Transient Imaging for Real-Time Tracking Around a Corner

Jonathan Klein[1, 2] and Martin Laurenzis[1] and Matthias Hullin[2]

[1]French-German Research Institute Saint-Louis, France
[2]University of Bonn, Institute of Computer Science II, Bonn, Germany

ABSTRACT

Non-Line-of-Sight imaging is a fascinating emerging area of research and expected to have an impact in numerous application fields including civilian and military sensing. Performance of human perception and situational awareness can be extended by the sensing of shapes and movement around a corner in future scenarios.

Rather than seeing through obstacles directly, non-line-of-sight imaging relies on analyzing indirect reflections of light that traveled around the obstacle. In previous work, transient imaging was established as the key mechanic to enable the extraction of useful information from such reflections.

So far, a number of different approaches based on transient imaging have been proposed, with back projection being the most prominent one. Different hardware setups were used for the acquisition of the required data, however all of them have severe drawbacks such as limited image quality, long capture time or very heigh prices. In this paper we propose the analysis of synthetic transient renderings to gain more insights into the transient light transport. With this simulated data, we are no longer bound to the imperfect data of real systems and gain more flexibility and control over the analysis.

In a second part, we use the insights of our analysis to formulate a novel reconstruction algorithm. It uses an adapted light simulation to formulate an inverse problem which is solved in an analysis-by-synthesis fashion. Through rigorous optimization of the reconstruction, it then becomes possible to track known objects outside the line of side in real time. Due to the forward formulation of the light transport, the algorithm is easily expandable to more general scenarios or different hardware setups. We therefore expect it to become a viable alternative to the classic back projection approach in the future.

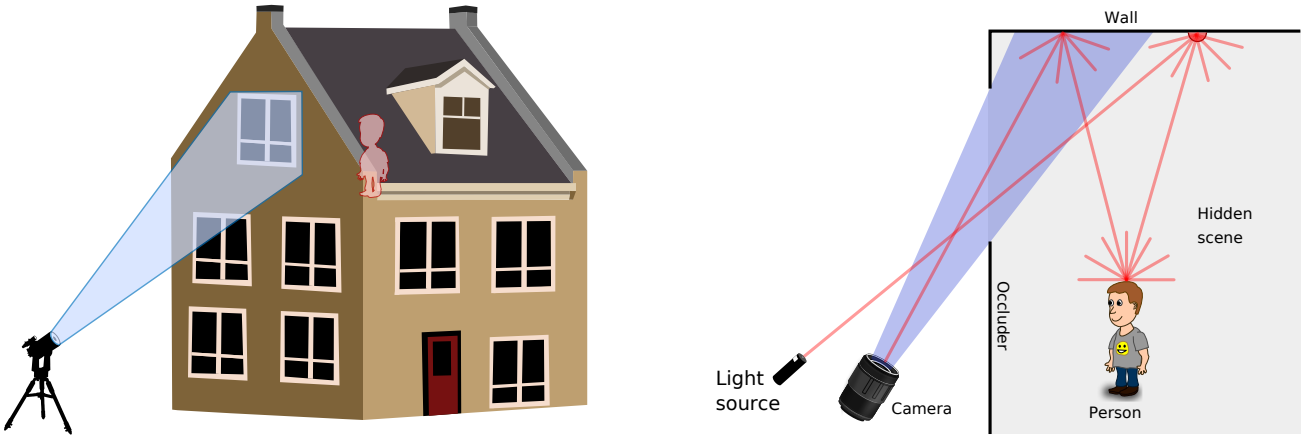
Keywords: Transient Imaging, Simulation, Rendering, Real Time Tracking, Non-Line-of-Sight, Analysis-by-Synthesis

1. INTRODUCTION

Non-line-of-sight (NLoS) imaging aims to gather information about a scene that is not directly visible, i.e. that is occluded by some kind of obstacle. In the past, this problem was often solved by choosing a wavelength in the electro-magnetic spectrum that is capable of penetrating certain materials (e.g. an occluding wall) while still being reflected by other materials (e.g. the object of interest behind the wall).[1,2]

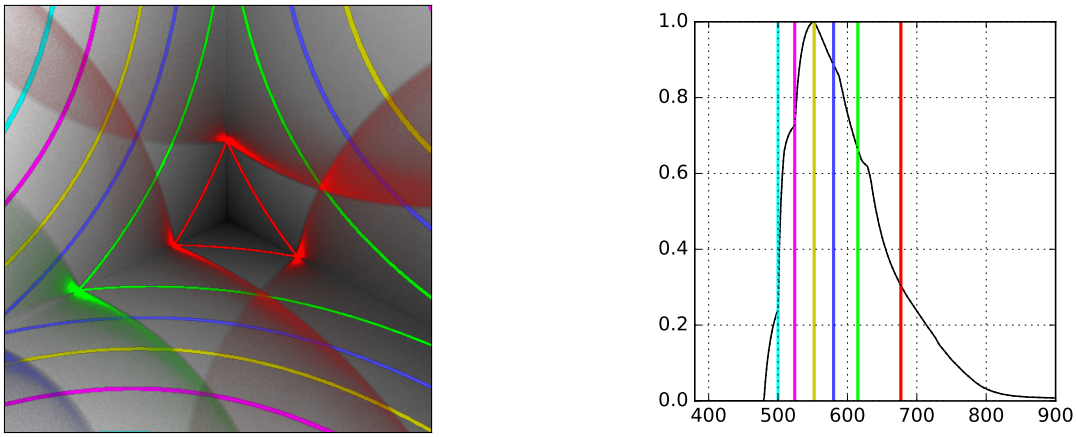
In contrast, the relatively new approach of NLoS imaging attempts to look around the occluder by analyzing light that has undergone multiple reflections. This scenario was first proposed by Kirmani et al.[3] and is depicted in Figure 1. The system consists of two main parts: A camera and a laser light source, both pointed at wall inside the scene (e.g. the back wall of a room) that takes the role of a (diffuse) reflector. From this reflector, the light is reflected towards the object of interest and back to the reflector, where it can be perceived by the camera. The challenge of reconstruction lies in the nature of the three diffuse reflections (wall, object, wall): Firstly, only a small portion of the light is reflected towards the object and only a small part of this light is reflected back to the observed part of the reflector, leading to a total intensity loss proportional to the fourth power of the distance from the object to the reflector. A strong laser and a low light capable camera are therefore required to reduce the signal noise to a level that allows reconstruction. Secondly, all angular information of the reflected light is lost, leading to a very low-frequent intensity distribution on the reflector (see Figure 7). It is notable, that both problems immediately vanish, if a specular reflector, i. e. an ordinary mirror, is used.

NLoS imaging could become a key technology in many different areas, including remote observation, traffic security, endoscopy and many other scenarios, where direct sensing is infeasible, dangerous or even impossible.



(a) (b)

Figure 1: Non-Line-of-Sight Imaging. a) A typical use case of NLoS imaging. An observer stands outside a building, the direct sight into the room is blocked by a wall. However, using NLoS imaging he is still able to see the person inside the room. b) The fundamental principle of NLoS imaging: A camera and a laser are pointed at the back wall of the room. After three diffuse reflections (back wall, object of interest, back wall) the light can be percieved by the camera.



(a) (b)

Figure 2: Visualization of a transient image. The scene consists of a three-plane corner illuminated by an ultra short pulse from a point light. a) The colors depict the various parts of the scene illuminated at the different times, the alpha value encodes the intensities. b) The total light the scene reflects at each time. The colored bars mark the time stamps from a).

Although it might not outperform existing solutions in every aspect, we expect it to find its very own niche in many of these fields.

At the present moment, NLoS imaging is still a rather new technology and only experimental prototypes exist. The main problem remains the loss of angular information, which makes a direct reconstruction impossible. So far the main tool to overcome this problem is the so called *transient imaging* (see Figure 2). The idea of transient imaging is to have a time resolved image of an ultra short light pulse traveling through a scene. Resulting from the high speed of light, the usual time scale of a transient image is in the nanosecond scale, which requires highly specialized hardware for capturing. However, time resolved images add another dimension of information which can be used for the reconstruction of the hidden scenes. We therefore perform an in depth analysis of the transient light transport before we use these insights to propose a new reconstruction scheme for NLoS imaging. The analysis is based on an advanced light simulation which will later, in a different form, act as a main ingredient of our new method.

2. TRANSIENT LIGHT TRANSPORT ANALYSIS

Different hardware setups have been proposed for the recording of transient images. Velten et al. uses a streak camera in combination with a femtosecond laser to record a single line at a time.[4] The line is smeared out over the whole sensor during the integration time, resulting in light reaching a different part of the sensor depending on the time of arrival. By scanning multiple lines after each other a whole transient image is captured. So far this approach results in the highest quality of images, however the capturing process is time consuming and the required hardware extremely expensive.

Amplitude-modulated continuous wave (AMCW) time-of-flight (ToF) cameras are commonly used for depth imaging (e.g. in the Microsoft Kinect v2), but can also be utilized to measure transient images. They illuminate the scene with a modulated light source and measure the phase difference between the outgoing and incoming light. Often, a pixel will not only receive light from a single distance but a mixture of possibly many distances, leading to wrong depth estimates (the so called *multipath effect*). As the measured signal depends on the modulation frequency, multiple frequencies can be used to untangle the returns and correct the measurements.[5,6] Going one step further even complete transient images can be reconstructed.[7,8] This method of acquisition can be fast (up to real time) but the image quality is also inferior to thos captured by streak cameras.

A relatively new development is the use of single-photon avalanche detector (SPAD) cameras. They are capable of sensing individual photons, however only with a certain probability which can be controlled via the voltage by the user. As a probabilistic system, the sensor will not necessary trigger at the first photon that arrives, giving it a chance to sense later once. Each pixel has an independent counter, that stops upon triggering and thus records the arrival time of the photon. If repeated often enough, the probability distribution of a photon arriving at a certain time can directly be measured and correlates with a transient image. However these devices are not only very expensive but due to the stochastic process also heavily affected by noise.

As of today, all systems capable of measuring transient images have severe drawback which is why we perform our analysis on simulated data. As the simulation of transient light transport is an extension of the ordinary light transport, we can build upon years of research that led to algorithms which can render images that are not distinguishable from pictures taken by a real camera. This gives us confidence that our simulated transient images are also physically correct, even if there is no practical way to compare them to ground truth data. On the contrary, carefully simulated data can be used as ground truth comparison for different hardware setups to estimate their systematical errors. As it is free of errors such as noise, this ground truth data is also very well suited to explore the theoretical limits of systems that process transient data which are otherwise affected by those errors as well. Lastly, we also have very fine control over the simulated scene. Precise manufacturing of test scenes can be a difficult task whereas their modeling on a CAD software is relatively easy. This way we have access to the whole spectrum from very simple to very complex test scenes while still always knowing the ground truth geometry.

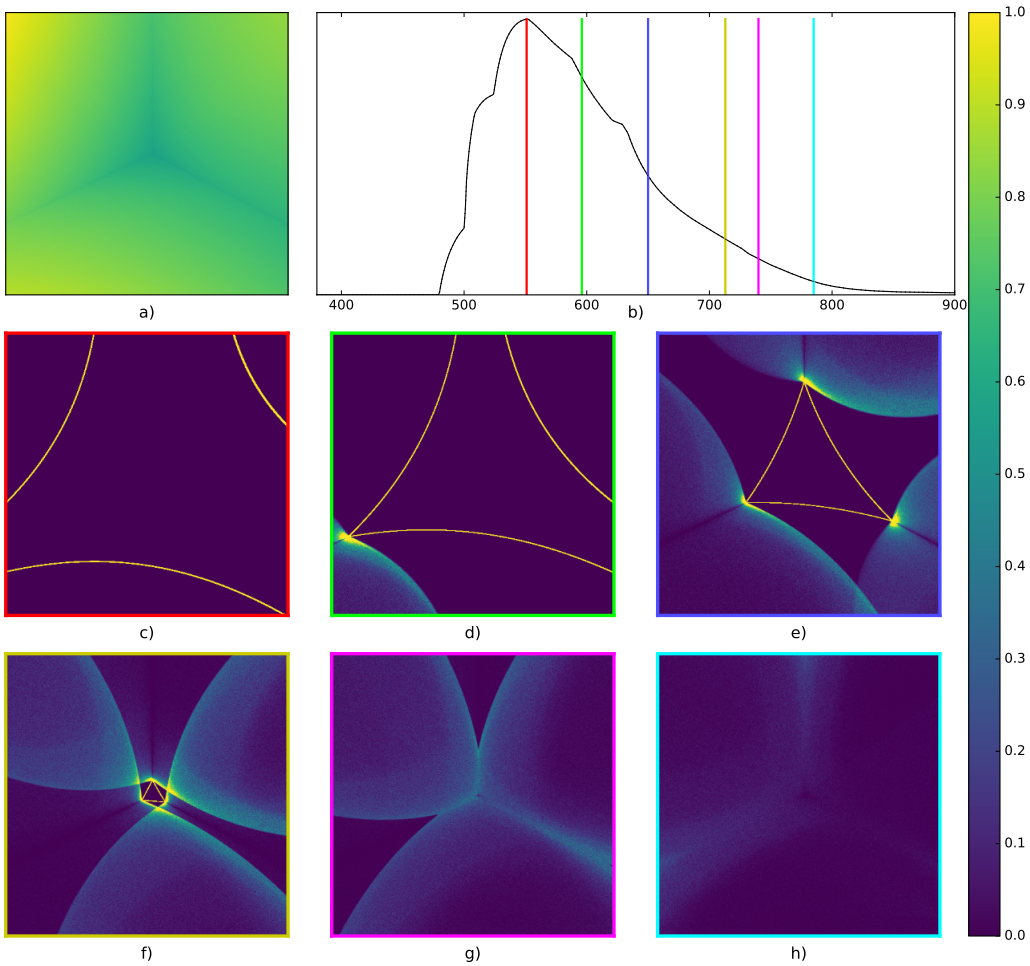


Figure 3: Transient Light Transport Analysis Test Scene 1. A corner consisting of 3 planes is illuminated by a point light. a) The total intensity of each pixel, integrated over time. b) Total intensity per time, integrated over the pixels. c)-h) Several time slices of the scene. The colors correspond to the markers in b). The values are scaled for better contrast, but are consistent for c)-h).

2.1 Simulation

Even though there have been some previous work directly attempting to simulate the transient light transport,[9,10] the closely related problem of simulating AMCW camera data has been payed far more attention so far.[11–14] This is due to the fact, that simulated measurements help to correct the multipath effect which is of huge commercial interest of the manufacturers.

Our implementation is based on pbrt, a state-of-the-art global illumination renderer with strong focus on experimental algorithms and rendering research.[15] One if its extensions is *toftracer*,[16] which is written by Microsoft research and adds the computation of path lengths inside the path tracing algorithm and was originally used to compute histograms of transient light transport for single pixels. We combine pbrt, the toftracer extension, our scene converter for the Blender 3D modeling suite and a post processing script to build our transient rendering pipeline which is capable of rendering transient images for arbitrary complex scenes. For the images shown here, we computed 8192 samples at a resolution of 512×512 pixels, leading to a render time of about 45 hours on a Intel Xeon E5-2690 CPU running 40 logical processors at 3.00 GHz on a system with 256 GB RAM. Our system is, as pbrt, CPU based and therefore does not profit from a high end GPU.

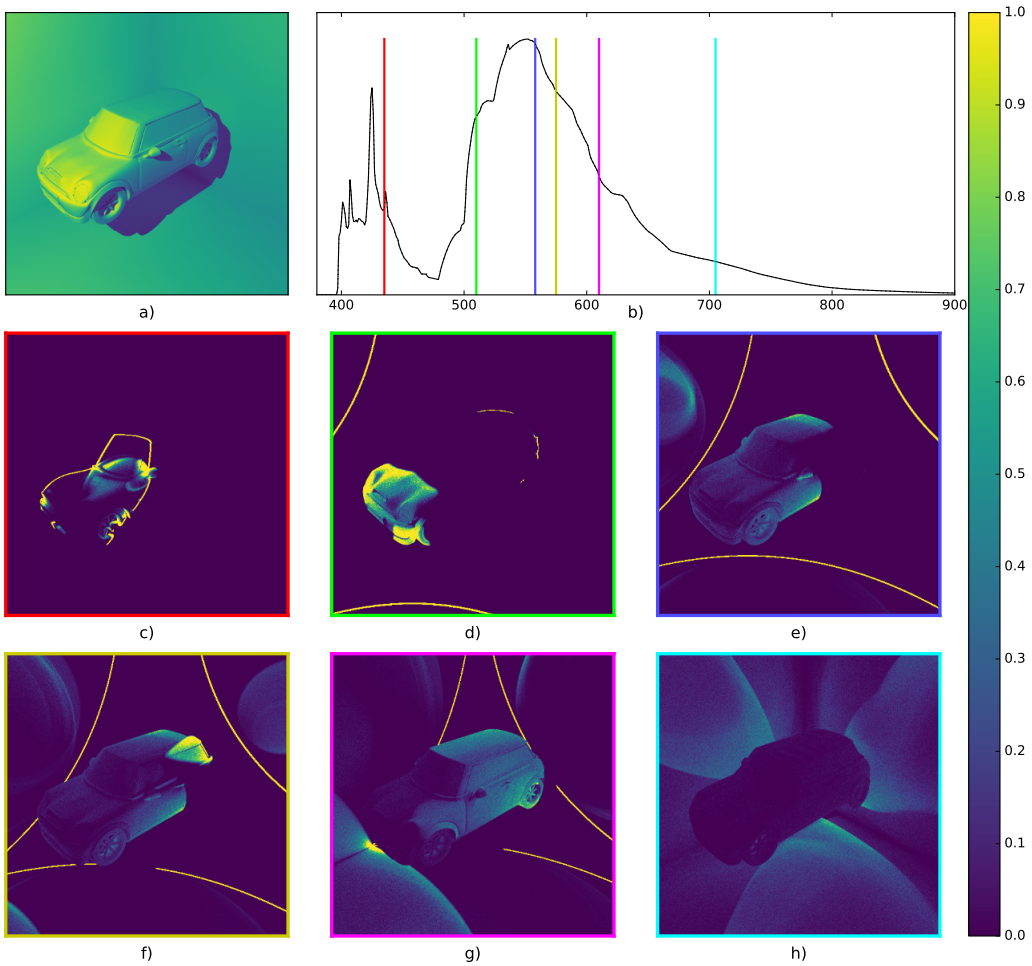
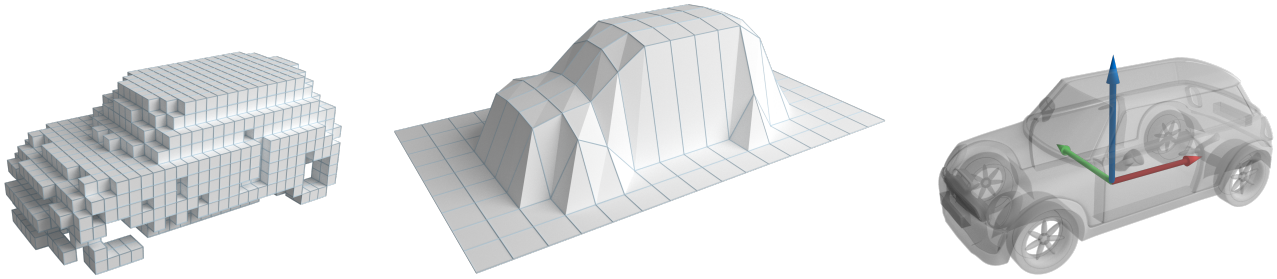


Figure 4: Transient Light Transport Analysis Test Scene 2. The sub figures are analog to Figure 3.

2.2 Results

We perform our analysis on two scenes with varying complexity. Figure 3 shows our first test scene. A corner made out of 3 planes is illuminated by a point light which is placed next to the camera. As this is a three-dimensional time-resolved data set, it is best viewed as a video (see supplementary material); in thepaper we are constraint to only show several frames.

Figure 3a shows the transient image integrated over the time domain, which results in an ordinary intensity image and gives a good intuition of the scene geometry. Figure 3b shows the transient image integrated over the spatial domain, resulting in an 1D function that describes the total amount of light reflected by the scene at a given time. These plots are often useful to locate major events in the time domain, but other interesting events can be completely invisible in this plot. In the first frame (Figure 3c), three wave fronts start to travel across the walls. Each first appearance of a wave front in the transient image corresponds to a sudden intensity increase in Figure 3b; as the light diminishes with the distance, the increase than slowly declines. In Figure 3d the first indirect reflections become visible as a much broader and weaker secondary wave front. These are not solely second order bounces but also higher order bounces, as the additional path lengths close to the corner can be arbitrary small. In Figure 3e the secondary waves from different walls start to overlap and it is clearly visible, that the corners itself experience very few indirect bounces, an effect called *ambient occlusion*. In Figure 3f, the last direct reflections can be seen, in 3g and 3h only higher order reflections remain. Note how the ambient occlusion effect is now reversed and only the short path lengths close to the corners result in considerable light amounts.



(a) (b) (c)

Figure 5: Different levels of reconstruction with varying amounts of degrees of freedom. a) Voxel grid, >10.000 DoF. b) Height map, >1000 DoF. c) Object position, 3 DoF.

Figure 4 shows the more complex second scene where a floating car model was added in fornt of the corner. Due to the added geometry, the interreflections are much more complicated now. The general light distribution in time (Figure 4b) looks similar to the last scene, however the complex signal of the car is added and the signal of the wall is changed slightly due to shadowing.

In Figure 4c the care is now the closest object to the camera and thus first illuminated. Due to its complex geometry it has many interreflections, which are best seen between the windshield and the hood. In Figure 4d the wave fronts on two of the wall appear, together with a much broader secondary wave front on the car, that was reflected from the walls. Note how the left and bottom side of the car are illuminated by different walls. In Figure 4e, a reflection of the car on the left wall is visible, while the primary wave front now completely passed the car. In Figure 4f, indirect reflections from the car to the right wall and from the right wall to the car become visible at the same time. The transient image does not only account for the time delay for the distance from the light source to the scene but also also for the delay induced by the distance from the scene to the camera, which is why both of those indirect reflections have very similar path length. In Figure 4g the reflections between the wall are again visible (as in Figure 3), but more importantly the primary wave front on the bottom wall has a hole where the shadow of the car is. However the indirect wave front in Figure 4h is continuous, as the bottom wall is not occluded by the car when the light comes from the left wall.

These two scenes demonstrate, how our simulation is capable of simulating quite complicated effects in arbitrary scenes. This makes it an extremely useful tool to gain better understanding of the transient light transport.

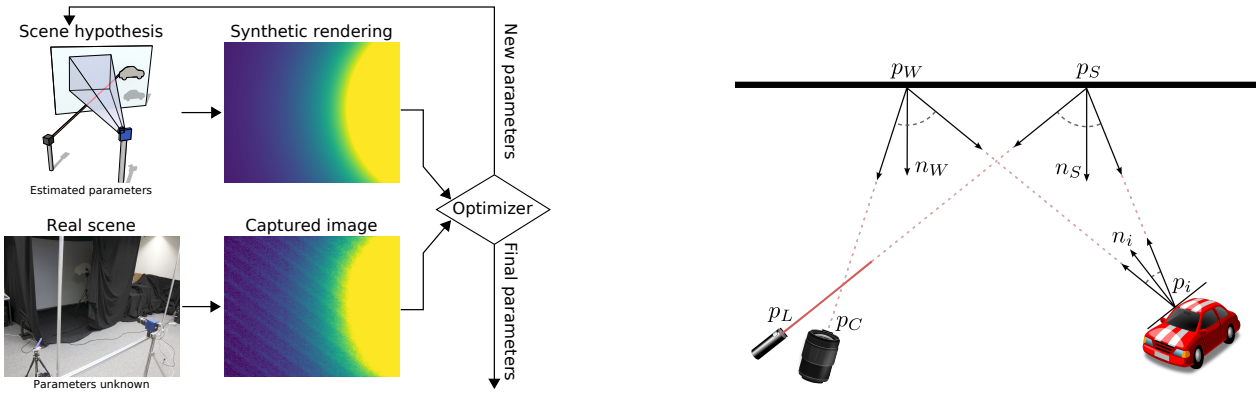
3. NON-LINE-OF-SIGHT IMAGING

We now use the insights gained through the simulation to build a new reconstruction scheme for the problem of indirect vision. Most solution approaches reported so far use a back projection scheme as in computed tomography,[17] where each intensity measurement taken by the imager votes for a manifold of possible scattering locations.[18–22] This *explicit* reconstruction scheme is computationally efficient, in principle real-time capable,[22] and can be extended with problem-specific filters.[20,21] However, it assumes the availability of ultrafast time-resolved optical impulse responses, whose capture still constitutes a significant technical challenge.

In contrast, *implicit* methods state the reconstruction task in terms of a problem-specific cost function that measures the agreement of a scene hypothesis with the observed data and additional model priors. The solution to the problem is defined as the function argument that minimizes the cost. In the only such method reported so far,[23] the authors regularize a least-squares data term with a computationally expensive sparsity prior, which enables the reconstruction of unknown objects around a corner without the need for ultrafast light sources and detectors.

3.1 Computer graphics driven approach

Our approach is based on the insight, that a scene reconstruction can per performed on different levels, resulting in a different amount of degrees of freedom (see Figure 5). In this example the scene to be reconstructed solely



(a) (b)

Figure 6: Our Analysis by synthesis approach. a) The optimizer adopts the parameters of the scene hypothesis until the synthetic rendering matches the captured image. These parameters are then believed to accurately describe the real scene and returned. b) Simplified light transport in the three bounce setup.

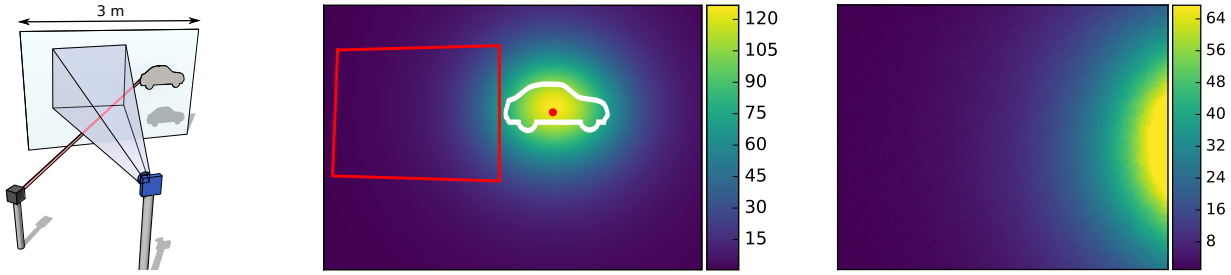
consists of a car. On the first and most detailed level, the scene is modeled as a voxel grid (as used in e.g. Velten et al.[21]), which results in a enormous number of degrees of freedom (one per voxel) when at least a reasonable resolution is chosen. This level gives a great amount of detail, however it is usually very hard to reconstruct. On the second level the scene is modeled as a height map (as used in e.g. Heide et al.[23]), which is a very reasonable approximation, if it can only be viewed from one direction (as it is the case in our basic setup, see Figure 1b). The data structure then is only two-dimensional which drastically decreases the number of degrees of freedom. In the last and most simplified level, the scene is modeled as a car object which has always the same shape, but varies in position. Then the only degrees of freedom remaining are the position which can greatly simplify the reconstruction. However this level can only be used for scenes consisting of exactly one object with known geometry. This may render it useless in some cases, but is a valid assumption in many others (e.g. humans can be approximated by a single shape reasonably well). We use this approach that was previously unseen in literature for our reconstruction scheme.

Inspired by our foregoing analysis we perform the reconstruction in an analysis-by-synthesis fashion. As the forward problem (the rendering of a given scene) is already solved, we formulate the inverse problem by using the rendering part inside a numerical optimization, as depicted in Figure 6a. This approach has been applied to a number of other problems before (e.g. Nayar et al.[24]), but to our knowledge we are the first to introduce it to this particular problem.

In order to perform the rendering, we must know the scene geometry (mainly the position and orientation of our camera system and the reflector), the shape of the object and the properties of the involved materials. In the future, the scene geometry could be acquired by an on-the-fly calibration step by measuring the distance and orientation of the reflector (possibly with a depth camera) and assuming that our camera and laser form a rigid system with known properties. In our experiment we perform the calibration by manually measuring the setup. We assume that the shape of our objects (e.g. humans, cars, ..) are either known or similar enough to be modeled by a reference mesh. In an extension, this reference model could have additional degrees of freedom to adapt its shape in certain way (e.g. a scaling factor). As a last step we assume that all materials can be modeled as Lambertian surfaces with a varying albedo, which we find to be accurate enough in our experiments. However, it is also be possible to use arbitrary BRDFs inside the rendering to account for arbitrary materials if the material BRDFs are known or can be measured.

The optimization starts with an initial estimate of the object's position (usually somewhere in the middle of the scene). The light transport is computed and compared to the image acquired by the camera. The optimizer then iteratively adjusts the object translation, until the rendering is close enough to match the captured image. The best fitting position is then returned as the result.

The pixels of both images are interpreted as entries of huge vectors and compared by computing the euclidean



(a) (b) (c)

Figure 7: Shape of our camera signal. a) Schematic overview over the scene. b) Light reflected from the object onto the wall. The red rectangle is the view frustum of the camera, the red dot the laser spot on the wall. The white car silhouette is the object's shape and size projected onto the wall. c) What the camera actually sees. The intensity units are arbitrary, but consistent between b) and c).

distance between them. This leads to the following least square minimization problem:

$$p_{\text{opt}} = \arg\min_{p} \left\{ \left\| I_{\text{meas}} - S\left(p\right) \right\|_2 \right\} \tag{1}$$

with p_{opt} as the solution (the optimal transformation parameters), I_{meas} as the image as measured by the camera and $S\left(p\right)$ as the rendering depending on a parameter set p.

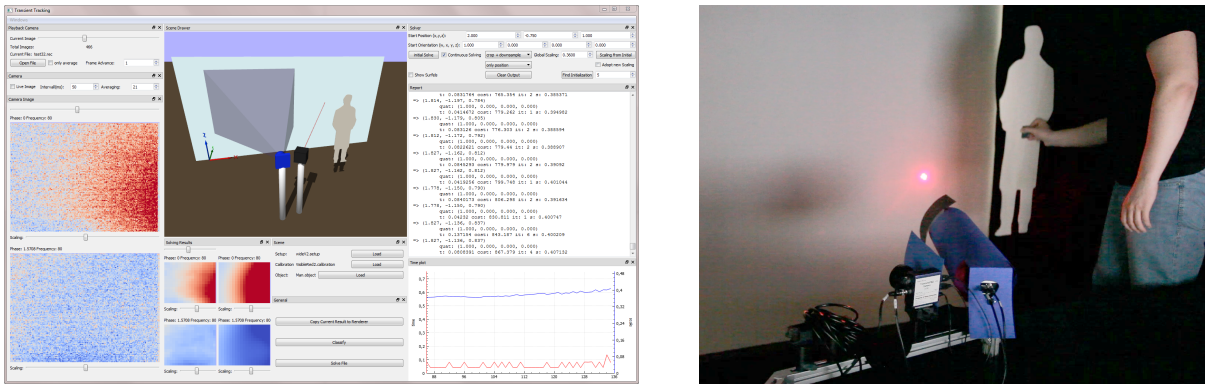
As the rendering of the scene happens inside the optimization, a high rendering speed is crucial. We therefore drastically simplify the very general path tracing approach used before to a fixed three-bounce scenario (see Figure 6b). As there is no possibility for diverging rays, this approach can be computed very efficiently on a GPU. We split the object shape into small *surfel* (surface elements) which only have a position, size and orientation, but no shape. We then can compute the light transport for each surfel independently and add the intensities to retrieve to total light reflected by the object. According to the identifiers used in Figure 6b, the distances D and transported light I from p_L to p_C are computed by:

$$D\left(p_L, p_C\right) = \left\| p_L - p_S \right\|_2 + \left\| p_S - p_i \right\|_2 + \left\| p_i - p_W \right\|_2 + \left\| p_W - p_C \right\|_2 \tag{2}$$

$$\begin{aligned} I\left(p_C, p_L\right) =& a_i \cdot \frac{1}{\left\| p_W - p_i \right\|_2^2} \cdot \frac{1}{\left\| p_i - p_L \right\|_2^2} \cdot \epsilon\left(p_L\right) \\ & \cdot \cos\angle\left(n_W, p_i - p_W\right) \cdot \cos\angle\left(n_i, p_W - p_i\right) \\ & \cdot \cos\angle\left(n_i, p_S - p_i\right) \cdot \cos\angle\left(n_S, p_i - p_S\right) \end{aligned} \tag{3}$$

Where a_i is the surfel area, $\epsilon\left(p_L\right)$ the light emitted from the laser and $\cos\angle\left(.,.\right)$ the clamped angle between the vectors. With D and I we have all the quantities to compute the transient image of the scene as seen by our camera. As we use a PMD camera and the reconstruction of transient images from the phase images that the camera measures is difficult, we transform our synthetic rendering into phase images. To this end we measure the camera response to various distances and apply it in a straightforward process on the transient image. The resulting images can then directly be used in Equation 1.

Figure 7 shows the usual shape of our camera signals and gives an intuition of some of the problems that the optimizer has to deal with. Due to the diffuse reflections, the amount of light diminishes with the fourth power of the distance from the object to the wall. The result is a very high dynamic range of our camera signal, in case of the scene shown in Figure 7 almost all light is concentrated on the right side. Furthermore, the peak of the signal is not within the field of view of the camera in most cases, meaning that the strongest part of the signal remains unobserved.



(a) (b)

Figure 8: Tracking tool and experimental setup. a) The graphical user interface of our application provides allows the user to control the tracking process by specifying various options. Additional real time information about the tracking performance are shown. b) The tracking system consists of a camera and laser, both with applied lens hoods. The tracked object is a wooden shape that can be freely moved inside the scene.

3.2 Results

Our implementation is written in C++ and uses CUDA to run the rendering part on the GPU. A Qt GUI (see Figure 8) allows the user to change parameters, load calibration files and control the tracking process. Due to our rigorous optimizations, we achieve approximately 10 frames per second and therefore real time frame rates.

Our tracking hardware consists of a modified *PMDTec camboard nano* and a custom built laser light source attached to it (Figure 8). To avoid lens flare artifacts and stray light from the laser, lens hoods are added to the camera and the laser.

To evaluate the accuracy of our tracking results, the object is fixated on a tripod and moved along the three main axes. A series of images is recorded at each position and optimized independently which allows the computation of the average accuracy and variance at each position (see Figure 9). We find that while the results have a slight bias, the general movement of the object is still reconstructed faithfully.

4. DISCUSSION AND FUTURE WORK

We presented a framework to compute the transient light transport in arbitrary complex scenes. On two examples we performed a detailed analysis and showed the various effects that occur. Based on this insights we developed a novel reconstruction scheme for the problem of indirect vision around corners. Thanks to the drastic specialization and optimization, our light simulation is fast enough to be used insight a numerical optimization and is therefore suitable to solve the inverse problem of deducing scene properties from the measurements.

In future work, the limits of our approach should be evaluated. More degrees of freedom, such as rotation, scaling, moveable joints or other mesh transformations would greatly increase the number of scenes that can be represented reasonably well, but it is unclear, whether and when the induced signal changes are significant enough for the numerical optimization to find the correct set of parameters.

We expect that our analysis-by-synthesis approach will become an important alternative to the usual back projection approach. Especially in situations where more abstract scene parameters (such as the rigid transformation of an object) are to be reconstructed, we believe our approach to be superior to solutions that require a voxel grid or height map reconstruction as an intermediate step. Furthermore our approach is very flexible which allows it to be adopted to many kinds of different hardware (AMCW lidar, streak cameras, SPAD sensors, ...).

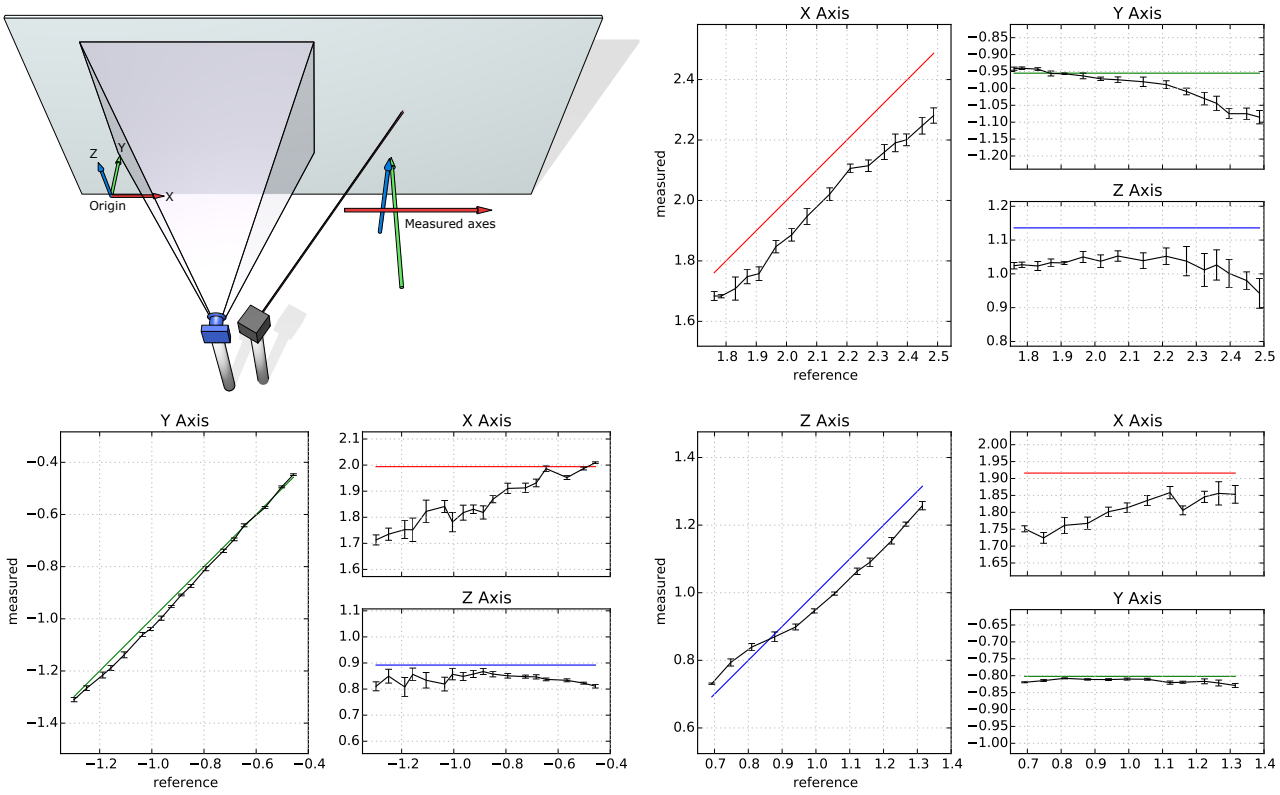


Figure 9: Accuracy Results. The object is moved in discrete steps along the three main axes; at each position multiple measurements are taken. a) Overview of the setup and range of each axis. b)-d) Reconstruction results along each axis. The colored lines are the reference value, the error bars show the variance.

5. References

[1] Adib, F. and Katabi, D., "See through walls with wifi!," *SIGCOMM Comput. Commun. Rev.* **43**, 75–86 (Aug. 2013).

[2] Sume, A., Gustafsson, M., Herberthson, M., Janis, A., Nilsson, S., Rahm, J., and Orbom, A., "Radar detection of moving targets behind corners," *Geoscience and Remote Sensing, IEEE Transactions on* **49**, 2259–2267 (June 2011).

[3] Kirmani, A., Hutchison, T., Davis, J., and Raskar, R., "Looking around the corner using transient imaging," in [*Proc. ICCV*], 159–166 (2009).

[4] Velten, A., Raskar, R., and Bawendi, M., "Picosecond camera for time-of-flight imaging," in [*OSA Imaging Systems and Applications*], (2011).

[5] Dorrington, A., Godbaz, J., Cree, M., Payne, A., and Streeter, L., "Separating true range measurements from multi-path and scattering interference in commercial range cameras," in [*Proc. SPIE*], **7864** (2011).

[6] Kirmani, A., Benedetti, A., and Chou, P. A., "Spumic: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods," in [*2013 IEEE International Conference on Multimedia and Expo (ICME)*], 1–6 (July 2013).

[7] Heide, F., Hullin, M. B., Gregson, J., and Heidrich, W., "Low-budget transient imaging using photonic mixer devices," *ACM Transactions on Graphics (Proc. SIGGRAPH)* **32**(4), 45:1–45:10 (2013).

[8] Peters, C., Klein, J., Hullin, M. B., and Klein, R., "Solving trigonometric moment problems for fast transient imaging," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* **34**, 220:1–220:11 (Nov. 2015).

[9] Smith, A., Skorupski, J., and Davis, J., "Transient rendering," Tech. Rep. UCSC-SOE-08-26, University of California Santa Cruz, School of Engineering (2008).

[10] Jarabo, A., Marco, J., Muñoz, A., Buisan, R., Jarosz, W., and Gutierrez, D., "A framework for transient rendering," *ACM Transactions on Graphics (SIGGRAPH Asia 2014)* **33**(6) (2014).

[11] Lambers, M., Hoberg, S., and Kolb, A., "Simulation of time-of-flight sensors for evaluation of chip layout variants," *IEEE Sensors Journal* **15**, 4019–4026 (July 2015).

[12] Keller, M. and Kolb, A., "Real-time simulation of time-of-flight sensors," *J. Simulation Practice and Theory* **17**, 967–978 (2009).

[13] Keller, M., Orthmann, J., Kolb, A., and Peters, V., "A simulation framework for time-of-flight sensors," in [*2007 International Symposium on Signals, Circuits and Systems*], **1**, 1–4 (July 2007).

[14] Meister, S., Nair, R., and Kondermann, D., "Simulation of time-of-flight sensors using global illumination," in [*Vision, Modeling & Visualization*], Bronstein, M., Favre, J., and Hormann, K., eds., The Eurographics Association (2013).

[15] Pharr, M. and Humphreys, G., [*Physically Based Rendering, Second Edition: From Theory To Implementation*], Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd ed. (2010).

[16] Pitts, P., Slaney, M., Benedetti, A., and Chou, P., "Time-of-flight tracer." https://toftracer.codeplex.com/ (11 2014).

[17] Pan, X., Sidky, E. Y., and Vannier, M., "Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?," *Inverse Problems* **25**(12), 123009 (2009).

[18] Buttafava, M., Zeman, J., Tosi, A., Eliceiri, K., and Velten, A., "Non-line-of-sight imaging using a time-gated single photon avalanche diode," *Optics Express* **23**(16), 20997–21011 (2015).

[19] Laurenzis, M., Klein, J., Bacher, E., and Metzger, N., "Multiple-return single-photon counting of light in flight and sensing of non-line-of-sight objects at shortwave infrared wavelengths," *Opt. Lett.* **40**, 4815–4818 (Oct 2015).

[20] Kadambi, A., Zhao, H., Shi, B., and Raskar, R., "Occluded imaging with time of flight sensors," *ACM Transactions on Graphics 2015* (2015). to appear.

[21] Velten, A., Willwacher, T., Gupta, O., Veeraraghavan, A., Bawendi, M., and Raskar, R., "Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging," *Nature Communications* **3**, 745 (2012).

[22] Gariepy, G., Tonolini, F., Henderson, R., Leach, J., and Faccio, D., "Detection and tracking of moving objects hidden from view," *Nature Photonics* **10**(1) (2016).

[23] Heide, F., Xiao, L., Heidrich, W., and Hullin, M. B., "Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2014).

[24] Nayar, S., Ikeuchi, K., and Kanade, T., "Shape from interreflections," **6**(3), 173–195 (1991).