Accelerated 3D Image Reconstruction for Resource Constrained Systems

Andreas Aßmann
EPS, Heriot-Watt University
STMicroelectronics R&D Ltd.
Edinburgh, UK
aa224@hw.ac.uk

Yun Wu
EPS, Heriot-Watt University
Edinburgh, UK
y.wu@hw.ac.uk

Brian Stewart
STMicroelectronics R&D Ltd.
Edinburgh, UK

Andrew M. Wallace
EPS, Heriot-Watt University
Edinburgh, UK

***Abstract*—We demonstrate an efficient and accelerated implementation of a parallel sparse depth reconstruction framework using compressed sensing (CS) techniques. Recent work suggests that CS can be split up into smaller sub problems. This allows us to efficiently pre-compute important components of the LU decomposition and subsequent linear algebra to solve a set of linear equations found in algorithms such as the alternating direction method of multipliers (ADMM). For comparison, a fully discrete least square reconstruction method is also presented.**

We also investigate how reduced precision is leveraged to reduce the number of logic units in field-programmable gate array (FPGA) implementations for such sparse imaging systems. We show that the amount of logic units, memory requirements and power consumption is reduced significantly by over 70% with minimal impact on the quality of reconstruction. This demonstrates the feasibility of novel high resolution, low power and high frame rate light detection and ranging (LiDAR) depth imagers based on sparse illumination.

***Index Terms*—Approximate Computing, FPGA, LiDAR, Compressed Sensing, Parallelization**

I. Introduction

The recovery of 3D information using light detection and ranging (LiDAR) at high resolutions and high frame rates is a challenging task. To achieve good signal-to-noise ratio (SNR) repeated illuminations with a laser require significant power and produce large amounts of raw data to be stored and processed with often expensive computational methods. This prohibits their use in resource constrained systems in particular when battery powered, e.g. drone scene mapping with simultaneous location and mapping (SLAM), augmented reality and virtual reality headsets and mobile phones. Depth reconstruction is often ill-posed and requires a solution of a large number of inverse problems. To enable better perception, the requirement for higher resolution in all three dimensions is obvious but requires significant compute power and memory to recover the desired information at scale and in real-time [1]. To maintain high frame rates, massively parallel compressed sensing (CS) could enable novel depth sensors to capture and recover 3D information fast enough for dynamic scenarios with lower amounts of measurements and thus memory to maximize limited data bandwidth. Sparse illumination further enables eye-safe long range low power LiDAR with block based parallelism [2]. However, the parallelism increases the complexity of the reconstruction process. Thus, each processing unit needs to have as few logic and memory gates as possible to minimize both power consumption and area.

Approximate computing (AC) is promising for energy efficiency in resource constrained systems and has been adopted for many areas such as signal processing [3], robotics [4], and machine learning [5]. Reduced precision (RP) computation for AC is one of the techniques that represents arithmetic data with less bits throughout the computational stack [6], specifically, memory and logic units. While [7] explored the fixed point precision ℓ_1-Solver for Model Predictive Control (MPC), the half floating point is investigated in [8] for 2D image recovery. When dealing with computational units in proximity to photon detectors it is crucial to decrease the processing chip power and thus direct heat to all photon sensitive elements to minimize device noise [9].

In this paper, we propose a ℓ_1 minimization based on RP for small scale basis pursuit de-noising (BPDN) or least absolute shrinkage and selection operator (*lasso*) problem formulations with an optimized lean alternating direction method of multipliers (ADMM) algorithm on a field-programmable gate array (FPGA) platform. This optimizer is suitable for compressive sensing image recovery, specifically time-of-flight depth sensing [2], [10]. We also apply RP to a discrete solver for sparsely illuminated depth sampling to investigate the impact of additional measurements over CS on the memory by applying RP. We analyze the effects on logic and memory resources with respect to their qualitative performance, as well as power savings and data compression throughout the sensing and compute stack.

Contributions. The contributions are summarized as follows:

- A lean discrete ADMM for compressive depth imaging, ℓADMM.
- A discrete least-squares method for sparsely illuminated depth recovery, dSparse.
- A comparative study of RP approximation effects on ℓADMM and dSparse for 3D image reconstruction.

This work was supported by STMicroelectronics R&D Ltd., the Engineering and Physical Sciences Research Council of the UK (EPSRC) under Grants EP/L01596X/1 & EP/S000631/1 and the UK MoD University Defence Research Collaboration (UDRC) in Signal Processing.

Before introducing the two discrete frameworks in Section III and the results in Section V, we introduce the problem and precision scaling below.

II. Background

A. Problem formulation

One promising approach to reduce large amounts of data in signal acquisition is CS. It exploits sparsity in the signal, $x \in \mathbb{R}^n$, and enables recovery of information from a smaller feature vector $y \in \mathbb{R}^m$, where $m < n$ with sampling constraints in the sensing matrix, $A \in \mathbb{R}^{m \times n}$, such that $y = Ax$ [11]. This problem can be recovered via BPDN i.e. re-formulated as

$$\min_x (\frac{1}{2}\|y - Ax\|_2^2 + \lambda \|Fx\|_1), \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ is the compressive sensing matrix, $\| \cdot \|_2^2$ is the Euclidean norm, $\lambda > 0$ is a regularization parameter to $\| \cdot \|_1 = \sum_i | \cdot_i |$ being the pseudo ℓ_1-norm and F a linear transformation. This particular formulation is also known as the *lasso* [12] or BPDN [13] and is useful in many other applications such as image reconstruction or de-noising [11], [12], [14], [15]. In this paper we investigate approximations to a popular algorithm to solve this optimization problem, ADMM [15], with the aim of reducing resource usage on FPGA type platforms to minimise logic units and memory and increase frame rate.

It was shown in [2] that this problem can be formulated as an ensemble of independent small blocks. We assume sparsity in the discrete cosine transform (DCT) domain encoded in A. To recover data, we apply Algorithm 2, to reconstruct two signals, x_Q, x_I, per block with identical A but independent measurement vectors y_Q, y_I. x_I is a vectorized intensity image, while x_Q is a proxy for intensity scaled distance per pixel as illustrated in Figure 1.

This yields a good estimate for depth as $x_D = x_Q./x_I$, if the assumption of a single primary reflective surface per pixel holds [2], [10]. It also reduces the amount of emitted laser power per measurement and only retains a small amount of data over traditional sampling methods.

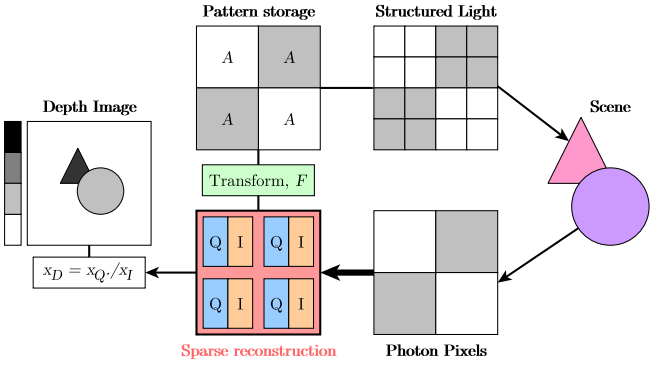
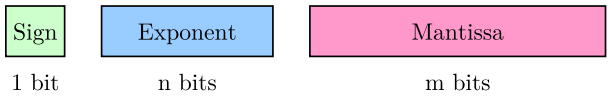


Fig. 1. Depth reconstruction using checkerboard compressive sensing (CBCS) [2] with low laser power and small memory requirements by using a CS depth reconstruction scheme. This work focuses on improving the highlighted parallel sparse reconstruction block by applying reduced precision to improve efficiency, compression and speed further.

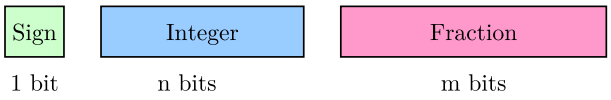
B. Reduced Precision Approximation

Reduced precision (RP) is an approximation technique where bit width in a data structure is shortened, enabling low cost and energy-efficient hardware implementations for computation and memory logic. By exploiting different arithmetic data types, it is expected to scale the area and power relative to the bit width and increase the density of operational units within the same area and power scope.

There are two major arithmetic data types: floating point and fixed point as shown in Figure 2. In iterative algorithms any



(a) Floating point representation (FP)



(b) Fixed point representation (FXP)

Fig. 2. (a) shows the structure of floating point. The number is represented by three parts, which are 1 bit sign, n bits exponent and m bits mantissa while the value is given by Sign $\times$ **M**antissa $\times$ $10^{\textbf{Exponent}}$. (b) shows the structure of fixed point. The number is represented by three parts, which are 1 bit sign, n bits integer and m bits fraction while the value is given by Sign $\times$ (**I**nteger + **F**raction).

approximation made by reducing the precision will cascade at each loop iteration and errors will accumulate. If the precision is not sufficient for the computations, the accumulated error will be unacceptable for the desired application. For depth imaging with sparse signal assumptions the iterative algorithm is performed for two separate quantities, which interact in the final recovery step. Thus, the effect of accumulated approximation effects is magnified. Specifically, with very low precision, significant signal components could be forced to zero too early and thus degrade consecutive computations. We aim to find the breaking points and optimal bit width for a lean ADMM framework and discrete sparse solver presented below and evaluate the effects of reduced precision on logic reduction, data compression, power, time and reconstruction quality.

III. Frameworks

A. ADMM

In the general case for BPDN, the standard ADMM implementation requires $O(n)$ flops with a soft-thresholding operator S and threshold λ/ρ shown in Algorithm 1 [15]. To increase overall stack efficiency, we identify components which is pre-computed and stored in look-up tables. It is obvious that the large matrix inversion term (Step 1 in Alg. 1) remains constant throughout and thus can be pre-computed. Further one can empirically find values for all regularization weights in this problem and determine a reasonable maximum iteration value, which allows to forego with computing the objective function after each iteration. We have modified

Algorithm 1 General ADMM *lasso* [15]

minimize $\frac{1}{2}\|y - Ax\|_2^2) + \lambda\|z\|_1)$
subject to $Fx - z = 0$
Iterate :
1: $x^{k+1} = (A^T A + \rho F^T F)^{-1}(A^T y + \rho F^T(z^k - u^k))$
2: $z^{k+1} = S_{\lambda/\rho}(Fx^{k+1} + u^k)$
3: $u^{k+1} = u^k + Fx^{k+1} - z^{k+1}$

an ADMM implementation from [15] and indicate what pre-computations are feasible for a compressive sensing problem, where we have full control over the sensing matrix, A. After careful analysis of the quadratic programming aspects of the ADMM implementation, inversion of A is often non-trivial but a LU decomposition is used and their inversions to compute the x-update. For a constant matrix A, these inversions can be pre-computed, $\{L, U\} \in \mathbb{R}^{m\times m}$, alongside $A^T y \in \mathbb{R}^n$ and $\lambda = \tau\max(|A^T y|)$, with τ for scaling. This allows us to break down and roll out the algorithm into discrete steps only involving simple matrix arithmetic as shown in Algorithm 2.

Algorithm 2 ℓeanADMM *lasso* for $m < n$

Input: $A, A^T y, L^{-1}, U^{-1}, y$
Output: x
Initialization : $const\{\alpha, \lambda, \rho, \kappa = \lambda/\rho\}, zeros\{z, q, u\}$
1: **for** $k = 0$ to k_{max} **do**
2: $q^{k+1} = A^T y + \rho(z^k - u^k)$
3: $x^{k+1} = q^{k+1}/\rho - 1/\rho^2 * A^T(U^{-1}(L^{-1}(Aq^{k+1})))$
4: $\hat{x}^{k+1} = \alpha x^{k+1} + (1-\alpha)z^k$
5: $xu^{k+1} = \hat{x}^{k+1} + u^k$
6: $z_1^{k+1} = \max\{0, xu^{k+1} - \kappa\}$; $z_2^{k+1} = \max\{0, -xu^{k+1} - \kappa\}$
7: $z^{k+1} = z_1^{k+1} - z_2^{k+1}$
8: $u^{k+1} = u^k + (\hat{x}^{k+1} - z^{k+1})$
9: **end for**
10: $x = z^{k_{max}}$
11: **return** x

B. Pseudo-inverse least-squares: dSparse

In this paper we assume a fairly small problem size with a block image vector size of $n = 16$. Given such small block sizes, one could also afford to make more than n measurements without the usual compressive sensing penalties of very large sensing matrices with too many measurements for real time operation. With $m > n$ measurements e.g. $m = 48$, the problem is oversampled but we retain reductions in emission power for each sparse measurement. We enforce that all pixels are covered at least once in the pseudo random binary pattern sequence, A_d, to compute a pseudo-inverse. The linear system of equations can be then solved with an approximated least squares solution. Let $A_{aux} = (A_d^T A_d)^{-1} A_d^T \in \mathbb{R}^{n\times m}$, then an image vector is simply

$$x = A_{aux} y. \tag{2}$$

with a pre-generated pattern sequence defining $A_d = [(A_d)_1, ..., (A_d)_m]$, A_{aux} is readily pre-computed and stored alongside A. The sequence of sparse measurements constitutes a linear set of equations, where each pixel solution is a discrete estimate for the centre-of-mass as a depth estimate. Due to having many contributions from various pixel combinations, there is more information available for depth estimation than relying on a single pixel sampling methodology. We call this approach dSparse. Furthermore, sampling time being a critical parameter, we note that this sparse oversampling approach will have a constant sampling time as resolution increases, while linear scans will continue to scale up. We illustrate this in a thought experiment.

Let's assume a system performs a line scan containing 1024 pixels and scans across 48 rows. In this case the linear scan and the sparse sampling will take the same amount of time. But, as the number of rows increases, so does the sampling time for the linear scan. Our sparse sampling approach remains constant. It also enables to scan the entire scene simultaneously, which in theory should minimise any motion artefacts in the sampling procedure over linear scans. As array sizes increase, this will have to be taken into consideration.

To enable high frame rate depth image reconstruction for high dynamic range and effective real-time operation of arrayed photon imagers such as presented in [16], sparse sensing helps to overcome power and data bandwidth limitations. Both cases $m < n$ and $m > n$ are massively parallelized as suggested in [2] and more memory efficient than per pixel histogram sampling. In non-compressive systems each pixel stores photon counts in a histogram, $h \in \mathbb{N}^l$, where l is the number of bins. To work with such measurements, the final container is $H \in \mathbb{R}^{n\times l}$, where l ranges from tens to thousands of bins depending on the system's timing resolution. This put significant strain on resource allocation. The parallelization of the problem into blocks enables fast processing times by solving many small problems rather than one big problem. Sparse depth imaging further helps to reduce total power draw both for emitter and reconstruction. This is especially important for photon sensitive measurements, where noise is introduced as temperature increases [9].

By investigating effects of reduced precision we aim to find out how approximate computing (AC) enables a flexible design for a parallel sparse reconstruction unit to enable efficient arrayed 3D imagers for LiDAR type applications. This can both enable a resource efficient and low power compressive depth imager for a discrete active sparse imager with a small footprint enabled by savings from reduced precision (RP). The hardware architectures and trade-offs is evaluated through re-configurable devices, such as FPGAs.

TABLE I
ARITHMETIC COMPLEXITY

	ℓADMM	dSparse
$\pm$	$mn + \sum_i^k (2mn + 11n)_i$	$2mn$
$\times$	$mn + \sum_i^k (2mn + 6n)_i$	$2mn$
memory access	$3mn + \sum_i^k (6mn + 43n)_i$	$4mn$

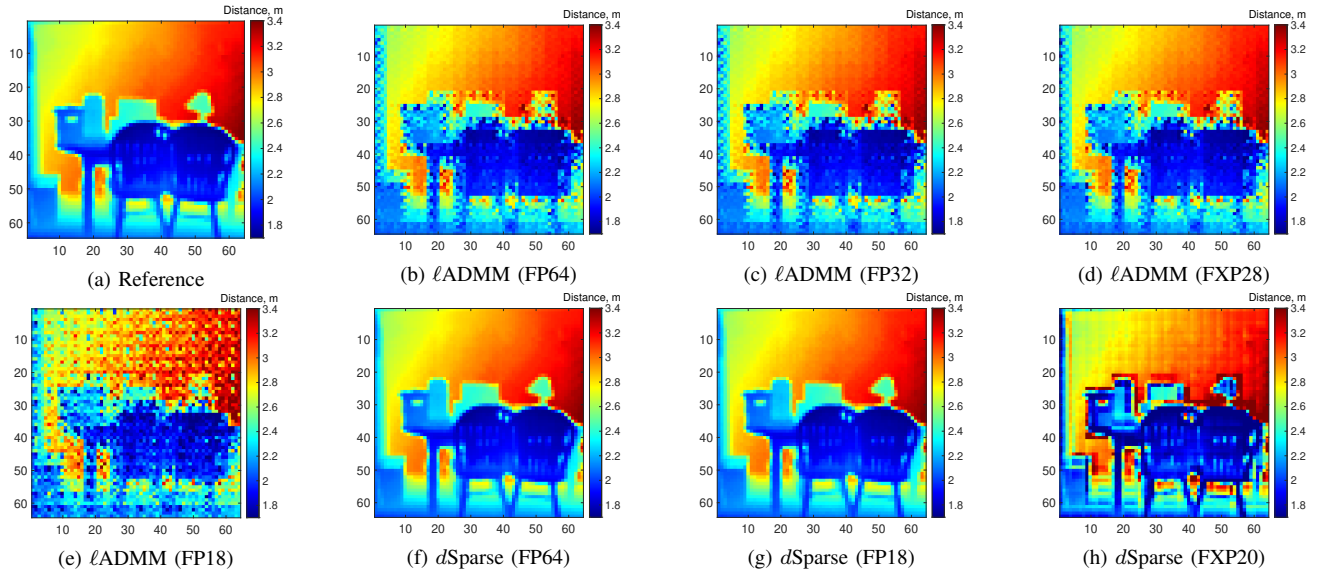
IV. Analysis

To evaluate the trade-off between computational complexity and reconstruction quality of the two algorithms, we evaluate the arithmetic complexity in Table I. Functions for

TABLE II

REDUCED PRECISION EFFECTS ON RESOURCE AND QUALITY METRICS FOR ℓADMM CS AND dSPARSE DISCRETE 3D IMAGE RECONSTRUCTION. FOR CUSTOM FLOATING POINT, BIT WIDTH = 1+E+M AND FOR CUSTOM FIXED POINT, BIT WIDTH = 1+I+F.

			Resources			Precision and Timing			Metrics				
		Fig.	LUT $10^3\times$	DSP	BRAM	Bit Width (bits)	Clock (MHz)	Latency $10^3\times$(cycles)	Dynamic Power (W)	Data Ratio	Time (ms)	PSNR (dB)	RMSE
ℓADMM $m=8$ $n=16$	FP64	3(b)	14.099	82	38	(1,52,11)	2.543	6.357	1.139	50.00%	2.500	23.99	0.21
	FP32	3(c)	6.375	36	4	(1,8,23)	2.156	5.377	0.460	25.00%	2.494	23.99	0.21
	FP18	3(e)	4.383	8	0	(1,6,11)	2.284	5.230	0.306	14.06%	2.290	17.98	0.43
	FP12	-	3.297	8	0	(1,5,6)	2.036	4.640	0.188	9.38%	2.279	4.54	2.00
	FXP28	3(d)	2.928	81	4	(1,17,10)	3.007	3.758	0.265	21.88%	1.250	24.06	0.21
	FXP23	-	2.005	57	4	(1,13,9)	2.736	3.758	0.188	17.97%	1.374	4.15	2.08
	FXP20	-	1.879	32	4	(1,11,8)	2.756	3.743	0.144	15.62%	1.358	3.26	2.26
dSparse $m=48$ $n=16$	FP64	3(f)	5.112	20	5	(1,52,11)	1.814	17.251	0.553	300.00%	9.510	307.88	0.00
	FP32	-	1.443	8	2	(1,8,23)	1.906	11.795	0.139	150.00%	6.188	134.48	0.00
	FP18	3(g)	0.753	2	1	(1,6,11)	1.874	15.282	0.057	84.38%	8.155	31.33	0.09
	FP12	-	0.564	2	0	(1,5,6)	1.684	15.283	0.040	56.25%	9.075	19.84	0.34
	FXP28	-	1.346	1	3	(1,17,10)	2.890	2.721	0.046	131.25%	0.942	25.43	0.18
	FXP23	-	1.272	1	3	(1,13,9)	2.858	2.625	0.043	107.81%	0.918	22.20	0.26
	FXP20	3(h)	1.299	1	1	(1,11,8)	2.864	2.177	0.034	93.75%	0.760	15.89	0.55



(a) Reference (b) ℓADMM (FP64) (c) ℓADMM (FP32) (d) ℓADMM (FXP28)

(e) ℓADMM (FP18) (f) dSparse (FP64) (g) dSparse (FP18) (h) dSparse (FXP20)

Fig. 3. Comparative figure illustrating various bit widths and their visual effects on final depth map reconstruction of simulated photon count data for a real depth scene from [17]. We show the breaking points for ℓADMM ($\rho = 1.2, \alpha = 1.4, \tau = 10^{-9}$) in (e) and dSparse in (h) for 18 bit floating and 20 bit fixed point representations respectively. We note that block artefacts in ℓADMM are primarily due to chosen basis and an identical pattern sequence for all blocks.

measurements, m, signal size, n and maximum iterations k are illustrated. As shown, considering the same m and n, the pseudo-inverse dSparse approach has a fixed number of operations and is of lower complexity compared to ℓADMM which scales with number of iterations.

It is clear that the block size and number of measurements will increase the complexity of ADMM significantly, in particular if the problem is scaled to full image resolutions rather than small blocks where the number of measurements is roughly $m = O(Slog(n/S))$, where S is the number of non-zero entries in pattern A. For illustrative purposes let $S = n/2$. Rather than solving (1) for $m_B = 8$ or less in parallel for each block ($n_B = 16$), a full frame image ($n_F = 64 \times 64 = 4096$) image would require $m_F > 600$ measurements, making the reconstruction via ADMM expensive and storage of A challenging. Some CS implementations rely on implicit generation of the sensing matrix, but adding more computational steps [18]. In general, CS techniques will reduce the amount of sampled data by increasing the computational complexity in the reconstruction. We try to reduce this impact on complexity, and thus resources, by applying reduced precision.

V. EVALUATION AND RESULTS

To demonstrate the effects of RP, we use simulated photon count data based on underlying real depth scenes from [17] and re-sample them in a compressive fashion using pseudo-random binary patterns. All datatypes were implemented in C++ and synthesized for a Xilinx FPGA UltraScale+ ZCU106 using Vivado 2019.1. We show the results for a variety of custom floating point and fixed point representations for both example applications in Table II. All comparative metrics are compared to full double precision and data ratio=$(m/n)(\text{bits}/64)$.

For ℓADMM we see dramatic savings of above 80% in resources with both custom data types for LUTs showing the significant saving potentials before the reconstruction quality degrades as shown in Figure 3(d). For ℓADMM 28 bit fixed point (FXP28) is the lowest precision before degradation occurs with the least resource usage. Additionally, when using fixed point representations we see a great reduction in latency by a factor of 2, with total computation times as low as

$1.25\,\mathrm{ms}$ per block for $k = 5$ iterations. We also see additional data compression with as low as 14% data usage versus full block non-compressive sampling. This is further enhanced by reducing the number of measurements for extremely high frame rates and very low sampling times.

In the discrete pseudo-inverse case, we observe similar trends for logic resources while the extra measurements memory can be offset by approximation to realise only 85% data usage over linear sampling at double precision. The latency is also significantly reduced with a fixed point data structure, making 23 bit fixed point (FXP23) the most efficient implementation. This demonstrates the potential for very fast ($< 1\,\mathrm{ms}$) depth computations using a sparse illumination approach with excellent reconstruction quality.

Using Vivado power estimation tools, the FPGA system has a static power consumption baseline of roughly $0.6\,\mathrm{W}$ consistent for both algorithms. We analyse the power savings for the dynamic power consumed above this baseline. For ℓADMM the power draw is reduced by 76% and for dSparse by over 90% before the reconstruction quality is adversely affected.

Both application examples demonstrate that significant reductions in resources, data usage, power and latency are achieved using RP with allowing for high frame rates of well above $500\,\mathrm{Hz}$ for ℓADMM and over $1\,\mathrm{kHz}$ for dSparse. This is comparable to x86 processing times presented in [2] but on real hardware. This result clearly illustrates the effectiveness of RP on sparse depth reconstruction. It enables great flexibility in system design for 3D imaging systems with control over resources and power efficiency using approximate computing techniques to counteract the increased cost of applying CS data acquisition while reducing the potential negative effects of heat to the photon sensitive elements when in close proximity.

VI. Conclusion

We have presented the effects of applying reduced precision as an approximate computing technique to depth reconstruction using small scale ℓ_1 (ℓADMM) blocked compressive sensing as well as a discrete least-square solver (dSparse).

With the optimal reduced precision for either approach with minimal reconstruction degradation, we achieve significant savings in logic resources versus double precision floating point arithmetic with much improved latency using fixed point data types. In both cases we achieve very fast per block computation times and significant power savings. In the compressive sensing case, we further achieve additional data savings reducing the overall data bandwidth even further. This demonstrates a pathway to dedicated logic for depth computation for resource constrained devices. This approach shows potential ways of mitigating the computational complexity of compressive sensing techniques, while benefiting from low power sparse illumination and lowering the data bandwidth throughout the computation stack.

We show that the combination of sparse imaging and reduced precision can enable power efficient, high resolution and high frame rate time-of-flight depth imaging for application in resource constrained devices.

References

[1] Julián Tachella, Yoann Altmann, Nicolas Mellado, Aongus McCarthy, Rachael Tobin, Gerald S. Buller, Jean-Yves Tourneret, and Stephen McLaughlin, "Real-time 3D reconstruction from single-photon lidar data using plug-and-play point cloud denoisers," *Nature Communications*, vol. 10, no. 1, pp. 4984, dec 2019.

[2] Andreas Aßmann, Brian Stewart, Joao F.C. Mota, and Andrew 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.

[3] K. Roy and A. Raghunathan, "Approximate computing: An energy-efficient computing technique for error resilient applications," in *2015 IEEE Computer Society Annual Symposium on VLSI*, July 2015, pp. 473–475.

[4] P. Pandey, Q. He, D. Pompili, and R. Tron, "Light-weight object detection and decision making via approximate computing in resource-constrained mobile robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 6776–6781.

[5] Ali Ibrahim, Mario Osta, Mohamed Alameh, Moustafa Saleh, Hussein Chible, and Maurizio Valle, "Approximate computing methods for embedded machine learning," in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 12 2018, pp. 845–848.

[6] Ankur Agrawal, Jungwook Choi, Kailash Gopalakrishnan, Suyog Gupta, Ravi Nair, Jinwook Oh, Daniel A. Prener, Sunil Shukla, Vijayalakshmi Srinivasan, and Zehra Sura, "Approximate computing: Challenges and opportunities," *2016 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–8, 2016.

[7] Ahmad Rateb, Sharifah Kamilah, and Rozeha A Rashid, "A generic top-level mechanism for accelerating signal recovery in compressed sensing," *Digital Signal Processing*, vol. 69, 07 2017.

[8] Adrian Wills, Adam Mills, and Brett Ninness, "Fpga implementation of an interior-point solution for linear model predictive control," *18th IFAC World Congress*, 01 2011.

[9] Eric A.G. Webster, Lindsay A. Grant, and Robert K. Henderson, "A high-performance single-photon avalanche diode in 130-nm CMOS imaging technology," *IEEE Electron Device Letters*, vol. 33, no. 11, pp. 1589–1591, nov 2012.

[10] Gregory A. Howland, Daniel J. Lum, Matthew R. Ware, and John C. Howell, "Photon counting compressive depth mapping," *Optics Express*, vol. 21, no. 20, pp. 23822, sep 2013.

[11] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, aug 2006.

[12] Robert Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, pp. 267–288, 1996.

[13] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[14] Mario A. T. Figueiredo, Robert D. Nowak, and Stephen 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.

[15] Stephen Boyd, Neil Parikh, Eric Chu, and Borja 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.

[16] Sarrah. M. Patanwala, Istvan Gyongy, Neale A. W. Dutton, Bruce. R. Rae, and Robert. K. Henderson, "A Reconfigurable 40nm CMOS SPAD Array for LiDAR Receiver Validation," in *International Image Sensor Workshop*, 2019.

[17] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in *European Conference on Computer Vision*, Florence, Italy, 2012, pp. 746–760, Springer, Berlin, Heidelberg.

[18] Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Jason N. Laska, Ting Sun, Kevin F. Kelly, and Richard G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, mar 2008.