Towards Edge-assisted Real-time 3D Segmentation of Large Scale LIDAR Point Clouds

Fraser McLean

The University of Edinburgh

Light detection and ranging (LIDAR) has become a cost-effective and accessible sensor for a broad range of resource-constrained devices, including mobile phones and drones. The technology is capable of providing unrivalled performance in several vision applications, such as Semantic and Instance Segmentation, where high accuracy is imperative to function correctly. Low inference time is crucial to many of these applications where real-time constraints must be adhered to.

Many of the best performing techniques for these vision tasks make use of intensive, state-of-the-art, deep neural networks which are able to provide high accuracy at the cost of inference time. When used out of the box, battery powered resource-constrained devices are ill-equipped to meet the processing demands for such intensive models whilst achieving real-time (ranging from 10Hz to 30Hz depending on the application). When paired with the scale of LIDAR data generation ($\sim$1.10GB/min for Velodyne HDL-64E LIDAR vs $\sim$375MB/min for 4K 30FPS video), completing such real-time processing solely on device is a challenging and impractical task.

We consider edge offloading as a potential approach to reconcile the conflicting requirements of inference accuracy and inference time. Edge offloading allows the high processing power of the cloud to be paired with the locality of on-device processing in order to allow high accuracy processing on resource-constrained devices in real-time. Specifically, we present an experimental characterization study exploring the benefit of edge-assisted LIDAR point cloud segmentation, considering a diverse set of embedded devices, state-of-the-art semantic segmentation models, and edge offloading techniques.

A range of NVIDIA embedded GPUs are used as part of the experimental setup (NVIDIA AGX Xavier, Jetson TX2 and Jetson Nano) to represent the spectrum of resource-constrained devices, and a series of SemanticKitti state-of-the-art models are also considered (including Cylinder3D and 3D-MiniNet). In order to evaluate the effectiveness of edge offloading, we construct an experimental setup through a direct wired Gigabit Ethernet connection between each of the resource-constrained devices and a simulated edge device. The network connection can be varied to simulate real network conditions ranging from bandwidths of 25Mbps to 250Mbps with 10ms or 30ms latencies.

We began by evaluating simplistic edge offloading techniques including direct offloading and compressed offloading (using Octree Compression). We found that significant inference time improvements could be made with compressed offloading when compared to on-device processing, however these alone were not substantial enough to meet even the least strict real-time requirements on any of the devices in any network conditions. Compressed offloaded was able to make use of an ensemble of the highest performing models on the edge to combine their predictions, and improve the overall accuracy performance achievable; taking advantage of the edge's high processing capabilities. Further to this, model partitioned offloading was introduced, and when paired with zstd compression was found to be a fairly effective offloading technique which was able to meet real-time requirements in the best network conditions, with no loss in accuracy. No ensemble was possible with partitioned offloaded due to model partitioning.

In an attempt to emulate a DASH-like edge offloading scenario whereby quality (or inference accuracy) is sacrificed in order to achieve real-time requirements, a small (and less accurate) on-device model was also introduced (using a modified version of 3D-MiniNet-Tiny). A confidence metric was created for it, such that it could be determined if the small model was confident in its prediction or not. The small model was then used in conjunction with both compressed offloading and partitioned offloading to allow real-time requirements to be made by processing the frames which the small model was most confident about on-device, and allowing the uncertain frames to be offloaded. This occurred at the cost of accuracy, which varied depending on the percentage of frames which had to be processed on-device in order to achieve real-time requirements. This allowed real-time processing to be achieved in all network conditions, with the amortised accuracy maximised where possible. Accuracy was found to range from 67.5% in the best network conditions to 51.7% in the worst, with an initial 66.1% accuracy achieved on Cylinder3D alone. Due to introduction of the ensemble, both compressed offloading and partitioned offloading are useful, as although partitioned is generally faster to offload, due to the ensemble the compressed generally gives greater accuracy for a given offloaded percentage - a trade off had to be made between the two approaches.