Bayesian methods for inverse problems with point clouds: applications to single-photon lidar

Julián Andrés Tachella

Electrical, Electronic and Computer Engineering Department
School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, United Kingdom

Under the supervision of

Prof. Stephen McLaughlin
Dr. Yoann Altmann
Prof. Jean-Yves Tourneret

A Thesis Submitted for the Degree of
Doctor of Philosophy (PhD) in Electrical Engineering

· November 2019 ·

Abstract

Single-photon light detection and ranging (lidar) has emerged as a prime candidate technology for depth imaging through challenging environments. This modality relies on constructing, for each pixel, a histogram of time delays between emitted light pulses and detected photon arrivals. The problem of estimating the number of imaged surfaces, their reflectivity and position becomes very challenging in the low-photon regime (which equates to short acquisition times) or relatively high background levels (i.e., strong ambient illumination).

In a general setting, a variable number of surfaces can be observed per imaged pixel. The majority of existing methods assume exactly one surface per pixel, simplifying the reconstruction problem so that standard image processing techniques can be easily applied. However, this assumption hinders practical three-dimensional (3D) imaging applications, being restricted to controlled indoor scenarios. Moreover, other existing methods that relax this assumption achieve worse reconstructions, suffering from long execution times and large memory requirements.

This thesis presents novel approaches to 3D reconstruction from single-photon lidar data, which are capable of identifying multiple surfaces in each pixel. The resulting algorithms obtain new state-of-the-art reconstructions without strong assumptions about the sensed scene. The models proposed here differ from standard image processing tools, being designed to capture correlations of manifold-like structures.

Until now, a major limitation has been the significant amount of time required for the analysis of the recorded data. By combining statistical models with highly scalable computational tools from the computer graphics community, we demonstrate 3D reconstruction of complex outdoor scenes with processing times of the order of 20 ms, where the lidar data was acquired in broad daylight from distances up to 320 m. This has enabled robust, real-time target reconstruction of complex moving scenes, paving the way for single-photon lidar at video rates for practical 3D imaging applications.

Acknowledgements

I would like to start by thanking my PhD supervisors: Yoann, Jean-Yves and Steve. Before starting the PhD, someone told me that the most important point when looking for a PhD was to find a good supervisor. At the end of my PhD, I can now say that I was lucky to find not only one good supervisor, but three great supervisors (and persons). I have learnt a very unique and distinct set of skills from each of you. Most importantly, thanks to all your support and guidance, I will always keep a beautiful memory of my PhD years.

I would also like to thank all the researchers that I have met during my PhD and have contributed in some way or another to this work: Marcelo Pereyra, Rachael Tobin, Aongus McCarthy, Aurora Maccarone, Gerald S. Buller, Miguel Márquez, Henry Arguello, Nicolas Mellado, Joshua Rapp, John Murray-Bruce, Charles Sauders and Vivek K. Goyal.

Finally, I would like to thank my family, who has supported me not only throughout this PhD, but also in every step that got me to this point. It is hard to realise how lucky I am to have such an unconditional support and endless motivation to never stop learning. Last but not least, to Pauline for her continuous off-stage support which also made this possible.

Research Thesis Submission

Please note this form should be bound into the submitted thesis.

Name:	Julián Andrés Tachella		
School:	Engineering and Physical Sciences		
Version: *(i.e. First, Resubmission, Final)*	First	Degree Sought:	PhD in Electrical Engineering

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

ONLY for submissions including published works

Please note you are only required to complete the Inclusion of Published Works Form (page 2) if your thesis contains published works)

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Limited Access	Requested	Yes		No		Approved	Yes		No	
E-thesis Submitted ***(mandatory for final theses)***										
Received in the SSC by *(name in capitals):*						Date:				

Publications related to the PhD thesis

International Journal Papers

- [**Tachella et al. 2019a**] **J. Tachella**, Y. Altmann, X. Ren, A. McCarthy, G. S. Buller, J.-Y. Tourneret and S. McLaughlin "Bayesian 3D reconstruction of complex scenes from single-photon lidar data", *SIAM Journal on Imaging Sciences*, vol. 12, no. 1, pp. 512-550, March 2019.

- [**Tachella et al. 2019b**] **J. Tachella**, Y. Altmann, M. Márquez, H. Arguello-Fuentes, J.-Y. Tourneret and S. McLaughlin "Bayesian 3D reconstruction of subsampled multispectral single-photon lidar signals", *IEEE Transactions on Computational Imaging (Early Access)*, September 2019.

- [**Tachella et al. 2019c**] **J. Tachella**, Y. Altmann, N. Mellado, R. Tobin, A. McCarthy, G. S. Buller, J.-Y. Tourneret and S. McLaughlin. "Real-time 3D reconstruction from single-photon data using plug-and-play point cloud denoisers", *Nature Communications*, vol. 10, pp. 4984, November 2019.

- [**Rapp et al. 2019d**] J. Rapp, **J. Tachella**, Y. Altmann, S. McLaughlin and V. K. Goyal, "Advances in single-photon lidar for autonomous vehicles", *to appear in IEEE Signal Processing Magazine*, 2020.

- [**Rapp et al. 2019e**] J. Rapp*, C. Saunders*, **J. Tachella***, J. Murray-Bruce, Y. Altmann, J-Y. Tourneret, S. McLaughlin, R. Dawson, F. Wong and V. K. Goyal "Seeing around corners with edge-resolved transient imaging", *Arxiv*, 2020. *The first 3 authors have contributed equally to the paper.

International Conference Papers

- [**Tachella et al. 2018**] **J. Tachella**, Y. Altmann, M. Pereyra, S. McLaughlin and J.-Y. Tourneret "Bayesian restoration of high-dimensional photon-starved images", in *Proc. 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018.

- [**Tachella et al. 2019f**] **J. Tachella**, Y. Altmann, S. McLaughlin and J.-Y. Tourneret "3D reconstruction using single-photon lidar data: Exploiting the widths of the returns", in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019.

- [**Tachella et al. 2019g**] **J. Tachella**, Y. Altmann, S. McLaughlin and J.-Y. Tourneret "On fast object detection using single-photon lidar data", in *Proc. SPIE Wavelets and Sparsity XVIII*, San Diego, USA, August 2019.

- [**Tachella et al. 2019h**] **J. Tachella**, Y. Altmann, S. McLaughlin and J.-Y. Tourneret "Fast surface detection in single-photon lidar waveforms", in *Proc. 27th European Signal Processing Conference (EUSIPCO)*, La Coruña, Spain, September 2019.

- [**Tachella et al. 2019i**] **J. Tachella**, Y. Altmann, S. McLaughlin and J.-Y. Tourneret "Real-time 3D color imaging with single-photon lidar data", in *Proc. International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Guadaloupe, West Indies, December 2019.

Acronyms and notation

Acronyms

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
APSS	Algebraic point set surfaces
ADMM	Alternating direction method of multipliers
CPU	Central processing unit
CNN	Convolutional neural network
DAG	Directed acyclic graph
FFT	Fast Fourier transform
FNR	False negative rate
FPR	False positive rate
GMRF	Gaussian Markov random field
GPU	Graphics processing unit
IRF	Impulse response function
MCMC	Markov chain Monte Carlo
MAP	Maximum a posteriori
DAE	Mean depth absolute error
IAE	Mean intensity absolute error
PPP	Mean photons per pixel
MSE	Mean squared error
MMSE	Minimum mean squared error
MSL	Multispectral single-photon lidar
NMSE	Normalised mean squared error

PD	Probability of detection
PFA	Probability of false alarm
RAM	Random access memory
RGB	Red, green and blue
RGB-D	Red, green, blue and depth
RJ-MCMC	Reversible jump Markov chain Monte Carlo
ROC	Receiver operating characteristic
SBR	Signal-to-background ratio
SPAD	Single-photon avalanche diode
SPL	Single-photon lidar
TCSPC	Time-correlated single-photon counting
TNR	True negative rate
TPR	True positive rate
TV	Total variation

Mathematical notation

x	A scalar quantity
$\boldsymbol{x}$	A vector
$\boldsymbol{X}$	A matrix
$\mathbb{R}$	Set of all real numbers
$\mathbb{Z}$	Set of all integers
$\mathbb{R}^d$	Euclidean d-space
$\|\|\cdot\|\|$	A norm
$\mathbb{1}_A(\cdot)$	Indicator function over the set A

Probability

$\mathcal{B}e$	Bernoulli distribution
$\mathcal{B}$	Binomial distribution
δ	Dirac delta distribution
$\mathcal{G}$	Gamma distribution
$\mathcal{N}$	Gaussian distribution

$\mathcal{P}$	Poisson distribution
π	Poisson random measure
$\mathbb{E}$	Expectation operator
N_i	Number of MCMC/optimisation iterations
N_{bi}	Number of burn-in iterations

Single-photon lidar

$\boldsymbol{Z}$	Lidar data
$\boldsymbol{z}_{i,j}$	Lidar histogram at pixel (i, j)
$z_{i,j,t}$	Photon count at pixel (i, j) and histogram bin t
$z_{i,j,\ell,t}$	Photon count at pixel (i, j), wavelength ℓ and histogram bin t
N_r	Number of pixel rows
N_c	Number of pixel columns
T	Number of histogram bins
L	Number of spectral bands/wavelengths
W	Number of measured spectral bands/wavelengths per pixel
$h(t)$	Impulse response function
$\boldsymbol{\Phi}$	Unordered set of points
$N_{\boldsymbol{\Phi}}$	Number of points
$\boldsymbol{c}_n$	Coordinate of nth point
$\boldsymbol{b}$	Vector of background levels
$\boldsymbol{B}$	Matrix of multispectral background levels
$b_{i,j}$	Background level at pixel (i, j)
$\tilde{\boldsymbol{b}}$	Vector of log-background levels
$\tilde{b}_{i,j}$	Log-background level at pixel (i, j)
$\boldsymbol{r}$	Vector of intensities
r_n	Intensity of nth point
$\boldsymbol{m}$	Vector of intensities
m_n	Log-intensity of nth point
$\boldsymbol{\Psi}$	Set of hyperparameters
w	Signal-to-background ratio
η_n	Broadening of the impulse response of the nth point

$\tilde{\eta}_n$	Log-broadening of the impulse response of the nth point
$\boldsymbol{P}$	Precision matrix of a multivariate Gaussian distribution
Δ_t	TCSPC timing resolution
Δ_b	Lidar bin width
Δ_p	Approximate spatial resolution of one pixel
T_a	Mean number of active (non-zero) histogram bins
T_h	Number of non-zero histogram bins in the support of $h(t)$
N_p	Binning window size (pixels)

Contents

Chapter 1

Introduction

Contents

1.1 Aims and objectives of the thesis

Reconstruction of three-dimensional (3D) scenes has many important applications, such as autonomous navigation [1], environmental monitoring [2] and other computer vision tasks [3]. While geometric and reflectivity information can be acquired using many scanning modalities (e.g., RGB-D sensors [4], stereo imaging [5] or full waveform lidar [2]), single-photon systems have emerged in recent years as an excellent candidate technology. The time-correlated single-photon counting (TCSPC) lidar approach offers several advantages: the high sensitivity of single-photon detectors allows for the use of low-power, eye-safe laser sources; and the picosecond timing resolution enables excellent surface-to-surface resolution at long range (hundreds of metres to kilometres) [6]. Recently, the TCSPC technique has proved successful at reconstructing high resolution 3D images

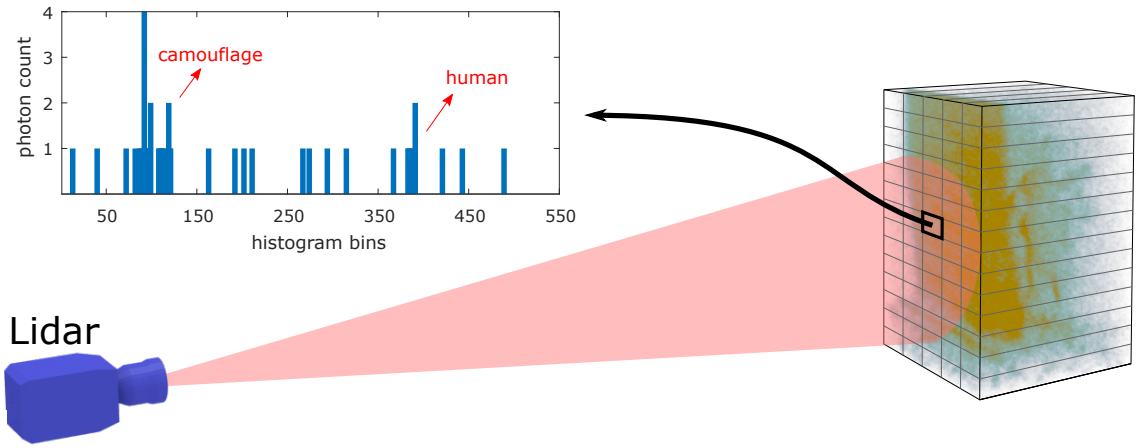


Figure 1.1 Illustration of a single-photon lidar dataset. The dataset consists of a man behind a camouflage net [10]. The graph on the left shows the histogram of a given pixel with two surfaces. The limited number of collected photons and the large number of spurious detections linked to the ambient illumination makes the reconstruction task very challenging. In this case, processing the pixels independently yields poor results, but they can be improved by considering a priori knowledge about the scene's structure.

in extreme environments such as through fog [7], with cluttered targets [8], in highly scattering underwater media [9], and in free-space at ranges greater than 10 km [6].

A TCSPC lidar system constructs a histogram of time delays between emitted and reflected pulses for each pixel. The presence of an object is associated with a characteristic distribution of photon counts in the histogram, as shown in Fig. 1.1. The position and number of detections provide depth and reflectivity information respectively. In scenarios where the light goes through a semitransparent material (e.g, windows or camouflage) or when the laser beam is wide enough with respect to the object size (e.g., distant objects), it is possible to record two or more surfaces in a single pixel.

Despite its remarkable ranging capabilities, this modality suffers from two inherent disadvantages: first, the number of photons coming back from the target of interest can be very small, as it is limited by the amount of laser power and acquisition time. Secondly, the recorded histograms contain spurious (non-informative) detections, due to background illumination sources (e.g., the sun). Signal processing methods tackle these limitations by exploiting prior knowledge on the scene to recover. In particular, reconstruction algorithms attempt to recover high-quality 3D point clouds using as few informative photons as possible. This task is challenging for several reasons:

- The photon detections follow Poisson statistics, hindering any direct use of standard signal processing methods designed for Gaussian noise.
- In scenarios with strong ambient illumination, the number of photons related to the target can be significantly smaller than the ones due to background illumination.
- Meaningful and well-defined models that capture correlations of 3D point clouds are difficult to construct.

- Making an all-encompassing algorithm that can handle varying sensing scenarios (such as those explained in Section 1.2.2) is a difficult task.
- The algorithms have to be able to process very large volumes of data while providing fast estimates for any posterior decision making.

While many methods have been proposed over the last decades, they impose restrictive assumptions on the scene to recover, hindering any practical deployment of single-photon lidar technology. In particular, most methods assume the presence of exactly one surface per pixel, as it greatly simplifies the reconstruction problem. However, this assumption does not hold in many real-world scenarios, being applicable only to indoor scenes. Moreover, they require execution times of the order of seconds or even minutes, prohibiting real-time decision making, which is essential in many important applications (e.g., self-driving cars). The goal of this thesis is to present new signal processing algorithms that remove these restrictive assumptions, paving the way for real-world deployment of single-photon lidar technology.

The rest of this chapter is organised as follows: Section 1.2 describes the basic working principles of single-photon lidar systems, identifying the main challenges. Section 1.3 summarises existing approaches to 3D reconstruction. The main contributions of this thesis are outlined in Section 1.4. Finally, Section 1.5 details the organisation of the thesis.

1.2 Single-photon lidar

1.2.1 Working principles

A single-photon lidar system comprises 3 main components: an illumination source, a single-photon detector and fast timing electronics, as shown in Fig. 1.2. The illumination source is generally composed of a diode laser, which can achieve root mean square pulse widths of the order of a few dozens of picoseconds. The most common choice of detector is the solid-state single-photon avalanche diode (SPAD), which consists of a reverse-biased photodiode biased above the breakdown voltage so that an individual photon incident on the SPAD can cause an avalanche of electrical charge carriers that is directly detectable as a digital signal.

The basic idea of TCSPC is that of a stopwatch: the laser starts a timer with each illumination pulse, and the timer is stopped with the detection of a photon. The time difference between the stop and start signals gives the photon's time-of-flight, which is easily converted to a measurement of the round-trip distance through multiplication by the speed of light. Due to timing uncertainty and the presence of nuisance detections of ambient light, illumination is repeated to build up a histogram of photon detection times, from which more reliable depth estimates can be determined.

Single-photon lidar systems have conventionally employed raster-scanned illumination, as the one illustrated in Fig. 1.2. A laser aimed at one spot in a scene repeatedly pulses for a certain

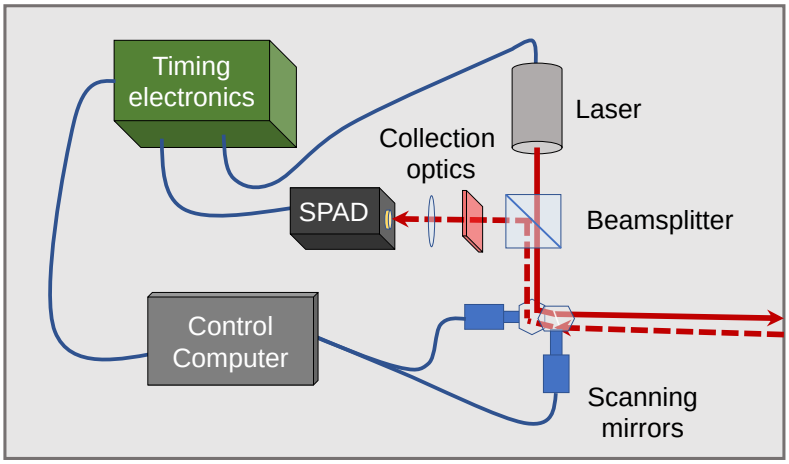


Figure 1.2 An example of a single-photon lidar system with raster-scanned confocal configuration. The laser illuminates one point in the environment at a time, directed by a pair of scanning mirrors. Light reflecting back from illuminated surfaces is directed towards a single-photon-sensitive SPAD detector. The time difference between the illumination and photon detection is recorded and processed.

dwell time before being redirected to the next spot by a pair of XY galvo mirrors [11]. Scanned illumination enables the use of a single-pixel or bucket detector, which is often in a confocal configuration (focused and co-axially aligned with the laser) to limit the number of photons undergoing multiple bounces or originating from ambient sources from being detected. The main drawback of this approach is the long time required to serially scan all the pixels.

A more recent approach has been to broadly illuminate a swath of the scene and achieve spatial resolution with an array of single-photon sensitive elements [12]. Detector arrays offer faster and parallelised acquisitions at the expense of lower depth resolution. The laser power is broadly diffused over a larger area, reducing the signal strength received at each pixel location.

A compromise between raster-scanning and array systems is the use of a line illumination and line array detectors. This alternative reduces the spatial scanning to a single dimension, while limiting the diffusion of the laser power [13].

1.2.2 Challenging sensing scenarios

The main challenges encountered in single-photon lidar waveforms are detailed below and illustrated in Fig. 1.3.

Few photons The number of detected photons may be small or even zero for several reasons: the number of illuminations is kept low for fast acquisition, the surface reflects very little light because it is weakly-reflective or far away, the detection efficiency is low, etc. Hence, photon-efficient systems work with exactly one photon per pixel (PPP) [14] or an average near 1 PPP [15, 16], resulting in many pixels with no detections.

Strong ambient light Signal estimation is particularly challenging if the ratio between the number of photons due to the laser and ambient illuminations, referred to as the signal-to-background ratio (SBR), is low. Even though optical methods (e.g., confocal configurations, bandpass filters) are used to limit the amount of ambient light that reaches the detector, strong daylight, especially when combined with a weak surface reflection, can result in far more detection events associated with background photons than from signal photons.

Absence of surfaces The most basic 3D reconstruction methods assume a single surface at each pixel location. If a pixel has no object in its line of sight (e.g., outdoor scenes), then the histogram contains only background detection events.

Multiple surfaces There may be reflections from multiple surfaces present at one pixel (e.g., Fig. 1.1). This may occur because the light passes through a semi-transparent material such as glass. Alternatively, the pixel size or field of view increases with distance (e.g., due to the laser divergence in a scanned setting), so the spot is more likely to cover multiple surfaces. This same principle is often used in foliage-penetrating airborne lidar used for terrain mapping [17, 18].

Pulse broadening Surfaces are generally assumed to be opaque and approximately normal to the illumination beam so that the reflected temporal response does not change across the imaged scene. However, sub-surface scattering or oblique-angled surfaces, especially at long distances (e.g., kilometres), return broadened pulse profiles, whose shape vary across measured pixels.

Attenuating media Particles in the beam path, such as fog, smoke, rain, or snow, affect the acquired light by scattering photons in different directions after both the illumination (forward path) and reflection (return path). To some extent, the result is similar to that of a signal weakened by additional attenuation and increased background due to scattered photons [7], although the near-range effects of scattering also reshape the temporal distribution of background, with more detections at earlier times [19]. Similar effects are also encountered for lidar in underwater environments [9].

Coarse time quantisation The ability to accurately resolve transient information depends on the width of the histogram bins. For raster-scanning systems, the bin resolution that can be achieved currently is of the order of picoseconds, which is typically much less than the duration of laser pulse, so quantisation effects on the depth estimation are negligible. However, the timing resolution of detector arrays is usually coarser for each element than for a single-pixel device due to hardware and readout constraints. The single-photon-sensitive elements and timing electronics can easily be constructed as separate elements for a single pixel, whereas in 2D arrays, the timing electronics must be integrated on-chip for each pixel, resulting in a trade-off between the fill factor

of the photo-sensitive detector and timing components. Poor depth resolution due to quantisation effects can make object detection and recognition more difficult (see Fig. 1.3h).

Dead time effects Unfortunately, the design of single-photon detectors makes it impossible to register every photon reaching the sensor. The circuitry required for single-photon sensitivity often precludes the ability to resolve numbers of photons, so only a single detection event can be registered even if multiple photons arrive at the detector simultaneously. In addition, the detector has a reset period known as a dead time following each detection in order to become single-photon sensitive again, during which no further photons can be registered.

One of the main implications of the dead time is that the sequence of detection times can no longer be described by a Poisson process: whether a photon will be detected now depends on the time of the most recent detection. Dead times are thus much more significant in the high-flux regime when the probability of a photon coming back per illumination event is very high, e.g., when imaging bright objects such as retro-reflective street signs. The dead time effect causes distortions in the detection time histogram, which may result in erroneous depth and reflectivity estimates, thereby making accurate localisation or object recognition more difficult. The simplest way to avoid dead time distortions is to attenuate the incident light, so that the probability of a photon arriving during a dead time is very low and the effect becomes negligible.

Dead and hot pixels Another current limitation of array manufacturing constraints is spatial non-uniformity. While raster-scanning imaging with a single-pixel detector has essentially identical system properties for each laser location, array elements have neither the same light sensitivity nor identical noise characteristics across the device. In particular, arrays often present "hot" pixels with overwhelming numbers of dark counts, or "dead" pixels with inadequate light sensitivity. The inputs from these pixels must then be omitted, or at least accounted for, in the reconstruction process.

1.2.3 Observation model

A lidar data cube of $N_r \times N_c$ pixels and T histogram bins is denoted by $\boldsymbol{Z}$, where the photon-count recorded in pixel (i, j) and histogram bin t is $[\boldsymbol{Z}]_{i,j,t} = z_{i,j,t} \in \mathbb{Z}_+ = \{0, 1, 2, \dots\}$, with $i = 1, \dots, N_r$, $j = 1, \dots, N_c$ and $t = 1, \dots, T$. The 3D reconstruction task consists of estimating a 3D point cloud from the measurements $\boldsymbol{Z}$. We represent the point cloud by a set of $N_{\boldsymbol{\Phi}}$ points $\boldsymbol{\Phi} = \{(\boldsymbol{c}_n, r_n) \quad n = 1, \dots, N_{\boldsymbol{\Phi}}\}$, where $\boldsymbol{c}_n = [x_n, y_n, t_n]^\mathsf{T}$ is the point location with $x_n \in \{1, \dots, N_r\}$, $y_n \in \{1, \dots, N_c\}$ and $t_n \in [0, T]$[1], and $r_n \in \mathbb{R}_+$ is the intensity (unnormalised reflectivity) of the point. For ease of presentation, we also denote the set of lidar depths values by $\boldsymbol{t} = [t_1, \dots, t_{N_\Phi}]^\mathsf{T}$ and the set of intensity values by $\boldsymbol{r} = [r_1, \dots, r_{N_\Phi}]^\mathsf{T}$.

[1] Some algorithms assume that the depth location t_n is continuous, whereas other methods discretise t_n according to the histogram binning, i.e., $t_n \in \{1, \dots, T\}$

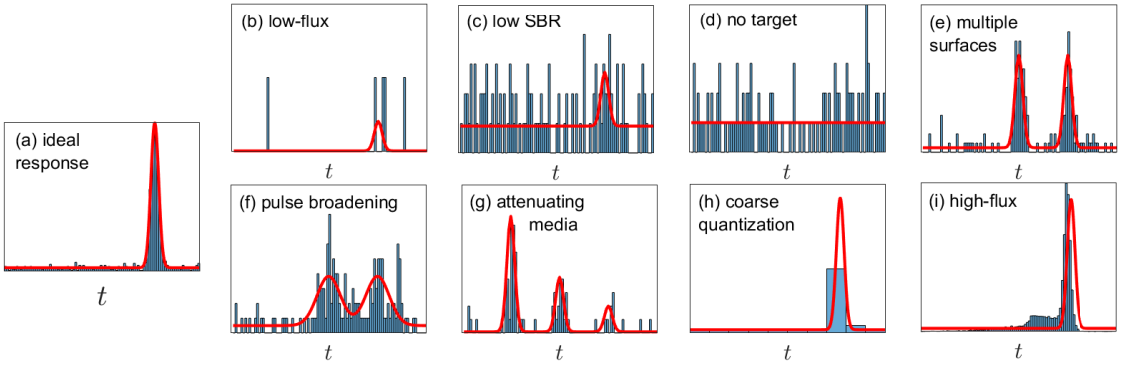


Figure 1.3 Examples of recorded histograms: (a) ideal case, (b) few photons, (c) strong background illumination, (d) absence of a target, (e) multiple surfaces per imaged pixel, (f) broadening of the impulse response, (g) highly attenuating media, (h) coarse quantisation, and (i) dead-time effects. The observed photon counts are shown in blue, whereas the underlying Poisson intensity (1.1) is shown in red.

If the light-flux reaching the single-photon detector is sufficiently low (i.e., detector dead time effects can be neglected), the observed photon count in bin t and pixel (i, j) follows a Poisson distribution, whose intensity is a mixture of the pixel background level $b_{i,j}$ and the responses of the surfaces present in that pixel, that is

$$z_{i,j,t}|\boldsymbol{\Phi}, b_{i,j} \sim \mathcal{P}\left(\sum_{\mathcal{N}_{i,j}} r_n h(t - t_n) + b_{i,j}\right) \tag{1.1}$$

where $t \in \{1, ..., T\}$, $h(\cdot) : \mathbb{R} \mapsto \mathbb{R}_+$ is the known temporal instrumental response and $\mathcal{N}_{i,j} = \{n : (x_n, y_n) = (i, j)\}$ is the set of points in pixel (i, j).Note that the impulse response of the system might vary across pixels in lidar arrays, and it can also appear broadened when measuring long distances. In this chapter, $h(t)$ is assumed to be fixed to simplify the presentation. Assuming mutual independence between the Poisson random variables in different time bins and pixels, the full likelihood can be written as

$$p(\boldsymbol{Z}|\boldsymbol{\Phi}, \boldsymbol{b}) = \prod_{i=1}^{N_r} \prod_{j=1}^{N_c} \prod_{t=1}^{T} p(z_{i,j,t}|\boldsymbol{\Phi}, b_{i,j}) \tag{1.2}$$

where $\boldsymbol{b} = [b_{1,1}, \ldots, b_{i,j}, \ldots, b_{N_r,N_c}]^\mathsf{T} \in \mathbb{R}_+^{N_r N_c}$ is the vectorised set of background levels. Note that $p(z_{i,j,t}|\boldsymbol{\Phi}, b_{i,j})$ in (1.2) is the Poisson distribution associated with (1.1).

1.2.4 Inverse problem formulation

The reconstruction task can be formulated as an inverse problem, where the aim is to recover the unknown scene parameters $\boldsymbol{\Phi}$ and $\boldsymbol{b}$ that generated the measurements $\boldsymbol{Z}$. This task is an ill-posed inverse problem due to the random noise affecting the measurements, the depth uncertainty related to the broad impulse response and coarse depth binning, and other potentially missing data issues,

such as the presence of hot or dead pixels. Hence, it is necessary to promote the solution to be in a set of plausible parameters, which can be defined using prior knowledge about them. In the context of Bayesian statistics, this knowledge is incorporated via prior distributions assigned to the unknown parameters. Following Bayes rule, the posterior distribution satisfies

$$p(\boldsymbol{\Phi}, \boldsymbol{b}|\boldsymbol{Z}, \boldsymbol{\Psi}) = \frac{1}{C(\boldsymbol{Z})} p(\boldsymbol{Z}|\boldsymbol{\Phi}, \boldsymbol{b}) p(\boldsymbol{\Phi}, \boldsymbol{b}|\boldsymbol{\Psi}) \tag{1.3}$$

where $\boldsymbol{\Psi}$ is a set of fixed hyperparameters, $p(\boldsymbol{Z}|\boldsymbol{\Phi}, \boldsymbol{b})$ is given by the observation model (1.2), and $p(\boldsymbol{\Phi}, \boldsymbol{b}|\boldsymbol{\Psi})$ is the prior distribution for the point cloud and background levels. The normalisation constant $C(\boldsymbol{Z})$ is defined as

$$C(\boldsymbol{Z}) = \int \int p(\boldsymbol{Z}|\boldsymbol{\Phi}, \boldsymbol{b}) p(\boldsymbol{\Phi}, \boldsymbol{b}|\boldsymbol{\Psi}) d\boldsymbol{\Phi} d\boldsymbol{b} \tag{1.4}$$

and it is generally intractable due to the high dimensional integrals. The unknown parameters $\boldsymbol{\Phi}$ and $\boldsymbol{b}$ can be estimated by computing statistics of (1.3), that is

$$(\widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{b}}) = \mathbb{E}\{f(\boldsymbol{\Phi}, \boldsymbol{b})\} \tag{1.5}$$

where the expectation is taken with respect to the posterior distribution of (1.3) and $f(\cdot)$ is some function chosen to minimise a certain risk [20]. For example, the marginal posterior mean corresponds to the mean minimum squared error (MMSE) estimator of the unknown parameters. In general, the high dimensional integrals of (1.5) are not available in closed form, and they are approximated using Markov chain Monte Carlo (MCMC) methods [21] or other variational approximations (e.g., expectation propagation [22] or variational Bayes [23]). A commonly chosen point estimator is the maximum a posteriori (MAP) estimator (also known as penalised maximum likelihood), that is

$$(\widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{b}}) = \underset{\boldsymbol{\Phi}, \boldsymbol{b}}{\arg\max} \log p(\boldsymbol{\Phi}, \boldsymbol{b}|\boldsymbol{Z}, \boldsymbol{\Psi}) \tag{1.6}$$

which can be computed using optimisation techniques [24], avoiding to resort to MCMC methods.

The computational methods presented throughout this thesis will use MCMC (Chapters 2 and 3), variational approximations (Chapter 3), numerical integration (Chapter 4) and optimisation techniques (Chapter 5) to solve the 3D reconstruction problem. A more detailed explanation about these inference machines is included in each chapter.

1.3 Existing approaches

The single-photon lidar literature contains a wide variety of 3D reconstructions algorithms, differing both in the assumptions about the signal model (1.2) and the regularisation assigned to

the unknown parameters. We distinguish three main families of algorithms. The first group of methods assumes exactly one object per pixel, reducing the 3D reconstruction problem to the estimation of depth, reflectivity and background images. The second group, assumes at most one object per pixel, where some pixels may not contain objects, The third group of algorithms, namely multi-depth methods, relax these assumptions and attempt to infer a more general 3D point cloud, relying on priors defined in 3D space.

1.3.1 Single-depth algorithms

In the case of a single surface per pixel, the reconstruction problem reduces to estimating a depth, intensity and background level per imaged pixel. The 3D point cloud is replaced by depth and intensity images. In terms of notation, the point cloud $\boldsymbol{\Phi}$ can be expressed as two vectorised images, $\boldsymbol{t} = [t_{1,1}, \ldots, t_{i,j}, \ldots, t_{N_r,N_c}]^\mathsf{T} \in [1,T]^{N_r N_c}$ and $\boldsymbol{r} = [r_{1,1}, \ldots, r_{i,j}, \ldots, r_{N_r,N_c}]^\mathsf{T} \in \mathbb{R}_+^{N_r N_c}$, where each element corresponds to the depth and intensity of a given pixel. Under this assumption, the negative log-likelihood function $g(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}) \propto -\log p(\boldsymbol{Z}|\boldsymbol{\Phi}, \boldsymbol{b})$ is

$$g(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}) = \sum_{i=1}^{N_r} \sum_{j=1}^{N_c} \sum_{t=1}^{T} r_{i,j} h(t - t_{i,j}) + b_{i,j} - z_{i,j,t} \log\left(r_{i,j} h(t - t_{i,j}) + b_{i,j}\right). \tag{1.7}$$

A standard estimation procedure is based on the maximum likelihood estimator (MLE), which determines the set of parameters that minimise the negative log-likelihood, i.e.,

$$(\hat{\boldsymbol{t}}, \hat{\boldsymbol{r}}, \hat{\boldsymbol{b}}) = \underset{\boldsymbol{t},\boldsymbol{r},\boldsymbol{b}}{\arg\min}\, g(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}). \tag{1.8}$$

As problem (1.8) is non-convex and often presents multiple minima, the background levels are assumed to be either known from a calibration procedure or negligible ($\boldsymbol{b} = \boldsymbol{0}$) to simplify the problem. Under the assumption of fixed $\boldsymbol{b}$ and $h(t - t_{i,j})$ negligible at the extremes of the histogram ($\sum_{t=1}^{T} r_{i,j} h(t - t_{i,j}) \approx r_{i,j}$), an approximate (non-iterative) solution of (1.8) is computed as follows:

- Discarding depth information (integrating photons across histogram bins), the intensity can be computed in closed form as

$$\widehat{r}_{i,j} = \max\left(0, \frac{-b_{i,j} T + \sum_{t=1}^{T} z_{i,j,t}}{\sum_{t=1}^{T} h(t)}\right) \tag{1.9}$$

  for each pixel (i, j) in the lidar data cube.

- The depth estimator $\widehat{\boldsymbol{t}}$ is given by cross-correlating the detection time histogram with the logarithm of $h(t)$, also known as the log-matched filter:

$$\widehat{t}_{i,j} = \underset{\tau \in [1,T]}{\arg\max} \sum_{t=1}^{T} z_{i,j,t} \log\left[\widehat{r}_{i,j} h(t - \tau) + b_{i,j}\right] \tag{1.10}$$

for each pixel (i, j).

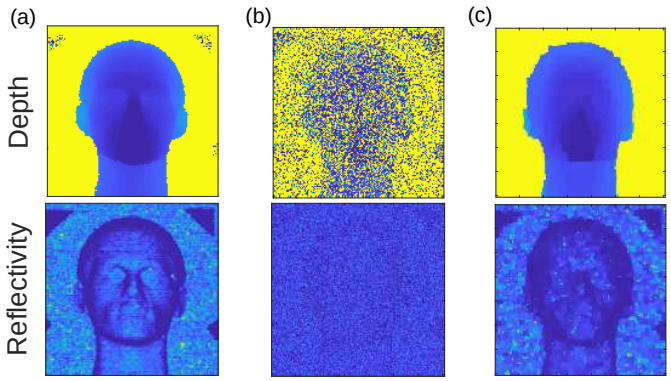


Figure 1.4 Example of a dataset containing one surface per pixel. The lidar dataset has approximately 3 PPP and an SBR of 0.5. The ground truth depth and reflectivity images are shown in (a). Depth and reflectivity estimates obtained with cross-correlation and a state-of-the-art single-depth algorithm [25] are shown in (b) and (c) respectively. As the number of recorded photons is very low, cross-correlation gives poor estimates, whereas algorithms using regularisation techniques obtain better reconstructions.

When the number of photons per pixel is low or the background levels are not negligible, the MLE does not provide reliable estimates, as illustrated in Fig. 1.4. The estimation can be improved by removing background photons with a pre-processing step and incorporating a priori information on the structure of $\boldsymbol{r}$ and $\boldsymbol{t}$, as explained in Section 1.2.4. In the case of MAP estimation, the reconstruction problem is written as

$$(\hat{\boldsymbol{t}}, \hat{\boldsymbol{r}}) = \underset{\boldsymbol{t},\boldsymbol{r}}{\arg\min}\, g\left(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b} = \boldsymbol{0}\right) + \lambda_{\boldsymbol{t}}\rho_{\boldsymbol{t}}(\boldsymbol{t}) + \lambda_{\boldsymbol{r}}\rho_{\boldsymbol{r}}(\boldsymbol{r}) \tag{1.11}$$

where $\lambda_{\boldsymbol{t}}\rho_{\boldsymbol{t}}(\boldsymbol{t})$ and $\lambda_{\boldsymbol{r}}\rho_{\boldsymbol{r}}(\boldsymbol{r})$ correspond to the negative log-prior distributions of the depth and intensity. The scalar hyperparameters $\lambda_{\boldsymbol{t}}$ and $\lambda_{\boldsymbol{r}}$ control the amount of spatial regularisation of the depth and intensity respectively. If the impulse response $h(t)$ is log-concave and convex regularisation terms $\rho_{\boldsymbol{t}}(\boldsymbol{t})$ and $\rho_{\boldsymbol{r}}(\boldsymbol{r})$ are chosen, problem (1.11) is convex and has a unique minimiser. However, in contrast to the standard Gaussian noise case, the objective (1.11) does not have globally Lipschitz gradient due to the data fidelity term g (with respect to $\boldsymbol{r}$). The solution of (1.11) can thus be obtained via optimisation programs that do not require global Lipschitz differentiability, such as SPIRAL [26] or PIDAL [27]. SPIRAL extends the standard proximal gradient scheme [28] with an adaptive step size that ensures convergence when the objective is only locally Lipschitz differentiable. In contrast, PIDAL is based on the alternating direction method of multipliers (ADMM) algorithm, which does not require the objective to be globally Lipschitz differentiable to ensure convergence [29].

Most of the single-depth algorithms [10, 12, 14–16, 30] propose a total variation regularisation (TV) for the depth and intensity, i.e., $\rho_{\boldsymbol{t}}(\boldsymbol{t}) = ||\boldsymbol{t}||_{\text{TV}}$ and $\rho_{\boldsymbol{r}}(\boldsymbol{r}) = ||\boldsymbol{r}||_{\text{TV}}$, where $||\cdot||_{\text{TV}}$ is defined as the ℓ_1 norm of the horizontal and vertical pixel differences [24]. TV regularisation is a

standard choice in the image processing literature, as it is a convex penalty that promotes piecewise constant images. All the proposed methods mostly differ in the background rejection step: Some methods rely solely on a ranked order mean filter [14, 15], and estimate $\boldsymbol{r}$ and $\boldsymbol{t}$ separately by first computing the intensities $\boldsymbol{r}$ from accumulated histograms (no depth information). Halimi et al. [10] proposed to estimate them jointly using PIDAL, but did not use a background rejection step. The more recent algorithm by Rapp and Goyal [25] uses an adaptive super-pixel approach to censor background photons and improve depth and reflectivity estimates in an iterative manner.

A slightly different approach was taken by Altmann et al. [16]. While similar regularisation terms were used, the algorithm estimates the background levels jointly with $\boldsymbol{r}$ and $\boldsymbol{t}$. To overcome the non-convexity of the problem, an MCMC sampling approach was used, which is ensured to visit the local maxima of the posterior distribution. This method computes MMSE estimates instead of marginal MAP estimates. Moreover, the hyperparameters can be also estimated within the Markov chain, avoiding the parameter tuning of other optimisation alternatives. However, this method suffers from very long execution times (dozens of minutes per lidar frame) due to the sequential nature and slow convergence of the MCMC algorithm.

The observation model (1.1) can be reformulated as a set of data points (time-tagged photons) generated from a mixture between a discrete uniform distribution (background photons) and a discrete distribution with shape $h(t - t_{i,j})/\sum_{t=1}^{T} h(t)$ [31, 32]. This representation is useful for deriving expectation maximisation (EM) algorithms [33], where the photon sources (background or signal) are the unobserved latent variables, and the depth $t_{i,j}$ is the parameter of interest. Despite the different observation model and inference approach, the proposed algorithms also rely on TV regularisation of $\boldsymbol{t}$ [31] and obtain similar results than other single-depth alternatives.

Finally, a different path is taken by the authors of [34], who use an end-to-end convolutional neural network (CNN) to estimate depth and reflectivity images. This method benefits from an additional high resolution 2D image, which is also injected into the CNN to improve the detail of the reconstructions. However, the performance achieved by this method was not significantly better than TV-based methods, such as [25].

To finish this part, we would like to mention that the single-depth assumption can be too restrictive for practical implementation of lidar systems, as multiple sensing scenarios contain a variable number of surfaces per pixel.

1.3.2 Target detection algorithms

Target detection algorithms focus on cases where at most one surface is present, which encompass a wide range of practical scenes. In this setting, simply thresholding the reflectivity estimates obtained by a single-depth algorithm is generally not robust to background illumination. Hence, specific target detection algorithms estimate an additional binary image indicating the per-pixel presence or absence of a target. Few algorithms have been proposed to tackle this problem. The

work by Altmann et al. [35] promotes spatial correlations in the detection map image using an Ising model, and inference is performed using a reversible jump MCMC algorithm (RJ-MCMC). As in [16], the hyperparameters are estimated within the Markov chain. Again, this method requires execution times of the order of hours, hindering any practical 3D imaging applications.

1.3.3 Multi-depth algorithms

The general multi-depth assumption includes the previously discussed algorithms as special cases, at the expense of solving a harder problem. We have identified two main strategies: The first approach aims at estimating a 3D volume of intensity values, where only a few non-zero values correspond to the 3D points. These methods estimate a vectorised data cube of intensities $\boldsymbol{r} = [\boldsymbol{r}_{1,1}^{\mathsf{T}}, \ldots, \boldsymbol{r}_{i,j}^{\mathsf{T}}, \ldots, \boldsymbol{r}_{N_r,N_c}^{\mathsf{T}}] \in \mathbb{R}_+^{N_r N_c T}$, having one intensity per histogram entry of pixel (i,j). In this model, the depth $\boldsymbol{t}$ is implicitly given by the non-zero entries of $\boldsymbol{r}$. The negative log-likelihood (1.2) is rewritten as

$$g\left(\boldsymbol{r}, \boldsymbol{b}\right) = \sum_{i=1}^{N_r} \sum_{j=1}^{N_c} \mathbf{1}_T^{\mathsf{T}} \boldsymbol{r}_{i,j} + b_{i,j} T - \boldsymbol{z}_{i,j}^{\mathsf{T}} \log\left(\boldsymbol{H} \boldsymbol{r}_{i,j} + \mathbf{1}_T b_{i,j}\right) \tag{1.12}$$

where $\mathbf{1}_T$ is a unitary vector of T elements, $\boldsymbol{z}_{i,j} = [z_{i,j,1}, \ldots, z_{i,j,T}]^{\mathsf{T}}$ is a vector containing the histogram entries at pixel (i,j) and $\boldsymbol{H} \in \mathbb{R}_+^{T \times T}$ is the convolution (Toeplitz) matrix associated with $h(t)$. Convex priors are then assigned to $\boldsymbol{r}$, resulting in the following minimisation problem

$$(\hat{\boldsymbol{r}}, \hat{\boldsymbol{b}}) = \underset{\boldsymbol{r}, \boldsymbol{b}}{\arg\min}\, g\left(\boldsymbol{r}, \boldsymbol{b}\right) + \lambda_{\boldsymbol{r}} \rho_{\boldsymbol{r}}(\boldsymbol{r}) \tag{1.13}$$

which in this case is convex and can be solved with SPIRAL or PIDAL. As only a very few entries of the intensity cube should be different from zero, Shin et al. [36] proposed an ℓ_1 norm regularisation for $\rho_{\boldsymbol{r}}(\boldsymbol{r})$ to promote sparse reconstructions. The algorithm, referred to as SPISTA, relies on a post-processing of the 3D point cloud to sparsify further the output. The main drawback of SPISTA is that no spatial correlation is promoted with the ℓ_1 norm. Halimi et al. [37] tackled this problem by considering the ℓ_{21} norm, which promotes correlation between reflectivity bins within a small depth interval (see Fig. 1.5). Moreover, to promote further spatial correlation between neighbouring pixels, this method considers an additional (volumetric) TV for the reflectivity cube. The algorithm, referred to as ℓ_{21}+TV, also relies in a post-processing step to sparsify the output.

Despite the advantage of having a unique solution, the volumetric formulation presents disadvantages:

1. The estimated values $\boldsymbol{r}$ are generally not sparse enough (over-estimation of the number of points) and the minimisation of (1.13) involves gradient steps with a dense computation over the complete cube.

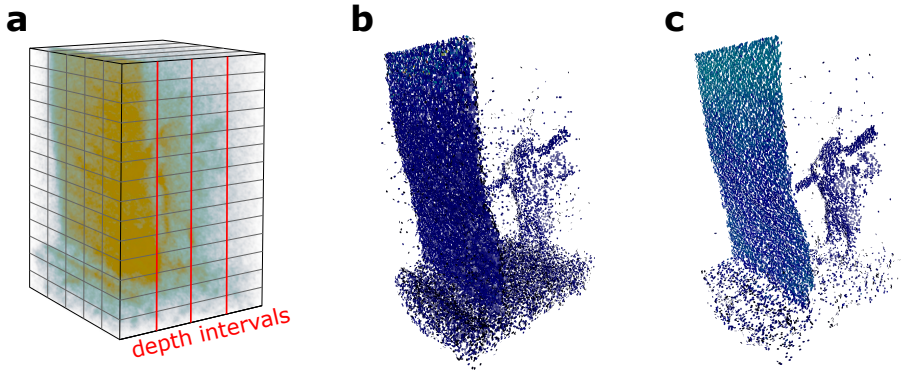


Figure 1.5 Example of multi-depth reconstruction algorithms based on a reflectivity cube. The photon detections are shown in (a). Reconstructions using (a) ℓ_1 and (b) ℓ_{21} regularisations. The depth intervals (red) for the ℓ_{21} regularisation are shown in (a).

2. The regularisation term does not capture well the manifold structure of the 3D point clouds. While the ℓ_1 regularisation does not consider spatial correlations, the ℓ_{21} relies on heuristics to define the depth intervals, as they are not a priori known. Moreover, a TV-based regularisation term promotes volumetric smoothness, which generally results in poor reconstruction quality and the need of empirical post-processing steps, as the reconstructed surfaces should be manifolds[2].

3. These methods suffer from large memory requirements, as they have to store a dense cube $\boldsymbol{r}$ (potentially many times due to the structure of the ADMM algorithm).

A second strategy directly estimates a 3D point cloud, where the dimension of the parameter space (i.e., the number of 3D points) is a priori unknown. The first step in this direction was the algorithm in [38], which infers the point positions using a reversible jump MCMC (RJ-MCMC) algorithm to handle the varying number of points per pixel. While this approach is able to find an a priori unknown number of surfaces and compute associated uncertainty intervals, it involves a prohibitive computation time. Moreover, it performs poorly when photon counts are relatively low, as it does not account for spatial correlation between neighbouring pixels. In later work, Hernandez-Marin et al. [39] proposed an extension to the latter algorithm, where a Potts model was used to regularise spatially the number of surfaces per pixel. However, the computational load of their algorithm remained prohibitive for large images and the correlation between the intensity and position of each object was not a priori modelled.

There have been other attempts to derive statistical models for lidar waveforms with an unknown number of objects per pixel, such as Mallet et al. with full waveform topographic lidar [40], where a marked point process was considered for each pixel separately. While they defined interactions between pulses in the same pixel, no spatial interaction between points of neighbouring pixels was considered.

[2]In some environmental monitoring applications, the recorded datasets can contain dense foliage. In such cases, if the spatial resolution is relatively low, the observed signal may not be well-described by manifolds.

1.4 Contributions

The methods proposed in this thesis tackle the shortcomings of the algorithms described in the previous section. We present novel modelling techniques that better capture the manifold structure of natural 3D point clouds, obtaining reconstructions with higher quality than previous methods. These algorithms are designed for the general multiple-surface per pixel setting, being applicable to a wide range of real-world applications. While the proposed methods are introduced in the context of single-photon lidar, they can be applied to other 3D scanning modalities or inverse problems involving structured point clouds. Each main contribution of this thesis is associated with an individual chapter.

Point process model We introduce a spatial point process that captures the manifold structure of 3D point clouds, coupled with an RJ-MCMC algorithm to infer the parameters of the model. We show that this method obtains state-of-the-art performance in inferring 3D point clouds from noisy data in the context of multi-depth single-photon lidar [41]. The general formulation and flexibility of this method allows us to account for a wide variety of sensing scenarios. In particular, we show applications to highly-attenuating media (underwater scenes) and peak broadening (long-range scenes) [42].

Multispectral lidar We extend the point process model to multispectral single-photon lidar data, introducing a carefully designed inference approach to handle the very high dimensionality of the data. As in the single-wavelength case, the resulting reconstruction algorithm can handle multiple surfaces per pixel having minimal memory requirements. Furthermore, we introduce a spectral subsampling technique, which reduces the number of necessary measurements. The proposed subsampling approach leads to faster acquisitions and reconstructions, lower memory requirements, and better reconstruction performance than other existing random subsampling schemes [43].

Fast surface detection We propose a Bayesian detection algorithm, which is able to distinguish whether a target is present or not at each imaged pixel. The method relies on a hierarchical Bayesian model that incorporates the Poisson observation model and other prior knowledge about the scene. The resulting algorithm can run in real-time, reducing the execution time of previous methods. Moreover, we present two post-processing schemes to achieve spatial correlation without losing the real-time capability. Experiments using lidar data demonstrate state-of-the-art detection performance [44, 45].

Plug-and-play point cloud denoising framework We introduce a plug-and-play framework for inverse problems involving the reconstruction of 3D point clouds. We show that off-the-shelf

point cloud denoisers from the computer graphics literature can be used as implicit regularisation terms. This idea extends the plug-and-play image processing techniques [46, 47] to point cloud restoration. The resulting algorithm benefits from both the probabilistic observation model and powerful manifold modelling techniques, obtaining state-of-the-art reconstructions in the context of multiple surfaces per pixel. A series of experiments using raster-scanning and lidar array technologies demonstrate the efficiency of the method [48]. We also present an extension to multispectral lidar that achieves real-time 3D colour reconstructions [49].

1.5 Organisation of the thesis

The thesis is organised as follows: Chapter 2 presents the novel spatial point process model for capturing correlations of 3D point clouds, designed for the lidar multi-depth setting. An RJ-MCMC algorithm is used to infer the point cloud parameters and quantify uncertainty. A comparison with other reconstruction methods using real lidar data shows competitive results. Chapter 3 extends this model to multispectral lidar systems. A subsampling scheme is proposed to reduce the number of necessary measurements to obtain multispectral reconstructions. Chapter 4 focuses on the target detection setting. This chapter presents two Bayesian target detection methods that can discard pixels without surfaces in real-time. Chapter 5 presents a plug-and-play reconstruction method that benefits from off-the-shelf point cloud denoisers to attain state-of-the-art reconstructions in real-time. The resulting algorithm also applies in the general multi-depth setting and its efficiency is demonstrated throughout a series of experiments with lidar videos (sequences of frames). Finally, Chapter 6 concludes the thesis and discusses future directions of research.

Chapter 2

Imaging complex 3D scenes

Contents

2.1 Introduction

As discussed in the previous chapter, multi-depth methods based on a dense cube of intensities [36, 37] fail to capture the manifold structure of the non-zero elements. In this chapter we follow a different path. We introduce a model that handles varying dimensions as in previous RJ-MCMC approaches [38], but also captures complex spatial interactions between points, both at a pixel level and at an inter-pixel level. Here we consider each surface within a pixel as a point in 3D space, which has a mark that indicates its intensity.

Natural lidar point clouds exhibit strong spatial clustering, as points belonging to the same surface tend to be close in range. Conversely, points in a given pixel tend to be separated as they correspond to different surfaces. Figure 2.1 shows an example of a synthetic lidar 3D point cloud to illustrate these phenomena. This prior information is added to our model using spatial point processes: repulsion between points at a pixel level is achieved with a hard constraint Strauss process and attraction among points in neighbouring pixels is achieved by an area interaction process, as defined in [50]. Moreover, the combination of these two processes implicitly defines a connected-surface structure that is used to efficiently sample the posterior distribution. To promote smoothness between the intensity of points in the same surface, we define a nearest neighbour Gaussian Markov random field (GMRF) prior model, similar to the one proposed in [51]. Inference about the posterior distribution of points, their marks and the background level is done by an RJ-MCMC algorithm, with carefully tailored moves to obtain high acceptance rates, ensuring better mixing and faster convergence rate. In addition to traditional birth/death, split/merge, shift and mark moves, new dilation/erosion moves are introduced, which add and remove new points by extending or shrinking a connected surface respectively. These moves lead to a much higher acceptance rate than those obtained for birth and death updates, as they propose moves to and within regions of high posterior probability. To further reduce the transient regime of the Markov chains and reduce the computational time of the algorithm, we consider a multiresolution approach, where the original lidar 3D data is binned into a coarser resolution data cube with higher signal power, lower number of points and same data statistics. An initial estimate obtained from the downsampled data is used as the initial configuration for the finer scale, thus reducing the number of burn-in iterations needed for the Markov chains to convergence.

We assess the quality of reconstruction and the computational complexity in several experiments based on synthetic lidar data and three real lidar datasets. The algorithm leads to new efficient 3D reconstructions with similar processing times to other existing optimisation-based methods. This method can be applied to scenes where there is only one object per pixel, thus generalising other single-depth algorithms. Moreover, the algorithm can also be applied to scenes where each pixel has at most one surface, generalising other target detection methods [35]. We refer to the proposed method as ManiPoP, as it aims to representing 2D manifolds with a 3D point process.

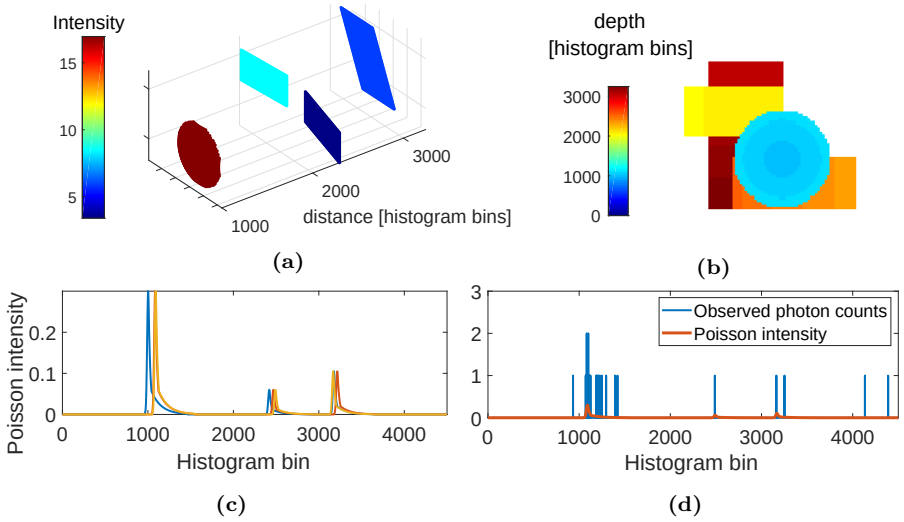


Figure 2.1 (a) depicts a synthetic 3D point cloud with $N_r = 99$ rows, $N_c = 99$ columns and $T = 4500$ bins. The scene consists of 3 plates with different sizes and orientations and one ball shaped object. The intensity represents the mean number of photons associated with each 3D point. (b) illustrates the depth of the first object for each pixel. (c) shows the intensity of 3 neighbouring pixels. The observed photon counts and underlying Poisson intensity of a pixel with 3 surfaces is shown in sub-figure (d).

The remainder of this chapter is organised as follows: Section 2.2 presents the Bayesian model considered for the analysis of multiple-depth lidar data. Section 2.3 details the sampling strategy using an RJ-MCMC algorithm. Section 2.4 discusses the proposed multiresolution approach and other implementation details to reduce the computational load of the algorithm. Results of experiments conducted on synthetic and real data are presented in Section 2.5. An extension of the algorithm that accounts for broadening of the instrumental response and highly scattering media is introduced in Section 2.6. Finally, Section 2.7 concludes the chapter.

2.2 Proposed Bayesian model

In this section we detail the prior distribution associated with the unknown parameters $(\mathbf{\Phi}, \boldsymbol{b})$. For ease of presentation, we denote the set of point coordinates as $\mathbf{\Phi}_c$ and the set of intensity values as $\mathbf{\Phi}_r$.

2.2.1 Markov marked point process

The set of point positions is defined as an unordered set $\mathbf{\Phi}_c = \{\mathbf{c}_n \quad n = 1, \ldots, N_{\mathbf{\Phi}}\}$, where each position is defined in the voxelised 3D space $\mathcal{T} = \{1, \ldots, N_r\} \times \{1, \ldots, N_c\} \times \{1, \ldots, T\} \subset \mathbb{Z}_+^3$. Following [50], the space of all point configurations can be defined as

$$\Omega = \bigcup_{N_\Phi \in \mathbb{Z}_+} \Omega_{N_\Phi} \tag{2.1}$$

where Ω_{N_Φ} denotes the space of configurations containing exactly N_Φ points. A Poisson point process is used as the basic building block for more elaborate point processes that capture spatial correlations. Points $\mathbf{\Phi}_c$ of a Poisson process with intensity $\lambda : \mathcal{T} \mapsto \mathbb{R}_+$, are independently distributed in $\mathcal{T}$. The number of points found in a subset B of $\mathcal{T}$ is a random variable distributed according to a Poisson distribution with mean $\lambda(B)$. Moreover, the numbers of points in K disjoint subsets $B_1, \ldots, B_K$ are mutually independent. The probability measure $\pi : \Omega \mapsto \mathbb{R}_+$ associated with a Poisson process on a subset A of the configuration space Ω is

$$\pi(\mathbf{\Phi}_c) = e^{-\lambda(\mathcal{T})} \sum_{N_{\Phi_c} \in \mathbb{Z}_+} \mathbb{1}_{\Omega_{N_\Phi}}(\mathbf{\Phi}_c) \lambda(\boldsymbol{c}_1) \ldots \lambda(\boldsymbol{c}_{N_{\Phi_c}}) \tag{2.2}$$

where $\lambda(\mathcal{T})$ is the expected total number of points, $\mathbb{1}_A(\cdot)$ is the indicator function defined in A. Interactions between points can be characterised using a normalised density $f : \Omega \to \mathbb{R}_+$, defined with respect to the Poisson reference measure π such that

$$\int_\Omega f(\mathbf{\Phi}_c) \pi(d\mathbf{\Phi}_c) = 1. \tag{2.3}$$

Multiple densities can be defined as

$$f(\mathbf{\Phi}_c) \propto f_1(\mathbf{\Phi}_c) \ldots f_r(\mathbf{\Phi}_c) \tag{2.4}$$

where $\propto$ means "proportional to" and r is the number of interactions, which will be fixed to $r = 2$ in this chapter.

We only consider (local) Markovian interactions between points. The benefits of this property are twofold: a) Markovian interactions are well suited to describe the spatial correlations in natural 3D scenes [52] and b) inference is performed using only local updates, which leads to a low computational complexity. We constrain the minimum distance between two different surfaces in the same pixel using the hard object process with density

$$f_1(\mathbf{\Phi}_c | d_{\min}) \propto \begin{cases} 0 & \text{if } \exists\ n \neq n' : x_n = x_{n'}, y_n = y_{n'} \\ & \text{and } |t_n - t_{n'}| < d_{\min} \\ 1 & \text{otherwise} \end{cases} \tag{2.5}$$

which is a special case of the repulsive Strauss process [50], where $d_{\min}$ is the minimum distance between two points in the same pixel. Attraction between points of the same surface in neighbouring pixels cannot be modelled with another Strauss process, due to a phase transition of extremely clustered realisations, as explained in [50,52]. However, a smoother transition into clustered configurations can be achieved by the area interaction process, introduced by Baddeley and Van Lieshout

in [53]. In this case, the density is defined as

$$f_2(\mathbf{\Phi}_c|\gamma_a,\lambda_a) = k_1 \lambda_a^{N_\Phi} \gamma_a^{-m\left(\bigcup_{n=1}^{N_\Phi} S(\boldsymbol{c}_n)\right)} \tag{2.6}$$

where λ_a is a positive parameter that controls the total number of points, $\gamma_a \geq 1$ is a parameter adjusting the attraction between points[1] and k_1 is an intractable normalising constant. The exponent of γ_a in (2.6) is the counting measure $m(\cdot)$ over the union of convex sets $S(\boldsymbol{c}_n) \subseteq \mathcal{T}$, defined as a cuboid with a face of 3×3 squared pixels and a depth of $2N_b + 1$ histogram bins centred around each point $\boldsymbol{c}_n$. The set $S(\boldsymbol{c}_n)$ determines a cuboid of influence around each point, allowing interactions with the 8 nearest neighbouring pixels, up to a distance of N_b bins in depth. As two points in the same pixel generally correspond to different surfaces, we set $d_{\min} > 2N_b$, thus constraining the minimum distance between two surfaces in the same pixel. The combination of the Strauss process and the area interaction process implicitly defines a connected-surface structure. Figures 2.2 and 2.3 illustrate the connected-surface structure via several examples. In the rest of this chapter, we fix $\lambda(\mathcal{T}) = 1$ and control the number of points with the parameter λ_a.

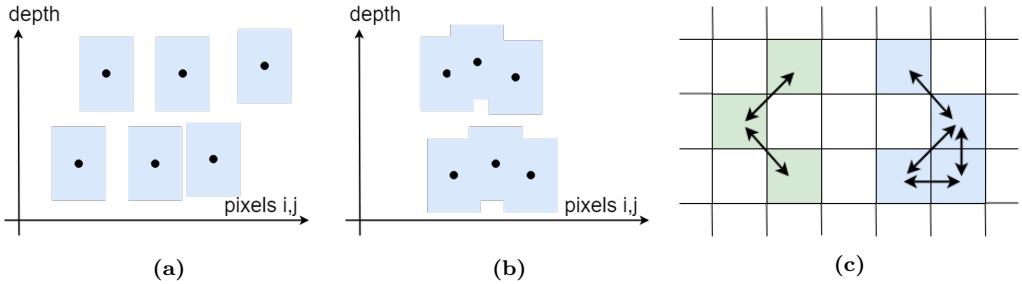


Figure 2.2 (a) and (b) show two different point configurations. Each point $\boldsymbol{c}_n$ is denoted by a black dot and the corresponding blue rectangle depicts the area of the convex set $S(\boldsymbol{c}_n)$. The configuration shown in (a) has a lower prior probability than the one shown in (b), as the union of all sets $S(\boldsymbol{c}_n)$ is smaller in (b) with respect to the counting measure. (c) shows the connectivity at an inter-pixel level. The green and blue squares correspond to pixels with points associated with two different surfaces, while the white squares denote pixels without points. For simplicity, in this example all points are considered to be present at the same depth. Note that each pixel can be connected with at most 8 neighbours.

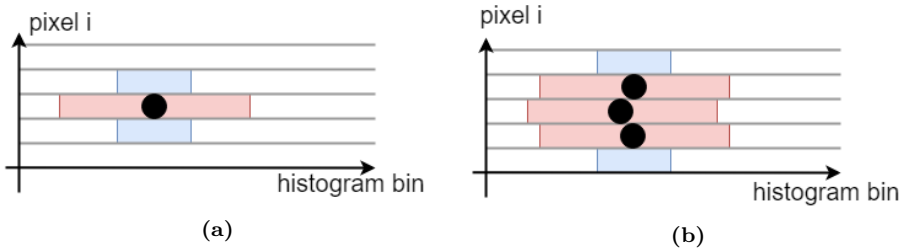


Figure 2.3 In both figures, the red colour denotes the space where no other points can be found (Strauss process) whereas the blue colour denotes the volume where other points are likely to appear (area interaction process). (a) Example of configuration with 1 point. (b) Example of configuration with 3 points.

[1]The special case $\gamma_a = 1$ corresponds to a Poisson point process (without considering a Strauss process) with an intensity proportional to $\lambda_a \lambda(\cdot)$.

The hyperparameters γ_a and λ_a of the area interaction process are difficult to estimate, as there is an intractable normalising constant in the density of (2.6) and standard MCMC methods cannot be directly applied. Although there exist ways of bypassing this problem (e.g., [54]), we fixed these hyperparameters in all our experiments to ensure a reasonable computational complexity.

After defining the spatial priors, the marked point process is constructed by adding the intensity marks r_n to the set $\mathbf{\Phi}_c$ with the density detailed in the next section.

2.2.2 Intensity prior model

In natural scenes, the intensity values of points within a same surface exhibit strong spatial correlation. Following the Bayesian paradigm, this prior knowledge can be integrated into our model by defining a prior distribution over the point marks. Gaussian processes are classically used in spatial statistics. However, the underlying covariance structure needs to consider too many neighbouring points to attain sufficient smoothing, which involves a prohibitive computational load. In order to obtain similar results with a lower computational burden, we propose to exploit the connected-surface structure to define a nearest neighbour Gaussian Markov random field, similar to the one used by McCool et al. in [51]. First, we alleviate the difficulties induced by the positivity constraint of the intensity values by introducing the following change of variables, which is a standard choice in spatial statistics dealing with Poisson noise (see [55, Chapter 4])

$$m_n = \log(r_n) \quad n = 1, \dots, N_{\Phi_c}. \tag{2.7}$$

Second, spatial correlation is promoted by defining the following conditional distribution of the log-intensities,

$$p(m_n | \mathcal{M}_{pp}(\boldsymbol{c_n}), \sigma^2, \beta) \propto \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_n - m_{n'})^2}{d(\boldsymbol{c}_n; \boldsymbol{c}_{n'})} + m_n^2 \beta \right) \right) \tag{2.8}$$

where $\mathcal{M}_{pp}(\boldsymbol{c}_n)$ is the set of neighbours of $\boldsymbol{c}_n$, $d(\boldsymbol{c}_n; \boldsymbol{c}_{n'})$ denotes the Euclidean distance between the points $\boldsymbol{c}_n$ and $\boldsymbol{c}_{n'}$, and β and σ^2 are two positive hyperparameters. The set of neighbours $\mathcal{M}_{pp}(\boldsymbol{c}_n)$ is obtained using the connected-surface structure, where each point can have at most 8 neighbours, as illustrated in Section 2.2.1. The distance between two points is computed according to

$$d(\boldsymbol{c}_n; \boldsymbol{c}_{n'}) = \sqrt{(y_n - y_{n'})^2 + (x_n - x_{n'})^2 + \left(\frac{t_n - t_{n'}}{l_z} \right)^2} \tag{2.9}$$

with $l_z = \Delta_p / \Delta_b$, which normalises the distance to have a physical meaning, where Δ_p and Δ_b are the approximate spatial resolutions of one pixel and one histogram bin respectively. This prior

promotes a linear interpolation between neighbouring[2] intensity values, as explained in [55]. Here we assume that Δ_p is constant throughout the scene. If the scene presents significant distortion, i.e., objects separated by a significant distance in depth, Δ_p should depend on the position by computing the projective transformation between world coordinates and lidar coordinates [5]. Following the Hammersley and Clifford theorem [55], the joint intensity distribution is given by the multivariate Gaussian distribution

$$\boldsymbol{m}|\sigma^2, \beta, \boldsymbol{\Phi}_c \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{P}^{-1}) \tag{2.10}$$

where $\boldsymbol{P}$ is the unscaled precision matrix of size $N_\Phi \times N_\Phi$ with the following elements

$$[\boldsymbol{P}]_{n,n'} = \begin{cases} \beta + \sum_{\tilde{n} \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{1}{d(\boldsymbol{c}_n; \boldsymbol{c}_{\tilde{n}})} & \text{if } n = n' \\ -\frac{1}{d(\boldsymbol{c}_n; \boldsymbol{c}_{n'})} & \text{if } \boldsymbol{c}_n \in \mathcal{M}_{pp}(\boldsymbol{c}_{n'}) \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

The parameter σ^2 controls the surface intensity smoothness and $\frac{\beta}{\sigma^2}$ is related to the intensity variance of a point without any neighbour. In addition, the parameter β ensures a proper joint prior distribution, as $\boldsymbol{P}$ is diagonally dominant, thus full rank [55].

2.2.3 Background prior model

Non-coherent illumination sources, such as the solar illumination in outdoor scenes or room lights in the indoor case, are related to arrivals of photons at random times (uniformly distributed in time) to the single-photon detector. The level of these spurious detections is modelled as a 2D image of mean intensities $b_{i,j}$ with $i = 1, \ldots, N_r$ and $j = 1, \ldots, N_c$. If the transceiver system of the lidar is monostatic[3] (e.g., the system described in [11]), the background image is usually similar to the objects present in the scene and exhibits spatial correlation, as background photons generally arise from the ambient light reflecting from parts of the targets and being collected by the system. Hence, we use a hidden gamma Markov random field prior distribution for $\boldsymbol{b}$ that takes into account the background positivity and spatial correlation. This prior was introduced by Dikmen and Cemgil in [57] and applied in many image processing applications with Poisson likelihood [58, 59]. In [57], the distribution of $b_{i,j}$ is defined via auxiliary vectorised image $\boldsymbol{u} = [u_{1,1}, \ldots, u_{i,j}, \ldots, u_{N_r,N_c}]^\mathsf{T}$ such that

$$b_{i,j}|\mathcal{M}_B(b_{i,j}), \alpha_b \sim \mathcal{G}\left(\alpha_b, \frac{\overline{b}_{i,j}}{\alpha_b}\right) \tag{2.12}$$

$$u_{i,j}|\mathcal{M}_B(u_{i,j}), \alpha_b \sim \mathcal{IG}(\alpha_b, \alpha_b \overline{u}_{i,j}) \tag{2.13}$$

[2]The combination of a local Euclidean distance with a nearest neighbours definition can be seen to approximate the manifold metrics [56].

[3]The transceiver system is monostatic when the transmit and receive channels are co-axial and thus share the same objective lens aperture.

where $\mathcal{M}_B$ denotes the set of 5 neighbours as shown in Fig. 2.4a, $\mathcal{G}$ and $\mathcal{IG}$ indicate gamma and inverse gamma distributions, α_b is a hyperparameter controlling the spatial regularisation and

$$\overline{b}_{i,j} = \left(\frac{1}{4} \sum_{(i',j') \in \mathcal{M}_B(b_{i,j})} u_{i',j'}^{-1} \right)^{-1} \tag{2.14}$$

$$\overline{u}_{i,j} = \frac{1}{4} \sum_{(i',j') \in \mathcal{M}_B(u_{i,j})} b_{i',j'} \tag{2.15}$$

We are interested in the marginal distribution of the gamma Markov random field $p(\boldsymbol{b}|\alpha_b)$ that integrates over all possible realisations of the auxiliary variables $u_{i,j}$. The expression of this marginal density can be obtained analytically (a detailed derivation can be found in Appendix A), i.e.,

$$p(\boldsymbol{b}|\alpha_b) \propto \int p(\boldsymbol{b}, \boldsymbol{u}|\alpha_b) d\boldsymbol{u} \tag{2.16}$$

$$\propto \prod_{i=1}^{N_r} \prod_{j=1}^{N_c} \frac{b_{i,j}^{\alpha_b - 1}}{\left(\sum_{(i',j') \in \mathcal{M}_B(u_{i,j})} b_{i',j'} \right)^{\alpha_b}}. \tag{2.17}$$

We fix the value of α_b, even if it could also be estimated using a stochastic gradient procedure as explained in [60], at the expense of an increase in the computational load. If the system is not monostatic, i.e., there is no prior assumption of smoothness in the background image, the value of α_b is set to 1.

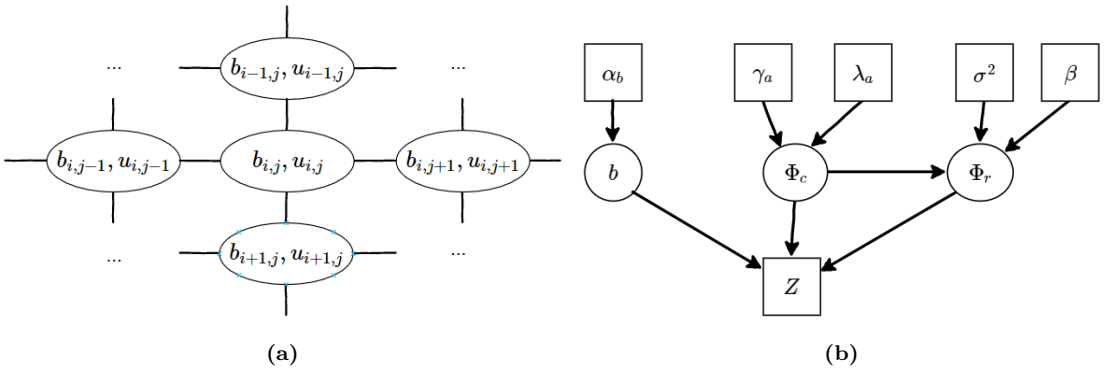


Figure 2.4 (a) illustrates the gamma Markov random field neighbouring structure $\mathcal{M}_B$. Each $b_{i,j}$ is connected to 5 auxiliary variables $u_{i',j'}$ as depicted by the continuous lines, including the one with the same subscript. Similarly, each $u_{i,j}$ is also connected to other 5 variables $b_{i',j'}$ as indicated by the continuous lines. (b) shows the directed acyclic graph (DAG) of the proposed hierarchical Bayesian model. The variables inside squares are fixed, whereas the variables inside circles are estimated.

2.2.4 Posterior distribution

The joint posterior distribution of the model parameters is given by

$$p(\boldsymbol{\Phi}_c, \boldsymbol{\Phi}_r, \boldsymbol{b}|\boldsymbol{Z}, \boldsymbol{\Psi}) \propto p(\boldsymbol{Z}|\boldsymbol{\Phi}_c, \boldsymbol{\Phi}_r, \boldsymbol{b}) p(\boldsymbol{\Phi}_r|\boldsymbol{\Phi}_c, \sigma^2, \beta) f_1(\boldsymbol{\Phi}_c|d_{\min}) f_2(\boldsymbol{\Phi}_c|\gamma_a, \lambda_a) \pi(\boldsymbol{\Phi}_c) p(\boldsymbol{b}|\alpha_b) \tag{2.18}$$

where $\boldsymbol{\Psi}$ denotes the set of hyperparameters $\boldsymbol{\Psi} = \{d_{\min}, \gamma_a, \lambda_a, \sigma^2, \beta, \alpha_b\}$, the likelihood of the observed data has been defined in (1.1) and (1.2), the Poisson reference measure is defined by (2.2), and the other densities are priors defined in (2.5), (2.6), (2.10) and (2.16). Figure 2.4b shows the directed acyclic graph associated with the proposed hierarchical Bayesian model.

2.3 Estimation strategy

Bayesian estimators associated with the full posterior in (2.18) are analytically intractable. Moreover, standard optimisation techniques cannot be applied due to the non-convex nature of the posterior distribution, caused by the likelihood and point process prior terms. However, we can obtain numerical estimates using samples generated by a Monte Carlo method denoted as

$$\{\boldsymbol{\Phi}^{(s)}, \boldsymbol{b}^{(s)} \quad \forall s = 0, 1, \ldots, N_i - 1\} \tag{2.19}$$

where N_i is the total number of samples. We use the MAP estimator of the point cloud positions and intensity values, i.e.,

$$\hat{\boldsymbol{\Phi}} = \arg\max_{\boldsymbol{\Phi}, \boldsymbol{b}} p(\boldsymbol{\Phi}, \boldsymbol{b} | \boldsymbol{Z}, \boldsymbol{\Psi}), \tag{2.20}$$

which is approximated by

$$\hat{s} = \arg\max_{s=0,\ldots,N_i-1} p(\boldsymbol{\Phi}^{(s)}, \boldsymbol{b}^{(s)} | \boldsymbol{Z}, \boldsymbol{\Psi}) \tag{2.21}$$

with $\hat{\boldsymbol{\Phi}} \approx \boldsymbol{\Phi}^{(\hat{s})}$. In our experiments, we found that the MMSE of $\boldsymbol{b}$, i.e.,

$$\hat{\boldsymbol{b}} = \mathbb{E}\{\boldsymbol{b} | \boldsymbol{Z}, \boldsymbol{\Psi}\} \tag{2.22}$$

achieves better background estimates than the MAP estimator in all of our experiments. This estimator can be approximated by the empirical mean of the posterior samples of $\boldsymbol{b}$, that is

$$\hat{\boldsymbol{b}} \approx \frac{1}{N_i} \sum_{s=N_{\text{bi}}+1}^{N_i} \boldsymbol{b}^{(s)} \tag{2.23}$$

where $N_{\text{bi}} = N_i/2$ is the number of burn-in iterations. In many applications, assessing the presence or absence of a target at a pixel level can be of special interest [35]. Here, we can use the Monte Carlo samples to estimate the probability of having k objects present in pixel (i, j), as

$$P(k \text{ returns in } (i,j) | \boldsymbol{Z}, \boldsymbol{\Psi}) = \frac{1}{N_i} \sum_{s=N_{\text{bi}}+1}^{N_i} \mathbb{1}_{k \text{ points in } (i,j)}(\boldsymbol{\Phi}^{(s)}). \tag{2.24}$$

Remark: If more detailed posterior statistics are needed, it is possible to fix the dimensionality of the problem using the estimate $\hat{\boldsymbol{\Phi}}$ and run a fixed dimensional sampler for additional N_i iterations.

Many samplers capable of exploring different model dimensions, i.e., different numbers of points, are available in the point process literature (a complete summary can be found in [21, Chapter 9]). The continuous birth-death chain method builds a continuous-time Markov chain that converges to the posterior distribution of interest. Alternatively, perfect sampling approaches generate samples using a rejection sampling scheme, which incurs a larger computational load. Finally, the RJ-MCMC sampler, introduced by Green in [61], constructs a discrete time Markov chain, where moves between different dimensions are proposed and accepted or rejected in order to converge to the posterior distribution of interest. We choose an RJ-MCMC sampler, as this option allows us to design application-specific proposals that speed up the convergence rate.

In addition, we propose a data augmentation scheme to sample the background levels. This technique introduces extra auxiliary (latent) variables $\boldsymbol{u}$ and generates samples in this augmented model space $(\boldsymbol{b}^{(s)}, \boldsymbol{u}^{(s)}) \sim p(\boldsymbol{b}, \boldsymbol{u}|\boldsymbol{Z}, \boldsymbol{\Phi}, \alpha_b)$, which is easier than sampling the marginal distribution $p(\boldsymbol{b}|\boldsymbol{Z}, \boldsymbol{\Phi}, \alpha_b)$. The resulting samples $\boldsymbol{b}^{(s)}$ are distributed according to the desired marginal density (detailed theory and applications of data augmentation can be found in [21, Chapter 10]).

2.3.1 Reversible jump Markov chain Monte Carlo

RJ-MCMC can be seen as a natural extension of the Metropolis-Hastings algorithm for problems with an unknown dimensionality. Given the actual state of the chain $\boldsymbol{\theta} = \{\boldsymbol{\Phi}, \boldsymbol{b}\}$ of model order N_Φ, a random vector of auxiliary variables $\boldsymbol{u}$ is generated to create a new state $\boldsymbol{\theta'} = \{\boldsymbol{\Phi}', \boldsymbol{b}'\}$ of model order $N_{\Phi'}$, according to an appropriate deterministic function $\boldsymbol{\theta'} = g(\boldsymbol{\theta}, \boldsymbol{u})$. To ensure reversibility, an inverse mapping with auxiliary random variables $\boldsymbol{u'}$ has to exist such that $\boldsymbol{\theta} = g^{-1}(\boldsymbol{\theta'}, \boldsymbol{u'})$. The move $\boldsymbol{\theta} \to \boldsymbol{\theta'}$ is accepted or rejected with probability $\rho = \min\{1, r\left(\boldsymbol{\theta}, \boldsymbol{\theta'}\right)\}$, where $r(\cdot,\cdot)$ satisfies the so-called dimension balancing condition

$$r\left(\boldsymbol{\theta}, \boldsymbol{\theta'}\right) = \frac{p(\boldsymbol{\theta'}|\boldsymbol{Z}, \boldsymbol{\Psi}) K(\boldsymbol{\theta}|\boldsymbol{\theta'}) p(\boldsymbol{u'})}{p(\boldsymbol{\theta}|\boldsymbol{Z}, \boldsymbol{\Psi}) K(\boldsymbol{\theta'}|\boldsymbol{\theta}) p(\boldsymbol{u})} \left| \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{u})}{\partial(\boldsymbol{\theta}, \boldsymbol{u})} \right| \tag{2.25}$$

where $K(\boldsymbol{\theta'}|\boldsymbol{\theta})$ is the probability of proposing the move $\boldsymbol{\theta} \to \boldsymbol{\theta'}$, $p(\boldsymbol{u})$ is the probability distribution of the random vector $\boldsymbol{u}$, and $\left|\frac{\partial g(\boldsymbol{\theta},\boldsymbol{u})}{\partial(\boldsymbol{\theta},\boldsymbol{u})}\right|$ is the Jacobian of the mapping $g(\cdot)$. All the terms involved in (2.25) have a complexity that depends only on the size of the neighbourhood, except the prior distribution (within the posterior term $p(\boldsymbol{\theta}|\boldsymbol{Z}, \boldsymbol{\Psi})$) of the intensity values defined in (2.10). Note that (2.10) involves the computation of the ratio of determinants of the precision matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$, which have a global dependency on all the points in Φ_r. To keep the computational complexity low, we address this difficulty by only considering a block diagonal approximation of $\boldsymbol{P}$, which includes only points in local neighbourhoods. The RJ-MCMC algorithm performs birth, death, dilation, erosion, spatial shift, mark shift, split and merge moves with probabilities p_{birth}, p_{death}, p_{dilation}, p_{erosion}, p_{shift}, p_{mark}, p_{split} and p_{merge}. These moves are detailed in the following subsections. Here we summarise the key aspects of each move, without specifying the full acceptance rate expression

of (2.25), which can be found in Appendix B.

Birth and death moves The birth move proposes a new point $(\boldsymbol{c}_{N_\Phi+1}, m_{N_\Phi+1})$ uniformly at random in $\mathcal{T}$. The intensity of the new point is computed according to the following scheme

$$\begin{cases} u \sim \mathcal{U}(0,1), b'_{i,j} = u b_{i,j} \\ e^{m_{N_\Phi+1}} = (1-u) b_{i,j} \dfrac{T}{\sum_{t=1}^{T} h(t)}. \end{cases} \tag{2.26}$$

This mapping preserves the total posterior intensity of the pixel, since

$$e^{m_{N_\Phi+1}} \sum_{t=1}^{T} h(t) + b'_{i,j} T = b_{i,j} T, \tag{2.27}$$

thus yielding a relatively high acceptance probability. Its reversible pair, the death move, proposes to remove one point randomly. In this case, the inverse mapping is given by

$$b'_{i,j} = b_{i,j} + e^{m_{N_\Phi+1}} \frac{\sum_{t=1}^{T} h(t)}{T}. \tag{2.28}$$

The acceptance ratio for the birth move reduces to $\rho = \min\{1, C_1\}$ with C_1 given by (2.25), where the posterior ratio is computed according to (2.18), $K(\boldsymbol{\theta'}|\boldsymbol{\theta}) = p_{\text{birth}}$, $K(\boldsymbol{\theta}|\boldsymbol{\theta'}) = p_{\text{death}}$, $p(\boldsymbol{u}) = \frac{\lambda(\cdot)}{\lambda(\mathcal{T})}$, $p(\boldsymbol{u'}) = \frac{1}{N_\Phi+1}$ and a Jacobian equal to $\frac{1}{1-u}$. The death move is accepted or rejected with probability $\rho = \min\{1, C_1^{-1}\}$, modifying $p(\boldsymbol{u})$ accordingly (i.e., changing $\frac{1}{N_\Phi+1}$ to $\frac{1}{N_\Phi}$).

Dilation and erosion moves Standard birth and death moves yield low acceptance rates, because the probability of proposing a point in a likely position is relatively low, as the detected surfaces only occupy a small subset of the full 3D volume $\mathcal{T}$. To overcome this problem, we propose new RJ-MCMC moves that explore the target distribution by dilating and eroding existing surfaces. The dilation move randomly picks a point $\boldsymbol{c}_n$ that has less than 8 neighbours, and then proposes a new neighbour $\boldsymbol{c}_{N_\Phi+1}$ with uniform probability across all possible pixel positions (where a point can be added). The new intensity can be sampled from the Gaussian prior, taking into account the available information from the neighbours, i.e., u is sampled from the conditional distribution specified in (2.8) and $m_{N_\Phi+1} = u$. The background level is adjusted to keep the total intensity of the pixel unmodified

$$b'_{i,j} = b_{i,j} - e^{m_{N_\Phi+1}} \frac{\sum_{t=1}^{T} h(t)}{T}. \tag{2.29}$$

If the resulting background level in (2.29) is negative, the move is rejected. The complementary move (named erosion) proposes to remove a point $\boldsymbol{c}_n$ with one or more neighbours. In a similar fashion to the birth move, a dilation is accepted with probability $\rho = \min\{1, C_2\}$, with C_2 computed

according to (2.25). In this case, $p(\boldsymbol{u}) = p(u_1)p(u_2)$ with

$$p(u_1) = \frac{1}{N_\Phi(2N_b+1)} \sum_{m \in \mathcal{M}_{pp}(\boldsymbol{c}_{N_\Phi+1})} \#\,\mathcal{M}_{pp}(\boldsymbol{c}_m) \tag{2.30}$$

where $0 \leq \#\,\mathcal{M}_{pp}(\boldsymbol{c}_m) \leq 8$ denotes the number of neighbouring points of $\boldsymbol{c}_m$. The expression of $p(u_2)$ is given by the conditional distribution defined in (2.8). The probability of u' is given by

$$p(u') = \frac{1}{\sum_{m=1}^{N_\Phi+1} \mathbb{1}_{\mathbb{Z}_+}(\#\,\mathcal{M}_{pp}(\boldsymbol{c}_m))} \tag{2.31}$$

and the transition probabilities are $K(\boldsymbol{\theta'}|\boldsymbol{\theta}) = p_{\text{dilation}}$ and $K(\boldsymbol{\theta}|\boldsymbol{\theta'}) = p_{\text{erosion}}$. The Jacobian term in the acceptance ratio (2.25) equals 1. An erosion move is accepted with probability $\rho = \min\{1, C_2^{-1}\}$.

Shift move The shift move modifies the position of a given point. The point is chosen uniformly at random and a new position inside the same pixel is proposed using a random walk Metropolis proposal defined as

$$u \sim \mathcal{N}(t_n, \delta_t)\,. \tag{2.32}$$

and $t'_n = u$. The resulting acceptance ratio is $\rho = \min\{1, C_3\}$, with C_3 computed according to (2.25), where $K(\boldsymbol{\theta'}|\boldsymbol{\theta}) = K(\boldsymbol{\theta}|\boldsymbol{\theta'}) = p_{\text{shift}}$, $p(u) = p(u')$ given by the Gaussian distribution of (2.32) and a Jacobian equal to 1. The value of δ_t is set to $(\frac{N_b}{3})^2$ to obtain an acceptance ratio close to 41%, which is the optimal value, as explained in [21, Chapter 4].

Mark move As in the shift move, the mark move refines the intensity value of a randomly chosen point. The corresponding proposal is a Gaussian distribution with variance δ_m

$$u \sim \mathcal{N}(m_n, \delta_m) \tag{2.33}$$

and $m'_n = u$. In this move, the acceptance ratio is $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$, where $K(\boldsymbol{\theta'}|\boldsymbol{\theta}) = K(\boldsymbol{\theta}|\boldsymbol{\theta'}) = p_{\text{mark}}$, $p(u) = p(u')$ given by (2.33) and a Jacobian equal to 1. As in the shift move, we set the value of δ_m to $(0.5)^2$ to obtain an acceptance ratio close to 41%.

Split and merge moves In lidar histograms with many photon counts per pixel, the likelihood function becomes very peaky and the non-convexity the posterior distribution becomes more difficult to handle. This non-convexity is related to the discrete nature of the point process, similar to problems where the ℓ_0 pseudo-norm regularisation is used, as discussed in [62]. In such cases, when one true surface is associated with two points, as illustrated in Fig. 2.5, the probability of performing a death move followed by a shift move is very low. To alleviate this problem, we propose a merge move and its complement, the split move. A merge move is performed by randomly

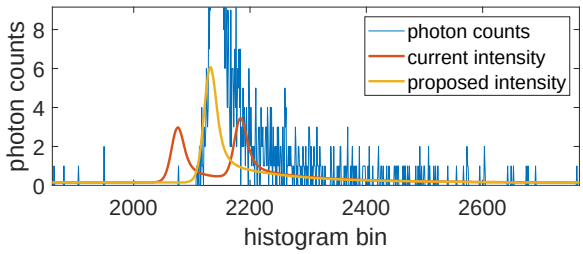


Figure 2.5 In scenarios where the sampler proposes two points (red line) instead of one (yellow line), the probability of killing one of them and shifting the other is very low. However, accepting a merge move has high probability.

choosing two points $\boldsymbol{c}_{k_1}$ and $\boldsymbol{c}_{k_2}$ inside the same pixel ($x_{k_1} = x_{k_2}$ and $y_{k_1} = y_{k_2}$) that satisfy the condition

$$d_{\min} < |t_{k_1} - t_{k_2}| \leq \text{length}_{h(t)} \tag{2.34}$$

where $\text{length}_{h(t)}$ is the length of the support of impulse response in bins. The support is obtained by thresholding $h(t)$ when it has negligible values. The merged point $(\boldsymbol{c'}_n, m'_n)$ is finally obtained by the mapping

$$\begin{cases} e^{m'_n} = e^{m_{k_1}} + e^{m_{k_2}} \\ t'_n = t_{k_1} \dfrac{e^{m_{k_1}}}{e^{m_{k_1}} + e^{m_{k_2}}} + t_{k_2} \dfrac{e^{m_{k_2}}}{e^{m_{k_1}} + e^{m_{k_2}}} \end{cases} \tag{2.35}$$

that preserves the total pixel intensity and weights the spatial shift of each peak according to its relative amplitude. For instance, if two peaks of significantly different amplitudes are merged, the resulting peak will be closer to the original peak which presents the highest amplitude. The split move randomly picks a point $(\boldsymbol{c'}_n, m'_n)$ and proposes two new points, $(\boldsymbol{c}_{k_1}, m_{k_1})$ and $(\boldsymbol{c}_{k_2}, m_{k_2})$, following the inverse mapping

$$\begin{cases} u \sim \mathcal{U}(0,1) \\ \Delta \sim \mathcal{U}(d_{\min}, \text{length}_{h(t)}) \\ m_{k_1} = m'_n + \log(u) \\ m_{k_2} = m'_n + \log(1-u) \\ t_{k_1} = t'_n - (1-u)\Delta \\ t_{k_2} = t'_n + u\Delta \end{cases} \tag{2.36}$$

which is based on the auxiliary variables u and Δ. This proposal verifies (2.35), ensuring reversibility. The acceptance ratio for the split move is $\rho = \min\{1, C_4\}$, with C_4 computed according to (2.25), where the Jacobian is $1/u(1-u)$, $K(\boldsymbol{\theta'}|\boldsymbol{\theta}) = p_{\text{shift}}$, $K(\boldsymbol{\theta}|\boldsymbol{\theta'}) = p_{\text{merge}}$, $p(u) = \frac{1}{N_\Phi}(d_{\min} + \text{length}_{h(t)})^{-1}$ and $p(u')$ is the inverse of the number of points in $\boldsymbol{\Phi}$ that verify (2.34). The acceptance probability of the merge move is simply $\rho = \min\{1, C_3^{-1}\}$.

p_{birth}	$1/24$	p_{death}	$1/24$	p_{dilation}	$5/24$	p_{erosion}	$5/24$
p_{shift}	$5/24$	p_{mark}	$5/24$	p_{split}	$1/24$	p_{merge}	$1/24$

Table 2.1 Move probabilities used in the RJ-MCMC sampler. These probabilities have been chosen through cross-validation.

2.3.2 Sampling the background

In the presence of at least one peak in a given pixel, Gibbs updates cannot be directly applied to obtain background samples, as the linear combination between the objects and the background level in (1.1) cancels the conjugacy between the Poisson likelihood and the gamma prior. However, this problem can be overcome by introducing auxiliary variables in a data augmentation scheme. In a similar fashion to [63], we propose to augment (1.1) by independent contributions, i.e.,

$$\begin{aligned} z_{i,j,t} &= \sum_{n:(x_n,y_n)=(i,j)} \tilde{z}_{i,j,t,n} + \tilde{z}_{i,j,t,b} \\ \tilde{z}_{i,j,t,b} &\sim \mathcal{P}(g_{i,j} b_{i,j}) \\ \tilde{z}_{i,j,t,n} &\sim \mathcal{P}(g_{i,j} r_n h(t - t_n)) \end{aligned}$$

where $\tilde{z}_{i,j,t,n}$ are the photons in bin t associated with the kth surface, and $\tilde{z}_{i,j,t,b}$ are the ones associated with the background. If we also add the auxiliary variables $u_{i,j}$ of the gamma Markov random field (as explained in Section 2.2.3), we can construct the following Gibbs sampler

$$\begin{cases} \tilde{z}_{i,j,t,b} \sim \mathcal{B}\left(z_{i,j,t}, \dfrac{b_{i,j}}{\sum_{n:(x_n,y_n)=(i,j)} \exp(m_n) h(t - t_n)} \right) \\ u_{i,j} \sim \mathcal{IG}(\alpha_b, \alpha_b \overline{u}_{i,j}) \\ b_{i,j} \sim \mathcal{G}\left(\alpha_b + \sum_{t=1}^{T} \tilde{z}_{i,j,t,b}, \dfrac{1}{T + \frac{\alpha_b}{\overline{b}_{i,j}}} \right) \end{cases} \tag{2.37}$$

where $\mathcal{B}(\cdot)$ denotes the Binomial distribution, $\overline{u}_{i,j}$ and $\overline{b}_{i,j}$ are defined according to (2.15) and (2.14) respectively. The transition kernel defined by (2.37) produces samples of $b_{i,j}$ distributed according to the marginal distribution of (2.16). In practice, we use only one iteration of this kernel.

2.3.3 Full algorithm

The RJ-MCMC algorithm alternates between birth, death, dilation, erosion, shift, mark, split and merge moves with probabilities as reported in Table 2.1. A complete background update is done every $N_B = N_r N_c$ iterations. After each accepted update, we compute the difference in the posterior density in order to keep track of the maximum density. After $N_{\text{bi}} = N_i/2$ burn-in iterations, we save the set of parameters $\mathbf{\Phi}$ that yield the highest posterior density and we also

accumulate the samples of $\boldsymbol{b}$ to compute (2.23). Algorithm 1 shows a pseudo-code of the resulting RJ-MCMC sampler.

Algorithm 1 ManiPoP

Input: lidar waveforms $\boldsymbol{Z}$, initial estimate $(\boldsymbol{\Phi}^{(0)}, \boldsymbol{b}^{(0)})$ and hyperparameters $\boldsymbol{\Psi}$
Initialisation:
$(\boldsymbol{\Phi}, \boldsymbol{b}) \leftarrow (\boldsymbol{\Phi}^{(0)}, \boldsymbol{b}^{(0)})$
$s \leftarrow 0$
Main loop: // Gather N_i samples from the posterior distribution
Sample the background levels every N_B iterations
while $s < N_i$ **do**
  if $\text{rem}(s, N_B) = 0$ **then**
    $(\boldsymbol{\Phi}, \boldsymbol{b}, \delta_{\text{map}}) \leftarrow$ sample $\boldsymbol{b}$ using (2.37)
  end if
  move $\sim \text{Discrete}(p_{\text{birth}}, \ldots, p_{\text{merge}})$
  $(\boldsymbol{\Phi}, \boldsymbol{b}, \delta_{\text{map}}) \leftarrow$ perform selected move
  map $\leftarrow$ map $+ \delta_{\text{map}}$
  Compute posterior statistics after burn-in iterations
  if $s \geq N_{\text{bi}}$ **then**
    $\hat{\boldsymbol{b}} \leftarrow \hat{\boldsymbol{b}} + \boldsymbol{b}$
    Save best point configuration
    if map $>$ map_{max} **then**
      $\hat{\boldsymbol{\Phi}} \leftarrow \boldsymbol{\Phi}$
      $\text{map}_{\text{max}} \leftarrow$ map
    end if
  end if
  $s \leftarrow s + 1$
end while
$\hat{\boldsymbol{b}} \leftarrow \hat{\boldsymbol{b}}/(N_i - N_{\text{bi}})$ // Normalise background levels NMSE estimator
Output: final estimates $(\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{b}})$

2.4 Efficient implementation

In order to achieve a computational performance similar to other optimisation-based approaches [36, 37], while allowing a more complex modelling of the input data, we have considered the following implementation aspects

1. Recently, the algorithm reported in [64] showed that state-of-the-art denoising of images corrupted with Poisson noise can be obtained by starting from a coarser scale and progressively refining the estimates in finer scales. We propose a similar multiscale approach to achieve faster processing times and better scalability with the total data size. This sequential procedure is detailed in Section 2.4.1.

2. In the photon-starved regime, the recorded histograms are generally extremely sparse, meaning that more than 95% of the time bins are often empty. Therefore, a histogram representation is inefficient, both in terms of likelihood evaluation and memory requirements. In [15],

the authors replaced the histograms by modelling directly each detected photon. Similarly, we represent the lidar data by using an ordered list of bins and photon counts, only considering bins with at least one count.

3. In order to avoid finding neighbours of a point to be updated at each iteration, we store and update an adjacency list for each point. This list allows the neighbour search only during the creation or shift of a point.

4. To reduce the search space, we add a pre-processing step that computes the matched-filter response at the coarsest resolution. The time bins whose values are below a threshold (equal to $\frac{0.05}{T}\sum_{t=1}^{T} z_{i,j,t} \sum_{t=1}^{T} \log h(t)$) are assigned zero intensity in the point process prior, i.e., $\lambda(\cdot) = 0$. In this way, the search includes with high probability objects in pixels with SBR higher than 0.05.

5. When the number of photons per pixel is very high, the binomial sampling step of (2.37) is replaced by a Poisson approximation, i.e.,

$$\sum_{t=1}^{T} \tilde{z}_{i,j,t,b} \sim \mathcal{P}\left(\sum_{t=1}^{T} \frac{b_{i,j} z_{i,j,t}}{\sum_{n:(x_n,y_n)=(i,j)} e^{m_n} h(t-t_n) + b_{i,j}}\right). \tag{2.38}$$

2.4.1 Multiresolution approach

Algorithm 2 Multiresolution ManiPoP

Input: lidar scene $\boldsymbol{Z}$, hyperparameters $\boldsymbol{\Psi}$, window size N_p and number of scales S
Initialisation:
$\boldsymbol{\Phi}_1^{(0)} \leftarrow \emptyset$
$\boldsymbol{b}_1^{(0)} \leftarrow$ sample from (2.37)
Main loop:
for $k = 1, \ldots, S$ **do**
  if $k > 1$ **then**
    $(\boldsymbol{\Phi}_k^{(0)}, \boldsymbol{b}_k^{(0)}) \leftarrow \text{upsample}(\hat{\boldsymbol{\Phi}}_{k-1}, \hat{\boldsymbol{b}}_{k-1})$
  end if
  $(\hat{\boldsymbol{\Phi}}_k, \hat{\boldsymbol{b}}_k) \leftarrow$ManiPoP$(\boldsymbol{Z}_k, (\boldsymbol{\Phi}_k^{(0)}, \boldsymbol{b}_k^{(0)}), \boldsymbol{\Psi})$
end for
Output: $(\hat{\boldsymbol{\Phi}}_S, \hat{\boldsymbol{b}}_S)$

We downsample the input 3D data by summing the contents over $N_p \times N_p$ pixel windows. This aggregation results in a smaller lidar image that keeps the same Poisson statistics, where each bin can present an intensity N_p^2 bigger (on average). Hence, a lidar data cube with higher SBR, has approximately N_p^2 less points to infer and a similar observational model (if the broadening of the impulse response can be neglected) is obtained. Note that no downsampling is performed along the depth axis, as the resulting SBR and computational complexity would remain unchanged, while achieving a worse resolution in depth. In this way, we run Algorithm 1 on the downsampled data

to get an initial coarse estimate of the 3D scene. This estimate is then upsampled and used as the initial condition for the finer resolution data. The point cloud $\mathbf{\Phi}$ is upsampled using a linear interpolator for fast computation. Following the connected-surface structure of ManiPoP, each of the estimated surfaces is upsampled independently of the rest. However, more elaborate algorithms can be also used, such as moving least squares (MLS), as detailed in [65]. These two steps can be performed in S scales, whereby, for each scale, the lidar data $\boldsymbol{Z}_k$ is obtained by aggregating $\boldsymbol{Z}_{k+1}$. Algorithm 2 summarises the sequential multiscale approach.

2.5 Experiments

We evaluated ManiPoP using synthetic and real lidar data. In all experiments, we denote the bin length as $\Delta_b = \frac{\Delta_t c}{2}$, where c is the speed of light in the scene medium and Δ_t is the bin width used in the TCSPC timing histogram. We also indicate PPP, which is proportional to the per pixel acquisition time. Our method is compared with the classical cross-correlation solution and two algorithms that estimate an intensity cube, SPISTA [36] and ℓ_{21}+TV [37]. In our experiments, we have slightly modified both SPISTA and ℓ_{21}+TV to attain better results, as explained in Section 2.5.2. The RJ-MCMC algorithm proposed in [39] was not considered in this comparison as its computational complexity is incompatible with large images (for a scene of $N_r = N_c = 100$ pixels and $T = 4500$ bins, the algorithm takes more than a day of computation). For visualisation purposes, all the intensity results obtained by different algorithms were normalised (post-processing step) under the condition $\sum_{t=1}^{T} h(t) = 1$, such that the estimated intensity has a value that reflects the amount of signal photons attributed to the corresponding 3D location. In the experiments, we used only 2 scales, a coarse one using a binning window of $N_p = 3$ pixels and the full resolution. The hyperparameters were adjusted with the following considerations

- The cuboid length N_b should be fixed according to the relative scale between the bin width and the pixel resolution. In our real data experiments, we set N_b to $8\Delta_p/\Delta_b$.
- The minimum distance between two points in the same pixel can be set as $d_{\min} = 2N_b + 1$, thus verifying the condition $d_{\min} > 2N_b$.
- The parameters controlling the number of points and the spatial correlation were set by cross-validation using many lidar data sets.
- For each scale, we scaled the impulse response by $\frac{\text{PPP}}{5\sum_t h(t)}$, such that all intensity values lie approximately in the interval $[0, 10]$. The regularisation parameters were then fixed to $\sigma^2 = 0.6^2$ and $\beta = \sigma^2/100$ by cross-validation in order to obtain smooth estimates.
- The hyperparameter controlling the smoothness in the background image $\boldsymbol{b}$ was also adjusted by cross-validation leading to $\alpha_b = 2$.

Hyperparam.	γ_a	λ_a	N_p	N_b	$d_{\min}$	σ^2	β	α_B
Coarse scale	e^2	$(N_r N_c/N_p^2)^{1.5}$	3	$3\Delta_p/\Delta_b$	$2N_b+1$	0.6^2	$\sigma^2/100$	2
Fine scale	e^3	$(N_r N_c)^{1.5}$	-	$3\Delta_p/\Delta_b$	$2N_b+1$	$0.6^2/3$	$\sigma^2/100$	2

Table 2.2 Hyperparameters values.

Table 2.2 summarises the different hyperparameter values for the coarse and fine scales. All the experiments were performed using $N_i = 25N_rN_c$ iterations in the coarse scale and finest scale.

2.5.1 Error metrics

Three different error metrics are used to evaluate the performance of the multi-depth algorithm. We compare the percentage of true detections $F_{\text{true}}(\tau)$ as a function of the distance τ, considering an estimated point as a true detection if there is another point in the ground truth/reference point cloud in the same pixel ($x_n^{\text{true}} = \hat{x}_{n'}$ and $y_n^{\text{true}} = \hat{y}_{n'}$) such that $|t_n^{\text{true}} - \hat{t}_{n'}| \leq \tau$. We also consider the number of points that were falsely created denoted as $F_{\text{false}}(\tau)$ (i.e., the estimated points that cannot be assigned to any true point at a distance of τ). Regarding the intensity estimates, we focus on target-wise comparison, by gating the 3D reconstruction between the ranges where a specific target can be found, keeping only the point with biggest intensity and assigning zero intensity to the empty pixels. We computed the normalised mean squared error of the resulting 2D intensity image as

$$\text{NMSE}_{\text{target}} = \frac{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}(r_{i,j}^{\text{true}} - \hat{r}_{i,j})^2}{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}\left(r_{i,j}^{\text{true}}\right)^2}. \tag{2.39}$$

Finally, we consider the NMSE metric for the background image

$$\text{NMSE}_{\boldsymbol{b}} = \frac{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}(b_{i,j}^{\text{true}} - \hat{b}_{i,j})^2}{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}\left(b_{i,j}^{\text{true}}\right)^2}. \tag{2.40}$$

2.5.2 Synthetic data

We evaluated the algorithm for two synthetic datasets: A simple one, containing basic geometric shapes and a complex one, based on a scene from the Middlebury dataset [66]. Both scenes present multiple surfaces per pixel. The first scene, shown in Fig. 2.6, has dimensions $N_r = N_c = 99$, $T = 4500$, $\Delta_b = 1.2$ mm and $\Delta_p \approx 8.5$ mm. The impulse response used in our experiments was obtained from real lidar measurements, with $\text{length}_{h(t)} = 518$ bins. The background was created using a linear intensity profile, as shown in Fig. 2.6. The resulting PPP was 11 photons, meaning that 99.75% of the bins are empty and approximately 4 photons per pixel are due to 3D objects. First we evaluated the performance with and without the proposed priors to show their effect on the final estimates. The algorithm was tested in the following conditions

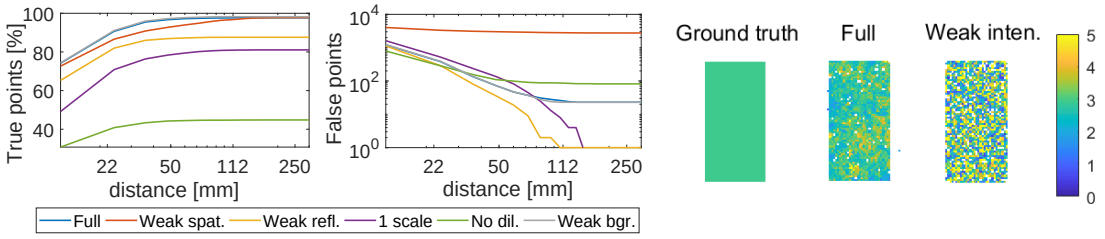
1. With all the priors as reported in Table 2.2.

(a) **(b)**

Figure 2.6 The 3D scene depicted in Fig. 2.1 consists in 3 plates with different sizes and orientations and one paraboloid shaped object. Left: Number of objects per pixel. Right: Mean background photon count $T\boldsymbol{b}$.

2. Without spatial regularisation ($\gamma_a = 1$).
3. With a weak intensity regularisation ($\sigma^2 = 100^2$).
4. With a softer spatial regularisation for the background levels ($\alpha_b = 1$).
5. Without erosion and dilation moves.
6. Only using the finest scale, adjusting the number of iterations to yield the same computing time.

The total execution time for all cases was approximately 120 seconds. Figure 2.7a shows $F_{\text{true}}(\tau)$ and $F_{\text{false}}(\tau)$ for all the configurations. The number of false points increases dramatically when the area interaction process is not considered, as the sampler tends to create many points of low intensity, mistaking background counts as false surfaces. The background regularisation does not affect the detected points significantly, but yields a better estimation of $\boldsymbol{b}$, leading to $\text{NMSE}_{\boldsymbol{b}} = 0.107$ for $\alpha_b = 1$ and $\text{NMSE}_{\boldsymbol{b}} = 0.0912$ for $\alpha_b = 2$. The number of true points detected without dilation and erosion moves or using only one scale decreases dramatically to 44% and 80% respectively. Figure 2.7b compares the estimated intensity of the biggest plate with different values of σ^2. The



(a) **(b)**

Figure 2.7 (a) shows the percentage of true and false detections. The intensity estimates for the vertical plate are shown in (b): Ground truth (left), estimates with $\sigma^2 = 0.6^2$ (center) and $\sigma^2 = 100^2$ (right).

NMSE obtained with $\sigma^2 = 0.6^2$ is 0.058, compared to 0.399 in the absence of correlation (i.e., when $\sigma^2 = 100^2$).

The second dataset was created with the "Art" scene from [66]. In order to have multiple surfaces per pixel, we added a semi-transparent plane in front of the scene. We simulated the

Method	Total time [seconds]	$\text{NMSE}_{\text{target}}$
SPISTA [36]	712	> 1
SPISTA+	8161	0.993
ℓ_{21}+TV [37]	2453	0.845
ℓ_{21}+TV group	2455	0.845
ManiPoP	**630**	**0.0999**

Table 2.3 Performance of ManiPoP, SPISTA, SPISTA+, ℓ_{21}+TV and ℓ_{21}+TV with grouping on the synthetic data. All the algorithms are implemented in MATLAB.

lidar measurements, as if they were taken by the system described in [11]. The scene consists in $N_r = 183$ and $N_c = 231$ pixels, and $T = 4500$ histogram bins. The bin width is $\Delta_b = 0.3$ mm and the pixel size is $\Delta_p \approx 1.2$ mm. In this complex scene, we compared ManiPoP with the optimisation algorithms SPISTA and ℓ_{21}+TV. SPISTA relies on the specification of a background level that was set to the true background value. It is important to note that this information is not available in real lidar applications, as the background levels depend on the imaged scene. We also show the results for the regularisation parameter that attained best results among many trials (the empirical rule for setting this parameter provided in [36] achieved worse results). We noticed that SPISTA provides large errors in the intensity estimates, as the algorithm can sometimes diverge in very low-photon scenarios. This is due to the fixed step size used in the proximal gradient scheme of SPISTA, which is not compatible with the non-Lipschitz globality of the gradient of the Poisson likelihood [67]. This problem can be solved using the SPIRAL [26] inner loop to compute the step size, yielding a modified algorithm, which we name SPISTA+. The ℓ_{21}+TV algorithm has 2 regularisation parameters that were adjusted in order to obtain the best results. It also relies on a thresholding step on the final estimates, as the output of the optimisation method is not sparse. Again, the thresholding constant was adjusted to achieve the best results. To further improve the results of ℓ_{21}+TV, we included a grouping step, similar to the one of SPISTA, which reduces the number of false detections by pairing similar ones in the same pixel. Instead of taking the maximum intensity as in [36], we summed the intensities of the grouped detections, as it achieved better intensity estimates. Figure 2.8 shows the 3D point clouds obtained for each algorithm whereas Fig. 2.9 shows $F_{\text{true}}(\tau)$ and $F_{\text{false}}(\tau)$. SPISTA finds 18% of the true points and around 5033 false detections, whereas SPISTA+ improves the detection to 34% and a 4267 false detections. ℓ_{21}+TV improves the detection rate to 57%, but also increases the false detections to 10^6. The grouping technique improves the results provided by ℓ_{21}+TV, reducing the false detections by a factor of 200. ManiPoP provides the best results, finding 92% of all the true points and 1852 false detections. As shown in Table 2.3, ManiPoP yields the best intensity estimates with the lowest execution time. Figure 2.10 shows the intensity estimate of the scene behind the semitransparent plane for each algorithm. SPISTA fails to provide meaningful intensity results, whereas SPISTA+ yields better estimates. As all the points that are behind the plane are grouped to yield a 2D intensity image, there is no difference between the ℓ_{21}+TV and ℓ_{21}+TV with grouping. Both

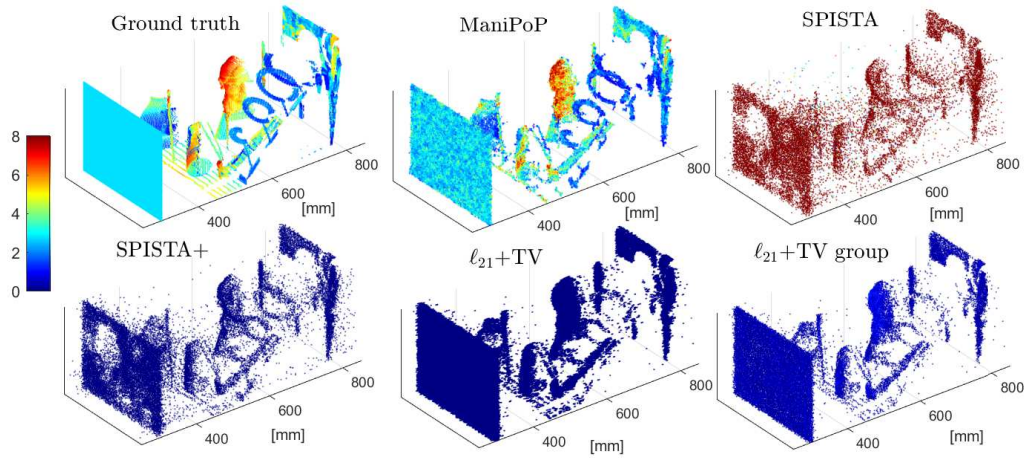


Figure 2.8 Estimated 3D point cloud by ManiPoP, SPISTA, SPISTA+, ℓ_{21}+TV and ℓ_{21}+TV with grouping.

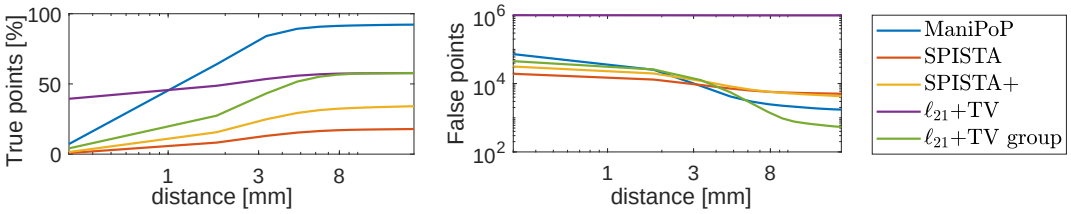


Figure 2.9 (a) Percentage of true detections for different algorithms as a function of maximum distance τ, $F_{\text{true}}(\tau)$. (b) Number of false detections, $F_{\text{false}}(\tau)$.

SPISTA+ and ℓ_{21}+TV with grouping show a negative bias in the mean intensity, which may be attributed to the effect of the ℓ_1 and ℓ_{21} regularisations respectively.

As both SPISTA+ and ℓ_{21}+TV with grouping improve the results of the original algorithms in all the evaluated datasets, we only show their results in the rest of the experiments throughout the thesis.

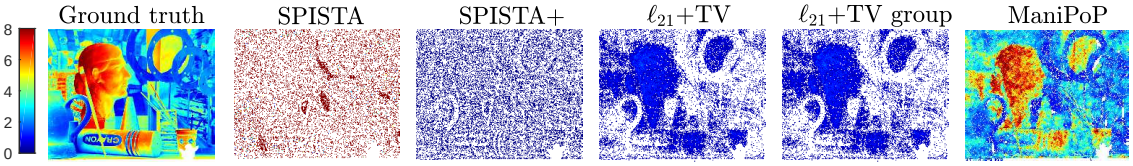


Figure 2.10 Intensity estimates of the surfaces behind the semi-transparent object.

2.5.3 Real lidar data

We assessed ManiPoP using 3 different lidar datasets: the multi-layered scene provided in [36, 68] recorded in the Massachusetts Institute of Technology, the polystyrene target imaged in Heriot-Watt University [16] and the camouflage scene from [37].

Mannequin behind scatterer The first scene consists of a mannequin located 4 meters behind a partially scattering object, with $N_r = N_c = 100$ pixels and $T = 4000$ bins. This lidar scene is publicly available online [68]. The scene has 45 PPP and the dimensions are $\Delta_p \approx 8.4$ mm and $\Delta_b = 1.2$ mm. Figure 2.11 shows the reconstructed point clouds for each algorithm. ManiPoP achieves a sparse and smooth solution, whereas the estimate of SPISTA presents more random scattering of points. The ℓ_{21}+TV output presents more spatial structure than SPISTA, but also fails to find the the border of the mannequin. The dataset contains a reference depth of the

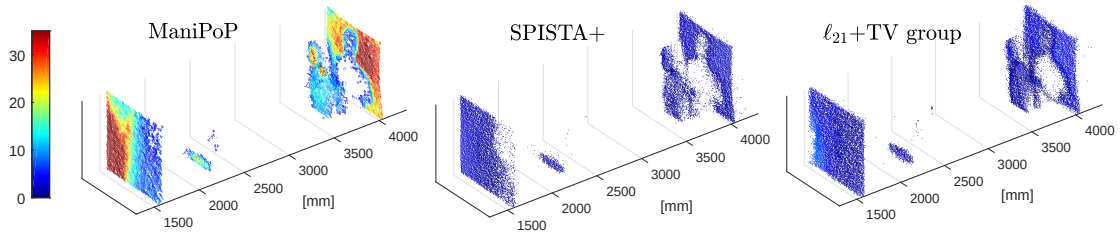


Figure 2.11 Estimated 3D point cloud by ManiPoP, SPISTA+ and ℓ_{21}+TV with grouping.

mannequin obtained using a long acquisition time. This reference was computed using cross-correlation on a cropped lidar cuboid where only the mannequin is present. Figure 2.13 shows the ground truth depth and the estimates obtained by ManiPoP, SPISTA+ and ℓ_{21}+TV with grouping. ManiPoP outperforms the SPISTA+ and ℓ_{21}+TV outputs, finding 97.9% of the reference detections, whereas SPISTA+ only detects 74.8% and ℓ_{21}+TV with grouping finds 92.8%, as shown in Fig. 2.12. The SPISTA+ and ℓ_{21}+TV with grouping algorithms detect 225 and 206 false points respectively, compared to the 432 points found by ManiPoP. This increase in false detections can be attributed to the scattering object that was (probably) removed when the reference dataset was obtained. The scattering effect can be also seen in Fig. 2.11, as it is possible to find some parts of the low intensity surface behind the mannequin. Despite not having a reference for reflectivity values of the target, we can say that ManiPoP attains significantly better visual results, as shown in Fig. 2.14. Both SPISTA+ and ℓ_{21}+TV with grouping underestimate the mean intensity. The total execution time of ManiPoP (146 seconds) was around 20 times less than SPISTA+ (2871 seconds) and slightly shorter than ℓ_{21}+TV with grouping (202 seconds).

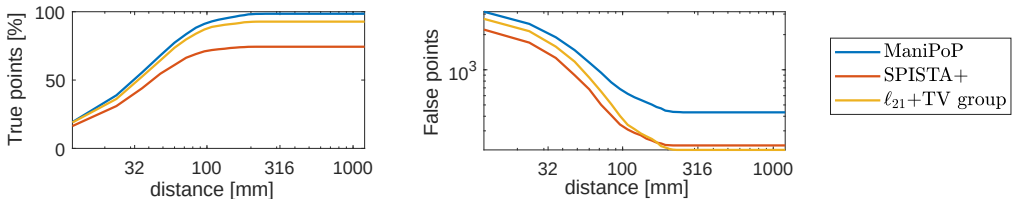


Figure 2.12 Percentage of true detections at a maximum distance τ for the mannequin behind scatterer dataset, $F_{\text{true}}(\tau)$, for ManiPoP, SPISTA+ and ℓ_{21}+TV with grouping. The number of false detections, $F_{\text{false}}(\tau)$, is shown in (b).

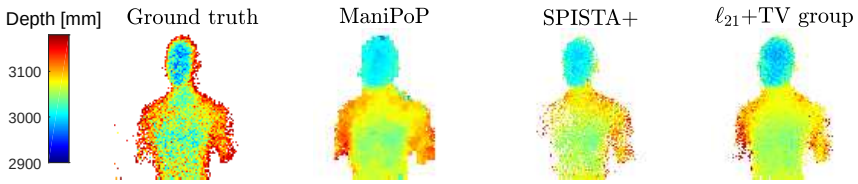


Figure 2.13 Depth estimates of the mannequin. From left to right: long acquisition reference, ManiPoP, SPISTA+ and ℓ_{21}+TV with grouping estimates.

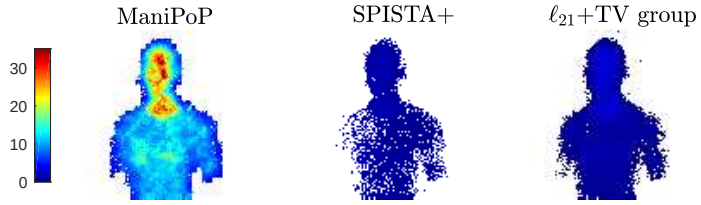


Figure 2.14 Estimated intensity by ManiPoP, SPISTA+ and ℓ_{21}+TV with grouping for the mannequin behind scatterer dataset. The colourbar illustrates the number of photons assigned to each point. Both SPISTA+ and ℓ_{21}+TV show a negative bias in the mean intensity.

Head with backplane The second dataset was obtained in Heriot-Watt University and consists of a life-sized polystyrene head at 40 meters from the imaging device (an image can be found in [16]). The data cuboid has size $N_r = N_c = 141$ pixels and $T = 4613$ bins. The physical dimensions are $\Delta_p \approx 2.1$ mm and $\Delta_b = 0.3$ mm. A total acquisition time of 100 ms was used for each pixel, yielding 337 PPP with approximately 23 background photons per pixel. The scene consists mainly in one object per pixel, only with 2 surfaces per pixel around the borders of the head. We compare ManiPoP with cross-correlation and the SPISTA+ algorithm for different acquisition times, i.e., many values of PPP. As no ground truth is available, we used as reference the cross-correlation estimate, manually dividing the lidar cube into segments with only one surface, using the largest acquisition time (100 ms). Although the dataset seems to have only one active depth per pixel, two surfaces per pixel can be found in the borders of the head, as shown in Fig. 2.15. As only a few pixels contain two surfaces, we also compared with Rapp and Goyal [25], which is a state-of-the-art 3D reconstruction algorithm under the single-depth assumption. Figure 2.16 shows the

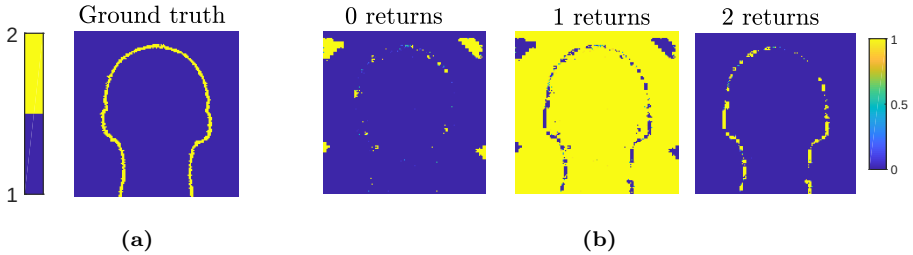


Figure 2.15 (a) True number of surfaces per pixel. (b) Probability of having $k = 0, 1, 2$ objects per pixel for an acquisition time of 1 ms.

Algo./Acq. time	100 ms (337 PPP)	10 ms (33.7 PPP)	1 ms (3.4 PPP)	0.2 ms (0.7 PPP)
SPISTA+ [36]	6769	6981	7191	8461
ℓ_{21}+TV group [37]	793	697	705	535.4
ManiPoP	322	229	201	173.4
Cross-corr.	**18**	**11**	**7.8**	**5.6**
Rapp and Goyal [25]	196.87	40	37	38.4

Table 2.4 Computing time of ManiPoP, SPISTA+, ℓ_{21}+TV, cross-correlation and [25] on the "head with backplane" dataset.

reconstructed 3D point clouds for an acquisition time of 1 ms whereas Fig. 2.17 shows $F_{\text{true}}(\tau)$ and $F_{\text{false}}(\tau)$ for acquisition times of 10, 1 and 0.2 ms. In the 10 and 1 ms cases, ManiPoP outperforms the other methods, finding almost all true points and providing relatively few false estimates. Cross-correlation (of the complete lidar cube) shows a significant error in depth estimates and fails to find 10% of true points, as it is only capable of finding one object per pixel. In the 0.2 ms case, there are only 0.7 PPP. Thus, the best performing algorithm is Rapp and Goyal, as the single-surface assumption plays a fundamental role to inpaint the missing depth information. ManiPoP performs in second place, finding 14% less true points than Rapp and Goyal.

The fastest algorithm is cross-correlation with less than 20 seconds in all cases. However, ManiPoP still requires less computing time than SPISTA+ and ℓ_{21}+TV with grouping. It is worth noticing that the ℓ_{21}+TV algorithm has a memory requirement proportional to 6 times the whole data cube due to the ADMM algorithm, which can be prohibitively large when the lidar cube is relatively big. The sparse nature of the ManiPoP algorithm only requires an amount of memory proportional to the number of bins with one photon or more plus the number of 3D points to infer.

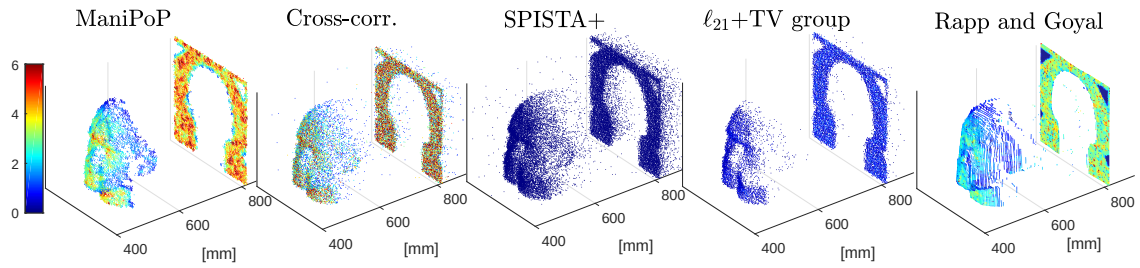


Figure 2.16 Estimated 3D point clouds using the polystyrene head dataset with an acquisition time of 1 ms. SPISTA+ and ℓ_{21}+TV underestimate the mean intensity, whereas ManiPoP, cross-correlation and Rapp and Goyal obtain similar mean intensity.

Head without backplane To further demonstrate the generality of ManiPoP, we studied the case where only one surface is present per pixel, but not all the pixels contain surfaces, which occurs in most outdoor measurements. If a single-surface per pixel algorithm is used [10,14–16,25], a non-trivial post-processing step is necessary to discriminate which pixels have active depths. We also

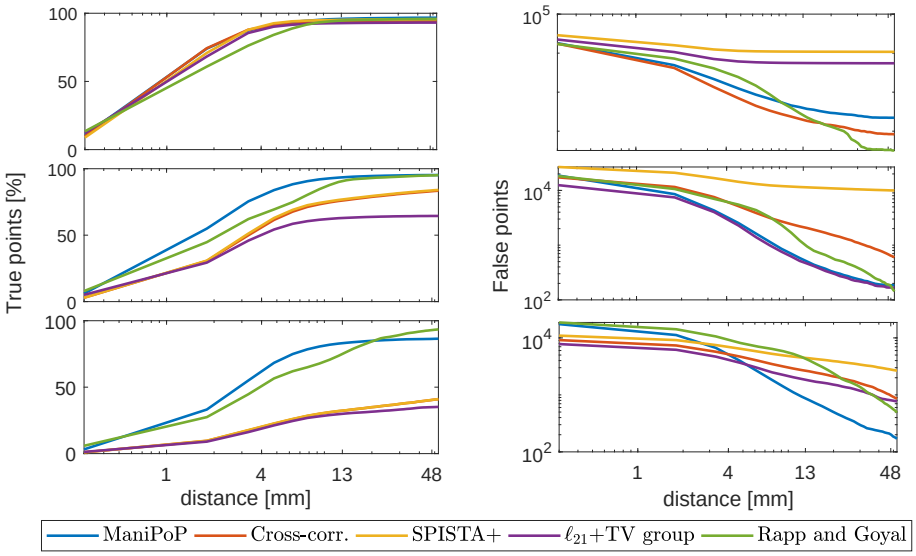


Figure 2.17 $F_{\text{true}}(\tau)$ and $F_{\text{false}}(\tau)$ for the polystyrene head using acquisition times of 10 ms (top), 1 ms (middle) and 0.2 ms (bottom). While all methods obtain good reconstructions in the 10 ms case, ManiPoP and [25] also achieve good reconstructions with acquisition times of 1 ms and 0.2 ms.

included the results obtained by the Bayesian target detection algorithm [35], which assumes at most one surface per pixel. To recreate this case using the polystyrene head dataset, we removed the backplane from the 1 ms dataset, obtaining a new 3D lidar cube that only contains the polystyrene head. Figure 2.19 shows the results obtained by ManiPoP and the competing algorithms. In the latter, we applied a global thresholding based on the recovered reflectivity values, such that only the target would be present in the final results. The value of the threshold was manually chosen to obtain the best results. ManiPoP obtains the best results, finding 95.2% of the points with only 24 false detections, whereas the single-depth method finds 93.1% of the points and 542 false detections and the detection algorithm obtains 86.0% of the points and 849 false detections. As shown in Fig. 2.18, the estimates of Rapp and Goyal degrade significantly towards the borders of the target, as the single-surface assumption imposes a false correlation with the background photons in neighbouring pixels where no surface is present. While the target detection algorithm performs similarly in terms of true and false point detections than ManiPoP, the depth and reflectivity estimates are worse. This result can be attributed to the lack of prior spatial correlation for the depth and reflectivity values of the target detection method [35].

Note that the samples generated by the RJ-MCMC chain are asymptotically distributed according to the posterior (2.18) and can thus be used to compute various uncertainty measures. For instance, Fig. 2.15 shows the probability of having $k = 0, 1, 2$ peaks for an acquisition time of 1 ms, computed according to (2.24). Another example is displayed in Fig. 2.20, which shows the position and log-intensity histograms that were computed using the samples from additional $N_i = 400 N_r N_c$ iterations in a fixed dimension (only allowing mark and shift moves).

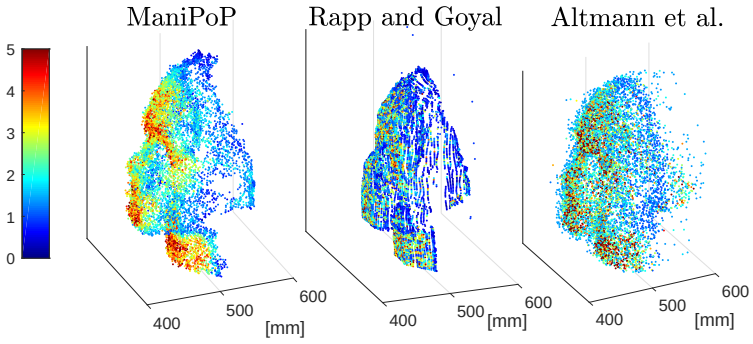


Figure 2.18 (a) 3D reconstructions in a target detection scenario (head without backplane with an acquisition time of 1 ms).

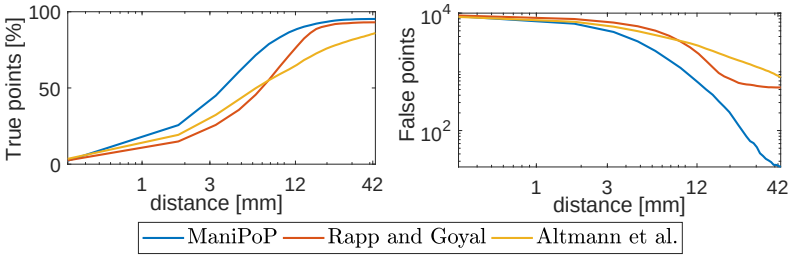


Figure 2.19 $F_{\text{true}}(\tau)$ and $F_{\text{false}}(\tau)$ for the head without backplane dataset with an acquisition time of 1 ms.

Human behind camouflage The last dataset consists of a man standing behind camouflage at a stand-off distance of 230 meters from the lidar system. An in-depth description of the scene can be found in [8,37]. An acquisition time of 3.2 ms was used for each pixel, obtaining 44.6 PPP, where approximately 13.3 photons correspond to background levels. The lidar cube has $N_r = 159$ and $N_c = 78$ pixels, and $T = 550$ histogram bins. The physical dimensions are $\Delta_p \approx 2.1$ mm and $\Delta_b = 5.6$ mm. We evaluated the performance of the algorithms for the per-pixel acquisition times of 3.2 ms and 0.32 ms. Figure 2.21 shows the reconstructions obtained by ManiPoP, SPISTA+ and TV+ℓ_{21} with depth grouping. In both cases, ManiPoP obtains a more structured reconstruction, without spurious detections and more dense reconstructions in the regions where the target is present.

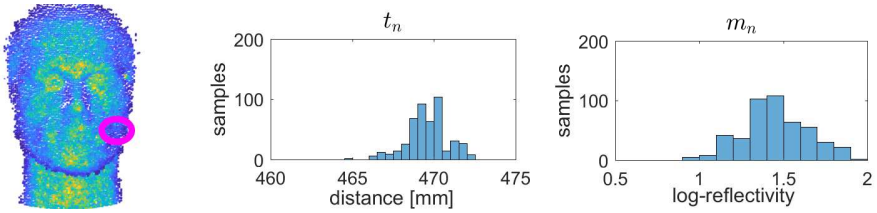


Figure 2.20 The center and right plots show the position and log-intensity histograms for the point encircled in violet in the left plot, using the head without backplane dataset with an acquisition time of 1 ms.

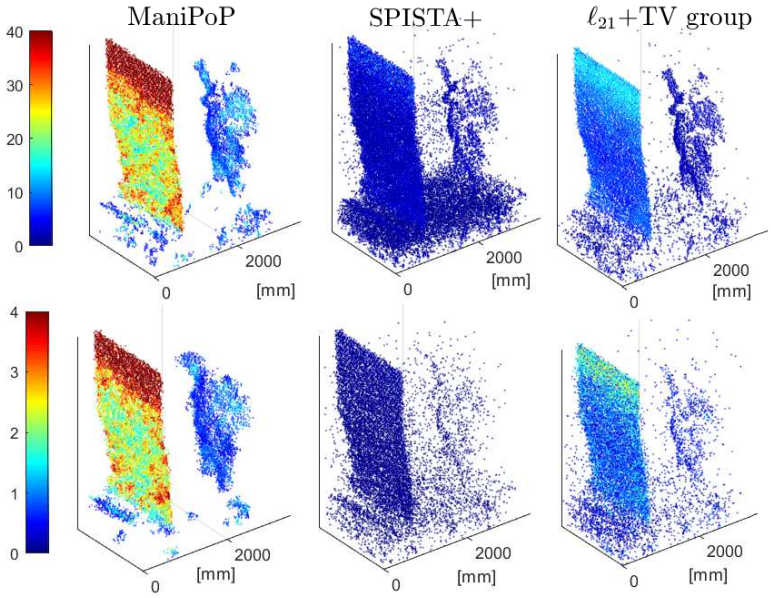


Figure 2.21 Estimated 3D point clouds using the camouflage dataset for per-pixel acquisition times of 3.2 ms (top row) and 0.32 ms (bottom row).

2.6 ManiPoP+

Despite estimating a varying number of surfaces per pixel, the observation model considered by ManiPoP (1.1) assumes a fixed impulse response $h(t)$ and negligible scattering of the light reflected onto the detector. The fixed impulse response assumption does not hold in very long-range scenes, where the observed $h(t)$ is broader when the target surface is not orthogonal to the illumination beam. The amount of broadening can be related to the angle between the laser beam and the imaged surface or to the local porosity of the object (light penetrating deeper into the object), as explained in [69]. Moreover, in scenes with highly attenuating media, e.g., underwater conditions, fewer photons are recorded as the target gets further away from the detector [9]. These additional effects can be incorporated in the observation model as follows

$$z_{i,j,t}|\mathbf{\Phi}, b_{i,j} \sim \mathcal{P}\left(g_{i,j}s_{i,j,t} + g_{i,j}b_{i,j}\right) \tag{2.41}$$

where the signal $s_{i,j,t}$ is expressed as

$$s_{i,j,t} = \sum_{n:(x_n,y_n)=(i,j)} r_n e^{-\alpha \Delta_b t_n} h_{\eta_n}(t - t_n) \tag{2.42}$$

with α the scattering coefficient (e.g., $\alpha \approx 0.6$ for clear water). The instrumental response of the device with width parameter $\eta \in [1, +\infty)$ is denoted by $h_\eta(t)$ and modelled using a Gaussian kernel as $h_\eta(t) \propto \sum_k h(k) \exp\left[-\frac{(t-k)^2}{2(\eta-1)^2}\right]$, where $h(t)$ is the instrumental response without broadening and is typically obtained during the calibration of the device. Note that the signal model in (1.1) can be recovered by assuming no attenuation (i.e., $\alpha = 0$) and no broadening of $h(t)$ (i.e., $\eta_n = 1$ $\forall n$).

In this section, we extend ManiPoP to account for these additional effects. The resulting algorithm, referred to as ManiPoP+, assigns a prior distribution to the additional peak broadening parameters η_n and relies on slightly modified RJ-MCMC moves, where the log-intensity m_n is replaced by the adjusted version $m_n - \alpha t_n$. The rest of the chapter presents the prior distribution of the broadening parameters and shows results in long-range and underwater conditions.

2.6.1 Broadening parameters

Points in a small neighbourhood of a surface usually present a similar amount of broadening, as the laser beam has a similar angle of incidence on them or they present similar porosity. Thus, similarly to the log-intensity, we assign to the set of $\tilde{\eta}_n$ a Gaussian Markov random field prior (2.10), which promotes spatial correlations between neighbouring widths, with hyperparameters $(\beta_{\tilde{\eta}}, \sigma^2_{\tilde{\eta}})$ instead of (β_m, σ^2_m). As with the log-intensity (2.7), we introduce the transformation

$$\tilde{\eta}_n = \log(\eta_n - 1) \tag{2.43}$$

to remove the constraint on the width, such that $\tilde{\eta}_n \in \mathbb{R}$ $\forall n$.

2.6.2 Long-range results

The dataset presented in [6] consists of the dome of a building, imaged using terrestrial lidar from a stand-off distance of approximately 3 kilometres. The lidar cube has a size of $N_r = 123$ and $N_c = 96$ pixels, and $T = 801$ histogram bins. There are 913 PPP and a mean SBR of 1.64. In this case, the medium is air with a negligible scattering effect, i.e., $\alpha \approx 0$. Figure 2.22 shows the reconstruction obtained by ManiPoP. The estimated point widths are consistent with the orientation of the surface with respect to the incoming laser. For example, the lower part of the roof presents a significant broadening η when the surface normal has a significant angle with respect to the laser.

Figure 2.23 compares the estimated Poisson intensities obtained by ManiPoP and ManiPoP+

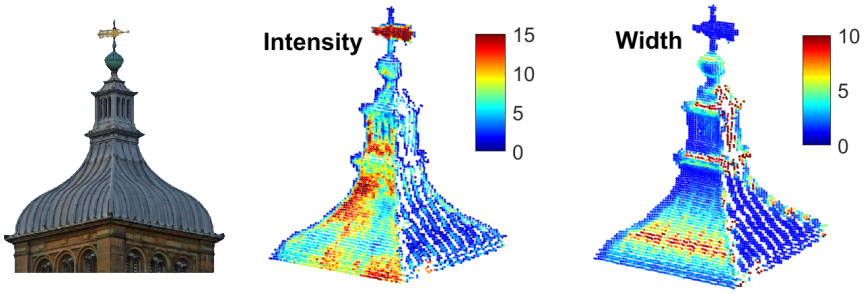
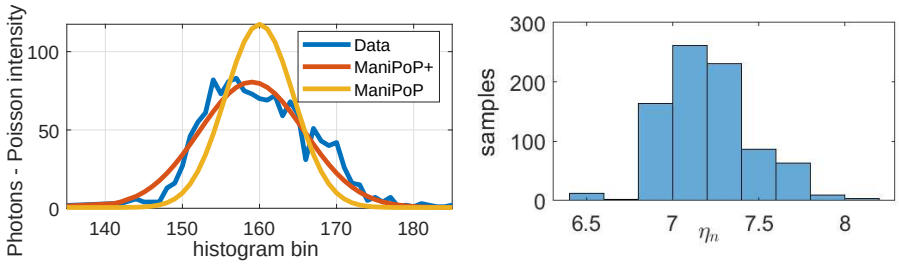


Figure 2.22 Right: RGB image of the imaged dome (taken from a closer distance). The estimated point cloud intensities and widths from the college dataset are shown on the middle and left figures, respectively. The incoming laser beam is orthogonal to the left hand side of the roof.



(a) (b)

Figure 2.23 (a) Estimated intensity for a pixel in the lower roof by ManiPoP and ManiPoP+. (b) Histogram of width samples obtained by the RJ-MCMC algorithm for the same pixel.

for one of those pixels. The ManiPoP algorithm does not take into account the broadening of the peak, thus underestimating the reflectivity by 5%, whereas ManiPoP+ provides an accuracy of 1%. Moreover, the estimation of the width does not significantly affect the computational load, as ManiPoP requires an execution time of 174 seconds, whereas ManiPoP+ requires 195 seconds.

2.6.3 Highly attenuating media results

The underwater scene presented in [9] is composed of a pipe inside a water tank, measured at a distance of 178 cm, as shown in Fig. 2.24. The measurements were repeated 3 times under varying concentrations of Maalox in the water, obtaining the scattering coefficients $\alpha \in [0.6, 3.9, 4.8]$. The lidar cube has $N_r = N_c = 120$ pixels and $T = 2500$ histogram bins, and the acquisition time was 100 ms in all cases.

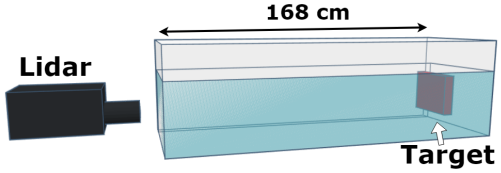


Figure 2.24 The underwater measurements were taken at a stand-off distance of 178 cm from the target, where 168 cm correspond to the water tank medium.

We compare the results with ManiPoP reconstructions followed by a simple post-processing correction of the estimated intensities, i.e., by dividing them by the attenuation factor based on the estimated range. Figure 2.25 shows the estimated point clouds for all the values of α for both algorithms. The reconstructions obtained by ManiPoP have a lower variation in the estimated intensity. Moreover, in the case with highest attenuation (i.e., $\alpha = 4.8$), ManiPoP fails to recover the backplane of the scene, as its mean intensity (without the exponential term correction) is too low and the algorithm considers it as belonging to the background.

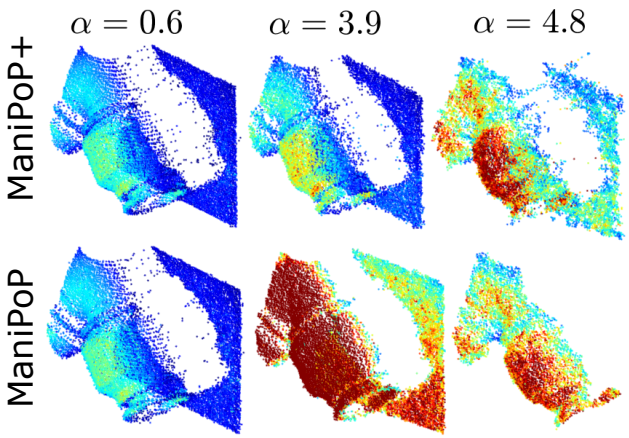


Figure 2.25 The reconstructions by ManiPoP+ are shown on the top row, whereas the reconstructions achieved by ManiPoP are shown in the lower row. The incoming laser beam is orthogonal to the backplane. All the intensities are shown in the same colourmap scale.

2.7 Conclusions

In this chapter, we have proposed a new Bayesian spatial point process model for describing single-photon depth images. This model promotes spatially correlated and sparse structures, which can be interpreted as a structured ℓ_0 pseudo-norm regularisation [70]. Finding the MAP estimate of these models is an NP-hard problem [71]. We overcame this problem by developing a stochastic RJ-MCMC algorithm with new moves that find a solution relatively fast. In addition, a multiresolution approach improved the estimates and reduced the execution time. ManiPoP yields good 3D reconstructions, with better depth and intensity estimates than other competing methods. In our experiments, we noted that for each dataset, a different set of hyperparameters and thresholding values is needed both for SPISTA and ℓ_{21}+TV, thus making user supervision compulsory, whereas the proposed algorithms use the same set of hyperparameters across all datasets. In extremely low-photon cases, i.e., less than one photon per pixel on average, ManiPoP might fail to recover the surface, thus performing worse than other single-depth 3D reconstruction algorithms [25]. The general formulation of ManiPoP can be easily extended to account for different observation models (e.g., (2.41)) and include additional parameters, such as peak broadening marks. Chapter 3

introduces an adaptation of ManiPoP for multispectral single-photon lidar.

The algorithm requires less execution time when compared to other optimisation [36, 37] and RJ-MCMC approaches [38,39]. Moreover, a more tailored C++ implementation with efficient handling of the connected-surface structure would further reduce the computing time considerably. A profiling analysis of the current code shows that around 70% of the total computational time is due to these computations. However, the Markovian structure of the algorithm places a fundamental limitation on the minimum execution time achievable. Chapters 4 and 5 present methods that by-pass this limitation, presenting algorithms that are designed for real-time reconstructions.

Chapter 3

Multispectral 3D imaging

Contents

3.1 Introduction

Multispectral lidar (MSL) systems gather measurements at many spectral bands, making it possible to distinguish distinct materials, as illustrated in Fig. 3.1. For example, spectral diversity was used in [18] to differentiate leaves from trunks and in [72] to estimate plant area indices and abundance profiles. The MSL modality consists of constructing one histogram of time delays per wavelength, as shown in Fig. 3.2. The spectral diversity can be obtained either using a supercontinuum laser source [72,73] or multiple lasers [74]. The detector generally consists of a spatial form of wavelength routing to demultiplex the channels [72–74] or wavelength-to-time codification [69].

Figure 3.1 An airborne MSL system can capture multiple objects per pixel and discriminate their materials. The multi-depth capability enables the recovery of information from photons reflected off different branches of the trees, ground, pedestrians or even from the interior of a car (i.e., photons which propagated across the windshield) at an intra-pixel level. Moreover, the multispectral information allows us to discriminate different properties of the materials of each 3D object (e.g., the leaves and trunk of a tree).

In the multispectral case, only single-depth algorithms have been proposed [59, 73, 75], with the exception of [72], which simply runs a single-wavelength multi-depth algorithm [38] on the most powerful wavelength. The single-depth assumption greatly simplifies the reconstruction problem, as it significantly reduces memory requirements and overall complexity. For example, the single-depth algorithm introduced in [75] estimates the reflectivities using accumulated histograms, avoiding to work on the entire dataset at once. In contrast, multi-depth algorithms generally have to access the full histograms, where such divide-and-conquer strategy cannot be easily applied. Datasets containing dozens of wavelengths can be prohibitively large for practical multi-depth algorithms, both in terms of memory and computing requirements. For example, a typical MSL hypercube with 32 wavelengths has more than 10^9 data voxels. To alleviate this problem, some compressive acquisition strategies have been proposed. While TCSPC technology hinders compressive techniques along the depth axis[1], reducing the number of measurements can be achieved by integrating multiple wavelengths into a single histogram [76] or measuring fewer histograms (i.e., subsampling) [75]. The wavelength-to-time approach proposed by Ren et al. [76] is not well-suited in the presence of multiple surfaces per pixel. This method compresses L histograms (associated with L wavelengths) into a single waveform by shifting in time the photon detections according to each measured wavelength. While significantly reducing the data size, the resulting likelihood becomes highly multimodal and extremely difficult to handle in the presence of multiple surfaces. Following another direction, different random subsampling schemes were studied in [75] without obtaining any significant differences in terms of reconstruction quality in the low-photon count regime.

In this chapter, we investigate a new pseudo-random subsampling scheme for low-photon count MSL data based on ideas from coded aperture design [77, 78]. By choosing the pixels measured for

[1] A coarse time-of-flight gating is applied to the photon detections, hindering measurements of an arbitrary subset of histogram bins or linear combination of them. However, other alternatives such as gated cameras [69] can provide such measurements.

each wavelength in a more principled way, we achieve better results than the completely random schemes of Altmann et al. [75]. Moreover, the novel subsampling strategy can be easily implemented in many lidar systems, reducing the total number of measurements, i.e., the time to acquire an MSL frame. Raster-scanning systems using a laser supercontinuum source [72,73] can be easily modified to measure only a subset of pixels per wavelength. More interestingly, the coded apertures can be applied to a single-wavelength lidar array by adding a multispectral filter array [79], simultaneously acquiring a different wavelength at each imaged pixel.

Furthermore, we propose a new method to perform 3D reconstruction from subsampled MSL data, which is capable of finding multiple surfaces per pixel. Although the method draws ideas from ManiPoP, we modify the Bayesian model and estimation strategy of ManiPoP to handle the significantly larger dimensionality of multispectral data. We introduce new RJ-MCMC proposals, which take into account the additional spectral dimension and improve the acceptance ratio, compared to the ones proposed in ManiPoP. Moreover, we propose an empirical Bayes approach to build the prior distribution associated with the background detections, which further improves the convergence of the RJ-MCMC sampler. In contrast to multi-depth methods that require storage of dense volumetric estimates [10,36], the memory requirements of the novel method are minimal (just the 3D points and spectral signatures are stored in memory), enabling the acquisition and processing of very large datasets (dozens of wavelengths and hundreds of pixels). The novel algorithm is referred to as MuSaPoP, as it models MultiSpectrAl lidar signals with POint Processes.

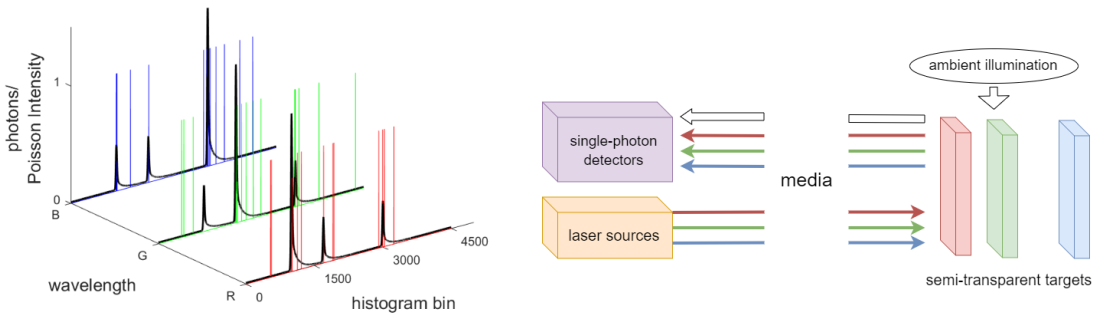


Figure 3.2 Example of an MSL system with 3 different wavelengths (red, green and blue). On the right, a schematic shows the working principles of a single-photon multispectral lidar device. The red, green and blue arrows illustrate the laser pulses sent by the laser sources and reflected by the target onto the single-photon detectors. The white arrow depicts the background photons emitted by an ambient illumination source that reach the detectors at random times. The figure on the left shows the collected histograms for a given pixel: the discrete measurements are depicted in red, green and blue, while the underlying Poisson intensity (i.e., the parameters to estimate from the data) is shown in black.

The remainder of this chapter is organised as follows: Section 3.2 recalls the observation model for single-photon MSL data. Sections 3.3 and 3.4 present the Bayesian 3D reconstruction algorithm and the associated RJ-MCMC sampler. Section 3.5 details the principled subsampling strategy. Experiments performed with synthetic and real lidar data are introduced and discussed

in Section 3.6. Conclusions are finally reported in Section 3.7.

3.2 Single-photon multispectral lidar

A full multispectral lidar data hypercube $\boldsymbol{Z} \in \mathbb{Z}_+^{N_r \times N_c \times L \times T}$ consists of discrete photon count measurements, where L is the number of acquired wavelengths. As in the single-wavelength case, the 3D point cloud is represented by an unordered set of points, that is

$$\boldsymbol{\Phi} = \{(\boldsymbol{c}_n, \boldsymbol{r}_n), n = 1, \ldots, N_\Phi\} \tag{3.1}$$

where $\boldsymbol{r}_n = [r_{n,1}, \ldots, r_{n,L}]^\mathsf{T} \in \mathbb{R}_+^L$ is the spectral response of the nth point. In the multispectral case, the observation model is

$$z_{i,j,\ell,t}|\boldsymbol{\Phi}, b_{i,j,\ell} \sim \mathcal{P}\left(g_{i,j,\ell}\left(\sum_{\mathcal{N}_{i,j}} r_{n,\ell} h_\ell(t - t_n) + b_{i,j,\ell}\right)\right) \tag{3.2}$$

where $b_{i,j,\ell}$ is the background level at wavelength ℓ, $g_{i,j,\ell} \in \{0, 1\}$ is a known binary variable that indicates whether wavelength ℓ at pixel (i, j) has been acquired/observed or not, $\mathcal{N}_{i,j} = \{n : (x_n, y_n) = (i, j)\}$ is the set of points corresponding to pixel (i, j), and $h_\ell(t)$ is the impulse response of the lidar device at wavelength ℓ, which can be measured using a spectralon panel during a calibration step and it is assumed to be fixed (i.e., negligible peak broadening effects). The likelihood of the full hypercube is

$$p(\boldsymbol{Z}|\boldsymbol{\Phi}, \boldsymbol{B}) = \prod_{i=1}^{N_r} \prod_{j=1}^{N_c} \prod_{\ell=1}^{L} \prod_{t=1}^{T} p(z_{i,j,\ell,t}|\boldsymbol{\Phi}, b_{i,j,\ell}). \tag{3.3}$$

The set of all background levels is denoted by $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L] \in \mathbb{R}_+^{N_r \times N_c \times L}$, which is the concatenation of L images $\boldsymbol{b}_\ell$, one for each wavelength. The cube of binary measurements is designated by $\boldsymbol{G} \in \{0, 1\}^{N_r \times N_c \times L}$, where $[\boldsymbol{G}]_{i,j,\ell} = g_{i,j,\ell}$. Note that the model used in the ManiPoP algorithm presented in the previous chapter can be obtained from (3.2) by setting all the binary variables to 1, and considering only one band, i.e., $L = 1$.

3.3 Multiple-return multiple-wavelength 3D reconstruction

The MuSaPoP model shares similar prior distributions with ManiPoP for the point positions and spectral responses, mostly differing in the prior distribution of the background levels. As detailed in Section 3.3.2, an empirical Bayes prior [20] is assigned to the background levels. This section details the main differences from ManiPoP.

3.3.1 Multispectral point cloud

Spatial configuration We adopt the spatial prior distribution of 3D points developed in the ManiPoP algorithm, as detailed in Section 2.2.1.

Reflectivity The spectral signatures are related to the materials of the surfaces [59]. As multispectral devices only acquire dozens of well-separated wavelengths, the spectral measurements within a pixel do not show significant correlation. Hence, although potential correlations between wavelengths could be modelled, we choose here to neglect this correlation to keep the estimation strategy tractable. As a consequence, we consider the following reflectivity prior model

$$p(\boldsymbol{\Phi}_r|\boldsymbol{\Phi}_c, \sigma^2, \beta) = \prod_{\ell=1}^{L} p(\boldsymbol{m}_\ell|\boldsymbol{\Phi}_c, \sigma^2, \beta) \tag{3.4}$$

where the set of spectral marks $\boldsymbol{\Phi}_r$ is separated into per-wavelength log-intensity vectors $\boldsymbol{m}_\ell = [m_{1,\ell}, \ldots, m_{N_{\boldsymbol{\Phi}},\ell}]^\mathsf{T}$ with $m_{n,\ell} = \log r_{n,\ell}$. As in ManiPoP, spatial correlations between log-intensity values (of the same wavelength) in neighbouring pixels are defined according to the distribution

$$\boldsymbol{m}_\ell|\boldsymbol{\Phi}_c,\sigma^2,\beta \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{P}^{-1}) \tag{3.5}$$

where σ^2 and β are hyperparameters controlling the level of smoothness and the precision matrix $\boldsymbol{P}$ is defined by (2.11).

3.3.2 Background levels

Independent prior distributions The background levels are assigned independent prior distributions at each band, in a similar fashion to the prior of the log-intensities. A natural choice would be to use one gamma Markov random field (as explained in Section 2.2.3) per wavelength. However, this prior is not well suited for MSL data as it introduces an undesired penalisation for large background levels, whose negative effects are amplified when considering multiple bands. This can be shown by inspecting the marginal distribution,

$$p(\boldsymbol{B}|\alpha_b) \propto \prod_{\ell=1}^{L}\prod_{i=1}^{N_r}\prod_{j=1}^{N_c} \frac{b_{i,j,\ell}^{\alpha_b-1}}{\left(\sum_{(i',j')\in\mathcal{M}_B} b_{i',j',\ell}\right)^{\alpha_b}} \tag{3.6}$$

where α_b is a hyperparameter controlling the degree of smoothness and $\mathcal{M}_B$ denotes the set of pixels in the neighbourhood of pixel (i, j). For an image of constant intensity c, we have $b_{i',j',\ell} = b_{i,j,\ell} = c$ for all pixels and spectral bands, yielding the density

$$p(\boldsymbol{B}|\alpha_b) \propto \prod_{\ell=1}^{L}\prod_{i=1}^{N_r}\prod_{j=1}^{N_c} c^{-1} = \prod_{i=1}^{N_r}\prod_{j=1}^{N_c} c^{-L}. \tag{3.7}$$

This dependency on the mean promotes reconstructions with lower background levels, decreasing the acceptance ratio of death and erosion moves (that propose to increase the background levels). In the case of ManiPoP, only one band is considered ($L = 1$). Thus, the bias towards smaller background levels does not impact the overall reconstruction significantly. However, in the MSL case ($L \gg 1$), the reconstruction quality is reduced, hindering the use of gamma Markov random fields.

Other alternatives such as Gaussian Markov random fields [55] cannot be sampled directly in closed form, requiring proposals with a rejection step, whose mixing and convergence scale poorly with the dimension of the spectral cube, as detailed in [80].

To alleviate these problems, we assign independent gamma priors, i.e.,

$$\begin{cases} p(\boldsymbol{B}|\boldsymbol{K},\boldsymbol{\Theta}) = \prod_{i=1}^{N_r}\prod_{j=1}^{N_c}\prod_{\ell=1}^{L} p(b_{i,j,\ell}|k_{i,j,\ell},\theta_{i,j,\ell}) \\ b_{i,j,\ell}|k_{i,j,\ell},\theta_{i,j\ell} \sim \mathcal{G}(k_{i,j,\ell},\theta_{i,j\ell}) \end{cases} \tag{3.8}$$

where $[\boldsymbol{\Theta}]_{i,j,\ell} = \theta_{i,j,\ell}$ and $[\boldsymbol{K}]_{i,j,\ell} = k_{i,j,\ell}$ are the shape and scale hyperparameters of the gamma distributions. Despite using independent priors, we can capture the spatial correlation by setting the hyperparameters $(\boldsymbol{K},\boldsymbol{\Theta})$ appropriately. More precisely, in a similar fashion to variational Bayes [23] or expectation propagation [22] methods, in order to simplify the estimation of $\boldsymbol{B}$, we specify (3.8) such that $p(\boldsymbol{B}|\boldsymbol{K},\boldsymbol{\Theta})$ is similar to another distribution $q(\boldsymbol{B}) = \prod_{\ell=1}^{L} q_\ell(\boldsymbol{b}_\ell)$ which explicitly correlates the background levels in neighbouring pixels and assumes mutual independence between spectral bands. Here, we use as a similarity criterion the Kullback-Leibler divergence

$$(\boldsymbol{K},\boldsymbol{\Theta}) = \arg\min_{\boldsymbol{K},\boldsymbol{\Theta}} \mathrm{KL}(q||p). \tag{3.9}$$

As discussed in the next subsection, solving (3.9) can be achieved by computing expectations with respect to $q(\boldsymbol{B})$.

Empirical Bayes approach To ensure that the prior model (3.8) is informative, a suitable distribution $q(\boldsymbol{B})$ should be chosen. Assuming that we have a coarse estimate of the point cloud (this information will be obtained using the multiresolution approach detailed in Section 3.4.2), we build the distribution $q(\boldsymbol{B})$ following an empirical Bayes approach, as illustrated in Fig. 3.3. First, one can discard almost all the signal photons in the dataset by removing the photons detected in the compact support of $h_\ell(t)$ around each point (see Fig. 3.3, central subplot). The number of bins that are not excluded at each pixel is referred to as $v_{i,j,\ell}$. Secondly, we integrate the remaining photons of each pixel, obtaining a coarse estimate of the per-pixel background photons, denoted

by $\bar{z}_{i,j,\ell,b}$. We then define $q_\ell(\boldsymbol{b}_\ell) \propto p(\bar{\boldsymbol{z}}_\ell|\boldsymbol{b}_\ell)p(\boldsymbol{b}_\ell|\alpha_b)$ with

$$\begin{cases} \bar{z}_{i,j,\ell,b}|b_{i,j,\ell} \sim \mathcal{P}(g_{i,j,\ell}v_{i,j,\ell}b_{i,j,\ell}) \\ b_{i,j,\ell} = \exp(\tilde{b}_{i,j,\ell} + \mu_\ell) \\ \tilde{\boldsymbol{b}}_\ell|\tilde{\alpha}_b \sim \mathcal{N}(\boldsymbol{0}, \tilde{\alpha}_b\boldsymbol{D}^{-1}) \end{cases} \tag{3.10}$$

where $\boldsymbol{D} \in \mathbb{R}^{N_rN_c \times N_rN_c}$ is a positive semidefinite matrix and $\tilde{\alpha}_b$ is a fixed hyperparameter controlling the degree of smoothness. In monostatic systems, a 2D Laplacian filter is chosen for $\boldsymbol{D}$ to promote spatial correlation [55], whereas in bistatic systems, $\boldsymbol{D}$ is replaced by the identity matrix, only penalising large background levels. The translation parameter μ_ℓ centres $\tilde{b}_{i,j,\ell}$ in the linear part of the exponential function and is defined as $\mu_\ell = \log(\frac{1}{N_rN_c}\sum_i^{N_r}\sum_j^{N_c}\frac{\bar{z}_{i,j,\ell,b}g_{i,j,\ell}}{v_{i,j,\ell}})$.

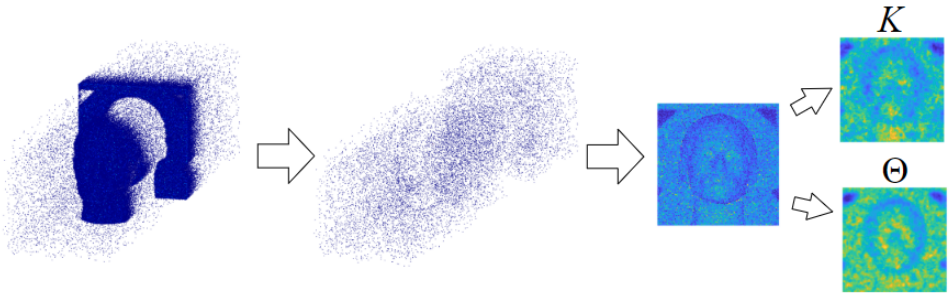


Figure 3.3 Computation of the hyperparameters for the priors of the background levels. First, the photons due to the signal are removed from the dataset using a coarse approximation of the point cloud. Secondly, the remaining photons are integrated per pixel, giving a noisy background image. Finally, this image is used to estimate uncertainty about the background levels, computing the hyperparameters $\boldsymbol{K}$ and $\boldsymbol{\Theta}$.

The prior for the background levels is chosen to minimise the Kullback-Leibler divergence in (3.9), where the correlated model q is given by (3.10). The minimisation of (3.9) can be written as

$$(\boldsymbol{K}, \boldsymbol{\Theta}) = \underset{(\boldsymbol{K},\boldsymbol{\Theta})}{\arg\min} -\mathbb{E}_q\{\log p(\boldsymbol{B}|\boldsymbol{K}, \boldsymbol{\Theta})\}. \tag{3.11}$$

Considering the product of independent gamma distributions in (3.8), the problem reduces to

$$(k_{i,j,\ell}, \theta_{i,j,\ell}) = \underset{(k_{i,j,\ell},\theta_{i,j,\ell})}{\arg\min} \frac{\mathbb{E}_q\{b_{i,j,\ell}\}}{\theta_{i,j,\ell}} - k_{i,j,\ell}\left(\mathbb{E}_q\{\log b_{i,j,\ell}\} - \log\theta_{i,j,\ell}\right) + \log\Gamma(k_{i,j,\ell}) \tag{3.12}$$

for all pixels $i = 1, \ldots, N_r$ and $j = 1, \ldots, N_c$ and wavelengths $\ell = 1, \ldots, L$. The expected values $\mathbb{E}_q\{b_{i,j,\ell}\}$ and $\mathbb{E}_q\{\log b_{i,j,\ell}\}$ cannot be obtained in closed form for the Poisson-Gaussian model of (3.10). Thus, we approximate them numerically by obtaining MCMC samples of $\tilde{b}_{i,j,\ell}$. As explained in [80], off-the-shelf sampling strategies (e.g., Hamiltonian Monte Carlo [21, Chapter 9]) do not scale well with the dimension of the problem, being inefficient when applied to large MSL datasets. Hence, we consider proposals from a Gaussian approximation of (3.10) (as detailed in [55]) using the perturbation optimisation algorithm [81], accepting or rejecting them according to the

Metropolis-Hastings rule [21,55]. We generate 10^3 samples $\{\tilde{b}_{i,j,\ell}^{(s)}, s = 1, \ldots, 10^3\}$ and compute the desired expected values as

$$\mathbb{E}_q\{b_{i,j,\ell}\} = \sum_s \exp \tilde{b}_{i,j,\ell}^{(s)} \tag{3.13}$$

$$\mathbb{E}_q\{\log b_{i,j,\ell}\} = \sum_s \tilde{b}_{i,j,\ell}^{(s)}. \tag{3.14}$$

Finally, the values of the hyperparameters are obtained by setting $\theta_{i,j,\ell} = \mathbb{E}_q\{b_{i,j,\ell}\}$ and minimising (3.12) with a one-dimensional Newton method. Note that (3.10) simply involves integrated photon counts (over the range dimension). Moreover, given the independence property of $q(\boldsymbol{B})$ among spectral bands, all the bands can be processed independently in parallel when sampling $\boldsymbol{B}$.

3.3.3 Posterior distribution

Following Bayes theorem, the joint posterior distribution of the model parameters is given by

$$p(\boldsymbol{\Phi}_c, \boldsymbol{\Phi}_r, \boldsymbol{B}|\boldsymbol{Z}, \boldsymbol{\Psi}) \propto p(\boldsymbol{Z}|\boldsymbol{\Phi}_c, \boldsymbol{\Phi}_r, \boldsymbol{B})\pi(\boldsymbol{\Phi}_c)p(\boldsymbol{\Phi}_r|\boldsymbol{\Phi}_c, \sigma^2, \beta)f_1(\boldsymbol{\Phi}_c|d_{\min})f_2(\boldsymbol{\Phi}_c|\gamma_a, \lambda_a)p(\boldsymbol{B}|\boldsymbol{K}, \boldsymbol{\Theta}) \tag{3.15}$$

where the set of hyperparameters is $\boldsymbol{\Psi} = \{d_{\min}, \gamma_a, \lambda_a, \sigma^2, \beta, \boldsymbol{K}, \boldsymbol{\Theta}\}$, the likelihood of the observed data has been defined in (3.2) and (3.3), the Poisson reference measure is defined by (2.2), and the other densities are priors defined in (2.5), (2.6), (3.5) and (3.8). Figure 3.4 shows the directed acyclic graph associated with the hierarchical Bayesian model.

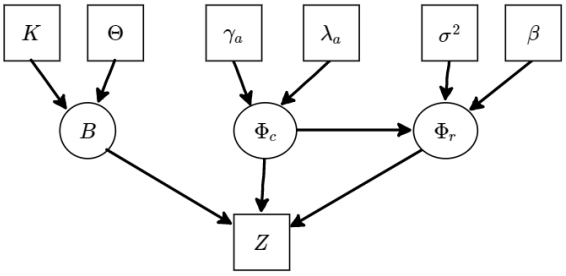


Figure 3.4 Directed acyclic graph (DAG) of the hierarchical Bayesian model. The variables inside squares are fixed, whereas the variables inside circles are estimated.

3.4 Inference

We compute the same posterior statistics as in ManiPoP: the point cloud positions and spectral signatures are estimated using the maximum-a-posteriori (MAP) estimator

$$\hat{\boldsymbol{\Phi}} = \underset{\boldsymbol{\Phi}, \boldsymbol{B}}{\arg\max}\, p(\boldsymbol{\Phi}, \boldsymbol{B}|\boldsymbol{Z}, \boldsymbol{\Psi}) \tag{3.16}$$

and the minimum mean squared error estimator is considered for $\boldsymbol{B}$

$$\hat{\boldsymbol{B}} = \mathbb{E}\{\boldsymbol{B}|\boldsymbol{Z}, \boldsymbol{\Psi}\}. \tag{3.17}$$

As in ManiPoP, we use a reversible jump MCMC algorithm that generates N_i samples of $\boldsymbol{\Phi}$ and $\boldsymbol{B}$ from the posterior distribution (3.15) denoted as

$$\{\boldsymbol{\Phi}^{(s)}, \boldsymbol{B}^{(s)} \quad \forall s = 0, 1, \ldots, N_i - 1\}. \tag{3.18}$$

These samples are then used to approximate the statistics of interest, i.e.,

$$\hat{s} \approx \underset{s=0,\ldots,N_i-1}{\arg\max}\, p(\boldsymbol{\Phi}^{(s)}, \boldsymbol{B}^{(s)}|\boldsymbol{Z}, \boldsymbol{\Psi}) \tag{3.19}$$

with $\hat{\boldsymbol{\Phi}} \approx \boldsymbol{\Phi}^{(\hat{s})}$, and

$$\hat{\boldsymbol{B}} \approx \frac{1}{N_i - N_{\text{bi}}} \sum_{s=N_{\text{bi}}+1}^{N_i} \boldsymbol{B}^{(s)}. \tag{3.20}$$

3.4.1 Reversible jump MCMC moves

For ease of presentation, we summarise the main aspects of each move, inviting the reader to consult Appendix C for the full expressions of the acceptance ratios.

Birth and death moves The birth move proposes a new point $\boldsymbol{\theta'} = (\boldsymbol{c}_{N_\Phi+1}, \boldsymbol{m}_{N_\Phi+1})$ uniformly at random in the 3D cube. The spectral signature of the new point is proposed by extracting a fraction $(1 - u_\ell)$ from the current value of the background level $b_{i,j,\ell}$ according to the SBR w_ℓ, that is for each wavelength ℓ

$$\begin{cases} u_\ell \sim \mathcal{U}(0,1), b'_{i,j,\ell} = u_\ell b_{i,j,\ell} \\ e^{m_{N_\Phi+1,\ell}} = w_\ell(1-u_\ell) b_{i,j,\ell} \dfrac{T}{\sum_{t=1}^{T} h_\ell(t)} \end{cases}, \tag{3.21}$$

where $\mathcal{U}(0,1)$ denotes the uniform distribution on the interval $(0,1)$. The death move proposes the removal of a point. In contrast to the birth move, we modify the background level according to

$$b'_{i,j,\ell} = b_{i,j,\ell} + e^{m_{N_\Phi+1,\ell}} \frac{\sum_{t=1}^{T} h_\ell(t)}{w_\ell T} \quad \forall \ell = 1, \ldots, L. \tag{3.22}$$

Dilation and erosion moves Birth moves have low acceptance ratio, as the probability of randomly proposing a point within or close to the surfaces of interest is very low. However, this problem can be overcome by using the current estimation of the surface to propose in regions of high probability. The dilation move proposes a point inside the neighbourhood of an existing surface with uniform probability across all possible neighbouring positions where a point can be

added. In contrast to ManiPoP, where the intensity samples are generated according to the prior distribution, the spectral signature is sampled in the same way as the birth move (3.21). The complementary move, erosion, proposes to remove a point $\boldsymbol{c}_n$ with one or more neighbours. In this case, the background is updated in the same way as the death move.

Mark and shift moves As in ManiPoP, the mark move updates the log-intensity of a randomly chosen point $\boldsymbol{c}_n$. Each wavelength is updated independently using a Gaussian proposal with variance δ_m as a proposal (also known as Metropolis Gaussian random walk), that is

$$m'_{n,\ell} \sim \mathcal{N}\left(m_{n,\ell}, \delta_m\right) \quad \forall \ell = 1, \ldots, L. \tag{3.23}$$

Similarly, the shift move updates the position of a uniformly chosen point using a Gaussian proposal with variance δ_t

$$t'_n \sim \mathcal{N}\left(t_n, \delta_t\right) \tag{3.24}$$

The values of δ_m and δ_t are adjusted by cross-validation[2] to yield an acceptance ratio close to 41% for each move, which is the optimal value for a one dimensional Metropolis random walk, as explained in [21, Chapter 4].

Split and merge moves Some pixels might present two points with overlapping impulse responses in depth. In such cases, a death move followed by two birth moves would happen with very low probability. Hence, as in ManiPoP, we propose a split move, which randomly picks a point $(\boldsymbol{c}_n, \boldsymbol{m}_n)$ and proposes two new points, $(\boldsymbol{c}'_{k_1}, \boldsymbol{m}'_{k_1})$ and $(\boldsymbol{c}'_{k_2}, \boldsymbol{m}'_{k_2})$. The log-intensity is proposed for each wavelength following the mapping

$$\begin{cases} u_\ell \sim \mathcal{B}(\eta, \eta) \\ m'_{k_1,\ell} = m_{n,\ell} + \log(u_\ell) \\ m'_{k_2,\ell} = m_{n,\ell} + \log(1 - u_\ell) \end{cases} \tag{3.25}$$

where $\mathcal{B}(\cdot)$ denotes the beta distribution and η is a fixed parameter. The new positions are determined according to

$$\begin{cases} s_\ell \sim \mathcal{B}\mathrm{e}(0.5) \\ \Delta \sim \mathcal{U}(d_{\min}, \mathrm{length}_{h(t)}) \\ t'_{k_1} = t_n + (-1)^{s_\ell} \Delta \dfrac{\sum_{\ell=1}^{L} (1 - u_\ell) e^{m_{n,\ell}}}{\sum_{\ell=1}^{L} e^{m_{n,\ell}}} \\ t'_{k_2} = t_n - (-1)^{s_\ell} \Delta \dfrac{\sum_{\ell=1}^{L} u_\ell e^{m_{n,\ell}}}{\sum_{\ell=1}^{L} e^{m_{n,\ell}}} \end{cases} \tag{3.26}$$

[2]Intensities are normalised to belong to a fixed interval across datasets. Hence, we can fix the variance of the proposal to achieve similar acceptance ratios.

where $\text{length}_{h(t)}$ is defined as the length in bins of the instrumental response at the wavelength with longest impulse response. The complementary move, merge, is performed by randomly choosing two points $\boldsymbol{c}_{k_1}$ and $\boldsymbol{c}_{k_2}$ inside the same pixel ($x_{k_1} = x_{k_2}$ and $y_{k_1} = y_{k_2}$) that satisfy the condition

$$d_{\min} < |t_{k_1} - t_{k_2}| \leq \text{length}_{h(t)} \quad \forall \ell = 1, \ldots, L \tag{3.27}$$

The merged point $(\boldsymbol{c'}_n, \boldsymbol{m}'_n)$ is obtained by the inverse mapping

$$\begin{cases} e^{m'_{n,\ell}} = e^{m_{k_1,\ell}} + e^{m_{k_2,\ell}} \quad \forall \ell = 1, \ldots, L \\ t'_n = t_{k_1} \dfrac{\sum_{\ell=1}^{L} e^{m_{k_1,\ell}}}{\sum_{\ell=1}^{L} e^{m_{k_1,\ell}} + e^{m_{k_2,\ell}}} + t_{k_2} \dfrac{\sum_{\ell=1}^{L} e^{m_{k_2,\ell}}}{\sum_{\ell=1}^{L} e^{m_{k_1,\ell}} + e^{m_{k_2,\ell}}} \end{cases}$$

which preserves the total pixel intensity and weights the spatial shift of each peak according to the sum of the intensity values.

Sampling the background In order to exploit the conjugacy between the Poisson likelihood and gamma priors for the background levels, we use a similar data augmentation scheme as in ManiPoP, which classifies each photon-detection according to the source (target or background), i.e.,

$$\begin{aligned} z_{i,j,t,\ell} &= \sum_{n:(x_n,y_n)=(i,j)} \tilde{z}_{i,j,t,\ell,n} + \tilde{z}_{i,j,t,\ell,b} \\ \tilde{z}_{i,j,t,\ell,b} &\sim \mathcal{P}(g_{i,j,\ell} b_{i,j,\ell}) \\ \tilde{z}_{i,j,t,\ell,n} &\sim \mathcal{P}(g_{i,j,\ell} r_{n,\ell} h(t - t_n)) \end{aligned}$$

where $\tilde{z}_{i,j,t,\ell,n}$ are the photons at bin t associated with the nth surface and $\tilde{z}_{i,j,t,\ell,b}$ are the ones associated with the background. In the augmented space defined by $(\tilde{z}_{i,j,t,\ell,n}, \tilde{z}_{i,j,t,\ell,b})$, the background levels are sampled as follows

$$\begin{cases} \tilde{z}_{i,j,t,\ell,b} \sim \mathcal{B}\left(z_{i,j,l,t}, \dfrac{b_{i,j,\ell}}{\sum_{n:(x_n,y_n)=(i,j)} r_{n,\ell} h(t - t_n) + b_{i,j,\ell}} \right) \\ b_{i,j,\ell} \sim \mathcal{G}\left(r_{i,j,\ell} + \sum_{t=1}^{T} \tilde{z}_{i,j,t,\ell,b}, \dfrac{\theta_{i,j,\ell}}{g_{i,j,\ell} T \theta_{i,j,\ell} + 1} \right) \end{cases} \tag{3.28}$$

where $\mathcal{B}(\cdot)$ denotes the Binomial distribution. The transition kernel defined by (3.28) produces samples of $b_{i,j,\ell}$ distributed according to the marginal posterior distribution of $\boldsymbol{B}$. In practice, we observed that only one iteration of this kernel is sufficient.

3.4.2 Full algorithm

We adopt a multi-resolution approach to speed up the convergence of the RJ-MCMC algorithm, in a fashion similar to ManiPoP. The dataset is downsampled by integrating photon detections

in patches of $N_p \times N_p$ pixels. The estimated point cloud at the coarse scale is upsampled using a simple nearest neighbour algorithm and used as initialisation for the next (finer) scale. In all our experiments we repeat the process for $S = 3$ scales with a window of $N_p = 3$ pixels. The background hyperpriors $\boldsymbol{K}$ and $\boldsymbol{\Theta}$ are initialised with non-informative values, i.e., $k_{i,j,\ell} = 0.01$ and $\theta_{i,j,\ell} = 100$ for all pixels (i, j) and wavelengths ℓ. In finer scales, these hyperparameters are computed using the algorithm detailed in Section 3.3.2. The multi-resolution approach is finally summarised in Algorithm 3.

Algorithm 3 Multiresolution MuSaPoP

Input: MSL waveforms $\boldsymbol{Z}$, hyperparameters $\boldsymbol{\Psi}$, window size N_p and number of scales S.
Initialisation:
$\boldsymbol{\Phi}_1^{(0)} \leftarrow \emptyset$
$\boldsymbol{B}_1^{(0)} \leftarrow$ sample from (3.28)
$(\boldsymbol{K}_1, \boldsymbol{\Theta}_1) \leftarrow$ non-informative hyperparameter values
Main loop:
for $k = 1, \ldots, S$ **do**
  if $k > 1$ **then**
    $(\boldsymbol{\Phi}_k^{(0)}, \boldsymbol{B}_k^{(0)}) \leftarrow \text{upsample}(\hat{\boldsymbol{\Phi}}_{k-1}, \hat{\boldsymbol{B}}_{k-1})$
    Compute hyperparameters $(\boldsymbol{K}_k, \boldsymbol{\Theta}_k)$ and SBR using Section 3.3.2
  end if
  $(\hat{\boldsymbol{\Phi}}_k, \hat{\boldsymbol{B}}_k) \leftarrow \text{MuSaPoP}(\boldsymbol{Z}_k, (\boldsymbol{\Phi}_k^{(0)}, \boldsymbol{B}_k^{(0)}), \boldsymbol{\Psi}_k, \text{SBR})$
end for
Output: $(\hat{\boldsymbol{\Phi}}_S, \hat{\boldsymbol{B}}_S)$

3.5 Subsampling strategy

Despite not being able to design compressive measurements along the depth axis, we can still reduce the number of measurements in the two spatial (horizontal and vertical) dimensions and in the spectral dimension [75]. Given the point positions, recovering their reflectivity profile reduces to a multispectral image restoration problem using measured data corrupted by Poisson noise. While many compressive sensing strategies have been proposed for measurements under this noise assumption [82–84], MSL datasets have an additional limitation if multiple surfaces per pixel are considered: photon-detections belonging to different wavelengths cannot be integrated into a single histogram, as the reconstruction problem generally becomes highly non-convex, preventing practical reconstruction algorithms[3]. Indeed, summing histograms associated with different wavelengths and including multiple peaks generates histograms with even more peaks (possibly overlapping), which makes the 3D reconstruction and the reflectivity estimation more difficult. For example, a histogram with 2 peaks could be due to a 2 objects of a single wavelength or a single object having two spectral bands. As a consequence, we only consider subsampling of depth histograms, which incorporates all of the practical sampling limitations. Following the formulation of the observation

[3] As mentioned in the introduction, the system presented in [69] considers the integration of photons belonging to different histograms, but is limited to one surface per pixel.

model (3.3), the subsampling strategy consists of choosing the binary coefficients $\boldsymbol{G}$ for a given compression level W/L, with W the average number of observed band per pixel. Several subsampling strategies have been proposed for different applications, such as halftoning and stippling [85], rendering [86], compressive spectral imaging [87–89], compressive computed tomography [90, 91], geometry processing [92], among others [93, 94]. These approaches exploit the sampling geometry of the sensing devices to design a set of criteria. Similar to coded aperture snapshot spectral imaging systems, the distribution of reflectivity profiles of 3D surfaces in real natural scenes suggests uniform sampling in the row, column and spectral axes. Following the design in Correa et al. [78], the coefficients are chosen according to the spatiotemporal characteristics of blue noise, which distributes the measurements in spectral and spatial dimensions as homogeneously as possible [95]. The binary cube $\boldsymbol{G}$ is obtained by minimising the variance of (weighted) measurements per local neighbourhood, i.e.,

$$\arg\min_{\boldsymbol{G}} \sum_{i=1}^{N_r}\sum_{j=1}^{N_c}\sum_{\ell=1}^{L} \sum_{(i',j')\in\mathcal{M}_{ss}^{(i,j)}} g_{i',j',\ell}\theta_{i-i',j-j'} \tag{3.29}$$

$$\text{subject to} \quad \sum_{\ell=1}^{L} g_{i,j,\ell} = W \quad \forall(i,j) \tag{3.30}$$

where $\mathcal{M}_{ss}^{(i,j)}$ denotes the set of pixels in a local neighbourhood of (i,j) and $\theta_{i,j}$ are the weights. In our experiments, we set $\mathcal{M}_{ss}^{(i,j)}$ to be a patch of 9×9 pixels with centre (i,j), and weights

$$\theta_{i,j} = \begin{cases} 0.4 & \text{if } |i| \leq 4 \text{ and } j = 0 \\ 0.4 & \text{if } i = 0 \text{ and } |j| \leq 4 \\ 0.2 & \text{if } 0 < |i| \leq 4 \text{ and } 0 < |j| \leq 4 \\ 0 & \text{otherwise} \end{cases} \tag{3.31}$$

The minimisation is simplified by dividing the data in slices of L/W contiguous bands and running the algorithm introduced in [78] per slice. As shown in Fig. 3.5, the blue noise strategy distributes the measurements uniformly in space, while other random strategies [75] tend to exhibit clusters, leaving some regions without measurements.

3.6 Experiments

To illustrate the efficacy of MuSaPoP, the new reconstruction algorithm is compared to other alternatives (based on the work conducted in [72]) using a synthetic dataset. Subsequently, the new subsampling scheme is compared with other random subsampling choices for a real MSL dataset. In all the experiments, the performance was measured using the following summary statistics:

- True detections $F_{\text{true}}(\tau)$ and false detections $F_{\text{false}}(\tau)$.

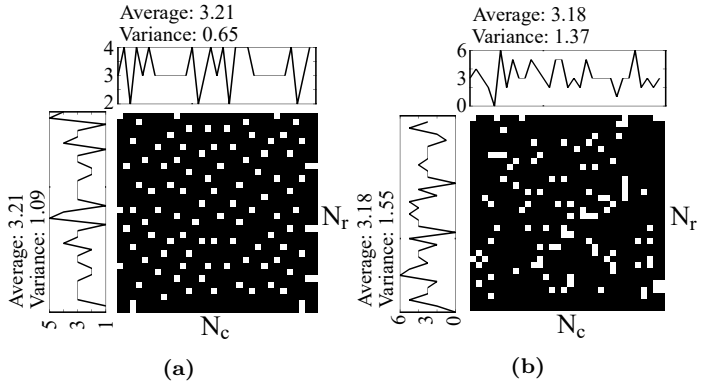


Figure 3.5 Subsampling strategies for a lidar cube with $L = 8$ wavelengths, $N_r = N_c = 32$ pixels and total compression of $1/L$, i.e., one observed band per pixel. The sampled pixels at the first wavelength are shown in white. A completely random strategy [75] is shown in (b), whereas the blue noise is shown in (a).

Hyperparam.	γ_a	λ_a	N_b	$d_{\min}$	σ^2	β	α_B
Scale k	e^2	$(N_r N_c / N_p^{2(k-1)})^{1.5}$	$3\Delta_p/\Delta_b$	$2N_b + 1$	$0.6^2/N_p^{2(k-1)}$	$\sigma^2/100$	2

Table 3.1 Hyperparameter values.

- Mean intensity absolute error at distance τ (IAE): Mean across all the points of the intensity absolute error $\sum_{\ell=1}^{L} |r_{n,\ell}^{\text{true}} - \hat{r}_{n',\ell}|$, normalised with respect to the total number of ground truth points. The ground truth and estimated points are coupled using the probability of detection $F_{\text{true}}(\tau)$. Note that if a point was falsely estimated or a ground truth point was not found, then they are considered to have resulted in an error of $\sum_{\ell=1}^{L} |r_{n,\ell}|$. The comparison is done with normalised intensity values, that is $\sum_{t=1}^{T} h_\ell(t) = 1$ for $\ell = 1, \ldots, L$.

- Background normalised mean squared error $\text{NMSE}_{\boldsymbol{B}}$: Mean of the normalised squared error of the estimated background at each wavelength, i.e.,

$$\frac{1}{L}\sum_{\ell=1}^{L} \frac{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}(b_{i,j,\ell}^{\text{true}} - \hat{b}_{i,j,\ell})^2}{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}\left(b_{i,j,\ell}^{\text{true}}\right)^2}. \tag{3.32}$$

3.6.1 Synthetic data

We first assessed the performance of MuSaPoP using a synthetic dataset created from the "Art" scene of the Middlebury dataset [66], as shown in Fig. 3.6. The measurements were obtained by simulating the single-photon multispectral lidar system of [59], whose bin width Δ_b is 0.3 mm. The generated dataset has $N_r = 283$ and $N_c = 231$ pixels, $T = 4500$ histogram bins and $L = 4$ wavelengths (red, green, blue and yellow), where only $W = 2$ wavelengths out of 4 were sampled per pixel using the coded aperture introduced in Section 3.5. The dataset has 10 PPP, where approximately 3.4 photons are due to the background illumination. As in monostatic lidar systems, the background levels are generated as a passive image of the scene (see Fig. 3.8).

Figure 3.6 Synthetic "Art" scene from Middlebury dataset with an additional semitransparent surface (dark blue plane).

We compared MuSaPoP to a two-stage algorithm that first estimates the point positions using ManiPoP by integrating the photons across wavelengths and then infers the spectral signatures with fixed point positions, similarly to the procedure suggested by Wallace et al. [72]. The resulting method, referred to as ManiPoP #1, is summarised in Algorithm 4. We also compared with ManiPoP in the strict single-wavelength setting, by choosing the most powerful wavelength and using the same total acquisition time per pixel than in the multispectral case (i.e., a per-pixel acquisition time $W = 2$ longer than the one considered for each wavelength in MuSaPoP). This second alternative is referred to as ManiPoP #2.

Algorithm 4 ManiPoP #1 [72]

Input: MSL waveforms $\boldsymbol{Z}$
Depth estimation: Accumulate photons across wavelengths $\bar{z}_{i,j,t} = \sum_\ell z_{i,j,\ell,t}$ for all pixels (i, j)
$(\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{B}}) \leftarrow$ManiPoP$(\bar{\boldsymbol{Z}})$
for $\ell = 1, \ldots, L$ and $\ell \neq w$ **do**
  Update $(\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{B}})$ using ManiPoP$(\boldsymbol{Z}_\ell)$ in a fixed dimensional setting (only using background and reflectivity moves)
end for

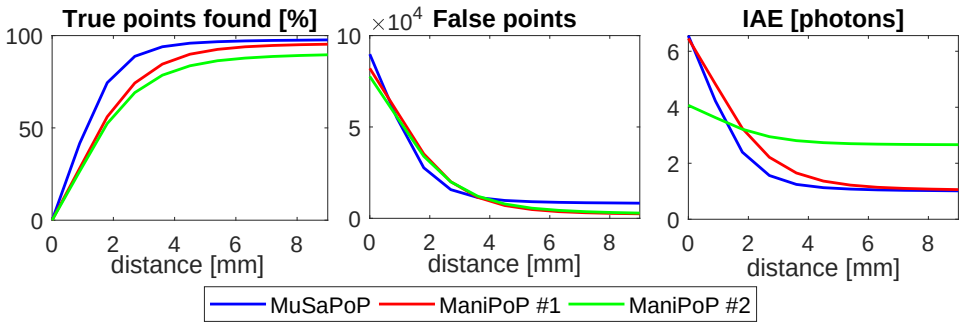


Figure 3.7 From left to right: $F_{\text{true}}(\tau)$, $F_{\text{false}}(\tau)$ and IAE(τ) for MuSaPoP and the two ManiPoP alternatives.

Figure 3.7 illustrates $F_{\text{true}}(\tau)$, $F_{\text{false}}(\tau)$ and IAE for both methods. MuSaPoP performs better than the other alternatives, as it finds 97.7% of the true points, whereas ManiPoP #1 and #2 only recover 95.34% and 89.6% respectively. ManiPoP #1 relies on an approximate impulse response

$\tilde{h}(t) = \sum_{\ell=1}^{L} h_\ell(t)$, which biases the depth estimates, as the true accumulated response varies across points depending on their spectral signature[4]. The bias degrades the performance in terms of average depth absolute error (computed for true detections within a distance of 9 mm from the ground truth point). MuSaPoP obtains an average error of 3.9 mm, whereas the estimates by ManiPoP #1 present an average error of 5.7 mm. Despite having double acquisition time for the single-wavelength, ManiPoP #2 fails to find points corresponding to materials that have very low reflectivity in the blue wavelength (e.g., the red helicoidal structure shown in Fig. 3.6). The MuSaPoP algorithm performs slightly worse in terms of false detections, finding 3 times more false points than the competitor methods. However, the false detections only represent 5% of the total number of ground truth points. In terms of intensity estimation, MuSaPoP obtains better results, having an asymptotic IAE of 1 photon, whereas the alternatives #1 and #2 provide IAE equal to 1.1 and 2.7. The estimated background levels are shown in Fig. 3.8. MuSaPoP yields a better background NMSE (0.04) than alternatives #1 (0.14) and #2 (0.79). The improvement in background estimation over the ManiPoP alternatives can be attributed to the use of an empirical Bayes prior instead of a gamma Markov random field. The total execution time was 811 s for MuSaPoP and 294 and 348 s for alternatives #1 and #2.

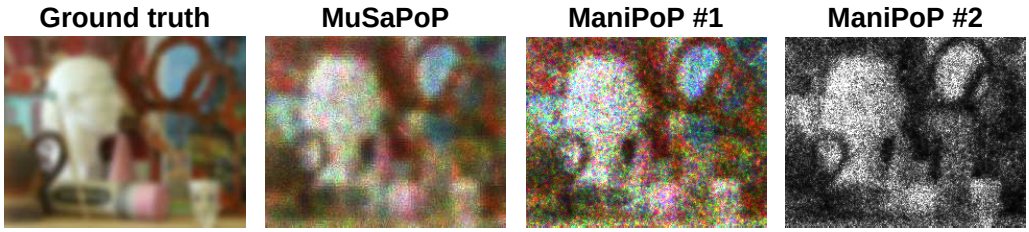


Figure 3.8 From left to right: Ground truth background levels, estimates obtained by MuSaPoP and the two ManiPoP alternatives. Only the red, blue and green channels were used to generate these images. MuSaPoP provides smooth estimates due to the empirical prior distribution described in Section 3.3.2. ManiPoP #2 only estimates one wavelength, which is shown in grayscale.

3.6.2 Real MSL data

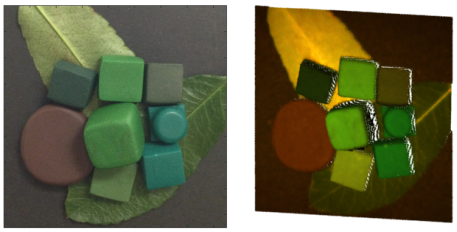
The blue noise subsampling scheme was evaluated on a real MSL dataset [59]. The scene consists of $L = 32$ wavelengths sampled at regular intervals of 10 nm from 500 nm to 810 nm, $N_r = N_c = 198$ pixels and $T = 4500$ histogram bins. The target is composed of a series of blocks of different types of clay and two leaves. Figure 3.9 shows an RGB image of the scene and the 3D reconstruction using acquisition times up to 10 ms per wavelength per pixel. We compare the blue noise codes mentioned in Section 3.5 with the random schemes introduced in [75], all yielding the same total number of measurements and acquisition time:

1. Random sampling without overlap: W out of L bands per pixel are sampled without replacement (i.e., for a given pixel, each wavelength is measured at most once).

[4]Note that this bias can be arbitrarily large depending on the variations of $h_\ell(t)$ across wavelengths.

2. Random sampling without overlap: For each wavelength, $W/L\%$ of the pixels are sampled without replacement.

3. Blue noise sampling method: For each wavelength, $W/L\%$ of the pixels are chosen following the scheme presented in Section 3.5.



(a) **(b)**

Figure 3.9 (a) is an RGB image of the target and (b) shows the 3D reconstructed scene (the colors were generated according to the CIE 1931 RGB color space).

The codes were evaluated for $W = 1, 2, 4, 8, 16$ bands per pixel and acquisition times of 0.1, 1 and 10 ms per measurement (i.e., the histogram of one wavelength), using as ground-truth the reconstruction obtained with all the measurements and an acquisition time of 10 ms. Figure 3.10 shows the percentage of true detections, IAE and background NMSEs for all codes, acquisition times and numbers of sensed bands per pixel W. All the evaluated compressive strategies yield good results, where a small improvement can be obtained by the use of blue noise codes. In terms of total number of estimated points, the blue noise codes achieve better performance in high compression scenarios $W = 1, 2$ and low acquisition times (0.1 and 1 ms). For example, for an acquisition time of 1 ms, almost all points are reconstructed using blue noise codes, whereas the random codes only yield around 97% of the ground-truth points. The choice of blue noise codes has a stronger impact in terms of IAE, achieving smaller IAE for all acquisition times and number of bands per pixel. Figure 3.11 shows the execution time for acquisition times of 10, 1 and 0.1 ms and different numbers of sensed bands. The RJ-MCMC sampler has a complexity proportional to the number of photon detections in the support of the impulse response around the 3D point being modified, whereas the background update has a complexity proportional to the total number of active histogram bins in the lidar scene. The background extraction step required around 15% of the total execution time, which could be significantly reduced if all the bands were processed in parallel instead of sequentially as it is done in the current implementation. All the experiments were performed using a Matlab R2018a implementation on a i7-3.0 GHz desktop computer (16GB RAM).

Finally, we compared the performance of MuSaPoP with the single-depth multiple-wavelength algorithm by Altmann et al. [75]. The algorithm is referred to as Depth TV and considers total variation regularisations for the background, reflectivity and depth images. Note that this method requires a (global) depth interval where all signal photons are found, which is given manually by

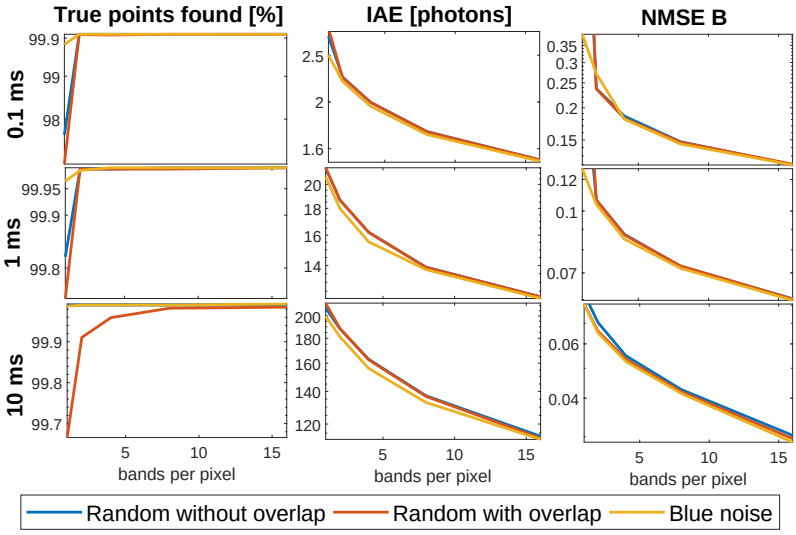


Figure 3.10 $F_{\text{true}}(\tau)$, $F_{\text{false}}(\tau)$ and IAE(τ) obtained with MuSaPoP for different acquisition times and sensed bands per pixel.

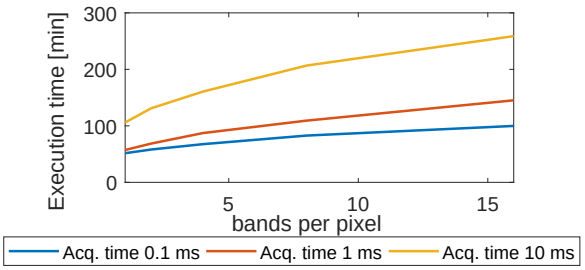


Figure 3.11 Total execution time for different number of sensed bands per pixel and acquisition times of 0.1, 1 and 10 ms.

the user. We also considered a target detection scenario by removing the backplane of the scene and keeping only photons associated with background levels. In this case, we post-process the Depth TV estimates, removing points with a mean normalised intensity below 10%, which gave the best results across the evaluated datasets. In both experiments, we used the blue noise codes with $W = 8$ wavelengths per pixel out of $L = 32$.

Figure 3.12 shows the 3D reconstructions obtained by MuSaPoP and Depth TV using an acquisition time of 10 ms. Figure 3.13 shows the performance of both algorithms in terms of true and false detections. MuSaPoP performs better in the 1 and 0.1 ms cases, whereas Depth TV obtains better depth estimates in the lowest acquisition time case (0.01 ms). However, in the 0.01 ms case without backplane, the intensity thresholding step does not remove backplane points, hence obtaining a very large number of false detections. This result illustrates the inefficiency of simple thresholding in target detection scenarios, whereas MuSaPoP includes these cases within its general formulation. Table 3.2 shows the performance of both algorithms in terms of IAE, background NMSE and execution time. MuSaPoP yields a better IAE than Depth TV (approximately half), as the latter tends to smooth out details within the blocks and leaves, as shown in Fig. 3.12. Moreover,

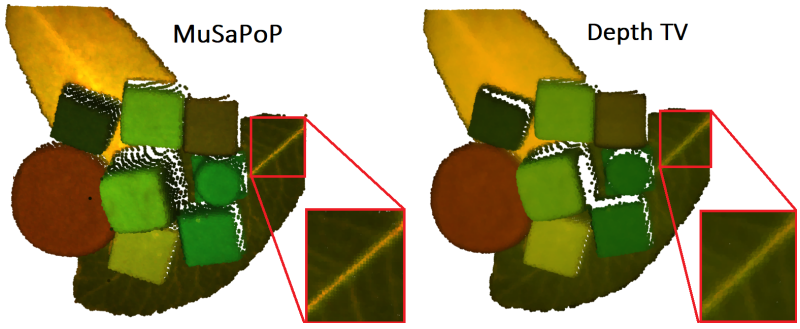


Figure 3.12 From left to right: 3D reconstructions obtained by MuSaPoP and Depth TV for an acquisition time of 10 ms. Note that Depth TV tends to smooth out fine scale details (zoom for better visualisation). Moreover, the thresholding step used in Depth TV removes some low intensity points in the borders of each 3D object.

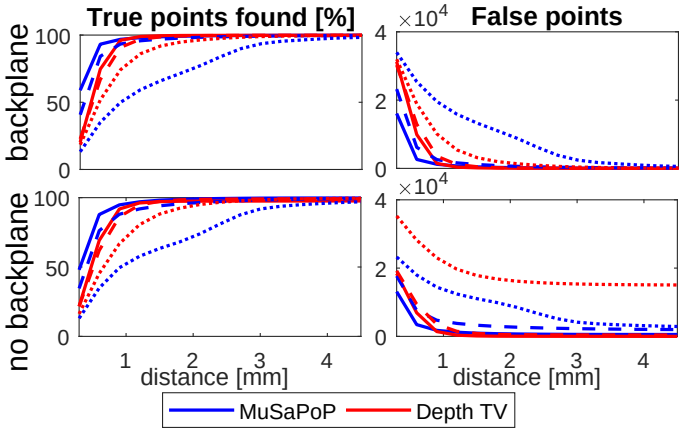


Figure 3.13 True and false point detections for MuSaPoP and Depth TV for the real MSL dataset with (top row) and without (bottom row) backplane. Solid, dashed and dotted lines represent the datasets with acquisition times of 1, 0.1 and 0.01 ms respectively.

in terms of background NMSE, Depth TV fails to provide good estimates in the low-photon cases, as it only considers photon counts within the global interval without signal returns. The execution time of Depth TV was significantly higher than MuSaPoP.

3.7 Conclusions

This chapter has presented a new 3D reconstruction algorithm from multispectral lidar data, which is able to find multiple surfaces per pixel. The novel method leads to better reconstruction quality than other alternatives, as it considers all measured wavelengths in a single observation model. While based on some ideas initially investigated in ManiPoP, MuSaPoP also relies on new strategies to deal with the very high dimensionality of the multispectral problem. The first novelty is the use of an empirical Bayes prior for the background levels, which speeds up significantly the RJ-MCMC algorithm. A second improvement is the adapted dilation/erosion and split/merge moves for the multispectral case, profiting from SBR estimates to increase the acceptance rate. Finally,

Backplane present		Yes			No		
Acq. time [ms]		1	0.1	0.01	1	0.1	0.01
IAE [photons]	Depth TV	36	3.7	0.5	45	4.9	0.8
	MuSaPoP	**14**	**1.7**	**0.3**	**19**	**2.4**	**0.4**
Bkg. NMSE	Depth TV	0.12	>1	>1	0.12	>1	>1
	MuSaPoP	**0.04**	**0.11**	**0.36**	**0.04**	**0.11**	**0.26**
Execution time [h]	Depth TV	19.8	17.9	17.7	19.3	17.9	17.7
	MuSaPoP	**2.8**	**1.4**	**1.0**	**1.5**	**1**	**0.8**

Table 3.2 IAE, background NMSE and execution time of Depth TV and MuSaPoP for the blocks and leaves dataset with and without the backplane.

the subsampling strategy further reduces both the algorithm's complexity and total number of measurements, leading to faster acquisitions and reconstructions. The sparse point cloud representation of MuSaPoP speeds up the computations proportionally to the number of measurements, whereas models based on dense intensity cubes [10, 36] would not achieve similar improvements.

Although MuSaPoP has minimal memory requirements, the execution time is of the order of minutes, hindering some MSL applications. The next chapter presents a target detection method that can discard non-informative histograms without surfaces in few milliseconds, and could be used as a pre-processing step that reduces the size of the data to be processed by MuSaPoP.

Finally, Chapter 5 will present a real-time multi-depth algorithm that can also be extended to handle MSL data, profiting from the subsampling scheme described in this chapter.

Chapter 4

Fast surface detection

Contents

4.1 Introduction

While the multi-depth assumption of ManiPoP and MuSaPoP applies to a very wide range of 3D scenes, a large subset of them contain at most one surface per pixel (e.g., close range imaging). This case also corresponds to many outdoor 3D imaging applications where a target might be present only at a subset of pixels. Hence, limiting the number of surfaces per pixel to 0 or 1 can significantly reduce the complexity of the reconstructions algorithms, while still tackling a wide range of practical imaging scenarios. As discussed in Section 2.5, simply thresholding the reflectivity estimates of a single-depth algorithm generally provides poor results. Moreover,

previous target detection methods [35] relied on an RJ-MCMC sampler with slow convergence, resulting in execution times of the order of hours for a single dataset.

In this chapter, we introduce a novel approach for the target detection problem that overcome these shortcomings. We propose an algorithm adapted to situations where the flux of photons originally emitted by the laser source is small (low PPP) and/or the ambient illumination level is high (low SBR). Following a Bayesian approach, the target detection problem is first formulated as a pixel-wise model selection and estimation problem, where prior distributions are assigned to each of the unknown model parameters. In contrast to the RJ-MCMC target detection algorithm [35], we reformulate the observation model such that the background parameters can be marginalised analytically while the other parameters (target range and reflectivity) can be marginalised from the posterior distribution using (finite sums of) one-dimensional integrals. We also present two post-processing alternatives to further improve the detection maps at a low additional cost. These additional steps can be seen as defining the prior probabilities of target presence (or equivalently the binary labels associated with the presence of targets) to account for the spatial organisation of objects in the scene. The resulting algorithm, which relies mostly on pixel-wise, low-dimensional integrations, is thus suited for real-time applications and can be implemented using parallel architectures.

The remainder of this chapter is organised as follows: Section 4.2 recalls the observation model under the target detection assumption and introduces an alternative model used in this chapter. Section 4.3 details the proposed Bayesian pixel-wise target detection method. Section 4.4 presents two post-processing approaches to incorporate spatial regularisation in the final detection map. Simulation results conducted using synthetic and real lidar measurements are presented and discussed in Section 4.5. Finally, conclusions are reported in Section 4.6.

4.2 Observation model

For ease of presentation, here we denote a single histogram of photon detections by the vector $\boldsymbol{z} = [z_1, \ldots, z_T]^\mathsf{T} \in \mathbb{Z}_+^T$. Under the assumption of at most one object per pixel, the general observation model (1.1) reduces to

$$z_t | r, t_0, b \sim \mathcal{P}\left(rh(t - t_0) + b\right) \quad \forall t = 1, \ldots, T, \tag{4.1}$$

where $r \in [0, \infty)$, being zero in the absence of target. The impulse response is assumed to be normalised ($\sum_{t=1}^{T} h(t) = 1$) in this chapter.

An equivalent model can be defined using the SBR, recalling that is defined as the ratio of the useful detected photons $w = r/(bT)$. Following this alternative parametrisation, the observation

model (4.1) can be rewritten as

$$z_t|w, t_0, b \sim \mathcal{P}\left(b\left(wTh(t-t_0)+1\right)\right) \quad \forall t = 1, \ldots, T. \tag{4.2}$$

The main motivation for using (4.2) instead of (4.1) is that gamma distributions are conjugate priors for b in (4.2) (and not in (4.1)), which allows a simple marginalisation of b, as will be seen in Section 4.3. Assuming independence between the different observations z_t conditionally to w, t_0 and b, the joint likelihood for a single histogram is expressed as

$$p(\boldsymbol{z}|w, t_0, b) = \prod_{t=1}^{T} p(z_t|w, t_0, b). \tag{4.3}$$

As can be seen from the two observation models (4.1) and (4.2), in the absence of surface in the field of view, i.e., when $r = 0$ or equivalently when $w = 0$, the observation model reduces to considering T random variables z_t drawn independently from a Poisson distribution with mean b, i.e.,

$$z_t|w = 0, t_0, b \sim \mathcal{P}\left(b\right). \tag{4.4}$$

This chapter presents a surface detection algorithm to decide whether $w = 0$ or $w > 0$. Note that the background level b and depth t_0 (if an object is present) are unknown in practice, which makes the detection task more difficult. The next section presents the proposed Bayesian strategy for this detection problem.

4.3 Detection strategy

We begin by introducing the prior distributions for a pixel-wise model, where every histogram is processed separately. Spatial regularisation techniques are discussed in the following section.

4.3.1 Prior distributions

Similarly to ManiPoP, MuSaPoP and other previous work [16, 35, 76], independent prior distributions are assigned to the background level and target reflectivity, i.e., $p(r, b) = p(r)p(b)$. In order to model the absence ($r = 0$) or presence ($r > 0$) of a target, we use a spike and slab prior distribution [96] for the signal intensity, that is

$$p(r|u, \alpha_r, \beta_r) = u\mathcal{G}(r; \alpha_r, \beta_r) + (1-u)\delta(r) \tag{4.5}$$

where $\delta(r)$ is the Dirac delta distribution centred at 0 and $u \in \{0, 1\}$ is a binary variable that indicates the presence ($u = 1$) or absence ($u = 0$) of a target. Moreover, $\mathcal{G}(r; \alpha_r, \beta_r)$ denotes a gamma density with known shape parameter α_r and rate parameter β_r. Note that (α_r, β_r) can

usually be adjusted from calibration measurements, as the dynamic range of r is primarily guided by the laser power used, the average distance between the lidar system and the scene, the scattering properties of the media, the efficiency of the detector and the pixel-wise acquisition time. The prior distribution for the binary label u is a Bernoulli distribution such that $P(u = 1) = P(u = 0) = 0.5$, expressing our absence of knowledge regarding this parameter.

The background level is modelled as in [16, 35] with a conjugate gamma distribution, that is

$$p(b|\alpha_b, \beta_b) = \mathcal{G}(b; \alpha_b, \beta_b) \tag{4.6}$$

with fixed hyperparameters (α_b, β_b). If limited information is available about the background levels, a weakly informative prior distribution can be defined for b (e.g., a heavy-tailed prior distribution). The resulting joint prior distribution with parametrisation based on b and w can be obtained from $p(r, b)$ by applying a standard change of variables yielding

$$p(w, b|u, \boldsymbol{\Psi}) = (1 - u)\delta(w)\mathcal{G}(b; \alpha_b, \beta_b) + uc_0(w)\mathcal{G}(b; \alpha_b + \alpha_r, \beta_b + \beta_r T w) \tag{4.7}$$

where $\boldsymbol{\Psi} = \{\alpha_r, \beta_r, \alpha_b, \beta_b\}$, $c_0(w) = (T\beta_r)^{\alpha_r} w^{\alpha_r - 1}(\beta_b + Tw\beta_r)^{\alpha_r + \alpha_b}\beta_b^{\alpha_b}/B(\alpha_r, \alpha_b)$ and $B(\cdot, \cdot)$ is the beta function. Since $\boldsymbol{\Psi}$ is fixed, it is omitted in all the conditional distributions in this chapter. Assuming no prior knowledge on the position of the target, we assign a uniform prior for the depth, i.e., $P(t_0) = 1/T$ for any t_0 in $\{1, \ldots, T\}$. However, this choice could be changed if additional information is available.

4.3.2 Decision rule

The proposed decision rule is based on the marginal posterior distribution of the label u, obtained by integrating out the parameters b, t_0 and w, considered here as nuisance parameters. Defining H_0 and H_1 as the absence and presence of the target respectively, the decision rule is

$$P(u = 0|\boldsymbol{z}) \underset{H1}{\overset{H0}{\gtrless}} \nu \tag{4.8}$$

where ν is a scalar threshold and

$$P(u|\boldsymbol{z}) = \sum_{t_0=1}^{T} \int\int p(w, b, t_0, u|\boldsymbol{z}) db dw \tag{4.9}$$

with $p(w, b, t_0, u|\boldsymbol{z}) \propto p(\boldsymbol{z}|w, b, t_0)p(w, b|u)P(t_0)P(u)$ using Bayes rule. Note that, as will be shown in Section 4.5, it is also possible to consider t_0 as a deterministic parameter and only marginalise (b, w), i.e., consider $P(u|\boldsymbol{z}, t_0)$ in (4.8), where the actual (unknown) value of t_0 is replaced by an arbitrary estimate.

4.3.3 Computation of marginals

In order to compute the marginal distribution $P(u|\boldsymbol{z})$ used in (4.8), we first integrate out the background level and target position, that is

$$p(w,u|\boldsymbol{z}) \propto \sum_{t_0=1}^{T} P(t_0) \int_0^\infty p(\boldsymbol{z}|w,t_0,b)p(w,b|u)P(u)db.$$

Due to the conjugacy between the observation model (4.2) and the prior distribution (4.7), the inner integral is available in closed form. The integration over the SBR is also available in closed form for $u=0$,

$$\begin{aligned} P(u=0|\boldsymbol{z}) &= \int p(w,u=0|\boldsymbol{z})dw \\ &= \frac{(1-\pi)}{\gamma}\Gamma(\bar{\boldsymbol{z}}+\alpha_b)(T+\beta_b)^{\bar{z}+\alpha_b} \end{aligned} \tag{4.10}$$

where $\bar{z} = \sum_{t=1}^{T} z_t$ is the total number of photons observed and γ is a normalisation constant. Finally, the marginal probability of the target being present is

$$P(u=1|\boldsymbol{z}) = \frac{c_1}{\gamma}\int_0^\infty f_1(w)\sum_{t_0=1}^{T}\exp(f_2(w,t_0))dw \tag{4.11}$$

with

$$\begin{aligned} f_1(w) &= w^{\alpha_r-1}\left(\beta_b+T+wT(\beta_r+1)\right)^{-\bar{z}-\alpha_r-\alpha_b} \\ f_2(w,t_0) &= \sum_{t=1}^{T} z_t \log(wTh(t-t_0)+1), \end{aligned}$$

and where γ is the same constant as in (4.10) and $c_1 = \pi\Gamma(\alpha_r)\Gamma(\bar{z}+\alpha_b+\alpha_r)(\beta_r T)^{\alpha_r}$. Since γ is the same in (4.10) and (4.11), it can be easily computed using $P(u=0|\boldsymbol{z})+P(u=1|\boldsymbol{z})=1$. The marginal distribution (4.11) involves an intractable integral. However, the sum can be computed with $\mathcal{O}(T\log T)$ floating point operations using the FFT, allowing the integral to be numerically approximated with a quadrature method (with a computational cost of R integrand evaluations). Thus, the overall complexity is $\mathcal{O}(RT\log T)$, which is close to cross-correlation if $R \ll T$. Note that if t_0 is not marginalised and replaced by a point estimate instead, (4.11) is simplified as the sum in the integrand reduces to one term.

4.4 Spatial regularisation

As discussed in the previous chapters, histograms corresponding to neighbouring pixels generally present similar numbers of surfaces. Thus, we propose to refine the pixel-wise detection method to

create a more homogeneous map of target presence. Such segmentation can be efficiently computed by solving a TV problem or using a multiscale approach as in ManiPoP. To speed up computations, spatial correlation is promoted using only the scalar marginal posterior probabilities $P(u = 1|\boldsymbol{z})$, such that we can compute them separately in parallel and then deploy a regularisation method working on this reduced set of variables (not the full model).

Remark: In contrast to ManiPoP, these approaches are not fully Bayesian, as the spatial correlation is not included within the posterior distribution, but as a post-processing step of the per-pixel detection probabilities.

4.4.1 Total variation regularisation

Spatial correlation can be achieved by finding the minimum perimeter detection map dividing regions with and without target, as illustrated in Fig. 4.1. This problem is solved by many existing segmentation algorithms (e.g., GrabCut [97]). The minimum-perimeter segmentation reduces to solving the following TV minimisation procedure [24]

$$\hat{\boldsymbol{u}} = f_{\text{th}}\left(\arg\min_{\boldsymbol{v}} ||\boldsymbol{v} - \boldsymbol{y}||_2^2 + \tau||\boldsymbol{v}||_{\text{TV}}\right) \tag{4.12}$$

where the input image $\boldsymbol{y}$ contains the log-ratios $y_{i,j} = \log P(u = 1|\boldsymbol{z}) - \log P(u = 0|\boldsymbol{z})$ of pixel (i, j), τ is a user-defined parameter which controls the amount of spatial correlation ($\tau = 5$ here) and $f_{\text{th}}(\cdot)$ is a hard thresholding operation, which assigns 1 to positive inputs and 0 otherwise. $||\cdot||_{\text{TV}}$ is the isotropic total variation operator, which is defined as the norm $||\boldsymbol{x}||_{\text{TV}} = ||\boldsymbol{D}\boldsymbol{x}||_{2,1}$ where $\boldsymbol{D}$ is the discrete gradient operator, which is defined by

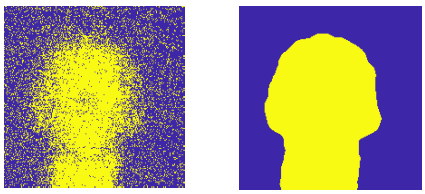
$$[\boldsymbol{D}\boldsymbol{x}]_{i,j,1} = \begin{cases} x_{i+1,j} - x_{i,j} & \text{if } 1 \leq i < N_r \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

$$[\boldsymbol{D}\boldsymbol{x}]_{i,j,2} = \begin{cases} x_{i,j+1} - x_{i,j} & \text{if } 1 \leq j < N_c \\ 0 & \text{otherwise} \end{cases}. \tag{4.14}$$

This algorithm can be easily parallelised, running one parallel thread per lidar pixel. Both the ℓ_2 penalty and TV operator only require the information of a local neighbourhood at each pixel.

4.4.2 Multiscale approach

For a fixed SBR, the detection performance depends on the number of photons collected in the histogram. In a similar fashion to ManiPoP, we integrate histograms in super-pixels (windows of 2×2 pixels), yielding approximately 4 times more photons and a similar SBR. By applying pixel-wise detection on a coarser scale, we can improve the performance, while reducing the number of



(a) (b)

Figure 4.1 Effect of the total variation post-processing. Yellow and blue pixels indicate the presence or absence of target respectively. (a) Detection map of a mannequin head using only pixel-wise detections. (b) Minimum-perimeter detection map. The TV-based post-processing step promotes correlation between neighbouring pixels, improving the detection performance.

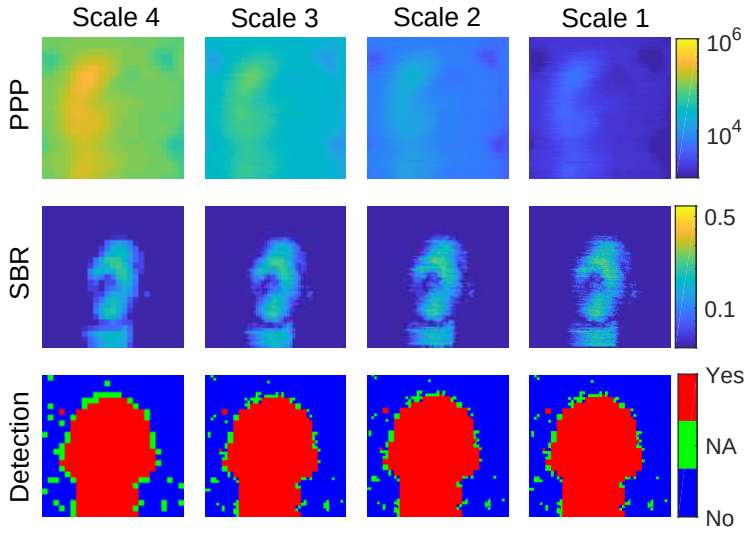


Figure 4.2 Integrating histograms within small windows of 2×2 pixels increases approximately 4 times the photons per histogram (top row), while reducing the amount of pixels to be processed 4 times. In most regions of the image, the SBR of the integrated waveforms does not change significantly (middle row), hence increasing the detection performance. The bottom row illustrates the detection strategy, starting from the coarse scale (scale 4) and refining the detection results sequentially in the uncertain regions (in green) using finer scales.

tests to be evaluated by a factor of 4. Figure 4.2 illustrates the multiscale approach. The worst case scenario corresponds to having only 1 pixel out of 4 which contains a target, where the SBR of the super-pixel is roughly 4 times smaller than at the finer scale. In such cases, the probability of target presence in the super-pixel is close to 0.5 (i.e., neither presence or absence of target is certain) and a more informed decision can potentially be made in the finer scale. Hence, super-pixels with $P(u = 0|\boldsymbol{z}) \leq 1 - \epsilon$ and $P(u = 1|\boldsymbol{z}) \leq 1 - \epsilon$ are left uncertain, and reprocessed as 4 individual pixels in the finer scale. This strategy starts at a coarse level of S scales, and is repeated until a decision has been made in all super-pixels or the finest scale is reached. The confidence level ϵ should be adjusted by the practitioner depending on the application. The number of scales should be set according to the size of the image and the expected detection detail. In all our experiments, we set $\epsilon = 0.05$ and $S = 4$.

4.5 Results

We evaluate the performance of the detection algorithm using synthetic and real lidar datasets. In all the results presented here, we assume that we know (from calibration measurements) the average number r_M of signal photons detected when observing an object of unit reflectivity under similar observation conditions as for the scene of interest. We then use this value to set $\mathbf{\Psi} = \{\alpha_r, \beta_r, \alpha_b, \beta_b\} = \{2, 2/r_M, 1, T/r_M\}$, which corresponds to a fairly informative prior for r and more weakly informative prior for b. The detection results are evaluated by the true positive rate (TPR), true negative rate (TNR), false positive rate (FPR) and false negative rate (FNR).

4.5.1 Synthetic data

First, we evaluate the pixel-wise detection method using individual synthetic histograms, generated with a Gaussian instrumental response with standard deviation $\sigma = T/100$. Receiver operating characteristic (ROC) curves are shown in Fig. 4.3a, where the effect of the threshold parameter ν is shown for different SBR values. As expected, the best choice is $\nu = 0.5$, which corresponds to a marginal posterior probability of 0.5. In all of the remaining experiments, we have fixed $\nu = 0.5$. Figure 4.3b shows the SBR vs PPP curves for TPR of 95%, without marginalising t_0 (i.e., the true value of t_0, estimated with the classical log-matched filter) and with the proposed marginalisation. The FPR of the detector is shown in Fig. 4.3c. While the sensitivity of the detector does not change significantly with the marginalisation of t_0 (see Fig. 4.3b), the probability of false alarm increases when t_0 is estimated using the standard cross-correlation. Figure 4.3d depicts a map of the empirical probability of detection (with t_0 marginalised) for various SBR and PPP. This figure gives an empirical bound on the minimum number of photons needed to detect a target with a given probability, for different levels of SBR, which can be used to adjust the acquisition time of the device in practice. In absence of a target, Fig. 4.3c shows that around 20 background detections are sufficient to correctly discard the histogram with high probability (>0.95).

Secondly, using a synthetic lidar scene (not individual histograms), we evaluate 3 detection procedures:

- **Single scale**: Decision rule in (4.8) applied independently to each pixel of the scene using (4.10) and (4.11).
- **Single scale + TV**: Computation of the probabilities (4.10) and (4.11) pixel-wise, TV-denoising and thresholding procedure as described in Section 4.4.1.
- **Multiscale**: Coarse-to-fine detection procedure as described in Section 4.4.2.

The synthetic scene is composed of a rectangular plane in the central region of the field of view and the corresponding reflectivity, background, and depth profiles generated are depicted in Fig. 4.4.

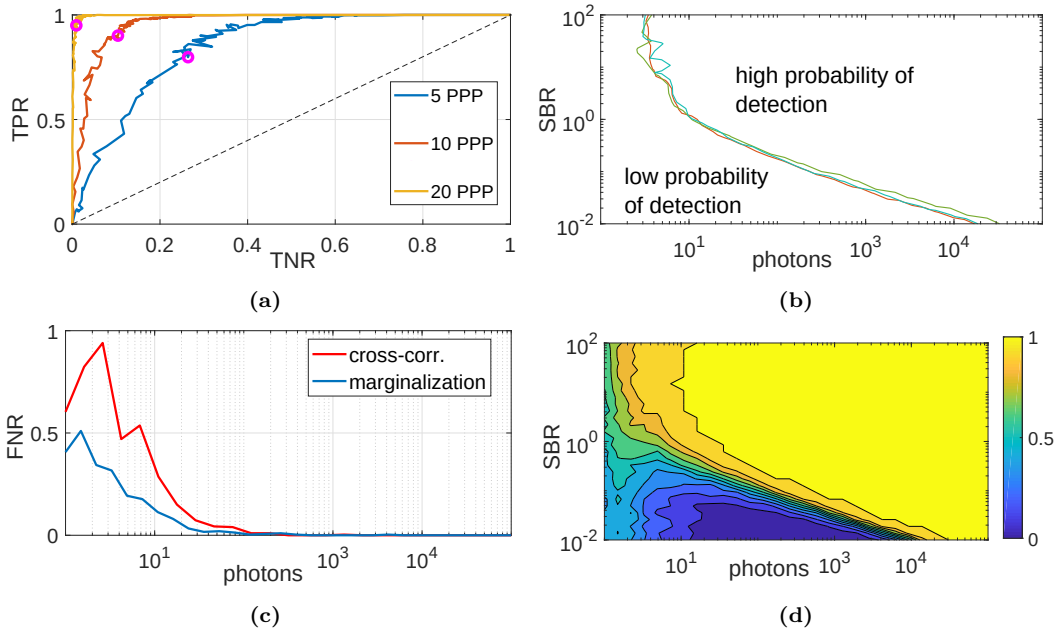


Figure 4.3 (a) ROC curve for different PPP as a function of the threshold level ν. The performance for $\nu = 0.5$ is shown in magenta for each curve. (b) Performance of the detection algorithm for different PPP and SBR. The solid lines correspond to a true positive rate of 95% with t_0 known (red), estimated by cross-correlation (blue) and marginalised (green). (c) FPR achieved by marginalising t_0 (blue curve) and by estimating t_0 via cross-correlation/matched filtering (red curve). (d) TPR as a function of the SBR and total number of photons.

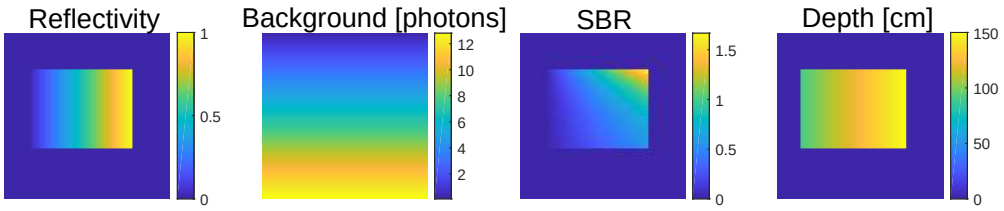


Figure 4.4 Synthetic lidar dataset. From left to right: reflectivity, per-pixel background photon-count, per-pixel signal-to-background ratio and ground truth depth.

Note that the rectangular shape does not present a constant depth profile but rather smooth variations mimicking a direction of observation that differs from the local normal to the surface. While the background levels vary in the vertical direction, the reflectivity profile changes in the horizontal direction, which allows us to vary the SBR across pixels, as well as the overall number of detected photons. With the parameters reported in Fig. 4.4, we obtain an average of 7.2 photons per pixel and an average SBR of 0.13.

Figure 4.5 depicts the detection results obtained using the 3 approaches mentioned above. First, note that the single scale approach provides a noisy detection map, with a high false alarm rate and a low detection rate in the low SBR region of the object. This can also be observed from Table 4.1 which reports the empirical TPR and TNR. Conversely, the two approaches using local information to regularise the detection problem provide less noisy detection maps and higher TPR. Moreover, while the TV regularisation can lead to underestimating the object size (in particular

in the low SBR region), the multiscale approach is slightly more robust. Interestingly, Table 4.1 also shows that the multiscale approach provides a reduction by 88% of the number of tests to be computed, when compared to the single scale and single scale + TV approaches, thus leading a significantly reduced computational time in serial architectures.

	TPR [%]	TNR [%]	Evaluations [%]
Single scale	65.64	77.37	100
Single scale + TV	84.32	**91.47**	100
Multiscale	**90.41**	84.74	**12**

Table 4.1 TPR, TNR and number of test evaluations per pixel for the evaluated algorithms on the synthetic dataset. For the multiscale approach, the uncertain regions are counted as regions where objects are present.

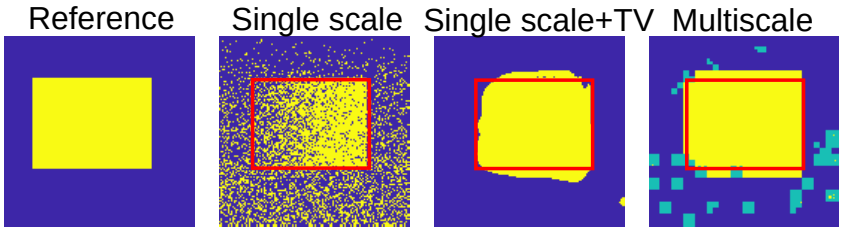


Figure 4.5 Target detection performance of the compared methods, target presence is indicated in yellow, while target absence is indicated in blue. The multiscale detection algorithm does not reach a decision for some pixels (depicted in greenish blue). To highlight the performance the evaluated algorithms, the red bounding boxes denote the limits of the ground truth object.

4.5.2 Real data

We compare the proposed detection algorithms with a single-depth [35] and multi-depth detection (ManiPoP) algorithms and the standard cross-correlation with reflectivity thresholding (see [35] for details) using real lidar dataset, which consists of a polystyrene head measured at a stand-off distance of 325 metres during midday (more details can be found in [35]). The hyperparameters of these two algorithms were chosen to obtain the best TNR/TPR trade-offs. The dataset consists of $N_r = N_c = 200$ pixels with $T = 2700$ histogram bins per pixel, a mean SBR of 0.29 with a 5th-95th percentile interval of $[0.05, 0.67]$.

Figure 4.6 shows the performance of the evaluated algorithms for 4 different per-pixel acquisition times (30, 3, 1 and 0.3 ms), which correspond to PPP of 900, 90, 30 and 9 photons, respectively. A ground truth detection map was obtained by choosing the majority among the detection maps of the evaluated methods in the 30 ms acquisition time case. TPR and TNR for all algorithms and acquisition times are reported in Table 4.2. These results show that although the single scale detection algorithm is applied pixel-wise, it generally provides better results than the cross-correlation method, a significant improvement in terms of TPR and TNR is obtained by using the additional denoising step presented in Sections 4.4.1 and 4.4.2, which accounts for spatial correlation between

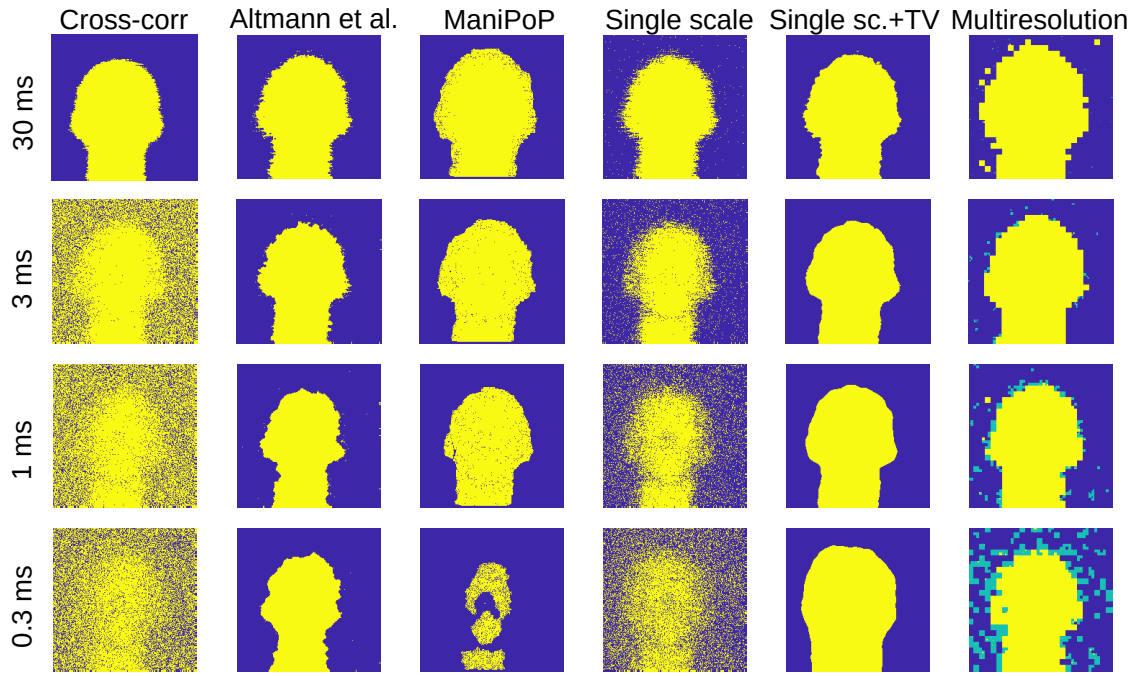


Figure 4.6 Target detection performance of the compared methods, target presence is indicated in yellow, while target absence is indicated in blue. The multiscale detection algorithm does not reach a decision for some pixels (depicted in greenish blue). To highlight the performance the evaluated algorithms, the red bounding boxes denote the limits of the ground truth object.

adjacent pixels. In contrast to the TV-based method presented in this chapter, applying a TV post-processing step directly to the cross-correlation output is challenging, as the per pixel detection probabilities are not available. Note that although the TV-based regularisation yields a small improvement in terms of TPR in the 0.3 ms case, the corresponding TNR is significantly smaller. Both the multiscale and TV approaches provide better detection maps than [35] and ManiPoP in the very low acquisition time scenario (0.3 ms).

The largest difference between the evaluated methods is the execution time. The previous detection method [35] requires more than 12 h for all datasets and ManiPoP requires more than 300 s. In contrast, a parallel implementation of cross-correlation has an execution time smaller

	TPR [%]				TNR [%]			
	30 ms	3 ms	1 ms	0.3 ms	30 ms	3 ms	1 ms	0.3 ms
Cross-corr.	91.67	95.65	89.44	80.96	98.26	34.72	34.48	42.17
Altmann et al. [35]	98.34	87.96	80.39	78.52	**99.67**	**99.83**	**99.81**	99.87
ManiPoP	99.39	94.74	86.05	27.41	90.83	98.16	99.64	**100**
Single scale	98.52	82.29	70.87	70.29	99.34	93.85	86.61	68.38
Single scale + TV	98.41	93.75	89.63	**98.34**	97.10	99.09	99.51	89.40
Multiresolution	**99.98**	**99.29**	**98.58**	97.77	78.82	90.50	90.73	72.78

Table 4.2 TPR and TNR for the evaluated detection algorithms. Pixels without a decision in the multiscale method were considered as indicating the presence of a target. The ground truth was obtained by choosing a consensus between the evaluated methods in the 30 ms case.

than 5 ms for all datasets (see Section 5.5). As evaluating (4.11) has a complexity of running $P \approx 20$ cross-correlations (using a standard quadrature method), we can expect execution times of the order of 100 ms in a parallel implementation of the proposed detection algorithms. This gap could be reduced further by using more advanced numerical integration schemes. Note that the TV step can be performed in parallel, only adding an overhead of a few milliseconds.

Table 4.3 illustrates how the proposed multiscale approach allows a general reduction of the number of tests performed to construct the detection maps in Fig. 4.6. While the single scale and single scale + TV approaches require a test per pixel, only 4% of that maximum number is used for the longest acquisition time. This gain generally reduces as the acquisition time decreases since the data become more uncertain. Note that, for the shortest acquisition time (most difficult scenario), the original overall number of tests in divided by 4.

Algo./Acq. time	30 ms	3 ms	1 ms	0.3 ms
Single scale + TV	100	100	100	100
Multiscale	4.0	3.4	7.7	24.5

Table 4.3 Percentage of detection evaluations (normalised by the number of pixels in the image) for different acquisition times.

4.6 Conclusions

This chapter has presented a fast target detection algorithm for single-photon lidar data. Unlike other competing algorithms, this algorithm is easily parallelisable and can be used as a pre-processing step to discard histograms without useful information. The TV approach can be used in parallel architectures, as the TV denoising step can be efficiently computed parallel, whereas the multiscale approach is well-suited for sequential architectures, where fewer evaluations of the decision rule (4.8) are necessary.

This algorithm can improve the reconstruction quality obtained by methods assuming one depth per pixel, as it removes histograms without surfaces from the data cube. It can also be used before multi-depth algorithms (e.g., ManiPoP and MuSaPoP) to reduce the computational load. Moreover, the performance bounds shown in Fig. 4.3 can be used to adjust the acquisition time depending on the minimum admissible SBR.

The next chapter presents an algorithm that is not limited to the target detection setting. As ManiPoP, it can handle the general multi-depth scenario and estimates all the model parameters, but can also perform the reconstruction in real-time.

Chapter 5

Real-time 3D imaging

Contents

5.1 Introduction

Recent advances in arrayed SPAD technology now allow rapid acquisition of data [98,99], meaning that full-field 3D image acquisition can be achieved at video rates, or higher, placing a severe bottleneck on the processing of data. Existing approaches, such as the ones presented in Chapters 1 to 3, are either too slow or not robust enough and thus do not allow rapid analysis of dynamic

scenes (e.g., road activity monitoring), and subsequent automated decision-making processes. In this chapter, we propose a new algorithm structure, differing significantly from existing approaches, to meet speed, robustness and scalability requirements.

Under a single-depth assumption, alternatives based on convex optimisation tools and spatial regularisation, such as SPISTA [36], ℓ_{21}+TV [37] or Rapp and Goyal [25] need several seconds to minutes runtime to converge for a single image. The parallel optimisation algorithm in [30] still reported reconstruction times of the order of seconds. Even the recent algorithm based on a convolutional neural network [34] does not meet real-time requirements after training.

As discussed in Section 2.5, multi-depth optimisation-based methods [10, 36] also require execution times of the order of minutes. While ManiPoP is also unsuitable for real-time applications, it improved the reconstruction quality and execution time of optimisation methods based on a intensity cube [10, 36]. This improvement, mostly due to ManiPoP's ability to model 2D surfaces in a 3D volume using structured point clouds, has motivated our new method which uses scalable point cloud denoising tools from the computer graphics community.

Here we introduce a new algorithm that can process dozens of frames per second, achieving state-of-the-art reconstructions in the general multiple-surface per pixel setting. The novel method efficiently models the target surfaces as two dimensional manifolds embedded in a 3D space. This is achieved using manifold modelling and point cloud denoising tools from the computer graphics community (see [100] for a complete survey). A typical computer graphics pipeline for 3D reconstruction consists of a 2-step process using a simple maximum likelihood algorithm to find a rough (initial) point cloud estimate from the data and then a second step using a point cloud denoising algorithm capable of rapidly processing millions of points. This strategy is efficient when the initial point cloud is dense and moderately noisy but it does not provide satisfactory results here due to the general poor quality of the initial point cloud (extracted from single-photon lidar waveforms) and the relatively small number of pixels of current SPAD arrays - e.g., the 32×32 array of the Kestrel Princeton Lightwave camera used in the experimental section of this chapter. Moreover, this strategy does not take into account a priori information on the observation model, such as presence of dead pixels [12, 101] or compressive sensing strategies [75, 102]. On the other hand, most image processing approaches work with the full lidar waveforms and use accurate observation models but their denoising tools are not tailored for point clouds, as mentioned above.

Here we propose a new inference scheme which benefits from the best of each strategy. The 3D reconstruction algorithm works directly on the lidar waveforms, making use of off-the-shelf computer graphics point cloud denoisers for distributed surfaces. We extend and adapt the ideas of plug-and-play priors [46, 103, 104] and regularisation by denoising [47], which have recently appeared in the image processing community, to point cloud restoration. The method iterates between gradient descent steps, which take into account the observation model, and denoising steps, which benefit from powerful point cloud denoising techniques from the computer graphics

literature (e.g., [105]). The resulting algorithm achieves real-time processing of 3D lidar videos due to the intrinsic parallel architecture of the gradient evaluation steps and of the point cloud denoising strategy.

The remainder of the chapter is organised as follows: Section 5.2 introduces the real-time reconstruction algorithm and Section 5.3 details its parallel implementation. Different choices of point cloud denoisers are discussed in Section 5.4. Section 5.5 presents results using raster-scanning lidar data and lidar array videos. Section 5.6 extends the real-time framework to multispectral lidar. Finally, Section 5.7 discusses conclusions.

5.2 Real-time 3D reconstruction algorithm

As in ManiPoP, we avoid the issues induced by the high dimensionality of the intensity parameters involved in SPISTA and ℓ_{21}+TV, while allowing for a variable number of surfaces per pixel. More precisely, here $\boldsymbol{t}$ and $\boldsymbol{r}$ are sets of variable size $N_{\boldsymbol{\Phi}}$. However, instead of relying on an RJ-MCMC sampler which is an inherently slow process, we use optimisation techniques that achieve fast convergence rates and can be easily parallelised. Hence, in this chapter, we use the vector notation $\boldsymbol{r}$, $\boldsymbol{t}$ and $\boldsymbol{b}$ as introduced in Section 1.2.3. We focus on MAP estimation, solving the problem (recall Section 1.3)

$$(\hat{\boldsymbol{t}}, \hat{\boldsymbol{r}}, \hat{\boldsymbol{b}}) = \underset{\boldsymbol{t},\boldsymbol{r},\boldsymbol{b}}{\arg\min}\, g\left(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}\right) + \lambda_{\boldsymbol{t}}\rho_{\boldsymbol{t}}(\boldsymbol{t}) + \lambda_{\boldsymbol{r}}\rho_{\boldsymbol{r}}(\boldsymbol{r}) + \lambda_{\boldsymbol{b}}\rho_{\boldsymbol{b}}(\boldsymbol{b}) \tag{5.1}$$

In contrast to single-depth algorithms that generally rely on a total variation regularisation for $\boldsymbol{t}$ and $\boldsymbol{r}$, as in plug-and-play denoising [46], we define them implicitly through their proximal operators. Nonetheless, instead of using image denoisers (or volume denoisers), we leverage point cloud denoisers from the computer graphics literature.

Reparametrisation

In a similar fashion to other optimisation algorithms assuming Poisson observation noise [106,107], we introduce the transformation

$$m_n = \log r_n \quad \forall n = 1, \ldots, N_{\boldsymbol{\Phi}} \tag{5.2}$$

and fix a maximum intensity $m_n \in (-\infty, \log r_{\max}]$. This change of variables and additional constraint ensure that the likelihood has a Lipschitz-continuous gradient with respect to $\boldsymbol{r}$. The vectorised set of log-intensity values is denoted by $\boldsymbol{m} = [m_1, \ldots, m_{N_\Phi}]^\mathsf{T}$. Analogously, we estimate the log-background levels, i.e., $\tilde{b}_{i,j} = \log b_{i,j}$, denoting the vectorised log-background image as $\tilde{\boldsymbol{b}} = [\tilde{b}_{1,1}, \ldots, \tilde{b}_{N_r N_c}]^\mathsf{T}$. The resulting negative log-likelihood function (recall (1.7)) under this

parametrisation is

$$g\left(\boldsymbol{t},\boldsymbol{m},\tilde{\boldsymbol{b}}\right)=\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}\sum_{t=1}^{T}e^{\tilde{b}_{i,j}}+\sum_{\mathcal{N}_{i,j}}e^{m_n}h_{i,j}(t-t_n)-z_{i,j,t}\log\left(\sum_{\mathcal{N}_{i,j}}e^{m_n}h_{i,j}(t-t_n)+e^{\tilde{b}_{i,j}}\right). \quad (5.3)$$

5.2.1 Proximal gradient steps

We follow the structure of the proximal alternating linearised minimisation for non-convex and non-smooth problems algorithm (PALM) [108] to solve problem (5.1). The resulting algorithm, referred to as RT3D, alternates between the optimisation of 3 blocks of variables ($\boldsymbol{t}$, $\boldsymbol{m}$ and $\tilde{\boldsymbol{b}}$), applying a proximal gradient update on each step, i.e.,

$$\begin{cases}\boldsymbol{t}^{*} & \leftarrow \boldsymbol{t}^{s}-\mu_t\nabla_{\boldsymbol{t}}g\left(\boldsymbol{t}^{s},\boldsymbol{m}^{s},\tilde{\boldsymbol{b}}^{s}\right)\\ \boldsymbol{t}^{s+1} & \leftarrow \arg\min_{\boldsymbol{t}}\lambda_{\boldsymbol{t}}\rho_{\boldsymbol{t}}(\boldsymbol{t})+\frac{1}{2\mu_t^s}||\boldsymbol{t}-\boldsymbol{t}^{*}||_2^2\end{cases} \quad (5.4)$$

$$\begin{cases}\boldsymbol{m}^{*} & \leftarrow \boldsymbol{m}^{s}-\mu_m\nabla_{\boldsymbol{m}}g\left(\boldsymbol{t}^{s+1},\boldsymbol{m}^{s},\tilde{\boldsymbol{b}}^{s}\right)\\ \boldsymbol{m}^{s+1} & \leftarrow \arg\min_{\boldsymbol{m}}\lambda_{\boldsymbol{m}}\rho_{\boldsymbol{m}}(\boldsymbol{m})+\frac{1}{2\mu_r^s}||\boldsymbol{m}-\boldsymbol{m}^{*}||_2^2\end{cases} \quad (5.5)$$

and

$$\begin{cases}\tilde{\boldsymbol{b}}^{*} & \leftarrow \tilde{\boldsymbol{b}}^{s}-\mu_b\nabla_{\tilde{\boldsymbol{b}}}g\left(\boldsymbol{t}^{s+1},\boldsymbol{m}^{s+1},\tilde{\boldsymbol{b}}^{s}\right)\\ \tilde{\boldsymbol{b}}^{s+1} & \leftarrow \arg\min_{\tilde{\boldsymbol{b}}}\lambda_{\tilde{\boldsymbol{b}}}\rho_{\tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{b}})+\frac{1}{2\mu_b^s}||\tilde{\boldsymbol{b}}-\tilde{\boldsymbol{b}}^{*}||_2^2\end{cases} \quad (5.6)$$

where μ_t, μ_m and μ_b are the step sizes for the depths, log-intensities and log-background levels respectively. The gradients with respect to the depth, log-intensity and log-background levels are denoted by $[\nabla_{\boldsymbol{t}}g\left(\boldsymbol{t},\boldsymbol{m},\tilde{\boldsymbol{b}}\right)]_n = \partial g\left(\boldsymbol{t},\boldsymbol{m},\tilde{\boldsymbol{b}}\right)/\partial t_n$, $[\nabla_{\boldsymbol{m}}g\left(\boldsymbol{t},\boldsymbol{m},\tilde{\boldsymbol{b}}\right)]_n = \partial g\left(\boldsymbol{t},\boldsymbol{m},\tilde{\boldsymbol{b}}\right)/\partial m_n$ and $[\nabla_{\tilde{\boldsymbol{b}}}g\left(\boldsymbol{t},\boldsymbol{m},\tilde{\boldsymbol{b}}\right)]_n = \partial g\left(\boldsymbol{t},\boldsymbol{m},\tilde{\boldsymbol{b}}\right)/\partial \tilde{b}_n$.

Depth denoising The key observation of this chapter is to extend the ideas introduced for plug-and-play denoising to 3D point clouds, replacing the proximal operator of (5.4) by the algebraic point set surfaces (APSS) algorithm [105, 109], i.e.,

$$\boldsymbol{t}^{s+1} \leftarrow \text{APSS}(\boldsymbol{t}^{*}). \quad (5.7)$$

The APSS algorithm fits a continuous surface to the set of points defined by $\boldsymbol{t}^{*}$, using spheres as local primitives. The algebraic spheres are parametrised by the vector $\boldsymbol{u}=[u_0,\ldots,u_4]^{\mathsf{T}}$, according

to the scalar field $\phi : \mathbb{R}^3 \to \mathbb{R}$, defined as

$$\phi_{\boldsymbol{u}}(\boldsymbol{c}) = [1, \boldsymbol{c}^\mathsf{T}, \boldsymbol{c}^\mathsf{T}\boldsymbol{c}]\boldsymbol{u}. \tag{5.8}$$

For each 3D point $\boldsymbol{c}_n = [i, j, t_n]^\mathsf{T}$, the local sphere is fitted by solving the following problem

$$\arg\min_{\boldsymbol{u}} \sum_{r=1}^{N_{\boldsymbol{\Phi}}} \omega\left(||\boldsymbol{c}_n - \boldsymbol{c}_r||_\Sigma\right) \phi_{\boldsymbol{u}}^2(\boldsymbol{c}_r) \tag{5.9}$$

where $\omega(t) = \left(1 - t^2\right)^4$ is a smooth compactly supported weight function and $||\boldsymbol{c}||_{\boldsymbol{\Sigma}} = \boldsymbol{c}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{c}$ is a metric of choice, with $\boldsymbol{\Sigma}$ a diagonal matrix with positive entries, which controls the degree of low-pass filtering of the surface. In particular, $\boldsymbol{\Sigma}$ was chosen with diagonal entries, i.e.,

$$\boldsymbol{\Sigma} = \begin{pmatrix} d_x & 0 & 0 \\ 0 & d_y & 0 \\ 0 & 0 & d_t \end{pmatrix}. \tag{5.10}$$

In all of the experiments, we set $d_x = d_y = 1$, such that only the 8 closest neighbouring pixels have strong weights, and d_t to be the minimum distance between two surfaces in the same transverse pixel, which is chosen according to the bin width of the lidar system to have a physical meaning (also to yield the best results, as shown in the experimental analysis conducted in Section 5.2.5). Interestingly, it is the same distance as the hard constraint between points in ManiPoP (Section 2.2.1). The fitting is performed in real-world coordinates, which equates to scaling the depth parameters by $\frac{\Delta_b}{\Delta_p}$. Figure 5.1 illustrates the surface fitting performed by APSS, and its pseudocode is presented in Algorithm 5. The implicit definition of the scalar field is evaluated in every pixel with at least 3 neighbours, filling any holes and dilating the existing surfaces. As in the *almost orthogonal* projection described in [110], we repeat the fitting process until there is no significant change in the projected point.

Intensity denoising The proximal operator of the log-intensity update in (5.5) is replaced by a denoising step using the manifold metrics. We simply consider a low-pass filter using the nearest neighbours of each point[1], as in [56]: each log-intensity m_n is updated as

$$m_n^{s+1} = \beta m_n^* + (1 - \beta) \sum_{n' \in \mathcal{M}(m_n^*)} \frac{m_{n'}^*}{\#\,\mathcal{M}(m_n^*)} \tag{5.11}$$

where β is a coefficient controlling the amount of filtering, $\mathcal{M}(m_n)$ is the set of spatial neighbours m_n and $\#\,\mathcal{M}(m_n)$ denotes the total number of neighbours. Hence, the proximal step is summarised

[1]While we use a simple nearest neighbours approach, it is possible to use the manifold metrics defined by the implicit mean least squares surface, as explained in [111]

Algorithm 5 Algebraic point set surfaces denoiser for lidar point clouds

Input: point positions $\boldsymbol{t}$
Main loop: Process each pixel (i, j) in parallel
Read depths of points in (i, j) and neighbouring pixels
Convert depth to real-world coordinates: $t_n \leftarrow t_n \frac{\Delta_b}{\Delta_p}$
Group points according to their depth into K clusters, such that each cluster centre t_i is separated to the rest by at least $2d_t$ and the minimum cluster size is 3
for $i = 1, \ldots, K$ **do**
  $s \leftarrow 1$
  $t_i^s \leftarrow i$th cluster centre
  while $s = 1$ or $|t_i^s - t_i^{s-1}| > 1$ **do**
    Fit an algebraic surface in the neighbourhood of $\boldsymbol{c}_n = [i, j, t_i^s]^\mathsf{T}$ by solving (5.9)
    $t_i^{s+1} \leftarrow$ depth of fitted surface at (i, j) using (5.8)
    $s \leftarrow s + 1$
  end while
end for
Convert depths back to histogram coordinates: $t_n \leftarrow t_i \frac{\Delta_p}{\Delta_b}$
Output: denoised point positions $\boldsymbol{t}$

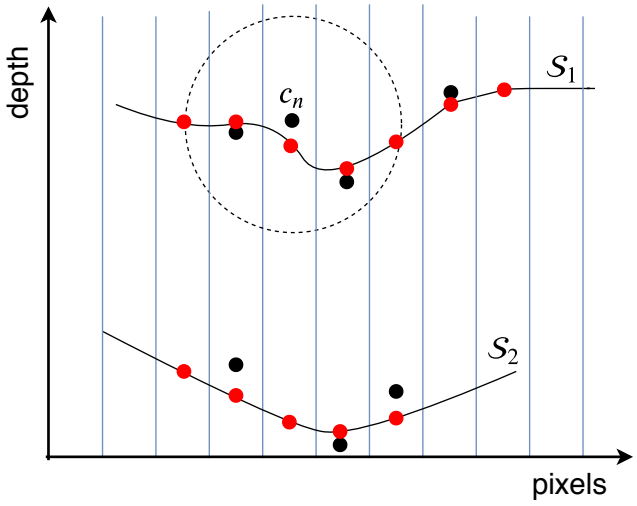


Figure 5.1 Illustration of the APSS denoising step. This example presents two surfaces $\mathcal{S}_1$ and $\mathcal{S}_2$ per pixel. The input and output points are depicted in black and red respectively. The algorithm fits a continuous surface (black line) using local spheres centred at each input point $\boldsymbol{c}_n$. The fitting is performed using a weighted least squares algorithm, where the weighting kernel is defined by a metric Σ (dashed-line circle). Note that the points in $\mathcal{S}_1$ are not affected by the ones in $\mathcal{S}_2$, as the weighting kernel vanishes at the points in $\mathcal{S}_2$. Thus, the denoiser can process an arbitrary number of surfaces per pixel.

as

$$\boldsymbol{m}^{s+1} \leftarrow \text{Manifold denoising}(\boldsymbol{m}^*) \tag{5.12}$$

More elaborate filters could also be applied, such as the bilateral filter (see Section 5.6). After the denoising step, we remove the points with intensity r_n lower than a given threshold $r_{\min}$. This step prevents the algorithm from growing surfaces without bounds. As discussed in Section 5.2.5, the threshold can be set to the minimum admissible reflectivity.

Background denoising The proximal operator used for $\tilde{\boldsymbol{b}}$ depends on the prior assumptions that can be made about the spatial configuration of the spurious detections. In bistatic raster-scanning, the proximal operator can be chosen as the identity operator (i.e., no regularisation). In monostatic raster-scanning systems or lidar arrays, we use a Gaussian Markov random field regularisation [55], i.e., $\rho_{\tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{b}}) = \tilde{\boldsymbol{b}}^{\mathsf{T}} \boldsymbol{P} \tilde{\boldsymbol{b}}/2$, where $\boldsymbol{P}$ is the Laplacian 2D filter. The proximal operator is thus

$$\tilde{\boldsymbol{b}}^{s+1} \leftarrow (\boldsymbol{I} + \lambda_{\tilde{\boldsymbol{b}}} \mu_b \boldsymbol{P})^{-1} \tilde{\boldsymbol{b}}^* \tag{5.13}$$

where $\boldsymbol{I}$ is the identity matrix. This denoising step can be quickly computed using the FFT[2]. The proximal operator can also be replaced by an off-the-shelf image denoising algorithm, such as non-local means [112] or BM3D [113], at the cost of a higher computational load.

5.2.2 Setting the step sizes

Assuming the number of points is constant, the step sizes should verify $\mu_t < \frac{1}{L_t}$, $\mu_m < \frac{1}{L_m}$ and $\mu_b < \frac{1}{L_b}$, where L_t^s, L_m and L_b are the Lipschitz constants of $\nabla_{\boldsymbol{t}} g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right)$, $\nabla_{\boldsymbol{m}} g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right)$ and $\nabla_{\tilde{\boldsymbol{b}}} g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right)$ respectively (see Section 5.2.3). The value of L_t can be upper bounded by the maximum eigenvalue of the Hessian matrix using Gershgorin circle theorem [114], i.e.,

$$L_t \leq \max_n \sum_{k=1}^{N_\Phi} \left| \frac{\partial g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right)}{\partial t_n \partial t_k} \right| \tag{5.14}$$

If the impulse response has a Gaussian shape, i.e., $h_{i,j}(t) \propto \exp\left[-(t/\sigma)^2/2\right]$, the partial derivatives can be computed analytically, leading to

$$L_t \leq \frac{1}{\sigma^2} \max_{i,j} \sum_{t=1}^{T} z_{i,j,t} \tag{5.15}$$

which only depends on the width of the impulse response and the maximum number of photons per pixel. The values of L_m and L_b are bounded by the maximum point intensity and background

[2]The matrix inverse is precomputed offline, only requiring to filter $\tilde{\boldsymbol{b}}^*$ at each iteration.

level, that is

$$L_m \leq \sum_{t=1}^{T} h_{i,j}(t) \max_n e^{m_n} \tag{5.16}$$

$$L_b \leq T \max_{i,j} e^{\tilde{b}_{i,j}}. \tag{5.17}$$

As the maximum log-intensity is bounded to $\log r_{\max}$, $L_m \leq r_{\max}$. The background levels are also upper bounded by $b_{\max}$, meaning that $L_b \leq 1/(Tb_{\max})$.

5.2.3 Convergence

The algorithm defined by steps (5.4) to (5.6) converges to a critical point of the objective function (5.1) if the following assumptions are fulfilled [108]:

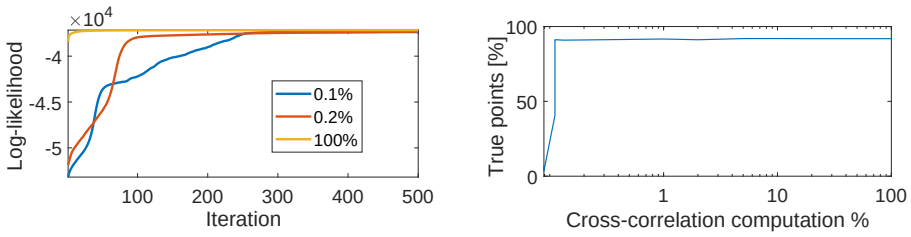
1. $g(\boldsymbol{t}, \boldsymbol{m}, \boldsymbol{b}) : \mathbb{R}^{N_\Phi} \times \mathbb{R}^{N_\Phi} \times \mathbb{R}^{N_r N_c} \mapsto \mathbb{R}$ is a C^1 (continuously differentiable) function.
2. The partial gradients used in (5.4) to (5.6) are Lipschitz. More precisely
   - For any fixed $(\boldsymbol{m}, \tilde{\boldsymbol{b}})$, the partial gradient $\nabla_{\boldsymbol{t}} g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right)$ is Lipschitz with constant L_t.
   - For any fixed $(\boldsymbol{t}, \tilde{\boldsymbol{b}})$, the partial gradient $\nabla_{\boldsymbol{m}} g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right)$ is Lipschitz with constant L_m.
   - For any fixed $(\boldsymbol{t}, \boldsymbol{m})$, the partial gradient $\nabla_{\tilde{\boldsymbol{b}}} g(\boldsymbol{t}, \boldsymbol{m}, \boldsymbol{b})$ is Lipschitz with constant L_b.
3. $\nabla g(\boldsymbol{t}, \boldsymbol{m}, \boldsymbol{b})$ is Lipschitz continuous on bounded subsets of $\mathbb{R}^{N_\Phi} \times \mathbb{R}^{N_\Phi} \times \mathbb{R}^{N_r N_c}$.
4. The regularisation terms $\rho_{\boldsymbol{t}}(\boldsymbol{t})$, $\rho_{\boldsymbol{m}}(\boldsymbol{m})$ and $\rho_{\tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{b}})$ are inf-bounded, proper and lower semi-continuous functions.

As explained in Section 5.2.2, conditions (1) and (2) are fulfilled by RT3D. Moreover, as $g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right)$ is also a C^2 function, condition (3) can be easily verified via the mean value theorem. The intensity denoiser given by (5.12) is a low-pass filter that can be linked to a quadratic penalty [55], i.e., $\rho_{\boldsymbol{m}}(\boldsymbol{m}) = \boldsymbol{m}^\mathsf{T} \boldsymbol{P} \boldsymbol{m}$ with positive semi-definite matrix $\boldsymbol{P}$, hence verifying condition (4). Similarly, the background prior $\rho_{\tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{b}})$ also verifies (4). However, it is not easy to verify whether the APSS denoiser can be linked with an explicit function $\rho_{\boldsymbol{t}}(\boldsymbol{t})$ or not[3]. Moreover, the algorithm changes the dimension of $\boldsymbol{t}$ and $\boldsymbol{m}$ at each step. While a complete convergence analysis is left for future work, the next subsection shows that the algorithm converges to fixed points in practical scenarios and is robust to different initialisations.

5.2.4 Initialisation

The initialisation step of the algorithm is designed to provide a coarse estimate, while being fast and easy to parallelise. If at most one surface per pixel is expected, then the classical cross-correlation

[3]Note that this is the case of many denoisers in the plug-and-play literature for image restoration [104]. Recent works relax the existence of $\rho_{\boldsymbol{t}}(\boldsymbol{t})$, studying fixed point convergence of plug-and-play ADMM using general denoisers as vector valued maps (not necessarily proximal maps) [104, 115].



(a) **(b)**

Figure 5.3 Robustness to the initialisation. The initialisation by cross-correlation is computed only using a fraction (%) of bins out of the total number of histogram bins. (a) Value of the log-likelihood as a function of the iterations of RT3D for different initialisations. In all cases, the algorithm converges to a similar maximum of the log-likelihood function. (b) True detections as a function of the computed bins in the cross-correlation initialisation.

can be applied. Figure 5.2 shows the initialisation (top row) and achieved reconstructions (bottom row) for different decimations of the cross-correlation initialisation. The decimation consists in discarding bins of the cross-correlation output before finding the one associated with the maximum. Decimating the cross-correlation function reduces to considering a reduced number of admissible ranges, which in turn reduces the computational complexity of the initialisation. For instance, Fig. 5.2 (b) uses only 3 admissible depths (i.e., the cross-correlation is computed only for 3 bins out of 4613). Yet, the algorithm yields the same reconstruction even if 0.11% of the total cross-correlation is computed. As shown in Fig. 5.3, the algorithm recovers the same quantity of true points for a wide range of initialisation, converging to the same likelihood value and point cloud configuration.

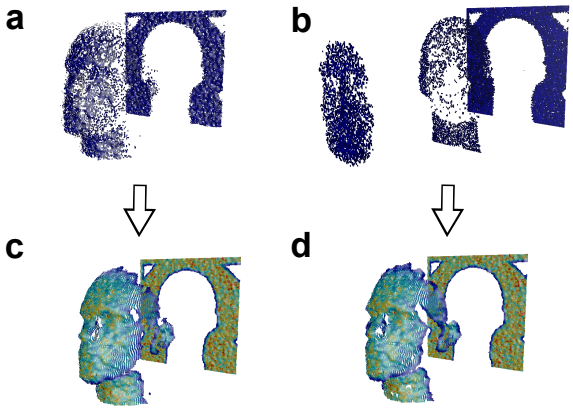


Figure 5.2 Reconstructions from different initialisations. Figures (a) and (b) show the initialisation of the algorithm when computing the 100% and 0.11% of the total cross-correlation. The reconstructions obtained after running the proposed reconstruction algorithm are shown in (c) and (d). Despite using different initialisations, the algorithm converges to similar reconstructions.

In a general setting where multiple surfaces may be present, we initialise the algorithm with a multi-surface extension of the classic cross-correlation. We propose 3 different alternatives depending on the sparsity of the recorded histograms:

- Lidar arrays can present dense histograms, so that we can use the Anscombe transform [116] to stabilise the variance of the Poisson noise[4]. After the transform, the matching pursuit algorithm [117] is used to find the K most prominent surfaces on each pixel, as summarised in Algorithm 6. The algorithm reduces to standard match filtering if $K = 1$.

Algorithm 6 Dense Anscombe matching pursuit initialisation

Input: lidar waveforms $\boldsymbol{z}_{i,j} = [z_{i,j,1}, \dots, z_{i,j,T}]^{\mathsf{T}}$, maximum number of surfaces per pixel K
Main loop: Process each pixel (i, j) in parallel
$\tilde{\boldsymbol{z}}_{i,j} \leftarrow 2\sqrt{\boldsymbol{z}_{i,j} + 3/8}$
$t_1, \dots, t_K \leftarrow$ Matched Pursuit using $\tilde{\boldsymbol{z}}_{i,j}$ and atoms given by the shifted impulse response $h_{i,j}(t)$
for $k = 1, \dots, K$ **do**
  $m_s \leftarrow \log(\sum_{t:h_{i,j}(t-t_k)\neq 0} z_{i,j,t})$
end for
$\tilde{b}_{i,j} = \log(\sum_{t\in\mathcal{T}} z_{i,j,t} / \sum_{t\in\mathcal{T}} 1)$ where $\mathcal{T} = \{t : h_{i,j}(t - t_k) = 0 \quad \forall k = 1, \dots, K\}$
Output: initial estimates $(\boldsymbol{t}^0, \boldsymbol{m}^0, \tilde{\boldsymbol{b}}^0)$

- Histograms collected using single-photon lidar systems with high temporal resolution (<20 ps), e.g., raster-scanning systems, generally present a large number of sparsely populated bins, hindering any dense computations using the Anscombe transform. In this case, we find the K most prominent peaks by iteratively using the cross-correlation estimate and removing the photons associated with the peak, as shown in Algorithm 7.

Algorithm 7 Sparse matching pursuit initialisation

Input: lidar waveforms $\boldsymbol{z}_{i,j} = [z_{i,j,1}, \dots, z_{i,j,T}]^{\mathsf{T}}$, maximum number of surfaces per pixel K
Main loop: Process each pixel (i, j) in parallel
for $k = 1, \dots, K$ **do**
  $t_k \leftarrow$ Cross-correlation maximum$(\boldsymbol{z}_{i,j})$
  $m_k \leftarrow \log(\sum_{t:h_{i,j}(t-t_k)\neq 0} z_{i,j,t})$
  $z_{i,j,t} \leftarrow 0 \quad \forall t : h_{i,j}(t - t_k) \neq 0$
end for
$\tilde{b}_{i,j} = \log(\sum_{\mathcal{T}} z_{i,j,t} / \sum_{\mathcal{T}} 1)$ where $\mathcal{T} = \{t : h_{i,j}(t - t_k) = 0 \quad \forall k = 1, \dots, K\}$
Output: initial estimates $(\boldsymbol{t}^0, \boldsymbol{m}^0, \tilde{\boldsymbol{b}}^0)$

- The main disadvantage of Algorithms 6 and 7 is that they have a complexity of $\mathcal{O}(KT \log T)$ and $\mathcal{O}(KT_aT)$ respectively, where T_a is the number of histogram bins with non-zero detections ($T_a \leq T$). Cross-correlation and other MP alternatives also present similar complexities. These complexities are worse than the one of the proximal gradient steps (as it will be explained in Section 5.3). Hence, the initialisation step becomes the processing bottleneck in lidar datasets with a large number of active bins per pixel. If there is at most one surface per pixel, a more efficient initialisation can solve this problem. Following the alternative interpretation of individual photon detections as samples from a mixture of a uniform distribution

[4]Note that the impulse response shape changes after the non-linear transformation given by the Anscombe transform. However, we noted that using the original $h(t)$ was enough to obtain good initialisations.

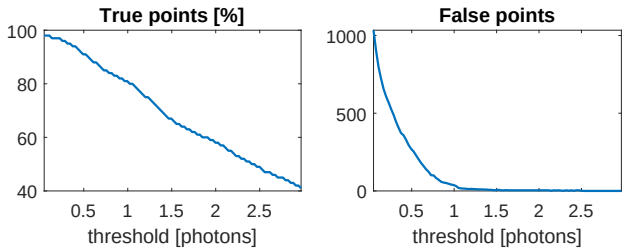


Figure 5.4 Reconstruction performance as a function of the intensity threshold for the head with backplane. The best performing values have a normalised intensity between 5% and 10%.

(background photons) and a normalised density $h(t-t_n)/\sum_{t=1}^{T} h(t)$ (signal photons) [31], we can use a robust mode estimator to find the location parameter t_n. In particular, we choose the half sample mode estimator [118], as shown in Algorithm 8. This estimator has a complexity of order $\mathcal{O}(T_a \log T_a)$, requiring significantly fewer operations than log-match filtering if $T_a \ll T$, without losing estimation performance.

Algorithm 8 Half sample mode initialisation (at most one surface per pixel)

Input: lidar waveforms $\boldsymbol{z}_{i,j} = [z_{i,j,1}, \ldots, z_{i,j,T}]^{\mathsf{T}}$
Main loop: Process each pixel (i,j) in parallel
Initialise lower and upper interval limits $t_l = 1$ and $t_u = T$
while $|t_u - t_l| > 1$ **do**
    Find bins $t_l \leq t_a \leq t_b \leq t_c \leq t_u$ such that the same number of photons is found in $[t_l, t_a]$, $[t_a, t_b]$, $[t_b, t_c]$ and $[t_b, t_u]$
    $t_l \leftarrow t_a$
    $t_u \leftarrow t_c$
end while
$t_1 \leftarrow t_l$
$m_1 \leftarrow \log(\sum_{t:h_{i,j}(t-t_1)\neq 0} z_{i,j,t})$
$\tilde{b}_{i,j} = \log(\sum_{\mathcal{T}} z_{i,j,t} / \sum_{\mathcal{T}} 1)$ where $\mathcal{T} = \{t : h_{i,j}(t-t_1) = 0\}$
Output: initial estimates $(\boldsymbol{t}^0, \boldsymbol{m}^0, \tilde{\boldsymbol{b}}^0)$

5.2.5 Setting the hyperparameters

In this section, we study the impact of the hyperparameters on the reconstruction performance, with the aim of providing basic guidelines to select them. We evaluate the performance using the head with backplane dataset. Figure 5.4 shows the number of true and false detections as a function of the intensity threshold. As we increase the threshold, the number of true detections decreases monotonically. In contrast, the number of false detections increases exponentially as the threshold tends to zero. The best performing values (in terms of true and false detections) are between 0.2 and 0.4 photons, coinciding with the reflectivity interval from 5% to 10%. This interval can be used as a guideline for setting $r_{\min}$. The execution time is not affected significantly by the threshold, as the complexity is mostly driven by the (fixed) number of photons.

The intensity update depends on the amount of filtering β in (5.11), which mostly impacts the

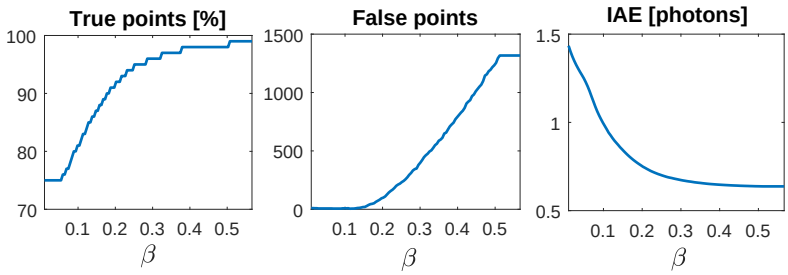


Figure 5.5 Effect of the amount of low-pass filtering on the reconstruction quality. Large values oversmooth the estimates, generating false detections and also incurring in a larger intensity error, whereas low values do not impose sufficient spatial correlation, reducing the number of true detections.

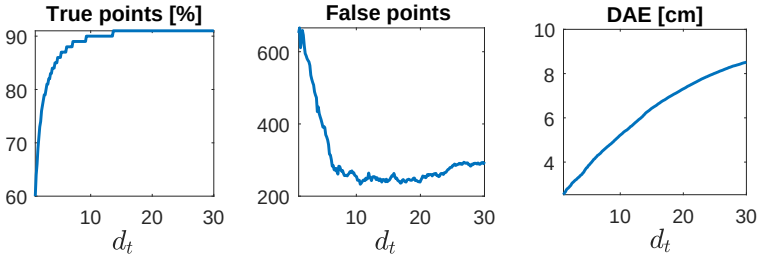


Figure 5.6 Effect of the APSS kernel size in the depth direction on the reconstruction quality. Low values fail to correlate neighbouring points, whereas large values oversmooth the depth estimates.

intensity estimation. Figure 5.5 shows the intensity absolute error (as defined in Section 3.6) as a function of $\beta \in [0, 1]$. Very small values of β mean negligible filtering, finding less points and resulting in a larger intensity error. Large values (close to 1) oversmooth the estimates, generating false detections and also resulting in a larger intensity error (this effect is reduced by the very smooth profile of a polystyrene head). Good values for β generally lie in the interval $[0.1, 0.3]$. Note that this interval might vary depending on the number of pixels of the array.

The depth update depends on d_t, the APSS kernel size in the depth direction. Figure 5.6 shows the impact of d_t in terms of true and false detections and mean depth absolute error (DAE). Small values of d_t result in poor reconstructions, as the kernel is too small to correlate neighbouring points, whereas large values oversmooth the depth estimates and may also mix different surfaces. The best choice lies around 8 and 10, which also has the physical meaning discussed above.

The background update depends on the hyperparameter $\lambda_{\tilde{\boldsymbol{b}}}$, which controls the degree of correlation between neighbouring background levels. Figure 5.7 shows the background estimation performance as a function of $\lambda_{\tilde{\boldsymbol{b}}}$ for the head without backplane dataset. While low values of $\lambda_{\tilde{\boldsymbol{b}}}$ do not impose sufficient correlation, large values of $\lambda_{\tilde{\boldsymbol{b}}}$ tend to oversmooth the estimates. While the best choices lie in the interval $[0.5, 2]$, the performance is not very sensitive to bad specifications of $\lambda_{\tilde{\boldsymbol{b}}}$.

While direct estimation of the hyperparameters from the observed data is not explored here, it would be interesting to study extensions of approximate message passing techniques [119, 120], methods based on the Stein unbiased risk estimator [121] (e.g., SUGAR [122]) or Bayesian tech-

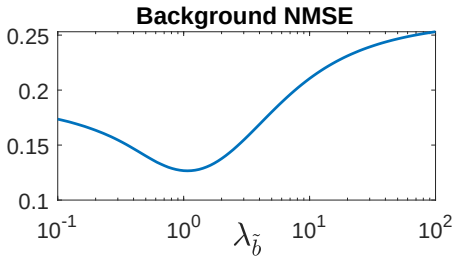


Figure 5.7 Effect of the amount of background regularisation $\lambda_{\tilde{b}}$ on the estimation of background levels.

niques [60] to this problem.

5.3 Parallel implementation

Pseudocode of the full algorithm is presented in Algorithm 9. The algorithm runs completely on a graphics processing unit (GPU), only exchanging the lidar waveforms and final output with the CPU. The parallel structures of the initialisation and main algorithm allow for efficient GPU implementation, as each parallel thread only requires the information of a local subset of photon measurements and 3D points. The number of iterations was fixed to $N_i = 50$, in order to have a similar execution time per lidar frame for real-time processing.

As the initialisation step processes every pixel independently, one parallel thread is executed per lidar pixel. For an initialisation with K surfaces per pixel, the general per-pixel complexity of the dense case is $\mathcal{O}(KT \log T)$, whereas the complexity of the sparse algorithm is $\mathcal{O}(KT_aT)$. If the half sample mode algorithm is used, the complexity of the initialisation reduces to $\mathcal{O}(T_a \log T_a)$, at the cost of finding at most one surface per pixel (during initialisation).

The gradient and denoising steps of the main algorithm have different parallel implementations. Each of the parallel threads processes one lidar waveform in the gradient steps of (5.4) and (5.5), as they can be processed independently of the rest due to the separable structure of the negative log-likelihood. The per-pixel complexity for the depth and log-intensity gradients is $\mathcal{O}(T_h)$ with T_h the number of non-zero bins in the compact support of the impulse response centred in the existing points ($T_h < T_a < T$), which is smaller than $\mathcal{O}(T \log T)$ needed for algorithms working on a dense intensity cube, especially when the number of histogram bins T is large. The background gradient step in (5.5) has a complexity of $\mathcal{O}(T_a)$. Both the APSS and intensity denoising steps run one thread per world-coordinates pixels, making use of the shared GPU memory (a gather operation [123]) to efficiently read the information of its neighbours. The main bottleneck for these steps is determined by the memory reads during the gather operation, which can be reduced by considering fewer neighbours at the cost of potentially degraded reconstruction. Note that RT3D has the same minimal memory requirements as ManiPoP. In contrast to alternatives that require the storage of a dense 3D cube of intensity estimates of size $\mathcal{O}(N_rN_cT)$, RT3D only stores

the estimated point cloud, generally of size $\mathcal{O}(N_r N_c)$.

The complexity of the algorithm is generally dominated by the gradient steps, which depend on the number of photons (active bins) per pixel. For example, the algorithm might run faster on a large array with few photon detections than a smaller array with densely populated histograms. To illustrate this, consider the execution times of the large raster-scan dataset (13 ms) and the Princeton Lightwave dataset (20 ms), shown in Table 5.1. While being significantly smaller, the $N_r = N_c = 32$ pixels array has dense histograms of 153 bins with non-zero counts. On the other hand, the $N_r = N_c = 141$ pixels raster scan dataset has a mean photon count of 3 photons per pixel, hence having approximately 3 active bins per pixel. Hence, the effective data size in the former case is $32 \times 32 \times 153 = 156672$, whereas in the latter is $141 \times 141 \times 3 \times 2 = 119286$ (where the last term in the multiplication is due to the bin number indicator in a sparse representation). The latter data size is smaller than the 32×32 array, hence the faster processing. Moreover, as the algorithm's complexity is driven by the amount of computation within a pixel, it is more intensive to process 153 bins than 4 active bins.

Algorithm 9 Real-time single-photon 3D imaging (RT3D)

Input: lidar waveforms $\boldsymbol{Z}$, number of iterations N_i, hyperparameters values and camera parameters Δ_b and Δ_p
Initialisation:
$s \leftarrow 0$
$(\boldsymbol{t}^0, \boldsymbol{m}^0, \tilde{\boldsymbol{b}}^0) \leftarrow$ Algorithm 6 (array) or Algorithm 7 (raster-scan)
Main loop:
while $s < N_i$ **do**
  $\boldsymbol{t}^{s+1} \leftarrow$ Point cloud denoising $\left(\boldsymbol{t}^s - \mu_t \nabla_{\boldsymbol{t}} g\left(\boldsymbol{t}^s, \boldsymbol{m}^s, \tilde{\boldsymbol{b}}^s\right)\right)$
  $\boldsymbol{m}^{s+1} \leftarrow$ Manifold denoising $\left(\boldsymbol{t}^s - \mu_m \nabla_{\boldsymbol{m}} g\left(\boldsymbol{t}^{s+1}, \boldsymbol{m}^s, \tilde{\boldsymbol{b}}^s\right)\right)$
  $\tilde{\boldsymbol{b}}^{s+1} \leftarrow \tilde{\boldsymbol{b}}^s - \mu_b \nabla_{\tilde{\boldsymbol{b}}} g\left(\boldsymbol{t}^{s+1}, \boldsymbol{m}^{s+1}, \tilde{\boldsymbol{b}}^s\right)$
  if the lidar system is mono-static **then**
    $\tilde{\boldsymbol{b}}^{s+1} \leftarrow$ Image denoising $\left(\tilde{\boldsymbol{b}}^{s+1}\right)$
  end if
  $s \leftarrow s + 1$
end while
Output: final estimates $(\boldsymbol{t}^{N_i-1}, \boldsymbol{m}^{N_i-1}, \tilde{\boldsymbol{b}}^{N_i-1})$

5.4 Beyond the APSS denoiser

We have focused on the APSS denoiser to target real-time performance, profiting from the parallel structure and closed-form updates. However, we could imagine other choices with different trade-offs between execution time, memory requirement and reconstruction quality [124]. For example, a straightforward alternative is the simple point set surface (SPSS) denoiser instead of APSS. The plug-and-play strategy provides a framework to incorporate different types of prior information, avoiding the need to develop specific algorithms for single-photon lidar. As explained in [100], APSS

only relies on a local surface smoothness prior, whereas more sophisticated denoisers exploit more complex prior knowledge about the scene's structure. If we want to capture non-local correlations between point cloud patches, we could use the denoiser in [125], which is based on a dictionary learning approach. Higher-level knowledge about the scene, such as the presence of buildings or humans could be also exploited through dedicated denoisers. For example, the algorithm in [126] uses planes to denoise point clouds of building facades, being adapted for remote sensing/outdoor applications. Finally, we could also profit from available 3D data using data-driven denoisers. In this direction, we can use algorithms that fit templates of possible objects [127] or profit from recent advances in graph convolutional neural networks [128], which are specially designed to handle point cloud structures [129, 130].

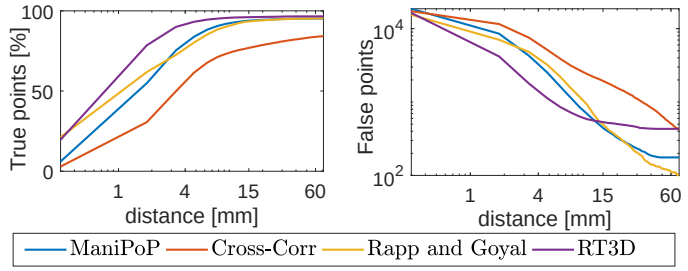
5.5 Results

First, we evaluate RT3D using 4 lidar datasets acquired with different raster-scanning systems, which have already been introduced in Section 2.5. Secondly, we demonstrate 50 frames per second reconstructions using the Princeton Lightwave lidar array camera, where the data is acquired in broad daylight from distances up to 320 metres. The raster-scanning datasets have a high spatial resolution (hundreds of pixels and thousands of histogram bins) and a low number of photon detections (less than 5 PPP), whereas lidar arrays record a large number of photons (more than 10 PPP) with low spatial resolution.

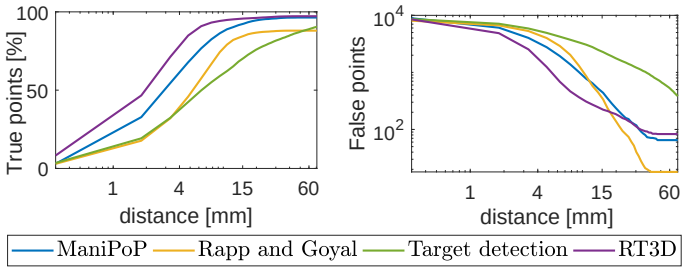
5.5.1 Raster-scanning results

We compare the reconstructions obtained with the standard cross-correlation, a state-of-the-art single-surface algorithm [25], 3 multi-depth reconstruction algorithms (SPISTA, ℓ_{21}+TV and ManiPoP), and a target detection algorithm [35]. We evaluated these algorithms using the raster-scanning datasets introduced in Section 2.5. Figures 5.9 to 5.11 show the 3D reconstructions obtained by the competing algorithms for each dataset, whereas their execution time are presented in Table 5.1. Figure 5.8 shows the percentage of true detections and number of false detections as a function of the maximum distance between a ground truth point and a reconstructed point.

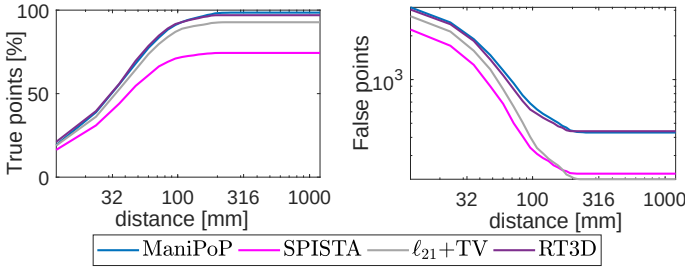
Figure 5.9 shows the results for the mannequin head with backplane dataset. Within a maximum error of 4 cm, RT3D finds 96.6% of the 3D points, improving the results of cross-correlation, which finds 83.46%, and also performing slightly better than the single-depth algorithm [25] and ManiPoP, which find 95.2% and 95.23%, respectively. The most significant difference is the processing time of each method: RT3D only takes 13 ms to process the entire frame, whereas ManiPoP and the single-surface algorithm require 201 s and 37 s, respectively. Whereas a parallel implementation of cross-correlation will almost always be faster than a regularised algorithm (requiring only 1 ms for this lidar frame), the execution time of RT3D only incurs a small overhead cost while



(a)



(b)



(c)

Figure 5.8 Number of true and false detections as a function of the maximum admissible distance between a ground truth point and a reconstructed one for (a) head with backplane (b) head without backplane and (c) mannequin behind scatterer.

	Head with backplane	Head without backplane	Human behind camouflage	Mannequin behind scatterer
Parallel cross-corr	1 ms	1 ms	NA	NA
SPISTA [36]	705 s	3362 s	1279 s (long. acq.) 1212 s (short acq.)	2871 s
ℓ_{21}+TV [37]	201 s	187 s	165 s (long. acq.) 182 s (short acq.)	202 s
Rapp and Goyal [25]	37 s	44 s	NA	NA
Altmann et al. [35]	12 h	12 h	NA	NA
ManiPoP	201 s	181 s	120 s (long. acq.) 102 s (short acq.)	146 s
RT3D	13 ms	11 ms	27 ms (long. acq.) 15 ms (short acq.)	40 ms

Table 5.1 Execution time of RT3D and other state-of-the-art 3D reconstruction algorithms. Some methods do not provide meaningful results in certain scenes. For such cases, the execution time is not available (NA). RT3D presents a higher computing time than a parallel implementation of the cross-correlation algorithm (which only applies in the presence of single peaks), but outperforms all the other reconstruction algorithms by a factor of about $\approx 10^5$. For all experiments, we used an i7-3.0 GHz desktop computer (16GB RAM) equipped with an NVIDIA Titan Xp GPU card and the codes provided by the authors of [25, 35–37].

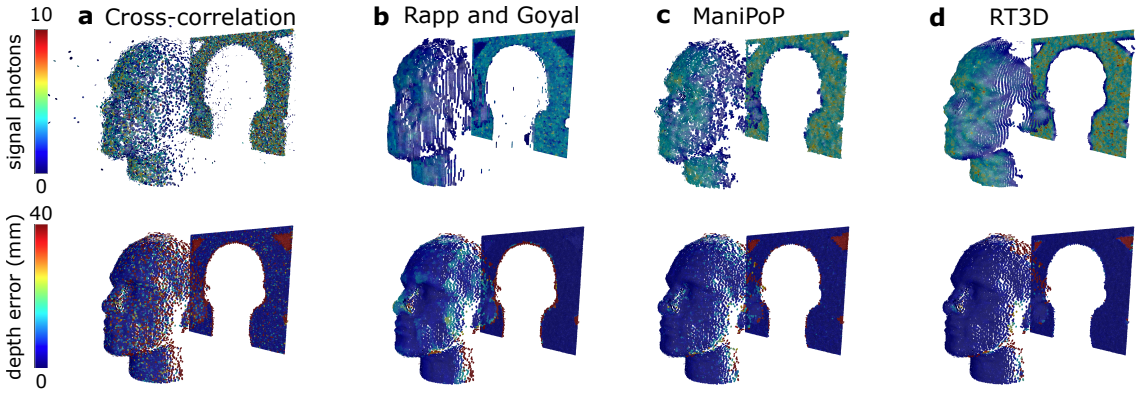


Figure 5.9 Comparison of 3D reconstruction methods. Reconstruction results of (a) cross-correlation, (b) Rapp and Goyal [25], (c) ManiPoP and (d) RT3D. The colour bar scale depicts the number of returned photons from the target assigned to each 3D point. Cross-correlation does not include any regularisation, yielding noisy estimates, whereas the results of Rapp and Goyal, ManiPoP and RT3D show structured point clouds. The method of Rapp and Goyal correlates the borders of the polystyrene head and the backplane (as it assumes a single surface per pixel), whereas ManiPoP and RT3D do not promote correlations between them.

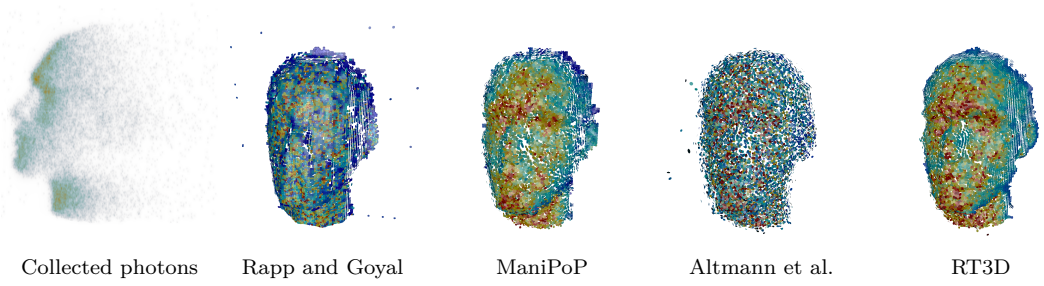


Figure 5.10 Comparison of 3D reconstructions achieved by RT3D and competing methods for the head without backplane scene. The colour scheme denotes the number of returned photons attributed to each 3D point.

significantly improving the reconstruction quality of single-photon data.

The head without backplane dataset presents at most one surface per pixel. In this case, if a single-surface per pixel algorithm [25] plus a thresholding step is used, the borders of the target are correlated with spurious detections in pixels without surfaces, yielding relatively poor estimates. The target detection algorithm of [35] takes into account the presence of pixels without any surfaces, but does not promote any correlation between detected points. Both RT3D and ManiPoP provide good results, correlating only points belonging to the target.

Figure 5.11 shows the mannequin behind scatterer and the human behind camouflage reconstructions obtained by SPISTA, ℓ_{21}+TV, ManiPoP and RT3D. Again, the best results are obtained by ManiPoP and RT3D. However, ManiPoP requires an execution time many orders of magnitude higher than RT3D.

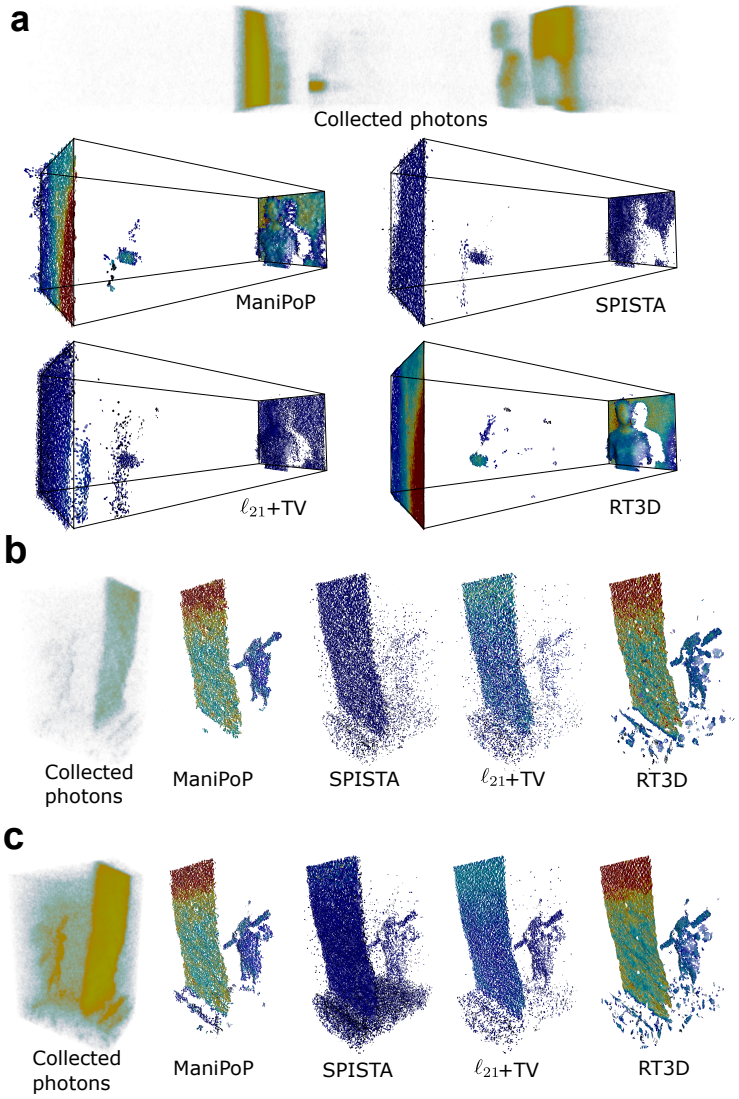


Figure 5.11 Comparison in the presence of multiple surfaces per pixel. Reconstructions achieved by RT3D and competing methods for the (**a**) mannequin behind scatterer and human behind camouflage datasets using acquisition times of (**b**) 0.32 ms and (**c**) 3.2 ms. In this scene, single-depth algorithms, including cross-correlation, cannot be applied, as they would only reconstruct the first object. In these cases, we evaluated SPISTA, ℓ_{21}+TV, ManiPoP and RT3D, which can handle multiple surfaces. The best results are obtained by ManiPoP and RT3D. However, ManiPoP requires an execution time many orders of magnitude higher than the novel method.

5.5.2 Lidar array results

To demonstrate the real-time processing capabilities of RT3D, we acquired, using the Kestrel Princeton Lightwave camera, a series of 3D videos with a single-photon array of $N_r = N_c = 32$ pixels and $T = 153$ histogram bins (binning resolution of 3.75 cm), which captures 150,400 binary frames per second. As the pixel resolution of this system is relatively low, we followed a super-resolution scheme, estimating a point cloud of $N_r = N_c = 96$ pixels. This can be easily achieved by defining a larger neighbourhood $\mathcal{N}_{i,j}$ in (5.3), mapping a window of 3×3 points in the finest

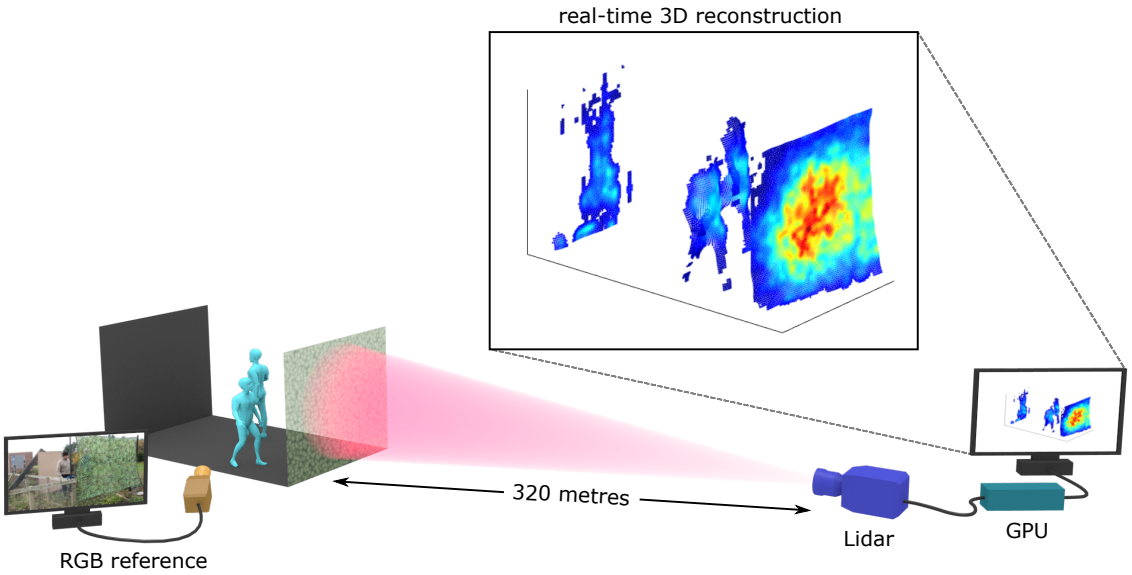


Figure 5.12 Schematic of the 3D imaging experiment. The scene consists of two people walking behind a camouflage net at a stand-off distance of 320 metres from the lidar system. An RGB camera was positioned a few metres from the 3D scene and used to acquire a reference video. The algorithm is able to provide real-time 3D reconstructions using a GPU. As the lidar presents only $N_r = N_c = 32$ pixels, the point cloud was estimated in a higher resolution of $N_r = N_c = 96$ pixels.

resolution (real-world coordinates) to a single pixel in the coarsest resolution (lidar coordinates). As processing a single lidar frame with the novel method takes 20 ms, we integrated the binary acquisitions into 50 lidar frames per second (i.e., real-time acquisition and reconstruction). At this frame rate, each lidar frame is composed of 3008 binary frames.

Figure 5.12 shows the imaging scenario, which consists of two people walking between a camouflage net and a backplane at a distance of approximately 320 metres from the lidar system. Each frame has approximately 900 photons per pixel, where 450 photons are due to target returns and the rest are related to dark counts or ambient illumination from solar background. Most pixels present two surfaces, except for those in the left and right borders of the camouflage, where there is only one return per pixel. A maximum number of 3 surfaces per pixel can be found in some parts of the contour of the human targets. The reconstruction videos can be found in youtube.com/watch?v=PzCcAoypUfMe.

Improvements by upsampling in small lidar arrays

Upsampling can bring additional details to the reconstructed objects, improving the estimates of naive upsampling in a post-processing step. Figure 5.13 shows the upsampled reconstructions with the RT3D method and cross-correlation. The cross-correlation output was upsampled by naively converting each detection into a 3×3 grid of points at the same depth. While the upsampled cross-correlation has a blocky appearance, RT3D captures additional details in the contours of the 3D target. Note that these contours are not always aligned with the coarse scale.

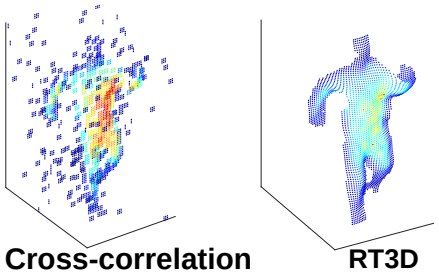


Figure 5.13 Comparison of 3D reconstructions of the 32×32 lidar array data by cross-correlation and RT3D. The upsampling strategy of RT3D brings additional details in the contours of the object, whereas a naive upsampling of the cross-correlation output presents a blocky appearance.

5.5.3 Operation boundary conditions

Finally, we study the performance of the algorithm as a function of PPP and SBR. We generate 100 synthetic lidar cubes for SBR values in $[0.01, 100]$ and mean photons per pixels in $[0.1, 100]$, using the ground truth point cloud, data cube size and impulse response from the head without backplane dataset. As a baseline, we compare with the standard cross-correlation algorithm. To account for the pixels without objects, we post-processed the output of cross-correlation by removing points below a normalised intensity of 10%. We consider the number of true and false detections, depth absolute error (only computed for true detections and reconstructions with more than 80% of detected points), intensity absolute error (normalised by the PPP to approximately lie between 0 and 1) and background NMSE. The results obtained are gathered in Fig. 5.14. RT3D performs well in a wider range of conditions, achieving reconstructions with ≈ 0.1 photons per pixel and up to an SBR of 0.01 (with 100 PPP or more). Cross-correlation generates many orders of magnitude more false detections than the new method. Interestingly, RT3D exhibits a sharper transition in the detection of true points, meaning that, for a given SBR, either none or most of the points will be found depending on the recorded photon count. It also achieves smaller depth and intensity absolute errors than cross-correlation in all conditions, as it exploits the manifold structure of the scene. Furthermore, it achieves a significantly smaller background NMSE, capturing the spatial correlation in the background image.

5.6 Extension to multispectral lidar

The RT3D algorithm can be easily extended to multispectral single-photon lidar by considering the extended observation model investigated in (3.2), where the negative log-likelihood function is

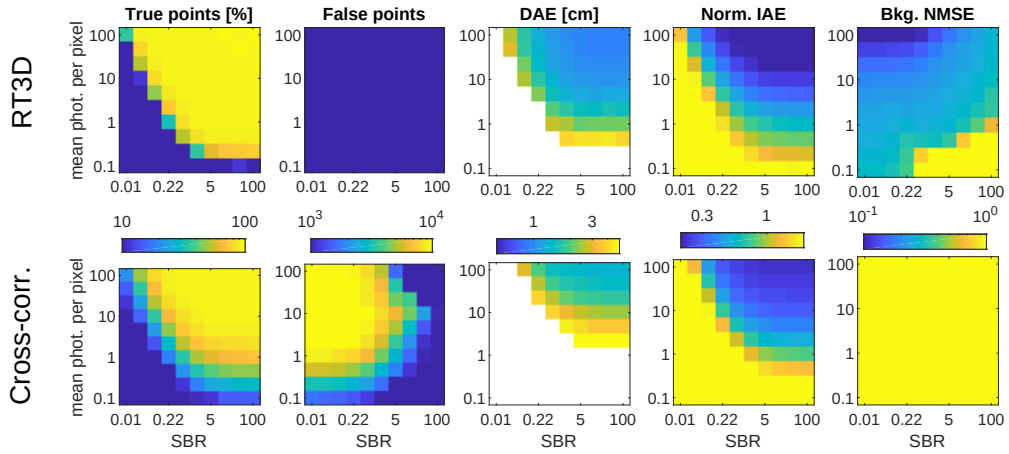


Figure 5.14 Comparison of RT3D and cross-correlation with thresholding in a target detection setting for different SBR and PPP. The depth absolute error is only displayed for reconstructions with more than 80% and is left blank otherwise.

$$g\left(\boldsymbol{t}, \boldsymbol{m}, \tilde{\boldsymbol{b}}\right) = \sum_{i=1}^{N_r}\sum_{j=1}^{N_c}\sum_{\ell=1}^{L}\sum_{t=1}^{T} g_{i,j,\ell} e^{\tilde{b}_{i,j,\ell}} + \sum_{\mathcal{N}_{i,j}} g_{i,j,\ell} e^{m_{n,\ell}} h_\ell(t - t_n) \ldots$$
$$- z_{i,j,\ell,t} \log\left(\sum_{\mathcal{N}_{i,j}} e^{m_{n,\ell}} h_\ell(t - t_n) + g_{i,j,\ell} e^{\tilde{b}_{i,j,\ell}}\right) \quad (5.18)$$

assuming that the impulse response $h_\ell(t)$ is known and only varies across wavelengths. The binary variables $g_{i,j,\ell}$ correspond to the coded apertures introduced in Section 3.5, where only W out of L wavelengths are measured per pixel.

Initialisation In order to reduce the complexity of the initialisation step, we first integrate the photons across wavelengths, i.e., $\bar{z}_{i,j,t} = \sum_{\ell=1}^{L} z_{i,j,\ell,t}$. If the impulse responses are aligned across wavelengths (i.e., $h_\ell(t)$ attain the maxima at the same histogram bin), we can apply the half sample mode (Algorithm 8), which does not require an analytical expression for $h(t)$. Both the intensity and background spectral vectors are initialised with the same value for all bands, obtained from the aggregated data.

Depth update The depth update remains as in the single-wavelength case, as detailed in Section 5.2.1.

Intensity update Here, instead of applying a simple low-pass filter as in the single-wavelength case, we use the bilateral filter [131], which profits from the additional colour information to preserve edges in the intensity profiles. Moreover, it presents a similar computational complexity to low-pass filtering when implemented in parallel (only a few neighbour intensity queries are

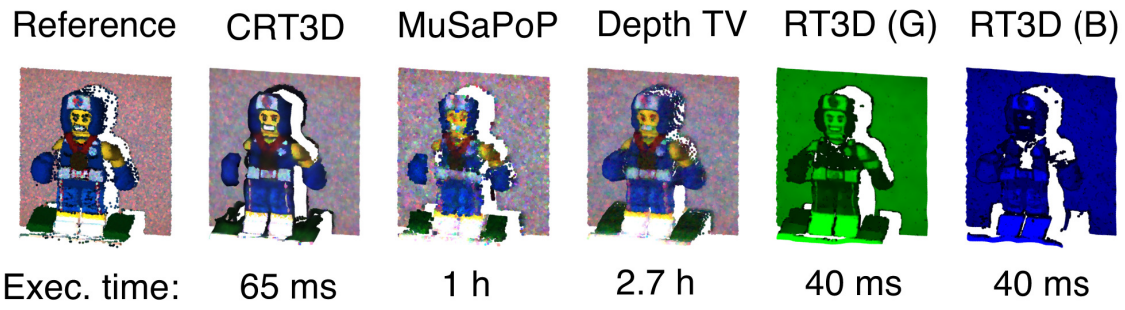


Figure 5.15 Ground truth point cloud and 3D reconstructions obtained by the competing methods for the dataset of strong ambient illumination and acquisition time of 1 ms. The execution time of each method is presented below the reconstruction.

required per input point). The bilateral filter is computed as

$$m_{n,\ell}^{s+1} = \frac{1}{C}\left(\beta m_{n,\ell}^{*} + \frac{1-\beta}{\#\,\mathcal{M}(m_{n,\ell}^{*})} \sum_{n' \in \mathcal{M}(m_{n,\ell}^{*})} m_{n',\ell}^{*} e^{-\frac{||\boldsymbol{m}_{n}^{*} - \boldsymbol{m}_{n'}^{*}||_2^2}{2\sigma^2}}\right) \tag{5.19}$$

where β plays the same role as in the single-wavelength case (5.11), σ^2 is another hyperparameter controlling the smoothness and C is a normalising constant, such that

$$C = \beta + \frac{1-\beta}{\#\,\mathcal{M}(m_{n,\ell}^{*})} \sum_{n' \in \mathcal{M}(m_{n,\ell}^{*})} e^{-\frac{||\boldsymbol{m}_{n}^{*} - \boldsymbol{m}_{n'}^{*}||_2^2}{2\sigma^2}}. \tag{5.20}$$

Background update As in the single-wavelength case, we choose a quadratic Laplacian regularisation for $\rho_b(\tilde{\boldsymbol{b}})$, applying a Wiener filter separately on each band (i.e., we assume prior independence across wavelengths), that is

$$\tilde{\boldsymbol{b}}_{\ell}^{s+1} \leftarrow (\boldsymbol{I} + \lambda_{\tilde{\boldsymbol{b}}_{\ell}} \mu_b \boldsymbol{P})^{-1} \tilde{\boldsymbol{b}}_{\ell}^{*} \tag{5.21}$$

separately for each wavelength $\ell = 1, \ldots, L$.

The resulting algorithm is referred to as CRT3D, as it can be interpreted as Colour extension of RT3D.

5.6.1 MSL Experiments

We evaluate CRT3D using the real MSL dataset introduced in [75], which has $N_r = N_c = 200$ pixels, $L = 4$ wavelengths (473, 532, 589 and 640 nm) and $T = 1029$ histogram bins. We used the coded apertures explained in Section 3.5 to choose only one wavelength $W = 1$ per pixel out of 4, yielding the same amount of data than a standard single-wavelength lidar dataset. The performance was assessed for two different acquisition times, 1 and 10 ms, and two different background illumination conditions, namely negligible background illumination and strong background illumi-

	low background			high background		
Backplane	yes	yes	no	yes	yes	no
Acq. time [ms]	10	1	1	10	1	1
PPP	101	10	2	100	10	5
SBR	80	80	22	2	2	1
Exec. time [ms]	170	51	35	251	65	41

Table 5.2 PPP, SBR and execution time of CRT3D for the evaluated datasets.

nation. Furthermore, as most of the pixels in the original scene consist of only one return, we created a new target detection dataset by removing all the returns linked to the backplane behind the lego figurine, only keeping background detections. Table 5.2 summarises the PPP and SBR of the evaluated datasets. A ground truth reference was obtained by applying the matching pursuit algorithm [117] on the most powerful wavelength (per pixel) in a very long acquisition time dataset (40 ms per pixel). The algorithm is compared with MuSaPoP, Depth TV [75] and RT3D using only 1 fully-sampled band. In the target detection case, we removed the points estimated by [75], which had a normalised reflectivity smaller than 0.1 (and provided the best results overall). We have not considered algorithms that rely on a dense multispectral intensity hypercube representation and use an ADMM algorithm [132], as they involve prohibitive memory requirements (e.g., more than 50 GB for the evaluated dataset). The performance was assessed using the metrics introduced in Section 3.6: the number of true and false detections found at a given distance τ and IAE for points found within a given distance τ of the ground truth. Figure 5.15 shows the 3D reconstructions for the low SBR, 1 ms acquisition time case, also including the reconstructions achieved by the single-wavelength algorithm RT3D, where the coded aperture is set to only acquire a specific wavelength (green or blue) across all the array. Depth TV provides good estimates of the figurine and backplane, but generates a false depth gradient in the contours of the objects due to the total variation regularisation. MuSaPoP does not suffer from this effect, but it provides noisier depth and intensity estimates. The reconstructions using only one wavelength miss some parts of the scene (e.g., the red collar in the green wavelength), as these surfaces have almost no returning photons at those wavelengths. CRT3D solves the aforementioned problems, separating well surfaces belonging to different objects and obtaining accurate estimates within each surface. Figure 5.16 shows true detections and IAE obtained by the different methods for the considered datasets. CRT3D performs better in terms of number of true points found. In terms of IAE, CRT3D obtains better results when considering points within a 5 cm error, but achieves a similar asymptotic IAE as Depth TV in the scenes with backplane, improving the IAE attained by MuSaPoP in all scenarios. Depth TV performs poorly when no backplane is present, as the convergence of the algorithm is degraded by pixels without surfaces and the thresholding step removes points of the figurine target. MuSaPoP obtains less false detections, followed by CRT3D in the scene without backplane and by Depth TV in the scenes with backplane. In terms of execution times, the competing methods require

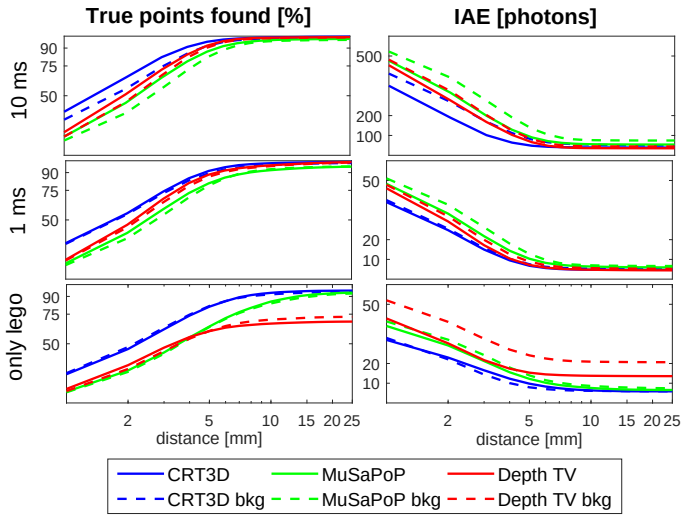


Figure 5.16 True detections and mean intensity absolute error for the evaluated datasets. Solid and dashed lines denote results obtained in the high and low SBR cases, respectively.

more than 1 hour per dataset (they are not easily parallelisable and rely on thousands of CPU sequential iterations), whereas the CRT3D has execution times of the order of few milliseconds (50 iterations) using a Titan Xp NVIDIA GPU card (see Table 5.2), being adapted for real-time sensing applications.

5.7 Conclusion

This chapter has introduced a real-time 3D reconstruction method that is able to obtain reliable estimates of distributed scenes using very few photons and/or in the presence of spurious detections. The resulting algorithm does not make any strong assumptions about the 3D surfaces to be reconstructed, allowing an unknown number of surfaces to be present in each pixel. We have demonstrated similar or better reconstruction quality than other existing methods, while improving the execution speed by a factor up to 10^5. The algorithm provides reliable real-time 3D reconstruction of scenes with multiple surfaces per pixel at long distance (320 m) and high frame rates (50 frames per second) in daylight conditions. Moreover, it can be easily implemented for general purpose graphical processing units [123], and thus is compatible with use in modern embedded systems (e.g., self-driving cars).

The method combines a priori information on the observation model (sensor statistics, dead pixels, sensitivity of the detectors, etc.) with powerful point cloud denoisers from the computer graphics literature, outperforming methods based solely on computer graphics or image processing techniques. Moreover, we have shown that the observation model can be easily modified to perform super-resolution. It is worth noting that the RT3D model could also be applied to other scenarios, e.g., involving spatial deblurring due to highly scattering media.

While we have chosen the APSS denoiser, the generality of our formulation allows us to use many point cloud (depth and intensity) and image (background) denoisers as building blocks to construct other variants. In this way, we can control the trade-off between reconstruction quality and computing speed (Section 5.4).

We have extended the single-wavelength method to MSL while keeping the real-time processing property. By using coded apertures, we reduce the amount of data to be acquired and processed. In the special case of $W = 1$ measured wavelength per pixel, the total amount of data is similar to the single-wavelength case. Hence, the MSL extension does not significantly increase the execution time, as the processing bottleneck is linked to the gradient steps, which have a complexity only affected by the number of photons per pixel.

Chapter 6

Conclusions and suggestions for future work

Contents

6.1 Conclusions

This thesis has presented multiple signal processing solutions to the single-photon lidar multi-depth imaging problem. Each solution offers different trade-offs in terms of a priori assumptions about the imaged scene, reconstruction quality, requirements on the number of signal photons, robustness to strong background illumination, execution time (complexity and convergence rate) and uncertainty quantification.

On one hand, the ManiPoP and MuSaPoP models provide good quality reconstructions using few photons, while also being able to compute uncertainty intervals on the estimated parameters (e.g., Figs. 2.20 and 2.23). On the other hand, RJ-MCMC inference for these models is an inherently slow process, which is not suited for real-time applications. The detection algorithms presented in Chapter 4 overcome the execution time bottleneck by profiting from a parallel model and inference scheme. While these methods can also quantify uncertainty, they impose a stronger assumption on the sensed scene than ManiPoP and MuSaPoP (at most one surface per pixel). The model based

on off-the-shelf point cloud denoisers presented in Chapter 5 relaxes this assumption without increasing execution time or losing in reconstruction quality, but does not provide meaningful uncertainty estimates.

The proposed methods also share many common ideas. The manifold models presented in Chapters 2, 3 and 5 have a number of parameters proportional to the number of points that define the manifold. They all promote correlations between points within a manifold, separating different surfaces by considering a minimum distance between points belonging to the same manifold ($d_{\min}$ in ManiPoP and MuSaPoP, and d_t in RT3D). Moreover, these models benefit from the manifold metrics to define correlations between the intensities of neighbouring points in the same surface.

Although the methods are presented in the context of single-photon lidar, their general formulation can be easily extended to other inverse problems involving 3D point clouds. As discussed throughout the thesis, we have not found standard tools from the image processing literature that can exploit efficiently existing correlations within manifolds[1]. In contrast, the methods presented herein can extract several surfaces from point clouds by exploiting the correlations within each surface.

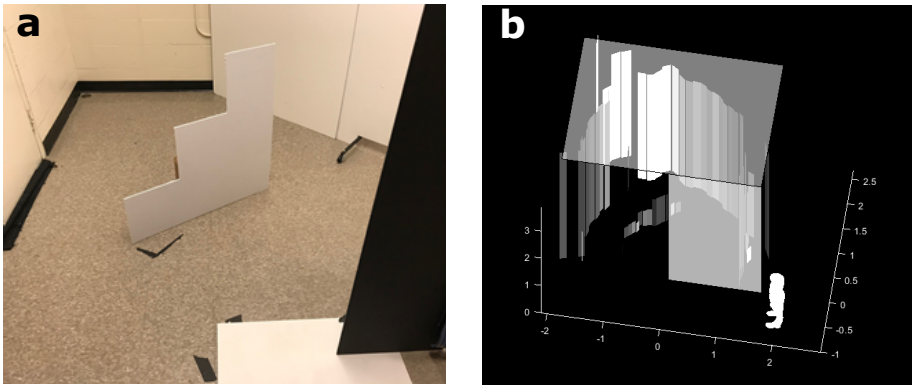


Figure 6.1 Non-line-of-sight imaging using the edge-resolved transient imager [**Rapp et al. 2019e**]. (a) RGB image of the hidden room. (b) Estimated contents using the SkellyPoP algorithm. The data is acquired from the position of the human figure in (b), where all the contents of the view are not in the line of sight of the camera.

The models presented in this thesis can be applied to inverse problems of the form

$$\boldsymbol{Z}|\boldsymbol{\Phi},\boldsymbol{\theta} \sim \mathcal{G}\left(f(\boldsymbol{\Phi},\boldsymbol{\theta})\right) \tag{6.1}$$

where $\mathcal{G}(\cdot)$ is related to the noise distribution (that can also depend on additional parameters), $\boldsymbol{\theta}$ is a vector of fixed dimension containing unknown model parameters (e.g., the background levels in the single-photon lidar problem) and $f(\cdot)$ is a function that maps the points of $\boldsymbol{\Phi}$ and additional parameters $\boldsymbol{\theta}$ into the measured data $\boldsymbol{Z}$. For example, in [**Rapp et al. 2019e**], we applied the

[1]We have not explored models based on curvelets and contourlets [133] in this thesis, which could also be useful for modelling surfaces. However, the resulting algorithms might not be as efficient as the methods presented here, as they would have similar or worse complexity than methods based on a cube of intensities (e.g., $\ell_{21} + TV$).

ManiPoP model to a non-line-of-sight imaging scenario. As illustrated in Fig. 6.1, the inverse problem consists of recovering the contents of a hidden room using measurements taken from the visible side. In this case, $\mathcal{G}(\cdot)$ is the Skellam distribution, the points in $\boldsymbol{\Phi}$ depict planar facets, which form 1D manifolds (walls or other pieces of furniture) in 2D space, $\boldsymbol{\theta}$ contains the unknown height and reflectivity of the ceiling, and f is a non-linear rendering function that maps the effect of hidden objects into the observed measurements.

6.2 Suggestions for future work

Single-photon lidar data raises multiple signal processing challenges. Despite the multi-depth tools presented in this thesis, there are many remaining challenges to be solved. These problems include the modelling of 3D point clouds for inverse problems which can be useful for various imaging modalities, some of them being described below.

6.2.1 Multi-depth imaging in turbulent media

Some important 3D imaging applications take place in highly scattering media. For example, long range scenes might present atmospheric turbulence, where the measurements suffer from spatial blurring. Underwater settings can also present similar phenomena. Hence, apart from the classical blurring along the depth axis (due to the timing electronics jitter), single-photon lidar data can exhibit blurring along the vertical and horizontal axes. An interesting direction of future work is to incorporate this scattering effect in the observation model, extending the algorithms presented here to account for the spatial blurring.

6.2.2 Compressive acquisition of lidar signals

In this thesis we have presented a subsampling scheme that reduces the necessary spectral measurements to recover multispectral 3D point clouds from multispectral lidar data. Compression along the vertical and horizontal axes (pixels) have also been proposed in single-photon single-pixel [102] and super-pixel [134] cameras. While some preliminary work [69] have proposed compression along the depth axis, this direction remains unexplored. Recent developments of lidar arrays [99] are limited by a memory transfer bottleneck, as these devices are not capable of outputting a stream of time-tagged detections at the rate they arrive to the detector. This limitation could be addressed with compressive learning techniques [135].

Ultimately, we seek a better understanding of the interplay between acquisition (photon budget, laser power, acquisition model), reconstruction complexity (memory requirements, parallel or serial architecture, execution time) and estimation performance.

6.2.3 Non-parametric detection of lidar signals

The detection methods presented in this thesis assume small deviations from the Poisson observation model. However, some lidar systems might present large variations due to imperfections of the timing electronics or high-flux conditions [30]. Non-parametric detection methods could be more robust to model misspecification by means of the kernel trick [136].

6.2.4 Inverse problems involving point clouds

Finally, it would be interesting to study the applicability of the proposed point cloud models to other inverse problems involving the recovery of an n-dimensional manifold from noisy and/or incomplete measurements. We identify two promising directions: sonar [137] and non-line-of-sight imaging inverse problems [138].

Sonar imaging systems attempt to recover a 2D manifold. Existing methods generally assume the presence of a single object per pixel to use image processing techniques, in a similar fashion to single-depth algorithms in the lidar problem. Hence, additional information could be recovered by using the methods described in this thesis.

Non-line-of-sight imaging systems [138] solve a tomography inverse problem, where the hidden 3D space is voxelised. Reconstruction methods [139] generally rely on a dense cube approach, as in SPISTA or ℓ_{21}+TV. Hence, it is likely that the improvements of ManiPoP and RT3D over the dense cube approaches in single-photon lidar would also appear in the non-line-of-sight setting.

Appendices

Appendix A

Marginal density of a gamma Markov random field

The marginal gamma Markov field joint density, $p(\boldsymbol{b}|\alpha_b)$, can be derived by integrating out the auxiliary variables $u_{i,j}$ from the complete joint density $p(\boldsymbol{u}, \boldsymbol{b}|\alpha_b)$, that is

$$p(\boldsymbol{b}|\alpha_{\boldsymbol{b}}) = \int_{\boldsymbol{u}} p(\boldsymbol{u}, \boldsymbol{b}|\alpha_b) d\boldsymbol{u} \tag{A.1}$$

For notation simplicity we replace the indices $i = 1, \ldots, N_r$ and $j = 1, \ldots, N_c$ for a unique linear index $n = 1, \ldots, N_r N_r$. The density $p(\boldsymbol{u}, \boldsymbol{b}|\alpha_b)$ can be expressed using Hammersley and Clifford theorem [55] as

$$p(\boldsymbol{b}, \boldsymbol{u}|\alpha_b) = \frac{1}{Z} \exp(\sum_{n=1}^{N_r N_c} -(\alpha_b + 1)\log(u_n) + (\alpha_b - 1)\log(b_n) - \sum_{n' \in \mathcal{M}_B(u_n)} \frac{4}{\alpha_b} \frac{b'_n}{u_n})$$

where Z is an intractable normalising constant. Then, (A.1) can be expressed as

$$\begin{aligned} p(\boldsymbol{b}|\alpha_b) &= \int_{\mathbb{R}_+^{N_r N_c}} p(\boldsymbol{b}, \boldsymbol{u}|\alpha_b) d\boldsymbol{u} \\ &\propto \prod_{n=1}^{N_r N_c} \left(\int_0^\infty u_n^{-(\alpha_b+1)} \prod_{n' \in \mathcal{M}_B(u_n)} e^{-\frac{4}{\alpha_b}\frac{b'_n}{u_n}} du_n \right) b_n^{\alpha_b - 1}. \end{aligned}$$

Considering that each integral has the following analytical solution

$$\int_0^\infty u_n^{-(\alpha_b+1)} \prod_{n' \in \mathcal{M}_B(u_n)} e^{-\frac{4}{\alpha_b}\frac{b'_n}{u_n}} du_n = \frac{\Gamma(\alpha_b)}{\left(\frac{4}{\alpha_b} \sum_{n' \in \mathcal{M}_B(u_n)} b'_n\right)^{\alpha_b}}$$

then

$$p(\boldsymbol{b}|\alpha_b) \propto \prod_{n=1}^{N_r N_c} b_n^{\alpha_b - 1} \frac{\Gamma(\alpha_b)}{\left(\frac{4}{\alpha_b} \sum_{n' \in \mathcal{M}_B(u_n)} b'_n\right)^{\alpha_b}} \tag{A.2}$$

$$\propto \prod_{n=1}^{N_r N_c} \frac{b_n^{\alpha_b - 1}}{\tilde{b}_n^{\alpha_b}} \tag{A.3}$$

where $\tilde{b}_n = \frac{4}{\alpha_b} \sum_{n' \in \mathcal{M}_B(u_n)} b'_n$ is a low-pass filtered version of b_n.

Appendix B

ManiPoP acceptance ratios

The birth move of point $(\boldsymbol{c}_{N_\Phi+1}, m_{N_\Phi+1})$ has an acceptance ratio given by $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$ with

$$r(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_1 & \text{if } |t_{N_\Phi+1} - t_n| > d_{max} \quad \forall n \neq N_\Phi + 1 : \\ & x_n = x_{N_\Phi+1} \text{ and } y_n = y_{N_\Phi+1} \\ 0 & \text{otherwise} \end{cases}$$

where C_1 is

$$C_1 = \prod_{t=1}^{T} \left(\frac{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m'_n} h(t - t'_n) + b'_{i,j}}{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m_n} h(t - t_n) + b_{i,j}} \right)^{z_{i,j,t}} \frac{p_{\text{death}}}{p_{\text{birth}}}$$
$$\lambda_a \gamma_a^{-m\left(S(\boldsymbol{c}_{N_\Phi+1}) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_{N_\Phi+1})} S(\boldsymbol{c}_{n'})\right)} \frac{1}{N_\Phi + 1}$$
$$\exp\left(-\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_{N_\Phi+1} - m_{n'})^2}{d(\boldsymbol{c}_{N_\Phi+1}; \boldsymbol{c}_{n'})} + {m_{N_\Phi+1}}^2 \beta \right) \right) \sqrt{\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}} \sqrt{\frac{1}{2\pi\sigma^2}}$$
$$\prod_{(i,j) \in \mathcal{M}_B(b_{i,j})} \left(\frac{b'_{i,j}}{b_{i,j}} \right)^{\alpha_b - 1} \left(\frac{\sum_{(i',j') \in \mathcal{M}_B(b_{i,j})} b_{i',j'}}{\sum_{(i',j') \in \mathcal{M}_B(b_{i,j})} b'_{i',j'}} \right)^{\alpha_b} \frac{1}{1-u}.$$

Similarly, the death move is accepted with probability $\rho = \min\{1, C_1^{-1}\}$, where the term $\frac{1}{N_\Phi+1}$ in the second line is replaced by $\frac{1}{N_\Phi}$. The dilation move of point $(\boldsymbol{c}_{N_\Phi+1}, m_{N_\Phi+1})$ is accepted with probability $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$ with

$$r(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_2 & \text{if } |t_{N_\Phi+1} - t_n| > d_{max} \quad \forall n \neq N_\Phi + 1 : \\ & x_n = x_{N_\Phi+1} \text{ and } y_n = y_{N_\Phi+1} \\ 0 & \text{otherwise} \end{cases}$$

where C_2 is

$$C_2 = \prod_{t=1}^{T} \left(\frac{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m'_n} h(t-t'_n) + b'_{i,j}}{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m_n} h(t-t_n) + b_{i,j}} \right)^{z_{i,j,t}} \frac{p_{\text{erosion}}}{p_{\text{dilation}}}$$
$$\lambda_a \gamma_a^{-m\left(S(\boldsymbol{c}_{N_\Phi+1}) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_{N_\Phi+1})} S(\boldsymbol{c}_{n'})\right)}$$
$$\frac{N_\Phi(2N_b+1)}{\sum_{m \in \mathcal{M}_{pp}(\boldsymbol{c}_{N_\Phi+1})} \#\,\mathcal{M}_{pp}(\boldsymbol{c}_m)} \times \frac{1}{\sum_{m=1}^{N_\Phi+1} \mathbb{1}_{\mathbb{Z}_+}(\#\,\mathcal{M}_{pp}(\boldsymbol{c}_m))}$$
$$\exp\left(-\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_{N_\Phi+1} - m_{n'})^2}{d(\boldsymbol{c}_{N_\Phi+1}; \boldsymbol{c}_{n'})} + {m_{N_\Phi+1}}^2 \beta \right) \right) \sqrt{\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}} \sqrt{\frac{1}{2\pi\sigma^2}}$$
$$\prod_{(i,j) \in \mathcal{M}_B(b_{i,j})} \left(\frac{{b'}_{i,j}}{b_{i,j}} \right)^{\alpha_b - 1} \left(\frac{\sum_{(i',j') \in \mathcal{M}_B(b_{i,j})} b_{i',j'}}{\sum_{(i',j') \in \mathcal{M}_B(b_{i,j})} {b'}_{i',j'}} \right)^{\alpha_b}.$$

A shift of the point $(\boldsymbol{c}_n, m_n)$ to the new position $\boldsymbol{c'}_n = (x_n, y_n, t'_n)^T$, has an acceptance probability of $\rho = \min\{1, r\,(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$ with

$$r\,(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_3 & \text{if } |t'_n - t_m| > d_{max} \quad \forall n \neq m : \\ & x_m = x_n \text{ and } y_m = y_n \\ 0 & \text{otherwise} \end{cases}$$

where

$$C_3 = \prod_{t=1}^{T} \left(\frac{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m'_n} h(t-t'_n) + b'_{i,j}}{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m_n} h(t-t_n) + b_{i,j}} \right)^{z_{i,j,t}}$$
$$\exp\left(-\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c'}_n)} \frac{(m_n - m_{n'})^2}{d(\boldsymbol{c'}_n; \boldsymbol{c}_{n'})} \right) \right)$$
$$\exp\left(\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_n - m_{n'})^2}{d(\boldsymbol{c}_n; \boldsymbol{c}_{n'})} \right) \right) \sqrt{\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}}$$
$$\gamma_a^{-m\left(S(\boldsymbol{c'}_n) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c'}_n)} S(\boldsymbol{c}_{n'})\right) + m\left(S(\boldsymbol{c}_n) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} S(\boldsymbol{c}_{n'})\right)}.$$

A mark update of point $(\boldsymbol{c}_n, m_n)$ to a new reflectivity $r'_n = \log(m'_n)$, is accepted with probability $\rho = \min\{1, C_4\}$, where

$$C_4 = \prod_{t=1}^{T} \left(\frac{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m'_n} h(t-t'_n) + b'_{i,j}}{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m_n} h(t-t_n) + b_{i,j}} \right)^{z_{i,j,t}} \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c'}_n)} \frac{(m'_n - m_{n'})^2}{d(\boldsymbol{c'}_n; \boldsymbol{c}_{n'})} + {m'_n}^2 \beta \right) \right) \exp\left(\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_n - m_{n'})^2}{d(\boldsymbol{c}_n; \boldsymbol{c}_{n'})} + m_n^2 \beta \right) \right).$$

The split move from $(\boldsymbol{c}_n = (x_n, y_n, t_n)^T, m_n)$ to $(\boldsymbol{c'}_{k_1} = (x_n, y_n, t'_{k_1})^T, m'_{k_1})$ and $(\boldsymbol{c'}_{k_2} = (x_n, y_n, t'_{k_2})^T, m'_{k_2})$ is accepted with probability $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$, where

$$r(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_5 & \text{if } |t'_n - t_m| > d_{max} \quad \forall n \neq m : \\ & x_m = x_n \text{ and } y_m = y_n \\ 0 & \text{otherwise} \end{cases}$$

and

$$C_5 = \prod_{t=1}^{T} \left(\frac{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m'_n} h(t-t'_n) + b'_{i,j}}{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m_n} h(t-t_n) + b_{i,j}} \right)^{z_{i,j,t}} \frac{p_{\text{merge}}}{p_{\text{split}}} \frac{1}{u(1-u)} N_\Phi(\#\text{ points in } \boldsymbol{\Phi} \text{ that verify (2.34)})^{-1} \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c'}_{k_1})} \frac{(m_{k_1} - m_{n'})^2}{d(\boldsymbol{c'}_{k_1}; \boldsymbol{c}_{n'})} \right) \right) \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c'}_{k2})} \frac{(m_{k_1} - m_{n'})^2}{d(\boldsymbol{c'}_{k_2}; \boldsymbol{c}_{n'})} \right) \right) \exp\left(\frac{1}{2\sigma^2} \left(\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_n - m_{n'})^2}{d(\boldsymbol{c}_n; \boldsymbol{c}_{n'})} \right) \right) \sqrt{\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}} \gamma_a^{-m\left(S(\boldsymbol{c'}_{k_1}) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c'}_{k_1})} S(\boldsymbol{c}_{n'})\right) + m\left(S(\boldsymbol{c}_n) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} S(\boldsymbol{c}_{n'})\right)} \lambda_a \gamma_a^{-m\left(S(\boldsymbol{c'}_{k_2}) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c'}_{k_2})} S(\boldsymbol{c}_{n'})\right)} (d_{max} + \text{length}_{h(t)}).$$

Finally, the merge move is accepted with probability $\rho = \min\{1, C_5^{-1}\}$.

Appendix C

MuSaPoP acceptance ratios

The birth move of point $(\boldsymbol{c}_{N_\Phi+1}, \boldsymbol{m}_{N_\Phi+1})$ has an acceptance ratio given by $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$ with

$$r(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_1 & \text{if } |t_{N_\Phi+1} - t_n| > d_{\min} \quad \forall n \neq N_\Phi + 1 : \\ & x_n = x_{N_\Phi+1} \text{ and } y_n = y_{N_\Phi+1} \\ 0 & \text{otherwise} \end{cases}$$

where C_1 is defined as

$$C_1 = \prod_{\ell=1}^{L}\prod_{t=1}^{T}\left(\frac{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m'_{n,\ell}} h_\ell(t-t'_n) + b'_{i,j,\ell}}{\sum_{\substack{n:x_n=i \\ y_n=j}} e^{m_{n,\ell}} h_\ell(t-t_n) + b_{i,j,\ell}}\right)^{z_{i,j,t,\ell}} \frac{p_{\text{death}}}{p_{\text{birth}}}$$
$$\lambda_a \gamma_a^{-m\left(S(\boldsymbol{c}_{N_\Phi+1}) \setminus \bigcup_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_{N_\Phi+1})} S(\boldsymbol{c}_{n'})\right)} \frac{1}{N_\Phi + 1} \left(\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}\frac{1}{2\pi\sigma^2}\right)^{\frac{L}{2}}$$
$$\prod_{\ell=1}^{L} \exp\left(-\sum_{n' \in \mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_{N_\Phi+1,\ell} - m_{n',\ell})^2}{2\sigma^2 d(\boldsymbol{c}_{N_\Phi+1}; \boldsymbol{c}_{n'})} - \frac{{m_{N_\Phi+1,\ell}}^2 \beta}{2\sigma^2}\right)$$
$$(1-u)^{-L} \prod_{\ell=1}^{L} \exp\left(g_{i,j,\ell} e^{m_{N_\Phi+1,\ell}} (1 - {w_\ell}^{-1}) \left(\sum_{t=1}^{T} h_\ell(t)\right)\right)$$
$$\prod_{(i,j) \in \mathcal{M}_B(b_{i,j})} \prod_{\ell=1}^{L} \left(\frac{{b'}_{i,j,\ell}}{b_{i,j,\ell}}\right)^{k_{i,j,\ell}-1} \exp\left(\frac{b_{i,j,\ell} - {b'}_{i,j,\ell}}{\theta_{i,j,\ell}}\right).$$

Similarly, the death move is accepted with probability $\rho = \min\{1, C_1^{-1}\}$, where the term $\frac{1}{N_\Phi+1}$ in the second line is replaced by $\frac{1}{N_\Phi}$. The dilation move of point $(\boldsymbol{c}_{N_\Phi+1}, \boldsymbol{m}_{N_\Phi+1})$ is accepted with probability $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$ with

$$r(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_2 & \text{if } |t_{N_\Phi+1} - t_n| > d_{\min} \quad \forall n \neq N_\Phi + 1 \\ & x_n = x_{N_\Phi+1} \text{ and } y_n = y_{N_\Phi+1} \\ 0 & \text{otherwise} \end{cases}$$

where C_2 is defined as

$$C_2 = \prod_{\ell=1}^{L}\prod_{t=1}^{T}\left(\frac{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m'_{n,\ell}} h_\ell(t-t'_n) + b'_{i,j,\ell}}{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m_{n,\ell}} h_\ell(t-t_n) + b_{i,j,\ell}}\right)^{z_{i,j,t,\ell}} \frac{p_{\text{erosion}}}{p_{\text{dilation}}} \lambda_a \gamma_a^{-m\left(S(\boldsymbol{c}_{N_\Phi+1})\setminus\bigcup_{n'\in\mathcal{M}_{pp}(\boldsymbol{c}_{N_\Phi+1})} S(\boldsymbol{c}_{n'})\right)} \left(\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}\frac{1}{2\pi\sigma^2}\right)^{\frac{L}{2}} \frac{N_\Phi(2N_b+1)}{\sum_{m\in\mathcal{M}_{pp}(\boldsymbol{c}_{N_\Phi+1})} \#\,\mathcal{M}_{pp}(\boldsymbol{c}_m)} \times \frac{1}{\sum_{m=1}^{N_\Phi+1} \mathbb{1}_{\mathbb{Z}_+}(\#\,\mathcal{M}_{pp}(\boldsymbol{c}_m))} \prod_{\ell=1}^{L}\exp\left(-\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_{N_\Phi+1,\ell}-m_{n',\ell})^2}{2\sigma^2 d(\boldsymbol{c}_{N_\Phi+1};\boldsymbol{c}_{n'})} - \frac{{m_{N_\Phi+1,\ell}}^2\beta}{2\sigma^2}\right) (1-u)^{-L}\prod_{\ell=1}^{L}\exp\left(g_{i,j,\ell}e^{m_{N_\Phi+1,\ell}}(1-{w_\ell}^{-1})\left(\sum_{t=1}^{T} h_\ell(t)\right)\right) \prod_{(i,j)\in\mathcal{M}_B(b_{i,j})}\prod_{\ell=1}^{L}\left(\frac{{b'}_{i,j,\ell}}{b_{i,j,\ell}}\right)^{k_{i,j,\ell}-1}\exp\left(\frac{b_{i,j,\ell}-{b'}_{i,j,\ell}}{\theta_{i,j,\ell}}\right).$$

A shift of the point $(\boldsymbol{c}_n, \boldsymbol{m}_n)$ to the new position $\boldsymbol{c}'_n = [x_n, y_n, t'_n]^T$ has an acceptance probability of $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$ with

$$r(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_3 & \text{if } |t'_n - t_m| > d_{\min} \quad \forall n \neq m \\ & x_m = x_n \text{ and } y_m = y_n \\ 0 & \text{otherwise} \end{cases}$$

where

$$C_3 = \prod_{\ell=1}^{L}\prod_{t=1}^{T}\left(\frac{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m'_{n,\ell}} h_\ell(t-t'_n) + b'_{i,j,\ell}}{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m_{n,\ell}} h_\ell(t-t_n) + b_{i,j,\ell}}\right)^{z_{i,j,t,\ell}} \prod_{\ell=1}^{L}\exp\left(-\frac{1}{2\sigma^2}\left(\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c'}_n)} \frac{(m_{n,\ell}-m_{n',\ell})^2}{d(\boldsymbol{c'}_n;\boldsymbol{c}_{n'})}\right)\right) \left(\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}\right)^{\frac{L}{2}} \prod_{\ell=1}^{L}\exp\left(\frac{1}{2\sigma^2}\left(\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_{n,\ell}-m_{n',\ell})^2}{d(\boldsymbol{c}_n;\boldsymbol{c}_{n'})}\right)\right) \gamma_a^{-m\left(S(\boldsymbol{c'}_n)\setminus\bigcup_{n'\in\mathcal{M}_{pp}(\boldsymbol{c'}_n)} S(\boldsymbol{c}_{n'})\right)+m\left(S(\boldsymbol{c}_n)\setminus\bigcup_{n'\in\mathcal{M}_{pp}(\boldsymbol{c}_n)} S(\boldsymbol{c}_{n'})\right)}.$$

A mark update randomly picks a point $(\boldsymbol{c}_n, \boldsymbol{m}_n)$ and proposes a new spectral signature $\boldsymbol{m}'_n$. Each

spectral log-intensity is accepted independently with probability $\rho = \min\{1, C_4\}$, where

$$C_4 = \prod_{\ell=1}^{L}\prod_{t=1}^{T}\left(\frac{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m'_{n,\ell}} h_\ell(t-t'_n) + b'_{i,j,\ell}}{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m_{n,\ell}} h_\ell(t-t_n) + b_{i,j,\ell}}\right)^{z_{i,j,t,\ell}}$$
$$\exp\left(-\frac{1}{2\sigma^2}\left(\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c'}_n)} \frac{(m'_{n,\ell}-m_{n',\ell})^2}{d(\boldsymbol{c'}_n;\boldsymbol{c}_{n'})} + {m'_{n,\ell}}^2\beta\right)\right)$$
$$\exp\left(\frac{1}{2\sigma^2}\left(\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_{n,\ell}-m_{n',\ell})^2}{d(\boldsymbol{c}_n;\boldsymbol{c}_{n'})} + m_{n,\ell}^2\beta\right)\right)$$
$$\prod_{\ell=1}^{L}\exp\left(g_{i,j,\ell}(e^{m_{n,\ell}} - e^{m'_{n,\ell}})(1-{w_\ell}^{-1})\left(\sum_{t=1}^{T} h_\ell(t)\right)\right).$$

The split move from $(\boldsymbol{c}_n = [x_n, y_n, t_n]^T, \boldsymbol{m}_n)$ to $(\boldsymbol{c'}_{k_1} = [x_n, y_n, t'_{k_1}]^T, \boldsymbol{m}'_{k_1})$ and $(\boldsymbol{c'}_{k_2} = [x_n, y_n, t'_{k_2}]^T, \boldsymbol{m}'_{k_2})$ is accepted with probability $\rho = \min\{1, r(\boldsymbol{\theta}, \boldsymbol{\theta'})\}$, where

$$r(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \begin{cases} C_5 & \text{if } |t'_n - t_m| > d_{\min} \quad \forall n \neq m : \\ & x_m = x_n \text{ and } y_m = y_n \\ 0 & \text{otherwise} \end{cases}$$

and

$$C_5 = \prod_{\ell=1}^{L}\prod_{t=1}^{T}\left(\frac{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m'_{n,\ell}} h_\ell(t-t'_n) + b'_{i,j,\ell}}{\sum_{\substack{n:x_n=i\\ y_n=j}} e^{m_{n,\ell}} h_\ell(t-t_n) + b_{i,j,\ell}}\right)^{z_{i,j,t,\ell}}$$
$$(u(1-u))^{-L} N_\Phi(\#\text{ points in } \boldsymbol{\Phi} \text{ that verify (3.27)})^{-1}\left(\frac{|\boldsymbol{P'}|}{|\boldsymbol{P}|}\right)^{\frac{L}{2}}$$
$$\prod_{\ell=1}^{L}\exp\left(-\frac{1}{2\sigma^2}\left(\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c'}_{k_1})} \frac{(m_{k_1,\ell}-m_{n',\ell})^2}{d(\boldsymbol{c'}_{k_1};\boldsymbol{c}_{n'})}\right)\right)$$
$$\prod_{\ell=1}^{L}\exp\left(-\frac{1}{2\sigma^2}\left(\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c'}_{k2})} \frac{(m_{k_1,\ell}-m_{n',\ell})^2}{d(\boldsymbol{c'}_{k_2};\boldsymbol{c}_{n'})}\right)\right)$$
$$\prod_{\ell=1}^{L}\exp\left(\frac{1}{2\sigma^2}\left(\sum_{n'\in\mathcal{M}_{pp}(\boldsymbol{c}_n)} \frac{(m_{n,\ell}-m_{n',\ell})^2}{d(\boldsymbol{c}_n;\boldsymbol{c}_{n'})}\right)\right)$$
$$\gamma_a^{-m\left(S(\boldsymbol{c'}_{k_1})\setminus\bigcup_{n'\in\mathcal{M}_{pp}(\boldsymbol{c'}_{k_1})} S(\boldsymbol{c}_{n'})\right)+m\left(S(\boldsymbol{c}_n)\setminus\bigcup_{n'\in\mathcal{M}_{pp}(\boldsymbol{c}_n)} S(\boldsymbol{c}_{n'})\right)}$$
$$\lambda_a\gamma_a^{-m\left(S(\boldsymbol{c'}_{k_2})\setminus\bigcup_{n'\in\mathcal{M}_{pp}(\boldsymbol{c'}_{k_2})} S(\boldsymbol{c}_{n'})\right)} 2(d_{\min}+L_h)\frac{p_{\text{merge}}}{p_{\text{split}}}.$$

Finally, the merge move is accepted with probability $\rho = \min\{1, C_5^{-1}\}$.

Bibliography

[1] J. Hecht, "Lidar for self-driving cars," *Optics and Photonics News*, vol. 29, no. 1, pp. 26–33, 2018.

[2] C. Mallet and F. Bretar, "Full-waveform topographic lidar: State-of-the-art," *ISPRS J. Photogrammetry Remote Sens.*, vol. 64, no. 1, pp. 1–16, 2009.

[3] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, "An overview of depth cameras and range scanners based on time-of-flight technologies," *Machine Vision and Applications*, vol. 27, no. 7, pp. 1005–1020, Oct 2016.

[4] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annual ACM Symposium on User Interface Software and Technology*, Santa Barbara, USA, 2011, pp. 559–568.

[5] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, USA, 2003.

[6] A. M. Pawlikowska, A. Halimi, R. A. Lamb, and G. S. Buller, "Single-photon three-dimensional imaging at up to 10 kilometers range," *Opt. Express*, vol. 25, no. 10, pp. 11 919–11 931, May 2017.

[7] R. Tobin, A. Halimi, A. McCarthy, M. Laurenzis, F. Christnacher, and G. S. Buller, "Three-dimensional single-photon imaging through obscurants," *Opt. Express*, vol. 27, no. 4, pp. 4590–4611, Feb 2019.

[8] R. Tobin, A. Halimi, A. McCarthy, X. Ren, K. J. McEwan, S. McLaughlin, and G. S. Buller, "Long-range depth profiling of camouflaged targets using single-photon detection," *Opt. Eng.*, vol. 57, no. 3, pp. 1 – 10, 2017.

[9] A. Maccarone, A. McCarthy, X. Ren, R. E. Warburton, A. M. Wallace, J. Moffat, Y. Petillot, and G. S. Buller, "Underwater depth imaging using time-correlated single-photon counting," *Opt. Express*, vol. 23, no. 26, pp. 33 911–33 926, Dec 2015.

[10] A. Halimi, Y. Altmann, A. McCarthy, X. Ren, R. Tobin, G. S. Buller, and S. McLaughlin, "Restoration of intensity and depth images constructed using sparse single-photon data," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 86–90.

[11] A. McCarthy, R. J. Collins, N. J. Krichel, V. Fernández, A. M. Wallace, and G. S. Buller, "Long-range time-of-flight scanning sensor based on high-speed time-correlated single-photon counting," *Appl. Opt.*, vol. 48, no. 32, pp. 6241–6251, 2009.

[12] D. Shin, F. Xu, D. Venkatraman, R. Lussana, F. Villa, F. Zappa, V. K. Goyal, F. N. Wong, and J. H. Shapiro, "Photon-efficient imaging with a single-photon camera," *Nat. Commun.*, vol. 7, p. 12046, 2016.

[13] D. B. Lindell, M. O'Toole, and G. Wetzstein, "Towards transient imaging at interactive rates with single-photon detectors," in *Proc. Int. Conf. Comput. Photography (ICCP)*, Pittsburgh, USA, May 2018, pp. 1–8.

[14] A. Kirmani, D. Venkatraman, D. Shin, A. Colaço, F. N. Wong, J. H. Shapiro, and V. K. Goyal, "First-photon imaging," *Science*, vol. 343, no. 6166, pp. 58–61, 2014.

[15] D. Shin, A. Kirmani, V. K. Goyal, and J. H. Shapiro, "Photon-efficient computational 3-D and reflectivity imaging with single-photon detectors," *IEEE Trans. Comput. Imag.*, vol. 1, no. 2, pp. 112–125, 2015.

[16] Y. Altmann, X. Ren, A. McCarthy, G. S. Buller, and S. McLaughlin, "Lidar waveform-based analysis of depth images constructed using sparse single-photon data," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 1935–1946, 2016.

[17] A. Wallace, C. Nichol, and I. Woodhouse, "Recovery of forest canopy parameters by inversion of multispectral lidar data," *Remote Sensing*, vol. 4, no. 2, pp. 509–531, 2012.

[18] E. S. Douglas, J. Martel, Z. Li, G. Howe, K. Hewawasam, R. A. Marshall, C. L. Schaaf, T. A. Cook, G. J. Newnham, A. Strahler, and S. Chakrabarti, "Finding leaves in the forest: The dual-wavelength echidna lidar," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 4, pp. 776–780, April 2015.

[19] G. Satat, M. Tancik, and R. Raskar, "Towards photography through realistic fog," in *Proc. Int. Conf. Comput. Photography (ICCP)*, Pittsburgh, USA, May 2018, pp. 1–10.

[20] C. Robert, *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, New York, USA, 2007.

[21] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov chain Monte Carlo*. CRC press, Boca Raton, USA, 2011.

[22] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. Conf. Uncertainty in Artificial Intelligence*, Seattle, Washington, USA, Aug. 2001, pp. 362–369.

[23] M. J. Beal, *Variational algorithms for approximate Bayesian inference*. PhD Thesis, University College London, UK, 2003.

[24] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numerica*, vol. 25, pp. 161–319, 2016.

[25] J. Rapp and V. K. Goyal, "A few photons among many: Unmixing signal and noise for photon-efficient active imaging," *IEEE Trans. Comput. Imag.*, vol. 3, no. 3, pp. 445–459, 2017.

[26] Z. T. Harmany, R. F. Marcia, and R. M. Willett, "This is SPIRAL-TAP: Sparse poisson intensity reconstruction algorithms: Theory and practice," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1084–1096, Mar 2012.

[27] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–3145, Dec 2010.

[28] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[30] F. Heide, S. Diamond, D. B. Lindell, and G. Wetzstein, "Sub-picosecond photon-efficient 3D imaging using single-photon sensors," *Sci. Rep.*, vol. 8, no. 1, p. 17726, 2018.

[31] Y. Altmann and S. McLaughlin, "Range estimation from single-photon lidar data using a stochastic EM approach," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Rome, Italy, Sep 2018, pp. 1112–1116.

[32] J. Rapp, R. M. A. Dawson, and V. K. Goyal, "Estimation from quantized gaussian measurements: When and how to use dither," *IEEE Trans. Signal Process.*, vol. 67, no. 13, pp. 3424–3438, July 2019.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[34] D. B. Lindell, M. O'Toole, and G. Wetzstein, "Single-photon 3D imaging with deep sensor fusion," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 113:1–113:12, Jul. 2018.

[35] Y. Altmann, X. Ren, A. McCarthy, G. S. Buller, and S. McLaughlin, "Robust Bayesian target detection algorithm for depth imaging from sparse single-photon data," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 456–467, Dec 2016.

[36] D. Shin, F. Xu, F. N. Wong, J. H. Shapiro, and V. K. Goyal, "Computational multi-depth single-photon imaging," *Opt. Express*, vol. 24, no. 3, pp. 1873–1888, 2016.

[37] A. Halimi, R. Tobin, A. McCarthy, S. McLaughlin, and G. S. Buller, "Restoration of multi-layered single-photon 3D lidar images," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Kos Island, Greece, Aug. 2017, pp. 708–712.

[38] S. Hernandez-Marin, A. M. Wallace, and G. J. Gibson, "Bayesian analysis of lidar signals with multiple returns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2170–2180, 2007.

[39] ——, "Multilayered 3D lidar image construction using spatial models in a Bayesian framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1028–1040, 2008.

[40] C. Mallet, F. Lafarge, M. Roux, U. Soergel, F. Bretar, and C. Heipke, "A marked point process for modeling lidar waveforms," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3204–3221, 2010.

[41] J. Tachella, Y. Altmann, X. Ren, A. McCarthy, G. Buller, S. McLaughlin, and J. Tourneret, "Bayesian 3D reconstruction of complex scenes from single-photon lidar data," *SIAM Journal on Imaging Sciences*, vol. 12, no. 1, pp. 521–550, 2019.

[42] J. Tachella, Y. Altmann, S. McLaughlin, and J. . Tourneret, "3d reconstruction using single-photon lidar data exploiting the widths of the returns," in *Proc. Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, May 2019, pp. 7815–7819.

[43] J. Tachella, Y. Altmann, M. Márquez, H. Arguello-Fuentes, J.-Y. Tourneret, and S. McLaughlin, "Bayesian 3D reconstruction of subsampled multispectral single-photon lidar signals," *To Appear in IEEE Trans. Comput. Imag.*, Apr 2019.

[44] J. Tachella, Y. Altmann, S. McLaughlin, and J. . Tourneret, "Fast surface detection in single-photon lidar waveforms," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, A Coruna, Spain, Sep. 2019, pp. 1–5.

[45] J. Tachella, Y. Altmann, S. McLaughlin, and J.-Y. Tourneret, "On fast object detection using single-photon lidar data," in *Proc. SPIE Wavelets and Sparsity XVIII*, vol. 11138. San Diego, USA: SPIE, Sep 2019, pp. 252 – 261.

[46] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. Global Conf. Signal and Inf. Process. (GlobalSIP)*, Austin, USA, 2013, pp. 945–948.

[47] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[48] J. Tachella, Y. Altmann, N. Mellado, A. Tobin, R.and McCarthy, G. Buller, J. Tourneret, and S. McLaughlin, "Real-time 3D reconstruction from single-photon lidar data using plug-and-play point cloud denoisers," *Nat. Commun.*, no. 10, p. 4984, 2019.

[49] J. Tachella, Y. Altmann, S. McLaughlin, and J. . Tourneret, "Real-time 3D color imaging with single-photon lidar data," in *Proc. 8th Int. Workshop Comput. Adv. Multi-Sensor Adap. Process. (CAMSAP)*, Guadaloupe, West Indies, Dec 2019, pp. 1–5.

[50] M. N. M. Van Lieshout, *Markov point processes and their applications*. World Scientific, London, UK, 2000.

[51] P. McCool, Y. Altmann, A. Perperidis, and S. McLaughlin, "Robust markov random field outlier detection and removal in subsampled images," in *Proc. Workshop Stat. Signal Process. (SSP)*, Palma de Mallorca, Spain, Jun 2016, pp. 1–5.

[52] J. Møller and R. P. Waagepetersen, "Modern statistics for spatial point processes," *Scandinavian Journal of Statistics*, vol. 34, no. 4, pp. 643–684, 2007.

[53] A. J. Baddeley and M. Van Lieshout, "Area-interaction point processes," *Ann. Inst. Statistical Math.*, vol. 47, no. 4, pp. 601–619, 1995.

[54] I. Murray, Z. Ghahramani, and D. MacKay, "MCMC for doubly-intractable distributions," *Preprint https://arxiv.org/abs/1206.6848*, 2012.

[55] H. Rue and L. Held, *Gaussian Markov random fields: Theory and applications*. CRC press, Boca Raton, USA, 2005.

[56] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[57] O. Dikmen and A. T. Cemgil, "Gamma Markov random fields for audio source modeling," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 3, pp. 589–601, 2010.

[58] Y. Altmann, R. Aspden, M. Padgett, and S. McLaughlin, "A Bayesian approach to denoising of single-photon binary images," *IEEE Trans. Comput. Imag.*, 2017.

[59] Y. Altmann, A. Maccarone, A. McCarthy, G. Newstadt, G. Buller, S. McLaughlin, and A. Hero, "Robust spectral unmixing of sparse multispectral lidar waveforms using gamma markov random fields," *IEEE Trans. Comput. Imag.*, 2017.

[60] M. Pereyra, N. Whiteley, C. Andrieu, and J.-Y. Tourneret, "Maximum marginal likelihood estimation of the granularity coefficient of a Potts-Markov random field within an MCMC algorithm," in *Proc. Workshop Stat. Signal Process. (SSP)*, Gold Coast, Australia, 2014, pp. 121–124.

[61] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[62] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley, "Non-parametric bayesian dictionary learning for sparse image representations," in *Proc. Advances in neural information processing systems (NIPS)*, 2009, pp. 2295–2303.

[63] M. Zhou, L. Hannah, D. B. Dunson, and L. Carin, "Beta-negative binomial process and poisson factor analysis." in *AISTATS*, vol. 22, La Palma, Canary Islands, 2012, pp. 1462–1471.

[64] L. Azzari and A. Foi, "Variance stabilization for noisy+ estimate combination in iterative poisson denoising," *IEEE Signal Process. Lett.*, vol. 23, no. 8, pp. 1086–1090, 2016.

[65] P. Lancaster and K. Salkauskas, "Surfaces generated by moving least squares methods," *Mathematics of computation*, vol. 37, no. 155, pp. 141–158, 1981.

[66] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.* Minneapolis, USA: IEEE, Jun 2007, pp. 1–8.

[67] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing.* Springer, 2011.

[68] https://github.com/photon-efficient-imaging/full-waveform/tree/master/fcns, 2019.

[69] X. Ren, P. W. R. Connolly, A. Halimi, Y. Altmann, S. McLaughlin, I. Gyongy, R. K. Henderson, and G. S. Buller, "High-resolution depth profiling using a range-gated CMOS SPAD quanta image sensor," *Opt. Express*, vol. 26, no. 5, pp. 5541–5557, Mar 2018.

[70] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[71] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[72] A. M. Wallace, A. McCarthy, C. J. Nichol, X. Ren, S. Morak, D. Martinez-Ramirez, I. H. Woodhouse, and G. S. Buller, "Design and evaluation of multispectral LiDAR for the recovery of arboreal parameters," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4942–4954, Aug 2014.

[73] Y. Altmann, A. Wallace, and S. McLaughlin, "Spectral unmixing of multispectral Lidar signals," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5525–5534, Oct 2015.

[74] G. Wei, S. Shalei, Z. Bo, S. Shuo, L. Faquan, and C. Xuewu, "Multi-wavelength canopy lidar for remote sensing of vegetation: Design and system performance," *ISPRS J. Photogrammetry Remote Sens.*, vol. 69, pp. 1 – 9, 2012.

[75] Y. Altmann, R. Tobin, A. Maccarone, X. Ren, A. McCarthy, G. S. Buller, and S. McLaughlin, "Bayesian restoration of reflectivity and range profiles from subsampled single-photon multispectral lidar data," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Kos Island, Greece, Aug. 2017, pp. 1410–1414.

[76] X. Ren, Y. Altmann, R. Tobin, A. Mccarthy, S. Mclaughlin, and G. S. Buller, "Wavelength-time coding for multispectral 3D imaging using single-photon lidar," *Opt. Express*, vol. 26, no. 23, pp. 30 146–30 161, Nov 2018.

[77] H. Arguello and G. R. Arce, "Colored coded aperture design by concentration of measure in compressive spectral imaging," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1896–1908, April 2014.

[78] C. V. Correa, H. Arguello, and G. R. Arce, "Spatiotemporal blue noise coded aperture design for multi-shot compressive spectral imaging," *JOSA A*, vol. 33, no. 12, pp. 2312–2322, 2016.

[79] P.-J. Lapray, X. Wang, J.-B. Thomas, and P. Gouton, "Multispectral filter arrays: Recent advances and practical implementation," *Sensors*, vol. 14, no. 11, pp. 21 626–21 659, 2014.

[80] J. Tachella, Y. Altmann, M. Pereyra, S. McLaughlin, and J.-Y. Tourneret, "Bayesian restoration of high-dimensional photon-starved images," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 747–751.

[81] C. Gilavert, S. Moussaoui, and J. Idier, "Efficient Gaussian sampling for solving large-scale inverse problems using MCMC," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 70–80, Jan 2015.

[82] M. Raginsky, R. M. Willett, Z. T. Harmany, and R. F. Marcia, "Compressed sensing performance bounds under poisson noise," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 3990–4002, Aug 2010.

[83] M. Raginsky, S. Jafarpour, Z. T. Harmany, R. F. Marcia, R. M. Willett, and R. Calderbank, "Performance bounds for expander-based compressed sensing in Poisson noise," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4139–4153, Sep 2011.

[84] Y. Li and G. Raskutti, "Minimax optimal convex methods for Poisson inverse problems under ℓ_q-ball sparsity," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5498–5512, Aug 2018.

[85] O. Deussen, S. Hiller, C. Van Overveld, and T. Strothotte, "Floating points: A method for computing stipple drawings," *Computer Graphics Forum*, vol. 19, no. 3, pp. 41–50, 2000.

[86] G. Liang, L. Lu, Z. Chen, and C. Yang, "Poisson disk sampling through disk packing," *Computational Visual Media*, vol. 1, no. 1, pp. 17–26, 2015.

[87] L. Galvis, E. Mojica, H. Arguello, and G. R. Arce, "Shifting colored coded aperture design for spectral imaging," *Appl. Opt.*, vol. 58, no. 7, pp. B28–B38, 2019.

[88] L. Galvis, D. Lau, X. Ma, H. Arguello, and G. R. Arce, "Coded aperture design in compressive spectral imaging based on side information," *Appl. Opt.*, vol. 56, no. 22, pp. 6332–6340, 2017.

[89] K. M. León-López, L. V. G. Carreño, and H. Arguello, "Temporal colored coded aperture design in compressive spectral video sensing," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 253–264, 2019.

[90] E. Mojica, S. Pertuz, and H. Arguello, "High-resolution coded-aperture design for compressive x-ray tomography using low resolution detectors," *Optics Communications*, vol. 404, pp. 103–109, 2017.

[91] K. Choi and D. J. Brady, "Coded aperture computed tomography," in *Proc. SPIE Adaptive Coded Aperture Imaging, Non-Imaging, and Unconventional Imaging Sensor Systems*, vol. 7468, San Diego, USA, 2009, pp. 99 – 108.

[92] C. Hinojosa, J. Bacca, and H. Arguello, "Coded aperture design for compressive spectral subspace clustering," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1589–1600, 2018.

[93] V. Surazhsky, P. Alliez, and C. Gotsman, "Isotropic remeshing of surfaces: a local parameterization approach," in *Technical Report, INRIA*, 2003.

[94] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz, "Realistic modeling and rendering of plant ecosystems," in *Proc. 25th annual conference on Computer graphics and interactive techniques*, Orlando, USA, 1998, pp. 275–286.

[95] D. L. Lau, R. Ulichney, and G. R. Arce, "Blue and green noise halftoning models," *IEEE Signal Process. Mag.*, vol. 20, no. 4, pp. 28–38, July 2003.

[96] J. Piironen and A. Vehtari, "Sparsity information and regularization in the horseshoe and other shrinkage priors," *Electron. J. Statist.*, vol. 11, no. 2, pp. 5018–5051, 2017.

[97] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Aug. 2004.

[98] M. Entwistle, M. A. Itzler, J. Chen, M. Owens, K. Patel, X. Jiang, K. Slomkowski, and S. Rangwala, "Geiger-mode APD camera system for single-photon 3D LADAR imaging," in *Proc. Advanced Photon Counting Techniques VI*, vol. 8375, Baltimore, USA, 2012, pp. 78–89.

[99] R. K. Henderson, N. Johnston, H. Chen, D. D. Li, G. Hungerford, R. Hirsch, D. McLoskey, P. Yip, and D. J. S. Birch, "A 192x128 time correlated single photon counting imager in 40nm CMOS technology," in *Proc. 44th European Solid State Circuits Conference (ESSCIRC)*, Dresden, Germany, Sep 2018, pp. 54–57.

[100] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," *Computer Graphics Forum*, vol. 36, no. 1, pp. 301–329, 2017.

[101] Y. Altmann, R. Aspden, M. Padgett, and S. McLaughlin, "A Bayesian approach to denoising of single-photon binary images," *IEEE Trans. Comput. Imag.*, vol. 3, no. 3, pp. 460–471, Sept 2017.

[102] M.-J. Sun, M. P. Edgar, G. M. Gibson, B. Sun, N. Radwell, R. Lamb, and M. J. Padgett, "Single-pixel three-dimensional imaging with time-based depth resolution," *Nat. Commun.*, vol. 7, p. 12010, 2016.

[103] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 408–423, Dec 2016.

[104] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, March 2017.

[105] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[106] J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett, "Poisson noise reduction with non-local PCA," *Journal of Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 279–294, Feb. 2014.

[107] W. Marais and R. Willett, "Proximal-gradient methods for Poisson image reconstruction with BM3D-based regularization," in *Proc. 7th Int. Workshop Comput. Adv. Multi-Sensor Adap. Process. (CAMSAP)*, Curacao, Dutch Antilles, Dec 2017, pp. 1–5.

[108] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1, pp. 459–494, Aug 2014.

[109] G. Guennebaud, M. Germann, and M. Gross, "Dynamic sampling and rendering of algebraic point set surfaces," *Computer Graphics Forum*, vol. 27, no. 2, pp. 653–662, 2008.

[110] M. Alexa and A. Adamson, "On normals and projection operators for surfaces defined by point sets," in *Proc. Eurographics Conference on Point-Based Graphics*, Aire-la-Ville, Switzerland, 2004, pp. 149–155.

[111] J. Liang and H. Zhao, "Solving partial differential equations on point clouds," *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. A1461–A1486, 2013.

[112] A. Buades, B. Coll, and J. . Morel, "A non-local algorithm for image denoising," in *Proc. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, San Diego, USA, Jun 2005, pp. 60–65.

[113] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[114] S. A. Gershgorin, "Uber die abgrenzung der eigenwerte einer matrix," , no. 6, pp. 749–754, 1931.

[115] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM Journal on Imaging Sciences*, vol. 11, no. 3, pp. 2001–2020, 2018.

[116] F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[117] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[118] D. R. Bickel and R. Frühwirth, "On a fast, robust estimator of the mode: Comparisons to other robust estimators with applications," *Computational Statistics & Data Analysis*, vol. 50, no. 12, pp. 3500–3530, 2006.

[119] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[120] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6664–6684, Oct 2019.

[121] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, 1981.

[122] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré, "Stein unbiased gradient estimator of the risk (SUGAR) for multiple parameter selection," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2448–2487, 2014.

[123] J. Sanders and E. Kandrot, *CUDA by example: An introduction to general-purpose GPU programming*. Addison-Wesley Professional, Ann Arbor, USA, 2010.

[124] V. Chandrasekaran and M. I. Jordan, "Computational and statistical tradeoffs via convex relaxation," *In Proc. of the National Academy of Sciences*, vol. 110, no. 13, pp. 1181–1190, 2013.

[125] S. Xiong, J. Zhang, J. Zheng, J. Cai, and L. Liu, "Robust surface reconstruction via dictionary learning," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 201:1–201:12, 2014.

[126] C.-H. Shen, S.-S. Huang, H. Fu, and S.-M. Hu, "Adaptive partitioning of urban facades," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 184:1–184:10, 2011.

[127] L. Nan, K. Xie, and A. Sharf, "A search-classify approach for cluttered indoor scene understanding," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 137:1–137:10, Nov. 2012.

[128] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, July 2017.

[129] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2018, pp. 820–830.

[130] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *Preprint at https://arxiv.org/abs/1801.07829*, 2018.

[131] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. on Comput. Vis., Bombay, India*, Jan 1998, pp. 839–846.

[132] A. Halimi, R. Tobin, A. McCarthy, J. Bioucas-Dias, S. McLaughlin, and G. S. Buller, "Robust restoration of sparse multidimensional single-photon lidar images," *IEEE Trans. Comput. Imag. (Early Access)*, 2019.

[133] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Proc.*, vol. 11, no. 6, pp. 670–684, 2002.

[134] A. Aßmann, B. Stewart, J. Mota, and A. Wallace, "Compressive super-pixel lidar for high-frame rate 3d depth imaging," in *Proc. Global Conf. Signal and Inf. Process. (GlobalSIP)*, Ottawa, Canada, 8 2019.

[135] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Advances in neural information processing systems (NIPS)*, Vancouver, Canada, 2008, pp. 1177–1184.

[136] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, "Kernel mean embedding of distributions: A review and beyond," *Foundations and Trends® in Machine Learning*, vol. 10, no. 1-2, pp. 1–141, 2017.

[137] Y. Petillot, I. T. Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *IEEE J. Ocean. Eng.*, vol. 26, no. 2, pp. 240–251, April 2001.

[138] M. O'Toole, D. B. Lindell, and G. Wetzstein, "Confocal non-line-of-sight imaging based on the light-cone transform," *Nature*, vol. 555, no. 7696, p. 338, 2018.

[139] M. La Manna, F. Kine, E. Breitbach, J. Jackson, T. Sultan, and A. Velten, "Error backprojection algorithms for non-line-of-sight imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1615–1626, July 2019.