INFORMATIK

3D Acquisition of Mirroring Objects

Research Report

MPI-I-2003-4-001 March 2003

Marco Tarini, Hendrik P. A. Lensch,
Michael Goesele, Hans-Peter Seidel

Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
tarini@iei.pi.cnr.it
lensch,goesele,hpseidel@mpi-sb.mpg.de

Abstract

Objects with mirroring optical characteristics are left out of the scope of most 3D scanning methods. We present here a new automatic acquisition approach, shape-from-distortion, that focuses on that category of objects, requires only a still camera and a color monitor, and produces range scans (plus a normal and a reflectance map) of the target.

Our technique consists of two steps: first, an improved environment matte is captured for the mirroring object, using the interference of patterns with different frequencies in order to obtain sub-pixel accuracy. Then, the matte is converted into a normal and a depth map by exploiting the self coherence of a surface when integrating the normal map along different paths.

The results show very high accuracy, capturing even smallest surface details. The acquired depth maps can be further processed using standard techniques to produce a complete 3D mesh of the object.

1 Introduction

In recent years, research on 3D acquisition led to the development of working techniques capable of producing digital 3D models of real world objects. These techniques differ in many aspects, including precision, scanning time, amount of required human interaction, economic affordability, type of resulting models, captured attributes, and so on. Another central aspect are the categories of objects that can be scanned by the specific techniques, i.e., which constraints limit the possible targets in terms of size, weight, complexity and surface properties.

In particular, theory and practice show that objects with mirroring surface properties are outside the scope of almost all current standard approaches for high quality acquisition. In all active scanning methods (e.g., laser scanning, structured light, time of flight – see [2]) the problem is that the active signal (e.g. projected pattern or laser) is mainly reflected along the specular direction so that almost no light reaches the sensor. In addition, specular reflections of an object are effected by the surface normal and can therefore lead to the effect that the projected pattern is perceived on the wrong position [1]. Shape from shading based approaches [26] require the surface to be Lambertian and don't work for glossy, especially not for mirroring surfaces. In shape from silhouette approaches [8], beside of the general problem of lacking precision, the silhouettes of mirroring objects are hard to detect automatically since the whole environment is reflected in the object.

However, mirroring objects may nevertheless be scanned by the approaches mentioned above if they are previously painted (or covered with white dust) to approximate Lambertian surface behavior. But this is clearly invasive and unpractical for many applications.

All in all, there is an urgent need to use optical techniques to measure the shape of an object with a specular surface, especially technical components, pieces of art or even everyday objects -like knifes, forks or keys- without changing their optical properties.

In this article we present a novel approach, *shape from distortion*, to reconstruct the surface of a mirroring object with high precision. It uses only standard, cheap hardware (a monitor and a camera), and care has been taken to achieve relatively short acquisition times. Our approach consists of two steps:

- an improved environment matte extraction that uses the interference of patterns of different frequencies in order to obtain sub-pixel accuracy relative to the projected pattern,
- followed by a step that converts the matte into a normal and a depth map by exploiting the self coherence of a surface when integrating the normal map along different paths.

The process is very modular: even if the second step requires the output of the first one as input, each can also be used separately in different contexts.

Before giving a more detailed overview over our technique in Section 3 we will briefly summarize related work by other researchers on the subject of specular objects scanning.

2 Previous Work on Mirroring Object Acquisition

Several research groups developed variations of standard range scanning techniques that are capable of scanning specular objects.

In Oren and Nayar [14], 3D surface profiles of specular objects are generated by observing the movements of virtual (i.e., reflected) features on their surface. Zheng and Murata [28] extended this approach by rotating a specular object on a turntable under the illumination of circular-shaped light sources. If there are no occlusions due to the object shape, a highlight can be observed on all surface points. While they can derive the complete geometry of an object in a single scan, the accuracy of the generated models is limited and special hardware is required.

In [1], a laser scanning system is proposed where a shield mask limits the direction of incident light. The prototype system is able to scan specular as well as Lambertian surfaces with high accuracy but at a very low speed since each position on the surface has to be seen under various angles (100 measurement points in 20 min [19]).

The quality of flat or nearly flat surfaces such as a die or sheet metal has often been measured by observing the reflection of regular grids (see for example [2] for an overview). Similar to reflection lines used in geometric modelling [10], the accuracy of these methods is high because measured surface normals are used to reconstruct the shape of the object. A similar method has also been proposed by [11] to measure the shape of the human cornea. Our method also benefits from this measurement principle in terms of accuracy and extends it to recover dense range maps of arbitrary shaped objects.

Park and Cho [16] place a specular object in a retroreflective hemisphere and illuminate its surface with a laser beam. The reflected beam is measured and the position on the object where it was hit by the retroreflected beam is calculated. The next beam is directed at this surface point under a different angle. Once the position and corresponding normal for one surface point is known, the shape and surface normals of a sufficiently smooth object can be reconstructed recursively.

The main problem of this approach is that errors are accumulted in each step and not corrected. With growing distance from the initial point the data becomes more and more unreliable.

In their recent work on this subject, Savarese and Perona [23, 24] use a 2D pattern consisting of points identified by the intersection of three coplanar lines, and manage to recover from their reflection some attributes of the mirroring surface, i.e. relationships between depth and normal. This can be used to locally reconstruct the surface itself, if an approximation of its distance is provided, or some a priori knowledge of the shape of the mirror (a sphere or a flat mirror) is known. Still, the detection and labelling of elements of the patterns is far from automatic and indeed quite problematic for not trivially shaped mirror. More important, the measurements are very sparse. Typically their number in around a dozen, so no assumption of point-to-point continuity is possible. Therefore, the technique can be useful as auxiliary rather that substitute for typical range scanning technique.

A technique that captures more the appearance of mirroring objects rather than their 3D geometry has been presented by Zongker et al. [29]. They introduced environment mattes which capture how light impinging from the environment is reflected by the object. We base our acquisition technique on environment matting and extend this approach. Environment mattes will be revisited and explained in more detail in Section 4.

3 Overview

In order to obtain the 3D geometry we observe images of a nearby monitor that are reflected in the mirroring object. The shape of the object is recovered by investigating the way the perceived reflection of the monitor is distorted. We call this process *shape from distortion*.

To represent the distortion we use an environment matte [29, 3] which is a general way to model the optical interactions between an object and its surroundings. Our system consists of, first, capturing a matte, and then using it to produce a range map representing the surface of the investigated object. The next two Sections are dedicated to one phase each:

Section 4 describes a new automatic way to capture the environment matte with high accuracy: we capture 5 images of the object while it reflects repetitive patterns of different frequencies displayed on a nearby monitor (Fig. 3.1a). These images are used to construct the matte with sub-pixel precision (Fig. 3.1b), i.e., for each pixel in the captured images we compute the exact 2D position of the point where the reflected ray hits the monitor.

In Section 5 we describe how the geometry of the object in the form of a depth and a normal map (Fig. 3.1c) is reconstructed from the resulting matte. Given the position of the monitor and a tentative normal at each pixel, the environment matte relates the normals to corresponding depth values, and vice versa. Since both are initially unknown we apply another constraint to find their absolute values: we minimize the incoherence of the surface that occurs integrating the normal map along different paths.

The final output of the presented algorithm is a standard range image, a 2D data structure that for each point stores the distance of the scanned object along the corresponding ray (Fig. 3.1d) enriched by a normal map, plus a reflectance map derived directly from the matte. A complete model of the object can be constructed out of multiple range maps using well known techniques for scan alignment [17, 22], merging [6, 20], and surface simplification [4].

The described steps of our approach are in practice preceded by a calibration procedure which is explained in Section 6. We need to recover the response of

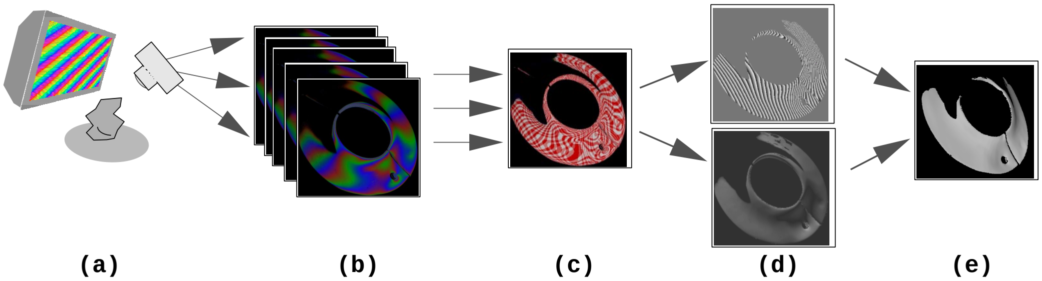

Figure 3.1: The big picture of *shape from distortion*: the target object is photographed (a) while reflecting some pattern displayed by a monitor. The captured images are used (b) to compute an environment matte (c) from which we extract a normal map together with a depth map (d) that are embedded into the final 3D range scan (e).

the camera to the intensity of the monitor with respect to the three color channels, since colored stripe patterns are applied during scanning. Further, a calibration of the intrinsic camera parameters is performed, and the relative monitor position is measured. This calibration has to be done only once and can be reused for subsequent scans as long as the monitor and the camera are not moved.

Finally, in Section 7 we show some experimental results and in Section 8 we conclude and give some directions for future work.

4 Matting extraction

In this section we describe an algorithm to acquire an *Environment Matte* that captures with sub-pixel precision the way mirroring objects reflect their environment.

4.1 Previous Matting Techniques

Matting is a category of 2D techniques that consists, in the general case, in capturing how the environment (in most cases a far background) is filtered, reflected, occluded, etc., by a target object (given a fixed illumination and viewpoint). This information is then typically used to synthesize images where the same foreground object is realistically shown in a different, possibly synthetic, environment.

The most simple matte, known as blue (or, more recently, green) screen matte [25], records for each pixel a real value in the interval $[0..1]$ modelling how much of the background is visible at that pixel. This technique is widely used in cinematography to replace the background of an arbitrary scene. The information is captured at each frame showing the object in front of a homogeneous blue, yellow or green background.

More information is recovered by an environment matte [29, 3], which stores at each pixel position (u, v) a texture coordinate $\vec{o}(u, v)$ into an environment texture corresponding to the texel that is visible at (u, v) after being reflected/refracted by the object. The object can then be placed into arbitrary backgrounds while maintaining its reflection characteristics. Actually the original environment matte does not store the coordinate of a single point but the rectangular area of the environment texture that contributes to the considered pixel. In our implementation we only use the main reflection direction.

In [29] Zongker et al. extracted the necessary information $\vec{o}(u, v)$ from a series of images of reflected binary encoded stripes displayed by a monitor that is placed at the position of the virtual texture. Here, the frequency of the pattern is doubled from frame to frame using 1, 2, 4, $\ldots, 2^n$ stripes. The low frequency patterns determine a rough approximation of the location of $\vec{o}(u, v)$, while higher frequency ones provide a more and more accurate refinement of that estimate. Similar approaches of spatio-temporal modulation of the illumination are commonly used in other 3D scanning applications, e.g. [21]. Unfortunately, in current approaches the accuracy of the system is limited by the resolution of the projected pattern with the highest frequency. Furthermore, little information is obtained from the images with lower frequencies which only roughly determine the position.

The environment matting technique has further been extended in two ways [3]: the first one increases the quality. It assumes that a pixel value at (u, v) results from an area of the texture weighted by a Gaussian rather than a rectangular region. A stripe with Gaussian profile is swept smoothly over the monitor. The stream of 100 to 300 frames is recorded by a video camera to obtain the position $\vec{o}(u, v)$ and the spatial extend of the Gaussian filter that has to be applied.

Another improvement reduces the effort of capturing the matte to just a single frame, allowing real-time mattes. However, it only works for background textures. The binary stripe pattern is replaced by smoothly-varying transition of pure red, green and blue color, a planar slice through the rgb cube. The resulting matte consists of the texture coordinates $\vec{o}(u, v)$ and an opacity value $\alpha(u, v)$. While the acquisition is considerably accelerated it also introduces a lot of noise. To improve quality, the recovered matte is filtered by combining neighboring pixels which limits the accuracy.

4.2 Our Matte

In our approach the matte is defined by the following attributes at each pixel (u, v):

- the 2D position $\vec{o}(u, v)$ where the ray through pixel (u, v), reflected by the object, intersects the monitor. $\vec{o}(u, v)$ will be recovered with sub-pixel precision with respect to the resolution of the monitor which is required for accurate 3D reconstruction.

- the reflectance $\alpha(u, v)$ as a color vector, encoding the perceived attenuation of light emitted by the monitor. For a perfect mirror $\alpha(u, v)$ will be $(1, 1, 1)^T$ while for points outside the silhouette of the object it will be $(0, 0, 0)^T$.

Regarding the content, our matte is very similar to the real-time captured matte [3]. However, our application scenario demands a much higher accuracy so that a single image is not sufficient to recover a high quality matte. Even more important, we have to avoid filtering and want to be able to reconstruct the value of $\vec{o}(u, v)$ for each pixel of the captured image totally independent on the values of neighboring pixels. This is because the origin of a reflected ray may change drastically for relatively small variations of the surface normal. Combining values

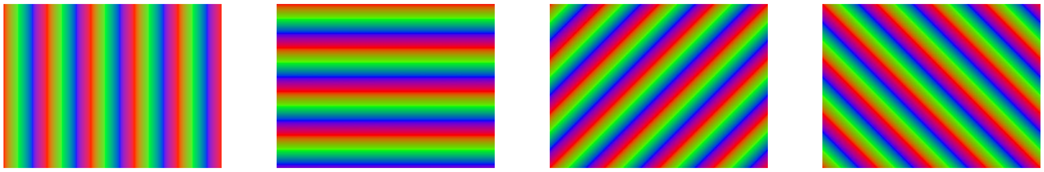

Figure 4.1: Patterns $M_{1..4}$ sent to the monitor to capture the reflected environment matte.

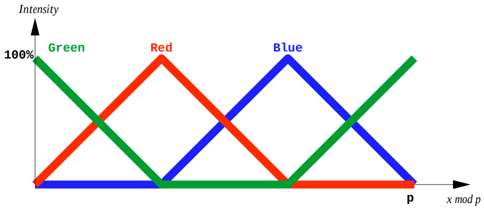

Figure 4.2: Each striped pattern shown in Figure 4.1 is perceived in the camera as a repetition of the following values of red, green, blue (up to a scale).

from neighboring pixels thus would introduce too large errors. Finally, we want to keep the number of required images as low as possible.

The new pattern scheme we apply for the matte extraction consists of four images. The patterns $M_{1...4}$ (visible in Figure 4.1) are striped along different directions. After calibration (see Section 6) the repeated period of the stripes is perceived by the camera as the periodic function shown in Fig 4.2: the signal is built out of hat functions for the RGB color channels displaced in such a way that they always sum up to one.

For each location (u, v), the quadruple of observed color values $c_i, i \in [1..4]$ for the four patterns is sufficient to determine $\vec{o}(u, v)$ robustly with sub-pixel precision: we first determine the position (or phase) $x_{1..4}$ inside one stripe for each of the four patterns; using the interference patterns we then recover the number of the stripe $y_{1..4}$ in which the point is located.

A perceived color pixel c_i identifies uniquely x_i as the abscissa, inside the function presented in Fig. 4.2, that best matches the RGB color values of c_i. We find x_i using the following algorithm:

1. The three color values are sorted yielding $c_i[\text{min}] \leq c_i[\text{med}] \leq c_i[\text{max}]$ (minimum, median, maximum).

2. The order of the color channels locates x_i inside one of the 6 region where each region is delimited by the intersection of the r, g, b hat functions of Figure 4.2.

3. The exact position of $x'_i \in [0..1)$ within the region is found by

$$x'_i = \frac{c_i[\text{med}] - c_i[\min]}{(c_i[\max] - c_i[\min]) + (c_i[\text{med}] - c_i[\min])} \tag{4.1}$$

   Considering the magnitude of $c_i[\text{med}]$ with respect to the minimal and maximal values takes into account darkening due to absorbed light. Additionally, it also works correctly if the minimum value is not zero. This reduces errors introduced by ambient lighting and color shifts. In an ideal case the minimum value would always be zero by construction of the signal.

4. The final x_i is given by the value of x'_i scaled and biased taking into account the region it belongs to.

$$x_i = \frac{\texttt{regionNr} - 1}{6} \pm \frac{x'_i}{6} \tag{4.2}$$

Repeating this algorithm for each color $c_{1..4}$ one ends up with four values $x_{1..4}$, each value x_i identifying where, inside a still unknown stripe y_i, the origin $\vec{o}(u,v)$ of the monitor point reflected to pixel (u,v) is located. The unknowns y_i are found by exploiting the interference of the four patterns.

In fact, the two diagonal patterns M_3 and M_4 have a different period for the stripes than M_1 and M_2, e.g., each row of M_1 has 5 repetitions while each row of M_3 has only 4. Since 4 and 5 are coprimes, there is only one solution $y_{1...4}$ that corresponds to the values $x_{1...4}$. To find it we need to solve the system

$$\begin{aligned} o_x &= (x_1 + y_1)/5 \\ o_y &= (x_2 + y_2)/5 \\ o_x + o_y &= (x_3 + y_3)/4 \\ o_x - o_y &= (x_4 + y_4)/4 \end{aligned} \tag{4.3}$$

where the unknowns are o_x, o_y, i.e. the coordinates of matte position $\vec{o}$, and the integer values $y_{1..4}$, with $y_{1,2} \in 0, \ldots, 4$ and $y_{3,4} \in 0, \ldots, 3$. Note that in the system all the variables are relative to an (u,v) position on the matte, therefore we omitted all the (u,v) indexes. The system is solved by

$$\begin{aligned} 4y_1 - 5(y_3 + y_4) &= 4x_1 - 5x_3 + 5x_4 \\ 4y_2 - 5(y_3 - y_4) &= 4x_2 - 5x_3 - 5x_4 \end{aligned} \tag{4.4}$$

Since the left sides of both equations represent integer values, we round the values of the two right sides, that are just constants, to their closest integer values

i_1 and i_2 respectively (the magnitude of the rounding can be used as a measure of our confidence for the value of $\vec{o}$ at that position). What remains to solve is

$$\begin{aligned} 4y_1 - 5d_1 &= i_1 \\ 4y_2 - 5d_2 &= i_2 \end{aligned} \tag{4.5}$$

where d_1 and d_2 are the sum and the difference between y_3 and y_4, i.e. $d_1 = (y_3 + y_4)$ and $d_2 = (y_3 - y_4)$.

The Equations 4.5 are two simple Diophantic Equations to be solved separately yielding

$$\begin{aligned} (y_1, d_1) &= i_1(3, 4) + k_1(5, 4) \\ (y_2, d_2) &= i_2(3, 4) + k_2(5, 4) \end{aligned} \tag{4.6}$$

for any $k_{1,2} \in \mathbb{N}$. We chose for k_1 (and k_2) the only value that puts y_1 (and y_2) in the required interval $[0..4]$.

With the values d_1 and d_2 we trivially compute values for y_3 and y_4. As an error recovery procedure, if the sum of d_1 and d_2 turns out to be odd, we know that one of the two roundings of i_1 and i_2 has been done in the wrong way: we can take the one which is further away from its closest integer value and re-perform its rounding in the opposite way (decreasing the confidence value accordingly).

Plugging the $y_{1..4}$ into (4.4) we get an overdetermined system to find the result $\vec{o} = (o_x, o_y)$ for the given pixel: the redundancy is used reduce errors.

In principle we could use the confidence values to reject bad or noisy measurements. However, we did not need to in practice, as the matte we computed showed high confidence for all the point with sufficient reflectance α, i.e. at all points where the original signal is not severely darkened (see Sec. 4.3 later).

4.2.1 Number of Stripes

The number of stripes used within one pattern is quite important. Ideally, the accuracy of the reconstructed $\vec{o}(u, v)$ will be increased with a higher number of stripes. Assuming that the x_i are always determined with the same accuracy no matter how many stripes are displayed, they represent a quantity varying at a higher rate resulting in higher precision.

But, if the stripes are too close, the computation of $y_{1...4}$ becomes more and more unstable. Since y_i are discrete values they are either computed correctly or deviate a lot from the correct solution. Their computation is very robust but if too many stripes are present the system may undergo more errors than it can handle. In these cases it would probably make sense to use the computed y_i of neighboring pixels to detect and correct errors, something we do not exploit currently.

In our implementation we use 5 stripes along the x axis for the vertical pattern and 4 repetitions for the horizontal one. The diagonal patterns are slanted by 45 degrees, resulting in 4 repetitions along the x axis. As required 4 and 5 are coprimes. Another desired property is that the patterns have more or less the same frequency. At the chosen frequency, the $y_{1..4}$ present little or no errors, while the $x_{1..4}$ still provides enough precision.

4.3 Color Shift

Our pattern scheme is naturally resistant against darkening because of the scaling in Step 3 of the algorithm in Section 4.2. That means a good matte can be captured even if captured color values do not sum up to one because only a fraction of the original brightness is reflected by the surface. The ability of the system to work independently of the obtained brightness is important, since it leaves some freedom in factors like camera exposure time, and gives robustness against absorption by the mirroring surfaces.

Aa another good quality, the zero-order continuity of the used periodic functions (see Fig. 4.2) gives to the system a relative resistance to the blurring resulting from the aliasing of the CCD.

On the other hand, the use of colored patterns makes the technique prone to errors due to color shift, mainly occurring as a consequence of two factors:

- reflective surfaces that color the light they reflect, such as golden surfaces; or
- differences in the color irradiated by a pixel of the monitor into different directions.

These effects may vary from pixel to pixel over the image.

One way to make the matte acquisition more robust against color shifts is to take a reference image c_α of the object reflecting just a white image displayed on the monitor. The color information at each pixel is then used to normalize the subsequent values of the corresponding channels of the pixels $c_1..c_4$. Color shift due to any of the mentioned reasons will affect both the c_α and the other images in the same way. The error can therefore be drastically reduced by a per-pixel, per-channel normalization with c_α, at the cost of just one extra image. This normalization has proven to be sufficient for our cases. The values of c_α are also directly used as the the reflectance channel $\alpha(u, v)$ of the matte. Values that are too dark are clamped to 0 and ignored (they are considered to be either outside the silhouette of the object, or part of the object that is not mirroring, or a part of the object that is not lit by the monitor).

5 Geometry Reconstruction

In this section we describe how the geometry is reconstructed from the acquired environment matte. Like in conventional 3D range scanning (e.g., laser based) the geometry will be recovered as a range map, i.e., at every pixel of the environment matte we compute a depth value (the distance between the principle point of the camera and the surface point visible at the considered pixel).

To reconstruct the geometry of the entire surface of the scanned object, multiple range maps are necessary, each obtained by a single matte captured from a different viewpoint. Since we produce the same output data as in conventional 3D scanning (range maps) the same tools can be used to post-process them in order to reciprocally align them, merge them, fill holes, and so on [17, 4].

The process of deriving a range map from an environment matte, i.e., the core process of shape from distortion, can be seen as a mixture of shape from shading and standard triangulation-based range scanning approaches. The transformation of a normal map into a depth map and vice versa is similar to the problems in shape from shading, where reflectance is used to compute a normal map which further is integrated into a depth map. Our method is similar to triangulation-based approach, since the depth to normal relation is given by the triangle determined by the camera, the surface point and the active light source, in our case the point of the monitor intersecting the reflected ray.

The major difference to both approaches is that none of the resulting quantities, depth values or normal directions, are measured directly but rather found by a stable optimization. The depth values are obtained by two major steps. At first, a random initial depth value is assigned to a random point and propagated through the whole matte, at the same time constructing both the depth map and a normal map. Both maps are further refined to obtain a more consistent field. In the second step we optimize the initial depth value with respect to the self coherence of the surface.

5.1 Data Structures and Terminology

We use the following explicit data structures, each organized as a 2D map of the same size as the environment matte (which is also the size of the input pictures):

- a distance scalar field $d(u, v)$, coding in each position the distance from the point of view to the surface of the object (along the viewing ray at position (u, v))
- a normal field $\vec{n}(u, v)$, coding the normal of that point

In the rest of this section we will furthermore use the following terminology:

- as in the previous section, $\alpha(u, v)$ and $\vec{o}(u, v)$ refer to the reflectance and the origin field of the input matte;
- $\vec{r}(u, v)$ denotes the viewing ray through the pixel (u, v) of the matte (obviously this data is not stored, but implicit given by the intrinsic and extrinsic camera parameters)
- the monitor function $\vec{m} : R^2 \rightarrow R^3$ maps the 2D position of a pixel on the monitor $\vec{o}(u, v)$ to the 3D position of the pixel in world space. How to determine $\vec{m}$ is described in Section 6.

At the end of the process, the output consists of the distance field $d(u, v)$, optionally enriched by the normal values of $\vec{n}(u, v)$, or/and by the reflectance $\alpha(u, v)$.

5.2 Converting Distance to/from Normals

Before solving the entire problem, let us first resolve a pair of related, trivial subtasks.

Let us assume that the current point of view $\vec{PoV}$ is in the origin. If we knew the depth at a single point $d(u, v)$ along the normalized ray $\vec{r}(u, v)$ from the origin as well as the origin of the reflected light on the monitor $\vec{m}(\vec{o}(u, v))$, then surface normal at that position $n(u, v)$ can directly be computed as

$$\vec{n}(u, v) = \frac{1}{2}\left(\frac{(\vec{m}(\vec{o}(u, v)) - d(u, v) \cdot \vec{r}(u, v))}{\|(\vec{m}(\vec{o}(u, v)) - d(u, v) \cdot \vec{r}(u, v)\|} - \vec{r}(u, v)\right) \tag{5.1}$$

(see Figure 5.1).

Likewise, given $\vec{n}(u, v)$ we can compute the only value for $d(u, v)$ that satisfies

$$d(u, v) \cdot \vec{r}(u, v) = \vec{m}(\vec{o}(u, v)) - k(2\vec{r}(u, v) \cdot \vec{n}(u, v))n(\vec{u}, v) + \vec{r}(u, v)) \tag{5.2}$$

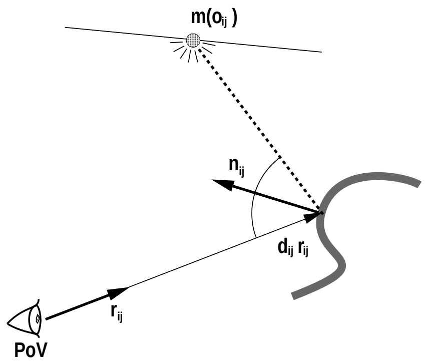

Figure 5.1: Relationship between the surface normal $\vec{n}(u,v)$ and distance $d(u,v)$ (along the ray $\vec{r}(u,v)$ given the 3D position $\vec{m}(\vec{o}(u,v))$ of the light origin (the monitor pixel at position $\vec{o}(u,v)$).

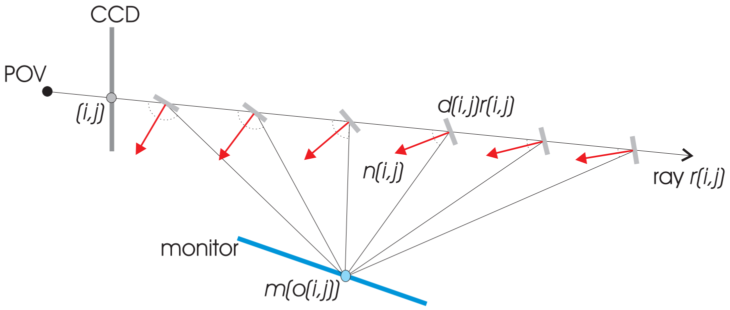

Figure 5.2: Given a ray and a 3D position of the light source, a one dimensional space of solutions, consisting of a surface normal and a distance, are possible.

for some $k > 0$. In other words, the knowledge of the value of the matte $\vec{o}(u, v)$ allows us to uniquely compute the normal from the depth of the surface point visible at (u, v), and the other way around.

There also exists a second way to deduce either of the two fields $\vec{n}(u, v)$ and $d(u, v)$ from the other under certain assumptions about surface continuity (see Section 5.4.1): given the values of $\vec{n}(u, v)$ we can integrate them to obtain the depth field $d(u, v)$ propagating an initial depth estimation, similarly to what is done in shape from shading approaches [26, 13]. Likewise, given the depth values $d(u, v)$ one can easily derive the normal at one point, e.g., as the normal of a plane fitted to the depth values of neighboring pixels.

5.3 Depth Extraction

Unfortunately, we know initially neither the values of $d(u, v)$ nor $\vec{n}(u, v)$. An initial approximation of the two fields is found by propagating an initial guess for the depth value at one point through the whole matte:

1. Start with an arbitrarily chosen initial point (u_0, v_0) inside the matte and assign a random distance $d(u_0, v_0)$ to it. How to determine a proper initial depth value is explained in Section 5.4. A boundary Γ is initialized consisting of only the point (u_0, v_0).

2. Compute the normals using point-wise Equation 5.1 of all boundary points.

3. Compute the depth $d(u_n, v_n)$ of every untouched point (u_n, v_n) neighboring to any boundary point (u_Γ, v_Γ), based on the depth value $d(u_\Gamma, v_\Gamma)$ using the surface tangent implicitly given by $\vec{n}(u_\Gamma, v_\Gamma)$. Add the point (u_n, v_n) to the boundary. If the point does not have any untouched adjacent points, remove it from the boundary.

4. Repeat from 2 until all points are processed;

For a more robust algorithm, before processing a point in Step 3, we wait until $k \geq N_p$ neighboring points around a given point $d(u_n, v_n)$ have been processed (apart from the initial one). Its depth value is then computed as the average of the depth predicted as above, estimated by the k different directions. This produces more accurate results, and is also a natural way to filter out error prone points: if a given point is never reached at least N_p times, we just drop it from the scan. In our implementation we used $N_p = 3$ out of the eight neighbors surrounding one point.

In a second phase we refine the initial approximation of both fields exploiting the dependency of one point on its complete neighborhood until two consistent fields are found:

1. Integrate the current normal field $\vec{n}(u, v)$ into the distance field $d(u, v)$;
2. $\forall(u, v)$ use Equation 5.1 to compute the new $\vec{n}(u, v)$ from the current $d(u, v)$;
3. repeat from 1 until convergence.

In this second phase we take into account the surface continuity along all directions, as opposed to the N_p directions during the boundary propagation in the initial phase.

Note that both phases are based in a point-wise computation of $\vec{n}$ from d (using Equation 5.1), and a local conversion of n to d (using normal integration). In theory, we could have used a dual approach that uses Equation 5.2 instead to obtain d from $\vec{n}$ per pixel, in conjunction with a derivation process (fitting normals to the depth field) to locally determine $\vec{n}$ from d. We prefer the former approach since both involved computations seem to be more robust than the ones of the latter.

5.4 Initial Depth

The discussed algorithm proved to converge fast in all tested cases. In fact it converges "too much": starting with different initial guesses of the depth $d(u_0, v_0)$ at an arbitrary chosen point, different surfaces are reconstructed.

In other words, the per-point ambiguity shown in Figure 5.1 results globally in an ambiguity between large, more or less flat objects far away or small, strongly curved objects close to the camera (see Figure 5.3). In fact, the normals of patches that are assumed to be far away from the camera tend to point towards the camera: it is easy to see that $\vec{n}(u, v)$ in Equation 5.1 converges to $-\vec{r}(u, v)$ when $d(u, v)$ tends to infinity.

The ambiguity consists of a single undetermined scalar value for the entire patch. We aim at solving it without requiring anything more than the information present in a single matte. Although the correct depth could easily be recovered by a second matte done with a second monitor position, this would increase the acquisition time and effort considerably since the monitor position has to be changed and thus recalibrated at each scan (it is not possible to just use two monitors, switching them on one at a time, because, in order for them to light two non-disjoint mattes, they would require a positioning that would make one obscure the other).

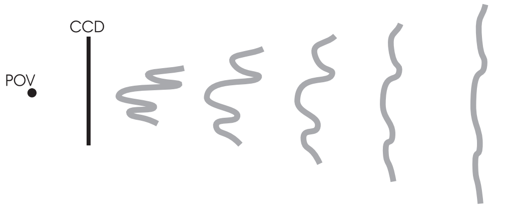

Figure 5.3: Intuitive sketch showing the relationship of the overall shape, size and position of the resulting range scan depending on the initial depth estimate.

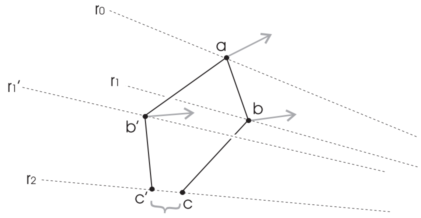

Figure 5.4: The basic case of incoherence that must be summed over the whole surface in all directions to evaluate the total incoherence.

The basic fact we will exploit to obtain the correct depth with just a single matte is that not all of the possible shapes solving the ambiguity are equally self coherent. The form of incoherence we address is the one depicted in Figure 5.4, where the depth estimate of a given points (A) along a ray $\vec{r}_0$ is propagated along two different paths, namely ABC and $AB'C'$, to the same ray $\vec{r}_2$, but leading to different depth values at that ray. For each of the two paths the depth along the starting ray determines the normal at that point, which in turn determines the delta of depth to the next ray. As already described, both in the initial estimation and in the refinement phases we average the C and C' depth values of any considered propagation directions to compute the final, most consistent depth value for C. However, the variance of the values for different paths can be used as the measure of incoherence we need. This measure is further integrated over all points in the matte.

Plotting a graph of this error measure versus the initial depth estimate (e.g. Figure 5.5 left) we always observed a well defined minimum. This minimum determines the correct depth. Since there is no obvious way to treat the function in an analytical way, we need to find this minimum with an iterative method.

A problem is that the error function tends to zero when the initial distance parameter goes to infinity. In fact, if the object is far enough then all the normals will approximately point towards the viewer (see Figure 5.3) and the normal map will be artificially coherent. A natural solution is to weight the measured incoherence with the inverse of the dot product of the view point and the normal

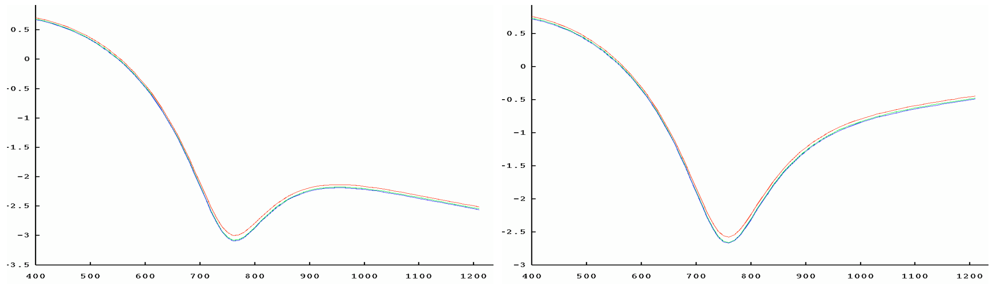

Figure 5.5: A plot of the computed incoherence integrated over the surface and all directions (y axis, in logarithmic scale) versus the initial depth estimation (x axis, in mm.). A minimum clearly exists, corresponding to the real distance of the object from the camera. The three plotted lines correspond to the incoherence of the surface right after the initial estimate, after one refinement, or after 5 refinements (note that the error decreases but the abscissa of the minimum stays the same). The function is shown without (left) and with (right) an extra weighting to take into account the flattening effect for far away objects (see text).

computed at the initial point with that depth, that is, the inverse of the typical delta of depth between two neighboring points. With this adjustment the error function proved to have just a single local minimum (see Figure 5.5 right), so we can apply the Golden Section Search method to find the minimum with great accuracy using few iterations. The weighting sometimes moves the computed minimum by a fraction of a millimeter, but we believe that this estimate is more accurate.

Time can be saved if during each iteration of the Golden Ratio Search only the initial step of the matte to depth map conversion (see Section 5) is executed. The refinement iterations afterwards reduce the error but we never observe any significant change of the position of the minimum (see Figure 5.5).

This measure of self coherence makes intuitively sense and gives good results, that is, the estimated distance to the camera is comparable, but much more accurate, to those that we measured with any empirical method on the physical object. We tested the overall accuracy of the method by using our scanning technique over a known object (see Section 7.2).

Although the method produces highly accurate results it unfortunately breaks down for too small regions. If one region consists of too few connected, usable points the measure of self coherence is no longer reliable, reducing the overall accuracy.

5.4.1 Continuity Assumption

During both the normal-to-depth or depth-to-normal conversion, the algorithm assumes that the surface is first-order continuous.

In practice, we do not really have to worry about gradient discontinuity (angles that are totally steep): real object scans never present them and even when the object itself present very steep angles, these are made a little smoother by the blurring inherent in the image acquisition system. In fact, the problem only occurred when testing the technique on synthetic images, never when using real pictures.

Another case of discontinuity arise for the zero-order (apparent) depth discontinuity due to self-occlusion. Typically depth discontinuities can be observed as separated regions/patches in the matte. At a depth discontinuity the normal of the surface will almost invariably be oriented orthogonally to the view direction, and therefore during matte acquisition it would not show the reflection of the monitor, which in all useful configuration is in front, not behind, the object with respect to the camera. Parts of the object not reflecting the monitor are black in the matte and are therefore automatically ignored by the algorithm.

5.4.2 Disjoint Patches

If the extracted matte presents more than one connected region of usable pixels (we will refer to them as matte *patches*), than each of them will be handled separately, each with a differently determined optimal initial depth. The result will be different patches within the same range and normal map. These patches, even if they are disjointed, are reciprocally aligned (that is, their relative position is fixed), and they can be treated as a single scan for any further processing if they are large enough for a reliable depth estimation.

5.4.3 Interreflections

One more issue is how to solve the problem caused by multiple reflections. The matte captures the reflections of a reflection as well as a simple reflection. In principle, if properly exploited, this could be an advantage rather than a problem, since double reflections provide information on both surfaces reflecting the ray. However it is not clear how to analyze those observations. It would be important, if not to exploit multiple reflections, to at least detect and ignore them.

Some factors make detection possible. First of all, parts of the object lighted (in the pictures, and therefore in the matte) by multiple reflections tend to be separated from those showing single reflections. So, in the vast majority of cases, either the captured colors of all pixels in the patch are due to multiple reflections or all pixel values are due to a single reflection. Another observation is that a patch caused by multiple reflections produces depth maps that show an incoherence (in the sense explained in Section 5.5) several order of magnitude bigger than single reflection patches. Finally, multiple reflection patches typically are also significantly darker. At present, we did not experiment using some, or a combination, of these factors to detect multiple reflection patches automatically.

6 Calibration

In order to reference the measured data, it is necessary to perform the following calibration steps:

- calibration of the intrinsic camera parameters
- calibration of the response curve of the monitor/camera combination
- calibration of the monitor position and shape relative to the camera

The first two steps must be done only once per choice of equipment (camera, monitor) and per lens setting whereas the position of the monitor must be determined at the beginning and whenever one of the two is moved. While our method has been designed so that a given position of camera-monitor could be easily used for multiple scans by just moving the target object, sometimes it may be desirable to move one of them to get a larger scan (i.e. more of the target object reflecting the monitor).

6.1 Intrinsic Camera Parameters

The intrinsic camera parameters such as focal length, lens distortion and central projection point must be known in order to relate a pixel position in the image to a position on space. It is not sufficient to rely on data in the camera and lens specification as the focal length changes for example with the subject distance. We used a standard method similar to Zhang et al. [27] to recover the intrinsic parameter of our camera system.

6.2 Photometric Calibration

Our approach uses color information to infer the screen position that illuminates a surface point. Similar to a shape from shading system, we therefore have to ensure

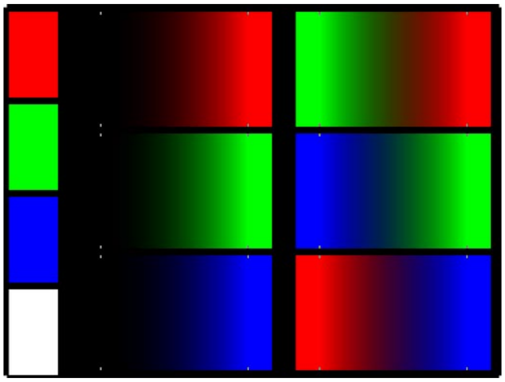

Figure 6.1: Test image used for the calibration procedure. Left: pure red, green, blue, and white color patches. Middle: single color ramps. Right: dual color ramps as used in the patterns.

that the acquisition system is photometrically calibrated. In particular, we have to ensure that the system is linear, i.e., that an image of a linear ramp displayed on the monitor is captured by the camera system again a linear ramp. Furthermore, crosstalk between the color channels in the captured images should be avoided. For our scanner, we therefore have to calibrate both the display and the acquisition system.

Within our imaging pipeline there are two places that require calibration:

- the display system where the RGB values of pixels in the original patterns are interpreted by the computer (which might include application of a gamma factor or an ICC profile) and displayed on the monitor

- the acquisition system where the camera will normally detect a different RGB signal as the spectral responses of the CCD sensors do usually not match the spectral emission of the monitor phosphors. After some color processing by the camera (possibly including the application of a response curve), the raw image signal is available.

To calibrate the display system, we first determine the response curve resp. the opto-electronic conversion function (OECF) of the camera using some standard method (e.g., [18, 7, 12]). If the camera does not have a linear response, applying the inverse of the response curve transfers the images into a linear camera color space.

To remove crosstalk between the three color channels, which is mainly caused by the different spectral characteristics of the monitor and the camera, a simple basis transform in the linear camera color space is sufficient. An image such as Figure 6.1 containing pure red, green, and blue color patches is displayed on

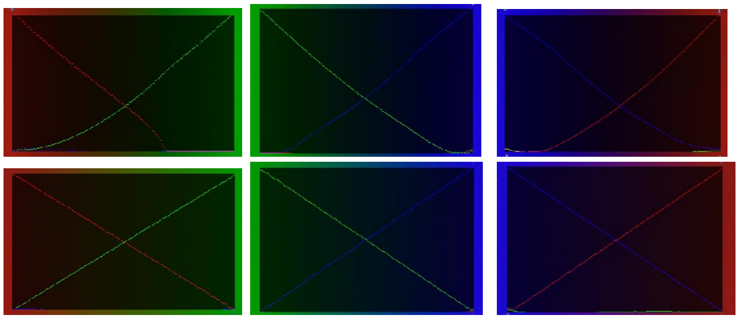

Figure 6.2: The three color ramps red to green. green to blue and blue to red. Top: their first approximation, i.e., linear ramps sent to the monitor and captured by the camera lead to irregular, non linear functions. Bottom: the final ramps after calibration are linear and well separated.

the monitor, captured with the camera and linearized. The corresponding image values R, G, and B are assembled in a basis transform matrix M. The inverse of M can be used to separate the colors:

$$\begin{pmatrix} r_{sep} \\ g_{sep} \\ b_{sep} \end{pmatrix} = \begin{pmatrix} r_{cam} \\ g_{cam} \\ b_{cam} \end{pmatrix}^T M^{-1} = \begin{pmatrix} r_{cam} \\ g_{cam} \\ b_{cam} \end{pmatrix}^T \begin{pmatrix} R_r & R_g & R_b \\ G_r & G_g & G_b \\ B_r & B_g & B_b \end{pmatrix}^{-1} .$$

In theory, this calibration depends also on the reflective material of the target object, different materials requiring different separation. In practice, we found out that, using the precautions about color shift presented in Section 4.3, a single matrix is usually good enough for any material.

To calibrate the display system, we have to capture the response curve of the monitor by displaying a linear ramp for each color channel varying from 0 to the maximal value (see Figure 6.1, middle). By inverting the captured ramp, we can now generate patterns that are displayed as linear ramps on the monitor and are captured and interpreted as linear ramps with a clear separation of the channels.

Any error made (or not compensated for) by the above calibration steps can lead to artifacts such as crosstalk between the color channels and to non-linearities in the captured matte. The quality of the calibration can therefore be judged by analyzing images of dual color ramps which are used in the patterns (see Figure 6.1, right). During our experiments, we achieved a high linearity of the linear ramps and a good separation of the color channels as can be seen in Figure 6.2.

6.3 Geometric Calibration

In the reconstruction phase, we need to know the monitor function $\vec{m}$ (see Section sec:reconstruction), that is, the exact location of each pixel on the monitor relative to the camera in order to interpret the captured mattes correctly. This task can be split in two parts: modeling the position of each pixel relative to the monitor (a one-time calibration) and recovering the position of the monitor. In our case it was sufficient to model the monitor as a flat rectangle, with pixels evenly spaced on it. If the monitor is not flat or regular enough, it might be necessary to model the shape of the monitor more accurately.

An obstacle in detecting the monitor position is the fact that typically the monitor is not directly visible from the camera. Instead, both the monitor and the camera are facing a similar direction, so that the camera can capture monitor light reflected by the target object. The monitor position measurement can still be easily performed using a single picture, with the aid of some mirroring sample objects with known, very simple, geometry such as mirroring spheres.

Given their size and position in a captured image (and the intrinsic camera parameters), it is easy to compute their position relative to the camera using basic projective geometry (see for example [5]). The position of each corner of the reflected monitor on the spheres identifies a reflected ray (the normal of the sphere is known) along which the real monitor corner must lie. Using several spheres in a single image, it is possible to compute the corners as the intersection of these rays. The monitor (modeled as a rectangle of known dimensions) is fitted in those corners. If a flat mirror is used as calibration object, the position of the monitor can be determined in a similar way.

For our experiments we used two metal spheres with 60 mm diameter for the calibration. The spheres are roughly located at the same position and have approximately the same size as the objects to be scanned. This way it is possible to capture the matte after the position calibration without changing the focus setting of the lens which would invalidate the intrinsic calibration.

The calibration remains valid as long as the camera and the monitor are left in place. Fortunately, our technique requires a single matte (which is captured with images all shot leaving both the monitor and the camera in the same position), in order to extract a range scan. Given that the object can be moved, it is possible to keep the camera and monitor fixed and just rotate the object to perform all the needed scans.

7 Results

In this section, we present some results obtained using our technique to scan three objects – a polished metal sphere with 60 mm diameter and two identical plumbery objects, to one of which we added some artificial dents. As the two phases of our algorithm can also be used separately, we will present the results for each phase individually.

7.1 Matte Capture and Extraction

Our measurement setup consists of a standard 19” CRT monitor for displaying the patterns and a high quality Kodak DCS 560 digital still camera (see Figure 7.1). Due to their huge view dependent color shift, most current flatscreen monitors cannot be used for display although they are much lighter and easier to move during setup. All experiments have been performed in our photo studio [9] which has been specially designed to keep the influence of the environment on such measurements as low as possible.

Figure 7.2 shows some results of our matte capture and extraction algorithm. At first, we show the object reflecting the white monitor image c_α (a), and two of the four patterns c_1, c_4 (b,c). In order to illustrate the quality of the extracted matte, we colored each of its points using the reconstructed origin field $\vec{o}(u,v)$ to index a checker pattern (d). Note that the stripes on the sphere are of equal width proving that there is no systematic error in our system when deriving the position of the reflected monitor pixel from image colors. The irregularity in the the middle of the image corresponds to a real, extremely tiny imperfection of the used mirroring sphere. In Subfigure (e), a photograph instead of the checker pattern is used for the mapping. The resulting image can be compared with a real picture showing the original object reflecting the same photograph displayed by the monitor (f). Apart from a color mismatch which we did not correct for, the two images are almost identically.

All mattes have been computed on a standard PC (Athlon 1.4 GHz 512 MB).

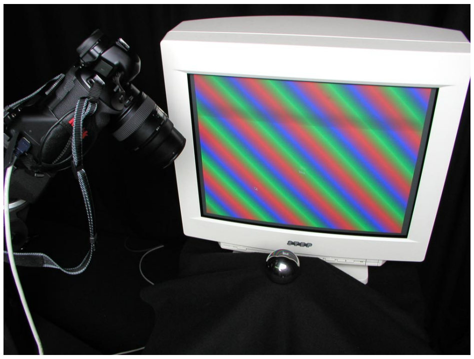

Figure 7.1: The setup used during our experiments. The patterns are displayed on a 19” CRT monitor and reflected off the target object (here the metal sphere in front of the monitor). The images are captured with a high quality digital still camera.

Computing a matte with over 100K points takes less than a second.

7.2 Geometry Reconstruction

A summary of the results of the geometry reconstruction phase is given in Figure 7.3. To check the accuracy of the reconstructed geometry, the 60 mm metal sphere was used as a test target (that diameter of that sphere was known with very high precision). As the table shows, in a range-scan of 75k range points the RMS to an optimal sphere with 60 mm diameter is as small as $21\mu m$. Figure 7.4 shows how the error is distributed over the scan of the sphere. The scan shows a very high accuracy and very low noise. We believe that these small errors are mainly due to inaccurate intrinsic camera parameters, and secondarily due to the errors in the monitor position estimation.

As a test for a more complicated object consisting of multiple, independent patches, we reconstructed the geometry of the plumbery objects. For the first one without dents whose matte is visible in Figure 7.2, we show the reconstruction of the normal field $\vec{n}(u, v)$ and the distance field $d(u, v)$ in Figure 7.5. The figure further displays the corresponding geometry (rendered adding a trivial connectivity to the range points identified by the distance field). Finally, in Figure 7.6 we show some results for the dented plumbery object, concerning both the acquired matte and the reconstructed geometry.

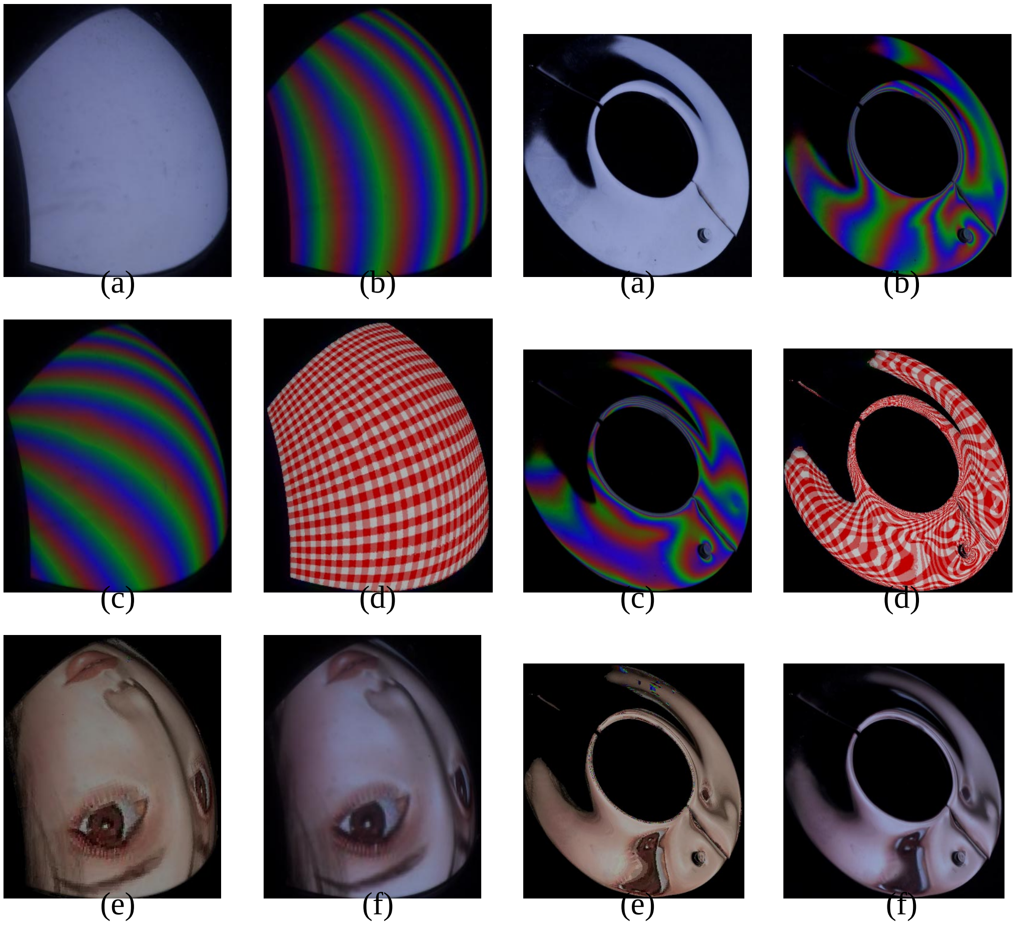

Figure 7.2: Matte captured for a mirroring sphere and a plumbery object. Subfigures (a), (b) and (c) show the input images reflecting patterns c_α, c_1, c_4. In Subfigure (d) and (e), the captured matte is shown using the origin field $\vec{o}(u, v)$ to index a checkered board (d), or an image (e). As reference, (f) shows pictures of the real objects mirroring the image. See Picture 7.6 for another example.

	60mm. sphere matte	plumbery object (clean)	plumbery object (dented)
Number of pixels	75K	38K	108K
Number of Patches	1	2	2
Time Initial Depth Est. (s)	11.9	7.6	20.0
Time Refinement (s)	5.2	3.0	7.5
Time Total (s)	17.1	15.1	27.5
Est. Initial Depth (mm)	756.716	819.640	998.426
Measured Coherence before Refinement (nm)	385	284	220
Measured Coherence after Refinement (nm)	275	263	210
RMS error (μm)	21	—	—

Figure 7.3: Results for three test cases: a matte of the metal sphere, of the plumbery object, and of the dented plumbery object. When multiple patches are present, all data refer to the computation of the largest patch. The coherence row reports, in nanometers, the quantity described in Section 5.4 before it is weighted, averaged over all valid scan points. The number of pixels corresponds also to the number of range points in the resulting range map. The errors for the plumbery objects cannot be quantified for lack of reference data.

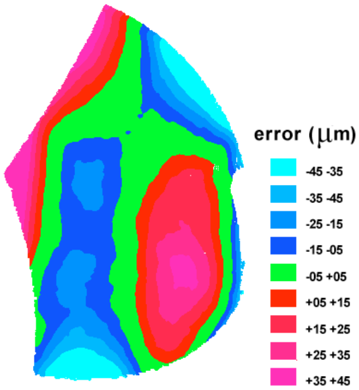

Figure 7.4: Error distribution in a scan of a sphere with 60 mm diameter similar to Figure 7.2 left. The error is shown as signed distance to an optimal sphere in μm.

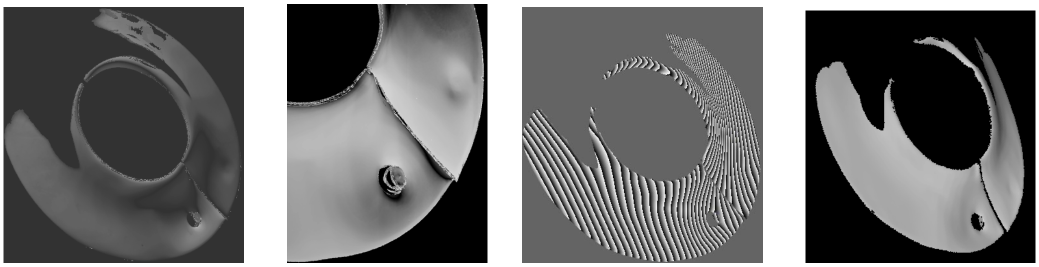

Figure 7.5: Geometry reconstructed from the extracted matte of the plumbery object visible in Figure 7.2. From left to right: the close-up of the normal field $\vec{n}(u, v)$ rendered with Lambertian surface properties reveals even small details. The distance field $d(u, v)$, visualized with isovalued lines. The resulting geometry, $32K$ faces, rendered from a slightly different point of view. (see Picture 7.6 for another example).

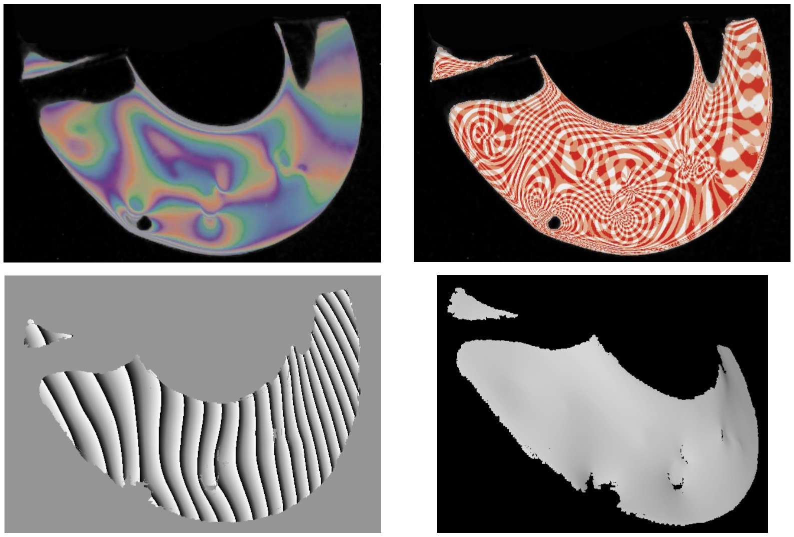

Figure 7.6: Results of measuring the dented plumbery object. From top left to bottom right: one of the original pictures shot to capture the matte, the extracted matte indexing a checkered pattern, the resulting depth map (isolines) and a rendering of the reconstructed geometry.

8 Conclusion

The presented method is a large step towards recovering the surface geometry of a mirroring surface – an important problem not only in computer graphics but also in industrial applications [2]. Due to the basic acquisition principle (an environment matte is related to the first derivative of a surface), the achieved precision is very high and even small details can be recovered accurately.

The acquired scans are range maps that can be dealt directly with existing, well studied techniques developed in other contexts of range scanning. The acquisition time for a single scan is quite reasonable and only standard, cheap hardware components are required.

The technique consists from two components that can also be used independently and in different contexts: The proposed environment matting scheme shows an efficient way to improve the precision of environment matting to sub-pixel accuracy. In cases where the matte extraction is difficult due to the reflection properties of a targeted object, using a more slow and robust environment matting method can improve the acquired matte. Such a matte can nevertheless be used as input for the surface reconstruction.

9 Future Work

The size of the scanned area is limited by the fact that only surface points are scanned for which the reflection of the monitor is seen by the camera (a limitation similar to the ones common in triangulation based 3D scanners). An easy extension of our work would therefore be to replace the monitor with a hemispherical projection screen or a cave. In principle, this requires only an additional calibration phase to recover the geometry of the projection screens.

This limitation poses also new problems on the still largely open field of scan planning: to maximize the surface area acquired by a single scan, one must maximize the portion of the object reflecting the monitor into the camera. A locally optimal position could be automatically found by some appropriate search. Such an automated planning system be especially useful in combination with a robot controlled acquisition environment [15].

One more issue is how to deal with multiple reflections. As already noted, this, if properly exploited, could be an advantage rather than a problem, since double reflections provide information on both surfaces reflecting the ray. But, while detecting multiple reflections is comparatively easy (as discussed in Section 5.4.3), it is quite difficult to integrate them into a coherent depth map.

Another promising research issue consists in the possible use of a similar shape from distortion approach to reconstruct transparent objects such as glass or some types of plastics. The basic concept would be to capture a matte with light from the monitor shining through the object. Its shape is reconstructed by looking at the distortions caused by refraction and reflection. This would be a further significant and very useful extension of the scope of 3D acquisition in general, as transparent objects represent, similarly to mirroring ones, an extremely troublesome yet needed category. It will likely be very challenging to reconstruct the geometry, partly because refractions are more complicated then reflections, partly because each light ray affected by the object at all is likely to undergo multiple refractions and reflections during its monitor-to-camera path. As a side effect, work in this direction could be helpful in exploiting and using multiple reflection matte patches.

Bibliography

[1] Mitsuru Baba, Kozo Ohtani, Makoto Imai, and Tadataka Konishi. New laser rangefinder for three-dimensional shape measurement of specular objects. *Optical Engineering*, 40(1):53–60, January 2001.

[2] Frank Chen, Gordon M. Brown, and Mumin Song. Overview of three-dimensional shape measurement using optical methods. *Optical Engineering*, 39(1):10–22, January 2000.

[3] Yung-Yu Chuang, Douglas E. Zongker, Joel Hindorff, Brian Curless, David H. Salesin, and Richard Szeliski. Environment matting extensions: Towards higher accuracy and Real-Time capture. In *Proc. of ACM Computer Graphics and Interactive Techniques (SIGGRAPH'00), New Orleans*, pages 121–130, July 2000.

[4] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computer & Graphics*, 22(1):37–54, 1998.

[5] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *Proc. of 7th Int.Con. on Computer Vision (ICCV'99), Kerkyra, Greece*, pages 434–442, September 1999.

[6] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proc. of ACM Computer Graphics and Interactive Techniques (SIGGRAPH'96), Los Angeles*, pages 303–312, August 1996.

[7] P. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proc. of ACM Computer Graphics and Interactive Techniques (SIGGRAPH'97), Los Angeles*, pages 369–378, August 1997.

[8] Charles R. Dyer. *Foundations of Image Understanding*, chapter 16, pages 469–489. Kluwer, 2001.

[9] M. Goesele, W. Heidrich, H. Lensch, and H.-P. Seidel. Building a Photo Studio for Measurement Purposes. In *Proc. of the Int. Fall Workshop on Vision, Modeling, and Visualization (VMV'2000), Saarbrücken, Germany*, pages 231–238, November 2000.

[10] H. Hagen, S. Hahmann, Th. Schreiber, Y. Nakayiama, B. Wördenweber, and P. Hollemann-Grundstedt. Surface interrogation algorithms. *IEEE Computer Graphics and Applications*, 12(5):53–60, 1992.

[11] Mark A. Halstead, Brian A. Barsky, Stanley A. Klein, and Robert B. Mandell. Reconstructing Curved Surfaces From Specular Reflection Patterns Using Spline Surfce Fitting of Normals. In *Proc. of ACM Computer Graphics and Interactive Techniques (SIGGRAPH'96), New Orleans*, pages 335–342, August 1996.

[12] ISO 14524 Photography - Electronic still picture cameras Methods for measuring opto-electronic conversion functions (OECFs), 1999.

[13] Reinhard Klette, Karsten Schlüns, and Andreas Koschan. *Computer vision : three-dimensional data from images*. Springer, 1998.

[14] Michael Oren and Shree K. Nayar. A theory of specular surface geometry. *The Int. Journal of Computer Vision*, 24(2):105–124, September 1996.

[15] Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. Scanning physical interaction behavior of 3D objects. In *Proc. of ACM Computer Graphics and Interactive Techniques (SIGGRAPH'01), New Orleans*, pages 87–96, August 2001.

[16] Won S. Park and Hyung S. Cho. Measurement of the 3-dimensional shapes of specular objects using Recursive Triangulation. In *Proc. of Pacific Conference on Manufacturing (PCM'96), Seoul, Korea*, volume 1, pages 389–394, October 1996.

[17] K. Pulli. Multiview registration for large data sets. In *Proc. of Int.Conf. on 3D Digital Imaging and Modeling (3DIM'99), Ottawa, Canada*, pages 160–168, October 1999.

[18] Mark A. Robertson, Sean Borman, and Robert L. Stevenson. Dynamic Range Improvement Through Multiple Exposures. In *Proc. of the Int. Conf. on Image Processing (ICIP'99)*, pages 159–163. IEEE, October 1999.

[19] Kevin Robinson. Laser measures shiny objects in 3-D. *Photonics Spectra*, 35(6):28–29, June 2001.

[20] C. Rocchini, P. Cignoni, F.Ganovelli, C. Montani, P. Pingi, and R. Scopigno. Marching intersections: an efficient resampling algorithm for surface management. In *Int.Conf. on Shape Modeling and Applications (SMI'2001). Genova, Italy*, pages 296–305, May 2001.

[21] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. A low cost optical 3D scanner based on structured light. In *Porc. of Annual Conf. of the European Assoc. for Computer Graphics (EUROGRAPHICS 2001), Manchester, UK*, pages 299–308, September 1999.

[22] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proc. of Int.Conf. on 3D Digital Imaging and Modeling (3DIM'01), Quebec City, Canada*, pages 145–152, May 2001.

[23] Silvio Savarese and Pietro Perona. Local Analysis for 3D Reconstruction of Specular Surfaces. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV'01), Vancouver, Canada*, pages 594–603, December 2001.

[24] Silvio Savarese and Pietro Perona. Local Analysis for 3D Reconstruction of Specular Surfaces - Part II. In *Proc. of European Conf. on Computer Vision (ECCV'02), Copenhagen, Denmark*, May 2002.

[25] Alvy Ray Smith and James F. Blinn. Blue screen matting. In *Proc. of ACM Computer Graphics and Interactive Techniques (SIGGRAPH'96), New Orleans*, pages 259–268, August 1996.

[26] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, August 1999.

[27] Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Int. Conf. on Computer Vision (ICCV'99), Corfu, Greece*, pages 666–673, September 1999.

[28] Jiang Yu Zheng and Akio Murata. Acquiring a complete 3D model from specular motion under the illumination of circular-shaped light sources. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):913–920, August 2000.

[29] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In *Proc. of ACM Computer Graphics and Interactive Techniques (SIGGRAPH'99), Los Angeles*, pages 205–214, 1999.