AFRL-AFOSR-VA-TR-2018-0422

Super Resolution Remote Imaging using Time Encoded Remote Apertures

Andreas Velten
UNIVERSITY OF WISCONSIN SYSTEM

11/28/2018
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTB1
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
31-10-2018	final	1 August 2016 - 31 July 2018

4. TITLE AND SUBTITLE
Super Resolution Remote Imaging using Time Encoded Remote Apertures

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Andreas U Velten

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Wisconsin-Madison
Research and Sponsored Programs
21 N. Park St., Ste. 6401
Madison, WI 53715-1218

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Remote Sensing

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Pubic Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The report summarizes our efforts on this project. We investigated a novel method of reconstructing images of scenes purely from their time responses. Our reconstructions are not affected by diffraction and thereby allow for imaging with high resolution over large (theoretically unlimited) distances. We describe several attempts at development of reconstruction methods and demonstrate a method that can reconstruct the geometry of realistic sparse swarms of objects. We demonstrate the method in simulation and experiment. We believe this work creates a viable new form of imaging with intriguing properties and definite strengths compared to existing approaches.

15. SUBJECT TERMS
Super resolution, remote sensing, imaging, target recognition

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
			UU	14	

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATE COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33315-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report. e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Final Report

The objective of this project is to explore the possibility of super resolution imaging using multibounce scattered light. It is based on the insight that time resolved multibounce light from a scene, i.e. light that has been reflected within the scene multiple times before traveling to the imaging system encodes spatial information about a scene in a unique way. Existing imaging systems create images by detecting a wave front of the EM wave traveling from the object. While a wave front diffracts and thus loses information as it travels, time encoded information remains unchanged. Our time encoded imaging approach thus provides us with a method that is not affected by diffraction and therefore does not have a diffraction limit. The objective of our three-year project was to perform research that lays the groundwork for a prolific research program investigating this phenomenon. The achievements we are reporting can be summarized as:

1. A framework to image sparse scenes i.e. scenes of few unconnected objects comprising:
    a. A method to reconstruct images of sparse scenes, that is robust to missing data, variations in object reflectance properties, and object occlusions.
    b. An experimental demonstration of the reconstruction method.
    c. A set of simulations to test the reconstruction method for a large number of randomly generated scenes.
    d. Mathematical guarantees regarding uniqueness and existence of a reconstruction.
2. Methods to reconstruct more general scenes made up of surfaces placed in a 3D scene space. Currently these methods only work for very simple scenes in simulation. Further research is required.
3. Methods to classify scenes for target recognition. These methods also require further research.

We are preparing a manuscript to publish the results of (1) while the other two items require further attention in follow up projects. In addition to our research output we have trained two graduate student researchers who have made substantial intellectual contributions to the project. We also now have a fully equipped functional lab with high end illumination and capture systems to further investigate the developed methods.

Theoretical Background

Fundamentals of imaging

In this section we establish a very fundamental conceptual understanding of what an imaging system does to obtain an image of an object. The purpose is to define terms for later use and avoid confusion. It may also help understanding of the imaging process to non-experts.

Consider an object emitting or reflecting light. Some of the light is detected by an imaging device at a distance d from the object. In order to reconstruct an image of the object the imaging device needs to establish the point of origin for any photon or light ray striking the device. To do this the device could measure different quantities. Measuring the position on the sensor aperture where the photon struck the sensor along with the angle the photon arrived from is one possibility. The eyes of certain animals like krill and to some extent the compound eyes of insects work in this way. An alternate set of measurements is to measure the positions where two or more photons strike the imaging device and compare the distances travelled by those photons from the object. With this knowledge one can the triangulate the position of their point of origin on the object. This is fundamentally what a lens does to project an image. It redirects all rays from a certain point on the object such that they constructively interfere on exactly one point in the image plane to create an image of that object point. Conventional imaging thus evaluates the length of a light ray as encoded in its phase (Figure 1). The accuracy with which that length can be established is directly related to the wavelength and gives rise to an airy disc on the sensor and the Rayleigh diffraction limit:

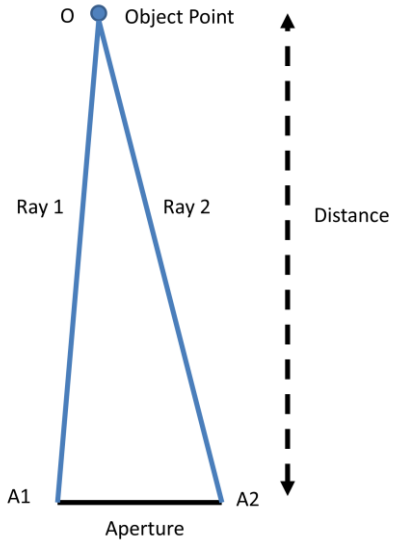


Figure 1 Fundamental relation between distance, resolution and aperture. To infer the position of an object point O within the focal plane the imaging system needs to determine the difference in length between Rays 1 and 2. For a constant aperture size this difference decreases linearly with distance.

$$r = 1.22\frac{\lambda d}{D}$$

where r is the distance between two barely resolvable points in the scene, λ is the wavelength of the light used, d is the distance between imaging system and scene, and D is the diameter of the aperture of the imaging system.

A pulsed or intensity modulated laser is a source with a high second order coherence. An object illuminated by a short pulse of light and observed by a lens-less sensor where every pixel can detect the time of flight of any light as well as its intensity can reconstruct an image similar to the image created by a lens. The resolution of such an image is described by the Rayleigh criterion, except that the wavelength λ is replaced by the time resolution τ of the sensor and light source. In this document this resolution is referred to as the transient Rayleigh Criterion:

$$r = 1.22\frac{\tau d}{D}$$

In LiDAR systems we do not typically use second order coherence for image formation, because for most current imaging scenarios τ is much larger than λ. Typically imaging LiDAR or gated viewing systems use phase and lens-based image generation methods and in addition use time of flight to obtain a better depth accuracy. The principles of TERA can be applied to first and second order coherence and we will explore both in this project.

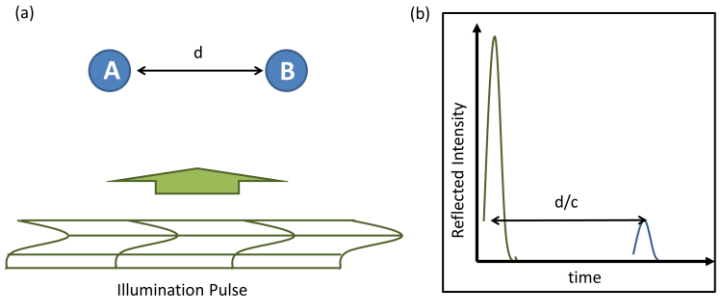


Figure 2 Time response of a scene containing two small objects A and B. The distance d between the objects can be inferred from the time response (b) of the scene even if the imaging system is too far away to resolve the scene spatially.

The fundamental problem of creating a conventional image is that one requires an extraordinarily accurate distance measurement to create an image of a given resolution. That accuracy has to increase linearly with imaging distance. Consider for example Figure 1. In order to establish the position of point O in the image plane, the imaging system has to measure the difference in distance between light rays 1 and 2. This is of course simply another way to express the Rayleigh criterion.

Time Encoded Remote Aperture (TERA) imaging

The situation changes when one allows for light rays to bounce between object points in the scene before travelling back to the detector. If these multiple reflections can be detected they encode the distance between two object points in the scene directly without the need for a triangulation step as in Figure 1. We then can in principle obtain a resolution that depends on τ or λ directly rather than via the Rayleigh criterion. Multibounce light paths can thus be used to computationally place the aperture into the scene itself if the scene is shaped such that light is reflected back and forth between objects in the scene. The resolution limit of such an image, the TERA Rayleigh criterion, is simply $r = \lambda$ for imaging with first order coherence or $r = \tau$ when imaging with second order coherence.

Figure 2 illustrates the way information about a scene is encoded in time using a very simplified example. Two small point objects A and B separated by an unknown distance are illuminated by a single plane wave for a short time (picoseconds or even femtoseconds). For simplicity we make the assumption that these objects are points, i.e. have a fixed diameter δ that is slightly smaller than the resolution of the final reconstructed image, and reflect a certain amount of light equally in all directions described by a reflectance factor ρ. Eventually we will extend this model to include connected surfaces rather than disconnected points. In this extended

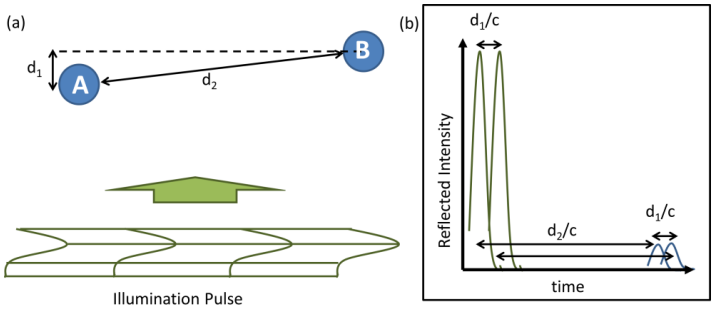


Figure 3 Time response with a different configuration of objects. The distances d_1 and d_2 are sufficient to completely reconstruct the scene and can be inferred from the time response.

model the diameter of each point or patch will be evident from its position.

The illuminations wave is emitted from a LiDAR imaging system far from the scene. The distance between the imaging system and the scene is so large that the distance between A and B is much below the Rayleigh diffraction limit for the used imaging aperture. All we detect is a single point in space with a time response that is shown in Figure 2b. It contains one impulse due to light directly reflected from objects A and B and one impulse due to light reflected from A towards B and vice versa and then back toward the detector. The time between these two pulses indicates the distance between A and B which is the only relevant parameter in this scene.

In Figure the position of A and B is altered and the detected time response changes as a consequence. The time response in Figure b allows us to conclude the distances d_1 and d_2. We also know the distance d_0 from the imaging system to the scene. This information is sufficient to completely reconstruct the relative positions of A and B in two dimensions except for a translation perpendicular to the optic axis and rotation around the optic axis of the imaging system. It is also clear that one can determine if the scene contains 1, 2, 3 or more spheres and it is intuitively possible to determine their positions for scenes with any number of points as long as the time resolution is sufficient to separate the responses of all the points involved and the points do not occlude each other. In the following sections we will lay out the steps necessary to move from this simplified version of TERA to a practical and realistic method that can be used in applications.

Inverse rendering and the radiative transfer equation

For scene geometries large compared to the wavelength and neglecting nonlinear optical effects, the interaction of light with a scene to create an image can be described by the rendering equation [25], [26]

$$f = K(h + Gf)$$

f describes light reflected by surfaces in the scene and part of f is used to generate the image detected by an imaging system. The global illumination term G describes how this light interacts with the objects in the scene. h represents the illumination of the scene and K describes the direct reflection of light by the surfaces in the scene. f depends on the geometry and surface properties of the scene. It may also include effects like subsurface scattering or diffraction. While an important component of the rendering equation is the direct light that travels from the light source to a scene surface and from there directly to the imaging device, a large and crucial portion of the signal detected by the imager depends on so-called indirect or global illumination. This includes subsurface scattering, as well as the large portion of light that is reflected multiple times by surfaces in the scene before finally ending up in the sensor. This multibounce light carries the information we are interested in. While this global illumination is important when rendering a realistic image of a scene it does not contribute to image resolution in conventional imaging techniques.

While the rendering equation typically describes the steady state of light transport that is perceived by humans, it can be extended to include a time dimension to model how fluctuations in the light source intensity propagate through the scene. Using this new time dimension opens up new possibilities for the modeling and investigation of light transport. Among the beneficiaries of this new data is the field of inverse rendering, i.e. the task of inverting the rendering equation and computing the scene geometry and other properties from the collected data. This is essentially what is done by the human brain intuitively when interpreting a two dimensional image of a three dimensional scene. Without a time dimension in the rendering equation this is an extremely difficult problem. TERA imaging can be seen as a form of inverse rendering where only the time evolution of f is detected by the imaging system.

1. Creation of a modeling environment

Initially we had planned to use DIRSIG as the software to base our computational simulations on. DIRSIG stands for Digital Imaging and Remote Sensing Image Generation model and is tool to create physically accurate renderings of scenes. DIRSIG has been tested for its accurateness in typical remote imaging scenarios and can simulate time of flight imaging data. There are several concerns with DIRSIG as a simulation platform. It is developed for standard remote imaging tasks and its photon mapping techniques seem to be sub-optimal for multibounce data. It is challenging to separate simulated returns into bounce components. The software is closed source, so there is no direct way for us to verify correctness or fix problems. We therefore are using our own implementation for forward rendering that is based on MATLAB for simple scenes. We are planning on incorporating GPU processing in this software in the near future. In addition we are making use of transient rendering software developed by Jarabo et. al. at the University of Zaragoza Specifically for the purpose of rendering multibounce time of flight imaging data [1].

In addition we created our own rendering tool for use in sparse scenes. For the simulations of sparse point clouds shown in section 5 of this document, none of the existing renderers is efficient. Existing renderers usually make the assumption that a scene has many objects in it and that most ray of light in the scene will eventually strike a surface and therefore needs to be rendered. Our point cloud scenes

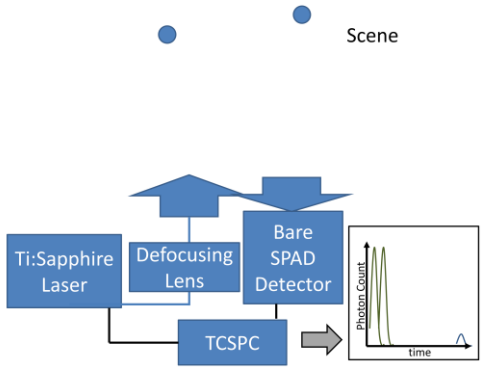


Figure 1: Experimental setup. The laser pulse is used to flash illuminate the entire scene. A lensless detector of 20 micron diameter is used to detection returning light.

2. Proof of concept experiment

As proof of concept we are designing an experiment to image and reconstruct a simple scene. The setup for this experiment is shown in Figure 1. We replaced the light source and part of the detection electronics compared to the setup that was described in the proposal. A frequency doubled pulsed Ti:Sapphire laser is used as a light source that provides sub 100 fs pulses at a repetition rate of 75 MHz. The frequency doubled beam has an average power of about 250 mW at 400 nm. This new light source is thus more powerful than the old system that delivered 50 mW. This beam is directed onto a scene and defocused to illuminate an area with about 20 cm diameter. The first scene consists of two white spheres of 1 inch diameter. The returned light is detected by a Single Photon Avalanche Diode (SPAD) detector. For two spheres the collected data includes three peaks as indicated in Figure 1. From the position of those peaks we can obtain an estimate of the position of the spheres.

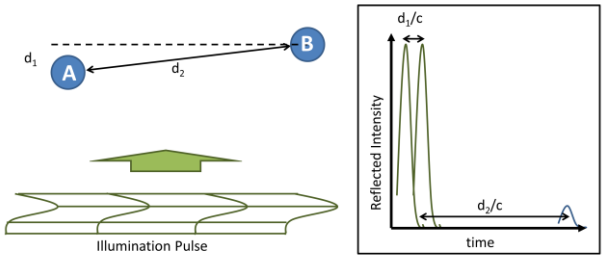


Figure 2: Expected return from a scene with two spherical objects A and B.

We have completed setup of the light source and frequency doubling. We have replaced the TCSPC unit of our setup and have been delayed by equipment lead times. Currently our SPAD setup is back to being operational and we will collect data for the proof of concept scene shortly.

4. Optimization based reconstruction

The main research component scheduled in year 1 was the investigation into models and reconstruction algorithms for TERA. We are currently working on different approaches for a reconstruction:

Second order Back-projection

We developed a second order back-projection algorithm to obtain approximate reconstructions of a scene. For this algorithm we divide the reconstruction space into a set of voxels $v(x)$. We assume the collected data is separated in photon numbers for first bounce returns $N_1(t)$ and second bounce returns $N_2(t)$. For each combination of voxel positions x_1 and x_2 we increment the two voxels by the number of photons collected that could have undergone reflections at these two locations.

$$v(x_{1,2}) = v(x_{1,2}) + N_1(d(s \to x_1 \to s)/c)N_1(d(s \to x_2 \to s)/c)N_2(d(s \to x_1 \to x_2 \to s)/c)$$

Where s is the location of the imaging system and d is the distance of the path connecting the points listed in the function argument. The thus computed values can be interpreted as a vote by each collected data sample towards the voxels that are consistent with light being detected in that particular time bin. For a large number of votes the reconstructed scene in the voxel space resembles the test scene. Results are shown in Figure 3.

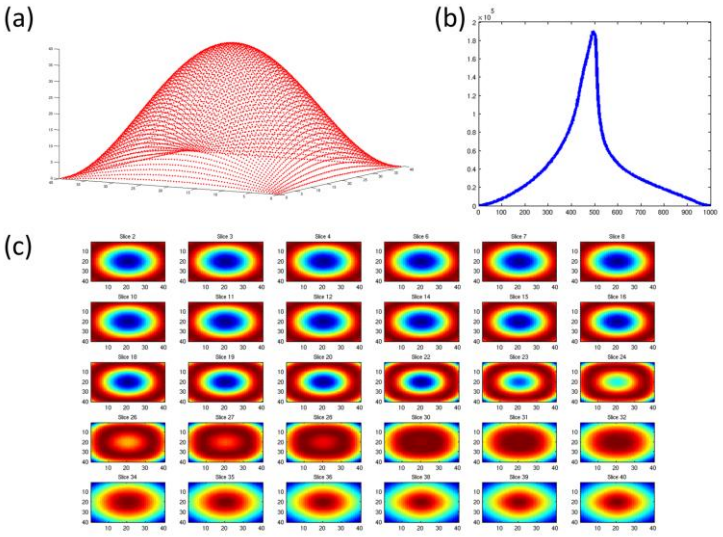


Figure 3: (a) Model of the scene to be reconstructed. The scene is illuminated from the bottom, along the z-axis. (b) Simulated second bounce time response of the scene. (c) Back-projection reconstruction. The individual images are slices through the reconstructed volume in the x-y-plane. Z is increasing from top left to bottom right.

The plot of the data shows that the shape is recreated and important features are preserved. The reconstruction resolution is quite low and more complex shapes would be hard to simulate without increasing the number of voxels. Because of the second order interaction the generation of the data and the back-projection are time intensive and treatment of larger voxel spaces would require significantly more computing power. From the equations governing the back-projection we can show that the back-projection result is invariant under exchange of the x and y coordinate and therefore the reconstruction result will always be rotationally symmetric even if the scene is not. For a not rotationally symmetric scene it is not completely clear at this point how the back-projected scene relates to the real scene. It is likely that a simple geometric transform on the true scene geometry, such as a rotation, describes the back-projected geometry. The back-projection thus has the disadvantages, that it provides low resolution and does not reconstruct all geometric information about the scene in the case of general geometries.

On the other hand, the back-projection approach has the advantage that its runtime does not depend on the complexity of the scene. It only depends on the number of voxels in the reconstruction volume. It also always provides a result that within its resolution limitations provides reliable information about the scene while other algorithms may converge to results that are not related to the true scene.

Analysis by Synthesis Reconstruction

A more complex, but potentially more powerful approach to the reconstruction is the use of analysis by synthesis (AbS) reconstruction methods. This group of methods relies on knowledge of a data generation model, i.e. knowledge of what measured time response would be detected from a given known geometry. AbS algorithms start with an initial guess for the geometry of the scene and generate the corresponding time response. This time response is compared with the measured time response from the true scene. The guess of the geometry is then adjusted and a new time response is generated and the process is repeated until the computed time response is identical to the measured time response. Different AbS algorithms differ primarily in the way the guess is adjusted in between iterations. A simple random adjustment represents the simplest and most general approach. More sophisticated algorithms attempt to improve adjustment by incorporating knowledge of how the scene geometry and the time response are related to determine the best direction and magnitude for an adjustment. In the first implentation we are using an AbS algorithm with random adjustments.

Multilevel gradient decent

In our AbS reconstruction approach we make use of the different intensity scales of the first and second bounce components of the data. Our scene consists of a set of points or nodes in a 3D volume. The z coordinate is the depth coordinate describing the distance from the imaging system. In each step our algorithm makes a small random modification to the coordinates of one point. In practice we expect the second bounce response from a particular pair of patches to be much weaker than the first bounce return from either patch. Modifying a z coordinate will result in a large change to the associated time response while changes to x and y will only affect the second bounce returns riding on top of the large direct returns. We are therefore able to optimize the z coordinates of the nodes in a separate loop from the x and y coordinates treating the second bounce returns as background. Z coordinates are adjusted first and an optimum is found. x and y coordinates are adjusted in a second loop while z is held constant.

We currently focus on scenes with 3 to 5 points. In those cases the algorithm finds the correct geometric configuration only in a certain percentage of cases. We therefor re-run it several times with different initial coordinates for the points until the correct solution is found. While for 3 points the correct answer is usually obtained after less than 5 runs, scenes with 4 and 5 points can take significantly more runs to complete. A random update AbS algorithm like the one here is therefore useful as a first step and can be applied to reconstruct very simple scenes. Because of the unfavorable scaling behavior to larger sets of points, we need different alternatives for more realistic scenes.

Our algorithm is set up to test both the agreement between the reconstructed and correct point cloud and the agreement between their time responses. If we come across an example where the time responses agree perfectly, but there is a difference between the point clouds we can use it as a proof by example that TERA reconstructions are not possible for arbitrary scenes. So far our code has not identified such an example.

Better optimization

To be able to reconstruct more complex scenes we are exploring more sophisticated optimization algorithms. We are using the nonlocal optimization toolbox in Matlab to apply a variety of advanced methods such as genetic algorithms to our problem.

Constrained scene setup

A second method to improve reconstruction performance is to constrain the space of possible reconstructions. It is very unlikely that TERA reconstructions of complete partially translucent volumes

will ever be possible or required. Instead we aim to reconstruct scenes consisting of two dimensional surfaces inside of three dimensional volumes. This means that along any line parallel to the z-Axis there can be only one point with a non-zero reflectance (and absorbance). All points before that point are in free space, and all points behind are occluded by the reflecting point and generate no measurable scattering return. Making this assumption about the nature of our scene reduces the scene parameter space and significantly simplifies the reconstruction algorithm.

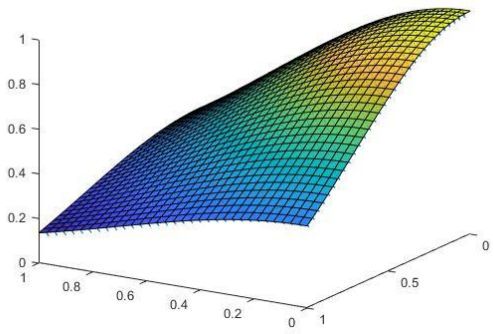
We implemented an optimization algorithm that assumes a set of surface patches that are arranged in a regular grid in x and y to form a continuous surface. The algorithm then varies the z-coordinate of every patch in an optimization loop. This algorithm works on larger grids than other presented approaches. We are currently experimenting with this method to obtain additional details.

Analysis by Synthesis Reconstruction and Proof of Invertability

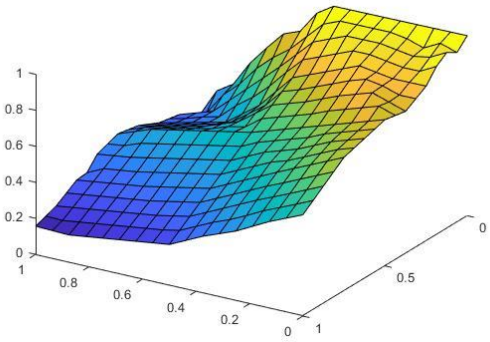
Our Analysis by synthesis work reliably for small scenes of less than 10 objects (or object patches). We have worked on a scale space optimization method to deal with the nonlinearity of the phase space we are reconstructing over. We were able to show that a traditional scale space approach is not applicable to our problem. We are able to show that reconstructions are always possible in principle in linear time for up to 5 objects by enumerating all possible configurations. Finally, we have identified research results from a related field that appear to prove that a reconstruction is possible for an arbitrary number of objects. We are currently waiting for publication of those results.

Constrained scene setup

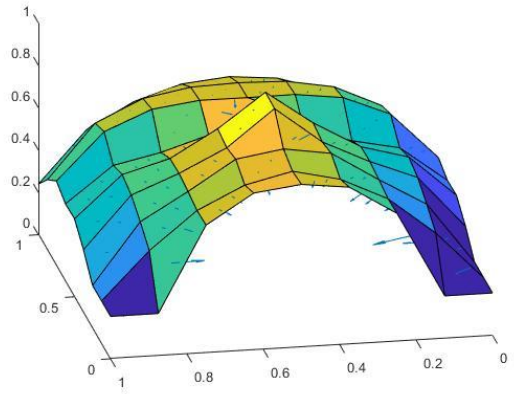
To simplify the Analysis by Synthesis reconstructions we have changed the parametrization of our scene. Rather than a set of objects in 3D space our constraint phase space reconstructs a surface in 3D space. Each surface patch is assigned a depth and the depths are adjusted. With this reduced parameter space we are able to reconstruct simple images with up to 32 by 32 pixels.



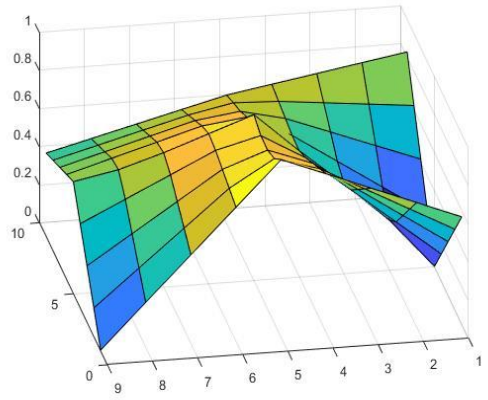
Input surface



Reconstruction



Input surface



Reconstruction

Reconstrctions of different surfaces using an Analysis by synthesis algorithm reconstructing a surface of 32 by 32 or 16 by 16 patches. The algorithm usually ends up in a local extremum by finds a shape that resembles the correct solution.

5. Point Cloud Reconstruction

This section describes a method to reconstruct geometries of general “sparse” scenes, i.e. scenes where individual geometry points (or patches) are fare apart from one another. As a result, each point will create a set of separate identifiable peaks in the data that can be extracted.

Consider an object emitting or reflecting light and an imaging system placed at a distance d from for the object. In order to infer the position of an object point O_i within the focal plane the imaging system redirects all rays from point O_i such that they constructively interfere on exactly one point in the focal plane. In other words the imaging system evaluates the length of a light ray as encoded in it’s phase. We call this phase as first order coherence. In this case, the resolution r of the imaging system is determined by the Rayleigh’s diffraction limit:

$r = 1.22\ \lambda\ d/D,$

where ”lambda” is wavelength of the emitted light, d is distance between focal plane of the imaging system and the object, and D is the diameter of the imaging system’s aperture. Another way of imaging is to measure the time of flight of intensity fluctuation We call these intensity fluctuation as second order coherence. One can use a short pulsed

laser to illuminate the object and lens-less time-of-flight detector to observe back-scattered light and reconstruct an image. The resolution of such an image is described by the Rayleigh criterion, except that the wavelength λ is replaced by the time resolution τ of the time-of-flight detector. We call it as a transient Rayleigh criterion:

$r_{trans} = 1.22\ \tau\ d/D.$

Theory

In this section, we apply results of [1], to show that when the measured time response of the scene contains sufficiently rich set of first and second bounces we can reconstruct the scene point cloud up to Euclidean congruence. Here we follow the same notation as [1]. Let p_1 ,.., pn ∈ R 3 be n point objects in the scene and p_0 ∈ R 3 be a position of the TERA imaging system. We call P = (p_1 ,.., p_n) as a scene configuration and S = (p_0, p_1,.., p_n) as a total configuration. The scene is illuminated with a pulsed laser and the detector observes returning signal from the scene. Let ping(first bounce) be a light that traveled $p_0 \rightarrow p_i \rightarrow p_0$ for any i = 1,..,n and let loop(second bounce) be a light that traveled $p_0 \rightarrow p_i \rightarrow p_j \rightarrow p_0$ for i 6= j.

Here we will make an assumption that we can detect the all signals. If we know the labeling of the signal, i.e. know from which target it came from, the reconstruction process is a well-known problem. However, we don't know the labeling of the signals, nor we know whether it is first bounce or second. Gkioulekas et al. showed that if the measurement contains sufficient amount of responses and the scene is a generic point configuration, then the point cloud is uniquely defined up to euclidean congruence.

Reconstruction algorithm

Armed with the theorem from previous chapter, we can design a reconstruction algorithm base on [1] with some modifications. The main difference from [1] is that we don't know whether the signal arose from singlebounce path or multi-bounce path. To address this problem, we exhaustively search over all possible combinations of single and multibounce path.

TRIBOND

Here we present modified TRIBOND algorithm. The algorithm consists of two parts: core finding and adding a vertex.

CORE FINDING

First step of the buildup algorithm is to find the core. The core of the embedding point cloud is overconstrained set of 5 points including the source. It can be broken down into 3 pieces: the base triangle, bottom tetrahedral and top tetrahedra. The base triangle is constructed using 2 first and 1 second bounces.

Bottom(blue) and top(red) tetrahedra each uses 1 first and 2 second bounces. Finally, one 2 bounce is used to test a bridge bond check between vertices's of two tetrahedra.

ADDING A VERTEX

After core is found, next step is to iteratively add vertex to the core. First, we choose any tetrahedra from the rigid substructure. The idea is to search over all possible combination of four distances from the remaining distance pool. Three distances will form one first and two second bounce distances. The remaining distance is used to test the bridge bond for the chosen rigid substructure.

MODIFIED TRIBOND

Algorithm Modified TRIBOND
Input: distance list from the timeresponse
1) Choose D=10 entries from the distance list
2) Search over all feasible four first and six second bounces distance combinations from the chosen D=10 entries
3) For each feasible combination to find core with 5 points including the system origin
Adding a vertex

1) Choose 4 points from the globally rigid substructure including the origin.
2) For all possible set of 4 distances from the list do: 2.1) pick 1 first bounce and 2 second bounces to construct tetrahera. 2.2) use remaining one second bounce to check the bridge bond
3) if bridge bond test fails, either choose other4 points from the globally rigid substructure or find another core.

LIGA

LIGA is an iterative stochastic algorithm based on a dynamic programming with backtracking. LIGA builds up a candidate structure by starting with a single vertex and adding additional vertices one at a time. The algorithm keeps a population of candidate structures at each size and uses promotion and relegation procedures to move toward higher quality structures.

Readers can find more details and the intuition behind the TRIBOND algorithm in [2] and the LIGA algorithm in [3].

MODIFIED LIGA

Algorithm Modified LIGA
Input: distance list from the time response
1) Start with a distance (from distance pool) and it's two vertices on the x-axis. Place one vertex at the coordinate origin.
2) For every sub-clusters do:
3) While distance pool is not empty do
4) PROMOTION PROCEDURE Apply line trial or planar trial or pyramid trial to add one or more vertices to currently chosen substructure. (Typical number of trials are n = 10000). Choose vertex with probability 1/cost.
5) RELEGATION PROCEDURE Choose a substructure with probability proportional to cost. Each vertex has cost equal to it's error contribution. Remove highest cost vertex and relegate the substructure.
6) End while loop
7) End for loop
8) Print the final cluster.

IV. Proof of concept simulations

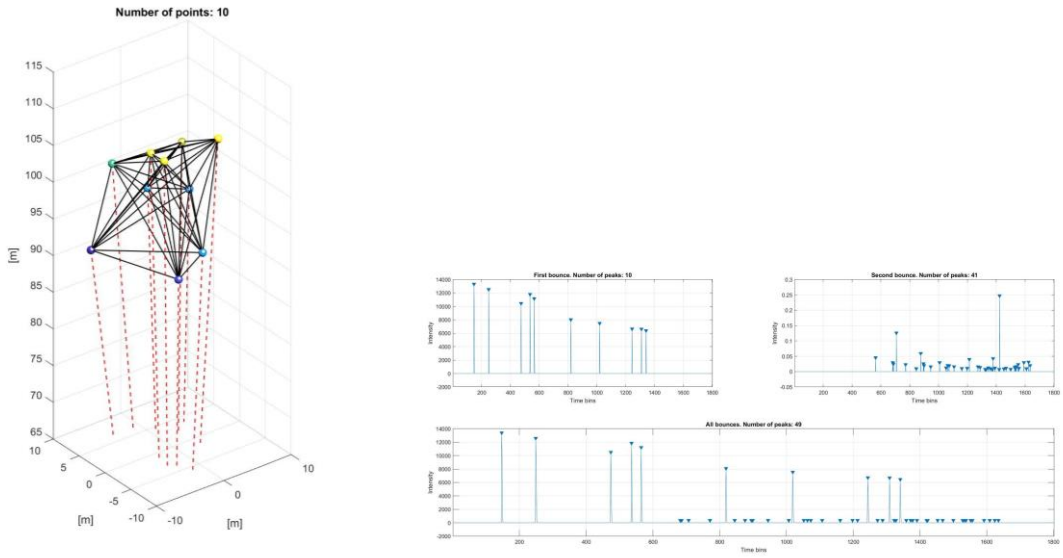
A. Forward model

To simulate the transient light transport measurements we use a renderer based on [4]. The renderer is successfully used in many time of flight applications([5]).

B. Simulation setup

To test the algorithm we generate simulated time responses. Again we consider following scenario. Let p_1 ,.., pn ∈ R 3 be n point objects in the scene and p_0 ∈ R 3 be a position of the TERA imaging system. Let di be the distance from pi to p0 for i = 1..n and di j be the distance from pi to pj for all i 6= j.

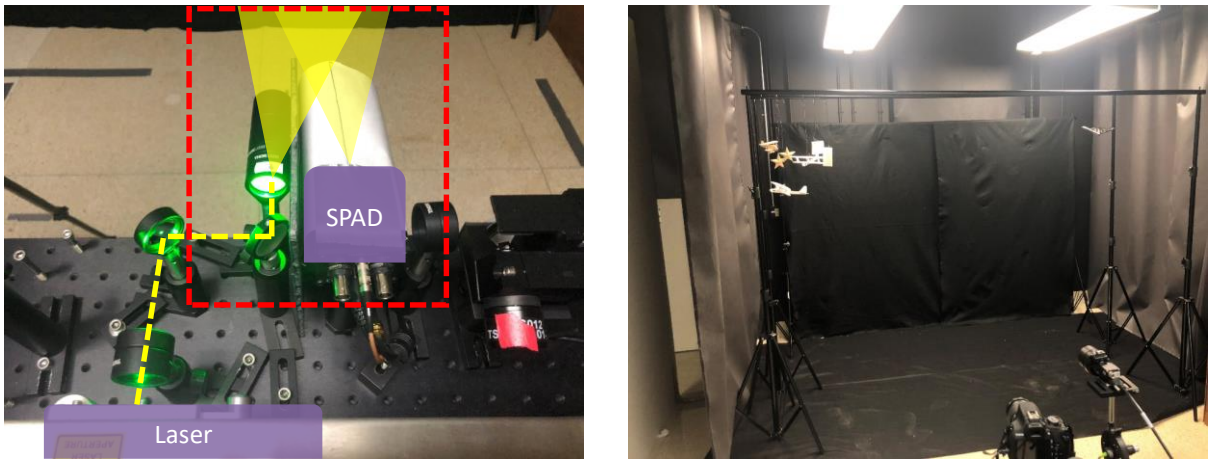
Let us assume that from the time response we acquired the time stamp list of photon arrival. We use this time stamp list as an input to the modified TRIBOND algorithm.



(left) Point cloud of 10 points along with reconstructions using the modified TRIBOND algorithm and simulated data. Simulated objects are 10 cm in diameter and spread over a space of 20 by 20 meters. (left) Simulated time response.

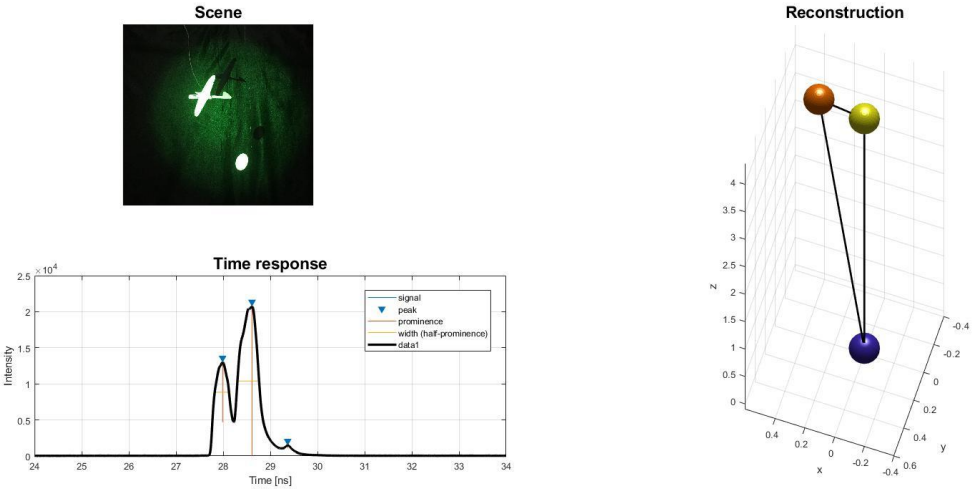
V. Proof of concept Experiments

In this section, we demonstrate the performance of TRIBOND and LIGA using experimental data. The experimental setup is shown in the figure below. The light source is ultrafast laser with 50[ps] pulse duration and 10MHz repetition rate. The wavelength is set to 532[nm]. The laser illuminates the scene which is roughly 2[m] away from the system. The scene contains Lambertian patches with sizes: 6x4[cm]

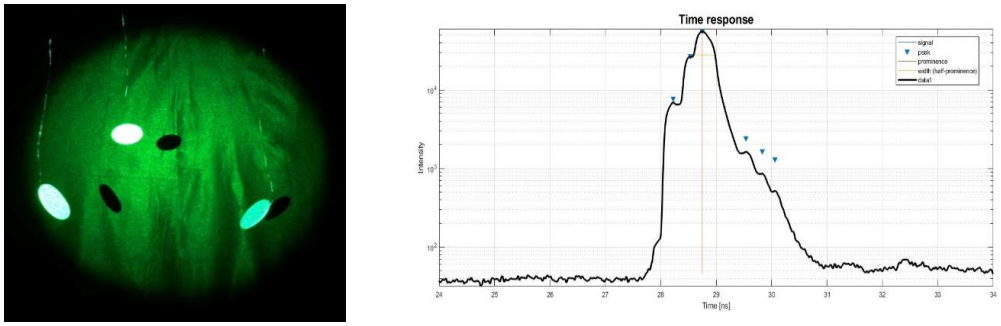


Setup of the TREA experiment.

and 3x2[cm]. The distance d between patches is varied. The returning light is collected by a SPAD, which has 20[mm] diameter active-area[paper]. At 532[nm], it has a photon detection efficiency of 30%. Note that we don't use any lenses in front of the SPAD. Figure displays measured time responses of the scene with different targets and varying distances.



Reconstruction of a 2 object real scene.



Reconstruction of a 3 Object Real Scene

Non-Line-of-Sight Imaging

While the main objective of TERA is to use multibounce light for super resolution, this study is closely related to the use of multibounce light for the reconstruction of scenes that cannot be seen directly and are only accessible via multibounce light. This field of Non-Line-of-Sight imaging shares many aspects with TERA. Both use similar detection hardware an light transport models. Our TERA models can also be used to reconstruct NLOS geometries and TERA methods are potentially useful to provide increased resolution NLOS reconstructions. Due to this strong overlap, our TERA research significantly contributed materially and intellectually to several ongoing NLOS imaging efforts in our group. Related manuscripts are listed below.

Products

Publications

We are currently in the process of creating a manuscript describing the point cloud reconstruction method described in this document. We are excited about the prospect of creating a new area of research based on this publication. Other publications are related to the presented work by using the same ideas for light transport analysis and sharing the same detection hardware as TERA:

Marco La Manna, Fiona Kine, Eric Breitbach, Jonathan Jackson, Talha Sultan, Andreas Velten,Error Backprojection Algorithms for Non-Line-of-Sight Imaging, IEEE Transactions on Pattern Analysis and Machine Intelligence
Preprint: https://minds.wisconsin.edu/bitstream/handle/1793/76968/TR1850.pdf?sequence=1

Xiaochun Liu, Ibon Guillen, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Diego Gutierrez, Adrian Jarabo, Andreas Velten. Virtual Wave Optics for Non-Line-of-Sight Imaging, Nature, Sent for Review by Editor

Preprint: https://arxiv.org/abs/1810.07535

Patents

No patents have been applied for.

References

[1] I. Gkioulekas, S. J. Gortler, L. Theran, and T. Zickler, “Determining generic point configurations from unlabeled path or loop lengths,” arXiv preprint arXiv:1709.03936, 2017.

[2] P. M. Duxbury, L. Granlund, S. [2] Gujarathi, P. Juhas, and S. J. Billinge, “The unassigned distance geometry problem,” Discrete Applied Mathematics, vol. 204, pp. 117–132, 2016.

[3] P. Juhas, L. Granlund, P. Duxbury, W. Punch, and S. Billinge, “The liga algorithm for ab initio determination of nanostructure, ”ActaCrystallographica Section A: Foundations of Crystallography, vol. 64, no. 6, pp. 631–640, 2008.

[4] A. Jarabo, J. Marco, A. Mu˜noz, R. Buisan, W. Jarosz, and D. Gutierrez, “A framework for transient rendering,” ACM Transactions on Graphics (SIGGRAPH Asia 2014), vol. 33, no. 6, 2014. C. Peters, J. Klein, M. B. Hullin, and R. Klein, “Solving trigonometric moment problems for fast transient imaging,” ACM Trans. Graph., vol. 34, no. 6, pp. 220:1– 220:11, Oct. 2015. [Online]. Available: http://doi.acm.org/10.1145/2816795.2818103