Appeared in the Proceedings of the Eurographics Symposium on Point-Based Graphics 2005
M. Pauly, M. Zwicker, (Editors)

A Practical Structured Light Acquisition System for Point-Based Geometry and Texture

Filip Sadlo, Tim Weyrich, Ronald Peikert, Markus Gross

Computer Science Department, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland

Abstract

We present a simple and high-quality 3D scanning system based on structured light. It uses the common setup of a video projector, a computer-controlled turntable and a single camera. Geometry is acquired using a combination of Gray code and phase-shift projections, and it is stored and processed in a point-based representation. We achieve high accuracy by careful calibration of camera, projector, and turntable axis. In addition, we make use of the projector's calibration and extend it to a calibrated light source, allowing for a simple reconstruction of material properties for each surface point. We alternatively use a Lambertian reflectance model, or fit a Phong reflectance model to the samples under different turntable orientations. The acquisition pipeline is entirely point-based, avoiding the need of triangulation during all processing stages.

Categories and Subject Descriptors (according to ACM CCS): I.4.1 [Image Processing and Computer Vision]: Scanning

1. Introduction

3D Reconstruction of real objects plays an important role in many fields. While applications in industry usually concentrate on accuracy but are tolerant to scanning artifacts as long as they do not affect operation, applications in graphics usually favor the correctness of the visual appearance and hence are tolerant to errors as long as they do not produce visible artifacts. However, the high sensitivity of the human visual system requires relatively high reconstruction accuracy also in graphics applications. Our system targets precision requirements of both, graphics and industry.

We have chosen to use structured light for acquisition because it allows robust and accurate acquisition of objects with arbitrary geometry and a wide range of materials. It has also the benefit that geometry and texture can be acquired with the same camera. Another benefit is the low cost compared to other systems, since video projectors are often disposable and because only a single camera is needed.

The representation of 3D objects as sets of surface elements or *surfels* [PZB*00] has several advantages compared to meshes or implicit representation. Many operations get simpler or more efficient on point-based representations. As structured light scanners usually acquire point clouds, it is a straightforward decision to store the result in a point-based representation. This way, we provide a suitable acquisition back-end for a point-based content creation pipeline as proposed by [ZPK*02] and [WPK*04].

In this paper, we present a point-based acquisition stage using custom hardware. The object is rotated by small known steps and for each position, the view is reconstructed using structured light. Unlike many approaches, we avoid the demanding and error-prone task of mutually registering the single reconstructions. Instead, we bank on the precise calibration of projector and camera and additionally calibrate the axis of the turntable. This allows us to produce consistent multi-view reconstructions (rings of overlapping views). Then some methods for the removal of artifacts are proposed and applied. We propose an efficient method for merging the overlapping reconstructions into a single-layered surfel representation. For the reconstruction of texture, we add photometric calibration to the already computed geometric calibration of the projector. This way, a calibrated light source is obtained which is used for the per-surfel reconstruction of either Lambertian texture or texture according to the Phong reflectance model. Finally, we present a method for analyzing the geometric accuracy of the system.

1.1. Related Work

Many approaches to 3D reconstruction have been used so far. There are mechanical probing systems but most systems are based on optical triangulation. Some image-based methods are passive, including visual hull, stereo-triangulation, shape from shading, shape from photo-consistency, voxel coloring, space carving [KS00] (see [Dye01] for an overview). Finally there are active optical methods including laser range scanning and structured light. Some of these systems also reconstruct texture ([SWI97], [PAD*98], [BMR01]; see [MR02] for an overview), while others acquire only geometry such as [RHL02]. Some systems make use of the temporal evolution of the reflected codes such as the space-time analysis by [CL95]. There are also semi-active methods such as [MPN*02] using backlighting and image-based appearance representation. The system of [ES03] uses a deformable model to refine an initial reconstruction and to fuse texture and geometry. Most systems produce mesh representations although some generate point clouds such as [SPM04].

There are several variants of structured light codifications [SPB04], [BMS98], [DRR03]. For better generality in terms of object properties, we work with binary monochrome codes only, instead of using color or gray levels for direct codification. Since we focus on static objects, we can make use of time-multiplexing. Therefore we have chosen the classical Gray code method combined with the phase-shifting method for the benefit of higher robustness.

Calibration of projectors is difficult compared to calibration of cameras. While cameras are often calibrated according to Zhang [Zha00] using views of a pattern of known geometry, different approaches have been used for the calibration of projectors, such as controlled movement of a projection plane [SM02] or simultaneous estimation of parameters [LBJ04]. We have chosen to use the Zhang model for camera and projector by calibrating the projector as an inverse camera.

2. System Overview

The acquisition system consists of a video projector that projects the structured light patterns, a turntable that rotates the object, a camera that takes images of the projections and a computer that controls the hardware and does the reconstruction. Since both the user and the projector need a video output, the system can be controlled by a single computer with multiple video output, or a second networked computer is used for operating the projector, as in our current setup.

We use an analog-input 1024x768 DLP video projector. It is beneficial if the projector allows for disabling features such as automatic synchronization and image size adaption. This way the calibration is not lost when the projector is turned off between scans. Another aspect is the projector's minimal image diagonal, or in other words, the minimal focal distance. This becomes important if small objects have to be acquired. We currently use an IEEE-1394 video camera with a resolution of uncompressed 1024x768 pixels to acquire the images. We prefer using monochrome cameras and illuminate the object (or calibration pattern) with red, green and blue projector light in order to acquire colors. The reason is that cameras using a Bayer pattern for color acquisition (each pixel has a red, green or blue filter attached) exhibit increased blur and artifacts due to undersampling and due to interpolation by the demosaicing process. This complicates calibration, smoothes the reconstruction and can produce artifacts comparable to Figure 7. For similar reasons, we also avoid rectification of the images and model the lens distortion instead.

Several rings of reconstructions are needed in order to get a complete reconstruction. This is accomplished by either tilting the object, tilting the turntable or by moving projector and camera to a new position. The optimal solution is to accomplish these transformations again around a calibrated axis and to apply the transformations to the reconstructions. But since this tends to be mechanically demanding, we tilt object or turntable by hand and each time acquire a ring of reconstructions by rotation of the turntable. If the turntable is tilted, it has to be recalibrated, but this is a fully automatic procedure and makes the resulting rings of reconstructions consistent, hence no registration is needed. If the object is tilted instead (as in our current application), no recalibration of the turntable is necessary but the resulting rings have to be mutually registered while treating each ring as a separate rigid model. Some methods for mutual registration of reconstructions are geometry-based such as *ICP* [BM92] and [Pul99], others are image-based such as [BMR01]. We use ICP. Some challenges of registration such as error accumulation play little role in this case because the rings are already fitting at high accuracy, leading to a simpler registration problem. After the rings have been merged to a single-layered surfel representation, reflectance samples for each surfel are collected into *lumitexels* [LKG*01] and used for texture reconstruction according to the Phong reflectance model, which is then rendered using PointShop3D [ZPK*02].

3. Calibration

Although often underestimated, precise calibration of the structured light system is the main prerequisite for a successful and accurate reconstruction. This is especially true for multi-view reconstructions, where a given surface point is reconstructed several times and the corresponding reconstructions have to meet. In order to increase accuracy and to ease calibration, we do separate intrinsic calibration of camera and projector and then do an extrinsic calibration step.

We use the Zhang calibration method as implemented by OpenCV's cvCalibrateCamera for intrinsic and extrinsic

calibration of the camera and the projector. This calibration model consists of focal length with respect to pixel widths and heights, the principal point and a radial and a tangential lens distortion modeled by two parameters each. As proposed by OpenCV, a chessboard pattern is used for calibration. However, the pattern detection algorithm of OpenCV fails under difficult lighting or oblique viewing conditions. This conflicts with our experience that strong perspective views at oblique angles increase the accuracy of (intrinsic and especially extrinsic) calibration. It also conflicts with our need to detect a printed pattern and a (distorted) chessboard projection in the same view (Section 3.3). Therefore, we propose an alternative detection procedure based on projective geometry.

3.1. Detection of Calibration Pattern

OpenCV's detection of the calibration pattern inside the images is done in two steps: First, approximative guesses of the chessboard corners are computed. Then, the corners are detected at sub-pixel accuracy from the guesses. The second step works well, but the first step fails under difficult light or viewing conditions. To ease the pattern detection, we put identical color marks onto the four corners of the chessboard pattern and an additional one for defining the origin, as shown by Figure 1. We have chosen colors instead of spatial marks because of the invariance of color with respect to projective transforms.

The color marks are segmented by a user-defined HSV color range, which has to be determined only once for each new pattern and new projector by interactive color picking. After the centroids of the marks have been computed, the origin mark is determined by their convex hull and used to select one of the four marks as origin mark. This is done by detecting which of the two diagonals (one as dotted line in Figure 1) is closer to the marker. Then the four corner marks are reordered accordingly and used to find the four outermost inner corners of the chessboard (next to the detected marks). This is done based on cross-ratio as described at the end of this section. The resulting positions are refined to sub-pixel accuracy using OpenCV. From these four corner positions, all inner corners of the pattern are found again using cross-ratio and also refined using OpenCV.

Positions inside the chessboard image are computed based on cross-ratio as follows (Figure 1): The origin of the pattern's coordinate system is at m_1, its first axis u is in direction from m_1 towards m_2 and the second axis v is in direction from m_1 towards m_3. And let u and v be in $[0, 1]$ if inside the polygon spanned by the corner marks m_i. For a point (u, v) in pattern coordinates, its image coordinates (s, t) are computed by first computing s_0, s_1, t_0 and t_1 from the corners and the vanishing points using equation (1) and then intersecting the gray lines in Figure 1.

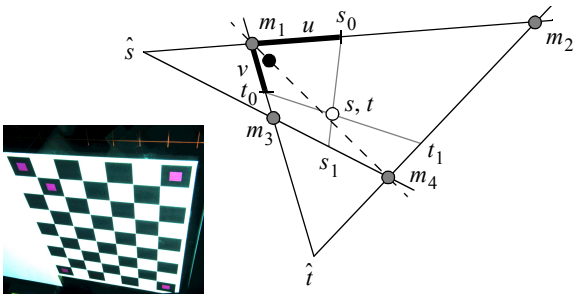

Figure 1: *Detection of calibration pattern. Left: Printed pattern with color marks (image postprocessed). Right: The gray dots represent the marks, the black dot represents the origin mark detected by their convex hull. Vanishing points $\hat{s}$ and $\hat{t}$ are computed from the corner marks.*

$$s_i = \frac{1}{1 + \frac{1-u}{u} \cdot \frac{\left\| m_{2i+2} - \hat{s} \right\|}{\left\| m_{2i+1} - \hat{s} \right\|}}, \quad t_i = \frac{1}{1 + \frac{1-v}{v} \cdot \frac{\left\| m_{i+1} - \hat{t} \right\|}{\left\| m_{i+3} - \hat{t} \right\|}} \tag{1}$$

3.2. Intrinsic Camera Calibration

The camera is calibrated using different views of a printed chessboard pattern as described in the previous section. We illuminate the pattern by the projector, in order to create similar lighting conditions as in the case of projector calibration (Section 3.3, Section 3.4). This allows to choose similar camera settings and color parameters for the detection in both calibration steps.

3.3. Intrinsic Projector Calibration

We calibrate the projector as an inverse camera. This means that instead of taking pictures of a chessboard with known geometry and detecting the corners inside the images, a chessboard pattern with known geometry is projected to different orientations and positions of a plane and the projections are measured with the calibrated camera.

We project color-marked chessboard patterns according to Section 3.1 and use the same corner detection algorithm to find the corners inside the images of the projections. Figure 2 shows such a camera image. In order to measure the 3D positions of the projected pattern with the calibrated camera, we attach a printed chessboard pattern to the projection plane and use extrinsic calibration to determine its position. Then, the corners of the projected pattern are detected at sub-pixel resolution inside the camera image and projected to the projection plane. The resulting points are fed into the OpenCV camera calibration routine.

Because the calibration of the camera is used for the calibration of the projector, the error of the camera calibration affects the accuracy of the projector calibration. This often

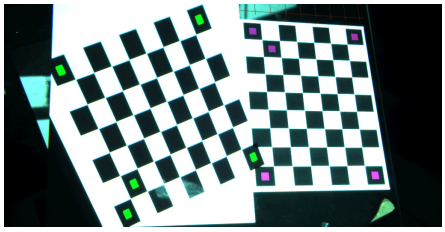

Figure 2: *Projector Calibration. Left: White projection area with projected pattern using green marks. Right: Printed pattern with purple marks on the same glass plate.*

leads to re-projection errors of the projector calibration that are one order of magnitude larger than those of the camera calibration. Hence extreme care has to be taken during camera calibration.

3.4. Extrinsic Calibration of Projector and Camera

The orientation of projector and camera relative to each other is determined in a similar way as in the projector calibration (see Section 3.3 and Figure 2), but with a single image. And this time both, not only the camera, are calibrated extrinsically.

3.5. Turntable Axis Calibration

Calibration of turntables is approached in different ways such as using markers permanently attached to the turntable [SPM04], or by fitting an axis to rotated reconstructions. We put a color-marked chessboard pattern horizontally on the turntable and do a full rotation in a given number of steps, typically twelve. For each step, the position and orientation of the camera relative to the pattern is computed by extrinsic calibration. Then a circle is fitted to the resulting ring of virtual camera positions. The rotational axis of the circle represents the axis of the turntable.

3.6. Luminous Projector Calibration

Because the object is illuminated by the projector for texture acquisition, we have to know the irradiance from the projector at a given point in space. For each new projector, we initially calibrate for the luminous intensity I of the projector pixels. This could be done using a calibrated reflection target, e.g. Spectralon. As our application does not require absolute physical quantities, we use gray cardboard instead and scale I to map color intensities into a useful range. We implemented two calibration modes. I is either assumed to be identical for all pixels, or I is determined on a per-pixel basis, capturing spatial intensity variations at the cost of more noise. The irradiance E at a given surface point is then computed as follows: $E = I/d^2$ with d its distance to the projector's center of projection.

4. Geometry Reconstruction

As already mentioned, geometry is reconstructed using structured light according to the Gray code and phase-shifting method described in Section 4.1. Normals are actually computed from the weighted positions of neighboring samples by plane fitting. They are computed separately for each view before merging the reconstructions according to Section 5.2. This avoids influence of registration errors.

4.1. Gray Code and Phase-Shifting

Structured light methods make use of a projection device to determine the z-depth of every illuminated camera pixel. This is done by optical triangulation of the camera ray with the corresponding projector ray that illuminated the surface element. There are several possibilities to structure the light in order to allow the identification of a projector pixel by its light. We have chosen to do time-multiplexing of gray level codes. Therefore, our method is limited to static objects.

In the standard Gray code algorithm (see Figure 3 left), the ray defined by a camera pixel is intersected with the plane defined by the corresponding projector column. However, this assumes no lens distortion inside the projector because otherwise the plane would be distorted. To make the reconstruction process more robust against lens distortion and decoding errors, we encode both, projector columns and rows. The plane-ray intersection problem becomes an overdetermined ray-ray intersection problem which also allows for the removal of artifacts by the ray-skew criterion as described in Section 4.2.

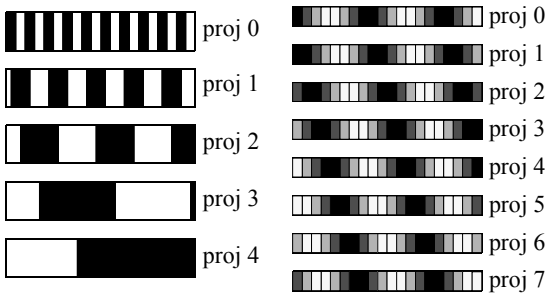

Figure 3: ***Gray-coded structured light*** *(left): Each projected image represents a bit-plane of the code. This way the temporal signal at a given object point represents the column of the corresponding projector pixel. The procedure is repeated with horizontal stripes to also encode the projector rows.* ***Phase-Shifting*** *(right): Shifted sinusoidal stripe patterns are projected to the object, producing a temporal sine signal on a given object point. The phase of this signal represents the projector column. The procedure is repeated with horizontal stripes to also encode the projector rows.*

Since the projected Gray codes are binary, the achievable precision is limited to integer projector coordinates. Therefore, the detected pixel relations are subsequently improved

using phase-shifting (see Figure 3 right). This fixes decoding errors in the least significant bits of the Gray code. In addition, phase-shift reconstruction even allows to determine the projector coordinates at sub-pixel accuracy. We also experimented with line-shifting [GBB*00], which achieves sub-pixel accuracy in camera coordinates rather than in the projector domain. However, in our setup it generally produced inferior results.

Due to errors in calibration and decoding, the two corresponding rays usually do not intersect. The method presented by [Güh01] addresses the problem by nonlinear least squares and analysis of the residual, while [HS97] give an overview and introduce a polynomial method. We compute the point of intersection as the point on the camera ray that is closest to the projector ray. We force the solution to the camera ray because its calibration can be assumed more accurate than that of the projector rays. This way the projector only contributes depth information, which meets the original intention. We use the distance between 'intersecting' rays (ray skew) for the removal of artifacts as described in the next section. We also visualize the ray skew by color-coding the reconstructions for visual verification of the calibration quality.

4.2. Artifact Removal

Here some methods for the elimination of geometric artifacts are described. They are all applied to the single-view reconstructions before they are merged to a single reconstruction according to Section 5.

Signal strength. A black reference image with all projector pixels black and a white reference image with all projector pixels white are taken for each view. They are used for elimination of camera pixels that receive no or too weak signals and they are also used for normalization of the gray-code signal. A camera pixel is eliminated if the white reference differs from the black reference less than a user-defined threshold. The threshold is chosen in order to reject mainly background and part of the shadows.

Ray skew. This method detects artifacts caused by decoding errors as well as artifacts that are produced by reflected or scattered codes. Assuming accurate calibration, it is unlikely that the projector ray corresponding to the falsely decoded code intersects with the camera ray as well as the correct projector ray would do. In other words, the ray skew tends to increase on decoding errors (Section 4.1). We use a threshold for the minimal distance between 'intersecting' rays and reject the reconstruction if the threshold gets exceeded. (Figure 4).

Sub-pixel variance. This method addresses artifacts that originate during phase-shifting from object regions with varying reflectance and curved (discontinuous) surface as described by [CL95]. The phase-shift signal is spatially integrated over the area of each camera pixel during acquisition. Assuming that the left half of the camera pixel looks, e.g., at black material while the right half observes a bright material, the signal integrated over the pixel will contain only codes from the right half, leading to wrong projector rays usually visible as depth error. As we acquired overlapping views, we are able to address this problem by eliminating surfels that lie on edges in image space: The pixel corresponding to the surfel is re-sampled at sub-pixel resolution using Lanczos interpolation, and its variance is computed. It is rejected if its variance exceeds a threshold. The method also removes sharp shadow boundaries.

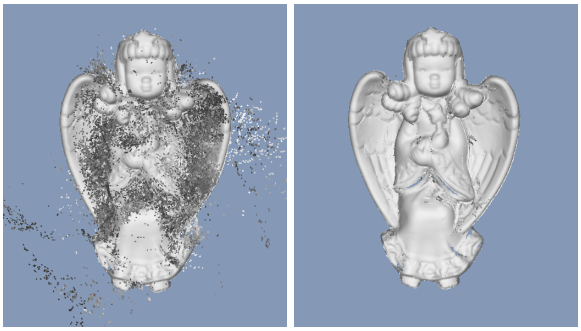

Figure 4: *Removal of artifacts based on ray skew and component size. Left: ray skew and component size is unlimited (*2,527,362 *surfels). Right: ray skew limited to 0.6 mm and component size to six surfels (2,053,543 surfels).*

We also remove outliers based on geometric criteria such as detecting small connected components of surfels and we also reject saturated pixels from the reconstruction process.

5. Merging Reconstructions

After the rings of reconstructions have been registered, we want to merge the overlapping reconstructions into a single-layered surfel representation in order to reduce storage and visualization cost. Doing so, we have to take care that texture quality and geometric accuracy remain as high as possible.

Blending of the overlapping textures and averaging the overlapping surfel positions would require that the overlapping reconstructions fit together at sub-pixel accuracy, otherwise the resulting reconstruction would get blurred or doubled. Therefore, we try to preserve accuracy by combining original patches of the overlapping reconstructions, as described in Section 5.2. This method relates to [TL94] and it also has a thinning effect because the patches have original resolution.

5.1. Bounded Projective Nearest Neighbor

We use a nearest neighbor search to get the overlapping surfels of a given surface element of the object. Instead of performing a standard nearest neighbor search, it is more efficient to perform the search in the M camera images. Because every point is reconstructed by a corresponding

camera pixel, we can make use of the calibrated setup. To find for a given point $\mathbf{p}$ and a given search radius r the nearest (or all) point reconstructions, we project $\mathbf{p}$ to all camera images, yielding a $\mathbf{p}'_i$ for each view i. Then all point reconstructions that have been reconstructed by camera pixels within a search radius r'_i around $\mathbf{p}'_i$ in image space are tested for whether they lie within distance r to $\mathbf{p}$ in world space. The search radius r'_i is computed from r by projection.

The complexity of the search is $O(M \cdot r'^2)$, where M is the number of camera images (r'^2 is limited by the number of pixels per image). In our current application, the search radius is small and constant. Therefore r' has a small upper bound, resulting in an algorithmic complexity of $O(M)$.

5.2. Best Surfels

This section describes our simple greedy approach to select patches from overlapping reconstructions in order to get a single layer of points and for each object region the best patch regarding the quality of geometry and texture. Our current approach is purely surfel-based, hence it does not maintain a volumetric representation such as [PDH*97].

The algorithm defines and selects the patches implicitly in two steps: First, the homologous points representing the same point on the surface are determined using the nearest neighbor search presented in Section 5.1. Then the best of these candidates is chosen for the resulting reconstruction.

The method is based on a simple idea: For each object point, take the surfel that has been reconstructed most orthogonally by its camera (regarding its surface normal computed from its neighbors). This addresses the fact that texture and geometry have highest resolution when acquired by perpendicular view. At the same time, this simple criterion forms the patches. Figure 5 shows the pseudo code of the algorithm while Figure 6 shows an example result. There are holes due to the greedy nature of the algorithm. However, the holes are small enough to disappear when rendered.

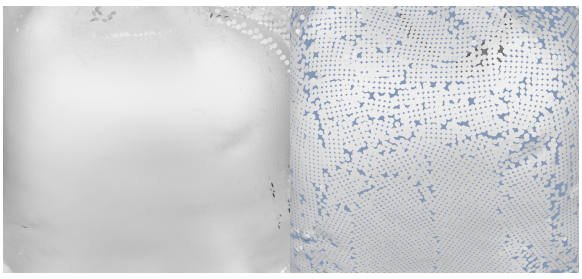

Figure 6: *Best Surfels. Left: before merging (2,053,543 surfels); Right: after merging (226,664 surfels). Search radius in world space = 1.2 mm. Search radius in image space = 1 pixel, in order to remove outliers.*

```
maps[mapCnt][width][height]: all depth maps (each of size
                             width x height)
mapDone[mapCnt][width][height]: init to false

for (map=0; map<mapCnt; map++) {
 for (y=0; y<height; y++) {
  for (x=0; x<width; x++) {
   if (!mapDone[map][x][y]) {
    mapDone[map][x][y] = true;
    for (otherM=map+1; otherM<mapCnt; otherM++) {
     - convert the point maps[map][x][y] to coordinate
       system of maps[otherM]
     - find nearest neighbor inside maps[otherM]:
       test only points where mapDone is false
     - if found: collect point
    }
    - collected points: candidates for best point
    - set candidate's mapDone's to true
    - decide best point, delete all other candidates
   }
  }
 }
}
```

Figure 5: *Best Surfels algorithm (greedy version). The depth maps hold the reconstructed points.*

In order to remove outliers (additionally to the methods described in Section 4.2), we propose a modification to the algorithm. This is achieved by modification of the nearest neighbor search presented in Section 5.1. Instead of computing the search range in images space r'_i using projection, it is set by the user to a range that is smaller. This way, the region where best surfel candidates are sampled is not spherical anymore (as would result from true 3D distance test), instead the search range is restricted in directions perpendicular to the views. This addresses outliers because it allows to choose a larger search radius that captures the outliers without thinning out too many neighboring surfels. Figure 6 shows an example result.

6. Texture Reconstruction

Most materials exhibit a certain amount of specular reflection. In the case of strong specular reflection, the effects are confined to a small region on the object for a given view and illumination. Since we acquire many overlapping reconstructions, one could remove the specular reflecting parts in all of them before computing the best surfels. Theoretically this would lead to consistent diffuse reconstructions of specular objects. However, we would lose the specular information. Additionally, most materials exhibit glossy reflection which leads to specular effects that cover large parts of the object. There it would not be possible to remove them without producing holes. Consequently, a specular reflectance model has to be fitted in these cases.

The diffuse reflection model can however be used to generate high-resolution reconstructions of objects that do not exhibit too strong specular reflection. The surfel reconstruction preserves the sub-pixel information discretized by the camera if the surfels are located on the viewing rays of the camera pixels and if each surfel gets its uniform color only from the corresponding pixel. This can be achieved for the

diffuse reflection model, because it can be robustly fitted to a single reflectance sample.

6.1. Shadow Removal

The scanner can generally not acquire shadowed object regions, because the structured light does not reach these parts. However, the reconstruction still may yield points in shadows, if they are indirectly illuminated by inter-reflection, sub-surface scattering and other effects. It is important to detect and remove shadows when computing texture. Furthermore, structured light reconstruction of indirectly illuminated regions and at shadow boundaries leads to geometric artifacts (see Figure 7 left). The explanation for the deformed boundaries is the same as the motivation for detecting sub-pixel variance in Section 4.2 and for avoiding Bayer tiling.

The shadows are detected using a depth-test relative to the projector. For each position of the turntable we get a reconstruction with corresponding viewpoint and projector position. After the best surfels have been determined, they are projected into each virtual view of the projector. Projector pixels that get more than one surfel projection are tested for shadow. If the surfel producing the projection belongs to the same depth map as the virtual view and if it is not nearest to the camera, it is removed given that the surfel is farther away from the nearest surfel than a user-defined threshold. The generated holes are filled with other surfels by recomputation of the best surfels.

Although some shadow surfels are eliminated by the described method, there are usually still shadow surfels remaining due to calibration errors, reconstruction errors, and the fact that the sampled surface elements have an extent. Therefore, instead of projecting each surfel to the projector and storing it only in the corresponding projector pixel, it is stored to a user-defined range around that projector pixel. This way the sensitivity for depth discontinuities is increased and at the same time, surfels that have been reconstructed at grazing viewing angles are removed.

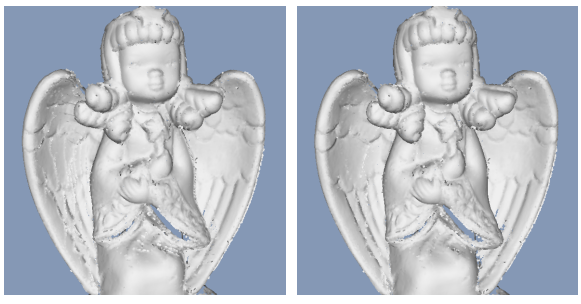

Figure 7: *Removal of shadows (seam artifacts). Left: no removal. Right: local range = 1 and minimal distance = 2mm (*18,452 *surfels removed).*

6.2. Reflectance Sampling

In order to compute the texture, we need to collect for each surface point all available and reliable reflectance samples from the acquired texture views. Reflectance is computed from the image intensities using the calibrated illumination model described in Section 3.6. According to [LKG*01], we call a surface point together with the corresponding samples a lumitexel. Not all views contribute a sample for a given surface point. There might be effects like occlusion and insufficient illumination or shadows that invalidate a sample in a given view. Insufficient illumination is detected using a user-defined threshold that rejects samples illuminated at grazing angles, in addition to the signal strength threshold described in Section 4.2.

We base our tests for occlusion and shadows on the surfel representation. Only surfels that belong to the final merged reconstruction are representing surface points and hence lumitexels. Therefore the points that are removed by the best surfels algorithm are only marked as such and not deleted, since they are needed here for sample selection.

Occlusion/Shadow test: The structured light method already performs a kind of implicit occlusion test in the sense that it can only acquire surface points which are visible by the camera. The same holds for the light source, so theoretically no shadows can be acquired. Therefore, we can decide if a point $\mathbf{p}$ gets occluded in a given view v_i by projecting $\mathbf{p}$ to that view (onto the camera pixel $\mathbf{p}'$) and by observing the surfel $\mathbf{s}_{pi}$ that has been reconstructed by the pixel $\mathbf{p}'$ of view v_i. Possible cases are:

- There has been no $\mathbf{s}_{pi}$ reconstructed: This means that reconstruction failed for that pixel or that it has been removed by one of the methods presented in Section 4.2. Another possibility is that it has been removed by the shadow removal process from Section 6.1. In any case, we reject the corresponding sample.
- $\mathbf{s}_{pi}$ has been reconstructed (and either selected by best surfels or not): This means either that v_i has an unobstructed view to the point $\mathbf{p}$ or that $\mathbf{s}_{pi}$ represents a surface point different to $\mathbf{p}$. To test if $\mathbf{s}_{pi}$ represents a different point, we measure the distance between $\mathbf{p}$ and $\mathbf{c}_i$ ($\mathbf{c}_i$ is the camera position of v_i) as well as the distance between $\mathbf{s}_{pi}$ and $\mathbf{c}_i$. If the difference $\left|\|\mathbf{p}-\mathbf{c}_i\| - \|\mathbf{s}_{pi}-\mathbf{c}_i\|\right|$ exceeds a user-defined threshold, the points are assumed to be distinct and the sample is rejected due to occlusion.

6.3. Lambertian Texture

We compute diffuse texture from a single reflectance sample according to the diffuse reflection model:

$$\rho_d = \frac{L}{\mathbf{n} \bullet \mathbf{u} \cdot E}, \qquad (2)$$

with, for a given object point, ρ_d the diffuse albedo, L the radiance observed by the camera, E the irradiance from the projector at that point, $\mathbf{n}$ its normalized surface normal, and $\mathbf{u}$ the normalized vector towards the light source. The surface normal is actually computed from the neighbors of the reconstructed point. $\mathbf{u}$ comes from the geometric projector calibration, and E from the luminous projector calibration according to Section 3.6. Figure 8 and Figure 9 (top left) show respective results.

Figure 8: *Single view reconstruction, Lambertian texture.*

6.4. Phong Texture

Alternatively we use the diffuse term together with a specular Phong lobe as reflection model:

$$\frac{L}{E} = \rho_d \cdot \mathbf{n} \bullet \mathbf{u} + \rho_s \cdot (\mathbf{u}^T(2\mathbf{n}\mathbf{n}^T - \mathbf{I})\mathbf{v})^n \,, \qquad (3)$$

with ρ_s the maximum specular albedo, $\mathbf{v}$ the normalized vector towards the camera, and n the specular exponent. Nonlinear least squares fitting of (3) to the reflectance samples is done simultaneously for red, green, and blue ρ_d and ρ_s, but with a single exponent n using the Levenberg-Marquardt method as suggested by [LFT*97]. An initial estimate is computed by linear fitting. This is achieved by fixing n for optimization and by solving the resulting linear least squares problem. This is repeated for exponentially increasing n and each time the residual is computed. The fit leading to the smallest error is chosen as result of the linearized fitting. Linear fitting can also be used as fall-back to nonlinear fitting.

Figure 9 shows a result of the Phong fit. The reconstruction still shows some artifacts in regions of sparse sampling and at points with erroneous normals due to outliers. However, the diffuse part of the specular fit provides a significantly improved texture compared to the Lambertian fit, as the Lambertian fit over-estimates brightness in specular regions.

Further improving the Phong fit is difficult due to the limited number of samples per lumitexel. One way to come around this would be to apply material clustering as presented by [LKG*01].

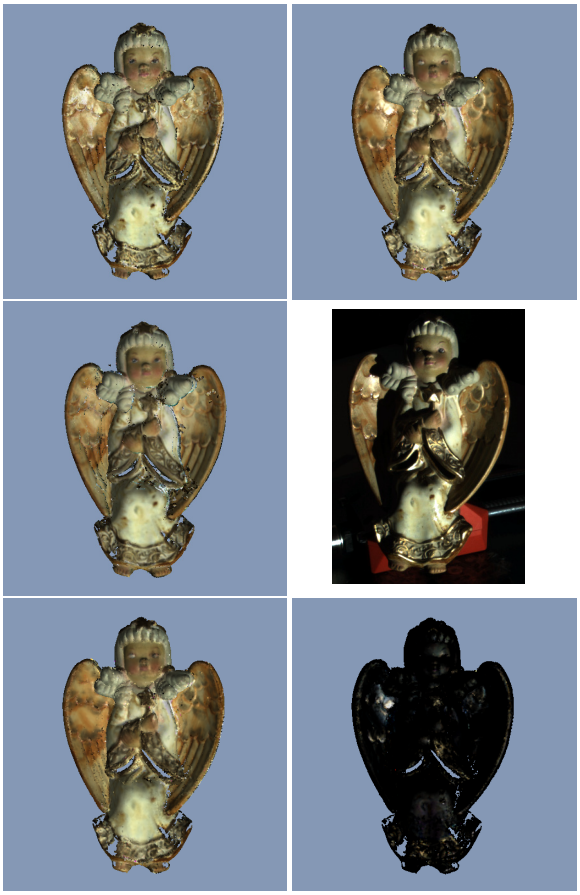

Figure 9: *Reconstructed Lambertian texture (top left), Phong texture (top right), diffuse component of Phong texture (bottom left), specular component (bottom right), blended overlapping reconstructions (center left) and original camera view (center right).*

7. Results

Figure 10 shows a single-ring reconstruction of a clay pot acquired from only 15 views. Figure 11 shows a telephone also reconstructed from 15 rotated views using the same setup. It can be seen that the clay pot produces better quality regarding texture and geometry because its material is mostly diffuse, in contrast to plastic material of the telephone.

Because we have chosen objects containing both, diffuse and specular materials and a lot of occlusion, the achieved results still contain artifacts such as outliers, false normals and holes. These are addressed by the post-processing stage as described by [WPK*04], although in our case, most of the holes originate from suboptimal setup and could be addressed by interactive control of the turntable and more appropriate tilting of the object in order to generate optimal views.

Figure 10: *Clay pot reconstructed from 15 rotary views (384492 surfels). Lambertian texture (left) and synthetic Phong texture (right) for showing geometric detail. The Lambertian texture exhibits stripe artifacts due to specular reflection of the clay surface.*

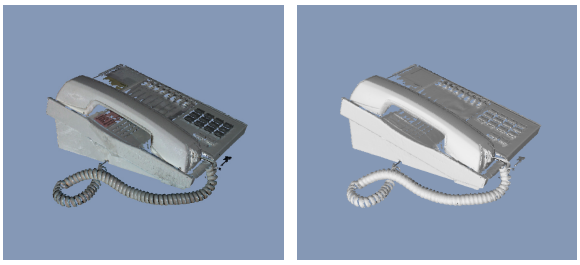

Figure 11: *Telephone reconstructed from 15 rotary views (317681 surfels). Lambertian texture (left) and no texture (right).*

7.1. Reconstruction Accuracy

For evaluation of the reconstruction accuracy, we used a steel sphere of 150 mm in diameter which had been manufactured with a tolerance of ±0.1mm and painted in white color. A single ring of 10 partial reconstructions at a uniform angular step was taken and merged according to Section 5. For the merged reconstruction, a center was computed by fitting a least-squares sphere to the surfels. We obtained a mean distance to this center of 74.947 mm and maximal errors less than 0.5 mm which is roughly the surfel spacing (red histogram in Figure 12a). For further analysis of these errors, we fitted a sphere of the same radius to each partial reconstruction. The resulting centers (Figure 12b) reveal a small systematic error in the turntable calibration, plus a (probably) mechanical effect happening between the first two scans. The blue histograms show the higher accuracy of each partial reconstruction. The green histogram is obtained by translating the partial reconstructions to a common center. This step cannot be done for general objects, it just illustrates that there is some potential left in the merging process. However, registration of surfel objects based on geometry or texture to extreme sub-surfel precision is difficult.

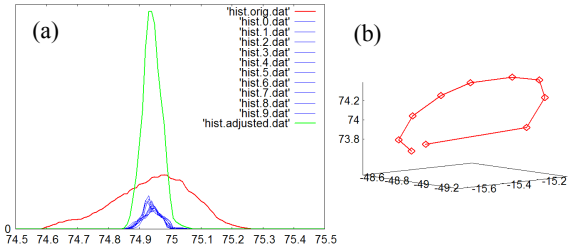

Figure 12: *(a) Histograms of surfel distances to sphere center (red: full reconstruction, blue: partial reconstruction, green: partial reconstructions merged by optimal translations). (b) Centers of LS fits for partial reconstructions. All units given in millimeters.*

7.2. Performance

The angel figurine of 180 mm height was reconstructed from 3x30 views. Each view contributed in average 28,082 surfels, leading to a raw reconstruction consisting of 2,527,362 surfels. 473,819 erroneous surfels were removed by the ray skew criterion and removed because they formed too small connected components (outliers). Another 1,826,879 surfels were discarded during merging. Finally, 18,452 surfels got discarded by the shadow removal process. Removing the shadows is the computationally most expensive step, it took 38 minutes. Texture reconstruction took 16 minutes and computation of the normals another 6 minutes. The best surfels merging procedure took again 6 minutes and the cost of remaining operations is negligible.

8. Conclusion

We proposed a method for calibration of the turntable axis, including a method for robust detection of regular calibration patterns. We extended the calibrated projector to a calibrated light source. Methods for robust reconstruction and removal of artifacts have been proposed as well as methods for merging multiple point-based reconstructions into a single-layered representation. We also contributed methods for fitting the Phong model to reflectance samples of single surface points. As future work, we plan to improve the best surfels algorithm from its currently greedy version to global optimization. Additionally to material clustering we also plan to fit the normals to the reflectance samples in order to improve small-scale normal variations.

References

[BMS98] BATLLE, J., MOUADDIB, E., SALVI, J.: Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. In *Pattern Recognition* 31, 7 (1998), 963–982.

[BMR01] BERNARDINI, F., MARTIN, I.M., RUSHMEIER, H.: High-quality texture reconstruction from multiple scans, *IEEE Transactions on Visualization and Computer Graphics* 7, 4 (2001), 318-332.

[BM92] BESL, P., MCKAY, N.: A Method for Registration of 3-D Shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.

[CL95]CURLESS, B., LEVOY, M.: Better optical triangulation through spacetime analysis. In *ICCV* (1995), 987–994.

[DRR03] DAVIS, J., RAMAMOORTHI, R., RUSINKIEWICZ, S.: Spacetime stereo: A unifying framework for depth from triangulation. In *CVPR'03* (June 2003), 359– 366.

[Dye01] DYER, C.R.: Volumetric Scene Reconstruction from Multiple Views. In L.S. Davis, editor, *Foundations of Image Understanding* (Kluwer, Boston, 2001) 469-489.

[ES03] ESTEBAN, C.H., SCHMITT, F.: Silhouette and stereo fusion for 3D object modeling. In *3DIM03* (2003) 46-53.

[Güh01] GÜHRING, J.: Reliable 3D surface acquisition, registration and validation using statistical error models. In *3DIM* (2001) 224-231.

[GBB*00]GÜHRING, J., BRENNER, C., BÖHM, J., FRITSCH, D.: Data processing and calibration of a cross-pattern stripe projector. In *IAPRS* 33, 5 (2000).

[HS97]HARTLEY, R., STURM, R.: Triangulation. In *Computer Vision and Image Understanding* 68, 2 (1997) 146-157.

[KS00] KUTULAKOS, K., SEITZ, S.: A theory of shape by space carving. *International Journal of Computer Vision* 38, 3 (2000), 197-216.

[LFT*97] LAFORTUNE, E.P., FOO, S.C., TORRANCE, K.E., GREENBERG, D.P.: Non-linear approximation of reflectance functions. In *Proc. SIGGRAPH '97* (August 1997), 117-126.

[LBJ04] LEGARDA-SAENZ, R., BOTHE, T., JUEPTNER, W.: Accurate procedure for the calibration of a structured light system. *Optical Engineering* 43, 2, (February 2004), 464-471.

[LKG*01] LENSCH, H.P., KAUTZ, J., GOESELE, M., HEIDRICH, W., SEIDEL, H.P.: Image-based reconstruction of spatially varying materials. In *Eurographics Workshop on Rendering (2001), 104–115.*

[MPN*02]MATUSIK, W., PFISTER, H., NGAN, A., BEARDSLEY, P., ZIEGLER, R., MCMILLAN, L.: Image-Based 3D Photography using Opacity Hulls. In *Proc. SIGGRAPH* (2002), 427-437.

[MR02]MARSCHNER, S., RAMAMOORTHI, R.: Acquiring Material Models Using Inverse Rendering. *Course notes for SIGGRAPH 2002 Course 39.*

[PZB*00] PFISTER, H., ZWICKER, M., VAN BAAR, J, GROSS, M.H., Surfels: Surface Elements as Rendering Primitives. *Computer Graphics,* In *Proceedings of SIGGRAPH 2000* (New Orleans, Louisiana, USA, July 23-28, 2000), 335-342.

[PAD*98] PULLI, K., ABI-RACHED, H., DUCHAMP, T., SHAPIRO, L.G., STUETZLE, W.: Acquisition and visualization of colored 3D objects. In *Proceedings of Fourteenth International Conference on Pattern Recognition* 1 (1998), 11-15.

[PDH*97] PULLI, K., DUCHAMP, T., HOPPE, H., MCDONALD, J., SHAPIRO, L., STUETZLE, W.: Robust meshes from multiple range maps. In *Proceedings of International Conference on Recent Advances in 3-D Digital Imaging and Modeling* (1997) 205-211.

[Pul99] PULLI, K.: Multiview registration for large data sets. In *Proceedings Second International Conference on 3-D Digital Imaging and Modeling* (1999), 160-168.

[RHL02] RUSINKIEWICZ, S., HALL-HOLT, O., LEVOY, M.: Real-Time 3D Model Acquisition. In *ACM Transactions on Graphics* 21, 3 (2002), 438-446.

[SPM04] SAINZ, M., PAJAROLA, R., MERCADE, A.: A Simple Approach for Point-Based Object Capturing and Rendering. *IEEE Journal of Computer Graphics and Applications (CG&A)* 24, 4 (2004), 24-33.

[SPB04] SALVI, J., PAGES, J., BATLLE, J.: Pattern codification strategies in structured light systems. *Pattern Recognition* 37, 4 (2004), 827-849.

[SM02] SHEN, T. S., MENG, C. H.: Digital projector calibration for 3-d active vision systems. *Journal of Manufacturing Science and Engineering* 124, 1 (February 2002), 126-134.

[SWI97]SATO, Y., WHEELER, M. D., IKEUCHI, K.: Object shape and reflectance modeling from observation. In *Proceedings of the 24th annual conference on computer graphics and interactive techniques* (August 1997), 379-387.

[TL94]TURK, G., LEVOY, M.: Zippered Polygon Meshes from Range Images. In *Proc. of SIGGRAPH* (1994), 311-318.

[WPK*04] WEYRICH, T., PAULY, M., KEISER, R., HEINZLE, S., SCANDELLA, S., GROSS, M.: Post-processing of scanned 3d surface data. In *Proceedings of Eurographics Symposium on Point-Based Graphics 2004* (2004), 85-94.

[Zha00] ZHANG, Z.: A flexible new technique for camera calibration. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (2000), 1330-1334.

[ZPK*02] ZWICKER, M., PAULY, M., KNOLL, O., GROSS, M.: Pointshop 3d: An interactive system for point-based surface editing. In *Proc. of SIGGRAPH 2002* (San Antonio, TX, July 2002), 322-329.