A Sparse Convolution Neural Network Accelerator for 3D/4D Point-Cloud Image Recognition on Low Power Mobile Device with Hopping-Index Rule Book for Efficient Coordinate Management

Qiankai Cao, Jie Gu
Northwestern University, Evanston, IL, USA

Abstract - This work presents the first 3D/4D sparse CNN (SCNN) accelerator for point cloud image recognition on low power devices. A special hopping-index rule book method and efficient data search technique were developed to mitigate the overhead of coordinate management for SCNN. A 65nm test chip for 3D/4D images was demonstrated with 7.09–13.6 TOPS/W power efficiency and state-of-the-art frame rate.

Introduction

Recently efficient hardware support of 3D/4D imaging for AR/VR applications has become critically important with LIDAR sensors being available for smartphone or tablet. Previously, a point cloud-based neural network (PNN) processor for simpler hand gesture recognition was developed through approximate sampling and grouping in 2D projected point-cloud, which may cause uneven distribution of sampled points with poor accuracy for large-scale 3D images [1]. A more recent work proposed a page-based memory management technique to handle non-uniform distribution of point cloud using page ID [2]. However, it is still based on dense format, hence incurring higher computing cost and data storage. As shown in Fig. 1, compared with 2D case, applications in 3D/4D space experience exponentially increase of the computation workload while also observe dramatic increase of sparsity, e.g. 97.5% in 3D or 99.9% in 4D cases. As in Fig. 1, considering the computation overhead of index/coordinate management for sparse input format, SCNN becomes advantageous at sparsity beyond 30%~40%, with benefits increasing with sparsity level. Hence, fundamentally, SCNN provides more efficient solution for high dimensional sparse images. Although a simulation based SCNN was proposed earlier for 2D image, it cannot be directly applied to 3D/4D point-cloud images [3]. This work presents the first comprehensive solution for SCNN for 3D/4D image recognition. As highlighted in Fig. 1, the contributions of this work are: (1) A 3D/4D SCNN accelerator based on the widely used record-holding Minkowski engine [4] was implemented on the silicon achieving state-of-the-art performance compared with prior 3D cases [2]; (2) A hardware friendly "rule book" solution for SCNN is developed leading to a speedup of 89.3X for 3D and 270.1X for 4D cases compared with conventional dense CNN; (3) To mitigate overhead of the coordinate management for SCNN, a hardware-efficient coordinate generation and search solution with octree data structure and computation skipping method are implemented rendering 12X speedup enhancing the benefits of sparse convolution; (4) A look-up table (LUT) based weight reuse scheme is utilized to reduce weight duplications leading to 26.9X saving of memory space.

SCNN Algorithm and Hardware Implementation

Fig. 2 shows the 3D/4D point cloud processing sequence and the chip architecture which contains (1) a 10 x 10 PE array as central compute engine, (2) a top controller for data flow management, (3) output accumulation and post processing modules for SCNN, (4) various memory banks with special indexing schemes to support the rule book and SCNN operations. The operation sequence includes coordinate management for rule book generation and subsequent sparse convolution, where input point cloud images are divided into sub-spaces for processing by the chip. The 8-bit reconfigurable PE array is designed to support both coordinate management and SCNN for compact chip implementation.

Fig. 3 shows details of the "rule book" flow at the core of SCNN and hardware implementation. For 3D/4D SCNN, the input pixels are saved with coordinate (X, Y, Z, T) associated with feature values eliminating the massive redundant zeros in the 3D/4D space. In SCNN mode, PE array is configured into MAC operations processing sparse inputs and kernel values. As spatial relationship is lost in sparse coding, a special map representing coordinate relationship is needed, referred as "Rule Book" [4]. A software implementation of Rule Book with expensive hash function is unsuitable for small-size ASIC accelerator due to the overwhelming memory operations for keyword search and high computation cost of hash functions. Hence, an efficient hardware friendly "hopping-index rule book" (HIRB) is developed in this work with multiple hopping of memory banks through use of data indexes. As shown in Fig.3, first, sparse inputs are loaded into PE array for MAC operation and the last 16-bits "end" address is sent to index memory to provide the stop address for current input. Second, the core index memory performs loading of multiple weights of different outputs for the same input until stop address is reached. Third, for MAC operation, to fetch the weight value accordingly, instead of duplicating the channel-wise weights, an 8-bit index indicating mapping between kernel and input shared by all channels is used so that weights stored in LUT are fetched according to index, rendering 13.5X~26.9X memory saving varying with channel number as in Fig. 1. Fourth, the multiple MAC outputs from the same input point are stored into different target addresses according to HIRB. This data flow not only solves the irregular sparse convolution data mapping, but also provide a general SCNN solution for variable dimensions, 2D/3D/4D and beyond.

Fig. 4 shows implementation of coordinate manager which is used to generate the HIRB for building spatial/temporal relationship among sparse points. Distance information between 3D/4D points are calculated by the PE array. Only two points with a distance lower than a threshold are recorded as neighbors denoted in the HIRB. As a brute-force sequential search incurs high compute cost, an octree data structure and data skipping technique are used to accelerate the operations. With entire space divided into subspace from the octree data structure and the sparse data stored in incremental orders of X, Y, Z, T, neighborhood searching is significantly narrowed. Moreover, partial distances in Z axis or T axis are calculated first and skipped if the partial distance is larger than a threshold. Overall, the use of octree data structure and distance skipping lead to a 12X saving on computing cost reducing coordinate management overhead from 67.5% to 14.7% of total operation further enhancing the benefits of SCNN.

Measurement Results

A 65nm test chip was fabricated and measured for the proposed SCNN accelerator as in Fig. 5. The chip operates from the nominal 300MHz/1V to 50MHz/0.5V with efficiency

from 0.78TOPS/W to 1.5TOPS/W without considering sparsity or 7.09TOPS/W to 13.6TOPS/W considering sparsity for 8-bit SCNN. The coordinate management takes 14.7% of runtime consuming 35% less power than SCNN. Fig. 5 shows examples of 3D/4D segmentation and corresponding mIOU scores for 3D and 4D point cloud images as well as the accuracy of this work on database ScanNet [5] and Synthia4D [6] with only 0.1% accuracy loss compared with FP results. Compared with conventional dense CNN, a speedup of 89.3X for 3D image or 270.1X for 4D image is achieved. Fig.6 shows comparison with prior point-cloud works. This is the first sparse convolution accelerator targeting 3D/4D point-cloud image/videos. While the raw framerate of 7.2fps is lower than [2] due to 5X larger CNN model size and 15X smaller PE array used in this work, a 7.5X higher framerate is achieved when normalized to similar model size and PE array due to the significant runtime reduction of sparse convolution. In addition, this is the only accelerator that also handles 4D point-cloud videos.

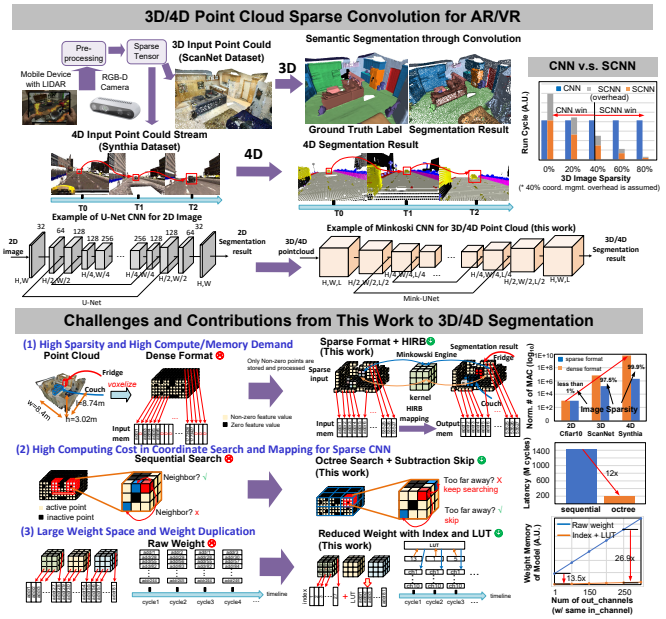


Fig. 1 3D/4D point cloud image recognition tasks, challenges and contributions from this work.

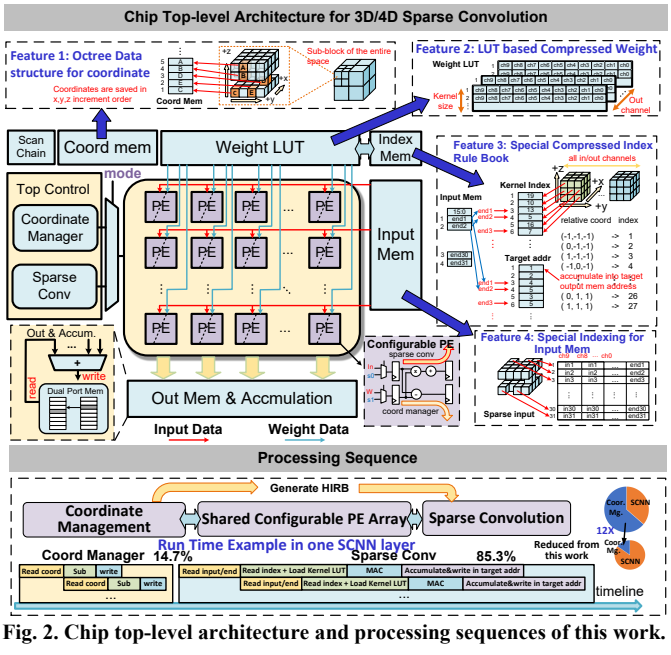


Fig. 2. Chip top-level architecture and processing sequences of this work.

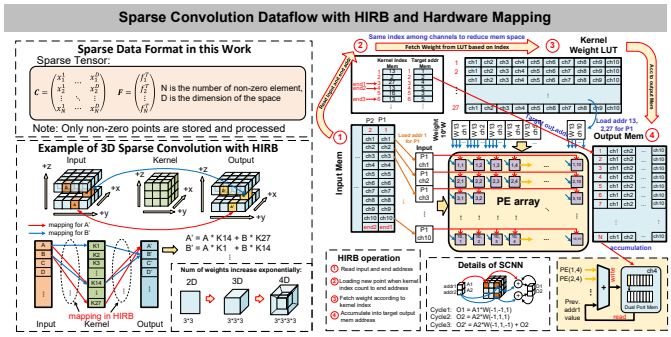


Fig. 3 Detailed description on SCNN operation, developed hopping-index rule book (HIRB) and hardware mapping in this work.

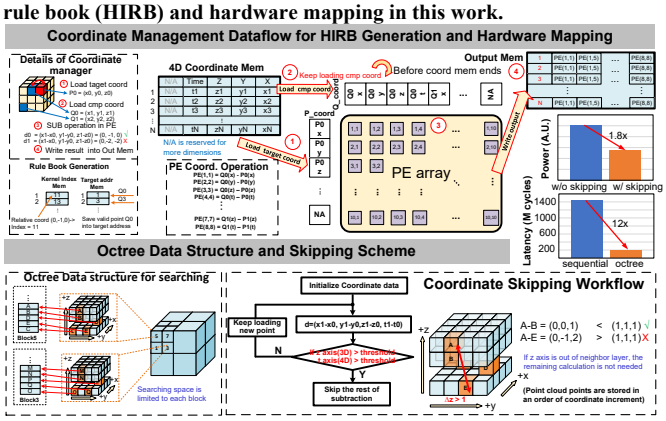
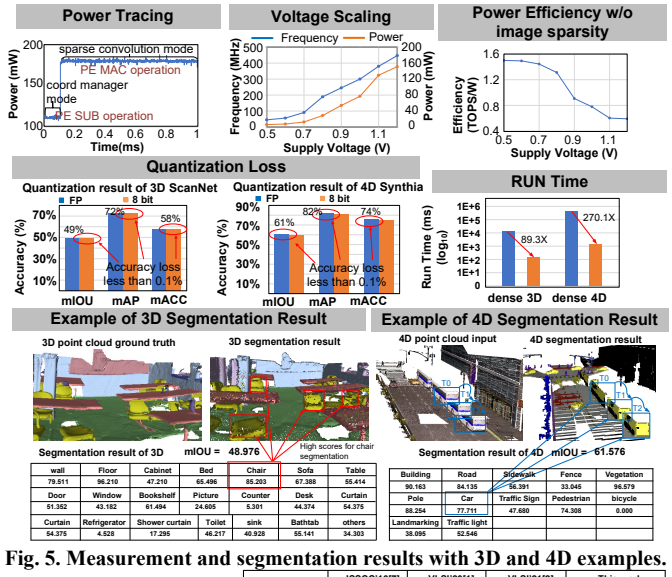


Fig. 4. Chip implementation of coordinate management for HIRB generation and associated speedup techniques.



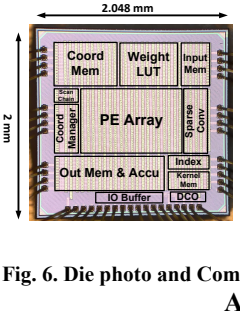
wall	Floor	Cabinet	Bed	Chair	Sofa	Table
79.511	96.210	47.210	65.496	85.203	67.388	55.414
Door	Window	Bookshelf	Picture	Counter	Desk	Curtain
51.352	43.182	61.494	24.605	5.301	44.374	54.375
Curtain	Refrigerator	Shower curtain	Toilet	sink	Bathtab	others
54.375	4.528	17.295	46.217	40.928	55.141	34.303

Segmentation result of 3D mIOU = 48.976

Building	Road	Sidewalk	Fence	Vegetation
90.163	84.135	56.391	33.045	86.579
Pole	Car	Traffic Sign	Pedestrian	bicycle
88.254	77.711	47.680	74.308	0.000
Landmarking	Traffic light			
38.095	52.546			

Segmentation result of 4D mIOU = 61.576

Fig. 5. Measurement and segmentation results with 3D and 4D examples.



	ISSCC'16[7]	VLSI'20[1]	VLSI'21[2]	This work
Dimensionality	2D	3D	3D	3D/4D
Process	65 nm	65 nm	65 nm	65 nm
Area (mm2)	3.5 x 3.5	16	16	4.1
Architecture	Eyeriss	PNN	PNN	SCNN + Coord Manager
Datatype	INT16	INT8	INT8	INT8
PE number	168	--	8*8*6*4	100
Frequency (MHz)	100- 250 MHz	150 MHz	200 MHz	300MHz
Supply Voltage(V)	0.82– 1.17	0.78– 1.2	0.78– 1.1	0.5– 1.2
On chip Memory	181.5 kB	362 kB	364 kB	108.5 kB
CNN Layer Power Efficiency (TOPS/W)	0.241 (1V, 16b)	5.62(1V, 8b)	2.59- 5.65	0.78[3]– 7.09[2] @1V 1.5[3]– 13.6[2] @0.5V
3D Framerate (fps)	--	≤ 44.4	84.8	7.2[3] (raw fps) 110.6[4] 638.9[5]
4D Framerate (fps)	--	--	--	0.69[6]

1) w/o considering input sparsity, 2) w/ considering input sparsity, 3) Minkowski@ SUN RGB-D, 4) Normalized with number of PEs, 5) Normalized with layers and number of PEs, 6) Minkowski@Synthia4D (25 3D image per 4D point cloud), 7) excluding IO loading time

Fig. 6. Die photo and Comparison table.

Acknowledgements

This work was supported in part by NSF under grant number CCF-1846424.

Reference

[1] S. Kim, et al. *VLSISymp, 2020*
[2] D. Im, et al. *VLSISymp, 2021*
[3] A. Parashar, et al. *ISCA, 2017*
[4] C. Choy, et al. *CVPR, 2019*
[5] A. Dai, et al. *CVPR, 2017*
[6] G. Ros, et al. *CVPR, 2016*
[7] Y. Chen, et al. *ISSCC, 2016*
[8] S. Song, et al. *CVPR, 2015*