Slimmer: Accelerating 3D Semantic Segmentation for Mobile Augmented Reality

Huanle Zhang[#], Bo Han[&], Cheuk Yiu Ip[*], Prasant Mohapatra[#]

[#]University of California, Davis
[&]George Mason University
[*]AT&T Labs - Research

3D Semantic Segmentation & Applications

Definition: 3D segmentation is a process where a given 3D input (e.g., 3D mesh or a **point cloud**) is divided into partitions that share the same local properties[1]

A point cloud

Scene Segmentation

An essential building block of Augmented Reality (AR). For example, a user

1. “Moves” objects and visualize how the scene looks like without actually moving them
2. “Plays” with objects in the scene
3. “Controls” objects by making a gesture
4. “Merges” objects into Virtual Reality (VR)

1. L.E. Carvalho, et al. 3D Segmentation Algorithms for Computerized Tomographic Imaging: a Systematic Literature Review. Journal of DIgital Imaging, 2018

Measurement of 3D Semantic Segmentation Model

Measurement Setup

- SparseConvNet[1]: One of the sparse convolutional networks
- ScanNet[2]: 3D Indoor scene segmentation dataset
- Dell Alienware laptop (6-core 2.90GHz i9 CPUs, 16GB RAM)

Measurement Metrics

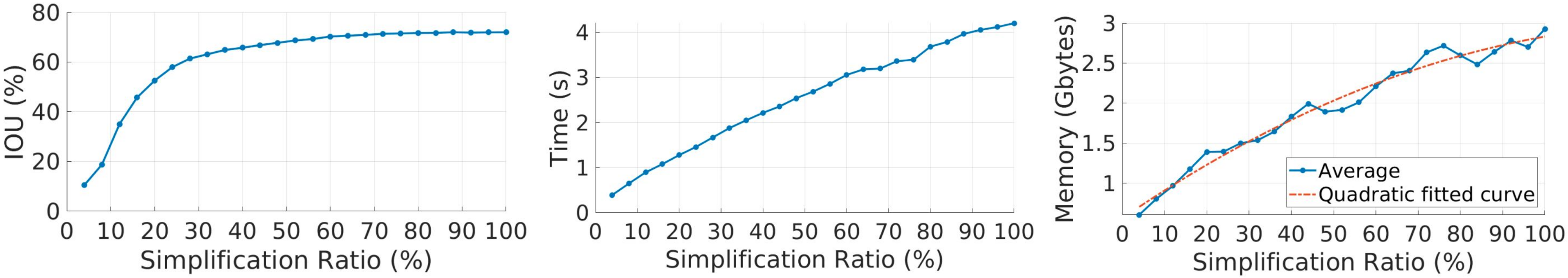
- Inference time
- Memory usage
- Accuracy: Intersection Over Union (IOU)

4.21 seconds, 2.83GB memory, 71.18% IOU per point cloud

Too costly for Mobile Devices

1. B Graham, et al. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In CVPR. 2018
2. ScanNet dataset. http://www.scan-net.org/

Motivation: Overheads of a Pre-trained DNN Model Grow Linearly with the Number of Points in the Input



(a) Accuracy (b) Inference Time (c) Memory Usage

Figure: Performance of the DNN model over sparsified point clouds

The pre-trained DNN model is untouched. We only sparsify the input point cloud

- **Model Accuracy**: IOU remains almost the same even when only circa 60% points are used.
- **Inference Time**: Inference time is approximately linearly correlated with the simplification ratio.
- **Memory Usage**: Memory usage is approximately linearly correlated with the simplification ratio.

Slimmer: Accelerating 3D Semantic Segmentation for Mobile Augmented Reality

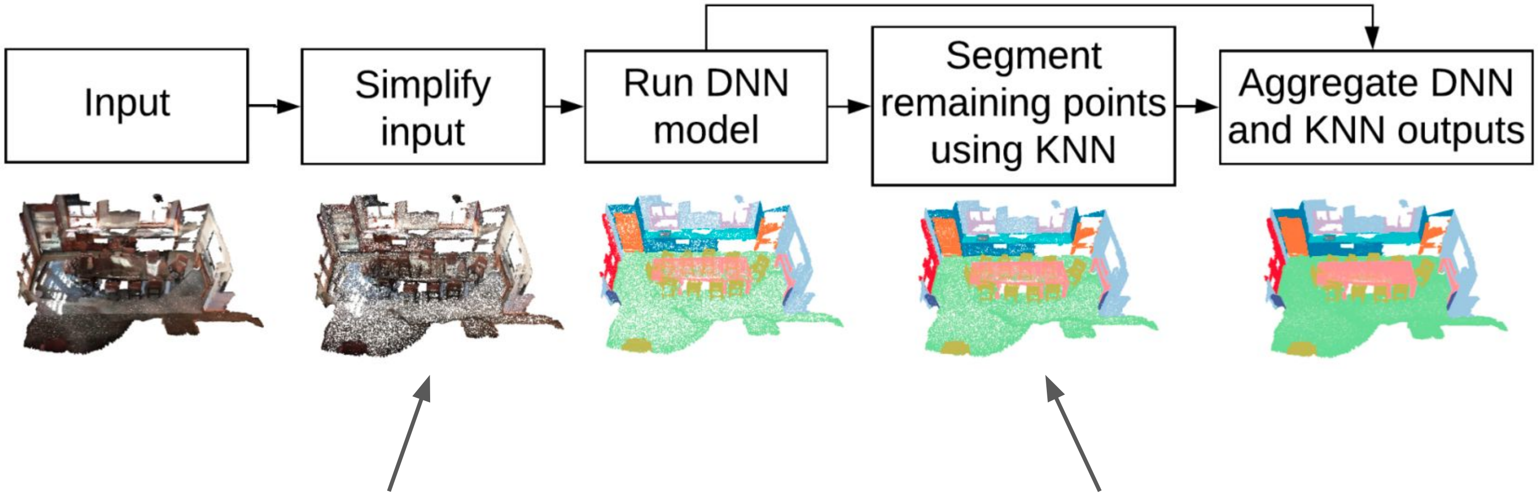
A generic and model-independent framework, for accelerating 3D semantic segmentation.

Idea: remove a fraction of points in the input, while keeping the pre-trained DNN models untouched.

Challenges

1. Determining a lightweight simplification method to sparsify the point clouds
2. A lightweight method to segment the removed points from the original full-size input

System Architecture of Slimmer



Core Component 1:
Simplifying Point Clouds

Core Component 2:
Segmenting the Removed Points

Component 1: Simplifying Point Clouds

Representative Simplification Methods

1. **Random** Simplification: Each point is independently kept with a given probability. Regards each point equally
2. **Grid** Simplification: Each point cloud is partitioned into grid cells of a given size. A point is randomly selected among the points in that cell. Favors sparse points than dense points
3. **Hierarchy** Simplification: An adaptive simplification through local clusters, which recursively splits the point set into smaller clusters until the clusters have less than a given size. Favors edge points than surface points.

Component 2: Segmenting the Removed Points

K-Nearest-Neighbor (KNN) to segment the removed points.

- For each removed point, we propose to infer its label by the majority label of its nearest neighbors that are in the simplified point cloud.

Algorithm of segmenting the removed points

n_t Number of points of a point cloud; n_s Number of points of the simplified point cloud; n_r Number of removed points

1. Construct a k-d tree (*k* = 3) for the simplified point cloud. $\mathcal{O}(n_s \cdot \log n_s)$
2. For each removed point, search its K nearest points in the k-d tree. $\mathcal{O}(n_r \cdot \log n_s)$
3. For each removed point, assign the majority label of its neighbors $\mathcal{O}(n_r)$

Total complexity = $\mathcal{O}(n_t \cdot \log n_t)$

QoE Improvement based on Simplification Ratio

Smaller simplification ratio -> Inference Time ⇩ Memory Usage ⇩ Accuracy ⇩

A **concave** function QoE to quantify system performance

$$QoE(r) = \alpha \cdot T(r) + \beta \cdot M(r) - I(r)$$

$T(r)$ Inference time reduction
$M(r)$ Memory usage reduction
$I(r)$ Accuracy loss
α β Weight for time, and memory

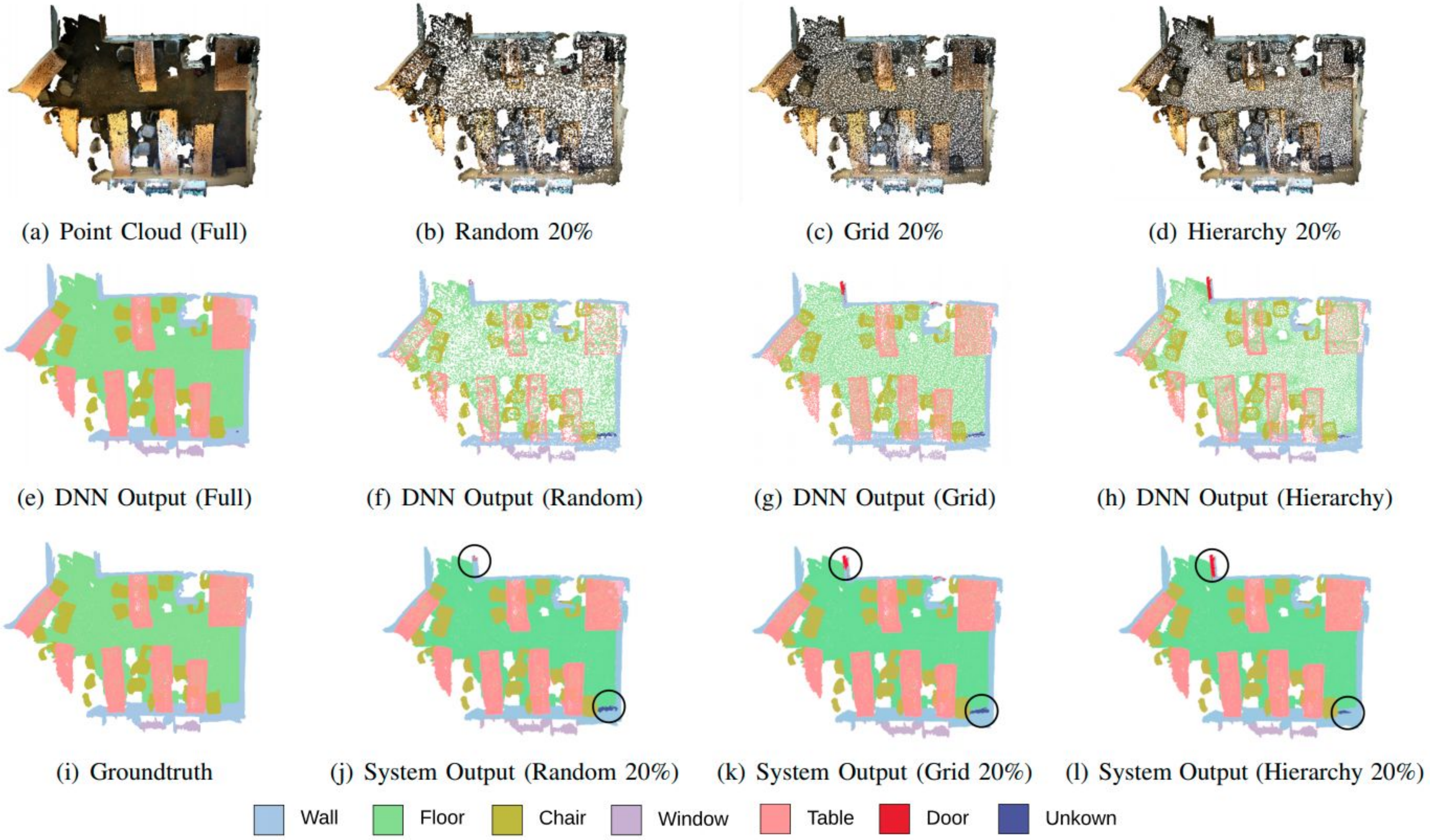
$$T(r) = 1 - \frac{T_S(r) + T_D(r) + T_R(r)}{T_D(100))}$$

Simplified to $QoE(r) = \lambda \cdot T(r) - I(r)$

$$I(r) = 1 - \frac{I_O(r)}{I_D(100)}$$

λ Weight for time, and memory

Visualization of Slimmer Outputs



(a) Point Cloud (Full) (b) Random 20% (c) Grid 20% (d) Hierarchy 20%

(e) DNN Output (Full) (f) DNN Output (Random) (g) DNN Output (Grid) (h) DNN Output (Hierarchy)

(i) Groundtruth (j) System Output (Random 20%) (k) System Output (Grid 20%) (l) System Output (Hierarchy 20%)

Evaluation

Experiment Setup

- Dell Alienware laptop (6-core 2.9 GHz i9 CPUs and 16 GB RAM)
- ScanNet indoor semantic segmentation dataset
- SparseConvNet DNN model of semantic segmentation

Evaluation Steps

1. Performance of the KNN
2. Performance of the simplification methods
3. QoE to explore the design space
4. Overall system performance

Evaluation: Segmenting Removed Points using KNN

Parameter: the number of neighbors K.

Result: we adopt **K = 1** considering the accuracy and processing delay

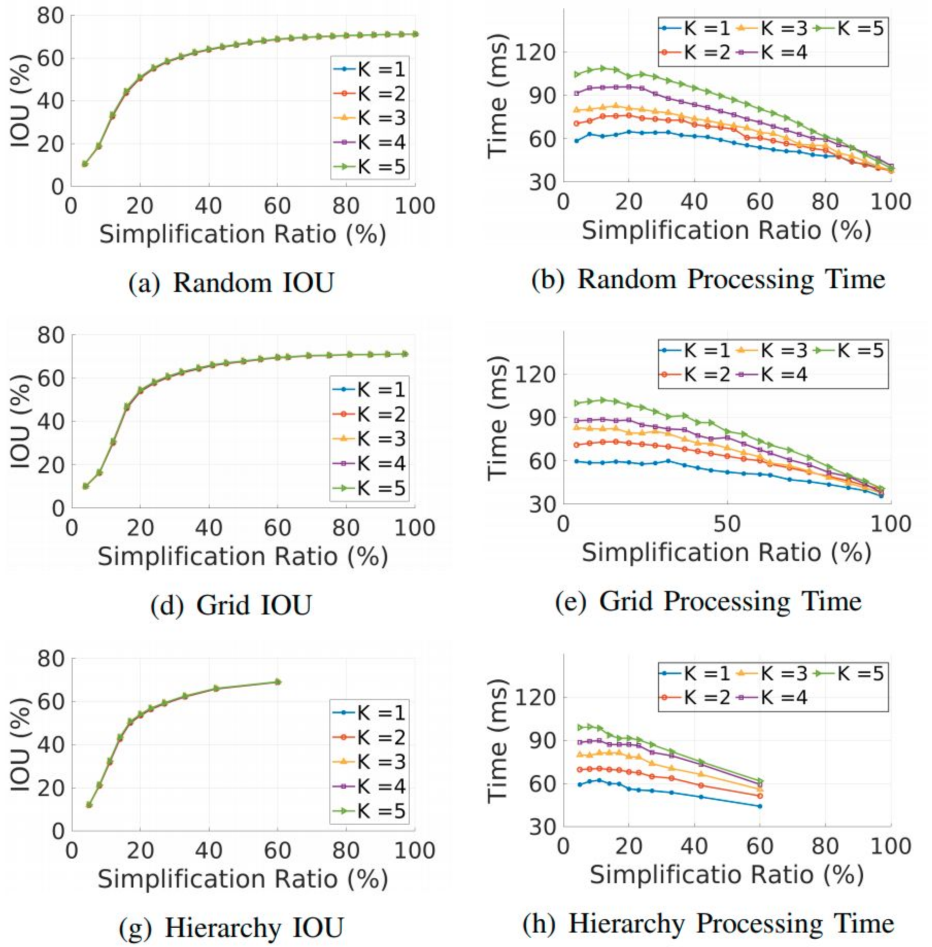
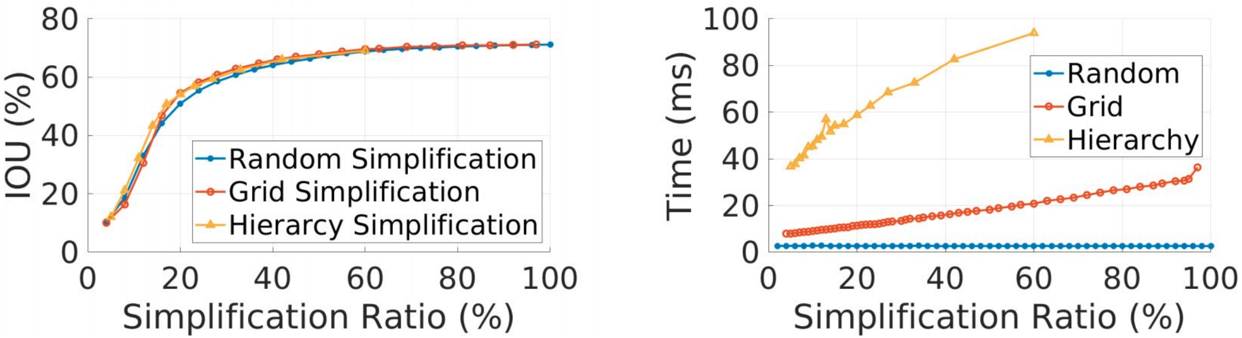


Figure: Study of different number K on performance of the random, the grid, and the hierarchy versus simplification ratio

Evaluation: Simplifying Point Clouds using the Random, the Grid, and the Hierarchy

Result: different simplification methods have advantages and disadvantage in terms of system segmentation accuracy and processing delay.



(a) System Segmentation Accuracy (b) Processing Delay

Figure: Study of the random, the grid, and the hierarchy simplification versus the simplification ratio.

Applying QoE to Compare Different Combinations of the Simplification Method and Ratio

λ : weight for inference time improvement

Result:

- The QoE curves are concave.
- Different simplification methods have different QoE curves for the same λ
- Optimal simplification ratio is smaller for larger weight λ

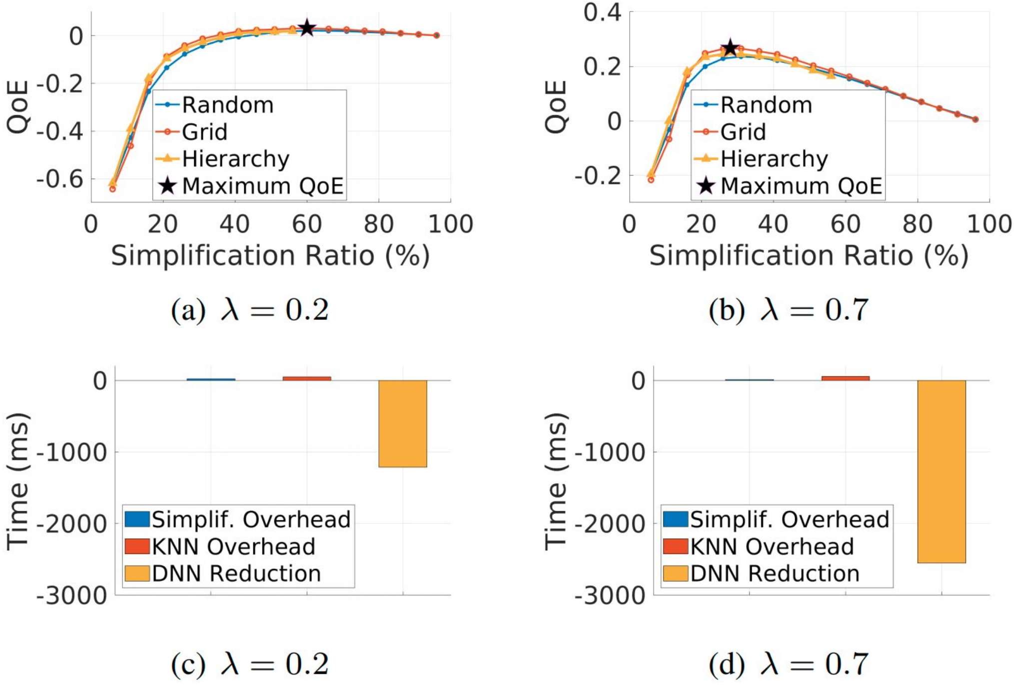


Figure: Leveraging QoE to investigate various design factors

Overall System Performance

Weight λ	0.00	0.05	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
Random QoE	0.000	0.000	0.002	0.022	0.051	0.087	0.133	0.183	0.236	0.294	0.353	0.414
IOU (%)	71.18	71.18	70.48	68.83	67.40	65.32	62.70	62.70	60.78	58.54	58.54	55.41
Accuracy Loss (%)	0.00	0.00	0.98	3.30	5.31	8.23	11.91	11.91	14.61	17.76	17.76	22.16
Time (s)	4.21	4.21	3.72	3.05	2.75	2.43	2.09	2.09	1.91	1.73	1.73	1.54
Time Improvement (%)	0.00	0.00	11.69	27.47	34.61	42.21	50.37	50.37	54.60	58.99	58.99	63.52
Memory (GB)	2.83	2.83	2.59	2.25	2.08	1.89	1.69	1.69	1.58	1.47	1.47	1.35
Memory Improvement (%)	0.00	0.00	8.48	20.73	26.69	33.26	40.42	40.42	44.23	48.20	48.20	52.31
Simplification Ratio (%)	100	100	80	60	52	44	36	36	32	28	28	24
Grid QoE	**0.001**	**0.002**	**0.009**	**0.032**	**0.065**	**0.109**	**0.156**	**0.210**	**0.267**	**0.326**	**0.389**	**0.453**
IOU (%)	71.22	70.93	70.42	69.62	67.03	66.13	62.90	62.90	60.85	60.85	58.24	58.24
Accuracy Loss (%)	-0.06	0.35	1.07	2.19	5.83	7.09	11.63	11.63	14.51	14.51	18.18	18.18
Time (s)	4.20	3.77	3.38	3.07	2.48	2.31	1.92	1.92	1.73	1.73	1.54	1.54
Time Improvement (%)	0.15	10.49	19.58	27.09	41.05	45.03	54.42	54.42	58.88	58.88	63.44	63.44
Memory (GB)	2.90	2.75	2.57	2.40	2.05	1.94	1.67	1.67	1.54	1.54	1.41	1.41
Memory Improvement (%)	-2.31	2.94	9.19	15.19	27.68	31.53	41.01	41.01	45.59	45.59	50.38	50.38
Simplification Ratio (%)	97	81	69	60	45	41	32	32	28	28	24	24
Hierarchy QoE	-0.030	-0.017	-0.004	0.022	0.056	0.098	0.141	0.192	0.246	0.307	0.370	0.434
IOU (%)	69.08	69.08	69.08	69.08	66.07	66.07	66.07	62.59	59.43	56.91	56.91	56.91
Accuracy Loss (%)	2.95	2.95	2.95	2.95	7.18	7.18	7.18	12.07	16.51	20.05	20.05	20.05
Time (s)	3.13	3.13	3.13	3.13	2.42	2.42	2.42	2.02	1.74	1.54	1.54	1.54
Time Improvement (%)	25.54	25.54	25.54	25.54	42.56	42.56	42.56	52.06	58.78	63.42	63.42	63.42
Memory (GB)	2.25	2.25	2.25	2.25	1.84	1.84	1.84	1.61	1.44	1.32	1.32	1.32
Memory Improvement (%)	20.73	20.73	20.73	20.73	34.99	34.99	34.99	43.27	49.21	53.36	53.36	53.36
Simplification Ratio (%)	60	60	60	60	42	42	42	33	27	23	23	23

Table: Details of the system performance of the random, the grid, and the hierarchy simplification versus the weight λ

Conclusion

1. Slimmer is a generic and model-independent framework to accelerate 3D semantic segmentation for mobile augmented reality
2. It can significantly reduce the inference time and memory usage, while remaining high accuracy for state-of-the-art DNN models of semantic segmentation
3. It does not require any modifications to pre-trained DNN models.
4. We propose a QoE metric to quantitatively compare design factors such as simplification method, and the simplification ratio.
5. It provides various tradeoffs between the inference time improvement, the memory usage improvement, and the accuracy loss, by adjusting the weight λ

Thanks

Questions and Answers