Fast back-projection for non-line of sight reconstruction

Victor Arellano
Universidad de Zaragoza , I3A

Diego Gutierrez
Universidad de Zaragoza , I3A

Adrian Jarabo
Universidad de Zaragoza , I3A

ABSTRACT

Recent works have demonstrated non-line of sight (NLOS) reconstruction by using the time-resolved signal from multiply scattered light. These works combine ultrafast imaging systems with computation, which back-projects the recorded space-time signal to build a probabilistic map of the hidden geometry. Unfortunately, this computation is slow, becoming a bottleneck as the imaging technology improves. In this work, we propose a new back-projection technique for NLOS reconstruction, which is up to *a thousand times* faster than previous work, with negligible quality loss.

CCS CONCEPTS

•**Computing methodologies** → **Reconstruction;** *Object recognition; Image processing; Volumetric models;*

KEYWORDS

Computational imaging, transient imaging, NLOS, reconstruction

ACM Reference format:
Victor Arellano, Diego Gutierrez, and Adrian Jarabo. 2017. Fast back-projection for non-line of sight reconstruction. In *Proceedings of SIGGRAPH '17 Posters, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: 10.1145/3102163.3102241

1 INTRODUCTION

One of the core applications of time-resolved imaging is the capability to robustly capture depth from a scene, by being able to track the time of arrival of photons. In the last years the applicability of time-resolved imaging has gone beyond directly visible geometry, to include non-line of sight (NLOS) imaging [Velten et al. 2012]. This technique allows reconstructing occluded objects by analyzing multiple-scattered light, even in the presence of turbid media.

Despite the recent advances in capture quality, the *reconstruction* step is still a bottleneck, limiting the applicability of this technology in the wild. The most widely used approach for reconstruction is back-projecting the space-time captured image on a voxelized geometry representation [Gupta et al. 2012; Velten et al. 2012]. However, the large amount of data being processed, together with the complexity of the reconstruction algorithms, impose computation times in the order of several hours.

In this work [Arellano et al. 2017] we propose a new back-projection reconstruction method that yields a speed-up factor of three orders of magnitude over previous NLOS reconstruction approaches, thus addressing the main pending issue limiting the applicability of recent approaches. Our techniques allows computing the probability maps of large datasets in the order of seconds with a minimum increase in error, while on the other hand allows significantly higher-resolution reconstructions with a negligible added cost.

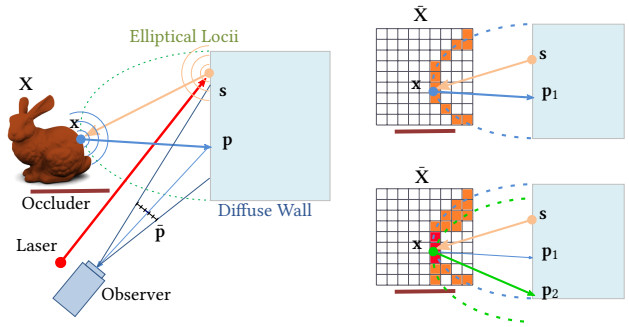

Figure 1: Overview of our method. Left: Our reconstruction setup. A laser pulse is emitted towards a diffuse wall, creating a virtual point light s illuminating the occluded scene. The reflection of the occluded geometry travels back to the wall, which is imaged by the camera. The total propagation time from a hidden surface point x forms an ellipsoid with focal points at s and p. Right: The intersection of several of these ellipsoids defines a probability map for the occluded geometry.

2 NLOS BACK-PROJECTION

The goal of NLOS reconstruction methods is to recover an unknown hidden scene $\mathbf{X} \in \mathbb{R}^3$ from the measurements on visible known geometry $\mathbf{P} \in \mathbb{R}^3$. Such $\mathbf{P}$ is defined by a bijective map $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ from pixel values $\bar{\mathbf{P}}$ imaged on the sensor. The scene is illuminated by a set of laser shots directed towards the visible geometry, hitting at positions $\mathbf{S}$; an image is captured for each shot. Figure 1 shows an example of our setup, where $\mathbf{s} \in \mathbf{S}$ is a virtual point light source created by the laser's reflection, and $\mathbf{p} \in \mathbf{P}$ is the projection of pixel $\bar{\mathbf{p}}$ in the known geometry $\mathbf{P}$ as $\psi(\bar{\mathbf{p}}) = \mathbf{p}$, with inverse mapping $\psi^{-1}(\mathbf{p}) = \bar{\mathbf{p}}$. It can be seen how the total propagation time from a hidden surface point $\mathbf{x}$ forms an ellipsoid with focal points $\mathbf{s}$ and $\mathbf{p}$.

In our particular context, the captured signal I is a time-resolved image indexed by the spatial and the temporal domains, measuring the light's time of arrival on each pixel

$$I_{\mathbf{s}}(\bar{\mathbf{p}}, t) = \int_{\mathbf{X}} L_o(\mathbf{s} \rightarrow \mathbf{x}, t_o)\, f(\mathbf{s} \rightarrow \mathbf{x} \rightarrow \mathbf{p})\, G(\mathbf{s} \rightarrow \mathbf{x} \rightarrow \mathbf{p}) d\mathbf{x}, \quad (1)$$

Back-projection methods aim to recover a discrete approximation $\bar{\mathbf{X}}$ of the unknown scene $\mathbf{X}$. The main idea is to build a probabilistic model where, assuming Lambertian reflectances, the probability of point $\mathbf{x}$ being part of the occluded geometry is modeled as

$$p(\mathbf{x}) = \sum_{\mathbf{s} \in \mathbf{S}} \sum_{\bar{\mathbf{p}} \in \mathbf{P}} I_{\mathbf{s}}(\bar{\mathbf{p}}, \tau(\mathbf{s} \rightarrow \mathbf{x} \rightarrow \mathbf{p}))\, G(\mathbf{s} \rightarrow \mathbf{x} \rightarrow \mathbf{p})^{-1}, \quad (2)$$

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGGRAPH '17 Posters, Los Angeles, CA, USA
© 2017 Copyright held by the owner/author(s). 978-1-4503-5015-0/17/07…$15.00
DOI: 10.1145/3102163.3102241

This probability map is later used to reconstruct the geometry, by using some operator over the voxelized representation, typically a three-dimensional Laplacian filter [Velten et al. 2012].

In practice, the most common straight forward form of computing the probability map consists of evaluating Eq. (2) for each unknown point $\mathbf{x} \in \bar{\mathbf{X}}$, which is very expensive. In the following, we introduce our novel formulation, which significantly reduces the theoretical complexity of the computations, and allows for a very efficient implementation in commodity hardware.

3 OUR METHOD

Our method builds on the observation that the set of points that can potentially contribute to $I_{\mathbf{s}}(\bar{\mathbf{p}}, t)$ from a given laser shot hitting at $\mathbf{s}$ is defined by the ellipsoid $E(\mathbf{s}, \mathbf{p}, t)$ with focal points at $\mathbf{s}$ and $\mathbf{p}$, and focal distance $t \cdot c$ (see Fig. 1). This means that the more ellipsoids intersecting at point $\mathbf{x}$, the higher the probability $p(\mathbf{x})$ of having occluded geometry at $\mathbf{x}$, since more light arriving at pixel $\bar{\mathbf{p}}$ can have potentially been reflected at $\mathbf{x}$. Following this observation, we pose Eq. (2) as an *intersection of ellipsoids* $E(\mathbf{s}, \mathbf{p}, t)$ with a voxelized representation of the scene, as

$$p(\mathbf{x}) = \sum_{\mathbf{s} \in \mathbf{S}} \sum_{\bar{\mathbf{p}} \in \mathbf{P}} I_{\mathbf{s}}(\bar{\mathbf{p}}, t)\, \text{isect}(\mathbf{x}, E(\mathbf{s}, \mathbf{p}, t)), \tag{3}$$

where $t = \tau(\mathbf{s} \rightarrow \mathbf{x} \rightarrow \mathbf{p})$, and $\text{isect}(\mathbf{x}, E(\mathbf{s}, \mathbf{p}, t))$ is a binary function returning 1 if the ellipsoid $E(\mathbf{s}, \bar{\mathbf{p}}, t)$ intersects voxel $\mathbf{x}$, and 0 otherwise.

The intersection operand allows us to compute $p(\mathbf{x})$ by testing the ellipsoid-voxel intersection directly. Instead of evaluating each voxel $\mathbf{x}$ against the captured data, we now simply project the captured data into the voxelized scene representation.

Posing the problem this way has two main benefits: On the one hand, it significantly reduces the complexity of the required computations by only testing on locations with signal information. On the other hand, our new formulation is equivalent to performing a voxelization of the full set of ellipsoids defined by the combination of tuples $\langle \mathbf{p}, \mathbf{s}, t \rangle$ (Fig. 1, right). Voxelization is a well-studied problem in computer graphics, which can be efficiently performed in modern GPUs [Schwarz and Seidel 2010].

In order to perform the voxelization of the ellipsoids we rely on hardware-based voxelization. To achieve this, we need to overcome two main problems: *i)* we need to create a large number of ellipsoids, which can be very expensive and memory consuming; and *ii)* hardware rasterization does not work with parametric surfaces beyond triangles, so we need to tessellate the ellipsoids before rasterization; this aggravates the cost/memory problem.

We address the first point by taking advantage of instanced rendering, which allows to re-render the same primitive while applying a different linear transformation to each instance. For this, we create a base sphere, which is later transformed in the target ellipsoid i by scaling, translating and rotating it, by using a standard linear sphere-to-ellipsoid transformation matrix $\mathbf{M}_i$.

Before rendering, we apply recursive geodesic tessellation to the base sphere. Ideally, we would like all triangles' sizes to be smaller than the voxel size, so that high curvatures are accurately handled. However, since the transformations required for each ellipsoid might vary, the final size of each rendered triangle is not known in advance; this means that we cannot set a particular tessellation level for all ellipsoids. We instead precompute a set of spheres $\mathbf{O}$ with different tessellation levels o, and dynamically choose what level o will be used as

$$\text{argmin}_{o \in \mathbf{O}} \; (\alpha_o \max(\text{Eig}(\mathbf{M}_i)) < \epsilon), \tag{4}$$

where α_o is the approximation error per triangle at tessellation level o, $\text{Eig}(\mathbf{M}_i)$ are the eigenvalues of transformation matrix $\mathbf{M}_i$, and ϵ is an error threshold, which we set to the voxel size.

4 RESULTS

We evaluate our method by using datasets from two different sources, captured with femto-photography (Fig. 2) and generated with transient rendering (Fig. 3). We compare our method against traditional back-projection [2012], using Velten et al.'s optimized implementation. All our tests have been performed on an Intel i5-6500 @3.2GHz with 8 GB of RAM equipped with a GPU Nvidia GTX 1060.

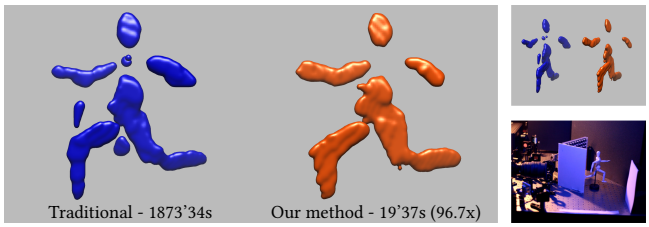

Figure 2: Reconstruction of a mannequin (see bottom right) captured with a streak camera and a femtosecond laser, reconstructed using traditional back-projection (left, in blue) and our method (right, in orange), which is two orders of magnitude faster while yielding similar quality. The inset on the top-right shows the same reconstructed object under a different camera angle (inset from [Velten et al. 2012]).

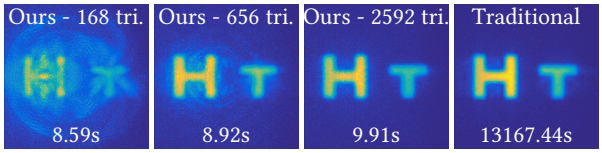

Figure 3: Reconstruction of a synthetic scene [Jarabo et al. 2014] using our method with increasing ellipsoid tessellation quality. The rightmost image shows the reconstruction result using using traditional back-projection. Our final reconstruction is comparable to traditional back-projection, computed with a speed-up of 1300x.

REFERENCES

Victor Arellano, Diego Gutierrez, and Adrian Jarabo. 2017. Fast back-projection for non-line of sight reconstruction. *Opt. Express* 25, 10 (May 2017), 11574–11583.

Otkrist Gupta, Thomas Willwacher, Andreas Velten, Ashok Veeraraghavan, and Ramesh Raskar. 2012. Reconstruction of hidden 3D shapes using diffuse reflections. *Optics express* 20, 17 (2012), 19096–19108.

Adrian Jarabo, Julio Marco, Adolfo Muñoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. 2014. A Framework for Transient Rendering. *ACM Trans. Graph.* 33, 6 (2014).

Michael Schwarz and Hans-Peter Seidel. 2010. Fast parallel surface and solid voxelization on GPUs. *ACM Trans. Graph.* 29, 6 (2010).

Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G. Bawendi, and Ramesh Raskar. 2012. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications* 3 (2012).