Exploiting Mirrors in Interactive Reconstruction with Structured Light

Emric Epstein, Martin Granger-Piché, and Pierre Poulin

LIGUM
Dép. I.R.O., Université de Montréal

Abstract

This paper describes how a mirror can be integrated as another view and another source of light patterns in an interactive reconstruction system with structured light, where the object, the camera, and the mirror can move. We show how a single pass of structured light can provide 3D points to accurately estimate the pose of a mirror, while also reconstructing 3D points on the object. We develop new structured light patterns that are unaffected by the reversed order created by some mirror configurations. We also describe hardware rendering support to avoid conflicting emitted/captured light patterns, and demonstrate how all the proposed realizations extend naturally for multiple mirror configurations. We finally conclude with results, discuss limitations, and suggest further improvements.

1 Introduction

Digitizing real objects has been a very active field of research for many years in computer vision [3, 21] and, more recently, in computer graphics [1, 9, 2].

One popular family of reconstruction techniques is structured light [18]. A temporal sequence of light patterns (*e.g.*, black and white vertical stripes) emitted by a video projector are cast onto an object, and the resulting images are captured by a camera. Each illuminated surface element that is visible in a camera pixel is thus associated with a temporal sequence of illuminated/non-illuminated states (*illumination code*) that uniquely identifies the corresponding projector pixels (or vertical stripe). The separation between two adjacent stripes defines a 3D plane, which is intersected with the ray issued from the camera pixel, to result in a 3D point. This requires that a common coordinate system is first extracted from the projector and camera calibrations. This process is illustrated in Figure 1(left).

Structured light shows strong advantages: simplicity, efficiency, precision, flexibility, etc. However structured light can only reconstruct the portions of an object that are both visible and illuminated. To acquire a more complete reconstruction, multiple object/projector/camera configurations are needed. This can be achieved with multiple calibrated cameras/projectors, at an obvious increased cost.

A more common solution consists in moving the object. It requires to precisely estimate the 3D pose of the object or the use of a precise robot arm. Temporal coherency and a robust stitching algorithm such as an ICP (Iterative Closest Point) can also merge together sets of reconstructed 3D points, assuming there is enough overlap between the sets. Such an efficient use of temporal coherency was demonstrated in an interactive system by Rusinkiewicz *et al.* [19]. Other less automatic approaches require user intervention to initially move the sets close enough together to help convergence of the ICP [9].

Moving a calibrated camera allows for the reconstruction of more 3D points from images of changing viewpoints. However, because the light patterns and object remain fixed with respect to each other, only 3D points along the static discretized separation planes can thus be added.

Moving the projector allows for the creation of new 3D points, as light patterns cast different illuminations on the object. Unfortunately a typically heavier projector is more difficult to manipulate, and thus can suffer from less robust calibrations. The moving projector must also be calibrated with respect to the camera coordinate system, which is more difficult than for a camera, as the projector usually does not capture images. Adding a camera to the projector [15] partly solves this problem, but the result is analogous to moving the object itself.

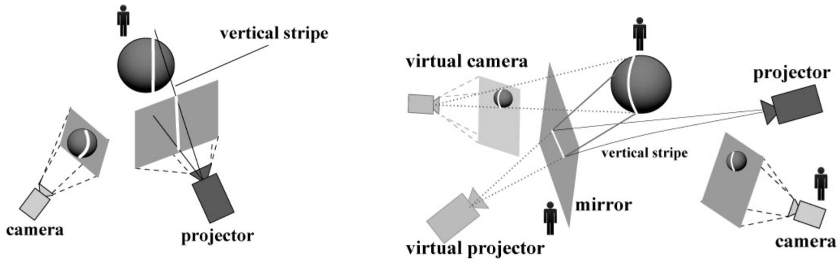

Figure 1: Left: Traditional reconstruction with structured light. Right: Proposed reconstruction with mirrors. The user can interact with the entities marked by a "person" logo.

2 Mirrors in Structured Light

Mirrors offer an alternative to moving either camera or projector. A planar mirror creates an image as seen from a virtual viewpoint, the camera viewpoint being transformed (reflected) by the supporting plane of the mirror. Similarly, light emitted by the projector and reflected by the mirror appears as emitted by a virtual projector.

While the vision and graphics communities have been aware for many years of the potential of mirrors in reconstruction algorithms, few publications have exploited mirrors, except for high-quality laser scanners [20] and specialized stereo cameras [5]. Unfortunately, these techniques require very precise calibrations.

A few recent publications exploiting mirrors in reconstructions include efficient extractions of BRDFs [7], multiple projection surfaces [13], and confocal imaging [8].

In the context of structured light reconstruction, a mirror creates, as illustrated in Figure 1(right) and Figure 2:

- a virtual projector to emit light patterns viewed directly in the camera;
- a virtual camera to view light patterns emitted by the projector;
- a virtual camera to view light patterns emitted by the virtual projector.

Mirrors also offer an interesting alternative in situations where the object cannot move (*e.g.*, large statue or rigid environment), and/or when the projector/camera must reside in a limited set of positions. They also allow for precise camera and projector calibrations to be computed once in a preprocessing step. In all these cases, the burden of the computations is mostly transferred to accurately estimating the 3D pose of the mirror.

In this paper, we present how mirrors can be advantageously integrated in an interactive reconstruction process with structured light. This is to our knowledge the first time mirrors are used in this context. After reviewing the basic mirror equations in the next section, we present our interactive structured light reconstruction system, followed by how we track and accurately estimate the 3D position of the mirror. A mirror can reverse the order in which the light patterns appear on the object. We describe how to adapt the light patterns creation of Hall-Holt and Rusinkiewicz [6] to avoid these problems. As the object may block portions of the mirror, we then show how OpenGL masks are efficiently used to identify reliable mirror regions for the reconstruction. The following section generalizes these concepts for handling interactions in presence of multiple mirrors and how to optimize coverage in limited mirror configurations. Finally, we discuss our results and present future directions to alleviate some of the remaining issues.

3 Mirror Equations

Planar mirror reflection has been well understood since the early days of computer graphics. A 3D point (center of projection of a camera/projector) or direction (view/projection directions) reflected in a planar mirror is computed as

$$[\text{mirrored } p]^T = \mathbf{M}_m [p]^T \qquad (1)$$

with

$$\mathbf{M}_m = \mathbf{T}(d)\mathbf{A}_m^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{A}_m \mathbf{T}(-d)$$

where d is the distance vector of the mirror supporting plane to the origin, and $\mathbf{A}_m$ is the alignment (rotations) of the mirror local coordinate system so it lies in the XY plane in the world coordinate system.

4 Interactive Structured Light Reconstruction System

Our interactive structured light reconstruction system is illustrated in Figure 2. The projector is hand-calibrated using measured points on our room corner; its walls and floor form our *background*. The camera is automatically calibrated using color points projected on this background. Both calibrations extract an affine projection matrix [4, 14]. The object is placed on top of a *stand*. Real color landmarks on the stand are tracked in 2D, and the best rigid transformation (translation and rotation of the stand constrained on the floor) is computed using least-squares fitting. Because we do not have access to a *gen-lock* device[1] to guarantee synchronization between the emitted light patterns and the image captured by the camera, we rely on a special coded light pattern (as captured in Figures 2, 3 and 4, top left corner). This *synchronization code* is projected on the background and deformed to always appear with the same dimensions in a specified corner of the camera image. It also ensures that all three color channels of the projector are properly captured in a single valid image, which is necessary for a DLP projector.

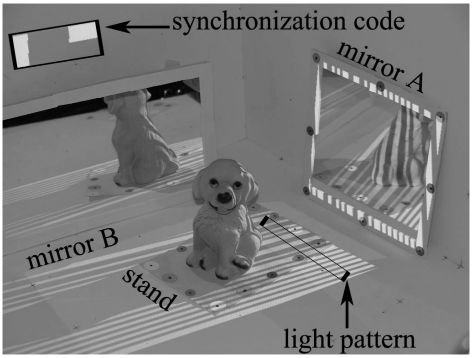

Figure 2: Our interactive reconstruction system.

5 Tracking a Mirror

Our planar mirror is marked with real color landmarks distributed over its non-reflective contour. These landmarks, similar to the ones on the stand, are chosen to be easy to detect and track (see mirror A in Figure 2). While the pose of the stand is robustly computed, thanks to the additional constraint that it lies on the background floor, the limited number of landmarks on the mirror and the precision of their image in the camera proved insufficient so far in our experiments to obtain precise 3D pose estimation.

Our current solution tracks the landmarks in 2D in the camera image, while projecting light patterns over the contour of the latest reliable 3D mirror position. From these light patterns, all reconstructed 3D points between pairs of robustly tracked adjacent landmarks are inserted in a principal component analysis (PCA) to estimate the new supporting plane for the mirror. Landmarks not successfully tracked due to occlusion or noise, are re-estimated from the computed 3D information. If they are effectively recovered from the image, they are reinserted for further tracking. To ensure that all reconstructed points are co-planar, we iteratively remove the points that are farthest from the supporting plane until the variance along the normal of the plane is below a fixed threshold. We use a co-planar measure similar to the one from Pauly *et al.* [12].

6 Light Patterns

We adapt the technique of Hall-Holt and Rusinkiewicz [6] to create unique optimized light patterns, whether their order in the image is reversed or not by the mirror/object configuration. Instead of freely following the edges in the graph

[1] However, one must consider that this solution usually limits the image resolution to a 640×480 interlaced video signal for both the projector and the camera [19].

of illumination nodes, we allow an edge to be traversed at most once.

This simple modification effectively eliminates every pair of adjacent illumination codes that has its reverse order already in the set of illumination codes. Consequently the number of possible codes (for a given number of images) is divided by 2. Hence, we project six images to generate about 400 illumination codes. We use between 150 and 250 of them to avoid identification problems due to aliasing in the camera image. To compute almost seven times more separation planes with the same illumination codes, we continuously shift (slide) the vertical light stripes by one pixel in our 1024×768 projector resolution.

Decoding the illumination code for each camera pixel is also different than in the technique of Hall-Holt and Rusinkiewicz [6]. Instead of checking intensity variations along a scanline, we consider for each pixel the six temporally consecutive images captured by the camera. The minimal and maximal intensity values for each pixel are first identified. Then the six values are distributed in one of the two classes created, according to the distance to these two extremal values. This also allows the system to reconstruct even in presence of textured objects.

7 Computing Masks

Depending on the object/camera/projector configuration, light patterns intended to be cast only onto the object may reach the mirror and reflect back on the object. The object may also block its image viewed in the mirror, or the image of the light patterns projected in the mirror. In all these cases, conflicting light patterns can produce erroneous reconstructions.

In a pre-processing step, the user specifies a bounding box enclosing the object and aligned with the stand. We use this bounding box, the pose estimated mirror, and the projector/camera calibrations to render masks using OpenGL stencil planes and multipass rendering [11] to eliminate conflicting information. Because the stencil plane values cannot be read back in OpenGL, we also compute images of object/mirror IDs while rendering the masks.

Masks (special ID) are computed for (i) the mirror, (ii) the object bounding box, and (iii) the conflicting region, *i.e.*, of overlap between the mirror and the bounding box. Such masks are computed for the camera as well as for the projector.

This is illustrated in Figure 3 for one mirror. The masks are also used in the case of multiple mirror configuration (Figure 4 and Section 9).

8 Computing 3D Points

Sets of light patterns are projected alternately on the object and on the mirror, using the appropriate masks to avoid conflicting light patterns. A set of six consecutive light patterns is necessary to uniquely identify the illumination codes, and therefore to extract 3D points.

To extract object colors and textures, we use the masks (*i.e.*, a seventh image) to properly illuminate the object, and assign a color to each reconstructed 3D point which is taken from the corresponding pixel in the camera image.

In rare occasions, when one image in a set is badly recovered, the seven consecutive light patterns are re-emitted until all seven images are properly gathered.

9 Multiple Mirrors

While the technique works well with one mirror, nothing prevents it from being extended to multiple mirrors. Tracking, pose estimation, mask rendering, and 3D point computation can all be performed, requiring only more processing time for each mirror configuration and careful detection for calibrations and mirror pose estimations.

At this moment, to simplify user interaction, we only allow one mirror to move at a time, while all the other mirrors are considered static, with their pose estimated at an earlier step.

Rendering in the case of multiple mirror interreflections is as easily done in OpenGL as previously stated, thanks to multipass rendering [11]. The masks for multiple interreflections can therefore be computed without technical difficulties. However as multiple mirror reflections reduce the number of valid pixels in the masks, we first make sure that a reasonable number of such pixels (about 400 was considered satisfying) are identified before computing the reconstruction. Otherwise, this portion of the structured light pass and all subsequent ones with the same initial path (*e.g.*, light patterns emitted to mirror A, reflected on mirror B, ...) are

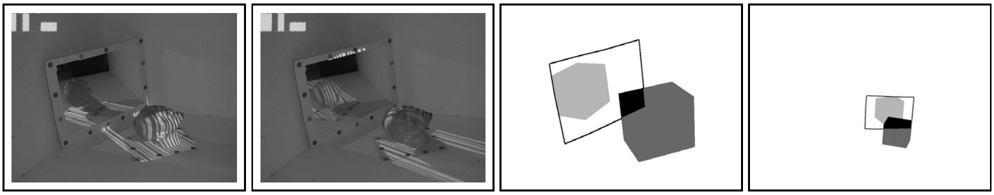

Figure 3: Avoiding conflicts with masks in single mirror configuration. From left to right: direct illumination; indirect illumination; mask for the camera; mask for the projector.

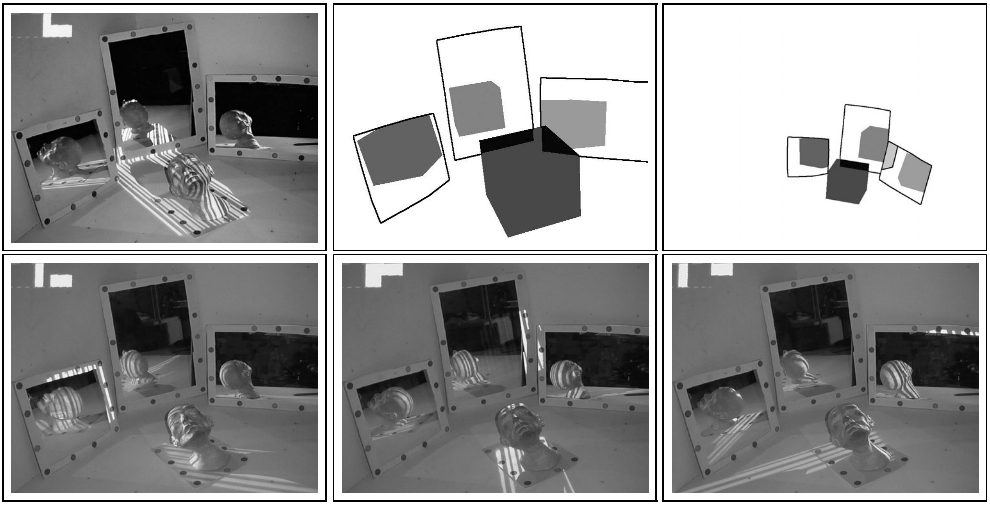

Figure 4: Avoiding conflicts with masks in multiple mirror configurations. (top left) Light patterns projected directly on the scene without touching the mirrors or the top of the head; (top center) mask for the camera; (top right) mask for the projector. Note that each region, including overlaps, is uniquely identified with a color (grey level) and stencil code. Light patterns projected on the (bottom left) left mirror; (bottom center) central mirror; (bottom right) right mirror.

simply ignored, cutting short the explosion of mirror configurations. We also define a maximal number of mirror reflections.

Naturally, at each level of interreflection, calibration, pose estimation, and planarity of the mirrors accumulate errors. While this is a potential concern as the number of interreflections increases, the ICP has proved fairly robust in the (rare) situations where this could be an issue.

Finally, each mirror reflection also attenuates the emitted light or reflected image, thus making it more difficult to differentiate illuminated from non-illuminated surface elements. We did not encounter a situation where this would be critical. The color of a reconstructed 3D point (Section 11) is extracted from the illuminated pixel of the real or virtual image (mirror) it is created from. Differences in the intensity of the colors might however be more observable during display. Automatically correcting for color attenuation in mirrors and incident light direction is part of our future work.

10 Optimizing Mirror Configuration

Positioning object/camera/projector/mirrors to optimize the number of potentially reconstructed 3D points quickly becomes a challenging task for the user, as the number of mirrors increases. Since we can efficiently render our masks and compute the total number of pixels potentially leading to new 3D points, optimizing the configuration of the static mirrors can be computed via simulation instead. To reduce the number of possible configurations, we allow mirrors to only lie on the floor, or to slide/tilt along the walls. A number of sampled configurations are first tested, and then refined where pixels with highest numbers are identified. Figure 5 shows a grey map of the number of reconstructed pixels as two mirrors are moved along the two background walls.

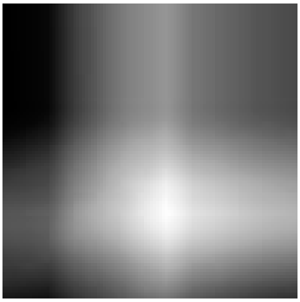

Figure 5: Grey map representing the number of pixels reconstructed as two mirrors are moved along the two background walls. The mirror movements are assigned to X and Y image coordinates.

Another more incremental solution consists in calibrating one real setup, and then letting the computer determine how one of these mirrors can be moved to increase its number of potentially reconstructed 3D points. We found these two basic schemes very useful in our prototype setup. They could be entirely automated in presence of robot-controlled mirrors.

11 Rendering the 3D Points

For interactive display of the current set of reconstructed 3D points, we use GL_POINTS in OpenGL. The color of each point is extracted from the pixel in the original image used to reconstruct it.

For quality rendering, we use splatting. The normal at a 3D point is generated by fitting a plane through the 40 closest points using PCA (Principal Component Analysis). The size of the splat is the distance between its center and the seventh closest point (user-specified). This information is displayed using the hardware EWA splatting of Pointshop3D [17]. To reduce noise in our results, we applied a small moving least-square (MLS) smoothing on our models.

12 Results

In our current implementation, a Panasonic PVGS-70 firewire camera with a 1/60 sec. shutter speed sends interlaced 720×480 images to our computer at 29.97 fps. Our MP4800 DLP Compaq projector supports a resolution of 1024×768 at 2100 lumens. The reconstruction system runs on a dual Xeon 2.4 GHz Pentium IV with 1 GB memory and a GeForce-4 Ti4200 graphics card. One processor is dedicated to video capture while the other is dedicated to image analysis, mirror tracking, and reconstruction.

Figure 6 shows how model surface coverage is noticeably improved when inserting more mirrors to the setup. Table 1 gives for this example approximate reconstruction times, number of reconstructed points, and number of structured light passes (*i.e.*, a sequence of seven images identified and used for reconstruction and color extraction).

Table 1: Statistics for the head statue

number mirrors	reconstruction time (sec)	number points	number passes
0	15	16k	14
1	26	21k	18
2	36	28k	28

Table 2 lists detailed results for the models in the color plate. The approximate time reported for reconstruction includes mirror calibration, stand pose estimation, and reconstruction. More examples can be found on the website associated with this paper.[2]

Models *A*, *B*, and *C* were obtained with a 3-mirror configuration similar to Figure 4, and by moving the stand 2 to 3 times. The tiger (model *C*) illustrates that we can reconstruct textured objects, which is a limitation of the original *stripe boundary code* decoding algorithm [6, 19]. The Chinese statue (model *D*) is reconstructed only with mirrors, *i.e.*, without moving the stand. This can prove very

[2] www.iro.umontreal.ca/labs/infographie/papers

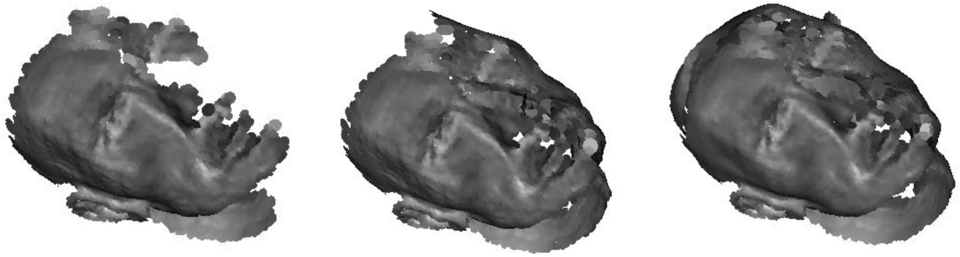

Figure 6: Reconstruction within a unique configuration; only the number of mirrors changes. (left) no mirrors; (center) 1 mirror; (right) 2 mirrors.

Table 2: Statistics for various reconstructions in the color plate

	model	number points	reconstruction time (mins)	number mirrors	stand positions	number passes
(A)	Dog	61k	10	3	3	46
(B)	Head	41k	7	3	4	30
(C)	Tiger	36k	15	3	4	155
(D)	Chinese statue	61k	30	2	1	265

useful when moving the real object is difficult or impracticable.

13 Conclusions and Future Work

This paper reports on integrating mirrors in an interactive reconstruction system using structured light. While our prototype yields limited reconstructions (even without the mirrors) compared to some finely-tuned structured light systems, it allowed us to investigate a number of important issues regarding mirrors in this context. The resulting system proves to be an interesting avenue to apply structured light in general, as well as in special situations where the object/projector/camera are limited in their positions/movements. It also demonstrates how mirrors can be efficiently exploited in most reconstruction algorithms.

Our contributions, besides introducing mirrors in structured light, include the design of unique illumination codes unaffected by the potential order inversion due to mirrors, efficient masking operations to avoid conflicts in identification of the illumination codes in all reflection modes, a measure of effective pixels to determine if mirror reconstruction could contribute sufficiently and to optimize the configuration of the mirrors, and the extension to multiple mirror configurations.

While we focused on an interactive system with the user getting immediate feedback (display) of the current reconstruction, nothing prevents our system from being more automatic. A mirror calibration feedback loop could be designed by projecting pixels of light on the mirror and measuring the difference between its expected and actual reflections over the known background. A technique by Mitchell and Hanrahan [10] could therefore help to correct even more the estimated mirror pose.

Synchronization with a *gen-lock* card should lead to faster results. Infrared light (easily emitted with any projector with a visible light filter) in conjunction with an infrared camera would help to reduce the constant light flickering, which is annoying in an interactive reconstruction mode [6]. Another solution can use a time-multiplexed light cancellation technique [16]. Multiplexing color light might also be another way to capture fewer images while obtaining the same information.

An extension involving curved mirrors and refractive lenses might allow for better focusing of light patterns on some regions of space. The optics of non-planar separation between illumination codes should prove quite challenging though, so we would have to pay particular attention to the focus distances of both camera and projector.

14 Acknowledgment

We would like to thank Mathieu Ouimet and Denis Vontrat for their help. We acknowledge financial support from FCAR and MITACS.

References

[1] F. Bernardini and H. Rushmeier. 3d model acquisition. In *State of the Art Reports – Eurographics*, pages 41–62, 2000.

[2] F. Bernardini and H. Rushmeier. The 3d model acquisition pipeline. *Computer Graphics Forum*, 21(2):149–172, 2002.

[3] Q. Chen and G. Medioni. A volumetric stereo matching method: Application to image-based modeling. In *Proc. Computer Vision and Pattern Recognition*, pages 29–34, June 1999.

[4] O. Faugeras. *Three-Dimensional Vision — A Geometric Viewpoint*. MIT Press, 1993.

[5] J. Gluckman and S.K. Nayar. Catadioptric stereo using planar mirrors. *International Journal of Computer Vision*, 44(1):65–79, 2001.

[6] O. Hall-Holt and S. Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Proc. ICCV 2001*, 2001.

[7] J.Y. Han and K. Perlin. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Trans. on Graphics*, 22(3):741–748, July 2003.

[8] M. Levoy, B. Chen, V. Vaish, M. Horowitz, I. McDowall, and M. Bolas. Synthetic aperture confocal imaging. *ACM Trans. on Graphics*, 23(3):825–834, August 2004.

[9] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. In *Proc. SIGGRAPH 2000*, pages 131–144, July 2000.

[10] D.P. Mitchell and P. Hanrahan. Illumination from curved reflectors. In *Computer Graphics (Proc. SIGGRAPH 92)*, volume 26, pages 283–291, July 1992.

[11] OpenGL. www.sgi.com/software/opengl/advanced98/notes/node125.html.

[12] M. Pauly, M. Gross, and L. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proc. of IEEE Visualization*, 2002.

[13] Claudio Pinhanez. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *Proc. Ubiquitous Computing 2001*, volume 2201, pages 315–331, 2001.

[14] R. Raskar. Camera calibration. *SIGGRAPH Course Notes 21*, July 2003.

[15] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. iLamps: Geometrically aware and self-configuring projectors. *ACM Trans. on Graphics*, 22(3):809–818, July 2003.

[16] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proc. SIGGRAPH 98*, pages 179–188, July 1998.

[17] L. Ren, H. Pfister, and M. Zwicker. Object space EWA surface splatting: A hardware accelerated approach to high quality point rendering. In *Proc. Eurographics 2002*, pages 461–470, 2002.

[18] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. A low cost 3d scanner based on structured light. *Computer Graphics Forum*, 20(3):299–308, 2001.

[19] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. *ACM Trans. on Graphics*, 21(3):438–446, July 2002.

[20] C. Samson, I. Christie, J.-A. Beraldin, and F. Blais. Neptec 3d laser scanner for space applications: Impact of sensitivity analysis on mechanical design. In *Proc. of SPIE: Optoelectronics, Photonics, and Imaging*, pages 29–34, May 2002.

[21] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, August 1999.