DETECTION AND TRACKING OF OBJECTS OUTSIDE THE LINE OF SIGHT USING SUPPORT VECTOR MACHINES

Sreenithy Chandran
Arizona State University
schand56@asu.edu

Abstract

Non Line of Sight Imaging is a recurring challenge with a wide variety of important applications. The most commonly proposed approaches to date for Non Line of Sight Imaging depend on exploiting time-resolved measurements, i.e., measuring the time it takes for short light pulses to transit the scene. This typically requires expensive, specialized, ultrafast lasers and detectors that must be carefully calibrated. We develop an alternative approach that uses only a 2D camera and spot light source to perform the imaging. By repeatedly simulating light transport through the scene, we render images are used to perform tracking using a Support Vector Machine.

1. Introduction

Non-line-of-sight (NLOS) imaging [15, 7] refers to estimation of the shape, texture, and reflectance of scene points that lie beyond the field of view of an imaging system. NLOS imaging of objects is a problem of fundamental importance to many fields of research with applications in robotic vision, defense, remote sensing, medical imaging, search and rescue missions and autonomous vehicles[2]. Despite recent advances in this domain, NLOS imaging has remained impractical due to processing requirements of existing reconstruction algorithms and the extremely weak signal of multiply scattered light. Improvement in NLOS imaging can prove to be a breakthrough for addressing security issues associated with self driving cars.

General approach of reconstructing the unknown object usually involves a back propagation scheme however this is based on the assumption of the availability of ultrafast time-resolved optical impulse responses. Usually in Time-of -Flight techniques there is a wall in front of the occluded object which is illuminated with short light pulses, and the returned light intensity is represented as a function of the illuminated and detected pixels on the wall and the time between illumination and detection. This reconstruction is usually viewed as an inverse rendering problem and an algorithm is then used to find the time-resolved light intensity measurements to form a three-dimensional reconstruction of the hidden scene. This can be seen as an inverse rendering problem and such an approach necessitates high system complexity and cost. Also this approach requires accurate calibration of imaging system parameters like illumination position, camera orientation like position, field of view etc,. Despite their obvious appeal, there are inherent challenges in the design of NLOS systems. In particular, typical surfaces (e.g., walls, floors, etc) diffusely reflect light, effectively removing beam orientation information, and rendering the problem of scene reconstruction poorly conditioned. In order to compensate for the losses induced by diffuse reflections, ini- tial demonstrations of NLOS imaging used ultrafast transient- imaging modalities. The system requirements posed by these systems, for trans- mission of very short, high power optical pulses on the transmitter side, and for very high temporal resolution on the receiver side, inevitably imply high system complexity and cost.

However for this work we explore the possibility of tracking without having to use the traditional TOF imaging methods and laser pulses but while harnessing the power of machine learning to learn from a collection of rendered images to perform tracking. This work draws inspiration from a previous work [8] that makes use of an implicit technique for detecting and tracking objects outside the line of sight in real time. Imaged using routinely available hardware (2D camera, spot light), the distribution of indirect light falling back onto the wall serves as our main source of information. This light has undergone multiple reflections; therefore, the observed intensity distribution is low in spatial detail. For our method instead of aiming to reconstruct the scene we perform tracking and detection using the light obtained.

In this work we make use of simulation data that is rendered of the reflective wall in a NLOS setup then we perform tracking using the data collected using support vector machine. Thus this work hopes to make the following contribution: Using synthetic measurements, to quantify the effect of object movement, and predict track the object under

which conditions the effect is significant enough to be detected.

2. Background

Imaging objects outside the line of sight can be represented by the following equation [10]:

$$\varsigma(x^{'}, y^{'}, z^{'}) = \int\int\int \frac{1}{r_l^2 r^2}\rho(x,y,z) * \delta(\sqrt[2]{(x^{'}-x)^2+(y^{'}-y)^2)+z^2} + \sqrt[2]{(x_l-x)^2+(y_l-y)^2)+z^2} - tc) dx.dy.dz$$

Here, ρ is the sought-after albedo of the hidden scene at each point in the three-dimensional half-space. The transient image is recorded while the light source illuminates position x_l, y_l on the wall with an ultra- short pulse. This pulse is diffusely reflected off the wall and then scattered by the hidden scene back towards the wall. The radiometric term $\frac{1}{r_l^2 r^2}$ models the square distance falloff using the distancer_l between x_l, y_l and some hidden scene point x, y, z, as well as the distance r from that point to the sampled detector position on the wall $x^{'}, y^{'}$. This simplifies to an inverse problem.

Explicit reconstruction of the scene assuming the the availability of ultrafast time-resolved optical impulse responses, whose capture still constitutes a significant technical challenge. Techniques proposed in literature include holography[1, 12], streak imagers [14], gated image intensifiers [9], single-photon avalanche diodes [4]. A streak camera uses a time-varying electric field to introduce a time-dependent spatial displace- ment to incoming photons. Single-photon avalanche diodes (SPADs) provide an alternative, less-expensive approach for measuring the transient. The SPAD has an infinite gain and hence, saturates whenever a photon is incident on the sen- sor; the time-stamp of this event measures the ToF of the photon. In contrast, implicit methods state the reconstruction task in terms of a problem-specific cost function that measures the agreement of a scene hypothesis with the observed data and additional model priors.

One of the first such work is by Heide et al.[5] in which the solution to the problem is defined as the function argument that minimizes the cost, the authors regularize a least-squares data term with a computationally expensive sparsity prior, which enables the reconstruction of unknown objects around a corner without the need for ultrafast light sources and detectors. They demonstrate NLOS shape reconstruction directly from a PMDs readout by using a generative

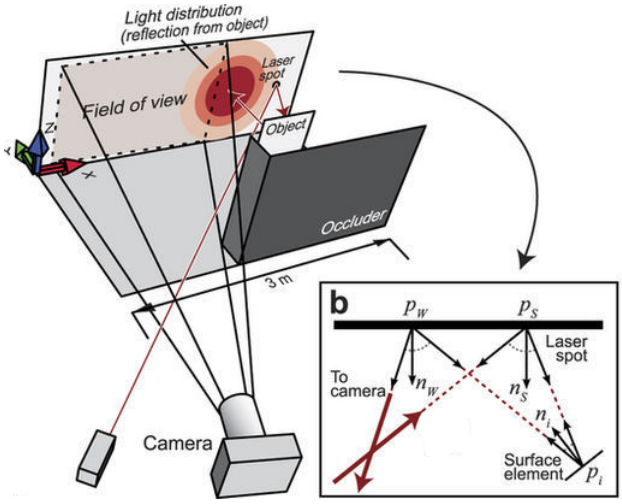


Figure 1. Experimental Setup used by Heide et al., the camera observes a portion of the wall, to the right of the cameras field of view, a collimated laser illuminates a spot that reflects light toward the unknown object[8]

model, instead of ellipsoidal constraints. Another common approach is to introduce surface priors to simplify the shape estimation problem. Kadambi et al. [6] use signal sparsity to recover distinct scene points.

A recent paper approaches this problem by performing real time object detection and tracking using the 2D intensity images [8] where a 2D camera and laser pointer was used to image the occluded scene in an analysis-by-synthesis sense. Here they perform NLOS pose estimation using just image intensities. In this case, the number of unknowns is smaller and image intensities are sufficient for estimating the hidden object Another recent paper performs object classification through scattering media with deep learning on time resolved measurements[13]. Toole et al. [10] in their recent paper propose a light cone transform (LCT) that expresses the confocal NLOS image formation model as a shift-invariant 3D convolution in the transform domain. The LCT is a computationally efficient way for computing the forward model and importantly, leads to a closed-form expression for the inverse problem. The paper along with the supplementary document provides a very detailed and comprehensive explanation of the methodology, setups and proofs of the image formation model used in the paper. Taking insights from these advances for imaging objects outside line of sight, this project proposes to carry out object detection and tracking of objects outside line of sight using SVM.

3. Method

This work proposes tackling the popular challenge of tracking an object outside line of sight without the use of time-of-flight technology and by formulating an optimiza-

tion problem based on a simplistic image formation model. By employing machine learning techniques to optimize and find the parameters that relate the light transport simulation that is generated and given a measurement of light scattered from the object to the wall, we handle the tracking task.

3.1. Light Transport Simulation

Using synthetic data for deep neural network training has shown promising results in the past[13], synthetic data have been utilized to generate training data for predicting object pose, semantic segmentation, and investigating object features. In this approach due to the limited availability of dataset we propose to make use of machine learning techniques like support vector machines to learn from synthetic data and use the parameters learned for tackling the tracking problem. Thus the synthetic data which comprises of rendered images, as formed using a spot light as the light source is the first step in this parametric object tracking problem.

By assuming that all light has undergone exactly three reflections, we get reduced overall computational complexity that is linear in the number of pixels and the number of surfels n. Each camera pixel observes a radiance value, L, leaving from a point on the wall, p_W , that, in turn, receives light reflected by the objects surfels. The portion contributed by the surfel of index $i\epsilon\{1...n\}$ is the product of three reflectance terms, one per reflection event; and the geometric view factors known from radiative transfer

$$L_i = \rho_0 . f_s(p_L - p_s, p_i - p_s)$$

$$\cdot \frac{(n_s \circ p_i - p_s).(n_i \circ p_s - p_i)}{\|p_s - p_i\|_2^2} . f_i(p_s - p_i, p_W - p_i).A_i$$

$$\cdot \frac{(n_i \circ p_W - p_i).(n_W \circ p_i - p_W)}{\|p_i - p_W\|_2^2} . f_W(p_i - p_W, p_c - p_W)$$

$where$

$$a \circ b = \frac{a^T.b}{\|a\|_2^2 \|b\|_2^2}$$

denotes a normalized and clamped dot product as used in Lamberts cosine law.

3.2. Rendering

A Monte Carlo (MC) renderer [11] used to synthesize a dataset which consists of the illumination formed on the wall as the position of the object is varied. 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. Ray tracing is based on following the path of a ray of light through a scene as it interacts with and bounces off objects in an environment. In this setup the light is assumed to follow three bounces. So we simulate scenes using this ray tracing methodology. After the data generation, machine learning algorithm is used to learn from the dataset and optimise the parameters for detection and tracking. This enables the second phase where the capture from the camera can be tracked using the trained model. A Monte Carlo (MC) model is used to synthesize a large training dataset for various positions of focus on the illumination wall of the laser and camera setup.The synthetic dataset generated is used for training purpose and is done using *Physically Based Rendering* (PBRT)[11].

Ray tracing 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. Ray tracing is based on following the path of a ray of light through a scene as it interacts with and bounces off objects in an environment. In this setup the light is assumed to follow three bounces. So we simulate scenes using this ray tracing methodology. For the purpose of rendering photo realistic images PBRT simulates the following objects and phenomena:camera(how and from where the scene is being viewed), rayobject intersections, light sources, visibility, surface scattering, indirect light transport and ray propagation.

Monte Carlo integration is a method for using random sampling to estimate the values of integrals. Randomness is used to evaluate integrals with a convergence rate that is independent of the dimensionality of the integrand. Hence ray tracing and Monte Carlo Integration is used for the synthetic data generation.

4. Support Vector Machines

In order to perform tracking using Support Vector Machine(SVM), feature extraction is the first step. Traditional computer vision techniques, such as Histogram of Oriented Gradients (HOG) and other features combined with sliding windows is generally used for object detection and tracking [3]. The feature vector used is a combination of spatial features, which are nothing else but a down sampled copy of the image patch to be checked itself, color histogram features that capture the statistical color information of each image patch and Histogram of oriented gradients (HOG) features, that capture the gradient structure of each image channel. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge position. The intensity difference images obtained is then used for feature extraction following which we use a linear support vector machine for classification.

5. Experiment

At the center of this work is an efficient renderer for three- bounce light transport. Like all prior work, we assume that the wall is planar and known, and so is the position of the spot light source. The object is represented as a collection of Lambertian surface elements (surfels), each characterized by its position, normal direction and area. As the object is moved or rotated, all its surfels undergo the same rigid transformation. We represent this transformation by the scene parameter p, which is a three-dimensional vector for pure translation, or a six-dimensional vector for translation and rotation. The irradiance received by a given camera pixel is computed by summing the light that reflects off the surfels. The setup is described below.

5.1. Experimental Setup

The problem is approached based on the assumption that the object is not in the direct line of sight of the light source or camera, it can only be illuminated or observed indirectly via a diffuse wall and all the observed light has undergone at least three diffuse reflections (wall, object, wall). Our experimental setup follows the arrangement reported in prior work [8]. The light from the spot light hits the illumination wall at a point (w1) falls on the hidden object (x) hits the wall back(w2) and is captured by the focused camera. This forms three-bounces in which light beams follow paths of the form

$$Laser \rightarrow w1 \rightarrow x \rightarrow w2 \rightarrow Camera$$

Suppose M and S(p) are vectors corresponding to the pixel values of the measured object at w2 and the one predicted by the simulation under the transformation parameter or scene hypothesis p, respectively. We find the parameter p that brings M and S(p) into the best possible agreement. This search for the best parameter p is done by performing machine learning algorithms on a synthetic dataset

In **Experiment 1** a sphere is placed in front of the reflective wall and and a spotlight source illuminates the reflective wall, the experiment is performed with and without ambient lights and the object is moved along the x, y and z direction.

In **Experiment 2** the same experiment is repeated with a square sheet which is placed in front of the reflective wall and and a spotlight source illuminates the reflective wall, the experiment is performed with and without ambient lights and the object is moved along the x, y and z direction.

In **Experiment 3** we rotate the object about its axis and obtain the intensity difference images, that is we place a sphere and a square which is rotated about its axis and then we perform the rendering.

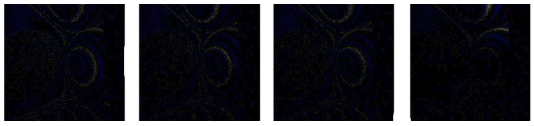


Figure 2. Difference images obtained when image is translated along the x direction

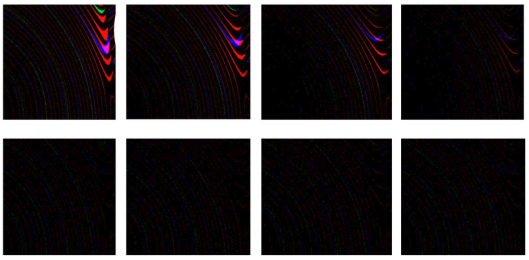


Figure 3. Difference images obtained when image is translated along the y direction

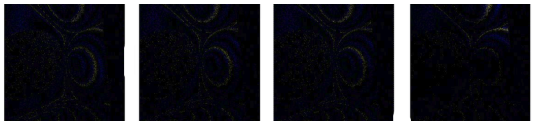


Figure 4. Difference images obtained when image is translated along the z direction

By varying position and location of the object, we obtained difference images that can be interpreted as partial derivatives with respect to the components of the scene parameter p. Since the overall light throughput drops with the fourth power of the object-wall distance, translation in Y direction caused the strongest change. Translation in all directions and rotation about the X and Z axes affected the signal more strongly than the other variations.

5.2. Dataset

For performing tracking we divide the entire room volume that is not within the direct line of sight to the camera setup into nine voxelized spaces. And each voxel is treated a different class and given a label from A to I. Then the object is placed at 100 different positions of varying (x,y,z) values in each voxel. Two different synthetic datasets was rendered for this project.
In **Dataset 1** we use a rectangular sheet as the occluded object and we position it at 100 different values of translation and rotation in each of the 9 voxels, thus Dataset 1 consists of 900 images.
In **Dataset2** we use a perfectly reflecting sphere as the oc-

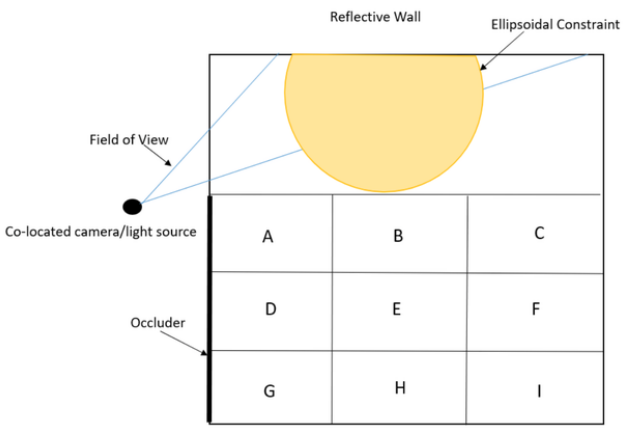


Figure 5. Division of NLOS scene into nine voxels

cluded object and repeat the same as in the Dataset 1. Thus we have a Dataset 2 comprising of 900 images.

5.3. Implementation Details

To carry out the tracking we we first perform image feature extraction and use the extracted features to perform classification and find which voxel the image captured by the camera lies in. The details of the implementation are given below.

5.3.1 Feature Extraction

In order to perform feature extraction we choose the feature descriptors that quantifies an image globally. Since the rendered images dont have the concept of interest points we thus, take the entire image for processing. So instead of using the local feature descriptors that quantifies local regions of an image, where interest points are determined in the entire image and image patches/regions surrounding those interest points are considered for analysis we use global feature descriptors such as the ones give below:

1. Color Histogram that quantifies color
2. Hu Moments that quantifies shape
3. Haralick Texture that quantifies texture

We extract these three global features for each of the images and concatenate the three global features into a single global feature. Since we use different global features, one feature might dominate the other with respect to its value. Thus, we normalize the features and save our features along with its label in a HDF5 file format.

5.3.2 Tracking

We perform the multi-class classification using SVM and also with other machine learning algorithms such as Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Decision Trees (CART), Random Forests (RF) and Gaussian Naive Bayes (NB) We make use of a K-fold cross validation with K=10 since it is the best way to predict a model's accuracy. Assuming Dataset 1 denotes images with sphere as the occluded object and Dataset 2 denotes rectangular sheet as occluded object, the classification was performed for four varying scenarios and the results are reported in Table 1.

1. Train images and test images from Dataset
2. Train images and test images from Dataset 2
3. Train images from Dataset 1 and test images from Dataset 2
4. Train images from Dataset 2 and test images from Dataset 1

5.4. Evaluation Metrics

Using the scores generated by our classification algorithms considered, we report top-1 classification accuracy (the fraction of images identified correctly), precision, recall, and F1 scores. We compare the performance of the algorithm against each other and report them. Precision and recall are defined as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

$$F1\ scores = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

In addition to quantifying the accuracy of our networks, we also observe the confusion matrix of the classifiers.

6. Results and Discussion

Table 1 compares the accuracy of the different machine learning algorithms across our key metrics. We can see that Naive Bayes gives the least accuracy compared to the other algorithms. In the case when the training data and test data are not chosen from the same dataset, that is when we train the algorithm on dataset with say, rectangular sheet, as the occluded object and try to track a sphere that is occluded from the camera, all of the algorithms fail. Due to the very low performance of all algorithms for such cases where the train and test data do not come from the same dataset we do not report the accuracy values.
Support Vector Classifier gives the best accuracy of about

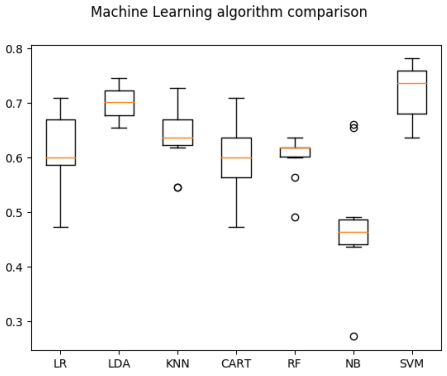
78% on the dataset with rectangular sheet as the occluded object and 76% on the dataset with a sphere as the hidden object . SVM with RBF kernel was used for classification, we obtained the best performance for RBF kernel, penalty parameter C of the error term equal to 100, kernel coefficient for RBF, gamma equal to 0.001. For the scenario when the training and test images are not from the same dataset the accuracy obtained was low and hence has not been reported.

Closely following SVM, LDA gives a significantly good performance. LDA can be seen as an efficient dimension reduction method since it transforms the original data into a low dimensional space determined by the *n-1* different eigenvectors where *n* is the number of different classes. Understanding the interpretations of the LDA basis vectors could potentially lead to efficient dimension reduction and less time for processing compared to SVMs. This would be of particular significance when we divide the NLOS scene into more voxels and thus more number of classes. The accuracy of LDA is followed by KNN and RF.

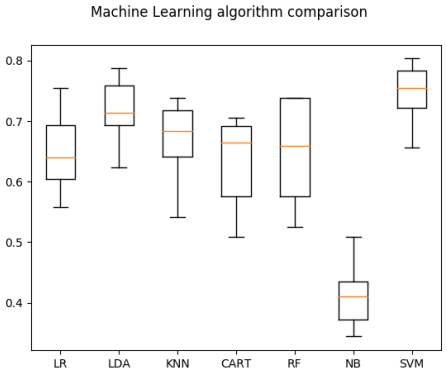
Usually, the characteristics of the dataset that we are working will be a guide to the decision of applying LDA as a classifier or a dimensionality reduction algorithm to perform a classification task. The main of Linear Discriminant Analysis is basically separate example of classes linearly moving them to a different feature space, therefore if the dataset is linear separable, only applying LDA as a classifier gives great results. However, if the dataset is not linear separable the LDA will try to organize the dataset in another space as the maximum linearly separability as possible, but there still will be examples of overlapping between classes because of non-linearly characteristic of data. In this case, we will need to use another classification model to deal with nonlinear data such as neural network with multiple hidden layers, neural network with radial basis function or SVM with nonlinear Kernels. Hence the performance of the LDA for this dataset can't be seen as an alternative to using SVMs.

Network performance as ranked by test set accuracy is consistent with performance on the other reported metrics as well. Although there is variation in the F1, precision, and recall scores, they tend to track closely with classification accuracy

The confusion matrix is used to evaluate the quality of the output of a classifier. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions. As seen from the confusion matrices there is a maximum value along the diagonal matrix which implies that all the classes were predicted with high probability irrespective of the positioning of the voxel. This is an important observation since it shows that that irrespective of the position of the voxel the SVMs were able to give a good performance.



(a) Dataset 1



(b) Dataset 2

Figure 6. Comparison chart of different machine learning classifiers (Y-axis: Accuracy)

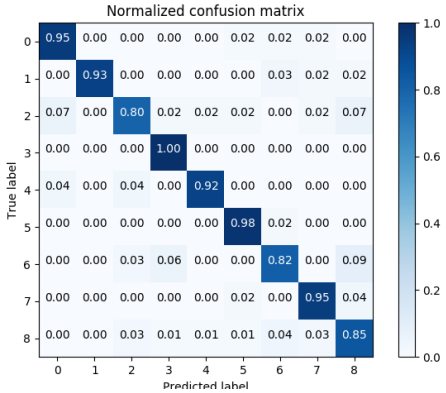


Figure 7. Confusion matrix for top-1 classification accuracy on the Dataset 1 using SVM with RBF kernel

Table 1. Tracking results Summary

Data Set	Algorithm	Testing Accuracy	Precision	Recall	F1-score
1	LR	0.61	0.45	0.53	0.47
1	LDA	0.70	0.70	0.67	0.68
1	KNN	0.63	0.56	0.55	0.54
1	CART	0.59	0.51	0.49	0.50
1	RF	0.60	0.54	0.53	0.53
1	SVM	0.78	0.73	0.73	0.74
1	NB	0.49	0.49	0.45	0.42
2	LR	0.64	0.60	0.58	0.58
2	LDA	0.72	0.72	0.72	0.71
2	KNN	0.66	0.65	0.65	0.65
2	CART	0.63	0.66	0.65	0.65
2	RF	0.65	0.64	0.64	0.64
2	SVM	0.76	0.75	0.75	0.73
2	NB	0.4	0.43	0.39	0.37

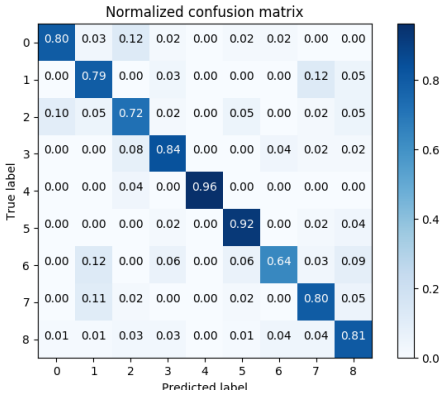


Figure 8. Confusion matrix for top-1 classification accuracy on the Dataset 2 using SVM with RBF kernel

7. Conclusion

The central finding of this work is that the popular challenge of tracking an object around a corner can be tackled without the use of time-of-flight technology. Using our dataset that comprises of 900 images with a sphere as an occluded object and another 900 images rendered with a rectangular box as the occluded object, we demonstrate that by formulating the NLOS problem based on a simplistic image formation model it is possible to track an occluded object only using 2D images with a spotlight as the light source.
In a room-sized scene, for small movements of the object along the x,y and z directions the machine learning algorithm gives good classification performance. Thus, this technique achieves even small scale repeatability, which puts it on par with the latest time-of-flight-based techniques. Also since we do not make use of the temporally resolved measurements of any kind, it has the unique property of being scalable to very small scenes (down to the diffraction limit) as well as large scenes (sufficient light power provided). Using difference images, for instance, we investigated the influence that parameter changes have on the signal. The same mechanism could also be used to obtain robustness estimates regarding additional unknowns in the scene model, such as non-diffuse object reflectance or the presence of additional objects. With these options, our approach offers a significant advantage over existing non-line-of-sight sensing techniques.
Our experiments have shown that SVM is a simple efficient and accurate approach for the multi-class classification problem and SVMs outperform other traditional machine algorithms by a significant margin. The precision of LDA approach is comparable to SVM, however, the time consumption of LDA approach is much less than SVM. Also as can be seen from the results when the train and test images are chosen from the same experimental setup the accuracies are of practical significance. However when we have to track an object in a scene that the SVM has not been trained on, the results obtained were not of any practical use.
We observe that the quality of the rendered images is an important parameter for obtaining on par performance. When the scenes are rendered with not with sufficient pixels per sample the results were not on par. As an extension we would like to experiment if there is a possibility of improving the performance of the classifiers if better machine learning algorithms were used. As a a future work we would also explore the possibility of using deep learning techniques for tracking. Also we would expand the dataset by rendering scenes with more than one object in the hidden scene with varying reflectivity of the wall as well as object and evaluate using the machine learning algorithms.

References

[1] N. Abramson. Light-in-flight recording by holography. *Optics letters*, 3(4):121–123, 1978.

[2] S. Chan, R. E. Warburton, G. Gariepy, J. Leach, and D. Faccio. Non-line-of-sight tracking of people at long range. *Op-*

tics express, 25(9):10109–10117, 2017.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[4] G. Gariepy, N. Krstajić, R. Henderson, C. Li, R. R. Thomson, G. S. Buller, B. Heshmat, R. Raskar, J. Leach, and D. Faccio. Single-photon sensitive light-in-fight imaging. *Nature communications*, 6:6021, 2015.

[5] F. Heide, L. Xiao, W. Heidrich, and M. B. Hullin. Diffuse mirrors: 3d reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3222–3229, 2014.

[6] A. Kadambi, H. Zhao, B. Shi, and R. Raskar. Occluded imaging with time-of-flight sensors. *ACM Transactions on Graphics (ToG)*, 35(2):15, 2016.

[7] A. Kirmani, T. Hutchison, J. Davis, and R. Raskar. Looking around the corner using transient imaging. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 159–166. IEEE, 2009.

[8] J. Klein, C. Peters, J. Martín, M. Laurenzis, and M. B. Hullin. Tracking objects outside the line of sight using 2d intensity images. *Scientific reports*, 6:32491, 2016.

[9] M. Laurenzis and A. Velten. Nonline-of-sight laser gated viewing of scattered photons. *Optical Engineering*, 53(2):023102, 2014.

[10] M. OToole, D. B. Lindell, and G. Wetzstein. Confocal non-line-of-sight imaging based on the light-cone transform. *Nature*, 2018.

[11] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[12] F. Quercioli and G. Molesini. White light-in-flight holography. *Applied optics*, 24(20):3406–3415, 1985.

[13] G. Satat, M. Tancik, O. Gupta, B. Heshmat, and R. Raskar. Object classification through scattering media with deep learning on time resolved measurement. *Optics express*, 25(15):17466–17479, 2017.

[14] A. Velten, R. Raskar, and M. Bawendi. Picosecond camera for time-of-flight imaging. In *Imaging Systems and Applications*, page IMB4. Optical Society of America, 2011.

[15] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications*, 3:745, 2012.