Semantics-Aware 3D Segmentation and Modeling System for Immersive Simulations and Training Scenarios

Anil Usumezbas, Bogdan Matei, Supun Samarasekera, Rakesh Kumar

SRI International

Princeton, NJ

anil.usumezbas@sri.com, bogdan.matei@sri.com, supun.samarasekera@sri.com, rakesh.kumar@sri.com

ABSTRACT

In recent years, 3D sensors have become increasingly ubiquitous, along with algorithms for integrating the measurements of these sensors over time to produce detailed and high-fidelity 3D models of both indoor and outdoor scenes. As large-scale 3D models become easier and cheaper to produce they still remain prohibitively large and cumbersome to manipulate, thus the emphasis has been slowly shifting from model production to effective storage, transfer, visualization and processing of these models, as well as facilitating their usability and usefulness when a human agent is interacting with them. To this end, we propose a novel and fully-automated system for understanding the distinct components of a large-scale 3D scene and the contextual interactions between such components in order to get a better understanding of the scene contents and to segment the scene into various semantic categories of interest. Imbuing existing 3D models with such semantic attributes is a critical first step in the broader 3D scene understanding problem, allowing automatic identification of different objects, parts of objects or types of terrain, which in turn allows for these categories to be targeted separately by simulation frameworks, as well as various downstream processes. We show that through the use of these semantic attributes, it is possible to: i) generate significantly more compact models without drastic degradations in quality and fidelity, allowing the deployment on mobile platforms with limited computational capabilities, ii) improve localization accuracy when estimating the full six degrees of freedom (6-DOF) pose of a mobile agent situated in the scene, and iii) provide agents with richer and smoother interactions with such 3D models during simulations and training scenarios.

ABOUT THE AUTHORS

Dr. Anil Usumezbas is currently an Advanced Computer Scientist in Vision and Robotics Laboratory from the Center from Vision at SRI International, Princeton, NJ. He received his Ph.D. in Electrical Sciences and Computer Engineering from Brown University in 2016. His research interests focus on 3D vision, specifically photogrammetry techniques, localization, odometry and mapping approaches, multiview geometry, semantic segmentation and object detection in 3D scenes as well as deep learning as it pertains to 3D data modalities. He has peer-reviewed publications in top conferences in his field.

Dr. Bogdan Matei is currently Technical Director in the Vision and Robotics Laboratory from the Center from Computer Vision at SRI International, Princeton, NJ. He received his Ph.D. in Electrical and Computer Engineering from Rutgers University in 2001. His technical interests are in the areas of 3D scene understanding, Fine-grained recognition from 2D/3D imagery, Image indexing and search, 2D / 3D object recognition/classification using deep learning, image geolocation in the wild, 3D modeling, 3D/2D object detection and recognition. Dr. Matei received the Best Student Paper award at Computer Vision and Pattern Recognition conference in 1999. Dr. Matei has published more than 25 peer-reviewed conferences and journals and he has ten granted patents. He is a Senior Member of the IEEE.

Mr. Supun Samarasekera is currently the SR. Technical Director of the Vision and Robotics Org. within the Center of Vision Technologies at SRI International. He received his M.S. degree from University of Pennsylvania. Prior to joining SRI, he was employed at Siemens Corporation. Mr. Samarasekera has 15+ years of experience in building integrated multi-sensor systems for training, security, and other applications. He has led programs for robotics, 3D

modeling, training, visualization, aerial video surveillance, multi-sensor tracking, and medical image processing applications. Mr. Samarasekera has received a number of technical achievement awards for his technical work at SRI.

Dr. Rakesh "Teddy" Kumar is currently Vice President of Information and Computing Science division and Director of the Center for Vision Technology at SRI International, Princeton, NJ. Prior to joining SRI, he was employed at IBM. He received his Ph.D. in Computer Science from the University of Massachusetts at Amherst in 1992. His technical interests are in the areas of computer vision, computer graphics, image processing, and multimedia. Dr. Kumar received the Sarnoff Presidents Award in 2009 and Sarnoff Technical Achievement awards in 1994 and 1996 for his work in registration of multi-sensor, multi-dimensional medical images and alignment of video to three-dimensional scene models respectively. He was an Associate Editor for the Institute of Electrical and Electronics Engineers (IEEE) Transactions on Pattern Analysis and Machine Intelligence from 1999 to 2003. Dr. Kumar has co-authored more than 50 research publications and has received over 35 patents.

Semantics-Aware 3D Segmentation and Modeling System for Immersive Simulations and Training Scenarios

Anil Usumezbas, Bogdan Matei, Supun Samarasekera, Rakesh Kumar

SRI International

Princeton, NJ

anil.usumezbas@sri.com, bogdan.matei@sri.com, supun.samarasekera@sri.com, rakesh.kumar@sri.com

INTRODUCTION

In the last decade, simultaneous breakthrough advances in both 3D sensing hardware technology and large-scale 3D model inference and/or integration algorithms, supported by visual odometry, simultaneous localization and mapping (SLAM), 3D point cloud alignment and related capabilities, have made large-scale and detailed models of indoor and outdoor scenes increasingly ubiquitous. More recently, with the practical advances in deep neural networks, the research focus in 3D computer vision has shifted from accurate capture and production of high-fidelity 3D models to meaningful processing of such models by down-stream processes such as dense modeling/meshing (Vanegas et. al., 2010), (Birdal and Ilic, 2017), (Wu et. al., 2015), (Ulusoy et. al., 2017); object/target detection (Zhou and Tuzel, 2018), semantic segmentation (Landrieu and Simonovsky, 2018), (Qi and Yi and Su and Guibas, 2017), model alignment/registration (Avidar et. al., 2017), (Lee et. al., 2017), model/mesh simplification (Zou et. al., 2017) among others. The broader goal of this collective research effort is to bring state-of-the-art capabilities for algorithmic reasoning and scene understanding closer to actual human cognition of 3D scenes, with practical approaches for imbuing these large-scale models with useful attributions that assist the aforementioned downstream processes. The first step towards this goal is solving the problem of *3D semantic segmentation*, namely the problem of delineating different categories of areas, objects and parts in a given scene to infer a basic attribute layer is semantically meaningful to a human agent. See Figure 1 for a result of the proposed approach. The 3D semantic models can be used for a broad set of applications ranging from mission planning and rehearsal using geo-specific simulation models, training using augmented or virtual reality systems, and GPS-denied navigation for tactical situations.

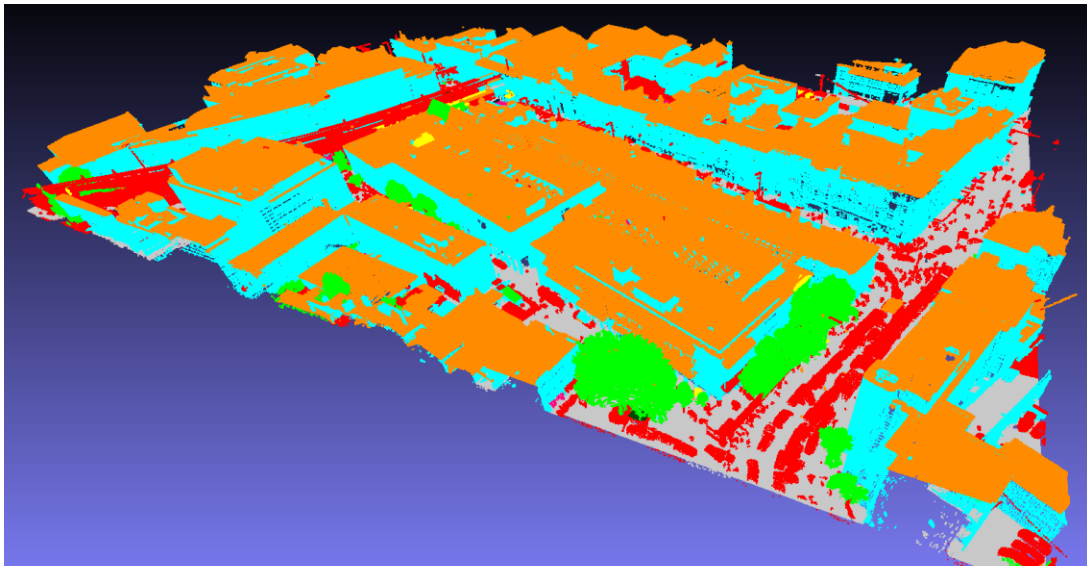

Figure 1. A scene from the public DublinCity benchmark, colorized by different semantic categories as inferred by the approach presented in this paper. Teal: facade, Orange: roof, Dark Green: grass, Light Green: tree, Yellow: bush, Gray: street, Pink: sidewalk, Red: clutter.

In this paper, we present a robust, state-of-the-art and end-to-end automated 3D segmentation system that targets large-scale outdoor scenes and effectively combines a number of previously unrelated approaches and techniques.

Specifically, the pipeline integrates: i) rule-based sequential algorithms for 3D point cloud data, ii) image-based deep neural networks, iii) 3D deep neural networks, both convolutional and graph-based, and iv) efficient raycasting to merge inferences in 2D and 3D. Once integrated, these components address the shortcomings of each other to produce state-of-the-art semantic segmentation results on the most recent and most comprehensive outdoor 3D semantics benchmark, DublinCity Dataset (Zolanvari et. al., 2019), (Laefer et. al., 2017).

The problem of 3D semantic segmentation on large-scale scenes contains a few fundamental challenges:

i) *Scene complexity*: Modeling entire scenes instead of individual, cleanly-separated objects is significantly more challenging due to high variability in scene content and composition, complex surface geometries, self-occlusions and the need for higher spatial resolution due to size disparities between different types of objects.

ii) *Training data acquisition and labeling*: Annotating 3D data is non-trivial and costly, and very few high-quality, large-scale benchmarks exist for outdoor scenes that contain labeled ground truth 3D models for training and evaluation.

iii) *Computational bottlenecks*: Increased dimensionality of 3D data puts additional strain on computational demands of deep neural networks and obtaining sufficiently high-resolution 3D results requires innovative designs.

Fully automated semantic segmentation approaches, both in 2D and 3D, have been studied extensively in computer vision and can be considered in two broad categories. The first, is a rule-based algorithm that encapsulates a set of declarative statements about the nature of scenes and semantic categories that are being targeted. These statements can range from being very simple and straightforward, e.g., streets should be at a locally minimum altitude, to being more complex and relational, e.g., building roofs should mostly be flat, horizontal and supported by vertical walls. Sequential algorithms that encapsulate a number of such constraints are very predictable in behavior and have high expressive power, however they are often brittle in the face of small, unexpected differences and typically do not generalize well to novel inputs. Furthermore, it becomes exceedingly difficult to integrate a high number of such constraints effectively, since each algorithmic piece interacts with all the rest, creating numerous edge cases and hard-to-predict errors that explicitly need to be considered and addressed.

The second category is a machine learning approach, consisting of either hand-crafted models, or more recently, deep neural network architectures that are very effective at parsing the implicit knowledge contained within annotated data, and is akin to human cognition observing many known exemplars to develop the means to recognize, understand and navigate previously-unknown scenes. Deep neural networks are excellent at capturing such information in a way that generalizes well to different types of scenes and scene compositions, but they require large amounts of training data to be effective, and do hardly any explicit reasoning, rendering the underlying mechanism for inferred results somewhat indecipherable. Due to lack of any explicit reasoning capability, these networks often make trivial errors that would be easy to fix in a sequential algorithm setting, but such fine-grained, local interventions are difficult to design and implement in the context of deep learning.

The main contribution of this paper is to present a novel, hybrid pipeline that combines the strengths of both types of approaches – the machine learning components extract knowledge that is implicit in annotations, while the rule-based components provide added reliability, generalizability and predictability in behavior. With this system, we are targeting various simulation and training applications where users inspect or align 3D models or even interact with them with game engines and/or VR-based simulations. Being able to automatically infer the semantic content of 3D scenes provide significant capabilities to such simulation and training systems. We demonstrate this capability with two sets of experiments: i) We show that the availability of semantic attributes allows for a variable-resolution mesh reconstruction and compression algorithm that can target different types of areas or objects operating at different resolutions for mission rehearsal and planning on devices with different processing capabilities, and ii) we show that geo-registration of images to a 3D reference model for augmented reality, GPS denied navigation and other applications can be done with improved accuracy if semantics are present. We evaluate the accuracy of the semantic inference results against a hand-annotated ground truth by using a portion of the DublinCity dataset as test samples.

RELATED WORK

Practical neural network architectures for 3D semantic segmentation can be considered in three broad categories. **Voxel-Based Approaches** are techniques that are the most intuitive extension to the 3D domain from the 2D image domain-based approaches and were the earliest to be developed. The idea is to quantize the 3D space into voxels, octrees or other regular or irregular grid-based structures, and then replicate the functionality of 2D pixel-based architectures in three dimensions. UNet-based architectures (Çiçek et. al., 2016) use a U-shaped encoder-decoder pair where the encoder maps the input voxel structure to a lower-dimensional vector, then the decoder maps the vector back to the same grid structure, with a semantic label attached at each grid location. The encoder and decoder portions jointly learn how to effectively represent an entire quantized scene or object with a latent vector that contains useful semantic information. VoxelNet (Zhou and Tuzel, 2018) also utilizes a regular grid and computes unified features for each voxel location using the point configurations inside each voxel. Octree Generating Networks (Tatarchenko et. al., 2017) learn to infer efficient octree structures from dense, regular voxel grids to selectively increase the level-of-detail and model quality in places where fine-grained details matter more. These methods have proved effective in segmenting individual objects or relatively small scenes, e.g. a single room, but sometimes suffer from quantization-related errors and are generally unable to effectively deal with larger scenes or very detailed objects before running into computational limits or having to divide the input into smaller chunks first. Regular grid structures are also not equipped to deal effectively with sparse data.

Point Cloud-Based Approaches directly ingest the unorganized 3D point cloud structure and fundamentally obey the permutation invariance of input points, while inferring a semantic label for each of them. PointNet (Qi and Su and Mo and Guibas, 2017) is a seminal early paper which presents a fully-convolutional neural network which computes local features around each point using multi-layer perceptrons, and a global feature for the whole input, which is then appended to each local feature. In this way, PointNet combines a global cue with local ones, resembling the idea of shape context, or other constellation-type models that have been in widespread use for years to characterize shape. PointNet architecture is not fully invariant to spatial transformations, but it uses a simpler version of itself as a Spatial Transformer Network to transform both input points and the computed features closer to a canonical “pose” to provide some robustness under small-scale transformations. Finally, the use of a global feature as well as a fully convolutional neural network requires PointNet to represent each input with a fixed number of points, limiting input size and level-of-detail significantly. PointNet++ (Qi and Yi and Su and Guibas, 2017) is an iteration on this idea that hierarchically encodes a point cloud by using the embedding vectors from one level as points themselves to encode the next level of embeddings. Thus, PointNet++ uses semantic embeddings to implicitly break the scene into smaller chunks, allowing it to handle much larger scenes or objects. SplatNet (Su et. al., 2018) is a more recent approach that interpolates data onto a permutohedral lattice, filters the data on this sparse lattice, then interpolate the filtered signal back to original points. The lattice interpolation provides a convenient way to jointly consider 3D point clouds with registered imagery when available. We use the intuitions provided in these approaches on how best to ingest point cloud data structure, as well as architectural details from PointNet to represent patches of points with appropriate embedding vectors, but we explicitly break the Euclidean scene into smaller fragments rather than implicitly do it at the embedding spaces like PointNet++.

Graph-Based Approaches use a graph-based representation of the scene with graph convolutions to infer labels, where nodes typically correspond to either individual points, local neighborhoods of points or patches of points and edges indicate either proximity or adjacency in some pre-defined neighborhood structure. Super-point Graph (Landrieu and Simonovsky, 2018) is a popular 3D segmentation approach that uses bottom-up geometric features computed from the covariance matrix of a local neighborhood of points to solve a traditional optimization problem and compute superpoints – patches of points that are maximally homogeneous in terms of these bottom-up features. A Voronoi partitioning and an accompanying connectivity analysis is used to transform the scene into a graph, and semantic labels are inferred for each node using both its own vector embedding as well those belonging to the neighbors of that node, in order to utilize surrounding context. We make extensive use of these ideas to formulate 3D segmentation problem as a mixture of graph-based inference and constraints based on first-order logic.

SYSTEM PIPELINE

The design of the system pipeline is predicated on the idea that reliably segmenting the scene into coarse categories enables further fine-grained segmentation of smaller entities and object parts. See Figure 2 for a complete diagram of the system pipeline, each module of which is described in greater detail in this section, in the order of data flow.

Pre-Processing: The goal of the pre-processing step is to increase the system’s robustness capability to operate in the presence of outlier points that are either low-confidence LIDAR measurements or artefacts from the photogrammetry algorithm. The presence of significant outlier points adversely affects the accuracy of ground level estimation, as well as distorting the geometry and context surrounding objects or areas of interest, making it harder for the system to infer the correct semantic labels. The difficulty in coming up with a fully-automated approach to remove such outliers is twofold: i) These outliers are not individual, isolated points, but rather are densely-populated patches of points that locally resemble a legitimate surface, and ii) it is commonplace for certain objects above ground level to be disconnected from the rest of the scene in a similar way, therefore only the outliers below ground level should be targeted and removed.

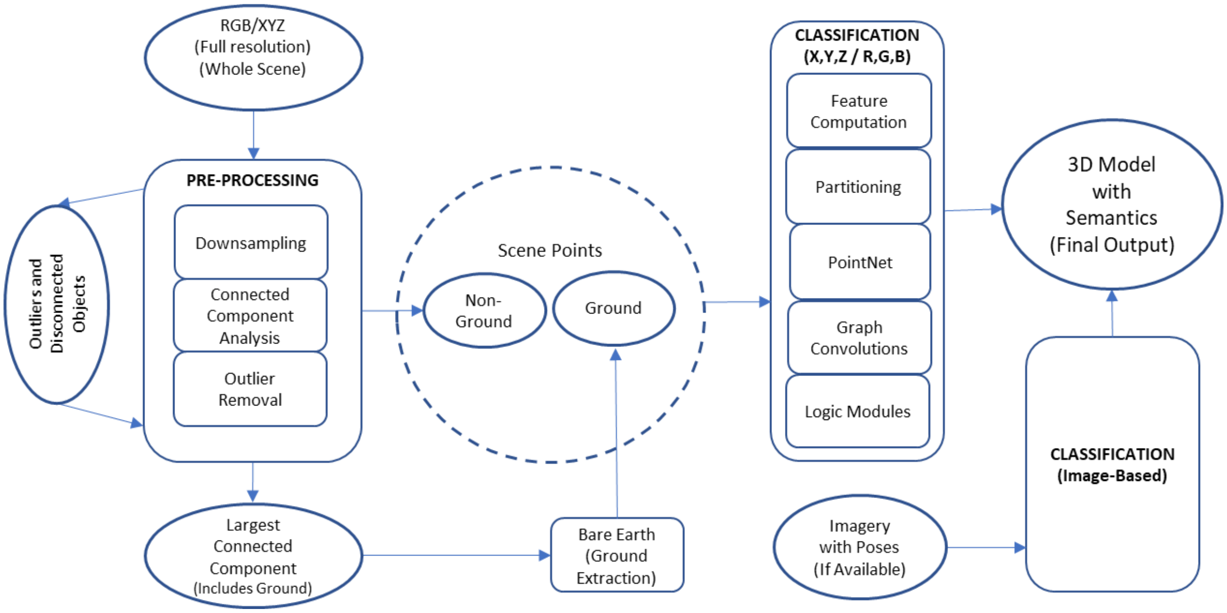
RGB/XYZ
(Full resolution)
(Whole Scene)
PRE-PROCESSING
Downsampling
Connected Component Analysis
Outlier Removal
Outliers and Disconnected Objects
Largest Connected Component (Includes Ground)
Scene Points
Non-Ground
Ground
Bare Earth (Ground Extraction)
CLASSIFICATION (X,Y,Z / R,G,B)
Feature Computation
Partitioning
PointNet
Graph Convolutions
Logic Modules
3D Model with Semantics (Final Output)
CLASSIFICATION (Image-Based)
Imagery with Poses (If Available)

Figure 2. An overview of the processing pipeline main blocks. Circles/ellipses indicate data and rectangles indicate processing modules.

The pre-processing module works as follows: The input point cloud is down-sampled to a spatially uniform resolution by computing a voxel grid, which is then used to do 26-connected component analysis on the raw point cloud. Resulting connected components are ordered from largest to smallest according the number of points contained, and largest n connected components are fused into “scene points” where n is selected such that scene points cover a sufficiently large portion of the raw input. The goal here is not to capture the entirety of non-outlier points, but rather to obtain a sufficiently large yet incomplete subset of the input scene that is devoid of any outlier points. The ground estimation algorithm can then be run on these scene points without being derailed by the outliers. Once a reliable estimate for the ground level is obtained, all the disconnected components above ground are added back, while the below-ground patches are discarded.

Ground Extraction: The ground extraction approach is, at its core, a rule-based binary ground vs. non ground classifier algorithm with robust interpolation capabilities that produces binary labels as well as a DEM representation of the terrain and follows an earlier work (Matei et al., 2008). It grows patches of 3D points from a number of seed points that have locally minimum altitudes; the region-growing itself takes into account smoothness of local point neighborhoods. The disjoint patches are eventually merged and filled to create the ground layer. Aside from allowing the below-ground outliers to be targeted and discarded, the extracted ground serves as a valuable frame of reference for all other objects present in the scene, establishing a useful coordinate system and allowing us to utilize height-above-ground elevation values and provide excellent performance in rapidly-changing terrain conditions such as hills, mountains and vineyards.

Superpoints: Scene Partitioning Using 3D Voronoi Diagrams: The idea of superpoints (Landrieu and Simonovsky, 2018) is analogous to superpixels (Achanta et. al., 2012): they are local groupings of 3D points that are homogenous in terms of a set of desired properties. In the use case of this pipeline, the desired properties are a set of local geometric features expressed in terms of eigenvalues of point distributions, as well as appearance features like color and texture. By formulating the point classification problem at the level of superpoints rather than individual points, the computational complexity and the size of the required training data is reduced drastically without impacting the performance, assuming that the superpoint partitioning is such that all the points in a given superpoint belong to one semantic category only. Once the features of interest are computed for each point, the superpoint partitioning is done using cut-pursuit optimizer (Landrieu and Obozinski, 2018), (Raguet and Landrieu, 2018). A regularization strength parameter determines how aggressive the grouping will be, and it is empirically selected for each sensor type. Using these superpoint partitions, a Delaunay triangulation of the complete 3D input point cloud is computed, resulting in a Voronoi diagram that dictates a graph topology on the superpoints where graph edges indicate a shared boundary between at least a single pair of points across two different superpoints. These edges are further weighted by "edge features" which are computed from the complete set of 3-dimensional offsets between neighboring points across two different superpoints. This graph becomes the intermediate representation for the input scene, where each node is a superpoint, see Figure 3 for an illustration. Defining neighborhoods this way instead of by proximity allows for very long-range interactions between different regions if the space between them is empty. This is a very compact and very powerful representation for effectively taking into account context during segmentation.

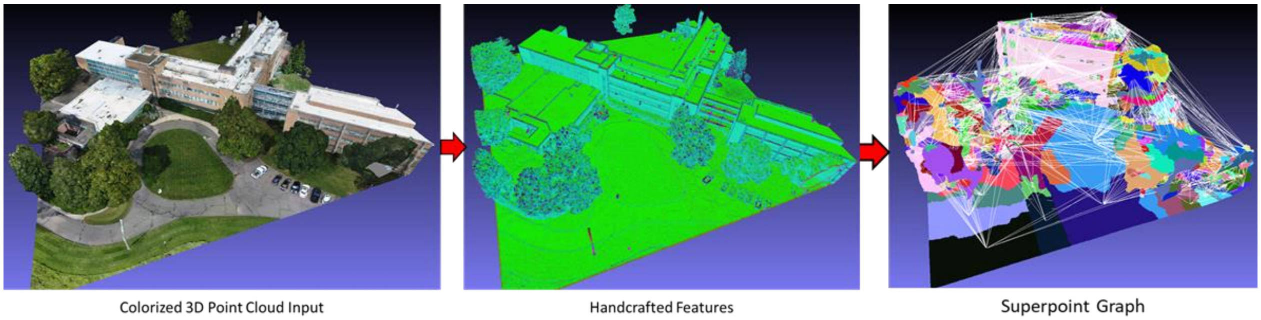

Figure 3. Input point cloud with RGB (left), color-coded local geometric features (center) and resulting superpoints and the neighborhood graph (right).

PointNet and Graph Convolutions: Graphs are among the most general data types to effectively represent entities and their relationships, therefore they lend themselves naturally to the problem of modeling context. In addition, they allow a functional combination relational structures known or computed a priori, with the end-to-end learning power of neural networks. In the context of deep learning, graph representations fully generalize the specific connectivity structures of other standard networks such as convolutional networks (CNNs) which are fully-connected special-case graphs, and recurrent neural networks (RNNs) which have a chain graph structure.

In the graph neural network paradigm, the standard neural network operations such as convolutions and pooling are replaced by a 4-step operation over the internal states of each graph node, as well as the graph itself:

i) Message passing: Each edge in the graph carries a message from a sender node to a receiving node. If the graph is not directional, such as the graph-based representations of outdoor scenes that are used in the pipeline for this paper, the messages are passed over each edge bidirectionally.
ii) Edge update: Edge features are updated with respect to each receiver node, according to the message carried.
iii) Node update: Updated edge features for each receiver node are aggregated, updating the internal state of the node
iv) Global update: Global attributes of the graph are updated, if any.

For graph edge weights and node update equations, we closely follow the formulation outlined in (Landrieu and Simonovsky, 2018). The edge weights are features that characterize the geometric properties of each node, i.e. superpoints, relative to its Voronoi neighbors. Refer to (Landrieu and Simonovsky, 2018) for more details. The latent vector that stores the internal state of each node prior to any message passing is generated by ingesting each superpoint patch into a PointNet architecture without the Spatial Transformer Network component. In a way, we use PointNet architecture in lieu of local geometric features, and the graph convolutions with the edge weights in Table 1 bring in global context.

Image-Based Segmentation and Efficient Raycasting: While 3D data is typically richer in information content, image data that is aligned to a 3D reference is not redundant – certain features are simply easier to detect on images, such as building facades and windows, which are mostly planar structures that lack geometric detail and therefore must be detected using color, texture or other forms of appearance. While 3D point clouds can also contain these attributes, the density and connectivity structures of image pixels make them better suited to find these types of objects. The problem is that these image-based segmentations will then need to be mapped onto the 3D model efficiently.

To this end, we use state-of-the-art deep neural networks for image-based detection and segmentation, together with an efficient raycasting approach to transfer the results back into 3D. We experiment with two different structures – building facades and windows. For building facades, we use a modified version of RetinaNet (Lin et. al., 2017), which is a network that is very similar to Region-based Convolutional Neural Networks (R-CNNs) (Girshick et. al., 2014) in that it utilizes a two-stage detection process, first stage computing the anchor locations and aspect ratios while the second stage regresses to find offsets from these anchors to final detections. Instead of using a bounding box representation, we alter this network design so that the second stage regresses boxes into general quadrilaterals instead. The general quad representation allows us to get very tight bounds on building facades, and can be trained on as little as a few hundred annotated images before it can learn to reliably find rectangular building facades.

Building windows are much smaller features, so trying to fit quads accurately around each window does not work as reliably as doing pixel-based segmentation to find them. To this end, we use a state-of-the-art variant of UNet architecture, namely Unified Perceptual Parsing Network (Xiao et. al., 2018). This hierarchical network design uses features at various semantic levels and identifies compositional structures, *i.e.* objects and their parts, among detected concepts. We trained this network on ADE20K (Zhou et. al., 2017), (Zhou et. al., 2019), which is the largest image segmentation dataset and benchmark to the best of our knowledge. Then we tested the network on aerial images of SRI Princeton campus – even though the training data hardly contains any aerial data, the results are very accurate, as shown in Figure 4.

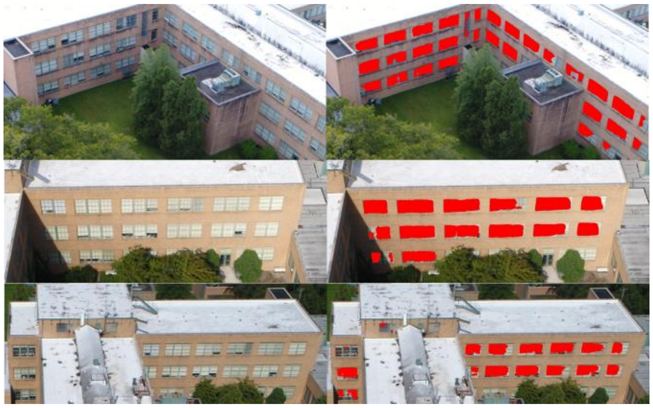

Figure 4. Image-based window segmentations using the presented pipeline. Window pixels are marked in red. Almost all instances of windows are detected accurately, even though the images are taken from an oblique, aerial view.

We transfer these image-based detection and segmentation results back into the 3D model through an efficient ray-casting algorithm that casts a ray from each pixel into the point cloud to attempt to find the 3D points corresponding to the detections or segmentations on the image. Since the points are infinitesimally small, the ray is likely to pass through the point cloud without hitting any actual points. Therefore, we represent the points with spheres of varying radius that these rays could potentially intersect with. It is also true that a given ray can pass through multiple such spheres, so we pick the intersecting point that is closest to the camera, which is a form of straightforward occlusion reasoning. Raycasting is a relatively expensive procedure, therefore we compute an octree structure on the point cloud, and use this octree structure to first find which cells intersect with a set of rays, and then inside each cell, we process a finer-detail intersection to find which exact points are hit inside a given cell. The radius of the sphere that is going to be represent each point need to be chosen in a way that takes into account: i) density of the point cloud, so that neighboring spheres don’t actually intersect with each other, but also don’t have large gaps between them, ii) perspective foreshortening, because the points closer to the camera should be represented with smaller spheres than those that are far, and iii) resolution of the camera image, since that determines how large a frustum each ray represents. See Figure 5 for an example segmentation results where window detections are merged from image-based segmentations using this raycasting approach, with other categories inferred from 3D.

Basic Neural Network Reasoning with First-Order Logic: One of the most fundamental shortcomings with deep neural networks is their lack of basic reasoning capabilities, which sometimes cause them to make trivial mistakes that either violate some common-sense facts or yield results that are hard to explain or inconsistent with respect to some external constraints that are not captured by the supervision. When annotations are abundantly available, this kind of logical reasoning is not as necessary, but for relatively rare and/or small object categories, it is more crucial.

Most existing approaches integrate such logical constraints into their pipeline either by post-processing inference results, or by adding additional feature channels to bring out desired properties manually. A much better way would be to embed such domain knowledge directly inside the training process of all related deep networks, so that they softly constrain the knowledge implicit in the supervision, rather than directly modifying inference results, or trying to capture some of the domain knowledge in the feature layers. This idea necessitates the logical constraints, which are normally thought of as operations on binary values, to be expressed in a continuous domain so as to allow for gradient backpropagation through the logical operators during training. This is the key idea that allows us to tightly integrate any rule or constraint into deep neural networks, as long as they can be expressed using standard first-order logic. Such continuous-domain versions of logical operators are referred to as "*groundings*" for first-order logic, and the choice of a good set of groundings is crucial in avoiding numerical instabilities as well as other common problems like vanishing gradients that become more pronounced as the gradients have to flow through more and more layers. For this system, we use the groundings described in (Sikka et. al., 2020), which maps logical binaries to (−∞,∞) range.

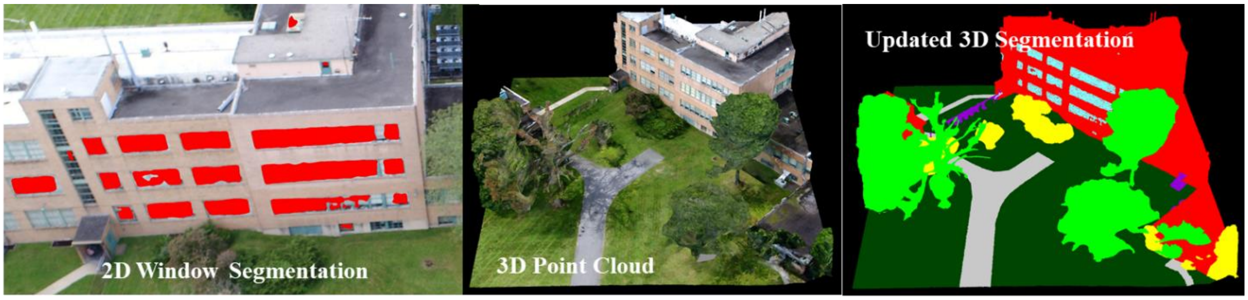

Figure 5. Image-based window segmentation results (left) are cast into the 3D point cloud model (center) to merge window segmentations with the other categories (right).

The first step when integrating a rule set or a specific piece of domain knowledge is to write the rules or constraints as declarative, first-order logic statements, making use of the 7 standard operators: and, or, not, equals, implies, for all, there exists. Figure 6 shows an example where 3 simple rules about the outdoor scenes are written down, and computational equivalents to certain semantic phrases such as "touching" and "too small" are defined. These modules can either be straightforward functions or other neural networks in their own right; regardless, gradients will flow through them so that the logical constraints can be made a part of the training process. When the rules are expressed in terms of these functions and first-order logic operators, the expressions imply a certain set of logical connections between the segmentation network and the rule grammar. This entire structure is connected to the latent vector that is used to infer semantic labels, constraining the form it can take, thereby constraining the latent space during training according the rule constraints. This entire construct allows the system to correct for basic common-sense mistakes, while also making the results of the system more explainable to a human agent.

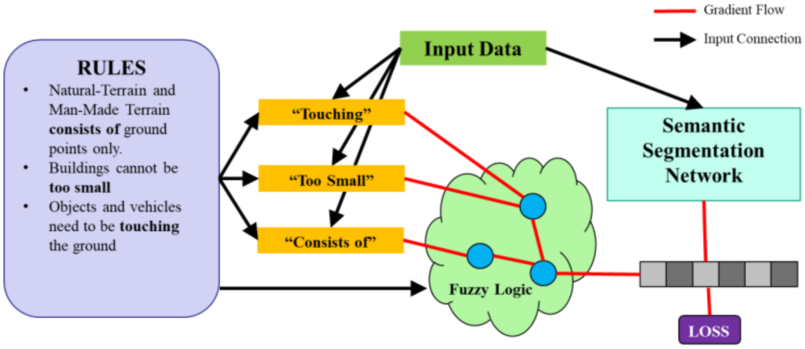
Gradient Flow
Input Connection
Input Data
RULES
• Natural-Terrain and Man-Made Terrain consists of ground points only.
• Buildings cannot be too small
• Objects and vehicles need to be touching the ground
"Touching"
"Too Small"
"Consists of"
Fuzzy Logic
Semantic Segmentation Network
LOSS

Figure 6. A schematic showing some example rules and how they interact with a deep neural network. Orange boxes are function implementations corresponding to the phrases they represent, and blue nodes are continuous-domain logical operators. Together they form the grammar for a given problem. Gradients flow through the red connections.

EXPERIMENTS AND RESULTS

We formally evaluate the system in two distinct ways. First, we quantitatively evaluate the accuracy of inferred semantic labels using annotated benchmarks – we simply train the system on a portion of the benchmark and test on the remaining portion. We present color-coded visualizations of the segmentation results for qualitative inspection, as well as two metrics for quantitative evaluation:

i) *Point classification accuracy*: Simply the percentage of 3D points that are accurately classified. Measures the accuracy of segmentation across all test data and all semantic classes.

ii) *Intersection over union*: This is a method for measuring how well the regions of each semantic category overlap, and is basically a ratio of the size of the intersection region over the union region. Here, size refers to number of points in the region, and intersection/union operations are done between inferred results and ground truth annotations.

Second, we prove the usefulness of these inferred labels in two different downstream processes: mesh reconstruction/compression and pose estimation. Specifically, we show that making these processes "semantics-aware", i.e., able to utilize the semantic attribute layer productively, there is demonstrable improvement in performance.

Qualitative and Quantitative Evaluation on Annotated Benchmarks

We use the largest publicly available outdoor 3D semantic segmentation benchmark, namely DublinCity (Zolanvari et. al., 2019), (Laefer et. al., 2017) for a formal evaluation of the system. This is a dataset captured by Urban Modelling Group at University College Dublin via the use of an ALS device on a helicopter, and covers one of the major areas in Dublin city center, about 2 km^2. To the best of our knowledge, it is the largest, densest and most accurate semantic segmentation benchmark that is publicly available. It also contains many labels organized in a hierarchy across 3 levels – level 1 contains the coarsest categories (building, vegetation, ground and clutter) while level 2 and level 3 contain parts of some of these objects (façade, roof, window, door) or finer-grained subcategories of others (tree, bush, sidewalk, street, grass). For the purposes of this evaluation, the results are reported on the first two levels.

Since there are no other reported results on this relatively recent and challenging benchmark, we compare the results of the presented system to a baseline method that is in widespread use in the 3D vision community: PointNet++ (Qi and Yi and Su and Guibas, 2017), which, as we mentioned earlier, is an extension of the original PointNet architecture that turns it into a hierarchical process so that larger scenes can be processed. See Tables 1 and 2 for a comparison of the system's metrics against this benchmark, and Figure 7 for some visualizations of segmentation results using level 1 and level 2 categories. In most categories and resolutions of data, the presented system is on par or better than the baseline, with the difference becoming more pronounced when data resolution is increased, and smaller-scale categories are targeted.

Table 1. Quantitative evaluation results on level 1 of DublinCity benchmark.

Resolution	Approach	Point Classification Accuracy	IoU Ground	IoU Building	IoU Vegetation	IoU Clutter
100cm	Proposed Pipeline	**83.68%**	0.346	**0.854**	**0.661**	**0.317**
	PointNet++	76%	**0.35**	0.77	0.38	0.00
30cm	Proposed Pipeline	**89%**	**0.427**	**0.91**	**0.6410**	**0.378**
	PointNet++	83%	0.38	0.82	0.00	0.00

Table 2. Quantitative evaluation results on level 2 of DublinCity benchmark.

Resolution	Approach	Point Classification Accuracy	IoU Roof	IoU façade	IoU Tree	IoU Bush	IoU Clutter	IoU Grass	IoU Street	IoU Sidewalk
100cm	Proposed Pipeline	**70.71%**	**0.612**	**0.562**	**0.603**	0.000	0.243	**0.196**	**0.199**	0.074
	PointNet++	64.52%	0.412	0.554	0.58	0.000	**0.35**	0.185	0.141	**0.105**
30cm	Proposed Pipeline	**69.02%**	**0.672**	**0.630**	**0.813**	0.000	**0.559**	0.161	**0.325**	**0.215**
	PointNet++	61.06%	0.455	0.563	0.653	0.000	0.42	**0.201**	0.203	0.17

Semantics-Aware Mesh Reconstruction and Compression: One of the common processes downstream from point cloud acquisition is meshification/dense surface reconstruction, and effective compression of such dense models. The semantic attributes inferred by the system allow different classes of objects to be targeted by modeling approaches in a way that is uniquely tailored to that object class. For instance, using the knowledge that buildings are mostly rectilinear with sharp boundaries, edges and corners, areas segmented with the "building" label can be processed with an algorithm that can safely make those assumptions. On the other hand, ground layer can be effectively represented in 2.5D, using a coarser resolution. This means that different parts of the same scene can be represented at different resolutions and level-of-detail, making optimal use of computational resources as well as transfer bandwidths and storage space. By making the meshing/surface reconstruction process aware of semantics in such a simple way gives the user an ability to produce more compact and lightweight models that can easily be loaded onto mobile platforms, while preserving distinct and salient features for each category when simplified. In contrast, generic mesh simplification algorithms lose these important features, or smooth over important details when attempting effective compression.

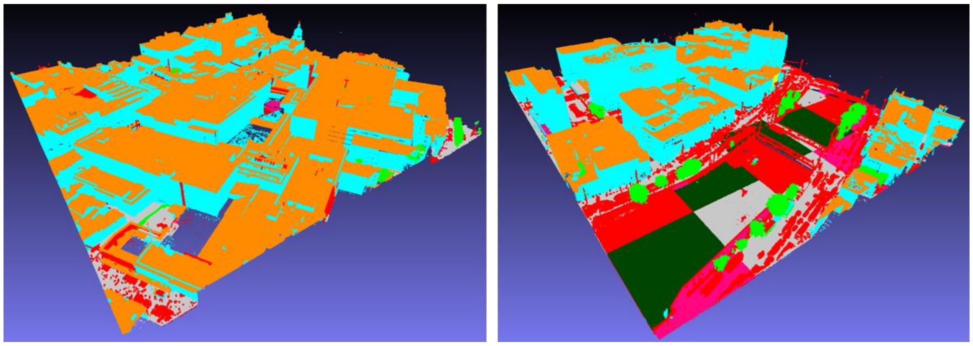

Figure 7. Example segmentation results of the system on 3 test tiles from DublinCity dataset. Orange: roof, Teal: façade, Red: clutter, Grey: street, Pink: sidewalk, Light green: tree, Dark green: grass, Yellow: bush.

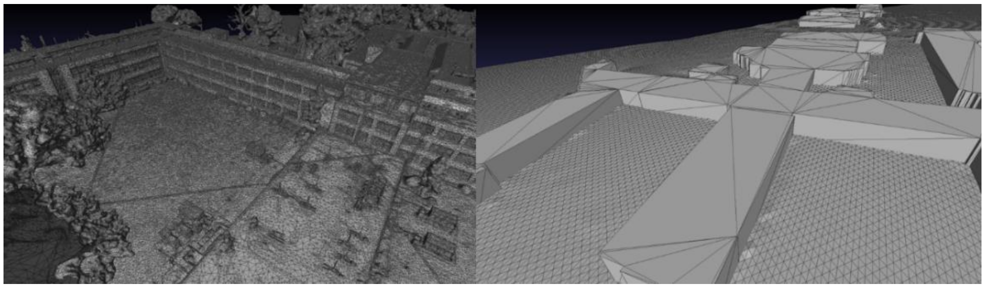

Figure 8. The original complete mesh (left) and the result of a semantics-aware mesh compression algorithm that compactly models ground and building and can throw out the remaining categories if they are not of interest (right). The original mesh contains 8,260,198 vertices and 16,321,204 faces in 1.25 GB file size, while the compressed result contains 83,338 vertices and 161,305 faces in 16.2 MB. The remaining classes can also be added to the simplified mesh at their original resolutions, or their simplified versions generated by standard, general-purpose mesh simplification algorithms.

To demonstrate this idea, we use the building and ground labels inferred by the system to decouple the meshing process for ground from those of the buildings. A low-resolution 2.5D representation for ground is used together with the rectilinear polygonization approach outlined in earlier work (Matei et. al., 2008) to model buildings at a higher resolution, and remove all other object categories from the meshification process, see Figure 8 for a comparison, and Figure 9 for an illustration of selectively varying the resolution of "ground" category.

Semantics-Aware Geo-registration of Ground to 3D Reference

Geo-registration refers to the problem of locating an agent in a larger-scale map of its surroundings, using the local data that is captured by its sensors. A variety of data modalities can be used as the reference map, including satellite or aerial imagery, but 3D models provide much richer information and therefore are better suited for the task when they are available. Often, the exact geographic location of such reference maps are known in advance, which in turn allows for the accurate localization of the said agent on a world map. Such an automatic localization capability is a powerful tool in the context of autonomous self-navigating robotic agents as well as AR/VR training and simulation scenarios where a human agent needs to be accurately placed inside particular scene over long periods of time.

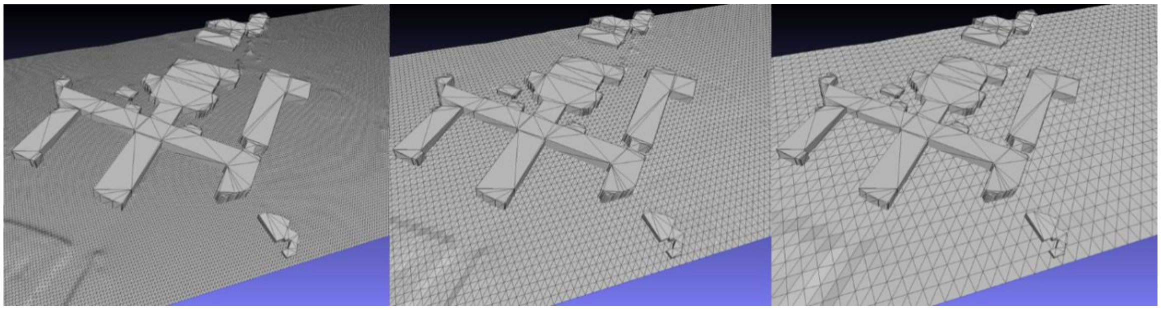

Figure 9. The presence of semantics allows for varying levels of detail for each category independent of other categories. Here, the resolution of 2.5D ground is progressively changed from 2m to 10m resolution while keeping the building resolution constant.

We show that, by providing a semantic understanding layer to both the reference 3D model and the local 2D images captured by an agent moving through a scene, it is possible to considerably improve the full 6-DOF pose estimation accuracy for the said agent, resulting in a more accurate motion trajectory that is computed automatically. In this experiment, the 3D reference is an aerial LIDAR capture and the agent is recording a video feed as it is traversing the scene. We use the full system to compute the semantics in the 3D model, and a standard image-based segmentation algorithm to compute the same semantic categories on each frame provided by the agent's video feed. Then we compute the 6-DOF pose of the agent's camera with and without the use of these semantic categories, and compare the results by using the estimated pose to render the 3D model from the viewpoint of the agent's camera. It can be seen that a significant improvement in the localization accuracy is obtained, even when relatively few semantic categories are used, as seen from Figure 10.

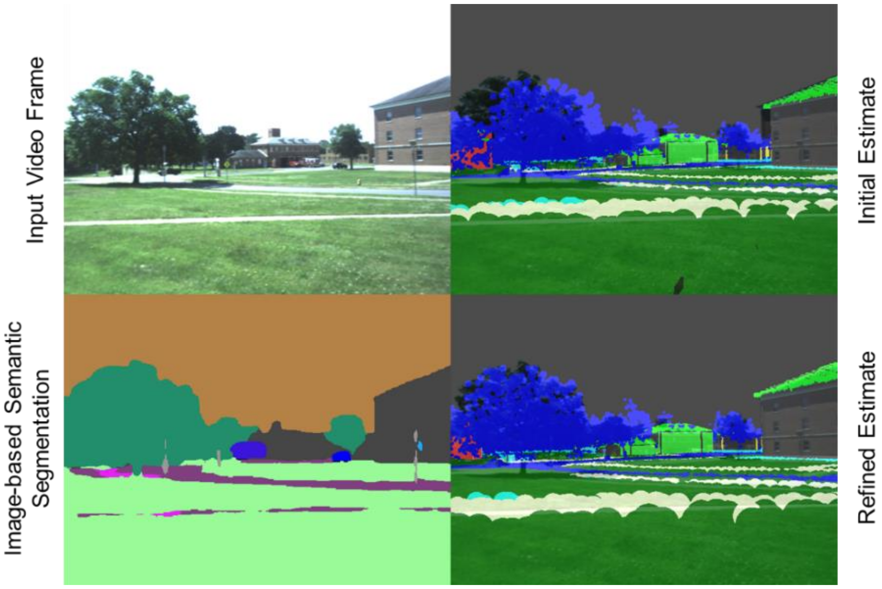

Figure 10. An input video frame (top-left) is overlayed on top of the 3D semantic reference layer (top-right) using the pose estimate obtained from image alone. When the semantic layer for the input image (bottom-left) is also used in the alignment process, the frame is better aligned with the 3D reference (bottom-right).

ACKNOWLEDGEMENTS

The work for this program was conducted with the support of the Office of Naval Research (ONR) Contract N00014-17-1-2949. The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official position, policy, or decision of the United States government.

REFERENCES

Achanta, R., & Shaji, A., & Smith, K., & Lucchi, A., & Fua, P., & Süsstrunk, S. (2012). SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(11), 2274-2282

Avidar, D., & Malah, D., & Barzohar, M. (2017). Local-to-global Point Cloud Registration Using a Dictionary of Viewpoint Descriptors. *IEEE International Conference on Computer Vision (ICCV)* (pp. 891-899)

Birdal, T., & Ilic, S. (2017). CAD Priors for Accurate and Flexible Instance Reconstruction. *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, (pp. 133-142)

Chen, X., & Ma, H., & Wan, J., & Li, B., & Xia, T. (2017). Multi-view 3D Object Detection Network for Autonomous Driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1907-1915)

Çiçek, Ö., & Abdulkadir, A., & Lienkamp, S. S., & Brox, T., & Ronneberger, O. (2016). 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (pp. 424-432)

Girshick, R., & Donahue, J., & Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 580-587)

Laefer, D. F., & Abuwarda, S., & Vo, A. V., & Truong-Hong, L., & Gharibi, H. (2017). 2015 Aerial Laser and Photogrammetry Survey of Dublin City Collection Record. *doi:10.17609/N8MQ0N*

Landrieu, L., & Obozinski, G. (2017). Cut Pursuit: Fast Algorithms to Learn Piecewise Constant Functions on General Weighted Graphs. *SIAM Journal on Imaging Sciences*, 10(4), 1724-1766

Landrieu, L., & Simonovsky, M. (2018). Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 4558-4567)

Lee, J. K., & Yea, J., & Park, M. G., & Yoon, K. J. (2017). Joint Layout Estimation and Global Multi-view Registration for Indoor Reconstruction. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 162-171).

Lin, T. Y., & Goyal, P., & Girshick, R., & He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (pp. 2980-2988)

Matei, B. C., & Sawhney, H. S., & Samarasekera, S., & Kim, J., & Kumar, R. (2008). Building Segmentation for Densely Built Urban Regions Using Aerial Lidar Data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1-8)

Qi, C. R., & Su, H., & Mo, K., & Guibas, L. J. (2017). Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 652-660).

Qi, C. R., & Yi, L., & Su, H., & Guibas, L. J. (2017). Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, (pp. 5099-5108)

Raguet, H., & Landrieu, L. (2018). Cut-Pursuit Algorithm for Regularizing Nonsmooth Functionals with Graph Total Variation. *arXiv preprint*, arXiv:1802.04383

Sikka, K., & Silberfarb, A., & Byrnes, J., & Sur, I., & Chow, E., & Divakaran, A., & Rohwer, R. (2020). Deep Adaptive Semantic Logic (DASL): Compiling Declarative Knowledge into Deep Neural Networks. *arXiv preprint*, arXiv:2003.07344

Su, H., & Jampani, V., & Sun, D., & Maji, S., & Kalogerakis, E., & Yang, M. H., & Kautz, J. (2018). Splatnet: Sparse Lattice Networks for Point Cloud Processing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 2530-2539)

Tatarchenko, M., & Dosovitskiy, A., & Brox, T. (2017). Octree Generating Networks: Efficient Convolutional Architectures for High-Resolution 3D Outputs. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (pp. 2088-2096)

Ulusoy, A. O., & Black, M. J., & Geiger, A. (2017). Semantic Multi-View Stereo: Jointly Estimating Objects and Voxels. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 4531-4540)

Vanegas, C. A., & Aliaga, D. G., & Beneš, B. (2010). Building Reconstruction Using Manhattan-world Grammars. *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, (pp. 358-365)

Wu, Z., & Song, S., & Khosla, A., & Yu, F., & Zhang, L., & Tang, X., & Xiao, J. (2015). 3D ShapeNets: A Deep Representation for Volumetric Shapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1912-1920)

Xiao, T., & Liu, Y., & Zhou, B., & Jiang, Y., & Sun, J. (2018). Unified Perceptual Parsing for Scene Understanding. *Proceedings of the European Conference on Computer Vision (ECCV)*, (pp. 418-434)

Zhou, B., & Zhao, H., & Puig, X., & Fidler, S., & Barriuso, A., & Torralba, A. (2017). Scene Parsing Through ADE20K Dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 633-641)

Zhou, B., & Zhao, H., & Puig, X., & Xiao, T., & Fidler, S., & Barriuso, A., & Torralba, A. (2019). Semantic Understanding of Scenes Through the ADE20K Dataset. *International Journal of Computer Vision (IJCV)*, 127(3), 302-321

Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end Learning for Point Cloud Based 3D Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 4490-4499)

Zolanvari, I., & Ruano, S., & Rana, A., & Cummins, A., & Smolic, A., & Da Silva, R., & Rahbar, M. (2019). DublinCity: Annotated LiDAR Point Cloud and its Applications. *British Machine Vision Conference (BMVC)*

Zou, C., & Yumer, E., & Yang, J., & Ceylan, D., & Hoiem, D. (2017). 3D-PRNN: Generating Shape Primitives with Recurrent Neural Networks. *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, (pp. 900-909)