Photon Mapping based Simulation of Multi-Path Reflection Artifacts in Time-of-Flight Sensors

Stephan Meister
Heidelberg Collaboratory for Image Processing
Institution1 address
stephan.meister@iwr.uni-heidelberg.de

Rahul Nair
rahul.nair@iwr.uni-heidelberg.de

Bernd Jähne bernd.jaehne@iwr.uni-heidelberg.de

Daniel Kondermann
daniel.kondermann@iwr.uni-heidelberg.de

Abstract

Time-of-Flight cameras which use specialized sensors and modulated infrared light are able to measure distances as well as intensity images. The resulting depth maps show various types of errors such as flying pixels, noise and incorrect depth. In this paper we present a novel method to simulate time-of-flight data, based on modern computer graphics. We modify the photon mapping global illumination algorithm to take the time-dependent propagation of modulated light in a scene into consideration. This allows us to correctly simulate the depth output of a time-of-flight camera given scene geometry and material properties. So far camera simulators have only been able to provide a physically correct simulation of the sensor electronics and the image creation process. The phenomenon of multi-reflection or multipath interference could not be fully simulated yet. These effects are caused by indirect light paths between camera and lightsource and are therefore dependent on scene geometry. Our method enables us to create ground truth for evaluation, denoising and analysis of such effects.

1. Introduction

Camera systems which are able to measure depths based on stereo vision, structured light or time-of-flight principles have recently attracted a lot of attention from researchers as well as practitioners. Time-of-flight cameras have many beneficial properties such as high frame-rates and robustness. Hence, they are interesting for various applications such as in outdoor environments. Depth maps created with these cameras show a variety of systematic and statistical errors such as depth noise, flying pixels or multipath interference.

Testing and validation of algorithms that use suboptimal depth maps as input is usually limited to images taken in controlled lab conditions, as only here sufficiently correct ground truth can be acquired. We believe that the capability to describe all time-of-flight errors and simulating them is a first step towards creating methods that are capable of overcoming these problems. Recent research by Wulff et al. [26] or Meister and Kondermann [15] suggest that computer generated images are in many cases suitable alternatives for algorithm validation. So far there are no methods which can simulate all time-of-flight related errors simultaneously.

We focus on the simulation of multipath interference, which has not yet been addressed in literature. These occur because the modulated light used by the camera is reflected from multiple surfaces inside the scene before reaching the camera sensor. (See Section 3.1) The problem of correctly simulating light propagation in a scene is well known in the computer graphics community. There, global illumination algorithms are used to simulate both direct and indirect light
as well as material properties like reflection, transmission, and others. These methods only assume that light propagation is instantaneous and each frame is in a steady state.

In this paper we propose a new method for simulating the response of time-of-flight cameras by using a modified photon mapping global illumination method, which also considers a finite speed of light.

2. Related Work

Multiple methods for the classification, measurement and removal of errors inherent in time-of-flight images have been described, for example by Plaue [18], Schmidt [22], Lindner and Kolb [14] or Falie and Buzuloiu [4].

The estimation and compensation of multipath interference in time-of-flight cameras has been the focus of recent investigation e.g. by Fuchs [5], Jiménez et al [8] or Dorrington et al. [2]. Most of the described methods assume that the involved materials have lambertian reflectance properties or take only simple one-time reflections into consideration.

A method for simulating the sensor of time-of-flight cameras has been described by Schmidt [21].

In contrast we simulate the light propagation in a scene. In computer graphics Smit et al.[24] where the first to postulate a generic time-dependent form of the rendering equation. Regarding time-of-flight cameras, this has only been performed by Keller, Kolb et al. [11] [10]. In their work they presented an extensive framework for time-of-flight simulation based on scanline-rendering.

Our proposed method enhances [10] by simulating multipath interference via photon mapping.

3. Methodology

In the following two sections, we will summarize the working principles of time-of-flight cameras as well as the photon mapping algorithm. Then, we will describe our modifications to the latter.

3.1. Time-of-Flight Cameras

Continuous-wave time-of-flight depth cameras use a variation of the following principle: The intensity of an active light source is modulated with a Megahertz-range frequency. The image sensor correlates the input signal created by the reflected light with the modulation reference signal to calculate the phase shift $\phi$. This phase-shift is proportional to the traveled light distance modulo the ambiguity range ($7.5 m$ for a modulation frequency of $20 MHz$) due to the periodicity of the modulation signal.

At least 4 raw frames ($I_{0-3}$) measured at 4 different phases are necessary to compute the depth.

For phase shifts $\lambda$ of $\frac{1}{2}\pi$, $\pi$ and $\frac{3}{2}\pi$, a modulation frequency $f$ and light speed $c$ the distance $d$ of an object to the camera can be computed as:

$$d \propto \tan\left(\frac{I_3 - I_1}{I_0 - I_2}\right) \cdot \frac{c}{f}$$

Different sensor models may capture more raw frames to compensate for noise or background illumination. More details on the CamCube sensor and a detailed error analysis can be found in [18], [22] and [23].

There are multiple sources of errors which cause the result of this equation to differ from the true object-to-camera distance. Apart from different random noise sources (electronics, photon noise, etc.) the most prevalent error is multipath or interreflection. It results in flying pixels at object corners or overestimated depths. Multipath error is caused by light which did not travel a direct path from the light into the scene and back to the camera but was reflected multiple times. In a computer graphics context such effects could be summarized as global illumination. Due to these errors the sensor reports a depth value which is typically higher than the true depth as well as dependent on intensity. Additional errors e.g. the wiggling error, which arise from the fact that the modulation signal is not perfectly sinusoidal can be removed by calibration. See the works by Rapp [19] or Erz and Jähne [3]. Similar depth image computations can also be made for pulse-based systems [13].

3.2. Photon Mapping Revisited

Photon mapping as introduced by Jensen [7] solves the global illumination problem of the rendering equation, which was first described in detail by Immel et al. [6] and Kajiya [9]. See Figure 1 for a visualization of the working principle. The problem is solved in two steps. First, the radiance of a scene is calculated by emitting virtual particles (photons) from each light source into the scene. Each time the photon intersects the scene geometry its position is recorded and from there it can be further reflected or transmitted according to the local material properties. This is done until a sufficient number of photons are deposited in the scene. Depending on the type of material reflectivity, the photons can be deposited in different maps to correctly simulate caustics or specular highlights. Figure 2 shows a sparse photonmap (with a low number of deposited photons) of the virtual box scene in Section 4.2.

In a second step, view rays are projected from the camera similar to classical raytracing. Around each scene-ray-intersection the radiance is calculated by accumulating photons in the vicinity and other light contributions like mirrored reflections.

3.3. Phase Modulated Photon Mapping

Our method modifies the photon emitting process, as well as the illumination integration in phase two of the algorithm.
Usually, for each intersection or photon event the (multi-spectral) intensity and position of the photon is saved in the photon map. In addition, we also save the traveled distance of each photon. Photons are placed until an appropriate termination criterion such as the number of deployed photons or reflection depth is met. The resulting photon maps represent the radiance in the scene (through photon density and individual color-energy per photon) as well as a light travel distance profile.

In the second step the individual light contributions for each pixel are accumulated to a final intensity $L_r$: Direct lighting (a light-scene-camera path) which typically contributes the largest amount of light, indirect lighting due to specular reflection (computed recursively via classical ray-tracing) and indirect lighting due to diffuse reflection. Indirect diffuse lighting as well as caustics are handled by integrating photons in an area around each intersection point. For each contribution $L_i$ the light travel distance $d$ is known, either through direct computation for direct light or from the data in the photon map for indirect light. This allows us to calculate the modulation factor $m_i$ for each contribution:

$$m_i = \left(0.5 \cos \left(\frac{d f \pi}{c} + \lambda\right) + 0.5\right) \cdot I_m$$

- $I_m$ unmodulated Intensity
- $d$ distance
- $f$ modulation frequency
- $c$ speed of light
- $\lambda$ phase shift between modulation and sensor signal
- $I_m = [0..1]$ modulation Intensity
- $O_m = [0..1]$ modulation Offset with $O_m + I_m = 1$

The final intensity $L_r$ observed in a pixel is then:

$$L_r = \sum_i L_i \cdot (m_i + O_m)$$

where the sum is performed over all direct and indirect contributions.

Usually, for each pixel multiple paths and light contributions are accumulated and the final response for a pixel is computed by averaging the contributions. This also involves the use of a spatial filter which can mimic the point-spread-function of a real camera system. This is similar to a supersampling approach and allows us to correctly simulate flying pixels due to mixed depth cues at a single sensor pixel. Currently, we assume a perfect sinusoidal reference signal but simulations using other waveforms could be implemented easily.

Like in most global illumination algorithms, it is possible to shift priority between speed of execution, memory consumption and physical realism. For correct time-of-flight simulation some algorithm parameters must be chosen than for pure artistic renderings. Most important is the number of photons to integrate over in the vicinity of each view-ray intersection point. Typically, a few dozen photons in a rather large area (projection size of several pixels) are sufficient for optically convincing render results. For time-of-flight simulation one must perform a statistically sufficient sampling not only over the light contributions at each point but also over the various photon paths. In other words, the sampling space is now two-dimensional and the number of photon contributions must be increased. Experiments suggest that a few thousand are sufficient. Additionally, the photon integration area must be kept small as otherwise the distance between the intersection point and the contributing photon could introduce an error in the traveled light distance. Both these facts make it necessary to compute sufficiently dense photon maps where the number of photons should be at least one order of magnitude higher than in the default parametrization (where 1 Million photons are usually enough even for complex scenes). This increased sampling must also be applied to the raytracing step.

Photon mapping is a statistical process and will converge to a correct solution under the assumption that the number of photons is sufficient. The algorithm itself is biased, which means that averaging multiple renderings with few samples will not necessarily generate a correct solution of the rendering equation. As each of the phase images is rendered independently, the statistical image noise can influence the produced depth maps significantly. Usually, much
like photon samples, a few hundred raytracing samples per pixel are enough for visually convincing results, but in our case a few thousands samples are necessary to create accurate depth maps. Caching of intermediate results and direct computation of all required phase shifts could reduce this problem in future versions of the algorithm.

Our modification of the photon mapping algorithm is based on the Exphotonmap Integrator (a general implementation of the photon mapping algorithm) of the open source LuxRender project [12], which itself is based on the pbrt render engine by Pharr and Humphrey [17].

Render time for a single phase image depends heavily on the used material shaders and scene geometry. The simulation of a single 200 by 200 pixel depth image with the yet unoptimized program can take about two hours on a 2.4 Ghz Xeon processor. The generated photon maps can be saved between different phase renderings which reduces computation time for all images but the first. The advantage of individual renderings is that motion blur can be simulated accurately by animating camera or scene motions for each phase image.

4. Experiments

We created two real scene setups as well as corresponding synthetic setups to test the algorithm. In this work we focus on the simulation of the PMDTec CamCube 3 sensor, although the principle can easily be applied to other cameras as well. To evaluate our results we compared the generated depth maps with results from a real PMDtec CamCube 3, the simulation method described by Schmidt [21], the simulator by Keller et al. [10], as well as ground truth data. The comparison with ground truth is important as it allows us a separate quantitative statistical and systematic error analysis of all involved imaging systems (synthetic and real).

Ground truth was provided by using objects with known geometry and 3d meshes whose position relative to the camera was estimated using manually annotated 2d-to-3d correspondences. The 2d-to-3d pose estimation and camera calibration was performed with an Levenberg-Marquardt optimization from the OpenCV image processing library [1]. The meshes have an accuracy of $\approx 1\text{mm}$ and reprojection errors of the internal and external camera calibration were below $\approx 0.5\text{pixel}$. Hence we assume that the error in the ground truth depth maps is lower than $\approx 1\text{mm}$ which is well below the standard error ranges of most time-of-flight cameras. All provided depth values are given as radial distance to the camera principle point. We also measured the lens distortions of the CamCube 3 using standard camera calibration techniques with a polynomial lens distortion model and rectified all used images.

For most time-of-flight cameras the distance between lightsource and camera optics is small but non-zero, so we separated our virtual camera and lights by a few centime-

Figure 4. Corner scene: Real time-of-flight intensity image (left), simulated phase images with $0, 0.5\pi, \pi$ and $1.5\pi$ phase shift (right). The depth change is only a few centimeters so the illumination change is only marginal.

Figure 5. Corner scene depth maps: ground truth(top left), real time-of-flight image(top right), Schmidt’s simulation(bottom left), our simulation(bottom right). All colormaps are scaled equally. The real and simulated values were shifted to the same mean depth (differences between Schmidt’s and Keller’s method were minimal). Our simulation is very close to the real cameras depth profile.

ters. This has the effect that the direct illumination typically dominates over other light contributions as there are few shadowed regions. Additionally, the LuxRender engine allows us to use more realistic area light sources instead of point lights which are used in other simulations. We did not consider additional light sources in the scene as their non-modulated light contributions can be rendered separately and added as constant offset to all phase images.

4.1. Corner Scene

As a first test environment we choose a simple scene with two perpendicular walls placed about 50 cm away from the
Figure 6. Depth profile of corner scene through the middle line of the sensor. All values are radial distances to the camera principal point. Our method correctly simulated the rounded corner caused by interreflection.

Figure 7. Relative depth error distributions of corner scene (defined as difference between measured/simulated depth) and ground truth. All axes have same scaling. The error distribution of our simulation resembles the real distribution more closely.

camera. An intensity image as well as four simulated phase images can be seen in Figure 4. Multipath reflections of the modulated light between the walls cause an overestimation in the measured distance as well as a shape change in the corner. Figure 6 shows a cross section of the sensors middle scanline. For our experiment we choose to simulate the shiny coated paper of the target by using a glossy material shader with mostly specular reflection. Details on the exact implementation of the material can be found in the LuxRender documentation. Photon Mapping was active for this simulation but the relative intensity of the direct and indirect specular illumination was about 3 orders of magnitude higher than the photon-map based indirect diffuse illumination, so the influence on the depth measurement is negligible. This allows us to evaluate the raytracing part of the illumination simulation.

Figure 5 shows that the simulated depth response is similar to the result of a real time-of-flight camera. For visualization purposes we shifted the depth values of our simulation to the same mean as the real time-of-flight image. The methods by Schmidt and Keller show similar behavior but due to missing multipath simulation the corner retains its sharpness which is less realistic.

Figure 7 shows the relative error distributions of the real time-of-flight depth values and all three simulations. We also computed the spearman correlation between the simulated depths and the real time-of-flight signal to clarify statistical similarities. See Table 1 for the correlation matrix.

The results indicate that even without a sophisticated sensor electronic simulation we are able to get a more accurate approximate reproduction of the depth bias than previous methods.

4.2. Test Box Scene

The second test scene consists of a wooden box containing multiple materials with different reflection properties like styrofoam or printed paper. The inked paper (Regions A,B and C in Figure 8) and the coated styrofoam are simulated with a regular glossy material shader and an additional low intensity bump map for additional surface detail. Experience shows that a lambertian reflectance model is actually not a very good approximation for unfinished wood, so we used a diffuse Oren-Nayar shader [16] with a high surface roughness in the used microfacet model. To separate the effects of indirect specular and indirect diffuse reflection we rendered two distinct simulations. The first one is a regular rendering with direct illumination. To reassess the correct behavior of the photon based illumination, we performed a second simulation, where we used only indirect illumination and changed the parameters in such a way that the photon map is the only light contribution.

Figure 9 shows a closeup of the different depth maps for the first simulation as a surface plot. The different effects the materials have on the measured depth are clearly visible: The black inked paper on the stairs (Point B in Figure 8) and slope (Point A) for example shows more distinct light reflections than the white paper (Point C). In these

<table>
<thead>
<tr>
<th></th>
<th>Real</th>
<th>Schmidt</th>
<th>Keller</th>
<th>Our</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>1.0000</td>
<td>−0.4576</td>
<td>0.5600</td>
<td>0.6301</td>
</tr>
<tr>
<td>Schmidt</td>
<td>−0.4576</td>
<td>1.0000</td>
<td>−0.1162</td>
<td>−0.2342</td>
</tr>
<tr>
<td>Keller</td>
<td>0.5600</td>
<td>−0.1162</td>
<td>1.0000</td>
<td>0.7246</td>
</tr>
<tr>
<td>Our</td>
<td>0.6301</td>
<td>−0.2342</td>
<td>0.7246</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Table 1. Spearman correlation matrix for real time-of-flight data, Schmidt’s method, Keller’s method and our simulation. Higher values denote better correlations.
regions the real time-of-flight camera measures a slightly higher depth. Our simulation shows the same effect as we have chosen higher reflectance values for the darker parts of the paper. In regions with sharp depth discontinuities flying pixels with depth values between the foreground and background are visible. Examples can be seen on the sides of the stairs or the slope (Area E in Figure 10). Our simulation shows a good match of these artifacts as well.

Figure 10 displays a cross section of the bottom of the box (Region D along the red line in Figure 8). The "bulge" in the real time-of-flight data is caused by the high reflectivity of the used wood under high angles of deflection. Most synthetic materials used in computer graphics appear darker under low angles of deflection. This is not a problem of the simulation but of the simplified material models. Therefore, future work should focus on using measured bidirectional brightness distribution functions (BRDFs).

We managed to correctly simulate the shallow rise in depth for pixels on the side wall (pixel values above 15) but not the "bulge". Without interreflection and only direct lighting the simulated depth is much closer to the ground truth, which is not a realistic behavior.

5. Conclusion

Based on a modification of the photon mapping algorithm, in this paper we solve the problem of physically correct multipath interference for phase modulated time-of-flight camera simulation. We have shown that the method can correctly simulate flying pixels, intensity dependent depth and depth errors due to multiple interreflections (multipath) based on the scene geometry. Furthermore, it is able to simulate time-of-flight depth distortions caused by different materials in the scene. Our simulation is so far the only one which accounts for all these effects. By taking into consideration the physical and geometric setup of a scene we can create more realistic images and depth maps for statistical analysis, evaluation, denoising and test of time-of-flight cameras. This enables researchers and practitioners to create synthetic test datasets for various environmental conditions, scenes or resolutions without the need of a real camera setup.

Future work will concentrate on optimizing the computational efficiency of the rendering process. An option would be to create all phase images simultaneously instead of us-
ing four distinct renderings at the cost of simulating motion artifacts. The described method of tracing the depth the light has traveled could also be applied to other global illumination algorithms such as metropolis light transport by Veach and Guibas [25] which are unbiased and make faster simulations of more complex setups possible.

Some of our experiments suggest that the results can be improved significantly when measured instead of modeled BRDFs are used to represent the materials in the scenes. Currently, we are working on custom material shaders which can use measured or otherwise reconstructed BRDFs, e.g. as described by Ruiters and Klein [20].

Additionally, a combination with other algorithms which simulate lenses or electronic characteristics of the sensor are also possible.

The modified LuxRender code will be made available on our website according to the GNU General Public License.

References