Parallel Block Compressive LiDAR Imaging

Andreas Aßmann, *Member, IEEE*, João F. C. Mota, *Member, IEEE*, Brian D. Stewart, and Andrew M. Wallace, *Fellow, IET*

***Abstract*—We propose an architecture for reconstructing depth images from raw photon count data. The architecture uses very sparse illumination patterns, making it not only computationally efficient, but due to the significant reduction in illumination density, also low power. The main idea is to apply compressive sensing (CS) techniques to block (or patch) regions in the array, which results in improved reconstruction performance, fast concurrent processing, and scalable spatial resolution. Using real and simulated arrayed LiDAR data, our experiments show that the proposed framework achieves excellent depth resolution for a wide range of operating distances and outperforms previous algorithms for depth reconstruction from photon count data in both accuracy and computational complexity. This enables eye-safe reconstruction of high-resolution depth maps at high frame rates, with reduced power and memory requirements. It is possible to sample and reconstruct a depth map in just 12 ms, enabling real-time applications at frame rates above 80 Hz.**

***Index Terms*—Compressive Sensing, LiDAR Imaging, 3D Image Reconstruction, Parallelization**

I. Introduction

THE next generation of autonomous robots, including self-driving cars, require safe and fast depth perception for reliable navigation in complex environments. Active sensors, in particular light detection and ranging (LiDAR) systems, are critical to achieving this task. LiDAR sensors operate by measuring the time it takes for a light wave to travel to a specific object and back. This round-trip, time-of-flight (ToF), is proportional to the distance of the object in relation to the speed of light. The most common forms of LiDAR systems mechanically scan a coherent light source in discrete steps. They provide accurate distance measurements at short range and across hundreds of metres [1], [2]. The mechanical nature of current LiDAR scanners and the requirement for high-precision calibration, however, makes their manufacturing process very expensive.

This has recently motivated the development of solid-state ToF arrays [3]–[5], which are smaller, have no moving parts, and can be mass produced. While solid-state LiDAR arrays have many more sensing elements than mechanical LiDARs, increasing the resolution of the acquired depth image requires a proportional increase in illumination and, thereby, of the emitted laser power. This is particularly important in scenarios with large fields-of-view (FoV), non-ideal reflectors, and complex scenes that are common for fully autonomous driving systems. Increasing resolution (amount of data sampled) and the illumination (emitted power), however, not only require complex processing schemes, which can lead to prohibitive operation times in dynamic applications, but also pose severe eye-safety challenges [6].

To overcome these limitations and to reduce the illumination density in LiDARs, [7]–[9] proposed the use of compressive sensing (CS). By leveraging structured illumination patterns, CS allows the acquisition of data in compressed form, but requires the solution of a complex optimization problem to reconstruct it, which has been a challenge to achieve in real-time or at video frame rates. For this reason, the scenarios considered in prior work are often limited to simple short range scenes not representative of real-world scenarios encountered by autonomous systems.

Problem Statement. The current limitations of solid-state LiDAR arrays motivate the problem that we aim to solve: *to design a LiDAR system that reconstructs high-resolution depth images, requires low illumination power, and is fast enough to reconstruct images in real-time.*

Our strategy to address this problem is to apply CS to two quantitative compressive measurements obtained directly from LiDAR data: the photon count and the depth-sum.

To enable the real-time operation of the resulting system, we propose to sense the scene in a block fashion. That is, we divide the scene into small blocks (or patches), each of which is sensed independently and concurrently. Such a block sensing scheme then enables us to either reconstruct the blocks independently (and in parallel) or to devise an heuristic that computes a good estimate of the scene.

Contributions. We summarize our contributions as follows:

- We propose a block model for depth imaging and an associated CS reconstruction framework for photon detector arrays with random sparse sampling patterns.
- We formulate the depth reconstruction problem under a rigorous framework and propose an algorithm that, in contrast with previous work, imposes no constraints on the content of the scene, imaging range, or application scenario.
- We present a de-blocking scheme for very high compression rates and low illumination density for sparse LiDAR imaging.
- We show that smaller block sizes enable independent parallel block compressive depth imaging at higher fidelity than prior work with processing times of the order of 10 ms for 128×128 depth maps.

This work was supported by STMicroelectronics R&D Ltd. and the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/L01596X/1.

Andreas Aßmann is with STMicroelectronics R&D Ltd. and was with the School of Engineering and Physical Sciences (EPS) and the Centre for Doctoral Training in Applied Photonics (CDTAP) while undertaking this work at Heriot-Watt University, Edinburgh, United Kingdom (e-mail: andreas.assmann@st.com).

Brian Stewart is with STMicroelectronics R&D Ltd., Edinburgh, United Kingdom

João F. C. Mota and Andrew M. Wallace are with the School of Engineering and Physical Sciences (EPS) at Heriot-Watt University, Edinburgh, United Kingdom (e-mail: {j.mota, a.m.wallace}@hw.ac.uk)

Our framework applies to sparsely sampled full histogram pixel data, an important distinction from single or few valued pixel data and normal intensity imaging. Part of this work was presented in [10], and the current paper extends it in several ways. We present a scalable system architecture capable of real-time depth reconstruction with greatly reduced memory and laser power requirements. Laser power per illumination can be reduced for eye safety or increased to improve reconstruction quality with adjustable pattern density. We validate our sparsity assumptions experimentally for different sparsity models, namely Daubechies wavelets (DWT), discrete cosine transform (DCT), and total-variation (TV) and elaborate the full signal and system models to the independent array blocking scheme for computationally efficient depth reconstruction. We expand our framework to leverage full-frame TV sparsity to reduce blocking artefacts in ultra low compression cases and propose a discrete reconstruction scheme. Greatly extended experiments assess the performance of the proposed parallel framework in much more detail running on a multi-core CPU.

Outline. The remainder of the paper is organized as follows. In Section II, we discuss related work on CS applied to ToF depth imaging, as well as on block CS applied to reducing computational cost. Section III provides background on the signal model and on the CS formulation of the problem. Section IV describes the proposed system architecture and formulates the block scheme applied to compressive depth recovery. Finally, in Section V, we evaluate how our assumptions hold in practice and compare our method against prior art in terms of speed and quality. Section VI concludes the paper.

II. Related Work

Depth imaging based on ToF principles requires processing large amounts of data because of the large numbers of histogram bins (temporal or depth dimension) and pixels (spatial dimension). Such a volume of data has motivated CS approaches, which we review next. Then, we summarize methods using block-based CS and discuss their limitations and trade-offs.

Histogram processing. To recover depth information from photon count data it is common to exploit a sparse signal model that assumes that an active sensor only receives very few surface reflections within the field-of-view of a single photon detector pixel, or in more recent work [11] [12] [13], within neighbouring pixels. These approaches achieve good depth reconstruction, but scale poorly with the number of pixels, n, and the length of the histogram, p, where p represents the number of discrete distance (time) steps. Efforts have also been made to accelerate the processing by utilising GPUs, e.g. [14], but limiting non-zero bins in a histogram to ≈ 256 bins to accommodate memory limitations.

Sliced spatial compressive depth. Depth images can be reconstructed from ToF measurements by assuming they are sparse in some basis as presented in [7]. In other words, the number of nonzero entries of a signal is much smaller than the number of zeros when transformed into another basis domain. This, however, implicitly assumes temporal sparsity, which implies that intensity masks are sparse and temporally clustered.

Depth images can also be recovered by solving p multiple independent 2D imaging problems, each of which is a standard CS problem [9], [15]. As each time-gated intensity slice is recovered sequentially, the complexity of reconstruction is $O(pn^3)$ where n is the number of pixels, using conventional CS algorithms. For example, for a modest $p = 512$ bins for short distance ranging, this results in a fairly low frame rate of 3 Hz for a depth map of size $n = 64 \times 64$ [15], highlighting the limitations in terms of scalability of this approach for higher spatial resolution and a wider operating range.

Masked spatial compressive depth. To minimise the computational burden in the reconstruction process, [16] introduced mask priors and constrained the problem to two Lambertian surfaces. The positions of the surfaces were estimated from the acquired histograms in a parametric fashion, and their shape recovered by solving two independent CS problems. However, the proposed method is limited to a small number of surfaces. A more general framework was presented in [8], which introduced a proxy, the so-called *time-of-flight sum* or simply depth-sum. This uses a two step recovery process, which implicitly assumes a few simple, planar surfaces enforced by a small TV-norm and subsequent basis thresholding in the wavelet domain. This is similar to explicit masking [16]. This approach is also very sensitive to ambient illumination. The acquisition times are further limited by the spatial light modulator (SLM) and by the problem size, a limitation of most single-pixel systems with a hard limit defined by the desired size of the final image.

Block compressive sensing. Blocking schemes are a common technique to distribute computationally expensive operations across smaller sub-problems and thus have naturally been applied to compressive sensing, in particular, to single-pixel cameras for intensity imaging [17], [18]. However, to the best of our knowledge, they have not yet been applied to compressive depth imaging. Most blocking formulations decrease reconstruction times, but require raster scanning the spatial-light-modulator, which increases the sampling time compared to normal CS.

Limitations. The use of two or few image reconstructions in order to recover depth is intriguing for its simplicity. However, for practical applications with a wide range of scenarios and non trivial surfaces across a wide operating range, e.g. 0-300 m, the assumption of a few simple surfaces at well separated depths is a major limitation. A more robust noise removal scheme is also required to deal with outdoor applications. Further, the acquisition time has to be reduced without limiting resolution, while processing time needs to be shortened by several orders of magnitude for real-time applications.

In our work, we address these shortcomings of compressive single-pixel depth recovery. We expand the concept of the depth-sum and formulate it more rigorously in Section III and extend it with block-independent sparsity regularisation, developing a system approach compatible with the emerging solid-state, photon detector LiDAR arrays, e.g. [5].

III. Background

We now provide background required to understand our approach. Although the depth-sum concept is based on [8],

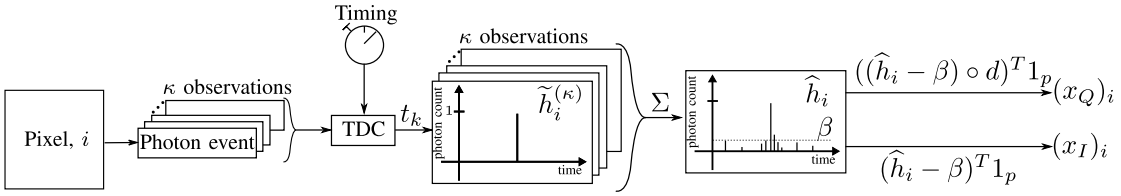


Fig. 1: Histogram acquisition for pixel i assuming one count per illumination cycle. This results in κ measurements, which are added together to produce the histogram estimate $\widehat{h}_i$. After eliminating noise β, we extract from $\widehat{h}_i$ the depth-sum $(x_Q)_i$ and the photon count $(x_I)_i$.

our models are more rigorous and precise. We start with a signal model associated with a single pixel and then generalize it to a full array. We next show how the former can be constructed from photon count measurements as two optimization problems.

Compressive depth signal model. We model a scene viewed from a LiDAR system as a 2D image $X_D \in \mathbb{R}^{N_x \times N_y}$ and represent its column-major vectorization as $x_D \in \mathbb{R}^n$, where $n := N_x \cdot N_y$. For each pixel i in the scene, a ToF LiDAR system collects direct distance measurements of an object observed at that pixel in the form of an histogram of the photon returns.

Ideal model. In the ideal case, we assume that, at each pixel i, there is a single photon return from an object at a specific distance. Representing the vector of possible, discretized distances by $d \in \mathbb{R}^p$, and assuming the system can detect objects in all p bins, pixel i of the depth image can be expressed as

$$(x_D)_i = (h_i \circ d)^T 1_p, \tag{1}$$

where $\circ$ denotes element-wise multiplication, $1_p \in \mathbb{R}^p$ the all-ones vector, and $h_i \in \mathbb{N}^p$ the basis vector with all entries but the singular distance bin index equal to zero, indicating the distance of the observed return. Note that in this ideal case only one photon measurement is necessary.

Probabilistic model. The model in (1) is ideal because, besides assuming no noise, it considers h_i to be a canonical vector, that is, it encodes the assumption of a perfect photon return on a single surface. A practical direct ToF system, however, registers photon counts in a much more haphazard manner, a process usually modelled as a Poisson random process [19], [20]. To compute the distance of an object more accurately, several measurements are collected for each pixel, and their average can be seen as an approximation of the canonical vector h_i in (1). Fig. 1 illustrates the process. For each pixel i, the system observes κ measurements and converts them to distances by using a time-to-digital converter (TDC) sampling device [5], [21], which discretizes time and thus the possible set of distances. We encode the κth measured distance, i.e., the distance covered by the respective photon, by a canonical (or one-hot) vector $\widetilde{h}_i^{(\kappa)} \in \mathbb{R}^p$, whose entries are all zeros except the one corresponding to the bin associated to the distance of the object. By summing all these vectors, we obtain a histogram (operator Σ in Fig. 1)

$$\widehat{h}_i = \sum_{k=1}^{\kappa} \tilde{h}_i^{(k)} . \tag{2}$$

Noticing that $(\widehat{h}_i)^T 1_p$ represents the total number of photon events (i.e. measurements) detected at pixel i, whenever there is a single return surface and the number of observed measurements is large enough, the ratio $\widehat{h}_i / (\widehat{h}_i)^T 1_p$ should converge to a canonical vector h_i. Replacing h_i in (1) by this quantity then yields the following estimate for the depth of an object observed at pixel i:

$$(\widehat{x}_D)_i = \frac{(\widehat{h}_i \circ d)^T 1_p}{(\widehat{h}_i)^T 1_p} = \frac{(\widehat{x}_Q)_i}{(\widehat{x}_I)_i} , \tag{3}$$

where we define $(\widehat{x}_Q)_i := (\widehat{h}_i \circ d)^T 1_p$ as the *depth-sum* (or ToF-sum [8] with distance conversion), and $(\widehat{x}_I)_i := (\widehat{h}_i)^T 1_p$ as the *photon event sum* for pixel i.

In LiDAR imaging, the total ToF recorded during a pattern exposure is related to the pattern stimulation in exactly the same way that the total photon count is spatially related to the pattern stimulation in standard intensity imaging.

However, because a single ToF measurement at a given pixel can correspond to photons associated with neighboring pixels (potentially reflected from different surfaces), we need to scale each ToF measurement by a factor that takes these cross-pixel interactions into account. Conveniently, this factor is exactly the photon event sum (intensity), $(\widehat{x}_I)_i$ [8].

Compressive measurements. So far, and as depicted in Fig. 1, we considered the acquisition of a histogram associated with a specific pixel. To reduce the number of illuminated pixels, however, we assume that each histogram measurement contains information from several pixels. More specifically, rather than directly measuring $(\widehat{x}_Q)_i$ and $(\widehat{x}_I)_i$, each measurement aggregates these quantities across a number of different active pixels, ρ. This reduces both sampling and laser emission power. *In other words, each measurement contains photon returns from $\rho < n$ randomly selected pixels. For simplicity, we assume a constant number ρ of activated pixels for all the measurements. Formally, we collect m measurements for both the depth-sum and photon event sum, each of which as*

$$(y_Q)_j = \sum_{i \in \mathcal{A}_j} (x_Q)_i + (\beta_Q)_j \tag{4a}$$

$$(y_I)_j = \sum_{i \in \mathcal{A}_j} (x_I)_i + (\beta_I)_j , \tag{4b}$$

where $j = 1, \ldots, m$ denotes the measurement number, $\mathcal{A}_j \subset \{1, \ldots, \rho\}$ indicates the set of active pixels that contributed to measurement j, and $(x_Q)_i$ and $(x_I)_i$ represent the *ideal* depth-sum and photon event sum. The quantities $(\beta_Q)_j, (\beta_I)_j \in \mathbb{R}_+$

represent noise from ambient photon influx, modelled as Poisson noise. The patterns $\mathcal{A}_j$ can be generated pseudo-randomly.

Relation to CS. Writing all the above quantities as vectors, (4) becomes

$$y_Q = Ax_Q + \beta_Q \tag{5a}$$
$$y_I = Ax_I + \beta_I\,, \tag{5b}$$

where the jth row of $A \in \{0,1\}^{m\times n}$ contains 1 in all the entries indexed by $\mathcal{A}_j$ and 0 elsewhere, and $y_Q, y_I, \beta_Q, \beta_I \in \mathbb{R}^m$. The ideal vectors $x_Q, x_I \in \mathbb{R}^n$ represent, respectively, the (vectorized) image of the total distance travelled by all the photons that are reflected by the most significant surface, and the (vectorized) image of the number of returned photons at pixel i. As the patterns $\mathcal{A}_j$ can be generated pseudo-randomly, A can also be pseudo-random. A key observation in [10] is that the vectors x_Q and x_I can often be modelled as independent, in the sense that x_Q and x_I can be reconstructed without taking each other into account, and as having sparse representations. That is, there exists a transform $\Theta \in \mathbb{R}^{q\times n}$ such that most entries of Θx_Q and Θx_I are zero or near-zero. Examples of Θ include the Wavelet and DCT transforms, or a difference matrix (which expresses the fact that x_Q and x_I have sparse gradients ([22], [23]). Such assumptions enable us to estimate x_Q and x_I concurrently using CS methods, as we explain next. Once we have estimates for these quantities, the depth estimate for pixel i is given by dividing them as in (3).

Reconstruction. Given that y_Q and y_I in (5) are linear measurements from x_Q and x_I, which are assumed to have sparse representations, the latter can be reconstructed using, for example, basis pursuit denoising (BPDN) [24]:

$$\underset{x_Q}{\text{minimize}} \quad \tfrac{1}{2}\|Ax_Q - y_Q\|_2^2 + \alpha_Q\|\Theta x_Q\|_1 \tag{6a}$$
$$\underset{x_I}{\text{minimize}} \quad \tfrac{1}{2}\|Ax_I - y_I\|_2^2 + \alpha_I\|\Theta x_I\|_1\,, \tag{6b}$$

where $\alpha_Q, \alpha_I \geq 0$ balance the competing terms in the objective, and $\|\cdot\|_1$ is the ℓ_1-norm.

CS theory has shown (e.g., [25]), x_Q and x_I can be reconstructed from (6a)-(6b) using much fewer measurements than the vectors' dimensions, i.e., $m \ll n$.[1]

While this makes the sampling process very efficient, solving (6a)-(6b) for high spatial resolution (large n) is too computationally intensive to be done in real-time.

IV. Proposed LiDAR System

As seen before, CS reduces the amount of data that needs to be acquired. However, reconstructing the object of interest from its measurements requires solving an optimization problem like (6a)-(6b). In our case, histograms can be acquired very efficiently in a compressed and aggregated form [cf. (5)], but the subsequent reconstruction of a depth image entails significant computation. We now describe our strategies to design LiDAR systems that are *efficient in both processes*, sensing and reconstruction. We adopt two different strategies to address this problem, each relying on different assumptions about the scene. Both strategies build on a block sensing model, explained next.

A. Block Sensing Model and Overall Scheme

Recall that each row of A in (5) specifies the pixels that contribute to a particular measurement, i.e., the set $\mathcal{A}_j$ for measurement $j = 1,\ldots,m$. In a solid-state ToF LiDAR system, such sets can be selected arbitrarily. Our key idea is then to select them such that the matrix A in (5) becomes block diagonal. More concretely, we partition the full depth image $x_D \in \mathbb{R}^n$ into B blocks (or patches, or arrays), each of length $n_B := n/B$:

$$x_D = \left(x_D^{(1)}, x_D^{(2)}, \ldots, x_D^{(B)}\right), \tag{7}$$

where $x_D^{(b)} \in \mathbb{R}^{n_B}$, where $b = 1,\ldots,B$, denotes an individual block index. The depth and photon event sums associated with $x_D^{(b)}$ are, respectively, $x_Q^{(b)}$ and $x_I^{(b)}$. By selecting A in (5) to be block diagonal, (5a) becomes

$$\begin{bmatrix} y_Q^{(1)} \\ y_Q^{(2)} \\ \vdots \\ y_Q^{(B)} \end{bmatrix} = \begin{bmatrix} A^{(1)} & & & \\ & A^{(2)} & & \\ & & \ddots & \\ & & & A^{(B)} \end{bmatrix} \begin{bmatrix} x_Q^{(1)} \\ x_Q^{(2)} \\ \vdots \\ x_Q^{(B)} \end{bmatrix} + \begin{bmatrix} \beta_Q^{(1)} \\ \beta_Q^{(2)} \\ \vdots \\ \beta_Q^{(B)} \end{bmatrix}, \tag{8}$$

where $A^{(b)} \in \mathbb{R}^{m_B\times n_B}$ and $\beta_Q^{(b)} \in \mathbb{R}^{m_B}$ for $b = 1,\ldots,B$. For simplicity, we set $m_B := m/B$. A similar model applies to the photon even sum in (5b). Each block operates independently with individual access to A, as shown in Fig. 2, in other words, the histogram quantities x_Q and x_I are sampled and processed independently in each block.

Reconstruction strategies. While the measurement structure in (8) senses the scene in an efficient and parallel manner, the reconstruction of the full vector x_Q in (7) requires solving a full CS problem involving all the blocks. For a large spatial resolution, such a problem can be computationally intensive and thus inadequate for real-time processing.

The key observation in [8] is that the full vector x_Q has small TV-norm and is sparse in the wavelet domain. In our first reconstruction strategy, we go one step further and assume that the blocks $x_Q^{(b)}$ and $x_I^{(b)}$ are, by themselves, sparse in a given dictionary.

Assumption 1: For all $b = 1,\ldots,B$, the blocks $x_Q^{(b)}, x_I^{(b)} \in \mathbb{R}^{n_B}$ have sparse representations in a given dictionary $\Theta \in \mathbb{R}^{n_B\times q}$, in the sense that there exist sparse $z_Q^{(b)}, z_I^{(b)} \in \mathbb{R}^q$ such that $x_Q^{(b)} = \Theta z_Q^{(b)}$ and $x_I^{(b)} = \Theta z_I^{(b)}$.

We provide extensive experimental evidence for this assumption in Section V-A. Because of the block-diagonal structure of A in (8), Assumption 1 enables us to reconstruct each block $x_Q^{(b)}$ [and $x_I^{(b)}$] independently and in parallel, allowing for significant speedups and thereby real-time reconstruction.

For our second strategy we assume that the *full* vectors x_Q and x_I have small 2D TV-norm independently:

[1] While most non-asymptotic CS results apply to constrained problems of the form $\min_x \|\Theta x\|_1$ s.t. $\|Ax - y\|_2 \leq \sigma$, for some $\sigma > 0$ (and $A\Theta^{-1}$ Gaussian), the formulations in (6a)-(6b) are easier to solve numerically. Both types of problems, however, are equivalent for properly selected α's and σ.

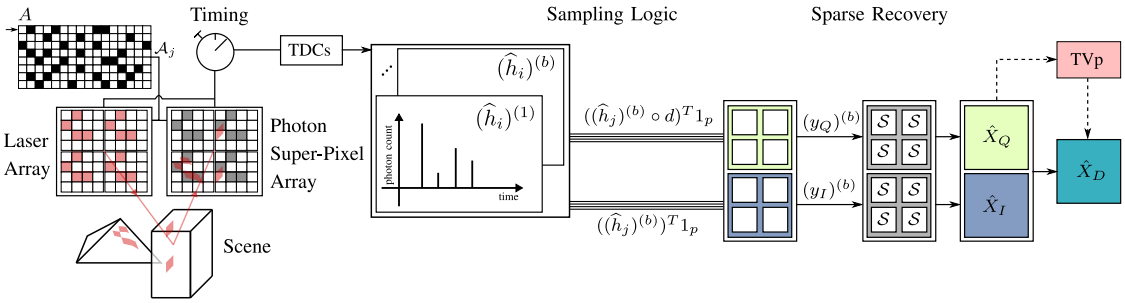


Fig. 2: Proposed compressive block LiDAR sampling. A laser array is partitioned into B blocks, each of which illuminates the scene independently. Information about the received photons is collected into histograms, from which $y_Q^{(b)}$ and $y_I^{(b)}$ in (5) are formed within each block b. A depth image $\widehat{X}_D$ is constructed by processing these measurements in parallel, which entails solving several instances of (6), represented by $\mathcal{S}$ and possibly with some post-processing, and forming $\widehat{X}_Q$ and $\widehat{X}_I$.

Assumption 2: Both $x_Q, x_I \in \mathbb{R}^n$ have small 2D TV-norm, in the sense that $\|x_Q\|_{\text{TV}}$ and $\|x_I\|_{\text{TV}}$ are small.

For a vector $x \in \mathbb{R}^n$, the 2D TV-norm is defined as $\|x\|_{\text{TV}} := \|Dx\|_1$, where each row of $D \in \mathbb{R}^{2n\times n}$ extracts either the vertical or horizontal difference at a given pixel of x. For computational efficiency, we assume periodic boundaries, so that products Dx and $D^T y$ can be computed via the FFT [23]. Similar to [8], our Assumption 2 apparently provides no computational advantage over e.g. DCT domain sparsity. Yet, the block sensing approach in (8) will enable us to design a good warm-start for the solution of a global TV problem. Specifically, we independently solve a TV minimization problem for each block, and the vector/image composed by the individual block solutions will be sufficiently close to the solution of (6a)-(6b) with A given as in (8).

Overall scheme. Fig. 2 gives an overview of the full pipeline of the proposed scheme. The sensing process is similar to [8], with several pixels in the laser array contributing to a single measurement. The main difference is the adaptation to solid-state arrays and the division of the laser array (and thus of the photon detector array) into B blocks that operate independently. Mathematically, this implements the block sensing matrix of (8). For each block b, several histograms are formed and added together [cf. (2)], yielding $(\widehat{h}_i)^{(b)}$ for pixel i of block b. Next, we extract from each of these histograms the depth-sum $(\widehat{x}_Q^{(b)})_i := \big((\widehat{h}_i)^{(b)} \circ d\big)^T 1_p$ and the photon event sum $(\widehat{x}_I^{(b)})_i := \big((\widehat{h}_i)^{(b)}\big)^T 1_p$. Because several pixels within each block are active, these quantities are actually aggregated from several pixels [cf. (4)-(5)], i.e., they comprise the measurements $(y_Q)^{(b)}$ and $(y_I)^{(b)}$. These measurements are then used to reconstruct $\widehat{x}_Q^{(b)}$ and $\widehat{x}_I^{(b)}$, associated to block b, via BPDN (6) and then tiled together to form the full images $\widehat{X}_Q$ and $\widehat{X}_I$. This reconstruction process is represented in the figure as $\mathcal{S}$. Finally, using the relation in (3), $\widehat{X}_Q$ is divided by $\widehat{X}_I$ point-wise to form the estimated depth image $\widehat{X}_D$. Note that all the processes before the formation of the full images $\widehat{X}_Q$ and $\widehat{X}_I$ can be parallelized.

Reconstruction Strategy 1. Our first reconstruction strategy relies on Assumption 1, which states that, for each block b, the quantities $x_Q^{(b)}$ and $x_I^{(b)}$ have sparse representations in a dictionary $\Theta \in \mathbb{R}^{n_B\times q}$. In this case, we reconstruct $x_Q^{(b)}$ and $x_I^{(b)}$ by solving BPDN (6a)-(6b) for each block b independently, which can be fully parallelized.

Given the measurements $y_Q^{(b)}$ and $y_I^{(b)}$ for each block b and respective measurement matrix $A^{(b)}$, we solve BPDN (6a)-(6b), represented by the map $\mathcal{S}$, to obtain $x_Q^{(b)}$ and $x_I^{(b)}$. All these reconstructions can be executed in parallel and use the same dictionary matrix Θ (even though the framework can be easily generalised to different dictionaries). Then, after joining all the blocks to form $\widehat{x}_Q$ and $\widehat{x}_I$, we use relation (3) to obtain the final vectorized depth image $\widehat{x}_D$ as illustrated in Fig. 2.

Reconstruction Strategy 2. Total-variation (TV) captures the notion that only a small number of neighbouring pixels in natural images have sharp variations in value [22], [23]. This concept, which can be expressed as a convex regularizer and thus handled efficiently, has been successfully applied to many inverse problems involving natural images [26]–[32].

Assumption 2, on which our second reconstruction strategy relies, states that the depth-sum x_Q *and* photon event sum x_I have small TV-norm. This assumption was also made in [8] for x_Q by solving (6a) with $\Theta = D$, where $D \in \mathbb{R}^{2n\times n}$ is a difference matrix. Here, however, we take advantage of the block sensing structure in (8) to compute the solutions of (6a)-(6b) faster, based on a two-step approach.

Block-TV. Using x_Q as an example, our goal is to solve (6a) with $\Theta = D$ and A having a block-diagonal structure [cf. (8)]:

$$\underset{x_Q}{\text{minimize}} \ \frac{1}{2}\sum_{b=1}^{B} \left\| A^{(b)} x_Q - y_Q^{(b)} \right\|_2^2 + \alpha_Q \|x_Q\|_{\text{TV}} \,, \quad (9)$$

where $\|x_Q\|_{\text{TV}} = \|Dx_Q\|_1$. The second term in (9) depends on the full image and does not decompose across blocks. However, since for any x_Q, there always holds

$$\sum_{b=1}^{B} \left\| x_Q^{(b)} \right\|_{\text{TV}} \leq \|x_Q\|_{\text{TV}} \,, \quad (10)$$

Assumption 2 implies that the left-hand side of (10) is expected to be small. In other words, a small total-variation of a full image implies a small total-variation of the blocks forming any partition of the image. Replacing $\|x_Q\|_{\text{TV}}$ in (9) by $\sum_{b=1}^{B} \left\| x_Q^{(b)} \right\|_{\text{TV}}$ we obtain an optimization problem that decomposes blockwise, with the bth problem taking the form

of a BPDN:

$$\underset{x_Q^{(b)}}{\text{minimize}} \ \frac{1}{2}\left\|A^{(b)}x_Q - y_Q^{(b)}\right\|_2^2 + \alpha_Q \|x_Q^{(b)}\|_{\text{TV}}\,, \qquad (11)$$

for $b = 1, \ldots, B$. Each of these problems can be solved in parallel.

B. De-blocking as a two-step approach

Based on these observations for total-variation and to reduce blocking for other basis functions we propose (using x_Q as an example) an extension to compute solutions of (6a) or (9) in two steps.

1) First, we solve (6a) or (11) for each block $b = 1, \ldots, B$ independently and tile all the blocks to form an estimate $\widetilde{x}_Q$.
2) Then, we solve (9) using an iterative solver, e.g., TVAL3 [33], using $\widetilde{x}_Q$ as a warm-start (i.e., as the initialisation of the algorithm).

Notice that step 1) reconstructs B vectors, each of size $n_B = n/B$, while step 2) reconstructs a single large vector of size n. However, because the vector $\widetilde{x}_Q$ obtained in step 1) should be a good approximation of the solution of (11), the algorithm in step 2) should require a small number of iterations. The overall process is thus efficient and, as we will show later, can be run in real-time.

While the idea of using a block-TV approach to reconstruct a full image [step 1)] has been discussed, for example, in [18], [34], our approach of using the solution of a block-TV approach to initialise the full TV problem (9) is novel. It can also reduce blocking artefacts when using prior CBCS solutions for other basis regularizations.

Next, we analyse several aspects of the proposed algorithms and discuss possible extensions.

C. Analysis and extensions

Data acquisition. We estimate data savings against a non-compressive system that acquires full histograms comprising np data points, as each pixel has p bins associated. In contrast, our system only acquires $2m$ measurements [m for y_Q and y_I each]. Furthermore, as explained later, we can estimate background noise by collecting a single histogram of size p. The savings in acquired data can then be expressed by the ratio

$$C = \frac{2m + p}{np}\,. \qquad (12)$$

Whenever $2m \ll np$, this ratio is very small. Although we lack a precise theoretical characterization of the number of measurements m required to reconstruct a s-sparse vector using BPDN (6) with a block diagonal matrix as in (8) and a generic dictionary matrix Θ, several asymptotic and non-asymptotic analyses of similar CS problems suggest $m \simeq s \log\big(n/s\big)$, e.g., [25]. The sparsity s of a vector, in our case x_Q or x_I, depends on the block size and the complexity of the scene, for example, the number of different surfaces. Expression (12) thus implies that there can be substantial savings in acquired data whenever the spatial and temporal resolutions of the system, respectively n and p, are large.

***d*Sparse.** The large savings in data acquisition expressed by (12) open up the possibility of going against CS principles to acquire more measurements m than the dimension n of the image. Indeed, if m is of the same order as n in (12), the ratio C is still small whenever p is large. In other words, acquiring more measurements m than the spatial dimension n still allows significant savings in comparison with a traditional LiDAR system that acquires full histograms for each pixel. We will refer to the regime in which the number of measurements m is larger than the dimension of the image, i.e., $m > n$, as *d*Sparse (from discrete sparse oversampling), and to the regime in which $m < n$ as *fully compressive* [35]. Compared to conventional LiDAR systems, as each measurement still contains the contributions of $\rho \ll n$ pixels, the average radiated output power of *d*Sparse is still small.

In the *d*Sparse regime, the linear systems in (5) become over-determined, as there are more equations than variables. In this case, instead of solving a BPDN problem, we estimate x_Q and x_I simply via least-squares [36], i.e., we set $\alpha_Q = \alpha_I = 0$ in (6). Due to the block sensing structure in (8), each block b can be reconstructed independently (and in parallel) by solving the linear system $\big({A^{(b)}}^T A^{(b)}\big)x = {A^{(b)}}^T y$, where $x = x_Q^{(b)}$ [resp. $x_I^{(b)}$] if $y = y_Q^{(b)}$ [resp. $y_I^{(b)}$]. Note that whenever $A^{(b)}$ is generated randomly (from a non-degenerate probability distribution), ${A^{(b)}}^T A^{(b)}$ has full rank with probability one.

Noise suppression. In practice, e.g. outdoors, there can be significant random photon influx from the sun (background rate). In this case, solely solving a BPDN problem may provide poor estimates. We thus propose two strategies to estimate and compensate for the background mean count rate, $\widehat{\beta}$.

1) In the first strategy, we aggregate a background noise histogram h_n with an additional pixel not in line with any photon emission. Exposing the detector to the same number of realisations as an instance of a regular histogram $\widehat{h}_i$, the active background compensation can be estimated as

$$\widehat{\beta}_{\text{active}} = \max\,(h_n) + \eta, \qquad (13)$$

where (h_n) denotes the dedicated noise histogram, and η is an offset parameter.
2) The second strategy uses no additional hardware. Instead, it allocates a small section of the histogram $\widehat{h}_i$ of an arbitrary pixel i to capture noise while the emitter is idle. For example, it can allocate additional l_n bins beyond the operating range. The passive background compensation is then

$$\widehat{\beta}_{\text{passive}} = \max_{p-l_n \le j \le p}\,(\widehat{h}_i) + \eta\,, \qquad (14)$$

where p is the length of $\widehat{h}_i$, i.e. number of bins.

Either noise compensation scheme is applied before storing the measurements $(y_Q)_j$ and $(y_I)_j$, and is deployed by replacing $\widehat{h}_i$ in (3) with $\widehat{h}'_i = \max\{0, \widehat{h}_i - \widehat{\beta}1_p\}$, where $\widehat{\beta}$ is either $\widehat{\beta}_{\text{active}}$ or $\widehat{\beta}_{\text{passive}}$, and $1_p \in \mathbb{R}^p$ the vector of ones.

V. Experimental Results

We now describe the experiments we conducted not only to validate our assumptions on the sparsity of the depth-sum and photon event sum (intensity) images, but also to compare the performance of our proposed scheme against prior approaches. We start by describing the common setup of all our experiments.

Scenes. We consider typical range scenarios for LiDAR imaging applications. A small dataset of 3 scenes from [37] is used to demonstrate real data compatibility with an operating range of 30 cm with sub-millimetre precision. For a more comprehensive evaluation of our parallel sparse imaging framework, we use a larger dataset containing a total of 75 scenes. We use 25 randomly selected scenes from [38] to illustrate a typical indoor short range application with an operating range of < 10 m and cm precision, e.g. for AR/VR. For automotive and outdoor applications we use a total of 50 scenes (5 each from each sequence respectively) from [39] with many participants and objects such as trees and signs in the foreground (0-50 m), buildings in the background (≤ 300 m) and more road focused scenes with a wide operating range of 0-300 m from [40].

Experimental setup. For our large dataset, histograms are simulated [41] using a sample time of 96 μs per pattern exposure (48 pulses for a maximum range of 300 m) with an ambient photon rate derived from incident sunlight at 1 klux (0.3 photons per bin) with a TDC presented in [5]. This dataset generation step takes up to 7 minutes per frame. All photon count data is re-sampled with our sampling framework. $\hat{\beta}$ is estimated passively for the real data [37] and actively for the synthetic data. Many algorithms exist to numerically find a solution for a BPDN problem such as the alternating direction method of multipliers (ADMM) [42], the fast iterative shrinkage thresholding algorithm (FISTA) [43], gradient projection for sparse reconstruction (GPSR) [44], and orthogonal matching pursuit (OMP) [45]. We used TVAL3 [33] to optimize for small TV and ADMM for sparsity in linear basis transforms. These specific algorithms were chosen for their good performance, adaptability and low execution time.

The parameters α_Q and α_I for ADMM were hand tuned. For synthetic data, the depth resolution is set at 1 cm for the indoor scene ($p = 1001$) and at 4 cm for outdoor scenes ($p = 7501$). The algorithm presented in [8] was implemented from scratch and hand tuned for depth recovery as best as possible. We used the code for for BCS-SPL (DCT) [18] provided by its authors and kept the default parameters. The published executables for RT3D [14] were used and hyper-parameters tuned for best performance. We note that histograms had to be post-processed for RT3D due to limitations in non-zero bins (active bins). We have applied an ordered threshold to discard the smallest values to retain 256 active bins, which effectively is a de-noising operation. We exclude this operation from the timing analysis.

Performance metrics. We assess the quality of the reconstructed depth vector, $y \in \mathbb{R}^n$, with the ground truth, $x \in \mathbb{R}^n$, using several figures of merit, namely the mean-squared error $\mathrm{MSE}(x, y) = \frac{1}{n}\|x - y\|_2^2$, peak signal-to-noise ratio $\mathrm{PSNR}(x, y) = 10 \log_{10} \max(x)^2 / \mathrm{MSE}(x, y)$, the signal-to-reconstruction error $\mathrm{SRE}(x, y) = 10 \log_{10} \|x\|_2^2 / \|x - y\|_2^2$ [13], and the structural similarity index measure (SSIM) [46]. In addition, we use the following metrics to assess the quality of depth reconstruction: 3-pixel-accuracy metric, which returns the percentage of good pixels according to a set of thresholds $\delta < \{1.25, 1.25^2, 1.25^3\}$, the absolute relative difference (ARD), the root mean square error with log scale (RMSE-LS) and a log-scale invariant root mean squared error (RMSE-LSI) [47].

A. Sparsity Assumptions

To validate our assumptions about the sparsity of the depth-sum and photon count, we evaluate the sparsity across our dataset for both signals.

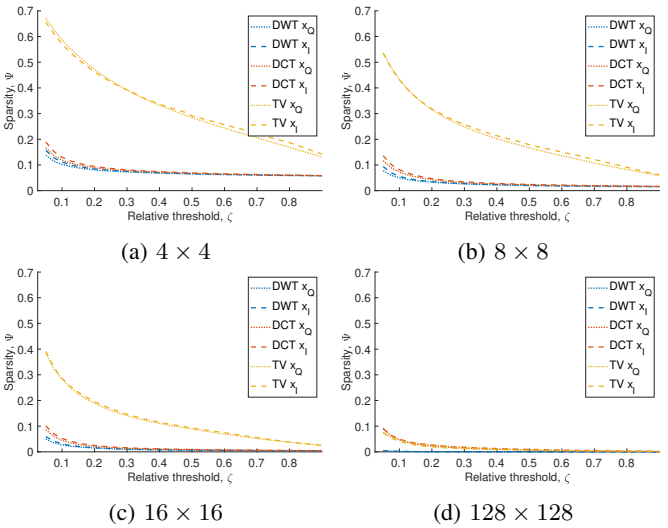
It is of particular interest how smaller block sizes below the often chosen 32×32 [17], [18] perform. Further, the assumption that x_Q is as sparse as x_I for natural scenes in [8] should be justified. We provide a more detailed analysis within this work's sampling and reconstruction framework, where there are no constraints on the number of surfaces in the scene but we assume that there are few surfaces per block.

Depth-sum Sparsity. A s-sparse vector $x \in \mathbb{R}^n$ has at most s nonzero entries. We define its sparsity ratio as

$$\Psi = 1 - \frac{s}{n}. \tag{15}$$

To test the assertion in [8] that x_I sparsity assumptions apply to x_Q, we used an estimate for reflectivity to generate a photon count image using the power model from [48] and an ideal depth image x_D. Then the depth-sum is $x_Q = x_I \circ x_D$. For linear transforms, Daubechies wavelets (DWT), DCT, and finally TV are considered in Fig. 3. Both x_Q and x_I are transformed and hard-thresholded [49] with a threshold value, τ, derived from a relative threshold ζ, such that

$$\tau = \zeta \max(x). \tag{16}$$



(a) 4×4 (b) 8×8

(c) 16×16 (d) 128×128

Fig. 3: Sparsity, Ψ, of DWT, DCT and TV norm of x_I and x_Q across relative threshold, ζ, for our dataset with specified block sizes (a)-(c) and (d) being full frame.

TABLE I: Performance evaluation of our sparse depth frameworks against prior art including a non-compressive approach. The **best overall** and *best CS* are highlighted. Scenes (128×128) are captured in B blocks with m_b measurements per block using an active pixel ratio of ρ/n for sparse frameworks. The sequential and parallel processing time are t_{seq} and t_{par} respectively. t_b is the block processing time in seconds, and the simulated sample time is provided as t_{samp}. All times are in seconds. The data ratio, C, represents the total number of measurements divided by the full histogram data.
(*Operating conditions as specified in [8]; †Running on NVidia RTX 2080 Ti (11 GB); ‡line-scanning with $\sqrt{n}$ steps.)

				higher is better				*lower is better*								
	m_b	B	ρ/n	PSNR, dB	SRE, dB	SSIM	δ_1	ARD	MSE	RMSE-LS	RMSE-LSI	t_b	t_{par}	t_{seq}	t_{samp}	C, %
RT3D [14]	16384	1	1	22.48	11.56	0.673	0.772	0.141	491.68	0.322	0.067	-	0.0396†	-	0.012‡	100.0
Howland* [8]	3277	1	0.5	13.75	2.83	0.179	0.210	2.074	2335.99	0.843	0.447	-	-	54.15	0.315	**0.005***
Howland [8]	8192	1	0.25	14.04	3.12	0.043	0.320	1.575	1834.59	0.967	0.502	-	-	107.2	0.786	0.013*
BCS-SPL$_{32}$ [18]	512	16	0.5	15.92	5.00	0.177	0.394	1.330	2410.33	0.619	0.144	-	-	0.221	0.049	0.019
CBCS$_4$-DWT	8	1024	0.5	18.75	7.83	0.183	0.805	0.154	639.13	0.313	0.064	0.00024	0.0847	0.508	0.0008	0.019
CBCS$_4$-DCT	8	1024	0.5	22.01	11.09	0.548	0.850	0.093	436.25	0.252	0.044	*0.00011*	0.0429	*0.225*	0.0008	0.019
CBCS$_4$-TV	8	1024	0.5	*25.12*	*14.21*	0.550	*0.857*	*0.081*	*265.72*	*0.201*	*0.031*	0.00767	2.8771	15.53	0.0008	0.019
CBCS$_4$-TV*	3	1024	0.5	22.44	11.52	0.334	0.839	0.112	379.37	0.234	0.039	0.00848	3.0479	17.55	**0.0003**	0.011
CBCS$_8$-DCT	32	256	0.5	19.67	8.75	0.377	0.822	0.131	789.13	0.333	0.077	0.00048	*0.0373*	0.251	0.0031	0.019
CBCS$_8$-TV	32	256	0.125	24.43	13.51	*0.553*	0.857	0.082	316.86	0.220	0.036	0.00976	0.9544	4.750	0.0031	0.019
CBCS$_4$-TV$_{dS}$	24	1024	0.5	**31.34**	**20.42**	0.879	0.864	0.055	**159.22**	**0.155**	**0.022**	0.00799	2.9002	16.43	0.0023	0.046
*d*Sparse$_4$	24	1024	0.5	30.76	19.84	**0.888**	0.867	**0.052**	373.60	0.227	0.040	**0.00002**	**0.0101**	**0.054**	0.0023	0.046
*d*Sparse$_8$	96	256	0.125	29.09	18.17	0.865	**0.868**	0.053	647.56	0.290	0.065	0.00012	0.0122	0.069	0.0092	0.046

The relative threshold, ζ, is swept across a range of 0.05 to 0.975. This ensures that the energy content is accurately considered and is equivalent to retaining ζn components of an ordered signal $x \in \mathbb{R}^n$. We show results for DWT, DCT and the TV-norm across our dataset in Fig. 3 with sparsity as defined in (15). From these results, in most basis transforms, x_Q can actually be sparser than x_I (i.e. $\Psi_Q < \Psi_I$) as the block size decreases, but is comparable to photon count intensity throughout. This is consistent for DWT, DCT and TV regularization alike. Sparsity as a whole decreases as the block size shrinks, i.e. more non-zero components are retained and therefore Ψ increases.

This indicates that it is possible to apply many of the sparsity assumptions which have been validated for natural intensity images and, by extension, photon count images to the depth-sum image. Importantly, as x_Q can be sparser than x_I for small block sizes $n_B \ll 128^2$ in some some basis regimes, the depth information contained in x_Q should be sufficiently sampled if bounds are set by the recovery of x_I.

B. Reconstruction Performance

We compare our parallel block compressive framework with a state-of-the-art GPU optimized full histogram processing approach which achieves real-time operation [14] and the single-pixel approach presented in [8], which uses a related signal model but only demonstrate indoor short range applications. We include short range data in our experimental scenes but we employ more complex (and realistic) scene compositions. Further, we compare our independent checkerboard compressive sensing (CBCS) formulation against block compressive sensing (BCS)-smoothed projected Landweber (SPL) [18], a block CS approach for intensity imaging, which proposes an optimal block size of $n_B = 32^2$, denoted BCS-SPL. The TV extension and de-blocking two-step approach TVp is evaluated as well as the discrete pseudo-inverse approach, *d*Sparse, for sparse random imaging.

Depth Reconstruction Results. The 75 scenes were reconstructed for each framework and the results were averaged

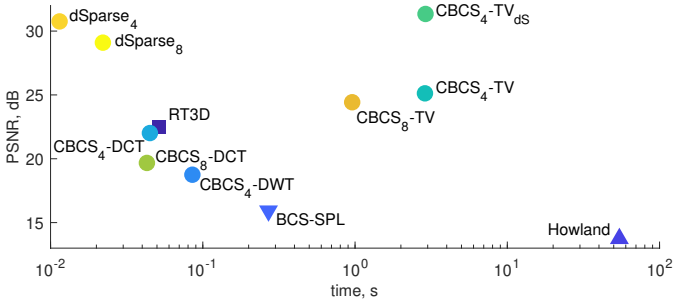
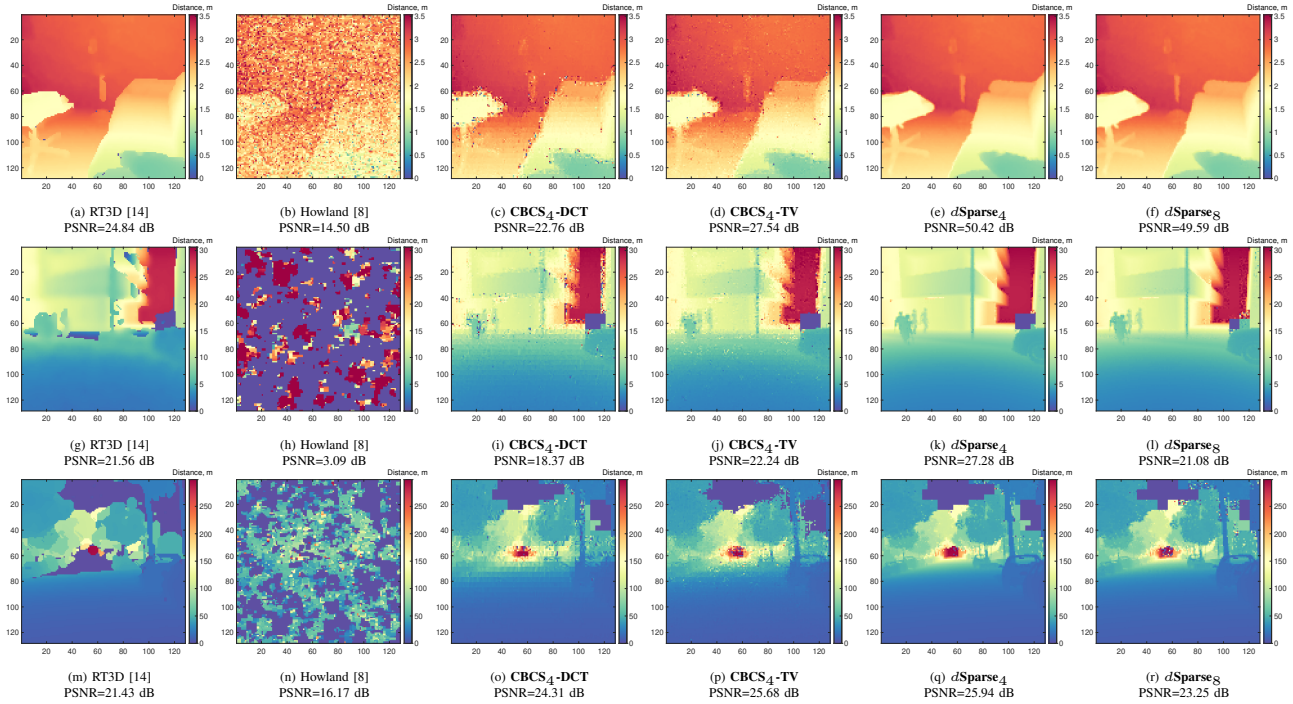


Fig. 4: Quality (PSNR) and frame time comparison for compressive depth reconstruction. The frame time is sample time and total processing time combined. Parallel processing time, t_{par}, is used where applicable.

across all scenes. The overall performance of all algorithms is presented in Tab. I. RT3D [14] was evaluated on a dedicated GPU (NVidia RTX 2080 Ti), while CBCS and *d*Sparse were processed in parallel on a general purpose CPU with 8 logical processing cores and no further optimizations (Matlab R2020a, 8×2.3 GHz Intel Core i9, 32 GB). Our proposed framework achieves reconstruction performance comparable to the much more complex framework of RT3D [14], with similar parallel execution times despite only running on an 8-core CPU.

We note that in theory each block can be reconstructed independently with dedicated logic for each block [35], resulting in theoretical reconstructions times of < 1, ms as indicated by the block time t_b. Such embarrassingly parallelization of our algorithm means that its execution time is determined only by the execution time of the slowest block. CBCS$_8$-DCT is the fastest compressive approach at 37.3 ms, closely followed by CBCS$_4$-DCT at 42.9 ms outperforming CBCS$_8$-DCT otherwise. In terms of quality, CBCS$_4$-TV is the best but also the slowest. The overall best approach in both time and quality, compared to all prior art, is *d*Sparse$_4$ with very short frame times, as illustrated in Fig. 4.

Translated into frame rates, which include sampling and



(a) RT3D [14] PSNR=24.84 dB (b) Howland [8] PSNR=14.50 dB (c) **CBCS$_4$-DCT** PSNR=22.76 dB (d) **CBCS$_4$-TV** PSNR=27.54 dB (e) ***d*Sparse$_4$** PSNR=50.42 dB (f) ***d*Sparse$_8$** PSNR=49.59 dB

(g) RT3D [14] PSNR=21.56 dB (h) Howland [8] PSNR=3.09 dB (i) **CBCS$_4$-DCT** PSNR=18.37 dB (j) **CBCS$_4$-TV** PSNR=22.24 dB (k) ***d*Sparse$_4$** PSNR=27.28 dB (l) ***d*Sparse$_8$** PSNR=21.08 dB

(m) RT3D [14] PSNR=21.43 dB (n) Howland [8] PSNR=16.17 dB (o) **CBCS$_4$-DCT** PSNR=24.31 dB (p) **CBCS$_4$-TV** PSNR=25.68 dB (q) ***d*Sparse$_4$** PSNR=25.94 dB (r) ***d*Sparse$_8$** PSNR=23.25 dB

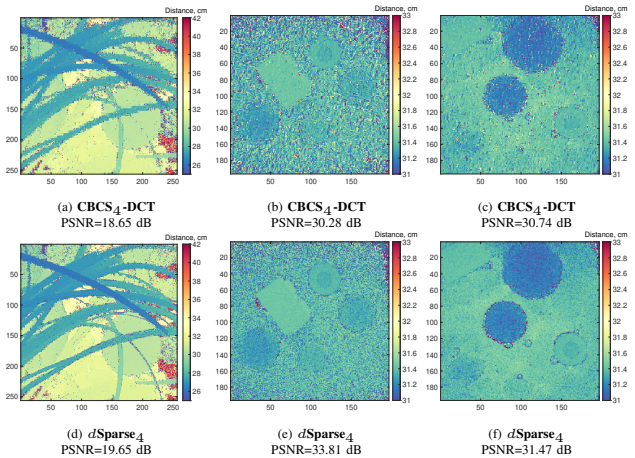
Fig. 5: Depth compressive sensing comparison for compressive depth recovery schemes. Scene examples from [38]–[40] - top to bottom: An indoor scene depicting a living room (a)-(f), a city scene with pedestrians (g)-(l) and a typical automotive scene with large dynamic range (0-300 m) in (m)-(r). All scenes were reconstructed using prior art and variations of our proposed framework (CBCS) and *d*Sparse with reconstruction quality indicated in PSNR for each individual scene.

processing times, CBCS$_4$-DCT can achieve 22 Hz and *d*Sparse$_4$ 80 Hz. RT3D [14] runs at 19 Hz for the considered scenes, assuming a typical line-scan sampling approach in $\sqrt{n}$ steps, due to safety limits in near-infrared (NIR) [6]. Furthermore, *d*Sparse$_8$ and CBCS$_8$-TV achieve optimal performance with a very low pattern density of only 12.5% pixels active, which provides a practical low power and low data approach to LiDAR imaging.

We compare our framework visually with typical scenes from each source dataset in Fig. 5. It is evident from this comparison that [8] only works at very short ranges, < 5 m (Fig. 5(b)), and struggles with noise. At longer range it fails because of their low surface count assumption. RT3D [14] performs well with some minor issues for low reflective large surfaces at long range (e.g. Fig. 5(m)). However, it is also the most complex algorithm with no compression, requiring 50 times the data volume of CBCS$_4$ with $m/n = 0.5$, and 20 times more than *d*Sparse with $m/n = 1.5$ for our outdoor scenes.

CBCS$_4$-DCT performs accurate depth recovery which retains depth gradients in blocks with minimal blocking effects in the DCT domain. The blocking primarily occurs in edge regions where a sudden depth variation occurs, implying a regularisation problem which favours smooth surfaces. Using the TV-norm, the block effects are reduced at the cost of computational effort but with overall excellent reconstruction performance for a compressive method. When we sparsely oversample spatially (while keeping compact compressive measurements aggregating time information with sparse illumination patterns) and reconstruct the scene information from an overcomplete set of subsamples the reconstruction quality is, as expected, excellent for *d*Sparse$_4$. This provides a high quality reconstruction for sparse illumination depth imaging but at the cost of additional sampling time (see Tab. I).

Real Data Demonstration. Next we demonstrate that our framework can be readily applied to real histogram data from other LiDAR sensor systems. We showcase the best overall compressive scheme (CBCS$_4$-DCT) and the best parallel sparse method (*d*Sparse$_4$) in Fig. 6.



(a) **CBCS$_4$-DCT** PSNR=18.65 dB (b) **CBCS$_4$-DCT** PSNR=30.28 dB (c) **CBCS$_4$-DCT** PSNR=30.74 dB

(d) ***d*Sparse$_4$** PSNR=19.65 dB (e) ***d*Sparse$_4$** PSNR=33.81 dB (f) ***d*Sparse$_4$** PSNR=31.47 dB

Fig. 6: Real data demonstration on scenes captured underwater from [37] with good reconstruction of sparse photon count data for our parallel sparse imaging methods.

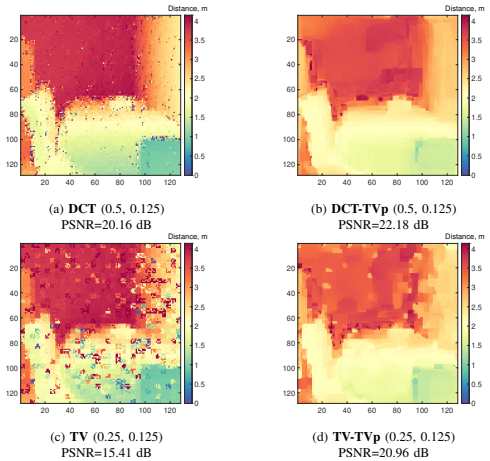
For this very sparse and low photon count data due to underwater imaging, the framework performs well and especially dSparse$_4$ recovers details and depth accurately.

De-blocking and global TV-norm. As noted earlier, CBCS can have some blocking artefacts when hard edges are present in a block. Further, we try to consolidate the TV-norm across the entire frame inline with most prior work stating low total-variation across natural scenes. We apply our second reconstruction strategy, TVp, utilizing a second TV-norm pass with initilization of a global TV optimization using their respective block solutions as priors. We highlight a few operating regimes in Tab. II and present an example case for both de-blocking of CBCS$_4$-DCT and full frame TV in Fig. 7.

TABLE II: Block prior total variation for de-blocking and global TV regularization using TVp. PSNR quality values versus ground truth are shown with parallel processing time, t, across the active pixel ratios. Highlighted entries are shown in Fig. 7.

CBCS$_4$	PSNR, dB						t, s		
m/n	0.25			0.5					
ρ/n	0.125	0.25	0.50	0.125	0.25	0.5	0.125	0.25	0.5
DCT	15.69	20.84	21.96	**15.94**	21.19	22.01	0.22	0.08	0.04
DCT-TVp	20.65	21.80	22.17	**21.29**	21.98	22.20	0.67	0.55	0.46
TV	**15.87**	20.38	22.40	19.39	24.01	25.12	3.01	3.21	3.03
TV-TVp	**21.67**	22.65	22.88	22.67	22.89	22.98	3.36	3.57	3.60

The second TV pass with prior block information has the largest effect in very high compression regions with low illumination density. Improvements of > 5 dB can be achieved across our dataset. This highlights the large compression potential of our approach alongside the flexibility to control illumination density, further increasing system efficiency.

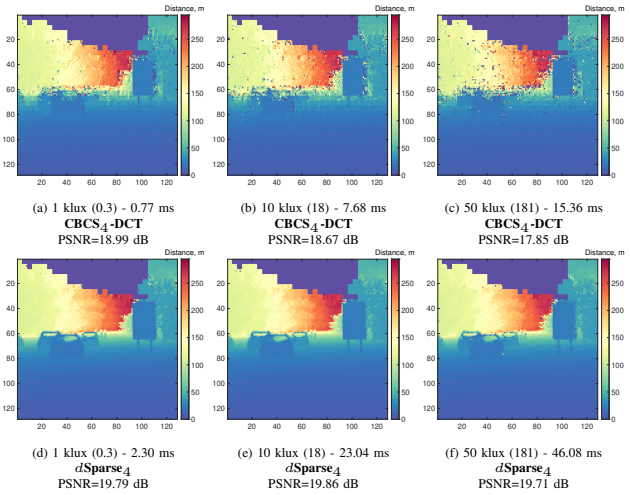


(a) **DCT** (0.5, 0.125) PSNR=20.16 dB
(b) **DCT-TVp** (0.5, 0.125) PSNR=22.18 dB
(c) **TV** (0.25, 0.125) PSNR=15.41 dB
(d) **TV-TVp** (0.25, 0.125) PSNR=20.96 dB

Fig. 7: De-blocking via global total variation regularisation and block solution prior applied to **CBCS$_4$**. In brackets are CBCS conditions (m/n, ρ/n) with quality improvement indicated by PSNR.

For the scene shown in Fig. 7, we can observe a clear de-blocking and de-noising effect for CBCS$_4$-DCT when presented with a very low illumination density of 12.5%. Meanwhile, when applied to 25% spatial compression and 12.5% illumination density using TV, the reconstruction improves dramatically with usable results with a significant reduction in block artefacts.

Background Noise Performance. CBCS and dSparse can sample the scene more rapidly, which in turn can allow for longer exposure times if required, e.g. in high noise scenarios to improve the signal-to-noise ratio (SNR), with the fully compressive scheme allowing for the longest exposure times. In terms of noise, the framework can maintain sufficient reconstruction quality for up to $50\,\mathrm{klux}$ with modest increases in exposure time illustrated in Fig. 8. For longer exposures the reconstruction quality is likely to increase.



(a) 1 klux (0.3) - 0.77 ms **CBCS$_4$-DCT** PSNR=18.99 dB
(b) 10 klux (18) - 7.68 ms **CBCS$_4$-DCT** PSNR=18.67 dB
(c) 50 klux (181) - 15.36 ms **CBCS$_4$-DCT** PSNR=17.85 dB
(d) 1 klux (0.3) - 2.30 ms **dSparse$_4$** PSNR=19.79 dB
(e) 10 klux (18) - 23.04 ms **dSparse$_4$** PSNR=19.86 dB
(f) 50 klux (181) - 46.08 ms **dSparse$_4$** PSNR=19.71 dB

Fig. 8: Noise effects on CBCS and dSparse for an automotive B-Road scene at 3 noise levels and their respective simulated sample time. Values in parenthesis after illuminance indicate mean ambient photon count per bin.

Summary. The results show that our proposed frameworks are competitive in terms of reconstruction quality and comparable precision to the non-compressive approaches, as shown by the plotted root mean squared error, RMSE$=\sqrt{\mathrm{MSE}}$, at distances up to 250 m in Fig. 9.

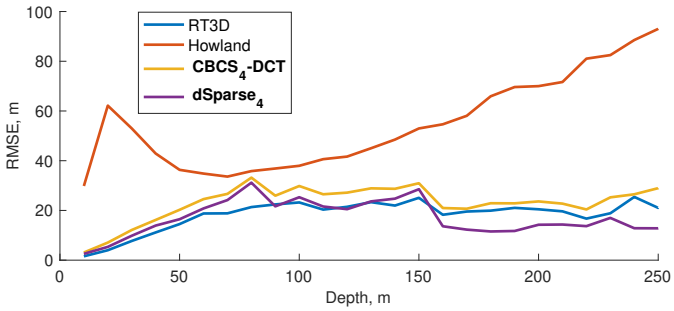


Fig. 9: RMSE as a function of range for our dataset, binned in 10 m increments. CBCS and dSparse outperform [8] dramatically with comparable performance to a full histogram processing approach [14] across this challenging dataset comprised of varying scene type.

Our framework can achieve 20 Hz with high sampling compression and 80 Hz for modest oversampling with flexible

illumination density running on 8 logical cores in parallel. Taking full advantage of the system architecture, we can theoretically reach extremely high processing rates of $1\,\mathrm{kHz}$, while enabling long exposure times for robust performance even in challenging situations. The framework performs well in all the presented scenarios, without any obvious degradation even at long distances, and no constraints on scene content or surface counts across the entire frame.

VI. Conclusion

We have presented a practical approach to real-time compressive depth sensing with an efficient sparse sampling scheme for long range LiDAR. The major limitations of prior art, namely the constraint to simple scenarios and very few surfaces and/or slow processing times are overcome by the use of a novel small-scale compressive depth framework.

A parallel block sparse LiDAR system architecture was presented to distribute the small-scale compressive depth reconstruction across a large solid-state photon detector array with a parallel processing architecture to enable effective exploitation of future large solid-state LiDAR arrays for high-resolution depth imaging. This architecture can scale well with resolution, as performance is block-size bound rather than tied to the problem size of the final frame resolution. Further, it accommodates a flexible random sparse illumination scheme allowing very low laser power per pattern exposure.

Our approach is capable of excellent depth reconstruction performance even at long range with scope for noise suppression and frame rates per block of over 80 Hz with very low processing latency of 10 ms without any major constraints on the application scope with a theoretical potential of < 1 ms processing time with dedicated per block processing logic.

The compressive sensing depth framework makes the major assumption of few surface returns in a small field-of-view and only returns a single depth value per pixel. It would be beneficial to extend this sparse depth framework to multi-return recovery. The basis transforms considered in this work were chosen primarily for speed and efficiency, but other basis functions, e.g. total generalized variation, may perform better and should be explored alongside efforts to reduce their computational and resource impact on the proposed system, for example with greedy optimization algorithms.

References

[1] A. M. Pawlikowska, A. Halimi, R. A. Lamb, and G. S. Buller, "Single-photon three-dimensional imaging at up to 10 kilometers range," *Optics Express*, vol. 25, no. 10, p. 11919, May 2017.

[2] C. Zhang, S. Lindner, I. M. Antolovic, J. M. Pavia, M. Wolf, and E. Charbon, "A 30-frames/s, 252 x 144 spad flash lidar with 1728 dual-clock 48.8-ps tdcs, and pixel-wise integrated histogramming," *IEEE Journal of Solid-State Circuits*, pp. 1–15, 2019.

[3] R. K. Henderson, N. Johnston, S. W. Hutchings, I. Gyongy, T. A. Abbas, N. Dutton, M. Tyler, S. Chan, and J. Leach, "A 256×256 40nm/90nm cmos 3d-stacked 120db dynamic-range reconfigurable time-resolved spad imager," in *IEEE International Solid-State Circuits Conference*, vol. 2019-Febru. Institute of Electrical and Electronics Engineers Inc., Mar. 2019, pp. 106–108.

[4] K. Morimoto, A. Ardelean, M.-L. Wu, A. C. Ulku, I. M. Antolovic, C. Bruschini, and E. Charbon, "Megapixel time-gated spad image sensor for 2d and 3d imaging applications," *Optica*, vol. 7, no. 4, 2020.

[5] S. M. Patanwala, I. Gyongy, H. Mai, A. Aßmann, N. A. W. Dutton, B. R. Rae, and R. K. Henderson, "A high-throughput photon processing technique for range extension of spad-based lidar receivers," *IEEE Open Journal of the Solid-State Circuits Society*, pp. 1–1, 2021.

[6] R. Thakur, "Scanning lidar in advanced driver assistance systems and beyond," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 48–54, 2016.

[7] P. T. Boufounos, "Depth sensing using active coherent illumination," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Mar. 2012, pp. 5417–5420.

[8] G. A. Howland, D. J. Lum, M. R. Ware, and J. C. Howell, "Photon counting compressive depth mapping," *Optics Express*, vol. 21, no. 20, p. 23822, Sep. 2013.

[9] M.-J. Sun, M. P. Edgar, G. M. Gibson, B. Sun, N. Radwell, R. Lamb, and M. J. Padgett, "Single-pixel 3d imaging with time-based depth resolution," *Nature Communications*, vol. 7, no. May, pp. 1–10, 2016.

[10] A. Aßmann, B. Stewart, J. F. Mota, and A. M. Wallace, "Compressive super-pixel lidar for high-framerate 3d depth imaging," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Ottawa, Canada, 2019.

[11] S. Hernandez-Marin, A. M. Wallace, and G. J. Gibson, "Multilayered 3d lidar image construction using spatial models in a bayesian framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1028–1040, 2008.

[12] 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 Transactions on Computational Imaging*, vol. 2, no. 4, pp. 1–1, 2016.

[13] 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 Transactions on Computational Imaging*, vol. 6, pp. 138–152, 2019.

[14] J. Tachella, Y. Altmann, N. Mellado, A. McCarthy, R. Tobin, G. S. Buller, J.-Y. Tourneret, and S. McLaughlin, "Real-time 3d reconstruction from single-photon lidar data using plug-and-play point cloud denoisers," *Nature Communications*, vol. 10, no. 1, p. 4984, Dec. 2019.

[15] M. Edgar, S. Johnson, D. Phillips, and M. Padgett, "Real-time computational photon-counting lidar," *Optical Engineering*, vol. 57, no. 03, p. 1, Dec. 2017.

[16] A. Colaco, A. Kirmani, G. A. Howland, J. C. Howell, and V. K. Goyal, "Compressive depth map acquisition using a single photon-counting detector: Parametric signal processing meets sparsity," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2012, pp. 96–102.

[17] Lu Gan, "Block compressed sensing of natural images," in *2007 15th International Conference on Digital Signal Processing*. IEEE, Jul. 2007, pp. 403–406.

[18] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proceedings - International Conference on Image Processing, ICIP*. IEEE, Nov. 2009, pp. 3021–3024.

[19] S. Pellegrini, G. S. Buller, J. M. Smith, A. M. Wallace, and S. Cova, "Laser-based distance measurement using picosecond resolution time-correlated single-photon counting," *Meas. Sci. Technol*, vol. 11, no. 00, pp. 712–716, 2000.

[20] S. Hernández-Marín, A. M. Wallace, and G. J. Gibson, "Bayesian analysis of lidar signals with multiple returns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2170–2180, 2007.

[21] N. Dutton, J. Vergote, S. Gnecchi, L. Grant, D. Lee, S. Pellegrini, B. Rae, and R. Henderson, "Multiple-event direct to histogram tdc in 65nm fpga technology," in *2014 Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, Grenoble, France, 2014.

[22] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[23] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[24] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[25] V. Chandrasekaran, B. Recht, P. Parrilo, and A. Willsky, "The convex geometry of linear inverse problems," *Found. Computational Mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[26] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[27] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, Mar. 2008.
[28] Tao Wan, N. Canagarajah, and A. Achim, "Compressive image fusion," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 1308–1311.
[29] M. A. Figueiredo and J. M. Bioucas-Dias, "Restoration of poissonian images using alternating direction optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3133–3145, Dec. 2010.
[30] J. Xu, J. Ma, D. Zhang, Y. Zhang, and S. Lin, "Improved total variation minimization method for compressive sensing by intra-prediction," *Signal Processing*, vol. 92, no. 11, pp. 2614–2623, Nov. 2012.
[31] S. Vishnukumar and M. Wilscy, "Single image super-resolution based on compressive sensing and improved tv minimization sparse recovery," *Optics Communications*, vol. 404, pp. 80–93, Dec. 2017.
[32] M. Vella and J. F. C. Mota, "Single image super-resolution via cnn architectures and tv-tv minimization," in *British Machine Vision Conference (BMVC)*, Birmingham, United Kingdom, Jul. 2019.
[33] C. Li, W. Yin, and Y. Zhang, "Tval3: Tv minimization by augmented lagrangian and alternating direction algorithms," 2010. [Online]. Available: http://www.caam.rice.edu/{∼}optimization/L1/TVAL3/
[34] J. E. Fowler, S. Mun, and E. W. Tramel, "Block-based compressed sensing of images and video," *Foundations and Trends in Signal Processing*, vol. 4, no. 4, pp. 297–416, 2012.
[35] A. Aßmann, Y. Wu, B. Stewart, and A. M. Wallace, "Accelerated 3d image reconstruction for resource constrained systems," in *2020 28th European Signal Processing Conference (EUSIPCO)*, Amsterdam, 2020, p. 565.
[36] E. J. Candes and J. K. Romberg, "Signal recovery from random projections," in *Computational Imaging III*, C. A. Bouman and E. L. Miller, Eds., vol. 5674. International Society for Optics and Photonics, Mar. 2005, p. 76.
[37] P. Chhabra, A. Maccarone, A. McCarthy, G. Buller, and A. Wallace, "Discriminating underwater lidar target signatures using sparse multispectral depth codes," in *2016 Sensor Signal Processing for Defence, SSPD 2016*, 2016.
[38] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Florence, Italy: Springer, Berlin, Heidelberg, 2012, pp. 746–760.
[39] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.
[40] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4340–4349.
[41] A. Aßmann, B. Stewart, and A. M. Wallace, "Deep learning for lidar waveforms with multiple returns," in *2020 28th European Signal Processing Conference (EUSIPCO)*, Amsterdam, 2020, p. 1571.
[42] S. Boyd, N. Parikh, E. Chu, and B. Peleato, "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.
[43] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
[44] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, ieee j," *Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.
[45] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*. IEEE Comput. Soc. Press, 1993, pp. 40–44.
[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[47] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in Neural Information Processing Systems*, vol. 3, no. January, 2014, pp. 2366–2374.
[48] W. Wagner, "Radiometric calibration of small-footprint full-waveform airborne laser scanner measurements: Basic physical concepts," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 6, pp. 505–513, 2010.
[49] D. L. Donoho and I. M. Johnstone, "Threshold selection for wavelet shrinkage of noisy data," *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, pp. A24–A25, 1994.

Andreas Aßmann (S'17-M'21) received the BEng (Hons) degree in Electrical and Mechanical Engineering from the University of Strathclyde, Glasgow, U.K., and his Doctor of Engineering (EngD) degree in Applied Photonics at Heriot-Watt University, Edinburgh, U.K., in 2021. He is a System Architect with STMicroelectronics R&D Ltd. at the Imaging Division in Edinburgh, U.K., since November 2020. His research interests include computational imaging and efficient signal processing for active imaging systems, in particular time-of-flight systems, and computer vision at large with an interest in compressive sensing and machine learning.

João F. C. Mota received the M.Sc. degree and the Ph.D. degree in Electrical and Computer Engineering from the Technical University of Lisbon in 2008 and 2013, respectively. He also received the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2013. He is Assistant Professor of Signal and Image Processing at Heriot-Watt University, Edinburgh. His research interests include theoretical and practical aspects of high-dimensional data processing, inverse problems, optimization theory, machine learning, data science, and distributed information processing and control. He was the recipient of the 2015 IEEE Signal Processing Society Young Author Best Paper Award.

Brian D. Stewart received his BSc in Electronics and PhD in Computer Vision from Dundee University in 1989 and 1992 respectively. During his time at STMicroelectronics R&D Ltd. he has focused on software development, modelling and image/signal processing and is a Design Architect within Imaging Division in Edinburgh, Scotland.

Andrew M. Wallace received his BSc and PhD degrees from the University of Edinburgh in 1972 and 1975 respectively. He is an Emeritus Professor of Signal and Image Processing at Heriot-Watt University. His research interests include LiDAR and 3D vision, image and signal processing, and accelerated computing. He has published extensively and has secured funding from EPSRC, the EU and other sponsors. He is a Chartered Engineer and a Fellow of the Institute of Engineering Technology.