Non-Line-of-Sight Imaging using Data-Driven Approaches

by

Matthew Tancik

B.S., Massachusetts Institute of Technology (2016)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2018

© Massachusetts Institute of Technology 2018. All rights reserved.
Non-Line-of-Sight Imaging using Data-Driven Approaches

by

Matthew Tancik

Submitted to the Department of Electrical Engineering and Computer Science on January 16, 2018, in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science

Abstract

Non-line-of-sight (NLOS) imaging is desirable for its many potential applications such as detecting a vehicle occluded by a building’s corner or imaging through fog. Traditional NLOS imaging techniques solve an inverse problem and are limited by computational complexity and forward model accuracy. This thesis proposes the application of data-driven techniques to NLOS imaging to leverage the convolutional neural network’s ability to learn invariants to scene variations. We demonstrate the classification of an object hidden behind a scattering media along with the localization and classification of an object occluded by a corner. In addition we demonstrate the use of generative neural networks to construct images from viewpoints that extend the original camera’s field of view.

Thesis Supervisor: Ramesh Raskar
Title: Associate Professor, MIT Media Lab
Acknowledgments

I would like to thank my adviser Prof. Ramesh Raskar for welcoming me to the Camera Culture group and assisting me on the path that culminated to this thesis.

I would also like to thank Prof. Fredo Durand and Aude Oliva for offering me the opportunity to conduct research in the CSAIL Computer Vision group.

This work would not have been possible without the help and guidance from the students I worked with over the last two years.

Thank you to my mentor, Guy Satat, who initially introduced me to the field of academic research my senior year. You have taught me so much about the field of imaging and I feel privileged to have had the opportunity to collaborate on multiple projects.

Thank you to Zoya Bylinskii for providing mentorship this past year. You have helped my both academically and through the boundless source of connections you seem to have.

I would like to thank all of those whom I have had the chance to collaborate with including, Spandan Madan, Tristan Swedish, Barmak Heshmat, Otkrist Gupta, Sami Alsheikh, and Adria Recasens.

Finally, none of this would have been possible without my friends and family. I want to thank my mom for supporting me during every turn of this multi-year process.
Contents

1 Introduction ........................................ 17

2 Related Work ........................................ 19
   2.1 Imaging Beyond Line of Sight ..................... 19
      2.1.1 Time of Flight Methods ....................... 19
      2.1.2 Non-Time of Flight Methods ................. 20
   2.2 Data-Driven Approaches to Imaging ................. 20
      2.2.1 Synthetic Data Generation .................... 21
      2.2.2 Calibration Invariant Imaging ............... 21

3 Imaging Through Scattering Media .................. 23
   3.1 Overview ......................................... 23
   3.2 Imaging Pipeline .................................. 26
      3.2.1 Measurement System .......................... 26
      3.2.2 Forward Model - Synthetic Data Generation . 27
      3.2.3 Learning Calibration Invariant Sensing with CNN . 29
   3.3 Evaluation ........................................ 35
   3.4 Discussion ........................................ 38

4 Imaging Around Corners ......................... 43
   4.1 Overview ......................................... 44
      4.1.1 Data-Driven Approaches to Imaging ............ 46
   4.2 Localizing and Identifying NLOS Objects .......... 47
4.2.1 Rendering Synthetic Dataset ........................................ 48
4.2.2 Geometric preprocessing ............................................. 49
4.2.3 Localization and Identification Models ......................... 50
4.3 Implementation Details ................................................. 50
4.3.1 Experimental Results .............................................. 52
4.4 Generative Occluded Scene Reconstruction ......................... 54
4.5 Discussion .............................................................................................................................. 55
4.5.1 Areas That Contain Information In The Scene .................. 55
4.5.2 Illumination Position and Camera Field-of-View .................. 57
4.5.3 Generative Models Trained on Real Data ......................... 57
4.5.4 Main Limitations ...................................................... 58

5 Conclusion ............................................................................................................................... 61

Bibliography ................................................................................................................................... 62
List of Figures

1-1 Examples of imaging geometries. a) A traditional imaging geometry where the scattered light from the target travels directly to the camera.

b) A scattering geometry where the scattered light from the target undergoes scattering in a medium before reaching the target.

c) A geometry where the target object is occluded. The light must undergo additional scatters in order to reach the camera.

d) A geometry where the object is not in the field of view of the camera. The light must undergo additional scatters in order to reach the camera.

3-1 Calibration-invariant object classification through scattering. a) Training phase is an offline process in which the user defines random distributions of physical model parameters (based on approximate measurements or prior knowledge). The distributions are used to generate synthetic measurements with an MC forward model. The synthetic data is used to train a CNN for classification.

b) Once the CNN is trained the user can simply place the camera (demonstrated here with a time sensitive SPAD array) and an illumination source in the scene, capture measurements (six examples of time resolved frames are shown), and classify with the CNN without having to precisely calibrate the system.
Comparison of SPAD measurement and forward model. The targets are two poses of a mannequin placed behind a paper sheet (diffuser). The data shows six frames (each frame is $32 \times 32$ pixels) of raw SPAD measurements, examples of two synthetic results generated by the MC forward model with similar measurement quality, and a synthetic result with high photon count and no additive noise. Note that differences between synthetic ex. 1, 2 and the raw measurement are due to the fact that the forward model was never calibrated to this specific setup. The synthetic images represent different instances chosen randomly from the dataset. The synthetic example with high photon count helps to distinguish between measurement (or simulated) noise and the actual signal as well as to observe the full signal wavefront.

3-3 CNN learns to be invariant to model parameters. The CNN is trained with the complete random training set (based on the MNIST dataset), and evaluated with test sets in which all model parameters are fixed except for one that is randomly sampled from distributions with growing variance. Three parameters are demonstrated (other parameters show similar behavior). a) Diffuser scattering profile variance $D_D \sim N(0, \sigma)$, $\sigma \sim U(1 - \alpha, 1 + \alpha)$ radians, b) Camera field of view $C_{FV} \sim U(0.15 - \alpha, 0.15 + \alpha)$ radians, and c) Illumination source position $L_P \sim U(-\alpha, \alpha)$ cm. The top plots shows the classification accuracy as a function of the parameter distribution variance in the test set. Red lines show the ranges used for training. The `X` marks point to specific locations sampled for PCA projections in the bottom part of the figure. PCA projections show a color map where each digit has different color. Performance is maintained beyond the training range and starts to slowly degrade further from it, as can be observed in PCA projection III where more mixing is apparent at a test range $\times 2.5$ larger compared to the training set.
3-4 Successful classification of handwritten digits behind scattering. The ‘0’ and ‘1’ digits from the MNIST dataset are cut out from cardboard and placed behind a paper sheet. Raw SPAD measurements are input into the CNN and correctly classified.

3-5 Lab experiments show successful estimation of hidden human pose. a) Three examples (rows) demonstrate target pose, raw SPAD measurement (first six frames), and the successful classification. b) Confusion matrix for classification of raw test set (10 samples per pose).

3-6 Classification among seven poses on synthetic test dataset. a) t-SNE visualization demonstrates the CNN ability to classify among the seven poses. b) Confusion matrix for classification on the synthetic test dataset.

3-7 Performance of the K-nearest neighbor approach on the clean dataset. Classification accuracy with varying dictionary size for a) nearest neighbor classifier, and b) K-nearest neighbors classifier.

3-8 Time resolution is more important than number of pixels for imaging through scattering. a) Classification accuracy vs. time resolution (for 32x32 pixels). b) Classification accuracy vs. number of pixels (for non time-resolved system).

3-9 Examples of spatio-temporal filters learned by the CNN. The network generates both a) spatial and b) temporal filters for inference.
Can we create an image as if seen from the '?' using self contained consumer flash photography? a) A target is occluded from the camera. The scene is illuminated by a flashlight, light scatters off of the wall to the occluded object, floor, and to the camera. The measured signal changes depending on the class and location of the hidden target. b) The Measurement is processed and fed into a CNN for inference. The CNN is trained using rendered data. c) The system localizes, identifies and tracks the occluded object in real time. d) A generative model reconstructs a photo-realistic rendering of the hidden scene from the desired virtual camera’s point of view.

Geometric preprocessing. a) Photograph from camera before processing. Purple dotted line outlines the floor area to be rectified. The orange circle indicates the location where the corner wall meets the floor. b) The photograph is first rectified and cropped (left column, grayscale). Then, it is background subtracted and scaled to $[-1,1]$ (right column, color). This is the input to the CNN. c) Two examples taken from the rendered dataset. The background subtracted examples demonstrate the light casted by the object which spreads throughout the measured area, not just the edge discontinuity.

Network architectures for localization, identification, and generative model. The input to all networks is $64 \times 64$ pixels scaled to $[-1,1]$. Convolution layers (yellow squares) are of size $5 \times 5$ with a stride of 1. Pooling layers (green arrows) are of size $2 \times 2$. A RELU is performed after each pooling layer and before each fully connected layer (blue square). a) The localization network has two outputs. b) The identification network uses softmax for classification. c) The generative model has a VAE architecture where the decoder is a series of fully connected layers. The embedding vector is composed of the mean (length 100) and variance (length 100) of a probability distribution, the latent vector is of length 20.
4-4 Test scene. a) Dimensions of scene from an above perspective. b) Dimensions of scene from a side perspective. c) Dimensions of the three target objects tested. d) Photographs of the scene. Arrows in panel a point to the two perspectives.

4-5 Occluded object localization and tracking in 2D. a) Localization accuracy on a grid. Three objects of different shapes are moved along a $5 \times 5$ grid of size $30 \times 30$ cm$^2$. The center of each grid point is marked by a colored plus sign. Each object and each grid point are measured 30 times, during which the object is perturbed (90 data points per grid point, 2250 data points overall). Localizations are marked by colored dots. The color of each dot corresponds to the associated ground truth grid position, such that correct localizations have the same color as the closest ground truth plus sign. As the object is further away, the localization accuracy is reduced. The inset shows the hallway geometry, with the localizations superimposed and an example of one of the targets for scale reference. b) Tracking a hidden object. In these examples an object is tracing an infinity sign and circular path twice. Each frame in the camera video stream is processed and used for localization independently. Color along the path represents the time axis.

4-6 Hidden Object identification. a) A confusion matrix showing the identification accuracies among the three classes on real data. The reported accuracies are averages over all locations. b) Identification accuracies on different grid positions.

4-7 Visualizing the occluded scene with a generative model. a) Photograph of the scene taken from the desired point-of-view, used for reference. b) Reconstruction results using a graphics program. The scene parameters are first estimated and then are used to render the image. c) Reconstruction results with a generative model. Different columns show results for different objects and positions from the test dataset.
4-8 Areas encoding information about the hidden scene. Brighter colors represent areas of higher sensitivity for the network in a) localization and b) identification. c) Comparing the importance of the area between the wall and corner (marked as 1 in the inset), and the area between the corner and the camera (marked as 2 in the inset). The table shows the localization and identification accuracies when only area 1 or 2 are available for the network. For localization both areas are equally important and using both significantly improves performance. For identification using just area 2 result in random prediction, and area 1 is essential for prediction.

4-9 Generative reconstruction results on trained real data. a) Scene photograph, showing the input camera (observing the floor) and the occluded scene. b) Input camera measurements. c) Ground truth measurements from the output camera (same camera used for training). d) Generative reconstruction, taking the background subtracted measurement as input and generate the scene from the desired point-of-view.
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Distributions for calibration and target parameters used in mannequin dataset.</td>
<td>27</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of different approaches on classification of the clean and realistic datasets. The CNN outperforms all methods in the clean dataset, and is the only method that achieves results that are better than random accuracy on the realistic dataset.</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of NLOS imaging techniques with a consumer camera.</td>
<td>45</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Imaging is the act of forming a representation of an object or scene. A familiar form of imaging is photography, such as from a mobile phone, where the goal is form a 2D representation of a scene using the visible wavelengths captured by an optical setup. This thesis discusses 2D imaging, but focuses more generally on imaging, where the goal is to extract any data from the scene. For example this could include determining the type or location of an object. This simplicity of the information can be meaningful in many applications; for example, an autonomous vehicle is more concerned with being able to localize and identify a pedestrian over visualizing the pedestrian’s outfit and hairstyle.

Non-line-of-sight (NLOS) imaging describes a particular type of imaging where the target object cannot be directly imaged by the camera. In a traditional image, photons are emitted into a scene (i.e. from the sun, indoor lighting, camera flash) and scatter off of the target. Some of the scattered photons then travel directly into the camera optics. In the NLOS situation, the photons that scatter off of the target undergo additional scattering events before reaching the camera optics, as shown in Fig. [1-1]. This thesis will consider multiple NLOS geometries; a geometry that contains a scattering medium such as an optical diffuser or fog, a geometry where the target object is occluded, and finally a geometry where the target is not in the field of view (FOV) of the camera. In each of these geometries, the information is obscured before reaching the camera. We formulate a data-driven approach to disentangle the
Figure 1-1: Examples of imaging geometries.  a) A traditional imaging geometry where the scattered light from the target travels directly to the camera b) A scattering geometry where the scattered light from the target undergoes scattering in a medium before reaching the target.  c) A geometry where the target object is occluded. The light must undergo additional scatters in order to reach the camera. d) A geometry where the object is not in the field of view of the camera. The light must undergo additional scatters in order to reach the camera.

introduced ambiguity and recover information about the scene. These methods have potential applications in many fields such as seeing around corners for self driving cars, search and rescue, human computer interaction systems, and imaging through fog.

Chapter 2 describes related work as it pertains to NLOS imaging and data driven techniques.

Chapter 3 presents a method for imaging through scattering. Using a time of flight camera and a data-driven approach we demonstrate pose classification of a mannequin occluded by a sheet of plain printer paper.

Chapter 4 presents a method for imaging around a corner. Using synthetically generated data, a model is trained to predict the identity and location of an object occluded by a corner. In addition, a separate model is trained to produce a 2D representation from a virtual camera which has the target object in the FOV.

Finally, chapter 5 summarizes the contributions of this thesis.
Chapter 2

Related Work

In this chapter, we discuss existing research relating to NLOS imaging (section 2.1) and data-driven approaches for imaging (section 2.2).

2.1 Imaging Beyond Line of Sight

NLOS imaging has been of interest to researchers for decades and has been explored widely in the context of vision related applications such as: image dehazing [1, 2, 3, 4], seeing through translucent objects [5], cloud tomography [6, 7], underwater imaging [8], tracking [9, 10] and recovery of material scattering properties [11]. It has also been discussed in the context of structured light [12].

Many active methods have been demonstrated in the context of imaging through scattering such as optical coherence tomography [13], wavefront shaping [14, 15, 16, 17], speckle correlations [18, 19], acousto-optic [20] and photo-acoustic [21, 22]

2.1.1 Time of Flight Methods

A Time of flight (ToF) camera is able to the time of arrival of photons. If the light source is pulsed or coded, the differences in arrival times correspond to differences in photon path length. This is because $\Delta d = c\Delta t$ where $\Delta d$ is the difference in distance, $c$ is the speed of light, and $\Delta t$ is the difference in time. Knowledge about the path
lengths can be used to infer information about scattering events and the geometry in the scene.

In the context of wide field NLOS imaging, ToF setups with either impulse \cite{23, 24} or phase \cite{25, 26, 27} based systems have been used. Different aspects of imaging have been demonstrated like full scene reconstruction \cite{28}, pose estimation \cite{29}, tracking \cite{9}, and medical applications such as seeing through tissue \cite{30}. NLOS imaging has been demonstrated with impulse based systems like streak cameras \cite{24} and single photon avalanche diodes (SPAD) \cite{31, 32}. A different approach uses only first-returning photons to recover geometry \cite{33}. In chapter \ref{chap:methods} we use a SPAD array with a pulsed laser in the imaging setup.

\section{2.1.2 Non-Time of Flight Methods}

Recently, regular cameras have been used for NLOS imaging. An object placed within a lit scene will change the way the light bounces within the scene. These changes are often imperceptible to the human eye, but are within the dynamic range of the camera. By modeling or understanding how an object will change the lighting in a scene, it may be possible to infer properties about the object.

A method by Klein et al. \cite{34} uses a laser pointer and a regular camera to track objects around a corner. A passive method (relies on ambient light) by Bouman et al. \cite{35} is able to perform one dimensional tracking of objects around a corner. In chapter \ref{chap:methods} we describe a method to track and identify an object occluded by a corner using a regular camera. We also describe a method to generate images from a virtual viewpoint using data captured by regular cameras.

\section{2.2 Data-Driven Approaches to Imaging}

To tackle the problem of calibration-invariant imaging we use a convolutional neural network (CNN). Recently, CNNs have become the main workhorse in many computer vision tasks. CNNs are especially appealing for our application due to their ability to capture invariants \cite{36, 37}, reduce dimensionality from noisy data \cite{38}, and classify
objects [39]. Initial uses of data driven approaches for imaging problems have been suggested in microscopy [40], compressive imaging [41], synthetic aperture radar [42], remote sensing [43, 44], dehazing [2], phase imaging [45], medical imaging [46], and classification with coherent light [47, 48]. In our case, the CNN was trained with synthesized data that includes variations in calibration parameters. By training the CNN with synthetic and diverse data, the network learns a model that is not only invariant under traditional transformations like translation, but also invariant to changes in calibration parameters within the training range, and nearly invariant beyond that range. Thus, we eliminate the need to precisely calibrate a computational model for NLOS object classification.

Data-driven techniques have also been applied to problems in the computational imaging community such as phase imaging [45], compressive imaging [49], tomography [50], microscopy [51], imaging [52], and imaging through scattering media [53]. Localization and identification of people around a corner has also been demonstrated with a data-driven approach based on active pulsed illumination and a time-resolved single photon avalanche diode camera [54].

2.2.1 Synthetic Data Generation

Using synthetic data to train CNN models is becoming more common in computer vision applications. Rendered data provides multiple advantages such as the ability to generate large datasets and the ability to introduce defined variations into the data. Rendered data has been used in computer vision applications such as optical flow [55], action recognition [56], text recognition [57], and tracking [58].

2.2.2 Calibration Invariant Imaging

Calibrating system parameters is required in many vision [59] and imaging systems. Having a calibration-free system relaxes many requirements on system design and usage scenarios, however it is usually very challenging. Some examples of calibration-free systems have been demonstrated in specific domains such as: augmented real-
ity [60, 61] and gaze tracking [62].

The requirement for calibration when performing NLOS imaging is directly related to the need of a physical model that explains the measurements. Such models simulate light transport and depend on the geometry and other physical parameters of the system. Since inverting NLOS geometries is an ill-posed problem, any mismatch between the physical model and the actual measurement will degrade performance. As a result, accurate calibration of imaging system parameters like illumination position, camera orientation etc. is needed. This limits many inversion based techniques to scale to real-world applications.
Chapter 3

Imaging Through Scattering Media

In this chapter we demonstrate an imaging technique that allows identification and classification of objects hidden behind scattering media and is invariant to changes in calibration parameters within a training range. The work presented in this chapter was completed in collaboration with Guy Satat, Otkrist Gupta, Barmak Heshmat, and Ramesh Raskar. The work was published under the title, *Object Classification through Scattering Media with Deep Learning on Time Resolved Measurement* in Optics Express (2017) [53].

In section 3.1 we provide an overview of the task. In section 3.2 we propose an imaging pipeline for human pose classification of a mannequin occluded by a sheet of paper. In section 3.3 we evaluate our method to other data driven methods. Finally, we conclude with a discussion about real-world applications and the advantages of time resolution in section 3.4.

### 3.1 Overview

Traditional techniques to image through scattering solve an inverse problem and are limited by the need to tune a forward model with multiple calibration parameters (like camera field of view, illumination position etc.). Instead of tuning a forward model and directly inverting the optical scattering, we use a data driven approach. With a Monte Carlo (MC) model, we synthesize a large dataset that contains random
Figure 3-1: Calibration-invariant object classification through scattering. a) Training phase is an offline process in which the user defines random distributions of physical model parameters (based on approximate measurements or prior knowledge). The distributions are used to generate synthetic measurements with an MC forward model. The synthetic data is used to train a CNN for classification. b) Once the CNN is trained the user can simply place the camera (demonstrated here with a time sensitive SPAD array) and an illumination source in the scene, capture measurements (six examples of time resolved frames are shown), and classify with the CNN without having to precisely calibrate the system.

realizations of all model parameters. This allows us to use a deep neural network for classification of objects that are hidden from the camera. We show that the network learns to be invariant to changes in these physical parameters, which effectively allows calibration-free imaging through scattering conditions. The method is evaluated with a time-resolved camera and multiple experimental results are provided including pose estimation of a mannequin hidden behind a paper sheet with 76.6% accuracy on real-world measurements.

The imaging procedure is shown in Fig. 3-1 and is partitioned into two halves. First (offline process) a Monte Carlo (MC) model is used to synthesize a large training dataset of potential measurements drawn from the distribution of all target variations and calibration parameters. The synthesized dataset is used to train a CNN. The resulting CNN is invariant to changes in calibration parameters within the training range, effectively allowing calibration-invariant object classification through scattering. This enables the second (online phase) in which the user can simply place the camera and illumination without calibration and classify, in real time, hidden objects.
Figure 3-2: Comparison of SPAD measurement and forward model. The targets are two poses of a mannequin placed behind a paper sheet (diffuser). The data shows six frames (each frame is $32 \times 32$ pixels) of raw SPAD measurements, examples of two synthetic results generated by the MC forward model with similar measurement quality, and a synthetic result with high photon count and no additive noise. Note that differences between synthetic ex. 1, 2 and the raw measurement are due to the fact that the forward model was never calibrated to this specific setup. The synthetic images represent different instances chosen randomly from the dataset. The synthetic example with high photon count helps to distinguish between measurement (or simulated) noise and the actual signal as well as to observe the full signal wavefront.
behind scattering media. We experimentally demonstrate this method with human pose estimation behind a scattering layer. Pose estimation behind scattering has many important applications in areas such as: privacy preserving human computer interaction systems (e.g. human pose estimation without access to face or body images) as well as search and rescue missions.

The key contributions of our approach are:

1. A measurement-independent training method that uses only synthetic data (based on a Monte Carlo renderer) to train a CNN; before acquiring any actual measurements.

2. A technique for imaging through scattering that is invariant to variations in system calibration parameters within the training range.

3. The technique allows real-time classification through scattering medium and beyond line of sight.

### 3.2 Imaging Pipeline

#### 3.2.1 Measurement System

The optical setup is shown in Fig. 3-1. A pulsed source (NKT photonics SuperK) with a repetition rate of 80 MHz and pulse duration of 5 ps is spectrally filtered to a band of 580 ± 10 nm. The camera is a single photon avalanche diode (SPAD) array (Photon Force PF32) with 32 × 32 pixels, and a time resolution of 56 ps. The laser is incident on the diffuser at ∼ 45°. The camera is focused on the diffuser (regular paper sheet which presents non-uniform scattering properties). A flexible mannequin is placed behind the diffuser (20 cm head to toe). A black screen separates the camera from the incident position of the laser on the diffuser (to prevent direct reflection from the diffuser to the camera). The optical setup demonstrates a reflection mode geometry. The first 64 time bins of the SPAD measurement are used, such that the data structure is of size 32 × 32 × 64 (the large number of frames guarantees consistency and flexibility
Table 3.1: Distributions for calibration and target parameters used in mannequin dataset.

<table>
<thead>
<tr>
<th>Calibration parameters</th>
<th>Diffuser</th>
<th>Camera</th>
<th>Noise</th>
<th>Target parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser</td>
<td>Diffuser</td>
<td>Camera</td>
<td>Noise</td>
<td>Target parameters</td>
</tr>
<tr>
<td>- Incident position</td>
<td>- Scattering profile</td>
<td>- Position</td>
<td>- Dark count</td>
<td>- Position</td>
</tr>
<tr>
<td>$L_p \sim U(-4,4)cm$</td>
<td>$D_d \sim N(0, \sigma), \sigma \sim U(0.8,1.2)rad$</td>
<td>$C_p \sim U(-1.5,1.5)cm$</td>
<td>$N_{DC} \sim U(3000,9000)$ photons</td>
<td>$T_{p_{x,y,z}} \sim U(-4,4)cm$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time resolution</td>
<td></td>
<td>- Scale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_{TR} \sim N(0, \sigma), \sigma \sim 56 + U(-5,5)ps$</td>
<td></td>
<td>$T_{s} \sim U(18,30)cm$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time jitter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_{TS} \sim U(0,3*56)ps$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Field of view</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_{FV} \sim U(0.1,0.2)rad$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Homography</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Normal distributions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

of the data structure). Several examples of the measurement frames are provided in Fig. 3-2.

3.2.2 Forward Model - Synthetic Data Generation

The proposed method is based on an MC model that renders the SPAD measurements. Since SPAD captures single photon events it fits well to an MC model that traces individual photons. MC is a very generic forward modeling technique that can be easily modified to simulate various system geometries like looking around corners and seeing through a scattering medium (which is the focus of this work). This is accomplished by modeling a wide range of physical parameters which can be broadly divided into target and calibration related.

The MC forward model is used to generate a dataset with random realizations
of potential measurements for different target instances and calibration parameters. For each new simulated data point, the above parameters are randomly sampled from their given distributions to define a specific system, geometry and target (Table 3.1 provides the list of parameters and distributions used here). A ray tracer is used to simulate propagation of individual photons from the illumination source, through the diffuser, onto the target, back to the diffuser and finally into the camera (see Algorithm 1). This process takes into account the propagation time. SPAD array measurements are only based on thousands of detected photons. Since there is no need to render the full time-dependent scene irradiance, the computational burden of ray tracing with MC is low (we simulate \(10^6\) photons for each data point, which takes \(\sim1\) second on a regular desktop computer). Fig. 3-2 compares raw measurements taken with the SPAD camera and instances of the forward model (chosen randomly from the synthetic dataset).

We note that while paper sheet is a strongly scattering media with multiple scattering events, it can be modeled as a single scatter event due to: 1) The propagation time through the paper \((\sim10\text{ps})\) [63] is much smaller compared to the time resolution of the SPAD camera. 2) The scene size (target feature size and scene length scales) are much larger compared to the scatterer thickness, so we can approximate the photon exit coordinate to be equal to the entrance coordinate. In cases where these assumptions don’t hold, it is easy to add a random walk simulation to the Monte Carlo renderer that would simulate the scattering process in the material (including the time and location dependency).

Each data point in the dataset corresponds to a specific example of a target measured by a system that is defined by a set of random target and calibration parameters (see table 3.1):

- The target is defined by a label and an instance (for example a specific appearance of a digit in a handwritten digits dataset), these are simply selected from the dataset. The dataset may or may not include variations in parameters such as scale and orientation. For improved robustness it is preferred to add variability in all parameters, this is achieved by scaling the target with parameters
that are drawn from distributions of plausible target size. Finally, the target is placed at a random 3D location behind the diffuser, the location is sampled from a uniform distribution which defines the non-line-of-sight volume of interest.

• The imaging system is defined by a realization of various calibration parameters that are sampled from random distributions. User input is involved only in determining the random distributions, which are defined based on approximate measurements, for example observation of the system geometry by the naked eye. If a parameter is easy to evaluate (for example the laser position on the diffuser) it can be modeled with a Gaussian distribution with the known mean and small variance. Or if it is hard to evaluate, it can be modeled with a uniform distribution.

Varying calibration parameters in the training data allows the CNN to be invariant to changes in those parameters within the training range (see section 3.2.3).

3.2.3 Learning Calibration Invariant Sensing with CNN

The synthetic random dataset generated with the MC forward model is used to train a CNN for classification of hidden objects behind a diffuser. CNNs are a natural fit for this task since: 1) they have been shown to perform well in classification tasks, 2) they are designed to be invariant to translations, and 3) learn to be invariant to other data transformations like scaling, rotation and, as demonstrated here, variations in the system calibration parameters.

Several neural network architectures were considered. The data structure in our case is composed of several frames, which is similar to the case of action recognition and gesture classification from short videos. Works such as [64, 65] indicated that convolutional architectures produce robust classification in that task. Thus, multiple convolutional architectures were evaluated including VGG [66], ResNet [67], and several custom shallower networks with various combinations of layers. All architectures performed similarly on the classification task with marginally better performance for VGG. The VGG topology was selected and modified by extension of convolution fil-
Figure 3-3: CNN learns to be invariant to model parameters. The CNN is trained with the complete random training set (based on the MNIST dataset), and evaluated with test sets in which all model parameters are fixed except for one that is randomly sampled from distributions with growing variance. Three parameters are demonstrated (other parameters show similar behavior). a) Diffuser scattering profile variance $D_D \sim N(0, \sigma)$, $\sigma \sim U(1 - \alpha, 1 + \alpha)$ radians, b) Camera field of view $C_{FV} \sim U(0.15 - \alpha, 0.15 + \alpha)$ radians, and c) Illumination source position $L_P \sim U(-\alpha, \alpha)$ cm. The top plots show the classification accuracy as a function of the parameter distribution variance in the test set. Red lines show the ranges used for training. The 'X' marks point to specific locations sampled for PCA projections in the bottom part of the figure. PCA projections show a color map where each digit has different color. Performance is maintained beyond the training range and starts to slowly degrade further from it, as can be observed in PCA projection III where more mixing is apparent at a test range $\times 2.5$ larger compared to the training set.
Algorithm 1 MC Forward Model

1: Initialize scene by randomly sampling:
2: Target: label, instance, position, size
3: Laser: incident position
4: Diffuser: scattering profile
5: Camera: position, time resolution, time jitter, field of view, homography parameters
6: for All photons do
7: Calculate initial intersection point with diffuser
8: Randomly sample diffuser local scattering profile
9: Randomly sample photon’s angle after diffuser
10: Calculate photon’s intersection point with target
11: if does not hit target then
12: continue to next photon
13: end if
14: Randomly sample angle after reflection from target
15: Calculate photon’s intersection point with diffuser
16: if does not hit diffuser then
17: continue to next photon
18: end if
19: Randomly sample diffuser local scattering profile
20: Randomly sample photon’s angle after diffuser
21: Map photon to camera sensor using homography
22: Randomly sample photon’s arrival time jitter
23: Store photon’s arrival time (with jitter) and location
24: end for
25: Bin recorded photons into discrete time frames.
26: Add dark count noise to measurement

 ters into time domain (3D space-time filters). Filters were resized to $3 \times 3 \times 10$ where the last index denotes the time dimension (see further details in section 3.4). The training time on 60,000 data points is approximately two hours on an Nvidia Titan XP GPU.

To evaluate our approach, we used the well-known MNIST dataset of handwritten digits. The goal is to evaluate the CNN ability to classify hidden objects while being invariant to changes in calibration parameters. To that end, 60,000 training samples and 10,000 test samples are synthesized with the MC forward model. Each data point is a realization of a different set of target and calibration parameters. The result on the test set is an overall classification accuracy of 74% (compared to 10%
Figure 3-4: Successful classification of handwritten digits behind scattering. The ‘0’ and ‘1’ digits from the MNIST dataset are cut out from cardboard and placed behind a paper sheet. Raw SPAD measurements are input into the CNN and correctly classified.

random guess accuracy). These simulations demonstrate the ability to classify objects hidden behind a scattering layer without calibration. As a proof of concept lab experiment, we cut two targets from cardboard shaped like zero and one digits, placed them behind a paper sheet, and measured the response with the SPAD camera. The two time resolved measurements were correctly classified as zero and one using the above network. The training dataset generation and network training were performed prior to this data acquisition. This demonstrates that our method is robust to variations in calibration parameters on raw data. Section 3.2.3 provides more challenging experimental results. As a proof of concept experiment, we cut the zero and one digits from cardboard and placed them behind a paper sheet. The network described above correctly classified the digits. Fig. 3-4 shows samples of two digits with the corresponding raw SPAD measurement and successful classification with the above network. The forward model and network training were performed prior to this data acquisition, which demonstrates calibration-invariant classification of hidden objects with raw data (the next section provides further experimental results).

In order to evaluate the extent of the network’s ability to handle changes in calibration parameters a set of controlled synthetic experiments were performed. We used the trained network with the MNIST dataset, and tested it with multiple test sets that were generated for the purpose of this evaluation. In each test set, all calibration parameters are held fixed (on the mean), except for one parameter that is randomly sampled from distributions with different variances. Thus, the CNN’s
sensitivity to variations in different parameters is probed independently. Specifically, for each calibration parameter to be investigated, multiple test sets are generated, each one with a different distribution variance. The variance is scanned starting from zero (i.e. just the mean) throughout the range that was used for training and then continues to grow beyond the training range up to at least $\times 2.5$ of the training range. Figure 3-3 demonstrates results for three calibration parameters (other parameters demonstrate similar behavior). As can be seen from the test accuracies, performance is maintained within the variance range used for training, and extended well beyond that range. This demonstrates the network ability to learn an invariant model to changes in the calibration parameters within the training range and nearly invariant beyond that range. For example, in Fig. 3-3(c) the network was trained with data that had the illumination position distributed uniformly within 5 cm from the mean. Yet, the test performance starts to slightly drop only after the illumination position may be found within 10 cm of the mean. Qualitative evaluation of these results are also presented in the bottom part of Fig. 3-3 with PCA projections of the activations from the penultimate layer of the CNN, these demonstrate sustained performance well beyond the training range.

This analysis shows that the network performance is maintained when the calibration parameters deviate from the mean within the training range. Furthermore, even if the network was trained under an assumption of certain ranges for system parameters, the performance degrades very slowly if the actual calibration parameters are outside the training range.

In order to demonstrate human pose estimation behind scattering medium, a flexible mannequin (length from head to toe $\sim 20$ cm) is placed behind a regular paper sheet (Fig. 3-1). We define three different poses for the mannequin using various positions of hands and legs (Fig. 3-5).

CNN training is accomplished by synthesizing 24,000 samples for training and 6,000 samples for validation. Translations and perturbations to the mannequin’s head and limbs are applied to create multiple instances of each pose. The test set is composed of 30 raw SPAD measurements, 10 per pose. For each measurement,
Figure 3-5: Lab experiments show successful estimation of hidden human pose. a) Three examples (rows) demonstrate target pose, raw SPAD measurement (first six frames), and the successful classification. b) Confusion matrix for classification of raw test set (10 samples per pose).

Figure 3-6: Classification among seven poses on synthetic test dataset. a) t-SNE visualization demonstrates the CNN ability to classify among the seven poses. b) Confusion matrix for classification on the synthetic test dataset.
the mannequin is moved around and the position of the hands, legs and head are adjusted. These raw measurements achieve 76.6% overall accuracy (compared to 33.3% random guess accuracy). Figure 3-5(a) shows examples of mannequin pose, SPAD measurements, and classification. Figure 3-5(b) shows the confusion matrix of this raw test set.

Here training is performed on one dataset (synthetic) and tested on another dataset (gathered by lab experiments). In general, it is challenging to train and test on different datasets and it is common to note performance degradation in such cases. The degradation in performance can potentially be mitigated with domain adaptation methods (e.g. [68]), we leave this to a future study.

To further explore the sensitivity to the number of poses we expanded the training set to include seven different poses (Fig. 3-6 bottom shows illustrations of the poses). The poses include a diverse combination of limb positions. For each label 8,000 training examples and 2,000 test examples were generated (total training set of 56,000 examples and 14,000 test set examples). Figure 3-6(a) shows a two dimensional student’s t-distributed stochastic neighbor embedding (t-SNE) [69] visualization of activations from the CNN penultimate layer generated on the test set. This visualization demonstrates that the network correctly separates the classes. Figure 3-6(b) shows the confusion matrix for this synthetic test set. The network is able to classify the seven classes with 91.86% accuracy. The synthetic test accuracy for the network trained only on the three poses (Fig. 3-5) achieved 96.7%. This indicates the ability to experimentally classify among more poses without significant decrease in accuracy.

3.3 Evaluation

To evaluate our approach we compare its classification performance to several other classification techniques. The classification task is based on the three mannequin poses. We create two datasets for evaluation, each one consists of 24,000 training examples and 6,000 test examples. The clean dataset demonstrates the algorithms’ sensitivity just to variation in calibration parameters (decoupling the sensitivity to
Table 3.2: Comparison of different approaches on classification of the clean and realistic datasets. The CNN outperforms all methods in the clean dataset, and is the only method that achieves results that are better than random accuracy on the realistic dataset.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Clean dataset</th>
<th>Realistic dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Example</td>
<td>33.3</td>
<td>33.3</td>
</tr>
<tr>
<td>KNN</td>
<td>53.0</td>
<td>30.0</td>
</tr>
<tr>
<td>SVM</td>
<td>57.1</td>
<td>20.0</td>
</tr>
<tr>
<td>Random forest</td>
<td>68.8</td>
<td>30.0</td>
</tr>
<tr>
<td>Single layer network</td>
<td>68.2</td>
<td>23.8</td>
</tr>
<tr>
<td>Our CNN</td>
<td>84.0</td>
<td>76.6</td>
</tr>
</tbody>
</table>

The results are summarized in table 3.2. While some of the traditional algorithms perform reasonably on the clean dataset, they fail on the realistic dataset. Our approach significantly outperforms the traditional methods on the clean dataset, and as demonstrated previously, it performs well on the lab measurement.

The details of the datasets are:

1. **Clean dataset**: This dataset aims to probe the ability to classify in extreme variation in calibration parameters in a noiseless measurement case. It is based on synthetic measurements with calibration parameters varying in ranges that are twice as large compared to the realistic dataset, and with $10^8$ photons without any additive noise (Fig. 3-2 shows two noiseless examples from this dataset). In this case both training and testing datasets are synthetic.

2. **Realistic dataset**: this is the dataset used for training the network described in section 3.2.3. It is based on renderings with $10^6$ photons with additive noise to approximate our SPAD measurements (see Fig. 3-2 synthetic examples 1 and 2). In this case the training is performed on the synthetic data and testing is based on the 30 lab measurements.

The different classification approaches that were used for comparison are:
1. **Mean example**: For each label we take the mean of the training data, such that we have one representative sample per label. Classification is performed based on nearest neighbor (closest sample in the dictionary to the measurement). This approach fails on both datasets.

2. **K-nearest neighbors**: Since this method may be sensitive to dictionary size, it is first evaluated on the clean dataset. We randomly choose varying number of samples from the training set to form different dictionary sizes. We consider two approaches here: a) Nearest neighbor — for each test point the chosen label is the label of the closest dictionary element. b) K-nearest neighbors (KNN) — for each test point the chosen label is the label of the majority of the K-nearest neighbors. K is chosen for each dictionary size with a validation set (taken from the training set). These results are presented in Fig. 3-7. The nearest neighbor approach shows decreased performance with increase in data size due to the increased ambiguity between dictionary elements. The K-nearest approach is able to overcome this limitation and provides classification accuracy in the range of 50% on the clean dataset, however it fails on the realistic dataset.

3. **Support vector machine (SVM)**: The SVM is evaluated with different kernels, and got the best performance with the linear kernel. After hyper parameters optimization we were able to achieve 57.1% classification accuracy on the clean dataset, and fail on the realistic dataset.

4. **Random forest**: A random forest is trained with 100 trees. The random forest achieves 68.2% accuracy on the clean dataset, and fails on the realistic dataset.

5. **Single layer network**: A neural network composed of one hidden layer. This network achieves 68.2% accuracy on the clean dataset, and like the previous methods, it fails on the realistic dataset.

This analysis presents the key difficulty and the requirement for both calibration invariance and robustness to noise. While some of the traditional approaches perform
Figure 3-7: Performance of the K-nearest neighbor approach on the clean dataset. Classification accuracy with varying dictionary size for a) nearest neighbor classifier, and b) K-nearest neighbors classifier.

reasonably well on the clean data, they fail on the realistic dataset. Our approach is the only one that achieves results that are better than random accuracy.

3.4 Discussion

While our approach is invariant to variations of calibration parameters within the training range, it still requires some approximate measurements or knowledge of system parameters and geometry. This limitation is somewhat mitigated by the fact that the network can operate well beyond its training regime (see Fig. 3-3 for examples). Another limitation is the need to synthesize a dataset and train the CNN on different types of geometries, which might slow down the process when arriving to a completely new setting. Faster hardware for data generation and CNN training can potentially address this in the future. Active acquisition systems like the ones used here, may suffer from interference with ambient illumination. This can be more challenging with single photon counting sensors. One possible solution is the use of narrow-band spectral filters to pass only the source’s wavelength. These filters are already used in systems such as LIDARs.

The measurement system suggested here uses time-resolved measurements with few spatial pixels (32 × 32). The importance of temporal resolution for classification when imaging through scattering media is evaluated with the suggested imaging
Figure 3-8: Time resolution is more important than number of pixels for imaging through scattering. a) Classification accuracy vs. time resolution (for 32x32 pixels). b) Classification accuracy vs. number of pixels (for non time-resolved system).

pipeline. The MC model is used to create training and test sets (based on the MNIST dataset) with different time resolutions. The result is plotted in Fig. 3-8(a), where we note that the performance degrades slowly until the time resolution nears 400ps and then degrades rapidly. In the scenes discussed and analyzed here, the time between the first and last signal photons spans roughly 500ps, so any time resolution better than that provides at least two frames with signal which allows the network to learn temporal filters. As seen from the measurements provided in Fig. 3-2, the spatial features have very little high frequency content, and therefore, unsurprisingly, low pixel count is sufficient for classification. To quantitatively evaluate this, we use the same pipeline to simulate no time dependency, while varying the pixel count. Figure 3-8(b) demonstrates that simply adding more pixels doesn’t improve the classification accuracy. This analysis is limited to the particular scene considered here and evaluates two extremes: low pixel count with varying time resolution and, no time resolution with varying spatial resolution. This demonstrates theoretical performance of commercially available hardware variants. We leave further analysis of potential hardware with e.g. high pixel count and significant time resolution to a future study.

The importance of time-resolved data for classification with CNN can be observed from the filters the network learns (Fig. 3-9). Inspection of these indicate that the network performs derivatives in the time domain. Similar spatio-temporal features have been demonstrated when using CNNs for action recognition in videos [64]. The
temporal features learned by our network combined with the strong dependency of classification accuracy on the SPAD’s time resolution shows that network inference is computed using information in both space and time.

Several aspects can be taken into account when considering the potential of this approach to scale into real-world applications:

- **Hardware:** Our hardware is a SPAD camera. Since SPAD cameras are manufactured with scalable semiconductor processes, they can be commoditized. Other approaches like phase based ToF systems are also a possibility (probably with significantly lower time resolution, which would impact its ability to classify). We note that SPAD cameras are especially useful for imaging through scattering due to several reasons:

  - They are single-photon sensitive which is extremely useful in NLOS geometries where the optical signal is very weak.
  - The time resolution of $\sim50\text{ps}$ corresponds to $1.5\text{cm}$ of spatial resolution, which is reasonable for room-sized scenes.
  - The low spatial resolution is not necessarily a drawback (see above analysis).

- **Real-time operation:** Since classification requires only a forward pass through the trained neural network, it can be performed in real time using specialized hardware (such as GPUs). The only caveat is the case of a completely new
scene that requires rendering new synthetic dataset and training a CNN. This requires anticipatory preparation before the real-time operation.

- **Flexibility**: The suggested forward model is based on an MC ray tracer. The MC model is very flexible and can render a wide range of optical geometries and materials.
Chapter 4

Imaging Around Corners

Vehicles, search and rescue personnel, and endoscopes use flash lights to locate, identify, and view objects in their surroundings. Here we show the first steps of how all these tasks can be done around corners with consumer cameras. Recent techniques for NLOS imaging using consumer cameras have not been able to both localize and identify the hidden object. We introduce a method that couples traditional geometric understanding and data-driven techniques. To avoid the limitation of large dataset gathering, we train the data-driven models on rendered samples to computationally recover the hidden scene on real data. The method has three independent operating modes: 1) a regression output to localize a hidden object in 2D, 2) an identification output to identify the object type or pose, and 3) a generative network to reconstruct the hidden scene from a new viewpoint. The method is able to localize 12cm wide hidden objects in 2D with 1.7cm accuracy. The method also identifies the hidden object class with 87.7% accuracy (compared to 33.3% random accuracy). This paper also provides an analysis on the distribution of information that encodes the occluded object in the accessible scene. We show that, unlike previously thought, the area that extends beyond the corner is essential for accurate object localization and identification. The work presented in the chapter was completed in collaboration with Guy Satat and Ramesh Raskar.

In section 4.1 we discuss an overview of the problem tackled and approach taken. In section 4.2 we describe and demonstrate a pipeline for localizing and identifying
Figure 4-1: Can we create an image as if seen from the ‘?’ using self contained consumer flash photography? a) A target is occluded from the camera. The scene is illuminated by a flashlight, light scatters off of the wall to the occluded object, floor, and to the camera. The measured signal changes depending on the class and location of the hidden target. b) The Measurement is processed and fed into a CNN for inference. The CNN is trained using rendered data. c) The system localizes, identifies and tracks the occluded object in real time. d) A generative model reconstructs a photo-realistic rendering of the hidden scene from the desired virtual camera’s point of view.

objects. In section 4.4 we describe and demonstrate a method for generated reconstructions of a scene from a new viewpoint. Finally, we conclude with a discussion about about the information content in the scene, the use of real data, and the method limitations in section 4.5.

4.1 Overview

The use of active illumination is common in computer and human vision. A car is equipped with headlights to help the driver or autonomous controller to drive at different times of day. A firefighter uses a flashlight or headlamp to see his surroundings while walking in a dark corridor. An endoscope uses bright illumination to navigate inside the body. Our goal in this paper is to extend these capabilities beyond the line of sight using consumer flash photography and data-driven recovery algorithms.

The task of seeing around corners has been accomplished with various measurement devices [24, 25, 9, 27]. Recently it was demonstrated using a traditional consumer camera with [34] and without [35] an active illumination source. Various computer vision applications have been demonstrated for non-line-of-sight (NLOS)
Table 4.1: Comparison of NLOS imaging techniques with a consumer camera.

<table>
<thead>
<tr>
<th>Method</th>
<th>Illumination</th>
<th># Light Bounces</th>
<th>Inference</th>
<th>Calibration sensitivity</th>
<th>Background subtraction</th>
<th>Required prior knowledge</th>
<th>Independent measurement of occluded area</th>
<th># Tracked objects</th>
<th>Localization</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>Flashlight (Broad)</td>
<td>3</td>
<td>Data driven</td>
<td>Easy: rectification</td>
<td>Robust: data driven</td>
<td>Required</td>
<td>Scene geometry</td>
<td>Potential object classes</td>
<td>Not required</td>
<td>1</td>
</tr>
<tr>
<td>Bouman[35]</td>
<td>Ambient</td>
<td>2</td>
<td>Geometric</td>
<td>Easy: rectification</td>
<td>Required</td>
<td>None</td>
<td>Not required</td>
<td>2</td>
<td>1D</td>
<td>2D with two corners</td>
</tr>
<tr>
<td>Klein[34]</td>
<td>Laser (Point source)</td>
<td>3</td>
<td>Geometric + Optimization</td>
<td>Challenge: require scene specific calibrated model</td>
<td>Required</td>
<td>Scene geometry</td>
<td>Specific object class</td>
<td>Not required</td>
<td>1</td>
<td>3D</td>
</tr>
</tbody>
</table>

imaging such as full scene reconstruction [24], and localization [54].

Recently, regular cameras have been used to image around corners. A method for tracking objects using a laser pointer was developed by Klein et al. [34]. A passive method (relies on ambient light) by Bouman et al. [35] is able to perform one dimensional tracking of objects around a corner. Table 4.1 summarizes the key differences between our approach and these two techniques. Two key properties of our approach are advantageous: 1) using flash photography is appealing for various practical scenarios as described before, and 2) our reliance on a data-driven technique allows us to demonstrate multiple computer vision tasks. Specifically, we locate the target in 2D, identify the object, and reconstruct a photo-realistic image of the hidden target area. Furthermore, since our approach is data-driven, it does not require extensive calibration similar to the one required by Klein et al. The localization accuracy reported by Klein et al. is 6.1cm (in 3D) compared to our reported accuracy of 1.7cm (in 2D).

To improve robustness of NLOS imaging with consumer flash photography, and demonstrate new capabilities like hidden object identification and photo-realistic hidden scene reconstruction, we turn to data-driven methods which have already proved useful in NLOS imaging [53, 54]. As in other computer vision applications, the main advantage of using such techniques is the inherent robustness and the fact that an accurate physical model is not necessary. Furthermore, this data-driven approach is a step toward general purpose hidden object classification.

The main challenge for data-driven solutions in the domain of NLOS imaging is the lack of large labeled databases. To tackle this challenge, we render a dataset for NLOS imaging, which is only limited by computational resources. To overcome the limitation
of transfer learning (ability to train on rendered data and operate on real-world data) we help the network to generalize by: 1) combining classical geometric knowledge into our imaging pipeline, and 2) introducing variations in rendering parameters into our dataset.

Here, we experimentally demonstrate NLOS imaging in a hallway with a consumer camera, coupled with a flashlight and a data-driven recovery algorithm. The main contributions of this paper are:

1. A technique to identify, localize, and track in 2D a hidden object around a corner using a combination of classical geometric processing and a data-driven approach.

2. A technique for full scene reconstruction that recovers the scene from a point of view that is not accessible to the camera based on a generative model.

3. An analysis of the spatial distribution of information on the occluded object in the visible scene.

4.1.1 Data-Driven Approaches to Imaging

Data-driven techniques have also been applied to problems in the computational imaging community such as phase imaging [45], compressive imaging [49], tomography [50], microscopy [51], imaging [52], and imaging through scattering media [53]. Localization and identification of people around a corner has also been demonstrated with a data-driven approach based on active pulsed illumination and a time-resolved single photon avalanche diode camera [54].

Using rendered data to train CNN models is becoming more common in computer vision applications such as optical flow [55], action recognition [56], overcoming scattering [58], text recognition [57], and tracking [58].

Here, we demonstrate the use of CNNs for NLOS imaging with a consumer camera. We show that rendering data using a graphics engine can be used to train a model
Figure 4-2: Geometric preprocessing. a) Photograph from camera before processing. Purple dotted line outlines the floor area to be rectified. The orange circle indicates the location where the corner wall meets the floor. b) The photograph is first rectified and cropped (left column, grayscale). Then, it is background subtracted and scaled to $[-1, 1]$ (right column, color). This is the input to the CNN. c) Two examples taken from the rendered dataset. The background subtracted examples demonstrate the light casted by the object which spreads throughout the measured area, not just the edge discontinuity.

that performs inference on real data. Finally, we demonstrate that a generative model can be used to reconstruct an occluded scene from a new point of view.

4.2 Localizing and Identifying NLOS Objects

As objects are added or removed from a scene, the measured irradiance changes with areas becoming dimmer or brighter. These changes still occur even when the object is moving in an occluded part of the scene. However, they are usually too subtle for the human eye to perceive, as our eyes are constantly adapting to our surroundings and ignore such information. In some cases, the changes in brightness are perceptible, but it can be hard to discern the cause of such changes. In either case this phenomenon is generally amplified with controlled illumination like flash photography. This section describes a technique to leverage such changes to localize and identify the hidden object.

In our experiments we are solving the problem of an ‘L’ shaped hallway. The
corner is occluding an object from the camera as shown in Fig. 4-1a. A flashlight is adjacent to the camera and is pointing at the wall opposite the camera. The camera is focused on the floor near the corner. In this arrangement, the useful signal is when light follows the path: light source → wall → object → floor → camera. Fig. 4-1a shows the geometry and example light path in the scene.

4.2.1 Rendering Synthetic Dataset

Many of the successes of deep learning systems in computer vision can be attributed to the availability of large scale datasets. The creation of such datasets is often difficult, time consuming, or impractical. For these reasons, our method utilizes rendered data for training and real data for testing and analysis. There are several advantages in using rendered data: First, the size of the dataset is bounded only by computation. Second, it is easy to increase diversity in the dataset by varying scene parameters such as material properties. And third, the labels are used in the generation of the data and are thus exact (noiseless labels).

To render photo-realistic measurements that account for the interaction of light with the hidden object, we use a physics-based ray tracing engine (Blender Cycles).
For robustness, the diversity of the dataset is increased by varying rendering parameters such as illumination, camera properties, material properties, and scene geometry.

The training dataset consists of 75,000 monochromatic images of $64 \times 64$ pixels. Generating a single image takes $\sim 5$ seconds on an Nvidia GTX 1080 GPU. Examples of the rendered samples are shown in Fig. 4-2.

### 4.2.2 Geometric preprocessing

**Rectification**  To increase robustness and allow the algorithm to operate independently of the camera position we first rectify the input image to a top-down view of the floor. The rectification involves placing the corner at the same pixel, thus easing the network to learn a model that is invariant to the camera’s point of view. Since this rectification process is rather simple, it is performed using traditional computer vision geometry. To rectify, we select four points, two on the back wall and two a known distance down the hallway closer to the camera. This quadrilateral is shown in Fig. 4-2a. The homography is calculated and used to project the quadrilateral into a $64 \times 64$ pixel square. Another approach would be to learn such transformation [70].

**Background Subtraction**  To further amplify the signal in the images we subtract the background from the measurements. Background subtraction has been a requirement in previous NLOS imaging when using consumer cameras [35, 34]. We explore several background subtraction techniques, including ground truth subtraction [34], subtraction of pixel-wise mean [35], and pixel-wise minimum of an input video. We note that all approaches require knowledge about the hidden scene, for ground truth we need to know the object is not there, and for mean or minimum subtraction we require the object to substantially move in the hidden scene during the measurement time. In our experiments, we found that subtraction of the ground truth provides better accuracies: ground truth subtraction: 1.71cm, 87.7%, mean subtraction: 12.26cm, 38.9%, minimum subtraction: 5.37cm, 79.2% for localization and identification accuracies respectively. Therefore results provided throughout this paper are based on ground truth background subtraction. Fig. 4-2 demonstrates raw measurement and synthetic rendering after rectification and background subtraction.
4.2.3 Localization and Identification Models

Two CNN models were trained for object localization and identification. The input image (following rectification and background subtraction) was scaled to $[-1, 1]$. Scaling individual images was necessary as the standard procedure of subtracting the dataset mean did not work in our case, due to the differences of the statistics between the real and simulated data.

Localization The localization model is a CNN regression network trained to predict the $(x, y)$ location of the object. The architecture of the model is diagrammed in Fig. 4-3a. The model was trained using a mean square error loss function for 7 epochs.

Identification A second model of similar architecture was trained to identify the type of the object. A softmax layer was used for identification, and training was based on the cross entropy loss for 20 epochs.

4.3 Implementation Details

Figure 4-4: Test scene. a) Dimensions of scene from an above perspective. b) Dimensions of scene from a side perspective. c) Dimensions of the three target objects tested. d) Photographs of the scene. Arrows in panel a point to the two perspectives.
Figure 4-5: Occluded object localization and tracking in 2D. a) Localization accuracy on a grid. Three objects of different shapes are moved along a $5 \times 5$ grid of size $30 \times 30 \ cm^2$. The center of each grid point is marked by a colored plus sign. Each object and each grid point are measured 30 times, during which the object is perturbed (90 data points per grid point, 2250 data points overall). Localizations are marked by colored dots. The color of each dot corresponds to the associated ground truth grid position, such that correct localizations have the same color as the closest ground truth plus sign. As the object is further away, the localization accuracy is reduced. The inset shows the hallway geometry, with the localizations superimposed and an example of one of the targets for scale reference. b) Tracking a hidden object. In these examples an object is tracing an infinity sign and circular path twice. Each frame in the camera video stream is processed and used for localization independently. Color along the path represents the time axis.
4.3.1 Experimental Results

To evaluate the method, an ‘L’ shaped hallway scene was constructed using poster boards. A set of objects of different shapes were placed in the occluded area. A Flea3 Point Grey camera captured 16-bit monochromatic images at 30fps. The scene was illuminated by an LED flashlight. Examples of the real world measurements can be seen in Fig. 4-2b. The scene setup is illustrated in Fig. 4-4

A test dataset was gathered in the above settings to evaluate the localization and identification accuracy. Three different objects (rectangular box with a base of 12cm, cube with a base of 18cm, and pyramid with a base of 12cm) were placed in 25 different positions on a uniform $30 \times 30 \text{cm}^2$ grid. The grid starts 10cm away from the corner. At each position, and for each object, 30 images were captured and processed (90 samples per grid point, totaling 2250 examples in the test dataset).

Localization and Tracking

Fig. 4-5a shows the computed locations alongside the ground truth. As can be seen from the localization distributions, when the object is close to the visible area the localization accuracy is higher (1.71 cm accuracy when the object is within 25 cm from the corner). When the object retreats further away the accuracy reduces (3.21 cm accuracy when the object is between 25-45 cm from the corner). The overall localization accuracy is 2.61cm across all examples in the test set. We note that when the object is closer to the illuminated wall, it is less likely to be properly illuminated by the reflection from the wall itself and thus contributes less to the measured signal on the floor. This leads to reduced localization accuracy.

Since the required computational steps are simple geometrical operations and a single inference pass through a shallow network, the method operates in real time. We track the object over time by localizing it in each frame independently. To that end, we recorded a video of the object tracing in an infinity sign and a circular motion for a total of 500 frames. The predicted locations are presented in Fig. 4-5b (note that these tracking plots did not require any smoothing or filtering).
Figure 4-6: Hidden Object identification. a) A confusion matrix showing the identification accuracies among the three classes on real data. The reported accuracies are averages over all locations. b) Identification accuracies on different grid positions.

Figure 4-7: Visualizing the occluded scene with a generative model. a) Photograph of the scene taken from the desired point-of-view, used for reference. b) Reconstruction results using a graphics program. The scene parameters are first estimated and then are used to render the image. c) Reconstruction results with a generative model. Different columns show results for different objects and positions from the test dataset.
Identification

The test dataset was also used for identification accuracy. Fig. 4-6a shows the confusion matrix accounting for all locations. Fig. 4-6b shows the overall identification accuracy as a function of location (evaluated independently for each grid point). The identification accuracy is correlated with the size of the object (highest for the tall rectangular box and lowest for the short cube). This is expected as larger objects reflect more light to the floor and increase the measurement signal-to-noise ratio. The identification accuracy across the entire grid is in the range of \([55.6, 100]\)% with a mean of 87.7% (compared to 33.3% random accuracy).

4.4 Generative Occluded Scene Reconstruction

Our goal in this section is to go beyond the demonstrated computer vision tasks (localization, tracking, and identification) to full scene reconstruction. The recovered object location and class are sufficient to render an image of the hidden object using the graphical renderer. The success of this approach is limited by the accuracy of the localization and identification networks. Previous work by Klein et al. [34] have demonstrated a similar capability with a consumer camera, but without object identification. Recent advances in generative models allow us to go beyond a graphics program and render a photo-realistic image of the hidden scene as if it was taken from a point of view not accessible to the camera (see Fig. 4-1).

The suggest approach is based on a variational auto-encoder (VAE) [71]. The VAE architecture is plotted in Fig. 4-3. The input to the generative model is the same as before (rectified, background subtracted image). The output is a rendering based on a point of view facing into the occluded area (Fig. 4-1).

Similar to the previous methods, the generative model can also be trained on rendered data. To that end, the desired output image is rendered along with the input image as part of the data generation pipeline described in section 4.2.1. Thus, the training dataset is composed of sets of input and output renderings (75,000 data points). The generative model is trained for 100 epochs.
Figure 4-8: Areas encoding information about the hidden scene. Brighter colors represent areas of higher sensitivity for the network in a) localization and b) identification. c) Comparing the importance of the area between the wall and corner (marked as 1 in the inset), and the area between the corner and the camera (marked as 2 in the inset). The table shows the localization and identification accuracies when only area 1 or 2 are available for the network. For localization both areas are equally important and using both significantly improves performance. For identification using just area 2 result in random prediction, and area 1 is essential for prediction.

Figure 4-7 shows the results for the generative model evaluating its performance on the test dataset (section 4.3.1). The figure compares: a) A reference photograph, taken with a camera at the desired position. b) A graphics rendering, using our data generation pipeline, with the estimated location and class as described in section 4.3.1. c) The output of the generative model. The figure plots several examples with different object shapes and positions.

The main advantage of this generative approach is its generality since it can reconstruct the hidden scene without implicitly going through the localization and identification steps, and more importantly it does not require a labeled dataset, just pairs of images. This concept is further discussed and demonstrated in section 4.5.3.

4.5 Discussion

4.5.1 Areas That Contain Information In The Scene

It is important to consider which areas in the visible part of the scene encode information about the occluded part. Bouman et al. [35] developed an analytical model showing that information about the angle of hidden objects are encoded in the area between the camera and the corner (marked as area 2 in Fig. 4-8c). The input to our
network includes a larger area that extends further, all the way to the wall facing the camera (marked as areas 1+2 in Fig. 4-8c).

To evaluate the contribution of different regions in the visible area for localization and identification accuracy, we block a small region (20 × 20 pixels) in the input frame and try to localize and identify the hidden object (this is similar to [72]). Repeating this process for multiple objects, positions, and different blocked regions in the visible area provides a sensitivity map of the network (a total of 800 examples from the rendered dataset were used for this evaluation). Areas to which the network is more sensitive encode more information about the hidden object.

The results for this sensitivity analysis for localization and identification is shown in Fig. 4-8(a,b). For localization, substantial information is encoded around the occluding corner (as predicted by Bouman et al. [35]). However, there is also significant sensitivity to the region that is between the corner to the further wall (area 1). For identification, most of the information is encoded beyond the corner in the area 1. To further evaluate this, we performed three tests: 1) only area 1 is visible, area 2 is blocked, 2) only area 2 is visible, area 1 is blocked (like in Bouman at al.), 3) both areas visible for baseline. We note that for localization when just one of the areas is visible the results are similar (equal contributions), however when both areas are available there is a dramatic boost in localization accuracy. For identification the results are sharper; the use of area 2 alone results in random accuracy, while area 1 alone is able to identify with 51% accuracy. It is expected that placing the separating line between the two areas in different locations would produce different results. We choose this particular separating line to compare our approach to Bouman et al. that used only area 2.

The model suggested by Bouman et al. requires a discontinuity in the scene (such as a corner) in order to capture high frequency information that encodes the object’s angle around the corner. Our approach is different, since we rely on a three-bounce light transport model. This model does not necessarily require such a discontinuity (although it helps), and can recover information about the hidden object from other regions in the visible scene. Furthermore, these regions contribute to our
Figure 4-9: Generative reconstruction results on trained real data. a) Scene photograph, showing the input camera (observing the floor) and the occluded scene. b) Input camera measurements. c) Ground truth measurements from the output camera (same camera used for training). d) Generative reconstruction, taking the background subtracted measurement as input and generate the scene from the desired point-of-view.

ability to localize the object in 2D and identify it.

4.5.2 Illumination Position and Camera Field-of-View

In our setup, we choose to illuminate a wall facing the camera and observe the floor in front of the camera. This configuration is not unique. For example, if there is no wall facing the camera, it would be possible to illuminate the floor instead [9]. In this case, it would be desirable to separate the camera field of view and the illuminated area to support the finite camera dynamic range. Other options may include illuminating the floor and observing the ceiling or vice-versa.

4.5.3 Generative Models Trained on Real Data

As a first step towards thinking about generalization, we approach dataset generation differently. The generative model demonstrated in section 4.4 was trained on rendered
data. Using rendered data is important in cases when it is hard to both gather and label datasets. However, the generative model approach does not require labels. Thus, it is possible to train such models on real data. The only requirement for such training data is the ability to access the occluded part of the scene to place a camera during training (we did not have such a requirement in our previous results). Such a requirement may be reasonable in the case of security cameras in places like alleys or museums. In such places it would be easy to gather a large dataset (i.e. saved video consisting of footage from different times of day and times of year) and create a system that provides security redundancy or reduced costs (less cameras). Or in manufacturing the ability to see part defects that are hidden from the camera.

To demonstrate this concept we placed two cameras in a conference room (see Fig. 4-9). One camera is observing the floor, similar to the input camera in previous sections. The second camera is observing a scene outside the field of view of the first camera (Fig. 4-9a). To train the system the cameras simply record synchronized videos that are then used to train the generative model. During the recordings a mannequin was moved around with different clothing in the occluded part of the room. A total of 14,600 frames were used for training (data gathering duration of less than 10 minutes). For testing we used a separate recording in which the mannequin was wearing different clothes. Several examples of the generative reconstruction are plotted in Fig. 4-9.

4.5.4 Main Limitations

This work has two main limitations:

The first limitation of our approach is the limited number of predefined objects it can identify, and the ability to recover only a single object. This is in part due to the limited number of classes generated during the dataset rendering step. This generalization challenge is not unique to NLOS imaging, but rather an artifact of current deep learning methods. In the future, improved low shot learning methods have the potential to ease the demand on large scale dataset generation. A different approach would be to develop alternative dataset generation pipelines as discussed in
While the approach is limited to predefined objects, this alone is enough for useful applications such as a car detecting a vehicle occluded by a building corner.

The second limitation of our approach is the requirement for known scene geometry. In many cases, the desired operating mode would be to approach an unknown scene and perform NLOS imaging. In future work, this limitation is likely to be solved in a similar manner to the limitation of predefined objects. Using datasets that data from many locations with different geometries, illumination conditions, materials, etc. would likely be required. Additional dimensions of data such as the scene depth can be used to assist future models. An alternative is to scale the generative model option.
Chapter 5

Conclusion

This thesis presented methods for NLOS imaging using data-driven techniques. We explained the advantages of using such techniques compared to traditional inversion based techniques. In chapter 3 we presented a method for classifying the pose of a mannequin occluded by a sheet of paper. Using a MC renderer to simulate thousands of time of flight measurements, we created a training dataset. The dataset was used to train a model that could then be used on real data. In chapter 4 we train a model on simulated data to classify and localize an object using a regular camera. we proposed using measurement data at the corner and beyond the corner to improve the results. Finally we demonstrate a method that uses a generative model to render images from viewpoints that extend the original camera’s field of view.

This thesis acts as a step towards practical NLOS imaging. Methods described can already be applied to real world applications, such as security systems. A generalization of this work can provide opportunities in applications such as vehicular safety, medical imaging, search and rescue, and manufacturing.
Bibliography


