




2020

Estimating Free-Flow Speed with LiDAR and Overhead Imagery

Armin Hadzic

University of Kentucky, armin.hadzic@uky.edu

Author ORCID Identifier:

 <https://orcid.org/0000-0002-8816-9366>

Digital Object Identifier: <https://doi.org/10.13023/etd.2020.061>

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

Recommended Citation

Hadzic, Armin, "Estimating Free-Flow Speed with LiDAR and Overhead Imagery" (2020). *Theses and Dissertations--Computer Science*. 95.

https://uknowledge.uky.edu/cs_etds/95

This Master's Thesis is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Armin Hadzic, Student

Dr. Nathan Jacobs, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

Estimating Free-Flow Speed with LiDAR and Overhead Imagery

THESIS

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in the College of
Engineering at the University of
Kentucky

By
Armin Hadzic
Lexington, Kentucky

Director: Dr. Nathan Jacobs
Associate Professor of Computer Science
Lexington, Kentucky 2020

Copyright© Armin Hadzic 2020
<https://orcid.org/0000-0002-8816-9366>

ABSTRACT OF THESIS

Estimating Free-Flow Speed with LiDAR and Overhead Imagery

Understanding free-flow speed is fundamental to transportation engineering in order to improve traffic flow, control, and planning. The free-flow speed of a road segment is the average speed of automobiles unaffected by traffic congestion or delay. Collecting speed data across a state is both expensive and time consuming. Some approaches have been presented to estimate speed using geometric road features for certain types of roads in limited environments. However, estimating speed at state scale for varying landscapes, environments, and road qualities has been relegated to manual engineering and expensive sensor networks. This thesis proposes an automated approach for estimating free-flow speed using LiDAR (Light Detection and Ranging) point clouds and satellite imagery. Employing deep learning for high-level pattern recognition and feature extraction, we present methods for predicting free-flow speed across the state of Kentucky.

KEYWORDS: transportation engineering, remote sensing, computer vision, deep learning

Author's signature: Armin Hadzic

Date: March 17, 2020

Estimating Free-Flow Speed with LiDAR and Overhead Imagery

By
Armin Hadzic

Director of Thesis: Nathan Jacobs

Director of Graduate Studies: Mirosław Truszczyński

Date: March 17, 2020

This work is dedicated to Admir. My brother has always been a source of encouragement, optimism, and guidance. He has been fundamental in shaping my life, and for that I could not be more grateful.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Nathan Jacobs. Working with him has been an enlightening experience that continually pushed the limits and scope of my knowledge. His enthusiastic teaching, mentorship, and candor has made graduate school the most dense and valuable learning experience in my life.

To my co-authors Weilian Song, Scott Workman, and others, I appreciate your generosity in sharing your work and knowledge. I have had the privilege of working with many insightful and generous coworkers, including: Hunter Blanton, M. Usman Rafique, Tawfiq Salem, David Jones, Sean Grate, Connor Greenwell, Yu Zhang, Virgil Barnard, and the rest of the lab. I was fortunate in being a part of this wonderful learning community that educated me and inspired me to pursue scholarship.

I would like to take the opportunity to thank my committee members Dr. Nathan Jacobs, Dr. Brent Harrison, and Dr. Mei Chen for their interest in my work and helpful feedback. And to my colleagues at the Johns Hopkins Applied Physics Laboratory, Ryan Mukherjee, Dr. Gordon Christie, Dr. Jeffrey Freeman, and Jeremy Ratcliff; it has been a privilege to work with you in the pursuit of helping others.

Last, I would like to recognize my parents Rasema and Suljo Hadzic for their support and generosity. To them, I owe my persistence, resilience, and passion for life. Their endless love and support has enriched my life.

Table of Contents

Acknowledgments	iii
Table of Contents	iv
List of Figures	vi
List of Tables	viii
Chapter 1 Introduction	1
1.1 Free-Flow Speed in Transportation Engineering	1
1.2 LiDAR and Remote Sensing	2
1.3 Estimating Free-Flow Speed using Deep Learning	2
1.4 Thesis Outline	4
Chapter 2 LiDAR Dataset Construction	5
2.1 Introduction	5
2.2 Related Work	6
2.3 Motivation	6
2.4 Methods	8
2.5 Data Fusion Examples	16
2.6 Conclusion	19
Chapter 3 Estimating Free-Flow Speed	20
3.1 Introduction	20
3.2 Related Work	21
3.3 Dataset	22
3.4 Methods	22
3.5 Evaluation	28
3.6 Conclusion	34

Chapter 4 Conclusion	35
4.1 Contributions	35
4.2 Future Work	36
Bibliography	37
Vita	42

LIST OF FIGURES

1.1	Airborne LiDAR (green) and satellite imagery (red) of a road segment are used to predict the free-flow speed of a given road segment.	3
2.1	Aerial LiDAR coverage of Kentucky by KyFromAbove [18] program. Each color corresponds to the flight identifier of a flight for a given LiDAR sweep.	7
2.2	LiDAR point cloud dataset construction process. The blue map of Kentucky is the KyFromAbove [18] aerial LiDAR tile distribution, where each blue square is a single LiDAR tile. Free-flow speed labels are used to select LiDAR tiles, and sample point clouds. The output of the process is a paired point cloud and raster center grid.	7
2.3	Example point clouds of different environments from the LiDAR dataset. The depicted point clouds are the same scenes shown in Figure 2.8 but are depicted at a different angle to illustrate the 3D capability of point clouds.	10
2.4	Example point clouds of different road types from the LiDAR dataset. The depicted point clouds are the same scenes shown in Figure 2.9, but are depicted at a different angle to illustrate the 3D capability of point clouds.	11
2.5	7x7 raster center points extracted for a single point cloud. The overhead image is sourced from the National Agriculture Imagery Program (NAIP).	12
2.6	Tile radial sampling approaches for an area depicted in (a). The reference image (a) is obtained from the National Agriculture Imagery Program (NAIP). Each strategy is designed to sample neighboring points around each center point selected from a 7x7 raster.	13
2.7	Tile sampling approaches for an area depict in Figure 2.6 (a). Distance weighted radial sampling with respect to a 7x7 center point raster for (a) 500 or (b) 100 points from a given center point. Sampled points from a linearly spaced 100x100 grid and 80x80 grid are shown in (c) and (d) respectively.	14
2.8	Example point clouds of different environments from the LiDAR dataset paired with satellite imagery. Point cloud coloring reflects the relative LiDAR intensities.	17
2.9	Example point clouds of different road types from the LiDAR dataset paired with satellite imagery. Point cloud coloring reflects the relative LiDAR intensities.	18

2.10	Example of a 5000 point densely sampled point cloud around a dirt road. Li-DAR light intensity is depicted in (a) and height in (b). The road example corresponds to the same road depicted in Figure 2.9 (c) and (d).	19
3.1	Image features are paired with point cloud features using a grid of raster center points (red dots), ensuring geo-spatial consistency between the two feature sets.	22
3.2	RasterNet general architecture diagram. Satellite images pass through a ResNet-based image encoder, while point clouds and raster center locations are passed through a point cloud encoder. Each cell of the point cloud feature map corresponds to a set of features of a local point cloud neighborhood. The two sets of features are then channel-wise concatenated before being passed through a shared model (ResNet block) to produce a free-flow speed prediction.	23
3.3	PointNet++ [33] style multi-scale grouping depicted for a point cloud (a) centered on a known free-flow speed label location. Grouping operations are performed around each of the 7x7 raster centers (red) at different scales and number of samples. Local point clouds (green) are grouped at varying sample sizes: 16 samples (b), 32 samples (c), and 128 samples (d).	25
3.4	Scatterplot of free-flow speed predictions on the test set from the RasterNet Learn model compared with known speed labels. Overlaid heatmap depicts higher point density in darker color. Optimal performance should follow the green line.	29
3.5	Ground truth and predicted speed maps for both Woodford (left small city) and Fayette (right larger city) counties in Kentucky, USA.	30
3.6	Ground truth and predicted speed maps for Campbell county Kentucky, USA.	31
3.7	Ground truth and predicted speed maps for Bell county Kentucky, USA.	32
3.8	Ground truth and predicted speed maps for Union county Kentucky, USA.	33

LIST OF TABLES

3.1	Simple Structural Features	24
3.2	Structural Statistics	26
3.3	Free-flow Speed Estimation Model Performances	28

Chapter 1

Introduction

1.1 Free-Flow Speed in Transportation Engineering

Managing roads and traffic over a large area is a challenging and expensive endeavor. According to the Kentucky Transportation Cabinet (KYTC) [3], the state of Kentucky (USA) has approximately 80,000 miles of public roads and 3 million licensed drivers. Kentucky roads are traveled 47.2 billion miles of each year. With this volume of traffic, the roads are in need of constant attention and maintenance. The KYTC requires state transportation engineering to follow a data-driven Highway Safety Improvement Program. The \$39 million program prioritizes highway engineering improvement projects that are targeted towards vehicle crash types and patterns, as supported by crash and road data. With Kentucky highway engineering costs accruing \$47 million of the \$700 million in total expenditures, reducing the cost of data collection could help the state reduce its \$2 million deficit [2]. Sourcing sufficient crash and road data for all state and local roads can be expensive and time consuming. One important type of road data required for evaluating designs and construction cost in transportation engineering is free-flow speed [29].

According to the USA Highway Capacity Manual 2010 (HCM) [25] free-flow speed is defined as the average speed of automobiles when operators are unaffected by delay or traffic volume. The HCM stipulates that, "the prevailing speed on freeways at flow rates between 0 and 1,000 passenger cars per hour per lane" [25] is the preferred method of measuring free-flow speed. This metric is directly used in not only determining the level of service (LOS) [26] and capacity of highways [25], but also the speed limits of roads.

The Manual of Uniform Traffic Control Devices [1] states, "When a speed limit within a speed zone is posted, it should be within 5mph of the 85th percentile speed of free-flow traffic." Furthermore, enforcement of posted speed limits directly impact driver behavior and in turn traffic flow [36]. Thus, understanding free-flow speed is useful in directly designing

traffic tools such as posted speed limits, but also in indirectly impacting driver behavior. An estimated 550,000 signs and 3,200 traffic signals across Kentucky [3] are designed or managed using road attributes such as free-flow speed. With the emerging capability of remotely controlled traffic devices and artificial intelligence, smart traffic devices have been shown to reduce travel time by 25%, delays by 40%, stops by 30-40% and emissions by 20% on a road network in Pittsburgh, Pennsylvania [37]. Smart traffic systems could lead to improvements in congestion control, reductions in crashes, and reductions traffic fatalities [10]. Analytic tools like state-wide free-flow speed mapping could become an integral component of such smart traffic systems (e.g. variable speed limits) [16].

1.2 LiDAR and Remote Sensing

Light Detection and Ranging (LiDAR) is a remote sensing method used to visually describe the topography of a scene using clouds composed of individual points. LiDAR operates by recording the distances measured by the surface reflectance of a pulsed laser on a given scene. Terrestrial LiDAR is used in autonomous driving [24] and remote sensing [34] applications due to its capability of describing 3D scenes at varying spatial resolutions. In the fields of robotics and autonomous driving, ground-based LiDAR has been used for tasks such as scene recognition [23], vehicle odometry, and environment mapping [47].

Airborne LiDAR is often used in remote sensing applications. In particular, overhead LiDAR can enable high resolution topographic mapping of natural geographies [7]. Similarly, Fedrigo et al. [9] have also shown the potential of combining LiDAR-derived structural profiles and species distribution models to create predictive maps of ecosystems in Australia. LiDAR-based mapping has also been used to describe man-made geographies such as the city of Dublin [20]. Since airborne LiDAR can depict both small (e.g. road shoulder) and large (e.g. city block) scale areas, it can be used to source descriptive features of roads. In this thesis, we explore using airborne LiDAR combined with overhead satellite imagery to estimate free-flow driving speed, as illustrated in Figure 1.1.

1.3 Estimating Free-Flow Speed using Deep Learning

Measuring free-flow speed across an entire state or country requiring extensive sensor networks or cumbersome manual estimation can strain transportation engineering budgets. Transportation engineers utilize free-flow speed modeling to reduce the need for directly measuring free-flow speed. Traditional approaches for free-flow speed modeling involved the use of highway geometric features such as lateral clearance, median type, number of

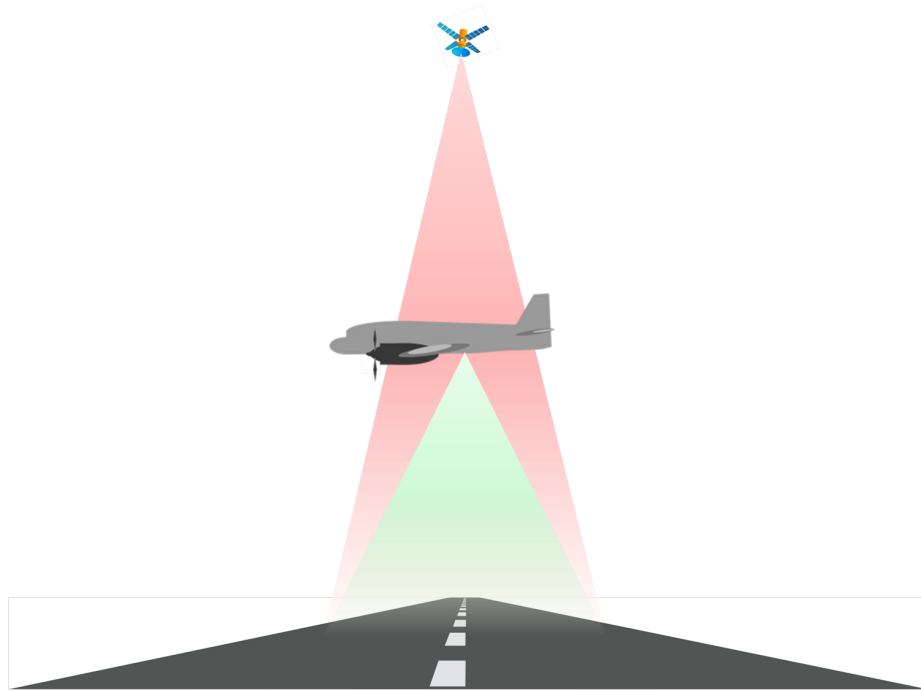


Figure 1.1: Airborne LiDAR (green) and satellite imagery (red) of a road segment are used to predict the free-flow speed of a given road segment.

access points, and lane [25]. For state-maintained roads, these geometric features are generally collected during road construction or improvement projects. Locally maintained roads generally do not record these features. Recent regression model approaches tend to be specific to certain network class type (arterial, local, collector) [36], or human geographies (urban and rural) [35]. Deardoff et al. [5] proposed a method of estimating free-flow speed directly from posted speed limits in Rapid City, South Dakota and neighboring highways. Since these methods are dependent on the availability of geometric highway features, they are limited to well documented roads. However, statewide free-flow speed estimation is possible with learning models using abundant visual data across an entire state.

Operators tend to drive vehicles based on visual feedback such as road size, prevalence of congestion, and scene. Recent approaches using convolutional neural networks found that satellite images of road segments can enable a model to identify the intrinsic visual patterns in roads that correlate with free-flow speed [38]. Combining these visual features with road features further improves prediction performance, but limits model applicability only to sufficiently documented roads. However, geometric road features can be extracted from LiDAR point clouds based on the spatial relationship between elements within a scene. Our approach utilizes deep learning models for large-scale free-flow speed estimation using ample raw visual data (LiDAR and satellite imagery).

1.4 Thesis Outline

The subsequent chapters of this thesis are outlined as follows:

- **Chapter 2** presents the processes for creating a real world LiDAR point cloud dataset for free-flow speed estimation using airborne LiDAR. We also investigate how point clouds could complement overhead imagery for describing roads.
- **Chapter 3** discusses a set of deep learning models that can be used to accurately estimate the free-flow speed. We introduce a raster-based data fusion method for geospatially pairing point clouds with an existing satellite imagery dataset. Our multi-modal models demonstrate the predictive performance of combining overhead image and point cloud features using a raster-based neural network.
- **Chapter 4** summarizes the contributions of the thesis and provides closing remarks. The chapter concludes by outlining potential future work in extending the LiDAR and overhead imagery dataset, exploring combined data representation, and inspiring new neural network architectures for free-flow speed estimation.

Chapter 2

LiDAR Dataset Construction

2.1 Introduction

Airborne LiDAR provides geometric representations of scenes in the form of point clouds. Point cloud representations of scenes are spatially consistent with scene structure seen in the natural world. This characteristic makes point clouds a useful depiction of large areas such as road segments and their surrounding environment. Geometric characteristics of roads such as lane width, slope, and curvature can be described by point clouds. The task of free-flow speed estimation has traditionally made use of many of these road characteristics [25]. Environment area descriptions of roads used in free-flow speed estimation [35], such as whether roads are in urban or rural areas, can also be encoded into point clouds. LiDAR point cloud's 3D representation of geography ensures that many physical road features can be extracted and incorporated into improving free-flow speed modeling.

We present a new point cloud dataset for estimating average free-flow driving speed. Our dataset is composed of airborne LiDAR point clouds of roads across the state of Kentucky and vehicle free-flow speeds. The dataset can be used to model the relationship between road topography and public free-flow driving speeds. Additionally, our dataset was designed to pair with satellite imagery from Song et al.'s [39] dataset, enabling data fusion driven methods. To the best of our knowledge, this is the first dataset developed for estimating free-flow vehicle driving speeds utilizing airborne LiDAR point clouds of urban and rural roads in the state of Kentucky.

2.2 Related Work

Recently, learning-based approaches have been introduced to address challenging automobile traffic problems. Remote sensing datasets have been developed in response to the volume of data required by these learning based approaches. Naphade et al. [27] presented a video surveillance dataset for vehicle speed estimation, anomaly detection, and vehicle re-identification for the 2018 NVIDIA AI City Challenge. For the 2019 NVIDIA AI City Challenge Naphade et al. [28] presented a video dataset using traffic cameras for large scale vehicle tracking and re-identification, in addition to anomaly detection. While both datasets utilize video to predict vehicle traffic characteristics, these characteristics are at individual vehicle scale. However, in order to address the challenge of free-flow speed estimation at a statewide scale, other sensing technologies are preferable over video due to its cumbersome size and lack of availability on most roads.

LiDAR has become increasingly popular in autonomous driving applications, making a strong case for introducing it to transportation engineering challenges. Road-side LiDAR has been utilized in extracting vehicle trajectories [40], estimating free-flow speed, and vehicle classification [41]. The superior spatial area coverage of airborne LiDAR over road-side LiDAR qualifies it for use in large scale applications. Airborne LiDAR is deployed for tasks such as road obstruction detection [17] and vehicle detection [22,46]. Inspired by the statewide airborne LiDAR applications, we developed a LiDAR point cloud dataset to cover roads across Kentucky.

Most similar to our work, Song et al. [38] constructed a dataset using $400 \times 400 m^2$ satellite imagery paired with the Kentucky Transportation Center free-flow speed data. We used the same Kentucky free-flow speed data for our dataset, but opted for high resolution point cloud data in lieu of large scale overhead imagery. Unlike overhead imagery, our dataset can be used at varying resolutions and scales. Our dataset allows for feature extraction to range from fine-grain lane markers, useful for tasks such as road safety evaluation [38], up to satellite image scale city centers.

2.3 Motivation

Image-only methods for estimating free-flow speed, such as Song et al. [38], are capable of utilizing two dimensional visual features for roadway pattern recognition. However, such methods lack sufficient features for identifying patterns in the height dimension. Spatial patterns in the height dimension can help extract high level road features such as road slope and lateral clearance, which can be useful for free-flow speed estimation. Height

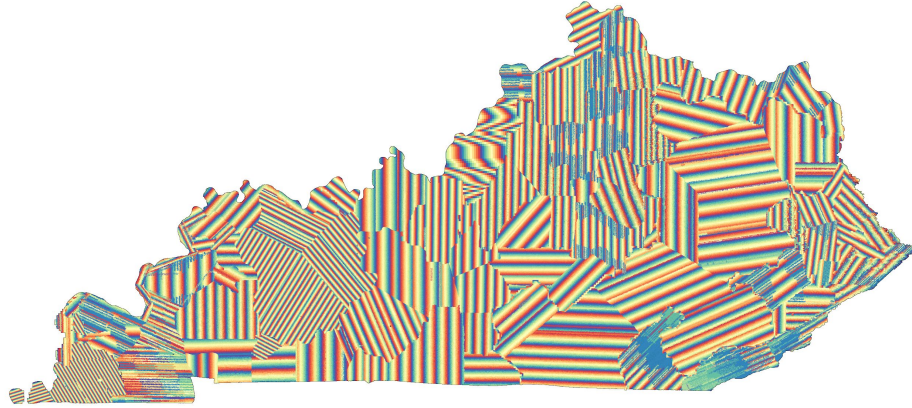


Figure 2.1: Aerial LiDAR coverage of Kentucky by KyFromAbove [18] program. Each color corresponds to the flight identifier of a flight for a given LiDAR sweep.

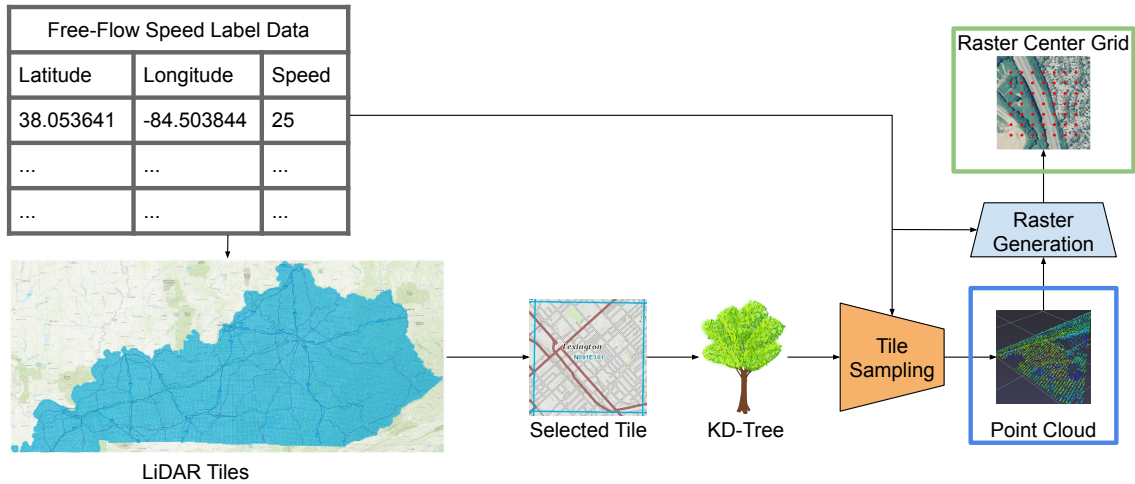


Figure 2.2: LiDAR point cloud dataset construction process. The blue map of Kentucky is the KyFromAbove [18] aerial LiDAR tile distribution, where each blue square is a single LiDAR tile. Free-flow speed labels are used to select LiDAR tiles, and sample point clouds. The output of the process is a paired point cloud and raster center grid.

dimension features can be represented by overhead LiDAR point clouds, which are capable of geometrically describing a scene in three dimensions. Additionally, point clouds could be paired with overhead imagery to provide both visual and 3D geometric representations of roads in order to further improve free-flow speed estimation. Our objective was to construct a point cloud dataset of roads across the state of Kentucky that could be used for free-flow speed estimation, and could be integrated into data fusion approaches. Moreover, models could utilize the 3D spatial information and gray-scale color intensities of road segments in this dataset to predict the free-flow speed of the road segments.

2.4 Methods

We present a new point cloud dataset for free-flow speed estimation of road segments. The dataset contains 3D point clouds of road segments that geospatially correspond with georeferenced free-flow speed labels. While this dataset was designed in such a manner that it could be paired with satellite imagery for data fusion approaches, it can also function as a standalone free-flow speed estimation dataset. Point cloud representations of road segments could be combined with representations from other modalities by coordinating data representations using geospatial coordinates.

The Kentucky Transportation Center aggregated HERE technologies speed data across uncongested periods to extract free-flow speeds of road segments and their corresponding geospatial coordinates. The speed data expressed free-flow speed of a vehicle on a road segment as the speed at which an operator would drive in uncongested traffic on weekdays during non-peak hours (9am-3pm), and outside of holiday hours. Free-flow speed data was recorded in 2014 and was averaged across the year for each respective road segment. Interpolating between the two ends of a road segment was used to reduce each segment to a single spatial location.

Training, validation, and test dataset partitioning followed the methodology established by Song et al. [38]. Roads within the borders of the following Kentucky, USA counties were held-out for the test set: Bell, Lee, Ohio, Union, Woodford, Owen, Fayette, and Campbell. All other Kentucky roads that were present in the label data but were not part of the test set were incorporated into the training and validation sets. The validation set was constructed from 1% of the random samples not included in the test set. Altogether, this dataset allows image-only, point cloud-only, and hybrid image and point cloud models to be trained and evaluated equally on the challenging task of free-flow speed estimation. The dataset is representative of rural, urban, highway and arterial roads across the state of Kentucky. Roads range in structure from one-way dirt/gravel roads to multi-lane highways. The point clouds allow for high level features to be extracted, such as change in elevation, road curvature, lane delineation markings, lane width, and proximity to neighboring structures.

An overview of urban (a) and rural (b) road segments represented as point clouds is shown in Figure 2.3. Point cloud colors in both figures portray the gray-scale light intensity of the airborne LiDAR. The 3D urban road point cloud example illustrates the impact the height dimension can have in describing a scene. The dark blue roadways are shown at lowest elevation compared to the roof tops of small buildings on sky scrapers (top right roof top). Rural landscapes depicted by point clouds can express the dynamic topogra-

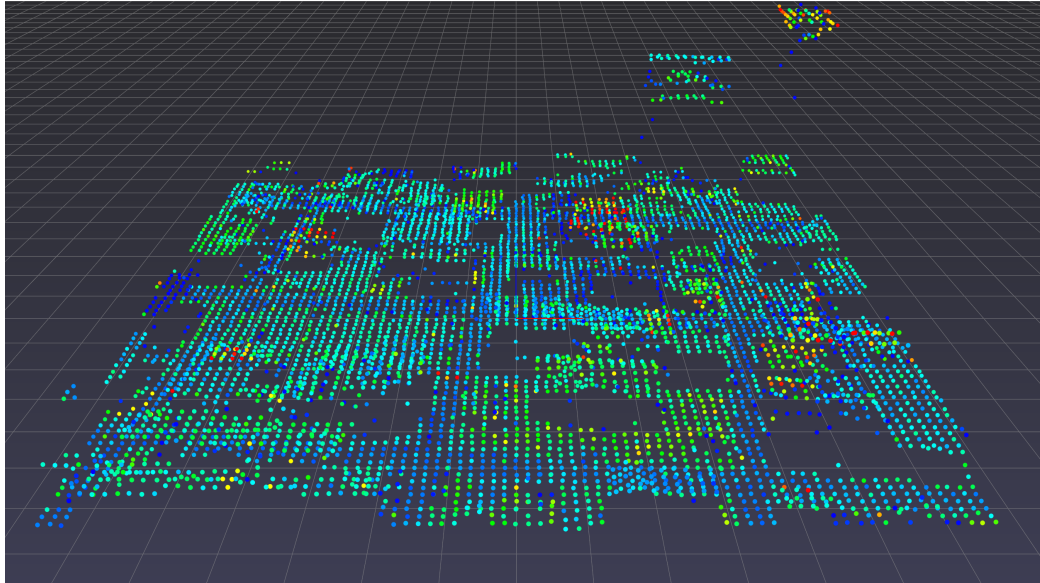
phy of landscapes neighboring roads, as shown in the rural road example. Additionally, examples of highway (a) and dirt road (d) point clouds are depicted in Figure 2.4. The highway example demonstrates that point clouds can describe road slope, a feature which can impact free-flow speed. Conversely, dirt roads represented by point clouds uniquely show an absence of clear road features, which could potentially be used to identify unpaved roads when compared with other road types. As depicted in Figure 2.2, construction of the point cloud dataset consisted of LiDAR tile selection, point cloud sampling, and raster generation.

2.4.1 LiDAR Tiles

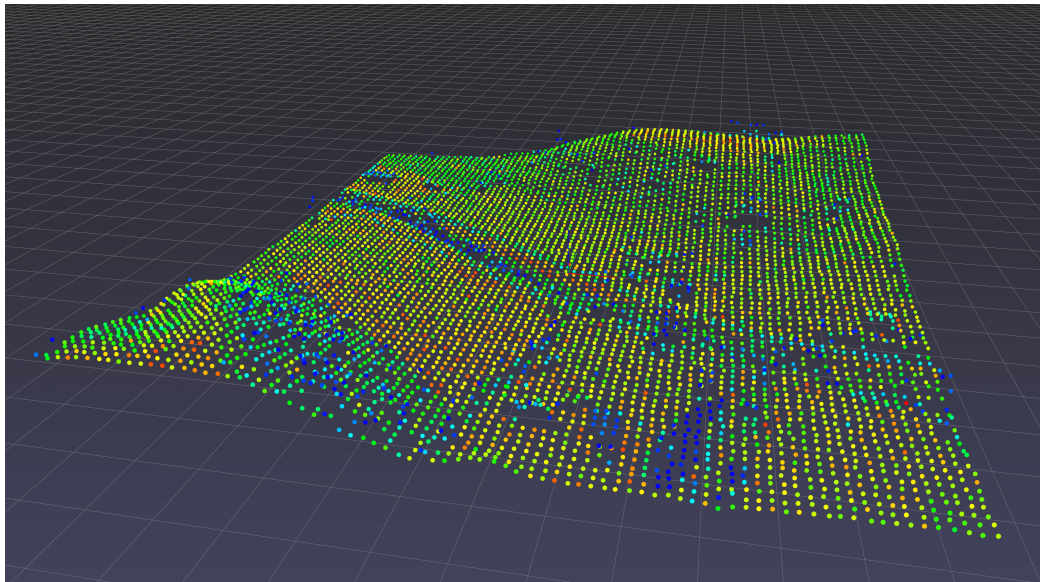
The Kentucky Division of Geographic Information’s KyFromAbove [18] program collected aerial LiDAR data over the state of Kentucky USA, with the objective of constructing a base-map of the Commonwealth of Kentucky. Aerial LiDAR data was collected by multiple flights and several aircraft in order to cover the entire state of Kentucky, as shown by the flight routes in Figure 2.1. From 2010 to 2017, LiDAR sweeps were conducted during leaf-off (winter to spring) seasons to minimize the impact of foliage on point cloud data. Flights were organized to ensure $\geq 20\%$ overlap to maximize the prevalence of usable data [19]. In accordance with the United States Geographical Survey (USGS) LiDAR Base Specification QL2, all points were expected to have vertical accuracy requirements of 95% confidence vertical accuracy $\leq 19.6cm$ for non-vegetated areas and 95% confidence vertical accuracy $\leq 29.4cm$ for vegetated areas [13]. The LiDAR data was stored as 31,135 LAZ tiles; each of which corresponds to a spatial region of Kentucky, as depicted in the blue map of Kentucky in Figure 2.2. We selected tile attributes that were visually descriptive of roads such as the three dimension point cloud spatial coordinates and 16-bit intensity values [19].

2.4.2 Point Cloud Sampling

We paired each free-flow speed label from the Kentucky Transportation Center data with the corresponding neighboring point cloud. The label data curated by Song et al. [38] paired each free-flow speed label with a corresponding WGS 84 (World Geodesic System) geo-spatial coordinates. Each label coordinate was transformed into the Kentucky State Plane (KSP) projection. Then an R-tree of LiDAR tile geo-spatial locations was used to perform a nearest neighbor search for the tile that contained the given KSP coordinate and label. Tile-label pairings were then sorted lexicographically by tile identifier, chunked into 16 sets, and distributed across 16 parallel machines to minimize computational runtime.

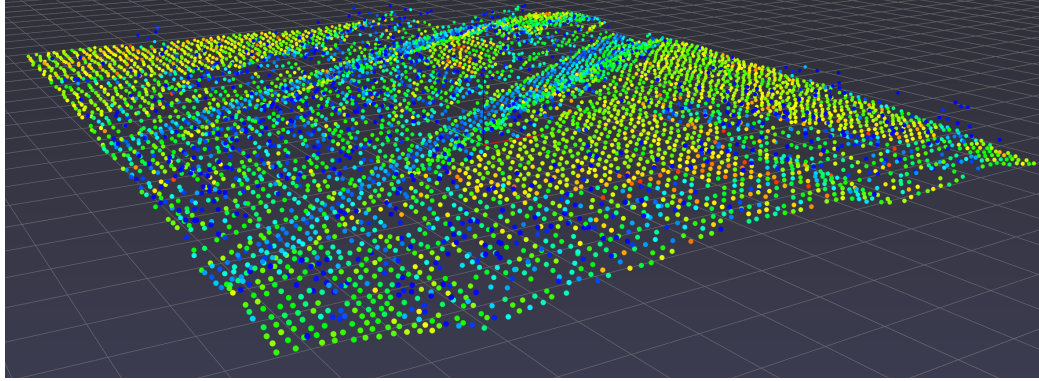


(a) Urban

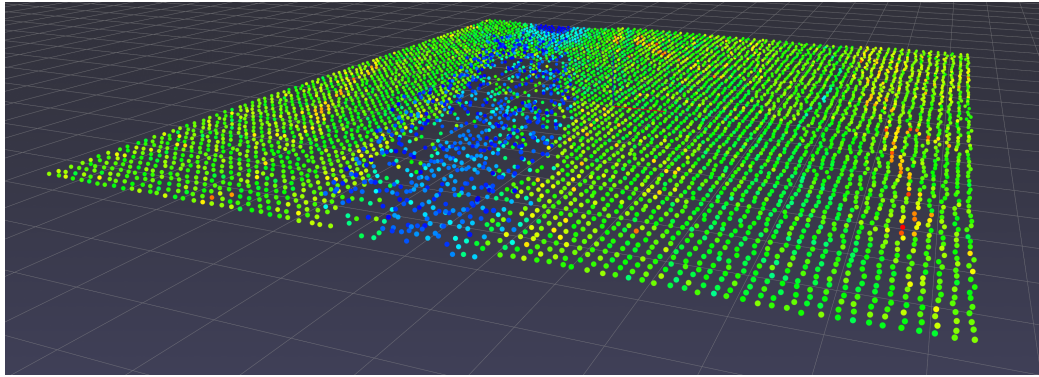


(b) Rural

Figure 2.3: Example point clouds of different environments from the LiDAR dataset. The depicted point clouds are the same scenes shown in Figure 2.8 but are depicted at a different angle to illustrate the 3D capability of point clouds.



(a) Highway



(b) Dirt Road

Figure 2.4: Example point clouds of different road types from the LiDAR dataset. The depicted point clouds are the same scenes shown in Figure 2.9, but are depicted at a different angle to illustrate the 3D capability of point clouds.

LiDAR tiles were then selected according to the tile identifier and 40% of the points were randomly sampled. The sampled points were read into a k-d tree for fast lookup. As detailed in section 2.4.4, multiple tile sampling strategies were examined and evaluated with respect to point cloud descriptiveness and number of points required. We found that an 80x80 point grid sampling method around the target label’s location was the most appropriate, as shown in Figure 2.7 (b). The 80x80 sampling grid was built by constructing a $400 \times 400 m^2$ bounding box centered on the target label location, and then 80 linearly spaced points were selected for each row and column. The tile’s k-d tree was then used to find the nearest point to each of the sampling grid’s point locations.

The x and y dimensions of the sampled point clouds were then reformulated to represent the difference between a given point and the label location point at the center of the point cloud. The z dimension of the point cloud and the point intensity values were normalized. All point clouds were then rotated such that the direction of travel of the target road was always pointed north, similar to how the imagery was aligned. Next, a 7x7 grid of raster

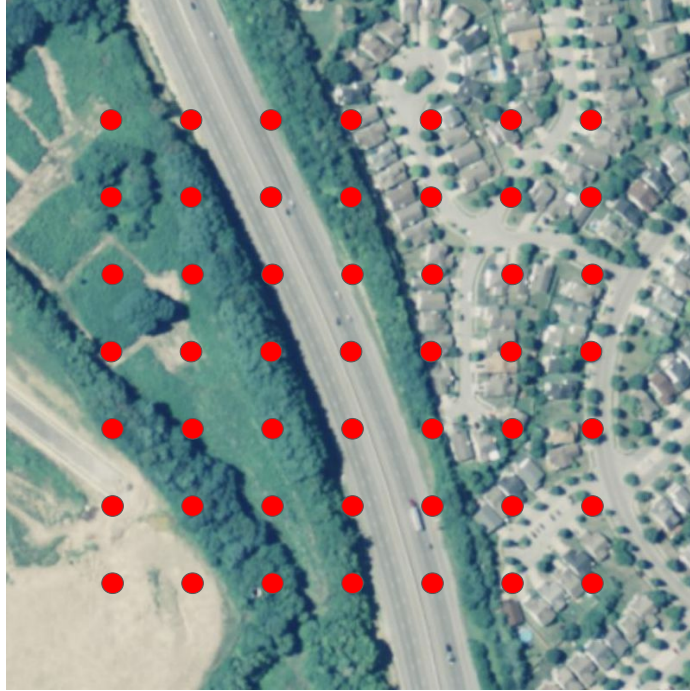
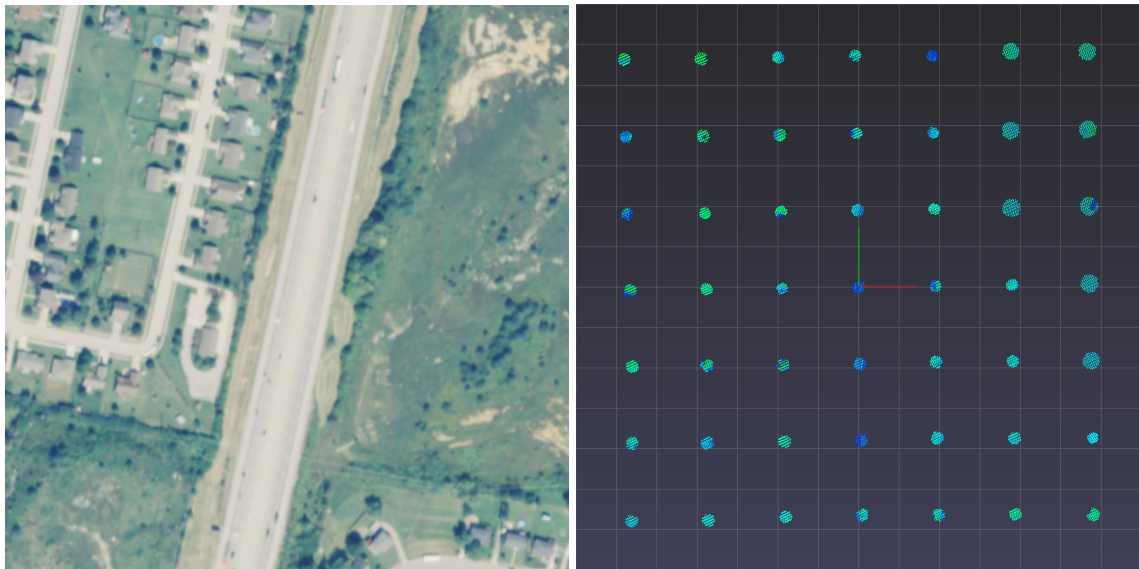


Figure 2.5: 7x7 raster center points extracted for a single point cloud. The overhead image is sourced from the National Agriculture Imagery Program (NAIP).

center locations was generated around the label location, as outlined in subsection 2.4.3. Each of the raster center locations and tile’s k-d tree were then used to sample the nearest point to a given location. Next, the raster center LiDAR points were centered in the same manner as the extracted point cloud. Last, the point clouds, raster center grids, and labels were batched and stored as binary files in order to minimize data loading runtime during training and evaluation.

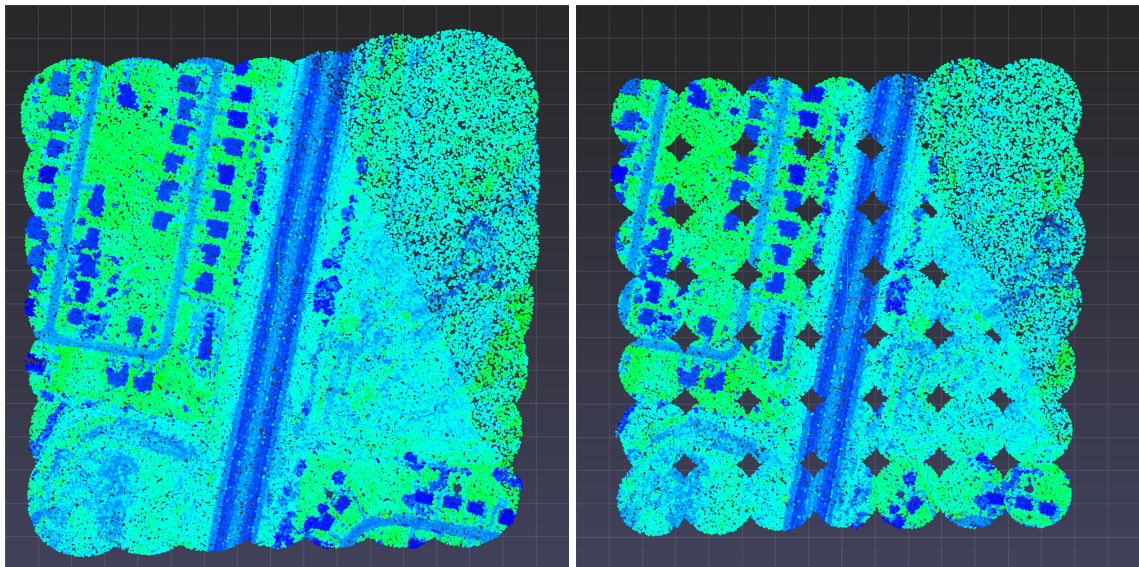
2.4.3 Raster Center Generation

For each point cloud around a target label, a raster center point grid was generated. The raster center point grid was used to bind point cloud features to a raster. The bound point cloud features could then be paired with image features in a spatially consistent manner. The raster center point grid was generated by constructing a bounding box with a $400 \times 400 m^2$ area centered on the target label’s geo-spatial location. Locations were then selected by linearly spacing 8 points $t_i \in t_0, t_1, \dots, t_8$ across the top edge of the bounding box, and repeating for the right edge of the bounding box. Halfway between any two adjacent locations, a raster center point was selected $c_i = \frac{r_i - r_{i-1}}{2}$. The resulting 7 point column and 7 point row were then used to populate a 7x7 grid of equally spaced points $c_j \in \{c_0, c_1, \dots, c_{49}\}$, as shown in Figure 2.5. Each point in the 7x7 grid expresses the



(a) Reference Image

(b) 100 Nearest Neighbors

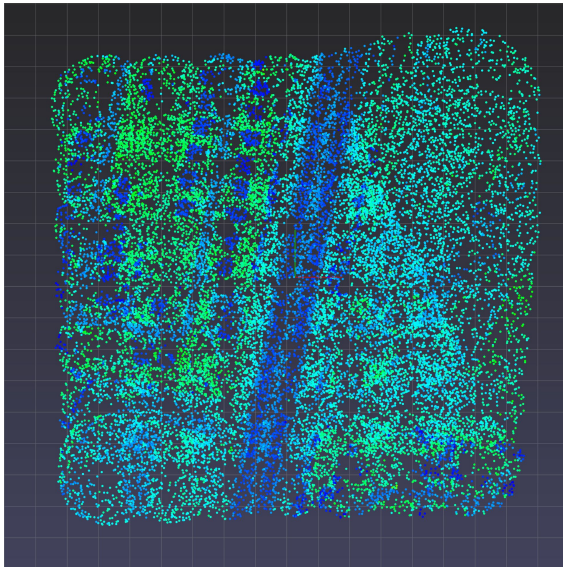


(c) 5000 Random Dense Sampling

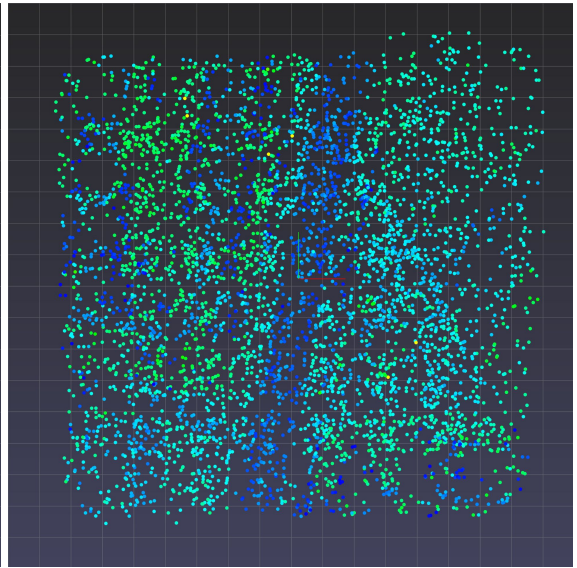
(d) 2000 Random Dense Sampling

Figure 2.6: Tile radial sampling approaches for an area depicted in (a). The reference image (a) is obtained from the National Agriculture Imagery Program (NAIP). Each strategy is designed to sample neighboring points around each center point selected from a 7x7 raster.

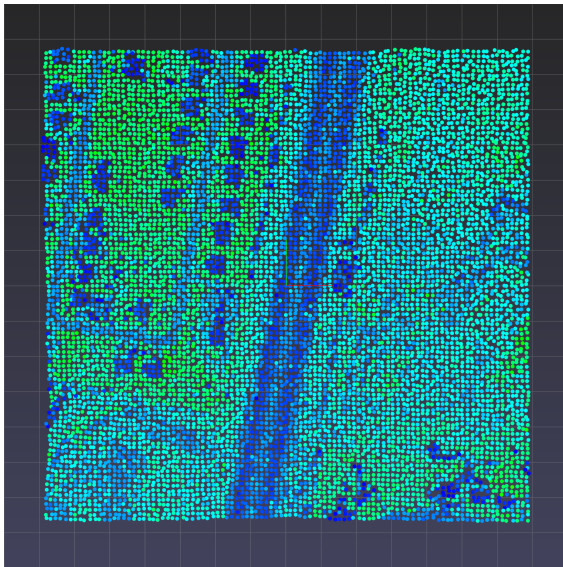
center point of a corresponding cell in a 7x7 raster grid. Similar to the label coordinates, the raster center locations were then transformed into the KSP projection in order to be in a common coordinate frame with the LiDAR tiles.



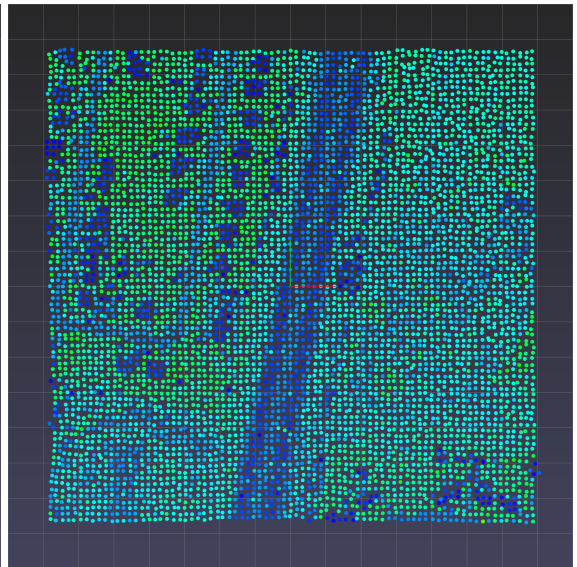
(a) 500 Distance Weighted Radial Sampling



(b) 100 Distance Weighted Radial Sampling



(c) 100x100 Grid Sampling



(d) 80x80 Grid Samplings

Figure 2.7: Tile sampling approaches for an area depict in Figure 2.6 (a). Distance weighted radial sampling with respect to a 7×7 center point raster for (a) 500 or (b) 100 points from a given center point. Sampled points from a linearly spaced 100×100 grid and 80×80 grid are shown in (c) and (d) respectively.

2.4.4 Tile Sampling Strategies

A tile sampling experiment was performed in order to select a tile sampling strategy that maximized the visual descriptiveness of the resulting point cloud, while using the fewest points necessary. Should a specific strategy yield point clouds that were visually descriptive, there is a higher likelihood that said samples would also be informative for a model trying to estimate free-flow speed. By nature, LiDAR point clouds lack consistent structure in their point distribution, meaning a high density of points might be necessary to adequately describe a scene. However, collecting dense point cloud samples leads to large datasets that can be difficult to use in a timely manner. Low number of points per point cloud was encouraged in order to reduce the training time of models using the dataset.

Since point cloud features would be grouped around the raster center points, our first sampling strategy involved selecting the 100 nearest neighboring points around each raster center point. An example of the raster nearest neighbor strategy is shown in Figure 2.6 (b). Comparing this sample to the corresponding satellite image (shown in Figure 2.6 (a)) suggests the sampling strategy is not visually descriptive, and likely a poor sampling strategy for describing the entire roadway scene.

Second, we experimented with dense random sampling k points around each raster center. As Figure 2.6 (c) illustrates, random dense sampling $k = 5000$ points around each raster center resulted in a highly descriptive point cloud when compared to the reference image Figure 2.6 (a). However, this sampling approach requires 254,000 points per sample. 254,000 points is high when considering that learning models using the popular point cloud benchmark dataset ModelNet40 [45] is commonly sample 1,024-5,000 sized point clouds [32, 33]. Similarly, when dense random sampling $k = 2000$ points the resulting point cloud was visually descriptive (see Figure 2.6 (e)) but required 98,000 points per sample. Interestingly, Figure 2.6 (e) depicts a challenging characteristic of LiDAR point cloud data. Namely, some local point clouds are densely packed while others are more scattered. As a result, for our next approach we tried to increase control of sampling around the raster centers.

Next, we experimented with weighted sampling of points to construct a point cloud. Let x_i be a point set $x_i \in \{x_0, x_2, \dots, x_n\}$ and $c_j \in \{c_0, c_1, \dots, c_{49}\}$ be a given raster center point. Weights were calculated by $w(x_i, c_j) = d(c_j, x_i)$. Weights w were assigned based on the distance (euclidean distance, d) between a given raster center point c_j and any other point x_i in the point cloud. This weighting scheme ensured that weighted sampling would select points that were closer to raster centers more often than points that were further away. We call this sampling strategy distance weighted radial sampling, as shown in Figure 2.6

$k = 500$ (d) and $k = 100$ (f). Setting $k = 100$ produced a point cloud that was not visually descriptive, while the $k = 500$ point cloud was moderately descriptive but required 24,500 points per sample. Neither of the radial sampling approaches produced a point cloud that was visually descriptive and that required few ($n \approx 5000$) points per sample.

Last, uniform grid sampling was used to uniformly describe a scene using a limited number of points. We started by creating a $400 \times 400 m^2$ bounding box centered on the target label. We then created a linearly spaced grid of length l that covered the entire bounding box area. Setting $l = 100$ (see Figure 2.7 (a)) resulted in a visually descriptive point cloud of the region depicted in Figure 2.6 (a), but required 10,000 points. However, setting $l = 80$ (see Figure 2.7 (b)) resulted in a similar visually informative point cloud using only 6,400 points. As a result, we chose to use $l = 80$ grid sampling strategy to generate point clouds for the point cloud dataset.

2.5 Data Fusion Examples

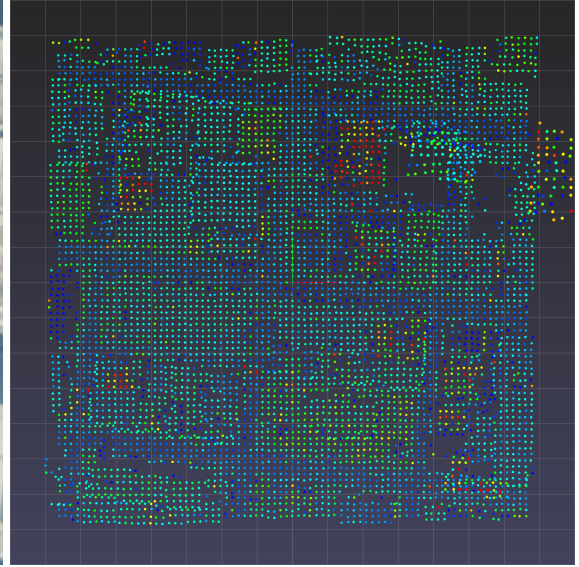
A qualitative analysis was performed on the point cloud dataset to evaluate whether visually descriptive point clouds could complement corresponding satellite images. A montage of point cloud examples from the dataset paired with satellite imagery of the same area are shown in Figures 2.8 and 2.9. Figure 2.8 (b) depicts an example of an urban environment where roofs (e.g. sky-scraper in red color) and streets are clearly distinguishable. Large building shadows, which can be problematic for some convolution neural network models, are shown affecting the satellite image in example (a). However, the urban road point cloud is unaffected by time of day lighting conditions. Example (d) shows a rural road where roads (dark blue lines) are distinguishable from grass and foliage from trees (speckled blue dots).

Figure 2.9 depicts an example of a highway (b) in Kentucky. The highway point cloud is topographically detailed, allowing streets and lanes to be clearly identified. More dense sampling can also be used to reveal fine grain details such as lane delineation. Road slope can be a useful features for estimating free-flow speed. The 2D highway satellite image (a) shows little indication of road slope. The 3D point cloud representation of the same road segment, depicted in Figure 2.4, reveals some road slope exists for the road segment. The 3D spatial features of the highway point cloud could complement the corresponding satellite image for data fusion approaches to free-flow speed estimation.

The dirt road (d) point cloud in Figure 2.9 is less clear than point clouds for all other aforementioned road types. While the neighboring forest (dark blue spots) is identifiable, the intensity of dirt roads is not distinguishable from the surrounding terrain. Examining



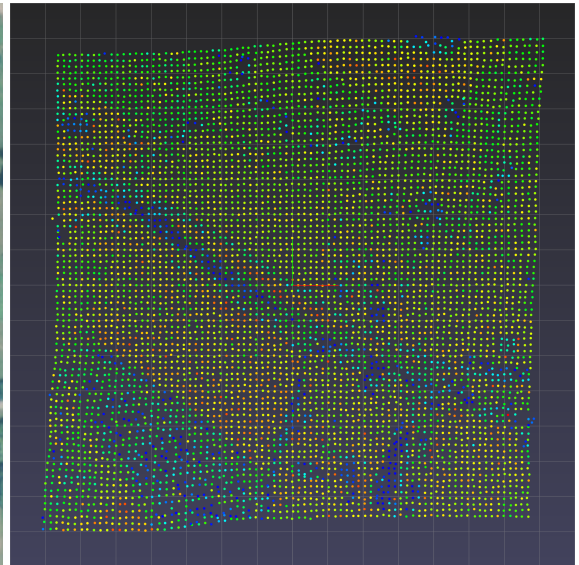
(a) Urban Satellite Image



(b) Urban Point Cloud



(c) Rural Satellite Image

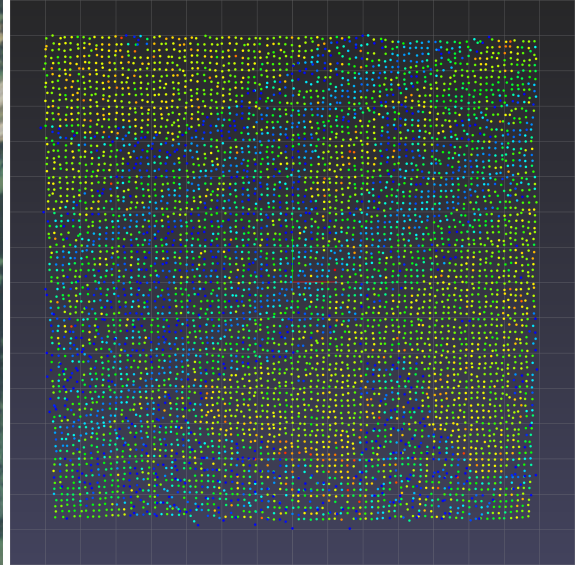


(d) Rural Point Cloud

Figure 2.8: Example point clouds of different environments from the LiDAR dataset paired with satellite imagery. Point cloud coloring reflects the relative LiDAR intensities.



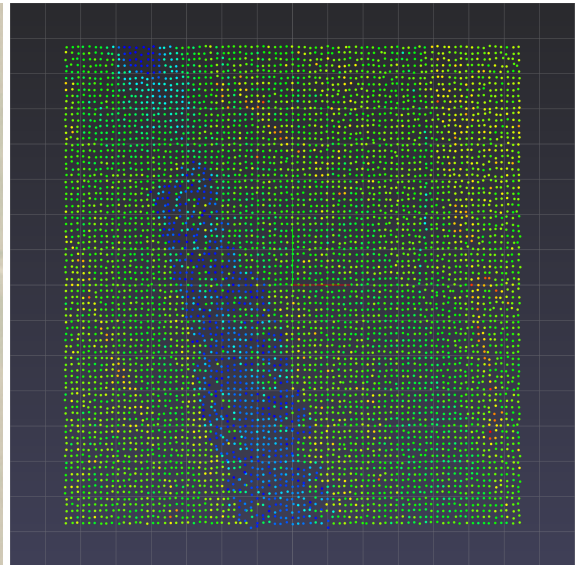
(a) Highway Satellite Image



(b) Highway Point Cloud



(c) Dirt Road Satellite Image



(d) Dirt Road Point Cloud

Figure 2.9: Example point clouds of different road types from the LiDAR dataset paired with satellite imagery. Point cloud coloring reflects the relative LiDAR intensities.

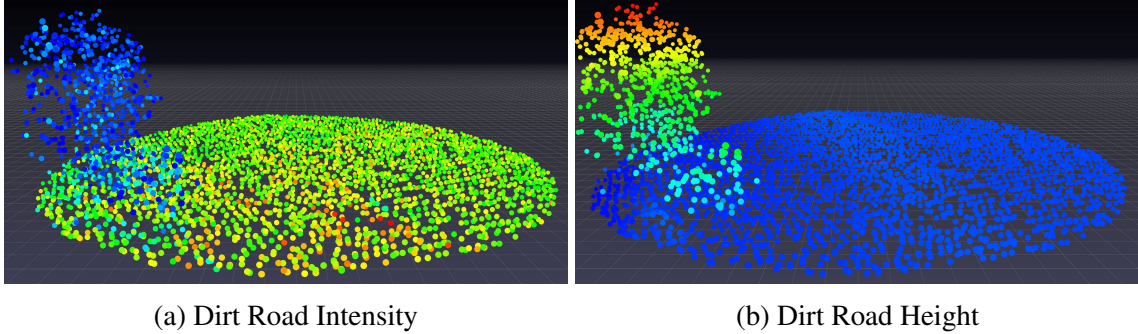


Figure 2.10: Example of a 5000 point densely sampled point cloud around a dirt road. LiDAR light intensity is depicted in (a) and height in (b). The road example corresponds to the same road depicted in Figure 2.9 (c) and (d).

the z-dimension of the dirt road in Figure 2.4 (b) suggests that z-dimension is not sufficiently descriptive in order to identify dirt roads either. However, highway roads (a) are distinguishable in the z-dimension, the same holds true for both urban and rural roads from Figure 2.3 (a and b respectively). These examples suggest that dirt roads can be particularly challenging to represent with grid sampled LiDAR point clouds. Changing tile sampling strategies to dense sampling for dirt roads does not produce more informative features, as shown by the uninformative light intensity (a) and height channels (b) of a dirt road in Figure 2.10. However, examining the corresponding dirt road satellite image from Figure 2.9 (c) suggests that dirt roads are visible from imagery, though with less clarity than non-dirt roads (image a and from Figure 2.8 images (a) and (c)). Pairing imagery with point clouds could be beneficial for predicting the free-flow speed of dirt roads.

2.6 Conclusion

We presented a new LiDAR point cloud dataset of Kentucky, USA for free-flow speed estimation. Our dataset provides 3D point clouds that can be paired with the Song et al’s dataset [38] satellite imagery dataset. Additionally, our dataset supplies raster centers for each point cloud, which allow locally (7x7 grid) sampled point cloud neighborhoods to be spatially consistent when paired with corresponding overhead image features. We also present multiple LiDAR tile sampling strategies for point cloud extraction to suit different dataset configurations. Last, through qualitative comparison we illustrate how point cloud representations of roads could complement overhead images of roads.

Chapter 3

Estimating Free-Flow Speed

3.1 Introduction

While image based approaches tend to perform well on many tasks, such as free-flow speed estimation [38], they can be limited by changes in season, cloud coverage, lighting, and viewpoint. Point clouds, on the other hand, can be a useful alternative for scene representation that is viewpoint invariant and robust to weather and lighting conditions. Scenes represented by point clouds have their own limitations, primarily related to their lack of rigid and consistent structure. We rectify the structural limitation of point clouds and lighting difficulties of images by employing data fusion. Data fusion is a method of combining multiple data sources (such as point cloud and satellite imagery) in order to produce more robust, informative, and consistent features than either source could provide individually. We organize local point cloud neighborhoods into a raster and pair them with spatially consistent image features. For a given road segment, our data fusion approach combines the structured visual features of overhead imagery with the geometric features of point clouds.

We introduce RasterNet, a new neural network architecture that combines point cloud and image features with spatial consistency for the task of free-flow speed estimation. RasterNet is trained and evaluated on a large free-flow speed estimation dataset composed of satellite imagery geospatially paired with point clouds. Our approach reaches state of the art performance on the Kentucky free-flow speed test set [38] without using provided highway geometric features. Our primary contributions can be summarized as follows:

- A method for fusing large-scale ($400 \times 400 m^2$) overhead satellite imagery and aerial LiDAR point clouds using a geospatially consistent raster structure.
- A series of neural network architectures for free-flow speed estimation using multi-modal learning, improving the state-of-the-art in free-flow speed prediction on Song

et al. [38] speed data.

- An evaluation of single modality and multi-modal free-flow speed estimation models using overhead imagery and/or point clouds.

3.2 Related Work

Point Cloud Representations: Qi et al’s seminal work on point cloud feature extraction with PointNet [32] introduced how deep neural networks could be used with point clouds for tasks such as classification and semantic segmentation. Later, Qi et al. presented an extension to PointNet with PointNet++ [33], which used furthest point sampling and multi-scale grouping to push the state of the art. Recently, VoxelNet [48] remedied an intrinsic problem of point clouds, a lack of structure. Voxelizing point clouds allowed conventional image-based convolutional neural networks (CNN) to be employed for object detection using point clouds. Furthermore, Weinmann et al. [42], Liu et al. [23], and Dubé et al. [8] demonstrated that point clouds could be represented by neighborhood structural statistics in order to improve performance on scene understanding and place recognition tasks.

Data Fusion Liang et al. [21] designed a method for multi-scale fusion of ground imagery with overhead LiDAR point clouds to perform object detection from multiple viewpoints and modalities. Similar to our own work, Jaritz et al. [15] used cross-modal autonomous driving dataset to perform unsupervised domain adaption for 3D semantic Segmentation. Their dataset combined real world terrestrial LiDAR point clouds and camera images for different times of day, countries, and sensor setups. The xMUDA cross-modal model performed data fusion by projecting 3D point cloud points onto the 2D image plane and sampling features at corresponding pixel locations. Jaritz et al’s dataset and method were designed for small spatial areas around a vehicle, while we perform data fusion of satellite imagery and airborne LiDAR point clouds of large $400 \times 400 m^2$ areas.

Vehicle Speed Estimation: Huang [14] used video surveillance data of traffic to perform individual vehicle speed estimation. We perform average free-flow speed estimation to form a macroscopic understanding of the traffic flow behavior and capacity of roads instead of individual vehicle speed characteristics. Most similar to our own work, Song et al. [38] performed free-flow speed estimation using satellite imagery and human annotated area type road features on the Kentucky free-flow speed dataset. Our RasterNet model is trained on the same satellite imagery and label data, but our approach replaces the provided highway geometric features with spatially consistent point cloud features.

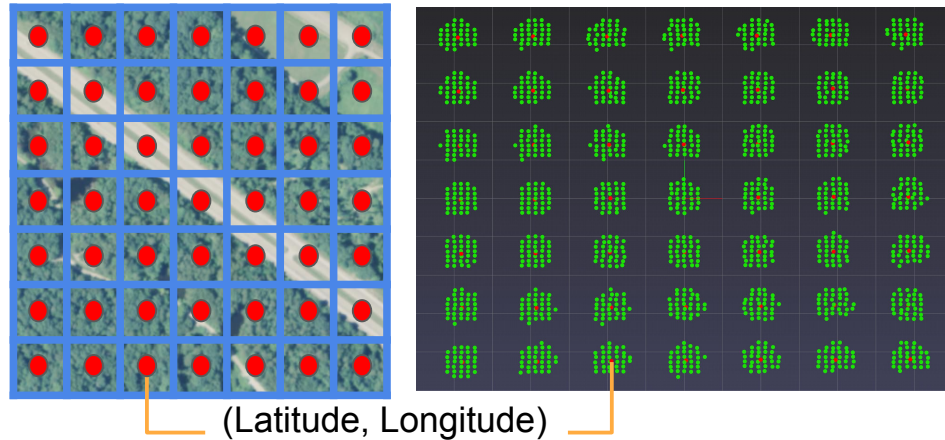


Figure 3.1: Image features are paired with point cloud features using a grid of raster center points (red dots), ensuring geo-spatial consistency between the two feature sets.

3.3 Dataset

The free-flow speed estimation dataset used to train each of the deep learning models contained a combination of the overhead image dataset developed by Song et al. [38] and the point cloud dataset described in Chapter 2. The overhead satellite imagery portion of the dataset was sourced from the National Agriculture Imagery Program (NAIP). Each satellite image covered a $400 \times 400 m^2$ area and was cropped to 224×224 . The geo-spatial coordinates corresponding to the free-flow speed labels (provided by Kentucky Transportation Center) were used to pair georeferenced point clouds and satellite images. The point cloud dataset also paired each point cloud with a 7×7 raster center grid. Each raster center grid (red dots) corresponded to a point cloud and was spatially consistent with the area depicted in paired imagery, as shown on the left in Figure 3.1. Further detail regarding the spatial correspondence between the overhead imagery, raster center grid, and point clouds is discussed in section 3.4.

3.4 Methods

Using the overhead image dataset created by Song et al. [38] and our own LiDAR point cloud dataset, we trained our models using both imagery and point clouds. We used an off-the-shelf CNN for image feature extraction. However, instead of using Xception [4] (23M parameters) as an image feature extractor like Song et al. [38], we chose the popular transfer learning network ResNet18 [11] (12M parameters) due to its impressive performance on the ImageNet [6] classification task using a limited parameter count. Combining the effi-

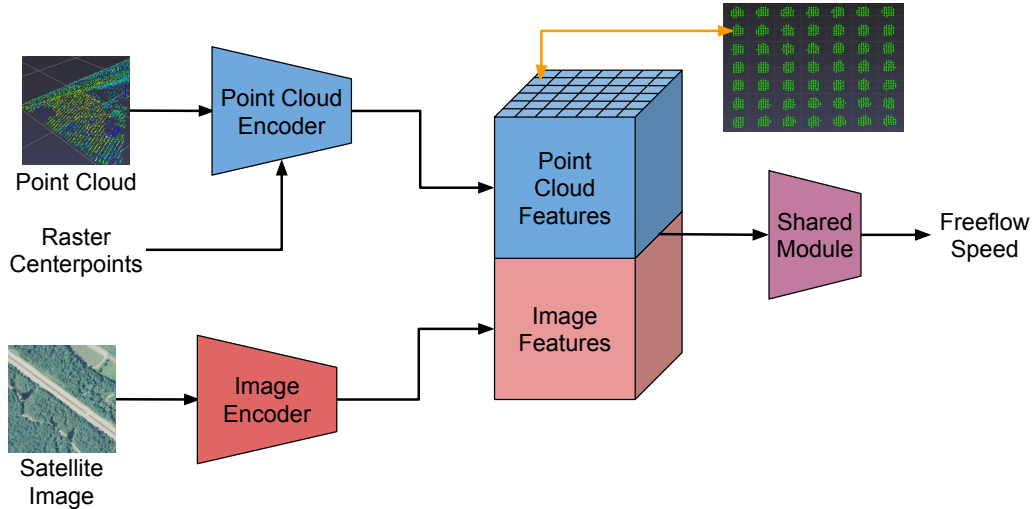


Figure 3.2: RasterNet general architecture diagram. Satellite images pass through a ResNet-based image encoder, while point clouds and raster center locations are passed through a point cloud encoder. Each cell of the point cloud feature map corresponds to a set of features of a local point cloud neighborhood. The two sets of features are then channel-wise concatenated before being passed through a shared model (ResNet block) to produce a free-flow speed prediction.

cient ResNet image feature extractor with a point cloud feature extractor, we formed a new network we call RasterNet, as depicted in Figure 3.2. The image encoder was built from the first 6 ResNet blocks and part of the 7th block (the residual sub-block). The image encoder was frozen and separated from the remainder of ResNet. This partitioning was done halfway through the 7th block because the downsample convolutions result in a $B \times C \times 7 \times 7$ set of features. The partitioning scheme ensured that the RasterNet network did not allow residual connections to ignore point cloud features. The remainder of ResNet was left unfrozen and was called the Shared Module, given that both the image features and the point cloud features would pass through said module. We developed three variations of RasterNet, each with a different point cloud encoder. Each variation was designed to explore alternative methods of extracting point cloud features. The three variations of RasterNet included: (1) RasterNet Simple, (2) RasterNet Statistics, and (3) RasterNet Learn. All three variations of RasterNet produced point cloud features that were spatially consistent with the image features, using the raster center points.

Raster center spatial consistency between imagery and point clouds was maintained using geo-spatial coordinates. The raster center grid ensured that image and point cloud feature correspondence was maintained. As images passed through a feature extractor a $B \times C \times 7 \times 7$ set of features was produced; where B and C are respective the batch and channel dimensions. Similarly, point cloud features were grouped around the raster center

Table 3.1: Simple Structural Features

Structural Feature	Equation
<i>Eigenvalues</i>	$\lambda_1^x, \lambda_2^x, \lambda_3^x$
<i>Means</i>	$\mu_{x_{i,1}}, \mu_{x_{i,2}}, \mu_{x_{i,3}}$
<i>Variances</i>	$\sigma_{x_{i,1}}^2, \sigma_{x_{i,2}}^2, \sigma_{x_{i,3}}^2$

points such that the point cloud feature extractor returned a $B \times C \times 7 \times 7$ set of features. The spatial correspondence established by the raster center grid between the overhead imagery and point clouds, ensured these two sets of high level features were spatially correlated. As a result, the image features and point cloud features could be concatenated channel-wise and passed through a shared module to produce a free-flow speed prediction.

3.4.1 Point Cloud Learned Encoder

The RasterNet Learn model used a modified PointNet++ [33] architecture as a learned point cloud feature extractor. PointNet++ was selected as a point cloud feature extractor because of its 90.7% accuracy on ModelNet40 with only 1.7×10^6 parameters, according to Prokudin et al. [31]. The publicly available PyTorch [30] implementation of PointNet++ from Wijmans [43] was modified so the second multi-scale grouping layer performed grouping around the raster centers of a given point cloud instead of using furthest point sampling. This modification ensured that the point cloud features, corresponding to each raster center that were extracted, could be channel-wise concatenated with image features while maintaining spatial consistency. After the second multi-scale grouping layer the remainder of Pointnet++ was replaced with a set of downsampling layers which reduced the number of collected features per raster center down to 16. The resulting point cloud features were the concatenated with the image features before being passed to the shared module for free-flow speed prediction, as shown in Figure 3.2.

3.4.2 Point Cloud Structural Feature Encoder

Inspired by Liu et al. [23] and work on place recognition using LiDAR point cloud structural features, we developed the RasterNet Statistics model that extracted point cloud structural statistic features instead of using a fully learned point cloud feature extractor (such

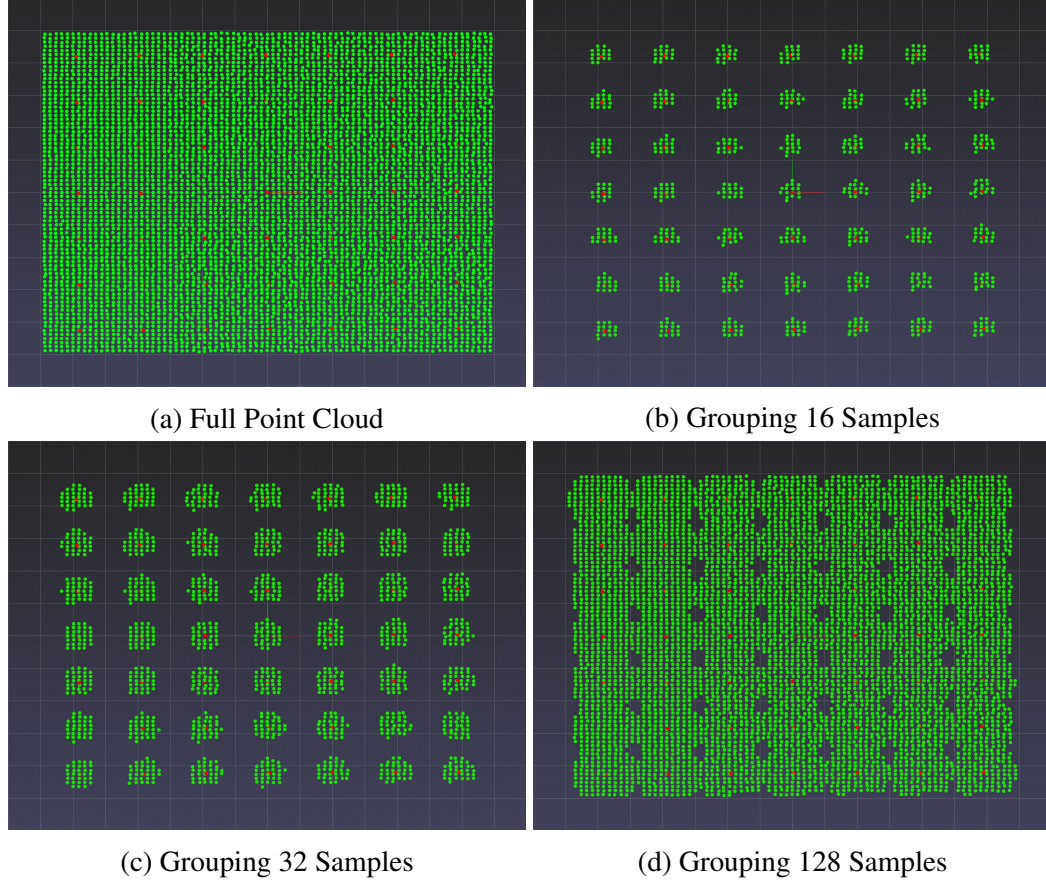


Figure 3.3: PointNet++ [33] style multi-scale grouping depicted for a point cloud (a) centered on a known free-flow speed label location. Grouping operations are performed around each of the 7×7 raster centers (red) at different scales and number of samples. Local point clouds (green) are grouped at varying sample sizes: 16 samples (b), 32 samples (c), and 128 samples (d).

as RasterNet Learn). The RasterNet Statistics model replaced the PointNet++ architecture from RasterNet Learn model with a single instance of multi-scale grouping, as depicted in Figure 3.3. Local point cloud neighborhood statistics were extracted by first calculating the covariance matrix of the 3D point cloud. Let $x = x_{i,j,k}$ be a three dimensional point cloud from the flattened raster ($n = 49$) of n point clouds, where $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, 3\}$, and $k \in \{1, 2, \dots, m\}$ for m points. The covariance matrix was assumed to be positive definite and symmetric, such that the corresponding eigenvalues $\lambda_1^x, \lambda_2^x, \lambda_3^x$ of the covariance matrix are all greater than zero. The local point cloud structural statistics were calculated according to the notation from by Weinmann et al.'s [42] and Liu et al [23], as shown in Table 3.2. Note, the 2D Scattering and 2D Linearity statistics required 2D xy-plane eigenvalues. 2D xy-plane eigenvalues were calculated by projecting the 3D point cloud to the 2D xy-plane and then calculating the two eigenvalues of the 2D covariance

Table 3.2: Structural Statistics

Structural Statistic	Equation
<i>Change of Curvature</i>	$C_{x_i} = \frac{\lambda_3^{x_i}}{\lambda_1^{x_i} + \lambda_2^{x_i} + \lambda_3^{x_i}}$
<i>Omni-variance</i>	$O_{x_i} = \frac{\sqrt[3]{\lambda_3^{x_i}}}{\lambda_1^{x_i} + \lambda_2^{x_i} + \lambda_3^{x_i}}$
<i>Linearity</i>	$L_{x_i} = \frac{\lambda_1^{x_i} - \lambda_2^{x_i}}{\lambda_1^{x_i}}$
<i>Eigenentropy</i>	$A_{x_i} = - \sum_{j=1}^3 \lambda_j^{x_i} \ln \lambda_j^{x_i}$
<i>Local Point Density</i>	$D_{x_i} = \frac{k_{x_i}}{\frac{4}{3} \prod_{j=1}^3 \lambda_j^{x_i}}$
<i>2D Scattering</i>	$S_{2D,x_i} = \lambda_{2D,1}^{x_i} + \lambda_{2D,2}^{x_i}$
<i>2D Linearity</i>	$L_{2D,x_i} = \frac{\lambda_{2D,2}^{x_i}}{\lambda_{2D,1}^{x_i}}$
<i>Verticality</i>	$V_{x_i} = v_{3,3}^{x_i}$
<i>Max Height Difference</i>	$\Delta Z_{x_i} = \max(x_{i,3}) - \min(x_{i,3})$
<i>Height Variance</i>	$\sigma_{x_i,3}^2 = \frac{\sum_{k=1}^m x_{i,3,k} - \mu_{x_i,3}}{m}$

matrix. Verticality is defined as the vertical component of the normal vector, which is the 3rd dimension of the covariance matrix’s 3rd eigenvector. Additionally, the mean of local point cloud x in the height dimension (3) was expressed as $\mu_{x_i,3}$.

Local neighborhood structural statistic features were then extracted from each of the three sets of 7x7 local point clouds. The three sets of 7x7 structural features were concatenated and passed through a downsampling convolution to produce a single set of 10 features corresponding to each of the 7x7 grid of point clouds. Last, the set of 10 structural features were then channel-wise concatenated with the 7x7 image features.

The third variant of RasterNet called RasterNet Simple was designed using nine simple local point cloud structural features from Table 3.1 instead of ten complex statistics inspired by Liu et al’s [23]. Similar to the RasterNet Statistics model, RasterNet Simple employed multi-scale grouping to extract three groups of point clouds at different scales. The simple structural features extracted from each of the three groups was composed of: (1) the mean of each of the three spatial dimensions, (2) the variance of each of the three dimensions, and (3) the three eigenvalues of the 3D point cloud’s covariance matrix. The three groups of

7x7 structural features were concatenated then passed through a downsampling convolution to simplify to a single 7x7 set of 10 structural features. Last, the 7x7 set of features was channel-wise concatenated with the 7x7 image features.

3.4.3 Loss Function

Similar to Song et al. [38] we formatted the free-flow speed prediction as a multi-class classification problem. Free-flow speeds were binned into 1mph bins, totaling 79 possible classes. We also experimented with a second binning strategy, where each speed was binned into 16 classes with bin increments of 5mph. Since free-flow speed prediction accuracy is measured within-5mph, binning in 5mph increments seemed like a natural binning strategy. We decided to work with cross-entropy loss due to its strong performance after experimenting with multiple loss functions: Huber regression loss, combined Huber and cross-entropy losses inspired by Workman et al. [44], Gaussian filtered cross-entropy loss, and weighted cross-entropy loss. Cross-entropy loss (L) with a Softmax activation was formulated as follows,

$$L(Y, \hat{Y}) = -\frac{1}{N} \sum_{i=1}^N \log \left(\frac{e^{y_i}}{\sum_j^C e^{\hat{y}_{i,j}}} \right). \quad (3.1)$$

Let $y_i \in Y$ be a positive class bin label for the i th sample from N training samples. The predicted probability from the distribution \hat{Y} for the i th sample from the j th class was expressed as $\hat{y}_{i,j}$. A given class $j \in \{0, 1, \dots, C\}$, where C was the maximum number of classes (either 16 or 79, depending on binning strategy). Some experimentation was done with class balanced weighting of cross-entropy, but it resulted in little improvement.

3.4.4 Implementation Details

Each network was trained with a learning rate of 1×10^{-6} that was scheduled for a reduction of 10 every 25 epochs. The Adam optimizer was used with weight decay of 0.1, since it resulted in the best performance. All RasterNet variant models also had an additional dropout layer before the last fully connected layer for added regularization. Early stopping was used for each model, where a model would halt training if the validation loss had not decreased for at least 3 epochs. Otherwise, each model was trained and validated for 100 epochs. Each model was evaluated on the set of weights corresponding to the smallest validation loss.

Table 3.3: Free-flow Speed Estimation Model Performances

Method	Architecture	Bin Width	Accuracy
<i>Song et al. Image Only [38]</i>	Xception	1	37.60%
<i>Song et al. Image + Road Features [38]</i>	Xception \oplus F	1	49.86%
<i>Image Only</i>	ResNet	1	42.01%
<i>Point Cloud Only</i>	Reduced PointNet++	1	34.08%
<i>Image + Point Cloud</i>	RasterNet Learn	1	49.38%
<i>Image + Simple Structural Features</i>	RasterNet Simple	1	50.28%
<i>Image + Structural Statistics</i>	RasterNet Statistics	1	47.75%
<i>Image + Point Cloud</i>	RasterNet Learn	1	50.47%
<i>Image + Point Cloud</i>	RasterNet Learn	5	46.46%

3.5 Evaluation

Following Song et al.’s [38] convention, free-flow speed estimation was evaluated using within-5mph accuracy of the predicted free-flow speed when compared with the known speed. While all models predicted speed in terms of bins (1mph or 5 mph), their accuracy was evaluated with respect to the raw average free-flow speeds.

3.5.1 Quantitative Evaluation

We started by evaluating the performance of a full ResNet model trained only on satellite imagery for the task of free-flow speed estimation in order to highlight the differences in image feature extractors between Song et al.’s Xception-based architecture and our choice of a ResNet architecture. The first 6 blocks and the residual sub-block of the 7th layer were frozen for all evaluated ResNet feature extractors. Additionally, we evaluated a standalone full PointNet++ model trained only using point cloud features to predict free-flow speed. Similar to the truncated RasterNet Learn feature extractor, the second multi-scale grouping layer was modified to extract features at raster center locations. The remainder of the PointNet++ architecture was left intact, with some modifications. The number of fully connected layers in the last multi-layer perceptron (MLP) (after the multi-scale grouping layer) was reduced to two layers in order reduce the number of trainable parameters and training time.

A summary of the average within-5mph test accuracy of each RasterNet variation, PointNet++, and ResNet are shown in Table 3.3. Interestingly, ResNet trained only with image features out performed the Song et al. Xception-based network. The reduced PointNet++ with raster center locations had the worst performance, likely due to the more de-

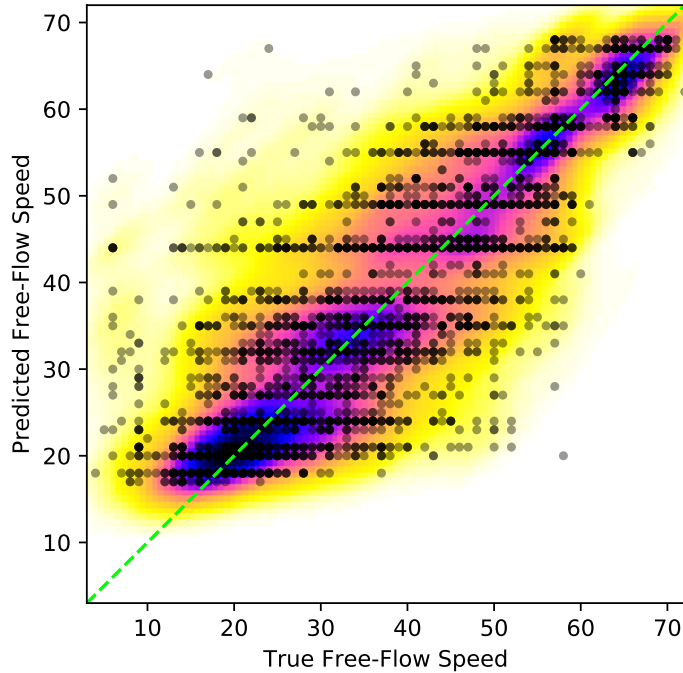
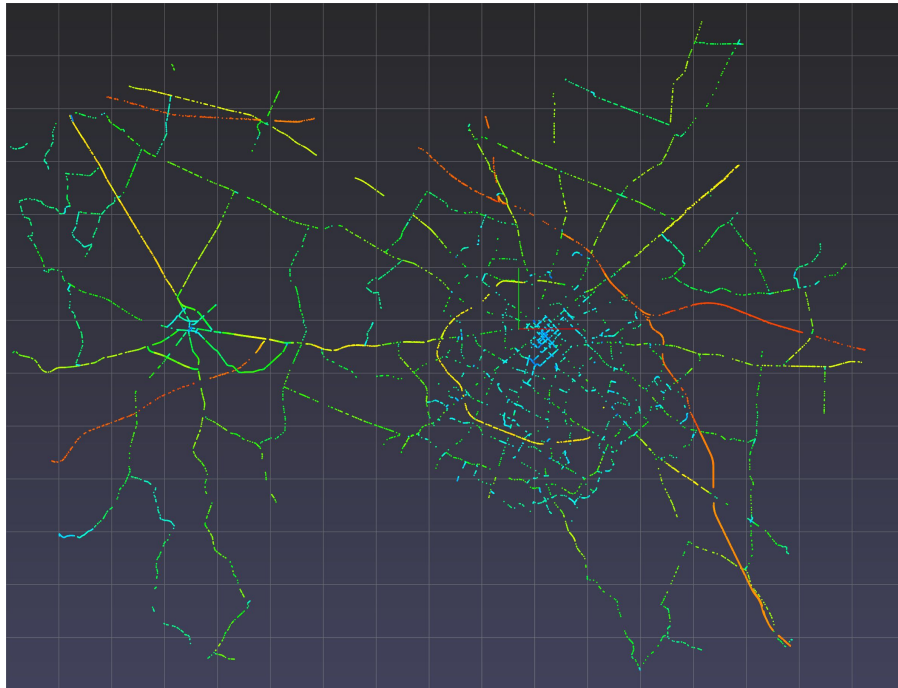


Figure 3.4: Scatterplot of free-flow speed predictions on the test set from the RasterNet Learn model compared with known speed labels. Overlaid heatmap depicts higher point density in darker color. Optimal performance should follow the green line.

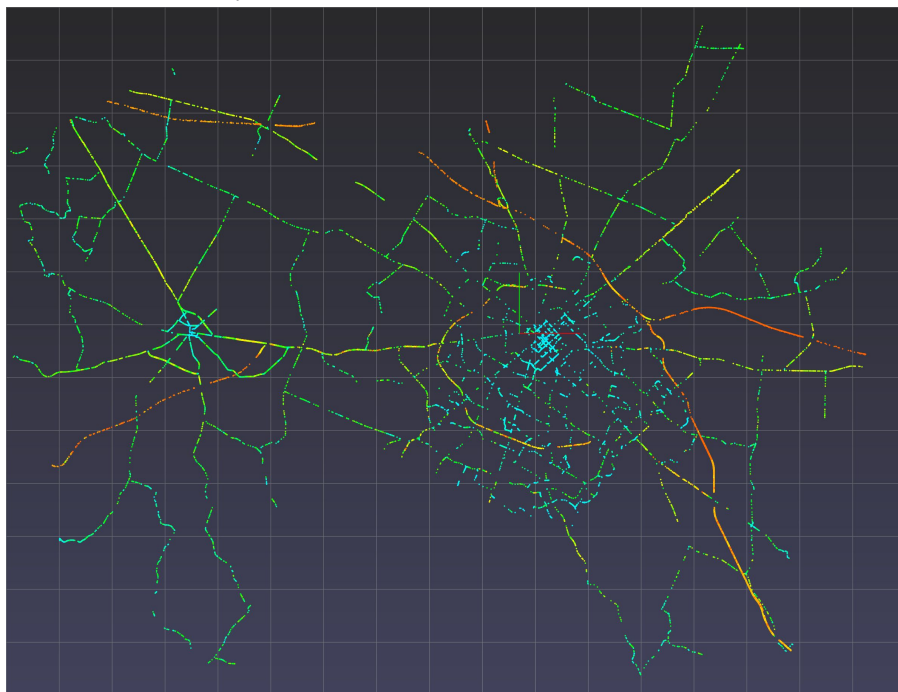
scriptive nature of image features. The RasterNet Learn model (image and point cloud features) was able to achieve similar performance to Song et al.’s image and road feature model (Xception concatenated with road features), despite not using provided highway geometric features. The RasterNet Learn variant using learned point cloud features with a bin size of 1, achieved the best performance and used fewer parameters than either of the other two RasterNet variants. Using the best performing model to predict within-5 accuracy with bin widths of 5, instead of 1, decreased performance. This binning behavior is likely attributed to the network being trained to have a more shallow understanding of the relationship between features due to less severe punishment for incorrect predictions.

3.5.2 Qualitative Evaluation

A qualitative evaluation was performed on the RasterNet Learn model on the test set, since it had the best quantitative performance. Examining Figure 3.4 the model’s prediction distribution followed the ground truth label distribution on the test set, as shown by the highest density of points (the darker colors) being on the green line. The prediction distri-



(a) Fayette and Woodford Counties Ground Truth



(b) Fayette and Woodford Counties Predicted

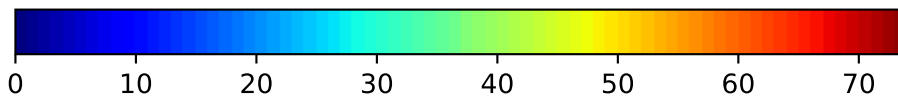


Figure 3.5: Ground truth and predicted speed maps for both Woodford (left small city) and Fayette (right larger city) counties in Kentucky, USA.

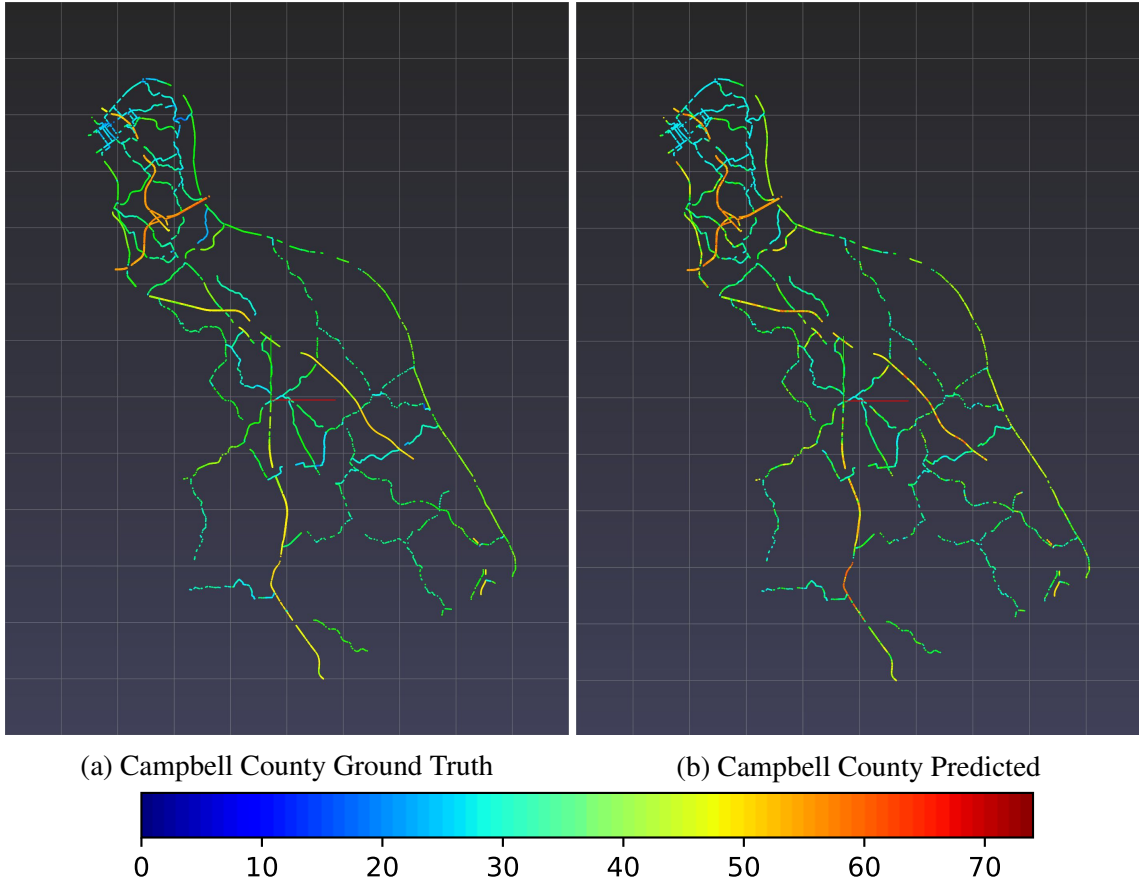
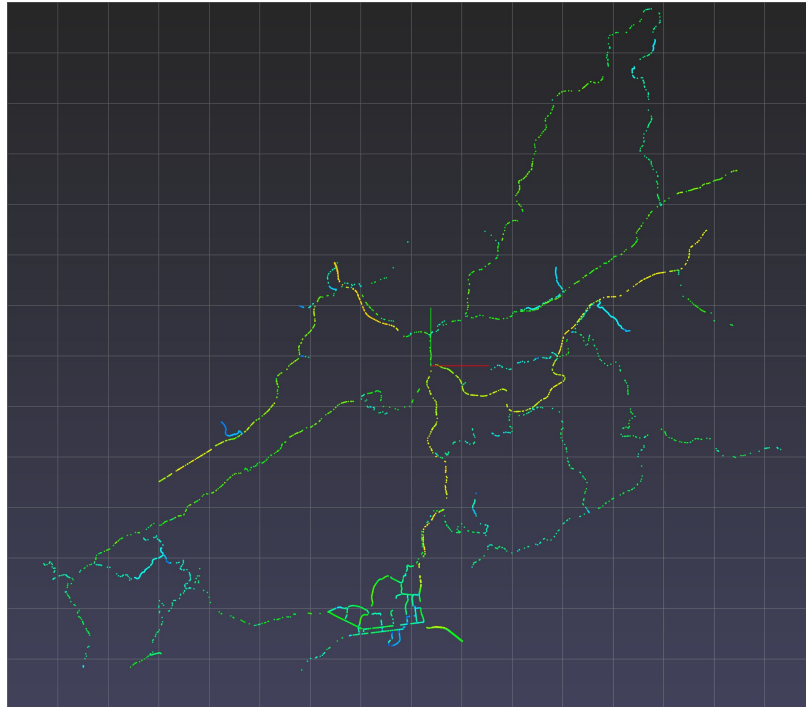


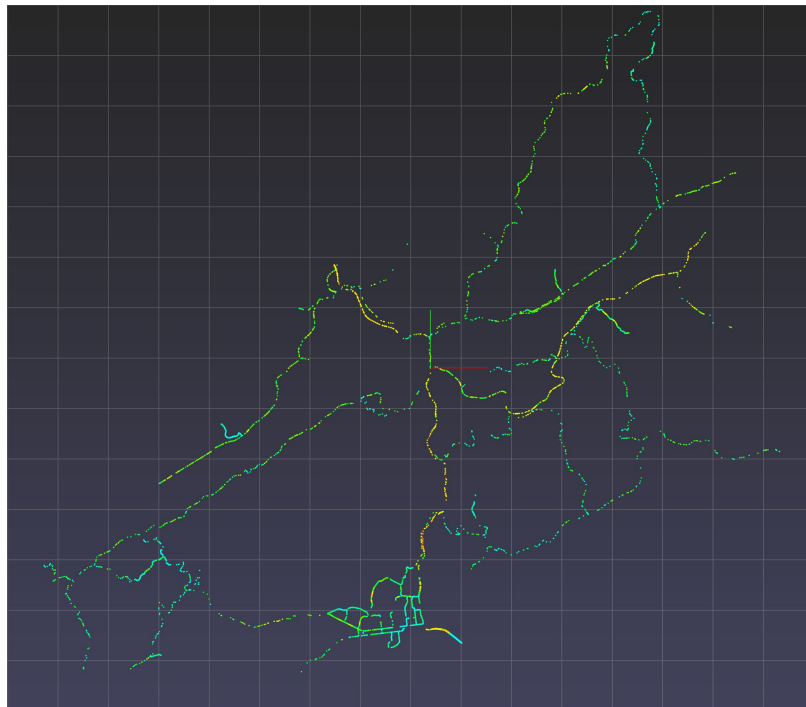
Figure 3.6: Ground truth and predicted speed maps for Campbell county Kentucky, USA.

bution also suggest that the model had difficulties predicting slow free-flow speeds, since it made no predictions below 17mph. Additionally, the scatter plot suggests that the model had difficulty distinguishing speeds between 40mph and 50mph, since it often predicted 45mph.

Evaluating the geographic relevance of the RasterNet Learn model’s test predictions was done by constructing free-flow speed maps. Free-flow speed maps visualize free-flow speed as it relates to geographic location. We generated free-flow speed maps with the ground truth and predicted free-flow speeds for 5 Kentucky counties from the test set: Fayette, Woodford, Campbell, Bell, and Union. Since Fayette and Woodford counties are closely connected, they were plotted on the same free-flow speed maps in Figure 3.5. Figure 3.5 (b) suggests that the model is capable of estimating free-flow speeds on highways accurately, as shown by interstate-75 and interstate-65 both being red in both maps. However, Figures 3.5 and 3.6 suggest that the model does not perform well at predicting slow speed areas since it rarely predicts speeds < 20 mph, in particular in urban centers such as the city center of Lexington Figure 3.5 (b). The model performs well in rural counties,



(a) Bell County Ground Truth



(b) Bell County Predicted

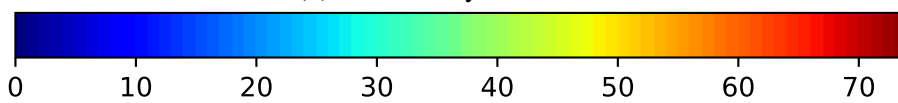
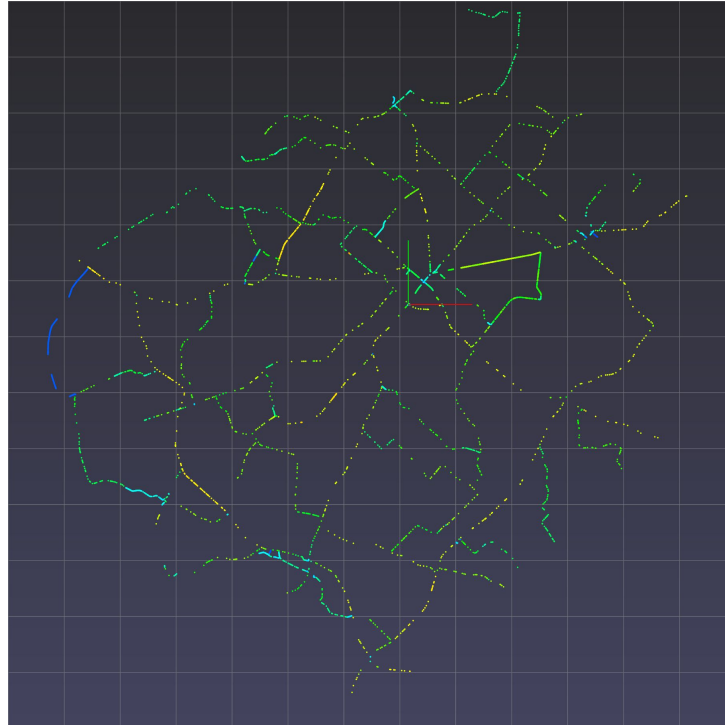
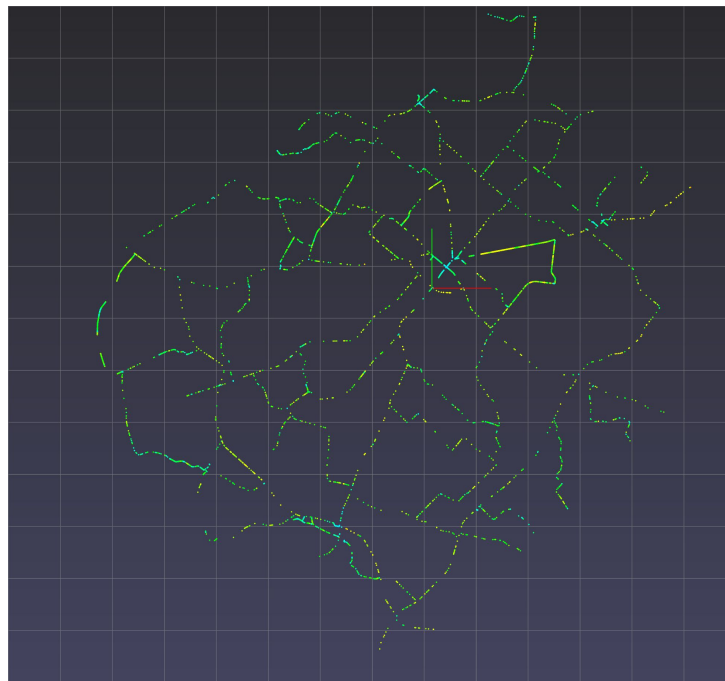


Figure 3.7: Ground truth and predicted speed maps for Bell county Kentucky, USA.



(a) Union County Ground Truth



(b) Union County Predicted

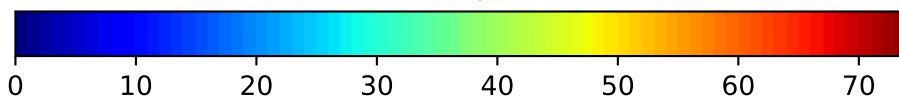


Figure 3.8: Ground truth and predicted speed maps for Union county Kentucky, USA.

such as Bell county in Figure 3.7, with speeds primarily ranging from 30-50mph. Note in Figure 3.8 (a), the road segment on the far left is dark blue, indicated free-flow speeds $< 20\text{mph}$. The predicted free-flow speed map in Figure 3.8 (b) suggests that the model predicts speeds $> 20\text{mph}$ for said road segment. The road segment in question is a dirt road, which suggests that the model has difficulty correctly evaluating low speed dirt roads.

3.6 Conclusion

We presented a new architecture, RasterNet, designed to combine satellite image features with LiDAR point cloud features using a structured raster for the task of free-flow speed estimation. Our RasterNet Learn variant using learned point cloud and overhead imagery achieved state of the art results of 50.47% within-5mph accuracy on the challenging Kentucky free-flow speed estimation task. Our comparative analysis reflected that point cloud structural features could be used to replace and outperform existing highway geometric features, allowing our model to be used on roads where these highway geometric features are not present. We also introduced a raster data fusion method for pairing overhead imagery with point clouds that both corresponded with free-flow speed labels. Our model was capable of producing large-scale free-flow speed maps across Kentucky, a potentially useful tool for transportation engineering and roadway planning. Through qualitative evaluation we showed that our RasterNet Learn model could predict free-flow speed on various road types and speeds.

Chapter 4

Conclusion

4.1 Contributions

This thesis presented a new Kentucky LiDAR point cloud dataset for the challenging task of free-flow speed estimation. The dataset included roads composed of various types, topographies, functional classes, elevations, and environment types. Point clouds provided a roadway scene representation that allowed geometric features to be extracted from the spatial correspondence of points. While highway geometric features are not always available for all roads in Kentucky, airborne LiDAR point clouds are abundant and cover roads across the entire state.

We also introduced a data fusion method for combining point cloud and image features using a raster structure. Pairing our point cloud dataset with the satellite imagery dataset from Song et al. [38] enabled our models to train on image features, point cloud features, or both sets of features. The raster center grid allowed our multi-modal approaches to geospatially align local point cloud neighborhood features with corresponding overhead image features.

We presented a new series of deep neural network architectures (RasterNets) for free-flow speed estimation using both overhead image features and point cloud features. RasterNet Learn, a RasterNet variant, used learned point cloud features to achieve state of the art performance on the Kentucky free-flow speed prediction dataset. Our model illustrated that provided geometric highway features could be replaced with high level point cloud structural features to improve estimation performance and coverage of state roads. We illustrated the capability of using our model to automatically generate large-scale free-flow speed maps on multiple counties in Kentucky, which could be expanded to include the entire state of Kentucky.

4.2 Future Work

While Kentucky includes an assortment of road types and environments, we would be interested in expanding the real world use cases of our dataset by including landscapes with more varied geographies and road types (ex. large bridges and multi-layered overpasses). Additionally, we would like to expand the existing dataset to include more instances of road types with limited number of examples, such as urban and unpaved dirt roads. Including more examples of rare road types could improve the predictive capability of the RasterNet models.

We would also be interested in exploring extensions of the raster data fusion method by including street-level LiDAR point clouds and panoramas. Improving the resiliency and capability of data-fusion approaches on more modalities further increase the RasterNet models performance. Additionally, we are interested in exploring alternative methods for data fusion, such as combined latent information representations using variational autoencoders. Latent data representations composed of features from a visual and spatial modality could be useful for encouraging alignment between point cloud and image features.

He et al. [12] have shown the potential of combined graph and convolutional neural networks for road understanding tasks (ex. lane and road type prediction). Graph-based approaches have the advantage of sequentially reasoning for multiple road segments. We would like to explore using graph neural networks for label smoothing with respect to a group of spatially neighboring predictions, which could improve estimation performance and consistency.

Bibliography

- [1] Federal Highway Administration. Manual on uniform traffic control devices. Technical report, US Department of Transportation, 2009. 1
- [2] Federal Highway Administration. Highway policy information. Technical report, US Department of Transportation, 2014. 1
- [3] Kentucky Transportation Cabinet. 2014-2035 kentucky long-range statewide transportation plan. Technical report, Kentucky Transportation Cabinet, 2014. 1, 2
- [4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 22
- [5] Matthew D Deardoff, Brady N Wiesner, and Joseph Fazio. Estimating free-flow speed from posted speed limit signs. *Procedia-social and behavioral sciences*, 16:306–316, 2011. 3
- [6] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 22
- [7] Annette Dietmaier, Gregory J McDermid, Mir Mustafizur Rahman, Julia Linke, and Ralf Ludwig. Comparison of lidar and digital aerial photogrammetry for characterizing canopy openings in the boreal forest of northern alberta. *Remote Sensing*, 11(16):1919, 2019. 2
- [8] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. Segmatch: Segment based place recognition in 3d point clouds. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5266–5272. IEEE, 2017. 21
- [9] Melissa Fedrigo, Stephen B Stewart, Stephen H Roxburgh, Sabine Kasel, Lauren T Bennett, Helen Vickers, and Craig R Nitschke. Predictive ecosystem mapping of south-

- eastern australian temperate forests using lidar-derived structural profiles and species distribution models. *Remote Sensing*, 11(1):93, 2019. 2
- [10] Thomas F Golob, Wilfred W Recker, and Veronica M Alvarez. Freeway safety as a function of traffic flow. *Accident Analysis & Prevention*, 36(6):933–946, 2004. 2
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 22
- [12] Songtao He, Favyen Bastani, Satvat Jagwani, Edward Park, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Samuel Madden, and Mohammad Amin Sadeghi. Roadtagger: Robust road attribute inference with graph neural networks. *arXiv preprint arXiv:1912.12408*, 2019. 36
- [13] Karl Heidemann, Hans. Lidar base specification. Technical report, U.S. Geological Survey, 2012. 9
- [14] Tingting Huang. Traffic speed estimation from surveillance video data: For the 2nd nvidia ai city challenge track 1. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 161–1614. IEEE, 2018. 21
- [15] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Émilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation, 2019. 21
- [16] Jake Kononov, Catherine Durso, David Reeves, and Bryan Allery. Relationship between traffic density, speed, and safety and its implications for setting variable speed limits on freeways. *Transportation Research Record: Journal of the Transportation Research Board*, 2280:1–9, 12 2012. 2
- [17] Mei-Po Kwan and Daniel M Ransberger. Lidar assisted emergency response: Detection of transport network obstructions caused by major disasters. *Computers, Environment and Urban Systems*, 34(3):179–188, 2010. 6
- [18] Kentucky Division of Geographic Information KyFromAbove. <http://kyfromabove-kygeonet.opendata.arcgis.com/>. vi, 7, 9
- [19] Kentucky Division of Geographic Information KyFromAbove. http://kygeonet.ky.gov/kyfromabove/pdfs/Specs_LiDAR_Production.pdf. 9

- [20] DF Laefer, S Abuwarda, AV Vo, L Truong-Hong, and H Gharibi. Aerial laser and photogrammetry survey of dublin city collection record. *New York University*. DOI, 10:N8MQ0N, 2015. 2
- [21] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 21
- [22] Yansong Liu, Sildomar T Monteiro, and Eli Saber. Vehicle detection from aerial color imagery and airborne lidar data. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1384–1387. IEEE, 2016. 6
- [23] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Yingtian Liu, Peng Yin, Hesheng Wang, and Yunhui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *ICCV*, 2019. 2, 21, 24, 25, 26
- [24] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-net: Towards learning based lidar localization for autonomous driving. In *CVPR*, 2019. 2
- [25] Highway Capacity Manual. Hcm2010. *Transportation Research Board, National Research Council, Washington, DC*, page 1207, 2010. 1, 3, 5
- [26] Richard Margiotta and Scott Washburn. Simplified highway capacity calculation method for the highway performance monitoring system. Technical report, Federal Highway Administration, Office of Policy and Governmental Affairs, 2017. 1
- [27] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, et al. The 2018 nvidia ai city challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 53–60, 2018. 6
- [28] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng Hwang, et al. The 2019 ai city challenge. In *CVPR Workshops*, 2019. 6
- [29] Chen Feng Ng and Kenneth A Small. Tradeoffs among free-flow speed, capacity, cost, and environmental footprint in highway design. *Transportation*, 39(6):1259–1280, 2012. 1

- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019. 24
- [31] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 24
- [32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 15, 21
- [33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. vii, 15, 21, 24, 25
- [34] Yonghua Qu, Ahmed Shaker, Lauri Korhonen, Carlos Alberto Silva, Kun Jia, Luo Tian, and Jinling Song. Direct estimation of forest leaf area index based on spectrally corrected airborne lidar pulse penetration ratio. *Remote Sensing*, 12(2):217, 2020. 2
- [35] Ch Ravi Sekhar, J Nataraju, S Velmurugan, Pradeep Kumar, and K Sitaramanjaneyulu. Free flow speed analysis of two lane inter urban highways. *Transportation research procedia*, 17:664–673, 2016. 3, 5
- [36] Ary P Silvano and Karl L Bang. Impact of speed limits and road characteristics on free-flow speed in urban areas. *Journal of transportation engineering*, 142(2):04015039, 2016. 1, 3
- [37] Stephen Smith, Gregory Barlow, Xiao-Feng Xie, and Zack Rubinstein. Surtrac: Scalable urban traffic control. In *Proceedings of Transportation Research Board 92nd Annual Meeting Compendium of Papers*. Transportation Research Board, January 2013. 2
- [38] Weilian Song, Tawfiq Salem, Hunter Blanton, and Nathan Jacobs. Remote estimation of free-flow speeds. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 791–794. IEEE, 2019. 3, 6, 8, 9, 19, 20, 21, 22, 27, 28, 35

- [39] Weilian Song, Scott Workman, Armin Hadzic, Xu Zhang, Eric Green, Mei Chen, Reginald Souleyrette, and Nathan Jacobs. Farsa: Fully automated roadway safety assessment. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 521–529. IEEE, 2018. 5
- [40] Yuan Sun, Hao Xu, Jianqing Wu, Jianying Zheng, and Kurt M Dietrich. 3-d data processing to extract vehicle trajectories from roadside lidar data. *Transportation research record*, 2672(45):14–22, 2018. 6
- [41] Yi Tian. *Self-Powered Intelligent Traffic Monitoring Using IR Lidar and Camera*. PhD thesis, Virginia Tech, 2017. 6
- [42] Martin Weinmann, Boris Jutzi, and Clément Mallet. Semantic 3d scene interpretation: A framework combining optimal neighborhood size selection with relevant features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):181, 2014. 21, 25
- [43] Erik Wijmans. Pointnet++ pytorch. https://github.com/erikwijmans/Pointnet2_PyTorch, 2018. 24
- [44] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. In *British Machine Vision Conference*, 2016. 27
- [45] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 15
- [46] Wei Yao, Stefan Hinz, and Uwe Stilla. Automatic vehicle extraction from airborne lidar data of urban areas aided by geodesic morphology. *Pattern Recognition Letters*, 31(10):1100–1108, 2010. 6
- [47] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2):401–416, 2017. 2
- [48] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 21

Belcan Engineering Group Inc.
2016–2017

Embedded Software Engineer
Lexington, KY

Tempur Sealy International Inc.
2013–2014

Software Engineering Co-op
Lexington, KY

Johnson Controls Inc.
2012

Software Engineering Intern
Lexington, KY

Publications

- [1] Weilian Song, Scott Workman, Armin Hadzic, Xu Zhang, Eric Green, Mei Chen, Reginald Souleyrette, and Nathan Jacobs. Farsa: Fully automated roadway safety assessment. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2018.