of the cameras. Tri-ocular or multi-ocular stereo gives more accurate results as more views (constraints) of the objects are available. The current best binocular stereo correspondence algorithm [28] evaluated in the Middlebury homepage [9] has the percentage of “bad” pixels of 0.16% – 0.97% depending on the test images.

The depth-from-focus method is sensitive to the aberrations of the lens system. Light rays have different focus points according to their distance to the optical axis. It results in distortion in the images in a form of barrel or pincushion distortion. A curvature of field is a significant reason, which means a planar object does not appear as a planar image. This leads to different and erroneous depth values because distinct parts of the object are not focused at the same distance. A method to reduce this symptom is to estimate the error by fitting a second order surface to the depth map and utilizes this error to construct a more accurate depth map [29]. However, the author did not present the error rate in the paper.

6. Absolute depth or relative depth

Some algorithms provide absolute (real) distance between the viewing camera and the objects, and they are able to estimate the actual size of the object; others measures relative depth by analyzing shading, edges and junction etc., providing a relative depth ordering between parts but no actual size. Most algorithms which rely on camera parameters can recover the real depth. It is also widely conceived that monocular depth cues cannot be used to estimate the real depth. As noted in section 2.13, this belief needs to be revised thanks to the machine learning techniques.

7. Dense or sparse depth map

This aspect focuses on the density of the depth map – whether each pixel of the image is assigned a depth level. A dense depth map is constructed using the global image features. This is desirable for the 3D television application since the entire image content needs to be presented. A sparse depth map offers only depth values for feature points. It is more suitable for 3D shape extraction. Some depth cues are able to generate both dense and sparse depth maps, depending on whether the specific algorithm makes use of local feature points or global structures.

8. Depth range

This aspect describes the effective depth range a human can perceive based on each individual depth cue. For example, occlusion works in all ranges; and atmospheric scattering works only at large distance. The depth ranges presented in the table are partly extracted from the work of Cutting and Vishton [30], illustrated in Figure 16. In the comparison table, we use the value of the depth range when the depth contrast is equal to 0.08. The resultant depth ranges conform in great line with other relevant literatures.

When considering the depth range which a 2D to 3D conversion algorithm can produce, it should be noted that the depth ranges of most relative depth cues are unlimited. What a relative depth cue provides is a depth ordering but no real magnitude. It is up to the user to choose the right depth range scale so that the recovered depth map creates the most realistic stereo effects. When a depth cue is suitable for absolute depth estimation (see aspect 6), the output depth range of the algorithms based on this depth cue coincides with the depth range humans can perceive using the same cue.

9. State of each depth cue

The state of depth cue describes approximately when the first or earlier algorithm based on certain depth cue was published in the realm of computer vision.

[9] figure adapted from [30]

Table 2: Multi-ocular Depth Cue Comparison

The Number of Input Images	***Depth Cues***	***Image acquisition***	***Image content***	***Motion presence***	***Real-time processing***	***Accuracy***	***Absolute/relative depth***	***Dense/Sparse depth map***	***Depth range***	***State of algorithm***	***Miscellaneous***
Two or More Images (multi-ocular)	**Binocular disparity**	**Active**: 2 images of the scene taken from different view points so that corresponding points can be observed	All	Yes	Yes	Percentage of pixels whose absolute disparity error is greater than 1: 0.16% - 0.97 % [28]	Absolute	Dense/sparse	< 30 meter	1976, Marr & Poggio [31]	Sensitive to occlusions; Disparity must not be too small; suffer from the lack of local surface texture due to smoothness of depth constraint
	Motion	**Active/passive:** Image sequences of moving objects or static scene taken by moving cameras	All	Yes	Yes	High accuracy achieved especially by algorithms which allows integration of multiple frames from image sequences	Absolute	Dense/sparse	< 30 meter	1979, Ullman [32]	Disparity is required to be small; optical flow method is sensitive to noise.
	Defocus	**Active**: 2 or more images taken by one camera using different camera parameters	Objects with complex surface characteristics (e.g. textured images either due roughness of the surface or reflectance variations).	No	Yes	Relative error rate: 0.5% when object is 2.5 meter away from camera [33]	Absolute	Dense	N/A	1987, Pentland [12]	
	Focus	**Active**: a series of images taken by one camera by varying the distance between the camera and objects	Objects with complex surface characteristics	No	No: 5.2 fps[26]	Relative error rate: 0.1% when object is 1.2 meter away from the camera [33]	Absolute	Dense	N/A	1987, Pentland [12]	Sensitive to the aberrations of lens system; Better performance for indoor scenes where the target is close by. It is applicable to images of small objects with a size up to hundreds microns.
	Silhouette	**Active**: Images taken by multiple cameras surrounding the scene	Foreground objects must be distinguishable from background	Yes	Yes: 4.2 fps [27]	Volume can be reconstructed well but texturing is not realistic.	Absolute	Sparse, only depth values of the foreground objects are recovered	Indoor size	1983, Martin & Aggarwal [34]	

Table 3: Monocular Depth Cue Comparison

The Number of Input Images	***Depth Cues***	***Image acquisition***	***Image content***	***Motion presence***	***Real-time processing***	***Accuracy***	***Absolute/relative depth***	***Dense/Sparse depth map***	***Depth range***	***State of algorithm***	***Miscellaneous***
One single image (monocular)	**Atmosphere Scattering**	Passive	Scene contains haze	No	N/A	Relative error rate: 10% for outdoor scenes	Relative	Dense	900 - 8000 meter; suitable for distant objects	1997, Cozman and Krotkov [15]	More suitable for outdoor images
	Defocus	Passive	Image contains one in-focus region and one out-of-focus region; either one of them is in foreground or background.	No	Yes	N/A	Relative	Dense	All ranges	1998, Elder & Zucker [35]	Not suitable for images of blurred textures.
	Shading	Passive	Image must not be too dark. The depth of regions such as shadow area which contains too little intensity information and cannot be recovered.	No	No	Mean error rate 4.6%, Maximum error 40% for cliffy surface [17]	Relative	Dense on surface	All ranges	1975, Horn [36]	Good estimates of local surface areas, but some of them have problems with variable albedo and spherical surfaces
	Linear perspective	Passive	Image contains geometric appearance	No	Yes	N/A	Relative	Dense	All ranges	1980, Haralick [37]	
	Patterned texture (incorporating relative size)	Passive	Some algorithm requires segmented texture region, other not	No	No	The average error is the angle between the estimated and actual surface normal = 8 degree [38]	Relative	Sparse (only on texels) /Dense	All ranges	1976, Bajcsy & Lieberman [39]	Many algorithms need texture segmentation beforehand
	Symmetric patterns	Passive	Non-frontal image of bilateral symmetric objects	No	N/A	N/A	Relative	Sparse, only depth values of symmetric objects are recovered	All ranges	1981, Kanade [40]	
	Static occlusion										
	- Curvature	Passive	All	No	Yes	N/A	Relative	Dense	All ranges	2005, Redert [22]	More omplex than the cue "transform"
	- Simple Transform	Passive	All	No	Yes: 40 fps [23]	N/A	Relative	Dense	All ranges	2005, Redert [23]	Low complexity, time-stable
	Statistical patterns	Passive	All	No	Yes	Average error: 0.132 in log scale (base 10); the training depth maps has a maximum range of 81 meter [25]	Absolute/Relative	Dense	All ranges	2002, Torralba & Oliva [24]	Especially good in estimating depth map of outdoor scenes

4 A new Trend: Pattern Recognition in Depth Estimation

It can be observed that a lot of the 2D to 3D conversion algorithms are still in the research phase. They are not yet ready for real-time use due to factors such as their high complexity or unsatisfactory quality. As well as improving the existing algorithms, a new trend in this field is to analyze the semantic content of the image and use this knowledge to help reconstruct the 3D object. The depth cue “statistical patterns” plays the central part in this trend. This chapter is dedicated to a more detailed description of a recent developed algorithm of Battiato [8] et al. using the image classification technique. The algorithm operates on a single color image. No a priori knowledge about the image content is needed. It is also claimed to be fully unsupervised and suitable for real-time applications.

Eight steps are involved in this algorithm. Throughout the process, two intermediate depth maps are constructed, the qualitative depth map and the geometric depth map. In the end, these two depth maps are combined together to generate the final depth map. The following paragraphs expound on these eight steps. Note that the fifth and sixth steps make use of the depth cue of the linear perspective and have been introduced in section 2.7. For the sake of completeness, these two steps are repeated here.

1. ***Color-based segmentation***
   Color-based segmentation identifies the chromatic homogeneous regions present in the image. The image is under-segmented so that main chromatic regions are retrieved and fine details are filtered out.

2. ***Rule-based regions diction to find specific areas***
   The segmented image in the RGB color model is converted to the HSI color model. The HSI model is more suitable for color description. Subsequently, the intensity values of the R, G, B, H and S components of each pixel in the image undergo various checks based on a set of color-based rules, which has been learned heuristically. These color-based rules are able to identify six semantic regions possibly present in the image: Sky, Farthest Mountain, Far Mountain, Near Mountain, Land and Other.

3. ***Qualitative depth map construction***
   Each semantic region is assigned a depth level, which corresponds to a certain gray level following the trend: Gray(Sky) < Gray(Furthest Mountain) < Gray(Far Mountain) < Gray(Near Mountain) < Gray(Land) < Gray(Other). The resultant image is termed the qualitative depth map (Figure 17).

4. ***Image classification to discriminate three categories: Indoor, Outdoor without geometric appearance, Outdoor with geometric appearance.***
   The six semantic regions obtain their own labels in this step, for example, the sky is labeled as ‘s’. The qualitative depth map is then sampled column-wise. Each column is represented by a label sequence which labels from top to down each region present in the column. A typical sequence could be “sml”, for instance, which indicates that the sample column consists of the 3 regions -Sky, Mountain and Land.

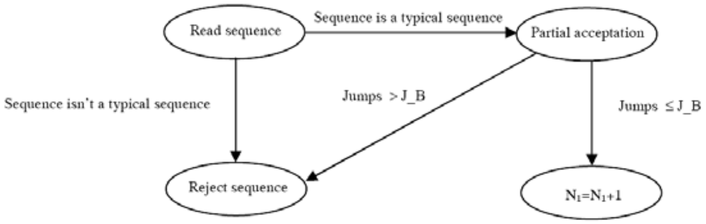

Figure 19: Determining the number of accepted sequences in an image

A sequence is accepted if it belongs to a typical landscape sequence and the number of jumps is smaller than a certain threshold (denoted by J_B in Figure 19). Finally, the following heuristics is applied to classify the image into three categories:

1) #Accepted_sequences > Threshold × #Sequence ⇒ OUTDOOR
2) #Sequences with the first region SKY > Threshold × #Sequences
   ⇒ OUTDOOR WITH GEOMETRIC APPEARANCE
3) Otherwise ⇒ INDOOR

5. ***Vanishing lines detection***
   Different vanishing line detection strategies are applied according to the category to which the image belongs. For Outdoor scenes, the vanishing point is put in the center

region of the image and a set of vanishing lines passing through the vanishing points are generated. For the categories Indoor and Outdoor with geometric appearance, a more complex technique is applied. Edge detection (using Sobel operators) and line detection (Hough transform) are conducted to determine the main straight lines. The vanishing point is chosen as the intersection point with the most intersections around it while the vanishing lines are the predominant lines passing close to the vanishing point.

6. *Geometric depth map construction*

Taking the position of the vanishing point into account, a set of horizontal or vertical gradient planes is assigned to each neighboring pair of vanishing lines. A gradient plane has a fixed depth level. There are more gradient planes close to the vanishing point than further away because human vision is more sensitive for the depth perception of objects close by (Figure 7).

7. *Consistency verification of detected regions*

In this step, the qualitative depth map is checked for consistency. False classified semantic regions are detected and corrected. For example, if between two “Sky” regions, there is a region of another type (e.g. a Mountain) with a vertical size larger than certain threshold, the second sky region is then identified as a “false” Sky and its type is changed to the same type of the upper zone (Figure 21).

8. *Fusion of the qualitative depth map and the geometric depth map*

The final depth map of INDOOR category image is just the geometric depth map. No fusion occurs. For OUTDOOR WITHOUT GEOMETRIC APPEARANCE, the final depth of pixel (x, y) is assigned the depth value in the qualitative depth map in all cases, except when it belongs to the $\text{Land} \cup \text{Other}$ category. In the latter case, pixel (x, y) obtains its depth value from the geometric depth map (see Figure 22). For the image category OUTDOOR WITH GEOMETRIC APPEARANCE, the final depth of pixel (x, y) is assigned the depth value in the geometric depth map for all cases, except when it is a *Sky* pixel, it then adopts the depth value in the qualitative depth map.

5 Discussion and Conclusion

A vast number of 2D to 3D conversion algorithms are dedicated to recover the “structure” or “shape” of objects in the images, which are understood to mean the 3D coordinates of a small set of points in the scene. These algorithms (e.g. shading, silhouette, symmetric patterns) can be possibly put into better use in computing the 3D motion of the camera or the objects, robot navigation, surveillance etc. rather than 3D video. For applications such as 3D TV, a dense depth map of all pixels in the image is conceivably better suited.

The multi-ocular depth cues take both spatial and temporal image information into account, which yield in general a more accurate result. The monocular depth cues are less accurate but do not require multiple images, which makes them more versatile. Image sequences where both objects and camera barely move can best resort to the monocular cues.

A single solution to convert the entire class of 2D images to 3D models does not exist due to the tremendous variations of the problem domain. The conversion problem is an ill-posed problem. It is often solved with strong enough constraints on the underlying problem domain. A new trend of the development of 2D to 3D conversion algorithms is to operate in association with robust algorithms for image semantics analysis and to design specialized conversion algorithm for each specific semantic entity.

It can be stated that no one cue is superb or indispensable for depth perception. Each cue has its own advantages and disadvantages. It is necessary to combine the suitable depth cues in order to achieve a robust all-round conversion algorithm. Some depth cues produce less detailed surface information (low frequency) due to reasons such as smoothness constraints (e.g. stereo), other depth cues offers a bettered detailed surface (high frequency), combing them may leads to a better result. The method based on image classification [8] is an example of depth cue fusion, where the depth maps derived from two complementary single cues enhance each other. The novel 2D to 3D conversion algorithm (e.g. [25]) based on supervised learning is in fact also one of the convincing ways of combining different depth cues. Its promising performance makes it certainly a new valuable research direction in this field.

It can also be concluded that serious considerations of systematic performance evaluation in quantitative aspects are needed. This would allow both the designers and the users of 2D to 3D conversion algorithms to know which ones are competitive in which domains. It would also stimulate researchers to devise truly more effective and efficient conversion algorithms.

Most 2D to 3D conversion algorithms for generating stereoscopic videos and ad-hoc standards are based on the generation of a depth map. However, a depth map has a disadvantage that it needs to be fairly dense and accurate. Otherwise local deformations in the derived stereo pairs are easy to happen. There are also approaches which do not work with a depth map. A recent example is the algorithm proposed by Rotem, Wolowelsky and Pelz [41], which creates the stereo pairs directly from the original video frames. Each pair is composed of the original image and a transformed image. The latter is generated by warping another image from the original video sequence using planar transformation. This method is claimed to be less prone to local deformation and the quality is so good that it is even suitable for applications where deformation is forbidden as in reconnaissance and medial systems. It is therefore helpful to also explore the alternatives than to confine ourselves only in the conventional methods based on depth maps.

6 Bibliography

[1] IJsselsteijn, W.A.; Seuntiëns, P.J.H.; Meesters, L.M.J. (2002) “State-of-the-art in Human Factors and Quality Issues of Stereoscopic Broadcast Television”, Deliverable ATTEST/WP5/01, Eindhoven University of Technology, the Netherlands.
URL: http://www.hitech-projects.com/euprojects/attest/deliverables/Attest-D01.pdf

[2] Trucco, E; Verri, A (1998) “Introductory Techniques for 3-D Computer Vision”, Chapter 7, Prentice Hall.

[3] Scharstein, D.; Szeliski, R. (2002) “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms”, International Journal of Computer Vision 47(1/2/3), 7-42.

[4] Ziou, D; Wang, S; Vaillancourt, J (1998) “Depth from Defocus using the Hermite Transform”, Image Processing, ICIP 98, Proc. International Conference on Volume 2, 4-7, Page(s): 958 - 962.

[5] Nayar, S.K.; Nakagawa, Y. (1994) “Shape from Focus”, Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 16, Issue 8, Page(s): 824 – 831.

[6] Matsuyama, T. (2004) “Exploitation of 3D video technologies ”, Informatics Research for Development of Knowledge Society Infrastructure, ICKS 2004, International Conference, Page(s) 7-14.

[7] Wong, K.T.; Ernst, F. (2004), Master thesis “Single Image Depth-from-Defocus”, Delft university of Technology & Philips Natlab Research, Eindhoven, The Netherlands.

[8] Battiato, S. ; Curti, S. ; La Cascia, M.; Tortora, M.; Scordato, E. (2004) “Depth map generation by image classification”, SPIE Proc. Vol 5302, EI2004 conference ‘Three-dimensional image capture and applications VI”

[9] Scharstein, D.; Szeliski, R. (2005), “Middlebury Stereo Vision Page”, web page visited in October 2005.
URL: http://www.middlebury.edu/stereo

[10] Han, M; Kanade, T. (2003) “Multiple Motion Scene Reconstruction with Uncalibrated Cameras”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 25, Issue 7, Page(s): 884 – 894

[11] Franke, U.; Rabe, C. (2005), “Kalman filter based depth from motion with fast convergence”, Intelligent Vehicles Symposium, Proceedings. IEEE, Page(s): 181 – 186

[12] Pentland, A. P. (1987) “Depth of Scene from Depth of Field”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No.4, Page(s) 523-531.

[13] Subbarao, M.; Surya, G. (1994) “Depth from Defocus: A Spatial Domain Approach”, the International Journal of Computer Vision, 13(3), Page(s) 271-294.

[14] Elder, J.H.; Zucker, S.W. (1998) “Local Scale Control for Edge Detection and Blur Estimation”, IEEE Transactions on Pattern Analysis and Machine Vison, Vol. 20, No.7.

[15] Cozman, F.; Krotkov, E. (1997) “Depth from scattering”, IEEE Computer society conference on Computer Vision and Pattern Recognition, Proceedings, Pages: 801–806

[16] Zhang, R.; Tsai, P.; Cryer, J.E.; Shah, M. (1999), “Shape from Shading: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence archive, Vol. 21, Issue 8, Pages: 690-706

[17] Kang, G; Gan, C.; Ren, W. (2005), “Shape from Shading Based on Finite-Element”, Proceedings, International Conference on Machine Learning and Cybernetics, Volume 8, Page(s): 5165 – 5169

[18] Forsyth, D.A. (2001) “Shape from texture and integrability”, ICCV 2001, Proceedings, Eighth IEEE International Conference on Computer Vision, Volume 2, Page(s): 447 – 452

[19] Loh, A.M.; Hartley, R. (2005) “Shape from Non-Homogeneous, Non-Stationary, Anisotropic, Perspective texture”, Proceedings, the British Machine Vision Conference 2005

[20] Francois, A.R.J.; Medioni, G.G.; Waupotitsch, R. (2002) “Reconstructing mirror symmetric scenes from a single view using 2-view stereo geometry”, Proceedings, 16th International Conference on Pattern Recognition, Vol. 4, Page(s):12 – 16

[21] Shimshoni, I.; Moses, Y.; Lindenbaumlpr, M. (1999), “Shape reconstruction of 3D bilaterally symmetric surfaces”, Proceedings, International Conference on Image Analysis and Processing, Page(s): 76 - 81

[22] Redert, A. (2005) Patent ID: WO2005091221 A1, “Creating a Depth Map”, Royal Philips Electronics, the Netherlands

[23] Redert, A. (2005) Patent ID: WO2005083630 A2, WO2005083630 A2, WO2005083631 A2, “Creating a Depth Map”, Royal Philips Electronics, the Netherlands

[24] Torralba, A.; Oliva, A. (2002), “Depth Estimation from Image Structure”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 24 , Issue 9, Pages: 1226 - 1238

[25] Saxena, A; Chung, S.H.; Ng, A. Y. (2005) “Learning Depth from Single Monocular Images”, Proceedings, 19th Annual Conference on Neural Information Processing Systems (NIPS 2005)

[26] Ahmad, M.B.; Tae-Sun Choi (2005) “Fast and accurate 3D shape from focus using dynamic programming optimization technique”, Proc. (ICASSP '05), IEEE International Conference on Acoustics, Speech, and Signal Processing; Page(s): 969 – 972, Vol. 2

[27] Li, M.; Magnor, M.; Seidel, H. P. (2003) “Hardware-Accelerated Visual Hull Reconstruction and Rendering”, Proceedings of Graphics Interface 2003, Halifax, Canada

[28] Sun, J.; Li, Y.; Kang, S. B.; Shum, H. Y. (2005) “Symmetric stereo matching for occlusion handling”, Proceedings, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2005)

[29] Blahusch, G.; Eckstein, W.; Steger, C. (2003) “Calibration of Curvature of Field for Depth from Focus”, Proc. of the ISPRS Workshop “Photogrammetric Image Analysis”, Vol. XXXIV, Part3/W8, Munich

[30] Cutting, J.; Vishton, P. (1995) “Perceiving Layout and Knowing Distances: the Integration, Relative Potency, and Contextual Use of Different Information about Depth”, In Epstein, W; Rogers, S (editors.) Perception of Space and Motion, Pages 69-117. Academic Press, San Diego.
URL: http://pmvish.people.wm.edu/cutting&vishton1995.pdf

[31] Maar, D.; Poggio, T. (1976) “Cooperative Computation of Stereo Disparity”, Science, Volume 194, Pages: 282-287

[32] Ullman, S. (1979) “The Interpretation of Visual Motion”, MIT Press

[33] Xiong, Y.; Shafer, S. A. (1993) “Depth from Focusing and Defocusing”, Proceedings of Computer Vision and Pattern Recognition 1993 and Proceedings of Image Understanding Workshop 1993

[34] Amartin, W. N.; Aggarwal, J. K. (1983) “Volumetric Descriptions of Objects from Multiple Views”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 5(2), Pages: 150-158

[35] Elder, J.H.; Zucker, S. W. (1998) “Local Scale Control for Edge Detection and Blur Estimation”, IEEE Transactions on Pattern Analysis and Machine Vision, Volume 20, No. 7

[36] Horn, B.K.P. (1975), “Obtaining Shape from Shading Information”, P.H. Winston (e.d.), the Psychology of Computer Vision, McGraw-Hill, New York, Pages: 115-155

[37] Haralick, R. M. (1980) “Using Perspective Transformation in Scene Analysis”, Computer Graphics and Image Processing (CGIP 13), Volume 13, Issue 3, Pages: 191-221

[38] Stone, J. V.; Isard, S. D. (1995) “Adaptive Scale Filtering: A General Method for Obtaining Shape from Texture”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 17, Issue 7, Pages: 713-718

[39] Bajcsy, R.K.; Lieberman, L.I. (1976) “Texture Gradient as a Depth Cue”, CGIP, 5(1): 52-67

[40] Kanade, T. (1981) “Recovery of the Three-Dimensional Shape of an Object from a Single View”, Artificial Intelligence, Volume 17, Pages: 409-460

[41] Rotem, E.; Wolowelsky, K.; Pelz, D. (2005) “Automatic Video to Stereoscopic Video Conversion”, SPIE Proc. Vol. 5664, Stereoscopic Displays and Virtual Reality Systems XII