2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Sensor Fusion for Semantic Segmentation of Urban Scenes

Richard Zhang[1] Stefan A. Candra[2] Kai Vetter[3] Avideh Zakhor[1]

Abstract— **Semantic understanding of environments is an important problem in robotics in general and intelligent autonomous systems in particular. In this paper, we propose a semantic segmentation algorithm which effectively fuses information from images and 3D point clouds. The proposed method incorporates information from multiple scales in an intuitive and effective manner. A late-fusion architecture is proposed to maximally leverage the training data in each modality. Finally, a pairwise Conditional Random Field (CRF) is used as a post-processing step to enforce spatial consistency in the structured prediction. The proposed algorithm is evaluated on the publicly available KITTI dataset [1] [2], augmented with additional pixel and point-wise semantic labels for *building, sky, road, vegetation, sidewalk, car, pedestrian, cyclist, sign/pole*, and *fence* regions. A per-pixel accuracy of 89.3% and average class accuracy of 65.4% is achieved, well above current state-of-the-art [3].**

I. INTRODUCTION

We propose an algorithm for semantic parsing of the environment using information from multiple sensing modalities, namely images and 3D point clouds. Semantic segmentation involves labeling every pixel in an image, or point in a point cloud, with its corresponding semantic tag. A semantic understanding of the environment facilitates robotics tasks such as navigation, localization, and autonomous driving.

In unimodal semantic segmentation, most approaches for both images and point clouds follow a *bottom-up* approach in a CRF framework. Notable exceptions include methods proposed by Farabet *et al.* [4] and Girshick *et al.* [5], which learn expressive features from convolutional neural nets. CRFs are a graphical model framework used to balance local beliefs while enforcing spatial consistency. For images, Tighe *et al.* [6] [7], Singh *et al.* [8], and Yang *et al.* [9] use an image retrieval system followed by non-parametric classification to produce superpixel classifications, with special considerations taken for objects [7], semantic context [8], and rare classes [9]. For point clouds, the same bottom-up pipeline is typically taken [10] [11] [12] [13] [14], starting with an oversegmentation, followed by regional feature extraction, classification, and a CRF.

A challenge in the *bottom-up* approach is integrating and enforcing *top-down* information and structure. To this end, Ladický *et al.* [15] propose using Robust P^N clique potentials [16] to incorporate object detections. In [17], clique potentials are extended into a multi-level hierarchy. However, using clique potentials increases run-time, and solving for optimal parameters remains an open problem.

In an alternative approach proposed by Munoz *et al.* [18] for images and applied by Xiong *et al.* [19] for point clouds, an initial classifier is first trained on the coarsest level of a hierarchical segmentation. A classifier is then trained on the subsequent segmentation level, with previous classifier outputs incorporated as input features. This process is repeated, and information can be passed downwards or upwards multiple times through the hierarchy. The method is extended to integrate information across multiple modalities by Munoz *et al.* [20], where classifications are passed to the subsequent level of the hierarchy in *both* modalities.

Several previous studies in semantic segmentation of urban 3D data using multiple sensors have been recently performed [21] [22] [3]. Sengupta *et al.* [21] and He *et al.* [22] first perform semantic segmentation in the image modality, ignoring point cloud features. The image labeling results are projected into 3D using point clouds extracted from stereo vision, and results are aggregated in 3D using a voting scheme. Cadena and Košecká [3] propose a solution using a CRF framework, with an emphasis on fast run-time. The method oversegments the image and extracts simple, low-dimensional features. An early fusion architecture is used, where separate feature sets are extracted for each segment based on sensor coverage. A CRF framework is used to enforce spatial consistency. The method is designed to run quickly on a coarse label set, but struggles when applied to finer object classes such as *pedestrian* and *fence*.

In contrast to previous work [3] [20] [17], we explicitly incorporate information from multiple scales up front. We first segment the image and point cloud on multiple scales. Feature vectors of the low-level segments are augmented by features from corresponding higher-level segments. This differs from [3], which uses localized features computed on a single scale only. Our approach precludes the need to train an expanding number of classifiers [20] or rely on complex graphical model machinery [17], but allows us to effectively integrate information from multiple scales.

Unlike [3], which uses early fusion, we use a late fusion architecture. Specifically, separate classifiers in each modality are first deployed. Then, for regions with overlapping sensor coverage, a fusion classifier is trained to merge the soft classification outputs from the separate unimodal classifiers. Late fusion enables us to effectively leverage the data from the entire training set, rather than just the subset with overlapping sensor coverage. Finally, we use a pairwise CRF as a post-processing step to enforce spatial

[1]Department of Electrical Engineering and Computer Science, University of California, Berkeley. {rich.zhang,avz}@eecs.berkeley.edu
[2]Department of Computer Science, University of California, Berkeley candrastefan@berkeley.edu
[3]Department of Nuclear Engineering, University of California, Berkeley and the Nuclear Science Division, Lawrence Berkeley National Lab kvetter@lbl.gov

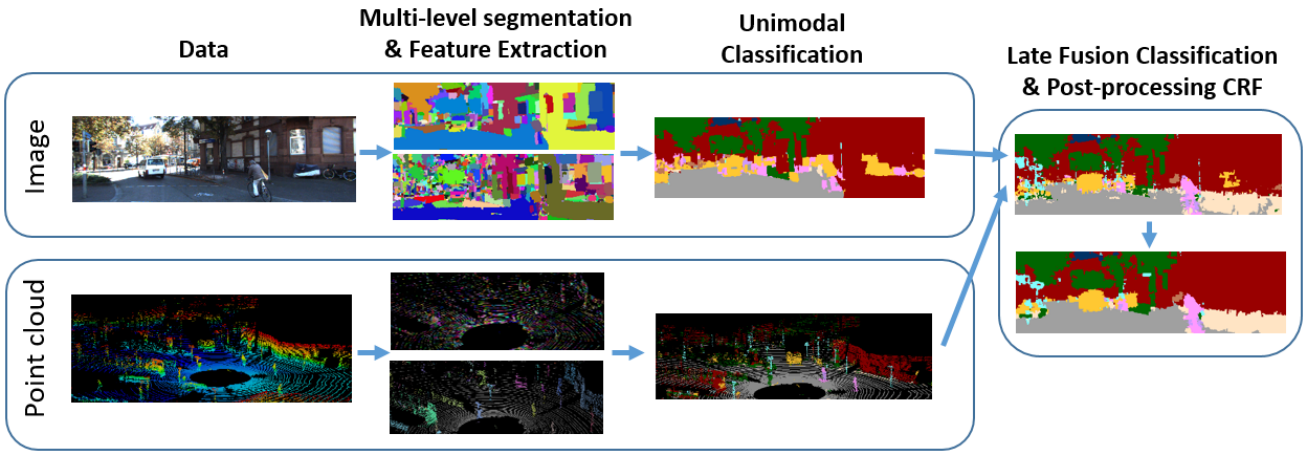
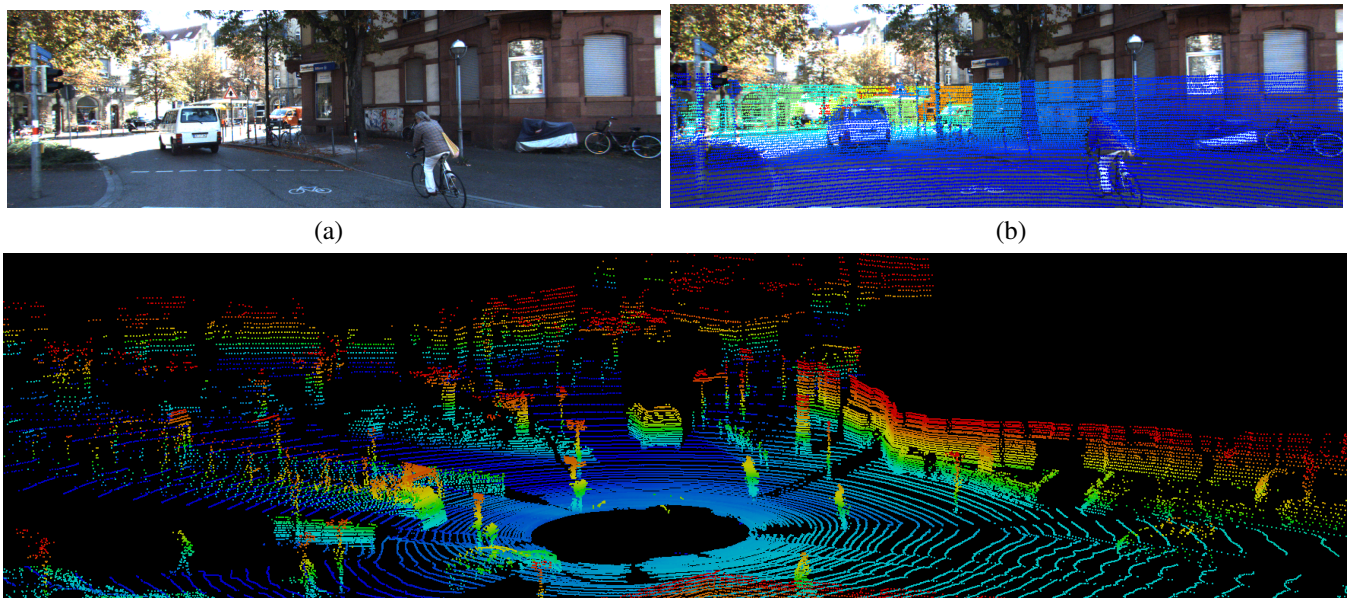


Fig. 1: Top-level pipeline.



(a) (b)

(c)

Fig. 2: Example (a) image; (b) point cloud projected onto image, colored by depth; (c) point cloud, colored by height.

consistency. We test our method on the publicly available KITTI platform [1] [2], which contains registered images and point clouds, and demonstrate significant improvement over the current state-of-the-art [3].

The outline of the paper is as follows. Section II provides an overview of the proposed algorithm. Section III details experimental results. We conclude and discuss future directions of study in Section IV.

II. ALGORITHM DESCRIPTION

A top-level pipeline of our algorithm is illustrated in Figure 1. The input to our algorithm is an individual acquisition comprised of a single image and its corresponding 3D point cloud, taken from the KITTI platform, as shown in Figures 2(a) and 2(c), respectively. The camera and lidar sensors are calibrated with respect to each other; Figure 2(b) shows the example point cloud projected onto an image. As seen in Figure 1, our algorithm is comprised of the following steps:

- *Multi-level segmentation:* Segmentation is first performed in each modality at two levels: a low or finer level and a high or coarser level. Segments on the low-level segmentation are used to perform inference.
- *Feature Extraction:* Each segment from the low-level is associated to an overlapping segment from the higher-level. Features are extracted from the low-level segments, and concatenated with the features from their associated high-level segments.
- *Classification and Fusion:* Segments are first classified using *unimodal* image or point cloud-only classifiers. Segments which are within the overlapping coverage of both sensing modalities are then reclassified by fusing the soft outputs from the unimodal classifiers.
- *Post-processing CRF:* A pairwise CRF is used as a post-processing step to enforce spatial consistency.

We describe each of the steps in more detail below.

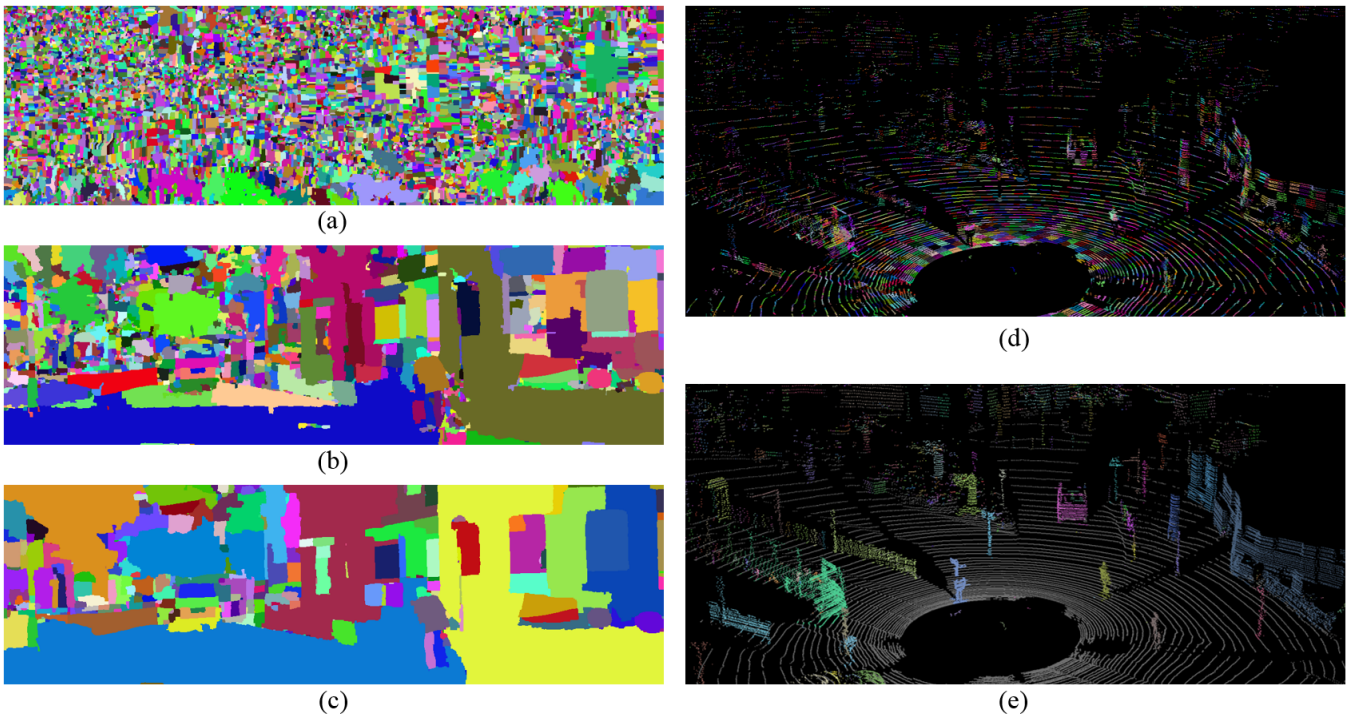


Fig. 3: Example image segmentation; Superpixels generated using thresholds t = (a) 0.06; (b) 0.1; (c) 0.2. Example point cloud segmentation; (d) Fine segmentation via VCCS [24]; (e) Coarse segmentation via connected component segmentation.

A. Segmentation

To capture information at multiple scales, two levels of segmentation are performed. Intuitively, the low-level segmentation should group segments which belong to the same class, but in a conservative manner to avoid label straddling from undersegmentation. The high-level segmentation should aim to capture larger object-level information.

In the image modality, we use the Multiscale Combinatorial Grouping (MCG) algorithm [23]. Thresholds of 0.10 and 0.20 for low and high levels, respectively, were empirically found to work well. Examples are of low and high level segmentations are shown in Figures 3(b) and 3(c), respectively.

In the point cloud modality, we use the Voxel Cloud Connectivity Segmentation (VCCS) algorithm [24] to produce fine-level segments. The algorithm uses region growing to produce uniformly sized *supervoxels*, while respecting object boundaries, inferred by large changes in local normals. We choose to produce supervoxels of dimension approximately 0.5m, smaller than the objects of interest in our application. We generate high-level segmentations from a connected component segmentation, similar to [25], resulting in a ground plane segment and connected component segments.[1] An example point cloud segmentation is shown in Figure 3. As seen in Figure 3(e), connected component segments can vary in size and encompass cars, people, or entire building façades on the order of 10m.

[1]Points are binned into a 0.1×0.1m 2D grid in the x and y dimensions. The difference in height of the points within each cell Δz is computed. Cells with Δz above and below 0.1m are labeled *occupied* and *unoccupied*, respectively. Connected components are extracted on the occupied cells. RANSAC plane fit is run on points in unoccupied cells to obtain a ground plane estimate. Points above the ground estimate which lie within occupied cells of the same connected component belong to the same segment.

B. Feature Extraction

We extract features for each segment at each scale. The features at the fine level contain bottom-up information, such as color and texture, but do not contain as much top-down shape information. Since we have segmented at multiple scales, each low-level segment can be associated to single high-level segment. The feature vector for each low-level segment is then augmented with features extracted from the associated high-level segment.

We compute features for each superpixel in an image, similar to those in [6], as summarized in Table I(a). In particular, we compute size, shape, position, color features. In addition, we compute a high-dimension Bag-of-words (BoW) descriptor by computing SIFT [26] and encoding using vector quantization. To incorporate contextual information, we dilate the bounding box of the superpixel by 10 pixels on each side and compute color and SIFT BoW features on this region, excluding the superpixel mask. The same features are computed on the high-level segments, with some exclusions. Specifically, the contextual features are not computed, since the segments are already large. The high-dimensional descriptors are also more suited for low-level bottom-up classification and not computed.

Table I(b) shows the features we choose to extract on the point cloud. We compute size, shape, position, and orientation features on both segmentation levels. PCA analysis

on a segment determines eigenvalues λ_i, where $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and corresponding eigenvectors $\mathbf{v_i} = (v_{ix} v_{iy} v_{iz})^T$. We define $\Lambda = \sum_{i=1}^{3} \lambda_i$. Similar to the image modality, the high-dimensional BoW descriptor, in this case Spin Images [27], is suited to bottom-up classification, and thus only computed on the low-level segmentation.

TABLE I: Features extracted.

(a) Image Features.

Type	Name	Dim	Low	High
Size/Shape	Area	1	✓	✓
	Equivalent Diameter	1	✓	✓
	Major/minor axes	2	✓	✓
	Orientation	1	✓	✓
	Eccentricity	1	✓	✓
Position	(x, y) - min, mean, max	6	✓	✓
	superpixel mask (8x8)	64	✓	✓
Color	rgb+lab (mean, std)	6	✓	✓
	rgb+lab (histogram)	48	✓	✓
High-dim	SIFT BoW	400	✓	
Contextual	contextual rgb+lab (mean, std)	6	✓	
	contextual rgb+lab (histogram)	48	✓	
	contextual SIFT BoW	400	✓	

(b) Point Cloud Features.

Type	Name	Dim	Low	High
Size	Length proxy - λ_1	1	✓	✓
	Area proxy - $\sqrt{\lambda_1 \lambda_2}$	1	✓	✓
	Volume proxy - $\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$	1	✓	✓
Shape	Scatter - λ_3 / Λ	1	✓	✓
	Planarity - $(\lambda_2 - \lambda_3)/\Lambda$	1	✓	✓
	Linearity - $(\lambda_1 - \lambda_2)/\Lambda$	1	✓	✓
Position	$z - z_{gnd}$ - min, mean, max	3	✓	✓
Orientation	Verticalness - v_{1z}	1	✓	✓
	Horizontalness - $\sqrt{1 - v_{1z}^2}$	1	✓	✓
High-dim	Spin image BoW	1000	✓	

C. Classification

The next step of our algorithm is to classify each segment. Classification is learning a mapping from feature vector of length N to a label $l \in \{1, ..., L\}$. A *soft* classifier learns a mapping from feature space $\mathbb{R}^N$ to a probability mass function (pmf) vector over L labels Δ^L, where $\Delta \in [0, 1]$. We have found the Random Forest (RF) classifier to be effective and use the implementation from [28].

A well-known challenge in semantic segmentation problems is class imbalance [7] [9]. In semantic segmentation problems, *stuff* classes tend to dominate the majority of the datapoints. Without addressing the class imbalance issue, a classifier typically learns to disregard the rare classes, resulting in zero or near-zero recall of those classes. This is addressed in [7] and [9] by subsampling their large training dataset at different rates based on object class occurrence, artificially boosting the representation of rare classes during classifier training. We take a similar approach, but address the issue by *reweighting* our samples rather than subsampling, due to a more limited training set size. Specifically, we calculate the probability distribution over label classes in the training set, mix it evenly with a uniform distribution, and reweight our training examples to reflect this new mixed distribution. This procedure is performed when training all of our classifiers. The performance of the unimodal classifiers is explored in Section III-C.

D. Fusion

There are two choices for fusion: *early* and *late*. In both cases, with two sensing modalities, three classifiers are trained: an *image* classifier, a *point cloud* classifier, and a *fusion* classifier, which operates on regions with overlapping sensor coverage. Let feature dimensions be N_{img} and N_{pc} in the image and point cloud domain, respectively.

In early fusion, unimodal classifiers P_{img} and P_{pc} are evaluated on segments which are covered by a single sensor.

$$P_{img} : \mathbb{R}^{N_{img}} \rightarrow \Delta^L, P_{pc} : \mathbb{R}^{N_{pc}} \rightarrow \Delta^L \quad (1)$$

In regions with overlapping coverage, a fusion classifier is trained and evaluated on a separate feature set of length N_{pc+img}, comprised of information from both modalities, e.g. the concatenation of the unimodal feature vectors.

$$P_{earlyfusion} : \mathbb{R}^{N_{pc+img}} \rightarrow \Delta^L \quad (2)$$

In a late fusion architecture, classifiers are first run on all segments in each sensing modality separately, as described in Equation 1, regardless of whether the segment falls under overlapping coverage. A second level of classification is then performed on segments with overlapping coverage to fuse the unimodal classification results. We use the concatenation of the output pmf vectors from the image and point cloud classifiers, and learn the following mapping.

$$P_{latefusion} : \Delta^{2L} \rightarrow \Delta^L \quad (3)$$

We run the fusion algorithm on extremely fine superpixels, as shown in Figure 3(a), to protect against label straddling. We take the *stacking* approach proposed in [18] to generate training data for the fusion classifier. The original training data is split into $k = 2$ folds. Unimodal classifiers are trained on each set of $k - 1$ folds and evaluated on the remaining fold. Those results are then used to train the fusion classifier.

An advantage of early fusion is that the joint feature space between the modalities is potentially more expressive. However, the learning problem becomes more difficult, as the classifier must learn a mapping from a higher-dimensional feature space $\mathbb{R}^{N_{pc+img}}$. Furthermore, since fusion only applies to *overlapping* regions, it does not enjoy the abundance of training data available to unimodal classifiers. In the case of KITTI data, as shown in Figure 2, the point cloud covers approximately 60% of the field-of-view (FOV) of the image, and the frontward facing camera only covers approximately 25% of the FOV of the point cloud. Thus, with a higher-dimensional feature space and less training data, the classifier in Equation 2 is prone to overfitting. The early fusion strategy proposed in [3] avoids this challenge by using simple, low-dimensional features. However, expressiveness is then inherently lost in comparison to the expressive features in our method. In addition, though not explored in this study, using late fusion would allow one to leverage outside image-only or point coud-only datasets, which is more abundant than datasets which contain both modalities.

In the late fusion architecture, the inputs to the fusion classifier are the outputs of the unimodal classifiers. The unimodal classifiers are trained on the *entire* training set in each modality, not merely the overlapping region, thus enjoying an abundance of training data. The unimodal pmfs serve as a compact and expressive *mid-level feature*, and the classifier mapping in Equation 3 can be more easily learned from less training data. The performance gain from late fusion is explored in Section III-D.

E. Post-processing Graphical Model

At this point, superpixels have been classified individually and likely contain some spatial discontinuities. There is an opportunity for additional refinement by considering the overall structure of the problem. Specifically, neighboring superpixels with similar depth values are more likely to have the same label. We incorporate this intuition with a post-processing CRF, a commonly used probabilistical framework to balance local beliefs and spatial consistency [8] [9] [10] [11] [12] [13] [14]. Using a CRF typically results in small quantitative improvement but large qualitative improvement, and will be discussed in Section III-E.

Let $\mathcal{V}$ describe the set of all superpixels, and $\mathcal{E}$ describe the set of all neighboring superpixels. We wish to infer the labeling $z_i \in \{1, ..., L\}$ of each superpixel $i \in \mathcal{V}$. Local beliefs are described as $x_i \in \Delta^L$, which are soft outputs from the late fusion classifier in Equation 3. The set of all labelings and local beliefs are denoted by $\mathbf{z}$ and $\mathbf{x}$, respectively. The energy function describes the affinity of all possible labelings $\mathbf{z}$, and is composed of *unary* potentials Ψ_i, which encode the likelihood of superpixel i belonging to a given class, and *pairwise* potentials Ψ_{ij}, which encode the likelihood of a pair of neighboring superpixels $(i, j) \in \mathcal{E}$ taking the same label. Parameter λ controls the relative importance between local beliefs and spatial consistency.[2]

$$E(\mathbf{z}, \mathbf{x}, \lambda) = -\sum_{i \in \mathcal{V}} \Psi_i(z_i|x_i) - \lambda \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(z_i, z_j) \quad (4)$$

The energy can be equivalently described as a probability distribution, where Z is the normalizing constant.

$$P(\mathbf{z}|\mathbf{x}, \lambda) = \frac{1}{Z} \exp(-E(\mathbf{z}, \mathbf{x}, \lambda)) \quad (5)$$

The most likely labeling is obtained by minimizing the energy, or equivalently, maximizing the probability.

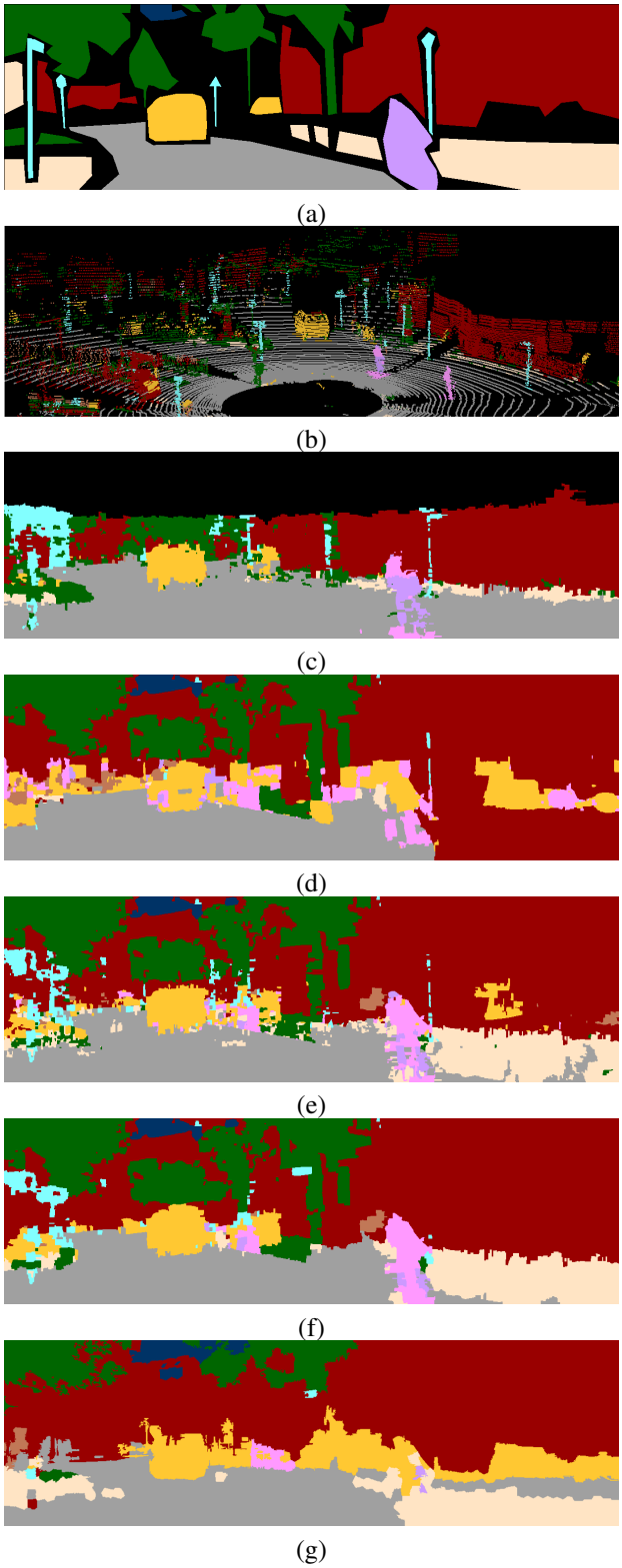
$$\mathbf{z}^* = \arg\min_{\mathbf{z}} E(\mathbf{z}, \mathbf{x}, \lambda) = \arg\max_{\mathbf{z}} P(\mathbf{z}|\mathbf{x}, \lambda) \quad (6)$$

The unary potential for superpixel i is the local belief x_i, weighted by superpixel perimeter C_i, which prevents small superpixels from overly influencing large neighbors.

$$\Psi_i(z_i = l|x_i) = C_i x_i^l \quad (7)$$

We define the pairwise potential between superpixels *(i,j)* to be associative, as neighbors are more likely to be of the same class. The potential is weighted by the length the shared

[2]Parameter λ=0.563 was set using cross validation and a parameter sweep.



(a) (b) (c) (d) (e) (f) (g)

Fig. 4: (a) Ground truth; (b) point cloud unimodal classification; (c) point cloud unimodal semantic segmentation projected onto image; (d) image unimodal semantic segmentation; (e) fused semantic segmentation; (f) our overall system; (g) [3]; Key: building, sky, road, vegetation, sidewalk, car, pedestrian, cyclist, sign/pole, fence.

boundary, denoted by C_{ij}, as neighbors with a large shared boundary are more likely to belong to the same object class. The potential is also weighted by a decreasing function of the difference in superpixel depths, $|d_i - d_j|$, as superpixels which are further apart spatially are less likely to be from the same class.[3]

$$\Psi_{ij}(z_i = l, z_j = m) = \delta(l, m) C_{ij} \exp\left(\frac{-(d_i - d_j)^2}{2\sigma_d^2}\right) \quad (8)$$

Because pairwise terms are *submodular*, Equation 6 can be efficiently and approximately solved using the well-known α-expansion algorithm [29].

III. Results

A. Dataset

We have used the publicly available KITTI dataset [1] [2], consisting of image and 3D point clouds. Previously published results on semantic segmentation of the KITTI dataset [21] [3] used a limited ground truthed dataset, comprised of 45 and 25 acquisitions on the training and test sets, respectively. Furthermore, all acquisitions on the test set were from a *single* sequence. We annotated a set of 252 acquisitions from 8 sequences, and split the dataset into 140 and 112 acquisitions for training and testing, respectively, with 4 sequences in each.[4] The split was chosen to balance object class representation in the sets. The labeled object classes were *building, sky, road, vegetation, sidewalk, car, pedestrian, cyclist, sign/pole* and *fence*. Performance was also evaluated over a coarser label set of *building, sky, ground, vegetation* and *object*, as used by Cadena and Košecká [3]. The *ground* class in the coarse label set consists of the *road* and *sidewalk* classes in the fine label set. The *object* class in the coarse label set is comprised of the *car*, *pedestrian*, *cyclist*, *sign/pole*, and *fence* classes.

B. Comparison to state-of-the-art

Results on the coarse and fine label sets on the images are shown in Tables II and III, respectively. The 1st rows show performance of current state-of-the-art [3].[5] The last rows show the results of our system.[6] On the coarse label set, as seen in Table II, the pixel-wise and class-average performance of our system is 93.4% and 89.8%, versus previous state-of-the-art [3] performance of 89.8% and 86.1%, respectively. This represents a 35.3% reduction in incorrectly labeled pixels. Of the *stuff* categories, *building* and *vegetation* classes see significant increases, from 90.2% and 86.3% to 95.0% and 92.8%, respectively, whereas *sky* performance decreases from 95.6% to 92.6%. The most dramatic change is the *object* class, which covers all *thing* classes, which increases from 61.0% to 70.1%.

[3] $\sigma_d = 5$m

[4] Images were annotated using ground truthing tool [30]. Point clouds were annotated using an extension of CloudCompare [31] built in-house.

[5] Results were obtained by separately retraining and testing on each label set using publicly available code and verification from authors.

[6] Results for fine label set were obtained by training and testing on the fine label set. Results for the coarse label set were obtained by mapping the fine label set results onto the coarse label set.

On the fine label set, as seen in Table III, the pixel-wise performance increase of our overall system is 89.3%, compared to 84.1% from [3], a reduction of 32.7% of incorrectly labeled pixels. A more dramatic increase is seen in the class-average accuracy, from 52.4% to 65.4%. Performance increases are seen on the *stuff* categories, but more dramatic increases are shown in the rare categories. For example, performance on the *pedestrian* class is more than doubled, from 28.6% to 65.1%. Recall performance on the difficult *cyclist*, *sign/pole*, and *fence* classes are dramatically increased from 4.0%, 2.5%, and 2.3% to 7.3%, 13.8%, and 43.2%, respectively.

The confusion matrix of our overall system is shown in Figure 5(d). High accuracy is achieved for *stuff* classes *building, sky, road*, and *vegetation*, all above 92.5%. The *sidewalk* class is difficult to distinguish from *road*, but our algorithm is able to take advantage of cues from multiple modalities to achieve high performance of 69.7%. The *cyclist* class only achieves 7% recall, but is often confused with the similarly looking *pedestrian* class 73% of the time. The *fence* class performs adequately at 43%, and is often confused with *vegetation*, due to its proximity in the test set, and sometimes with *building*, due the similarity in shape and appearance. The *sign/pole* class remains difficult, but *thing* classes *car* and *pedestrian* perform well.

The 2nd, 3rd, and 4th rows of Tables II and III show the performance of our system at various stages: after per-superpixel image classifiers, after late-fusion, and finally, after the post-processing CRF, and are discussed in further detail in Sections III-C, III-D, and III-E, respectively.

C. Effect of multiple scales

Table IV shows a performance comparison of the unimodal classifiers in the image and point cloud domains, given information at various scales: low-level only, high-level only, and low & high-level features together. In both domains, using low-level information results in higher performance than high-level only, due to increased likelihood of oversegmentation at the high-level. Performance increases in both domains are observed when multiple scales are used. This is especially evident in the point cloud domain, which shows a 2.9% increase in pixel-wise accuracy and a dramatic 9.8% increase in class-average accuracy. Performance for every class except *road* is increased when using information from multiple scales. In particular, the recall performance for the *cyclist* class increases from 0% when using low or high-level features only to 19.3% when using both scales. The *building*, *car*, *pedestrian*, *sign/pole*, and *fence* classes also see significant improvement.

An example segmentation using our image only classifier is shown in Figure 4(d), where ground truth is marked by Figure 4(a). In the example image, *stuff* classes such as *building*, *sky*, *vegetation*, *road* are generally classified correctly. However, small objects are difficult to accurately label, as parts of the cyclist are misclassified as *car*, and the pole on the left of the image is completely missed. Pieces

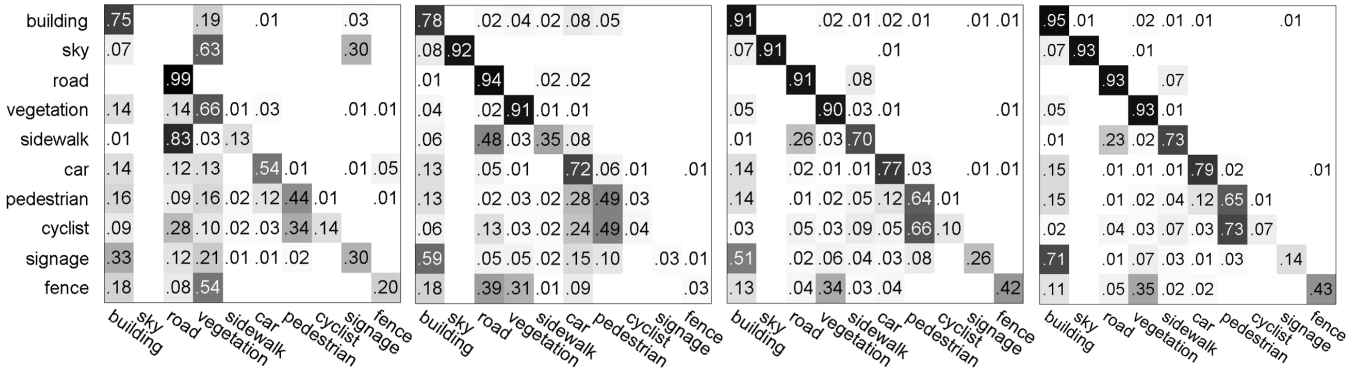


Fig. 5: Confusion matrices on *overlapping* sensor region; (a) Point cloud unimodal classifier; (b) Image unimodal classifier; (c) Late-fused classifier; (d) Confusion matrix of our overall system on *image* region.

TABLE II: Coarse label set performance comparison against state-of-the-art; glob: pixel-wise accuracy, class: class-average accuracy, bldg: building, sky: sky, gnd: ground, veg: vegetation, obj: object.

	Performance							Information Used		
	glob	class	bldg	sky	gnd	veg	obj	img	pc	crf
Cadena and Košecká [3]	89.8%	86.1%	90.2%	**95.6%**	97.4%	86.3%	61.0%	✓	✓	✓
Ours (image only)	89.4%	86.1%	87.5%	92.5%	91.9%	92.5%	66.1%	✓		
Ours (late fused)	92.8%	89.3%	93.5%	92.5%	98.1%	92.0%	**70.4%**	✓	✓	
Ours (CRF)	**93.4%**	**89.8%**	**95.0%**	92.6%	**98.5%**	**92.8%**	70.1%	✓	✓	✓

TABLE III: Fine label set performance comparison against state-of-the-art; glob: pixel-wise accuracy, class: class-average accuracy, bldg: building, sky: sky, road: road, veg: vegetation, side: sidewalk, car: car, ped: pedestrian, cyc: cyclist, sgn: sign/pole, fnc: fence.

	glob	class	bldg	sky	road	veg	side	car	ped	cycl	sgn	fnc
Cadena and Košecká [3]	84.1%	52.4%	92.5%	**95.7%**	92.5%	86.3%	51.5%	67.9%	28.6%	4.0%	2.5%	2.3%
Ours (image only)	83.5%	53.3%	87.5%	92.5%	94.5%	92.5%	34.5%	71.4%	49.0%	3.6%	4.1%	3.3%
Ours (late fused)	88.0%	64.8%	93.5%	92.5%	91.2%	92.0%	69.7%	76.5%	63.7%	**10.0%**	**16.6%**	42.2%
Ours (CRF)	**89.3%**	**65.4%**	**95.0%**	92.6%	**92.6%**	**92.8%**	**73.3%**	**78.7%**	**65.1%**	7.3%	13.8%	**43.2%**

TABLE IV: Effect of incorporating multiple scales; glob: pixel-wise accuracy, class: class-average accuracy, bldg: building, sky: sky, road: road, veg: vegetation, side: sidewalk, car: car, ped: pedestrian, cyc: cyclist, sgn: sign/pole, fnc: fence.

	Scale		Performance											
	Low	High	glob	class	bldg	sky	road	veg	side	car	ped	cycl	sgn	fnc
img	x		83.0%	52.6%	86.6%	92.8%	94.1%	**93.6%**	29.6%	67.8%	**49.3%**	**6.1%**	**5.1%**	0.8%
		x	81.1%	50.4%	**88.7%**	**93.5%**	90.1%	90.6%	20.6%	**71.4%**	42.0%	3.1%	3.8%	0.1%
	x	x	**83.5%**	**53.3%**	87.4%	92.5%	**94.5%**	92.5%	**34.5%**	**71.4%**	49.0%	3.6%	4.1%	**3.3%**
pc	x		67.1%	38.6%	72.9%	-	89.4%	53.9%	26.1%	36.0%	22.4%	0.0%	30.4%	16.5%
		x	59.5%	32.4%	68.0%	-	**97.8%**	11.7%	3.8%	34.6%	35.8%	0.0%	38.3%	1.7%
	x	x	**70.0%**	**49.8%**	**86.9%**	-	89.2%	**55.0%**	**26.2%**	**50.0%**	**49.0%**	**19.3%**	**51.7%**	**21.1%**

of the car are also misclassified, along with the sidewalk on the bottom right.

D. Fusion results

Figure 4(b) shows the point cloud classification for the example acquisition, and Figure 4(c) shows its projection onto the image. As seen in Figure 4(c), the point cloud unimodal classifier performs better at classifying the cyclist, which is labeled as *cyclist* and *pedestrian*. The pole on the left of the image and almost the entire car are classified correctly as well. Significant qualitative improvement is seen after fusion, as shown in Figure 4(e). Specifically, the cyclist is classified as the easily confusable *pedestrian* class, and the pole and car are correctly classified. Even the sidewalk on the bottom right side is correctly identified, despite being classified as *road* in the point cloud domain and *building* in the image domain.

Quantitatively, our system receives a strong boost after sensor fusion. In the fine label set, as shown in Table III, pixel-wise performance is increased from 83.5% to 88.0%, and class-average performance is increased from 53.3% to 64.8%, and performance increases are seen in almost every object class. At this stage, our system outperforms the current state-of-the-art [3] in both label sets, as shown in the 1st and 3rd rows of Tables II and III.

Figures 5(a) and 5(b) show the performance of the 3D point cloud and image unimodal classifiers, respectively, on the overlapping region of the two modalities. Figure 5(c)

shows the confusion matrix on the overlapping region after fusion. A few interesting cases where fusion greatly improves performance are discussed below.

The *sidewalk* class is difficult to distinguish from *road*. In the point cloud, they have near identical shape appearance, with the only distinguishing feature being a slight elevation difference. In the image domain, *sidewalk* is more accurately classified due to color and horizontal position features, but is still misclassified as *road* the majority of the time. After fusion, however, *sidewalk* is classified correctly 70% of the time, a dramatic increase from 13% and 35% in the individual domains, as seen in Figures 5(a)(b)(c).

The fusion classifier is also able to learn modes of failure. For example, recall on the *fence* class is 42% after fusion, higher than the combined recall of 20% and 3% in individual domains, as seen in Figure 5. The fusion classifier learns that a segment classified as *road* in the image and *building* or *vegetation* in the point cloud may in actuality be *fence*.

E. Post-proessing CRF

Figure 4(f) shows a final semantic segmentation after the post-processing CRF. Compared to Figure 4(e), there is a qualitative improvement, as the result is much more spatially consistent. Quantitatively, as seen in the last two rows of Tables II and III, slight improvements are made in both pixel-wise and class-average accuracy. For example, on the fine label set, global and class-average accuracies are increased from 88.0% to 89.3% and 64.8% to 65.4%, respectively.

IV. CONCLUSIONS

We proposed a semantic segmentation algorithm which fuses information from multiple modalities. The algorithm effectively integrated information at multiple scales by performing multiple levels of segmentation and augmenting feature vectors on low-level segments with their associated high-level segments. A late fusion architecture was used to effectively leverage information from single modalities. Finally, a CRF was used as a post-processing step to balance local beliefs while enforcing spatial consistency. The algorithm was validated on image and point cloud data from the KITTI platform, and shown to achieve state-of-the-art results on the problem of urban semantic segmentation.

A few challenges remain. The *cyclist* class is often confused for *pedestrian*, and *sign/pole* remains difficult to identify. Integrating top-down detectors for *thing* classes in both modalities would be a future direction of research. Another future direction is integrating temporal information from per-acquisition imagery and point cloud scans. Integrating our results with reconstruction algorithms would be an interesting research direction as well.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Homeland Security under Grant Award 2011-DN-077-ARI049-03.

REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," 2013.
[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.
[3] C. Cadena and J. Košecká, "Semantic segmentation with heterogeneous sensor coverages," in *ICRA*, 2014.
[4] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Scene parsing with multiscale feature learning, purity trees, and optimal covers," *ICML*, 2012.
[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.
[6] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *ECCV*, 2010.
[7] ——, "Finding things: Image parsing with regions and per-exemplar detectors," in *CVPR*, 2013.
[8] G. Singh and J. Košecká, "Nonparametric scene parsing with adaptive feature relevance and semantic context," in *CVPR*, 2013.
[9] J. Yang, B. Price, S. Cohen, and M.-H. Yang, "Context driven scene parsing with attention to rare classes," *CVPR*, 2014.
[10] D. Anguelov, B. Taskarf, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of markov random fields for segmentation of 3d scan data," in *CVPR*, 2005.
[11] D. Munoz, N. Vandapel, and M. Hebert, "Directional associative markov network for 3-d point cloud classification," in *3DPVT*, 2008.
[12] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *CVPR*, 2009.
[13] R. Shapovalov and A. Velizhev, "Cutting-plane training of non-associative markov network for 3d point cloud segmentation," in *3DIMPVT*, 2011.
[14] Y. Lu and C. Rasmussen, "Simplified markov random fields for efficient semantic labeling of 3d point clouds," in *IROS*, 2012.
[15] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr, "What, where and how many? combining object detectors and crfs," in *ECCV*, 2010.
[16] P. Kohli, P. H. Torr *et al.*, "Robust higher order potentials for enforcing label consistency," *IJCV*, 2009.
[17] L. Ladický, C. Russell, P. Kohli, and P. H. Torr, "Associative hierarchical crfs for object class image segmentation," in *ICCV*, 2009.
[18] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *ECCV*, 2010.
[19] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert, "3-d scene analysis via sequenced predictions over points and regions," in *ICRA*, 2011.
[20] D. Munoz, J. A. Bagnell, and M. Hebert, "Co-inference machines for multi-modal scene analysis," in *ECCV*, 2012.
[21] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr, "Urban 3d semantic modelling using stereo vision," in *ICRA*, 2013.
[22] H. He and B. Upcroft, "Nonparametric semantic segmentation for 3d street scenes," in *IROS*, 2013.
[23] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *CVPR*, 2014.
[24] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation supervoxels for point clouds," in *CVPR*, 2013.
[25] M. Himmelsbach, T. Luettel, and H. Wuensche, "Real-time object classification in 3d point clouds using point feature histograms," in *IROS*, 2009.
[26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *IJCV*, 2004.
[27] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," in *PAMI*, 1999.
[28] P. Dollár, "Piotr's Image and Video Matlab Toolbox (PMT)," http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html.
[29] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *PAMI*, 2001.
[30] D. Hoiem, "Software," http://web.engr.illinois.edu/ dhoiem/, accessed: 2014-09-14.
[31] "Cloudcompare," http://www.danielgm.net/cc/, accessed: 2014-09-14.