Co-inference for Multi-modal Scene Analysis

Daniel Munoz, J. Andrew Bagnell, and Martial Hebert

The Robotics Institute
Carnegie Mellon University

Abstract. We address the problem of understanding scenes from multiple sources of sensor data (*e.g.*, a camera and a laser scanner) in the case where there is no one-to-one correspondence across modalities (*e.g.*, pixels and 3-D points). This is an important scenario that frequently arises in practice not only when two different types of sensors are used, but also when the sensors are not co-located and have different sampling rates. Previous work has addressed this problem by restricting interpretation to a single representation in one of the domains, with augmented features that attempt to encode the information from the other modalities. Instead, we propose to analyze all modalities simultaneously while propagating information across domains during the inference procedure. In addition to the immediate benefit of generating a complete interpretation in all of the modalities, we demonstrate that this *co-inference* approach also improves performance over the canonical approach.

1 Introduction

With the advent of an increasingly wide selection of sensing modalities (*e.g.*, optical cameras, stereo/depth cameras, laser scanners, flash ladar, sonar), it is now common to obtain multiple observations of a given scene. In general, however, the sensor observations from different modalities often do not uniquely correspond to each other. Examples: 1) A laser scanner will never return any depth readings past a maximum range limit, while a camera can measure pixels infinitely far. 2) Range sensors, such as the X-Box Kinect, will often have missing depth information due to imperfect correspondences. 3) Scanning range sensors now commonly used on ground vehicles generate point clouds with highly variable point density in 3-D because of variations in depth and incidence angle coupled with complex scanning patterns. Further complicating matters is the fact that it is physically impossible for the two sensors to have the exact same viewpoint, and in practice the sensors are often physically far apart. As a consequence, objects are often visible in one sensor but occluded in the other(s).

In this work, we address these fundamental challenges that arise in scene analysis from multiple modalities. While our approach could be applied to multiple sensors, for clarity, henceforth we focus on understanding scenes from images and 3-D point clouds; however, our approach is not specific to this application and relies on general definitions and operators. In our application, we are given an image, a 3-D point cloud, and the camera parameters to project the

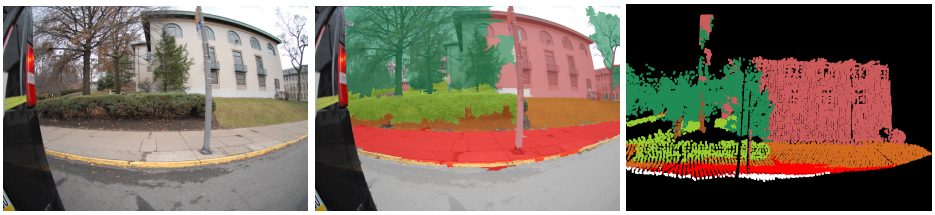


Fig. 1. Multimodal scene analysis. The reference scene (left) is observed with a camera and laser scanner and simultaneously classified in the image (middle) and 3-D point cloud (right). Color code: dark-red=sidewalk, white=road, light-green=shrub, dark-green=tree-top, brown=tree-trunk, light-red=building.

3-D points into the image plane. Our approach will simultaneously assign a semantic category (*e.g.*, building, car, *etc.*) to all elements in *both* domains, as illustrated in Fig. 1. The main contribution of this work is a technique for performing simultaneous/co-inference across domains when there is *not* a unique correspondence between modalities. A secondary contribution is a unique annotated dataset of images and 3-D point clouds of an urban environment for evaluation of algorithms for multi-modal vision tasks.

2 Motivation and related work

Two spatially adjacent scenes from our dataset are shown in Fig. 2 to highlight the challenges of this problem. Our dataset was collected with a laser scanner and camera mounted on a vehicle driving in an urban environment. As the vehicle moves, the laser scanner continuously collects samples and maps the 3-D points to a global reference frame. Because the laser scanner operates in a push-broom mode, the displacement is often on the order of tens of meters between the location of the scanner when it observes a 3-D point versus the location of the corresponding camera(s) into which the 3-D point is projected. Hence, there are often multiple 3-D points of different objects along the ray of the camera's (occluded) viewpoint, *e.g.*, the building behind the trees. In addition, the laser scanner samples the scene at a much sparser rate, *i.e.*, we have many more pixels than number of points. Currently, many datasets with combined image and depth data are post-processed in order to obtain a full-resolution depth image [1–3]. While interpolation might work well under appropriate conditions, an accurate and complete interpolation is impossible in general, especially in outdoor environments (*e.g.*, there is no depth for pixels past the maximum range of the sensor, and the density of measured points in 3D varies substantially).

The problem of analyzing scenes in combined 2-D and 3-D data has been investigated early in the literature [4, 5]; however, the problem has received an considerable increase in attention due to the ubiquity of data resulting from inexpensive sensors [6, 7]. The conventional way to approach this problem is to constrain the representation into only one of the modalities while integrating

Fig. 2. Example images and point cloud from our challenging dataset. The point cloud is colored by elevation. Colored circles are drawn to help the reader make correspondences between the domains.

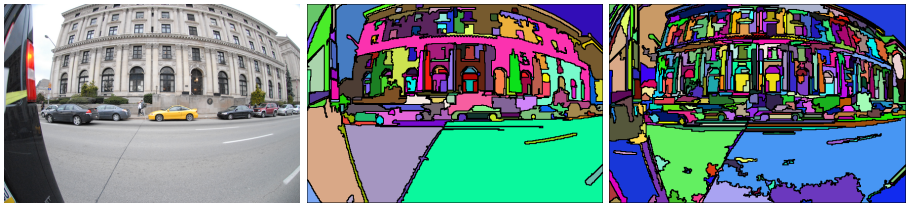
information from the other discarded domain as features. That is, the approach can be *2-D driven* [5, 8–12, 1], in that reasoning is done in the image while integrating 3-D features, or the approach can be *3-D driven* [7, 13–15], in that the predictions are made on the 3-D data while integrating 2-D features. These approaches are typically only applicable when the two modalities are in correspondence. In the commonly occurring case when there is a disparity between domains, constraining the modalities into a single representation can have negative consequences, as illustrated in Fig. 3. In the presence of this data mismatch, we instead propose to treat both modalities as first class objects, that is, we never discard data from either domain and we perform joint inference over all modalities. By coupling the inference over all modalities, we can propagate contextual information to and from data without correspondences, which would be discarded with the canonical approach, in order to aid predictions.

3 Approach

3.1 Overview

We wish to infer semantic labelings in both modalities simultaneously. In principle, we might define a single graphical model with edges linking nodes between modalities as well as high-order cliques over regions. Optimizing and learning parameters in such a graphical model is difficult because of the exponential number of label configurations and intractable structure. Instead, we follow the effective

Fig. 3. The effects of constraining the representation into a single domain. **Top** (a): Reference scene. **Middle** (b): 2-D driven approach. The image is segmented (left) and then back-projected into the 3-D point cloud (right) using occlusion reasoning. The 3-D region colors correspond to the 2-D segmentation, except the 3-D points colored black which are occluded with respect to the camera's viewpoint and are not associated with any 2-D region. **Bottom** (c): 3-D driven approach. The original 3-D point cloud is segmented (left) and then projected into the image plane (right) using occlusion reasoning. Note that not every pixel is associated with a 3-D region and that the the resulting 2-D regions are not connected due to occlusions, and sampling rates.



Reference image Coarser regions Finer regions

Fig. 4. Example hierarchical segmentation

sequential labeling approach of [16] for contextual scene understanding. In the next subsection, we briefly review a simplified version of the sequential labeling approach for a single modality and in the following subsection we extend it to the multi-modality scenario.

3.2 Inference in one modality

Given an image, [16] represents the scene through a hierarchical segmentation of coarse to fine regions (Fig. 4) and uses an iterative procedure that makes sequential predictions of the *distribution* of labels associated with each region. The procedure is designed to iterate over the different levels of the hierarchy and uses the previous predictions as contextual features to aid the next prediction.

More formally, let $\mathcal{X}_t \in \mathcal{X}$ be the set of regions describing an image at level t. For each level t, we wish to learn a predictor q_t whose predictions on $x_i \in \mathcal{X}_t$ match its true distribution vector $\hat{b}_i \in \mathcal{B} = \{b \in \mathbb{R}^K | b \geq 0, 1^T b = 1\}$ of K possible labels. We learn q_t by minimizing the KL divergence D_{KL} of the predicted distribution $b_{i,t} = q_t(x_i)$ to the region's empirical distribution $\hat{b}_i$ in training data:

$$\arg\min_{q_t} \sum_{x_i \in \mathcal{X}_t} D_{KL}(\hat{b}_i || q_t(x_i)). \tag{1}$$

Internally, q_t computes a descriptor using a feature function $f : \mathcal{X} \rightarrow \mathbb{R}^{d_1}$ to extract a fixed dimensional feature representation per region, such as color histograms (detailed in Sec. 4.3). In our experiments, we use a multi-class, MaxEnt model $q_t(x; \phi)[k] = \frac{\exp(\phi_k(f(x)))}{\sum_l \exp(\phi_l(f(x)))}$, where $\phi_k : \mathbb{R}^{d_1} \rightarrow \mathbb{R}$ is learned and returns the score for assigning the k'th label to the region.

The predictor q_t as described uses only features that are local to region x. In order to propagate label predictions from other levels of the hierarchy, we need to define additional features that encode contextual cues. We achieve this by defining a function $g : \mathcal{X} \times \mathcal{B} \rightarrow \mathbb{R}^{d_2}$ which encodes context for the given region from the given set of previous predictions. That is, if we define $B_t = \oplus_{\tau=1}^{t-1} \oplus_{x_i \in \mathcal{X}_\tau} \{b_{i,\tau}\}$, where $\oplus$ denotes the list concatenation operator, to be all previous predictions made over *all* regions in the hierarchy up to level t, then $g_t(x, B_t)$, can be viewed as contextual priors specific for region x. In practice, we use the contextual features $g_t(.,.)$ that describe the local, global,

and parent context suggested in [16]. Now, at each level t in the hierarchy we use the fixed-dimensional, augmented feature representation

$$\tilde{f}_t(x) = [f(x) \; ; \; g_t(x, B_t)] \in \mathbb{R}^{d_1+d_2}, \tag{2}$$

to train the predictor. Inference is performed by applying the predictors $\{q_t\}$, using the same feature functions $\tilde{f}_t$, in the order they were trained.

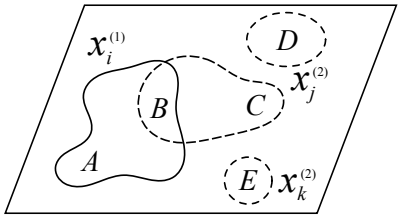
In the version of the algorithm described so far, the label distributions are propagated through the hierarchy in a top-down manner. However, in general, the procedure could traverse in any order, moving both up and down the hierarchy with g_t accordingly computing contextual features from regions below/above. Furthermore, multiple rounds of predictions could be performed at each level instead of a single one; in which case $g_t(x, \cdot)$, spatially pools previous predictions around region x, as described in [16]. An analogous method can be used for analyzing regions in 3-D point clouds [17]. Finally, the previous works use the technique of *stacking* [18] to train the predictors in order to avoid a cascade of overfitting due to the sequential nature of the training procedure.

3.3 Co-inference in multiple modalities

We denote by $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$ the set of regions in the hierarchical segmentations generated from two modalities, images and 3-D point clouds, respectively. A straightforward approach to analyze the modalities would be to construct two independent region hierarchies and to perform independent inference. However, instead of predicting over each domain separately, we want to couple the predictions so that information from one modality is propagated to the other. This is important because some domains are more apt at predicting certain categories than others. For example, as our experiments show, images are better for discriminating between physically similar things but with different texture (*e.g.*, road vs. sidewalk), and 3-D point clouds are better for semantically similar objects but at different scales (*e.g.*, buses vs cars). In order to use this inter-domain context, the predictors must incorporate this information at training-time. We now discuss how to modify the above sequential inference procedure to use the inter-domain context.

Inter-domain co-neighborhoods. First, we need a notion of correspondence between regions in different domains. We define an inter-domain co-neighborhood function $\eta_j : \mathcal{X}^{(i)} \to \wp(\mathcal{X}^{(j)})$, where $\wp$ is the power set operator. Given a region in one domain, this function simply returns a (potentially empty) set of neighboring regions in the other domain; we refer to this set of corresponding neighbors in the other domain as *co-neighbors*.

As previously discussed for our application, it would be unwise to directly use pixel and 3-D point correspondences; instead, we use the following approach. For each 3-D region in the 3-D segmentation $\mathcal{X}^{(2)}$, we project its points into the image plane, using z-buffering to maintain closest-to-camera ordering, resulting in a (partial) projected 2-D segmentation. Now, for any 3-D region $x^{(2)} \in \mathcal{X}^{(2)}$,



$$\eta_2(x_i^{(1)}) = x_j^{(2)}$$
$$\eta_1(x_k^{(2)}) = \emptyset$$
$$\nu(x_i^{(1)}, x_j^{(2)}) = B$$
$$|x_i^{(1)}| = A + B$$
$$|x_j^{(2)}| = B + C + D$$

Fig. 5. Synthetic example of inter-domain co-neighborhoods and overlaps. The solid outline is the only 2-D region $x_i^{(1)}$, and the dashed outlines are 2-D projections of the 3-D regions $x_j^{(2)}$ and $x_k^{(2)}$; note that the projection of $x_j^{(2)}$ is not simply connected.

$\eta_1(x^{(2)})$ returns all the 2-D regions that the projected segmentation of $x^{(2)}$ touches in the 2-D segmentation, and for any 2-D region $x^{(1)} \in \mathcal{X}^{(1)}$, $\eta_2(x^{(1)})$ returns all 3-D regions that $x^{(1)}$ touches in the projected segmentation. Figure 5 illustrates our co-neighborhoods.

Inter-domain overlap. Next, we need a notion of how much a region in one modality should influence a region in the other. We define an inter-domain overlap function $\nu : \mathcal{X}^{(i)} \times \mathcal{X}^{(j)} \rightarrow \mathbb{R}^+$, which assigns a non-negative value indicating a degree of correspondence between two regions in different modalities. We use the intersections of regions in the projected 3-D segmentation and the 2-D segmentation to define this overlap. Figure 5 illustrates inter-domain overlap.

Inter-domain context features. Using the above definitions, we define the fixed-length, inter-domain context feature function $h_t^{(i,j)} : \mathcal{X}^{(i)} \times \mathcal{B}^{(j)} \rightarrow \mathbb{R}^{K+1}$, which, for a given region in one domain, computes a contextual feature vector using its co-neighboring K-class predictions in the other domain. Formally,

$$h_t^{(i,j)}(x_k^{(i)}, B_t^{(j)}) = \sum_{x_l^{(j)} \in \eta_j(x_k^{(i)})} \frac{\nu(x_k^{(i)}, x_l^{(j)})}{|x_k^{(i)}|} \left[b_{l,t-1}^{(j)}, 1 \right]^T, \tag{3}$$

where $|x|$ is the area of the (projected) region as used in ν. In words, the first K values of this vector are the weighted average of the predictions of the co-neighboring regions in the other domain, where the weight is based on inter-domain overlap; and the last value is in $[0, 1]$ and is the fraction of overlap with the co-neighboring region(s). It is 0 when the first K values are 0, which happens when a region is observed in only one modality, and it is 1 when the first K values sum to 1, which happens when a region fully overlaps with co-neighbor region(s). This value is needed to disambiguate how much a region should trust its co-neighbors' predictions. For example, a co-context feature value of 0.2 could be due to high predicted probability and low overlap, or *vice versa*.

Algorithm 1 `train_co_inference`

1: **Inputs:** Labeled region hierarchies over N different modalities $\{\mathcal{X}^{(i)}\}_{i=1}^{N}$, Traversal sequence $[t_1, \ldots, t_T]$.
2: $Q^{(i)} = \emptyset, \forall i$ // predictors for each modality
3: $B = \emptyset, \forall i$ // predictions over all regions, in all domains encountered so far
4: **for** $t = t_1 \ldots t_T$ **do**
5: $B_t = \emptyset$
6: **for** $i = 1 \ldots N$ **do**
7: $q_t^{(i)} = \texttt{train\_predictor}(\mathcal{X}_t^{(i)}, B)$ // Solve Eq. 1 using Eq. 4 features
8: $Q^{(i)} \leftarrow Q^{(i)} \oplus \{q_t^{(i)}\}$ // Save for test-time
9: $[\mathcal{U}, \mathcal{V}] = \texttt{split\_data}(\mathcal{X}_t^{(i)})$ // $\mathcal{U} \cup \mathcal{V} = \mathcal{X}_t^{(i)}, \mathcal{U} \cap \mathcal{V} = \emptyset$
10: $q_U = \texttt{train\_predictor}(\mathcal{U}, B) \quad q_V = \texttt{train\_predictor}(\mathcal{V}, B)$
11: **for** $x \in \mathcal{U}$ **do**
12: $B_t \leftarrow B_t \oplus \{q_V(x)\}$
13: **end for**
14: **for** $x \in \mathcal{V}$ **do**
15: $B_t \leftarrow B_t \oplus \{q_U(x)\}$
16: **end for**
17: **end for**
18: $B \leftarrow B \oplus B_t$ // couple predictions among domains
19: **end for**
20: **Return:** Learned predictors for each modality $\{Q^{(i)}\}_{i=1}^{N}$

Putting it together. Given two hierarchical segmentations and a procedure for propagating information between regions in the different modalities, we can now jointly train the entire procedure. For simplicity in the explanation, we assume that the two hierarchies have the same number of levels. We train two sets of predictors $\{q_t^{(1)}\}, \{q_t^{(2)}\}$, one set for each hierarchy. Instead of training all the predictors for one domain first before starting to train the other, we instead train pairs of predictors at a time as we iterate over the levels. That is, we first train $q_{t-1}^{(1)}$ and $q_{t-1}^{(2)}$ before training $q_t^{(1)}$ and $q_t^{(2)}$. In order to couple the predictions and propagate context across domains, we augment our feature representation with the respective co-neighbors' predictions. That is, for each region $x^{(i)} \in \mathcal{X}_t^{(i)}$, we use the fixed-length feature representation

$$\hat{f}_t^{(i)}(x) = [\tilde{f}_t(x^{(i)}) \; ; \; h_t^{(i,j)}(x^{(i)}, B_t^{(j)})] \in \mathbb{R}^{d_1+d_2+K+1}, \tag{4}$$

when training $q_t^{(i)}$. Using $\hat{f}_t^{(i)}(x)$ in this way uses contextual information from the other modality j's previous predictions when training $q_t^{(i)}$. Algorithm 1 summarizes the training procedure in the simplest case of one example (observed with N modalities) and and using 2-fold stacking [18]; it is implied that each region x_i is associated with its empirical distribution $\hat{b}_i$. The test-time inference follows similarly, except we replace lines 7-16 with $B_t \leftarrow B_t \oplus \{q_t^{(i)}(x)\}, \forall x \in \mathcal{X}_t^{(i)}$, where $q_t^{(i)} = Q^{(i)}[t]$.

Although the presentation has focused on the image and point cloud setting, the general definitions of η and ν can be applied to any multi-modality scenario

	road	sidewalk	ground	building	barrier	bus-stop	stairs	bench
2-D	27.65	12.66	5.99	17.48	3.25	0.11	0.19	0.02
3-D	10.79	8.01	8.88	27.06	2.54	0.21	0.44	0.03
	shrub	tree-trunk	tree-top	small-veh.	big-veh.	bike	person	
2-D	2.46	0.79	17.89	7.22	1.78	0.03	0.62	
3-D	4.42	1.37	26.51	5.41	1.55	0.04	0.72	
	flag-pole	tall-light	short-light	post	sign	util-pole	wire	
2-D	0.01	0.17	0.03	0.27	0.24	0.22	0.57	
3-D	0.04	0.31	0.10	0.43	0.36	0.26	0.10	
	traff-pole	traff-signal	bag	trash	hydrant	mailbox	obstacle	
2-D	0.09	0.04	0.03	0.04	0.01	0.01	0.13	
3-D	0.14	0.06	0.03	0.06	0.01	0.02	0.10	

Table 1. Distribution (%) of categories in our dataset, per-domain.

for which there is an operational definition of the projection from one modality to another, which, in order to leverage information, must exist. For example, co-neighborhoods can be defined between samples that correspond to the same physical space (*e.g.*, in images and infrared) and/or time (*e.g.*, in audio and video). The key benefit of our approach is that we eliminate the constraint of requiring a unique correspondence between domains and that we can pass information in a softer manner through contextual features.

4 Experiments

4.1 Urban Image+Laser Dataset

We collected and annotated a dataset of 372 scenes (images and 3-D point clouds) obtained from a vehicle driving around an urban environment. The images were annotated using LabelMe [19], and the 3-D annotations are obtained by back-projecting these 2-D annotations; hence, the 3-D annotations are susceptible to subtle projection errors when objects are transparent/porous and/or have a high incident angle with the camera. 3-D points are mapped into a global reference frame and then registered to corresponding images; on average, 31,000 3-D points project into an image. Since the laser scans in push-broom mode, there exist scenes containing 3-D scan lines that do not cover the image due to when the vehicle moves slowly/stops. For each of the 372 scenes, the task is to assign each pixel and 3-D point to one of 29 semantic categories that typically occur in urban environments. The category names and their distributions are shown in Table 1. Other currently available "RGBD" datasets, *e.g.*, [1–3] consist primarily of range images from one viewpoint of the scene with co-located sensors and are often interpolated as a post-processing step to ensure the RGB and depth values have unique correspondence. In contrast, our goal is to evaluate the performance of scene understanding when there is mismatch between the two modalities.

4.2 Models

Given an image and a point cloud, our approach returns a complete labeling of both modalities simultaneously. We compare this approach with the natural baselines of using one modality in isolation and with augmented features computed in the other modality. As we build off the state-of-the-art hierarchical inference framework from [16, 17], we use their single-domain representations to build the baselines. Controlling for the same hierarchical representation, features, and predictors facilitates a fair comparison among six possible models:

1. **2D**: Hierarchical segmentation and features are computed only in the image; no 3-D data can be classified. This is the framework used in [16].
2. **2D+A**: Hierarchical segmentation and features are computed in the image. In addition, the 2-D regions are back-projected into the point cloud (Fig. 3(b)) and 3-D features are computed over these 3-D regions and appended to the feature descriptor. No 3-D data is classified with this model.
3. **3D**: Hierarchical segmentation and features are computed only in the point cloud; no 2-D data can be classified. This is the framework used in [17].
4. **3D+A**: Hierarchical segmentation and features are computed in the point cloud. In addition, the 3-D regions are projected into the image (Fig. 3(c)) and 2-D features are computed over these 2-D regions and appended to the feature descriptor. No 2-D data is classified with this model.
5. **Co**: Our proposed approach. Two hierarchical segmentations are separately constructed in the image and point cloud, with the same features computed over the regions as in **2D** and **3D**, respectively.
6. **Co+A**: Same as **Co**, but with each region's features augmented across domains as done in **2D+A** and **3D+A**

4.3 Segmentations, features, and predictors

All of the models require: 1) a hierarchical segmentation, 2) region features, 3) predictors. We use the efficient graph-based segmentation approach of [20] to construct 4-level region hierarchies in each domain. The nodes in the graph are defined over pixels/voxels and the edge similarity uses the difference in RGB/local geometry [21]. For the 2-D region features, we use average pooling of "soft" k-means quantized codes, as detailed in [22], where the codes are functions of the descriptor's distance to each cluster center. These quantizations are computed separately for texture, local binary patterns, SIFT, and color SIFT descriptors, as used in [23]. In addition, we compute simple geometry (area, perimeter, location) of each 2-D region and take the weighted average of adjacent regions' features [24]. For the 3-D region features, we also do soft-pooling over quantized spin images and local geometry [21], separately. In addition, we compute shape (local elevation, bounding box, geometry, orientation) of each 3-D regions [17]. We purposely do *not* use distance from the sensor to help reduce dataset bias of observing scenes from a vehicle on the road. Similar to [16], we optimize Eq. 1 with boosting, where the weak learners are vector regression trees, and are sequentially

trained using 10-fold stacking [18]. We iterate over the hierarchy from bottom $\rightarrow$ top $\rightarrow$ bottom with the sequence: $[\ell_1, \ell_1, \ell_2, \ell_2, \ell_3, \ell_3, \ell_4, \ell_4, \ell_3, \ell_3, \ell_2, \ell_2, \ell_1, \ell_1]$, where ℓ_1, ℓ_4 are the leaf and root levels, respectively.

4.4 Analysis

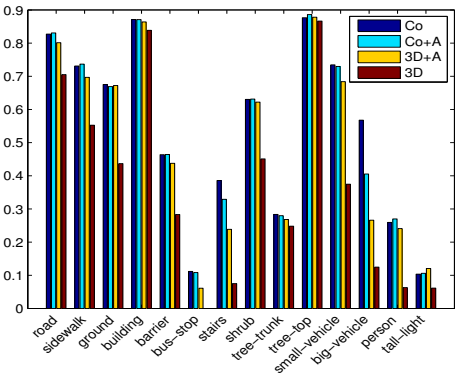
We evaluate our model on 5 different partitions (297-train/75-test) of the data, grouped by time[1]. As the models defined only over a single modality cannot make predictions on the other, we evaluate the performance on the points and pixels that correspond so that the comparisons between **Co** vs. **2D**/**3D** are consistent. However, note that **Co** will make predictions over the entire image and point cloud. As there is a severe (and unavoidable) imbalance in the number of samples per class, we evaluate the the per-class F_1 score, computed separately over pixels/voxels in each domain.

In Fig. 6, we present performance for each of the 6 models on the 3-D point clouds and images. We immediately see that feature augmentation in both domains is beneficial, especially in the 3-D point cloud. This result is expected as texture can help disambiguate among road, sidewalk, and ground in 3-D. Next, we see that in both domains **Co** $\geq$ **Co+A**, indicating that the information from the other domains can be encoded as our contextual features without a loss of representation power and avoids overfitting due to a larger, augmented feature representation. This is important as it simplifies the representation and computation time, *i.e.*, we do not need to duplicate the feature computation.

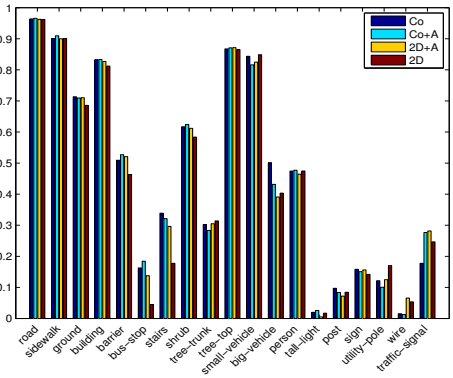
Figure 6 (c) shows an improvement in F_1 on all except one rare class in the 3-D point clouds. This improvement is due to the robustness of the representation: 1) There is bound to be back-projection errors when converting the 3-D point cloud into a 2-D segmentation from which 2-D features are computed. With the co-inference approach, we are more robust to these errors due to passing information as a distribution of labels, rather that encoding information in a large feature descriptor for which the spatial support could be poor. 2) As there is more image data than point cloud data, co-inference is indirectly passing larger amounts of global information to the 3-D point cloud, which is unavailable to **3D+A**. For example, the image component of **Co** examines the global context of all regions in the image, some of which might not have 3-D data. 3) As **Co** does not augment the features across domains, its feature dimension is smaller and less susceptible to overfitting (for **Co**, $f^{(2)}(x) \in \mathbb{R}^{98}$ and for **3D+A**, $f^{(2)}(x) \in \mathbb{R}^{693}$).

For images, Fig. 6 (d) shows a big gain in the big-vehicle class, modest improvements in 3 other classes and slightly better overall. The large improvement in the big-vehicle class can be explained through Fig. 7. For 2-D regions on the bus, corresponding 3-D regions have a large planar structure, similar to buildings for which they are often confused. Hence, simply augmenting the 3-D geometric features is not enough to disambiguate. By simultaneously reasoning in 3-D space, correct context can be propagated back into the image. Furthermore, we improve upon the vegetation that occlude each other. Fig. 6 also shows a

[1] This is needed to avoid testing on scenes that might overlap with the training data.



(a)



(b)

Label	**Co**	**3D+A**	Diff.
Road	.827	.802	**.026**
Sidewalk	.731	.697	**.034**
Barrier	.464	.438	**.026**
Bus-stop	.112	.061	**.051**
Stairs	.386	.239	**.147**
Tree-trunk	.284	.268	.015
Small-vehicle	.735	.684	**.051**
Big-vehicle	.568	.266	**.302**
Person	.260	.241	.019
Tall-light	.103	.120	-.017

(c)

Label	**Co**	**2D+A**	Diff.
Barrier	.509	.521	-.012
Bus-stop	.163	.138	**.025**
Stairs	.339	.297	**.042**
Small-vehicle	.844	.825	.019
Big-vehicle	.502	.391	**.111**
Person	.474	.465	.010
Tall-light	.020	.005	.015
Post	.097	.072	**.025**
Wire	.015	.066	**-.050**
Traffic-signal	.178	.282	**-.104**

(d)

Fig. 6. Per-class F_1 scores on our Image+Laser dataset, averaged over 5-folds. (a) Comparisons on the 3-D point clouds. (b) Comparisons on the images. Categories from (a) and (b) with at least a difference of 0.01 in F_1 are show in (c) and (d), respectively; differences of at least 0.02 are bolded. Categories not shown in (a) and (b) achieved 0.0 F_1 for all methods.

decrease in performance in the wire and traffic-signal classes because they are particularly hard classes to discriminate in 3-D point cloud data. As these two classes are physically very small and constitute a small fraction of the dataset, *none* of the 3-D models are currently able to detect them. Hence, since they cannot be discerned in the 3-D point cloud, co-inference cannot provide correct context for these classes. Note that the remaining classes which the 3-D models can predict are improved upon, on average.

In addition to achieving improved performance and complete understanding in both domains simultaneously, co-inference is more efficient in practice. On average, holding segmentation and feature computation constant, **Co** takes 0.46 s to classify the entire scene (image and point cloud), whereas using **2D+A** and **3D+A** takes 0.45 s + 0.39 s = 0.84 s. Furthermore, from a practical viewpoint, co-inference is simpler to implement than feature augmentation due to the special

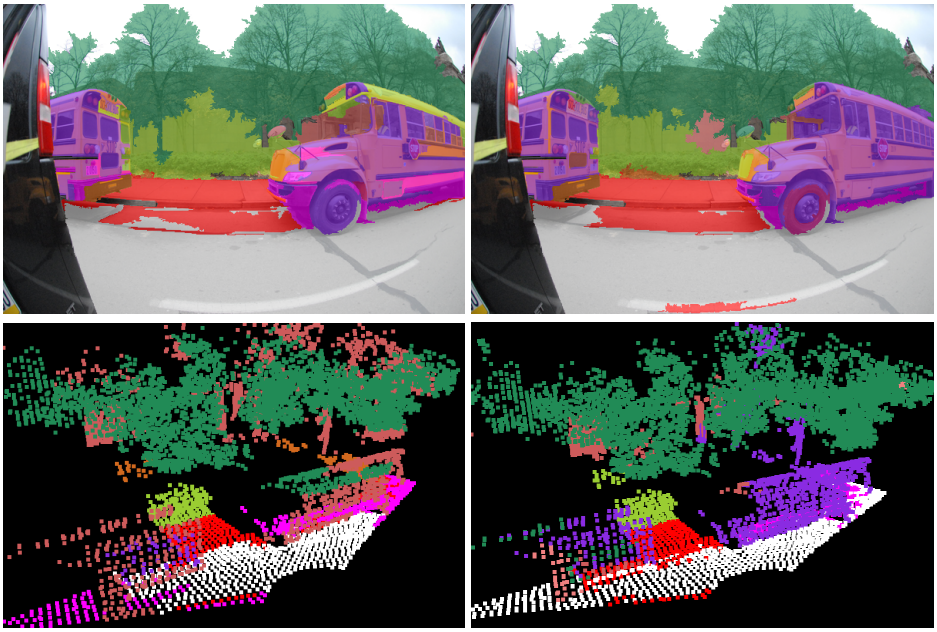


Fig. 7. Qualitative comparison of independently (top-left: **2D+A**, bottom-left: **3D+A**) and jointly (right: **Co**) trained models. The proposed co-inference approach does a much better job of identifying the big-vehicles (buses). Color code: same as Fig. 1, and pink=small-vehicle, purple=big-vehicle.

cases which must be accounted for; *e.g.*, when a region is observed in only one modality and features cannot be computed for it in the other modality.

5 Conclusion

This work addresses the problem of understanding scenes from multiple modalities when there is not a unique correspondence between data points across modalities. Instead of restricting our representation to a single modality and integrating information from the unselected ones, we treat both modalities as first class objects and propose a joint inference procedure that couples the predictions among all of the modalities. Our experiments demonstrate that our co-inference approach obtains improved predictions in all modalities compared to multiple, decoupled representations with the added benefit of efficiency and simplicity.

Acknowledgements

This work was conducted through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016.

References

1. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: 3DRR Workshop. (2011)
2. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A category-level 3-d object dataset putting the kinect to work. In: Consumer Depth Cameras in Computer Vision Workshop. (2011)
3. Liu, B., Gould, S., Koller, D.: Single image depth estimation from predicted semantic labels. In: CVPR. (2010)
4. Besl, P.J., Jain, R.C.: Invariant surface characteristics for 3d object recognition in range images. CVGIP **33** (1986)
5. Kweon, I.S., Hebert, M., Kanade, T.: Sensor fusion of range and reflectance data for outdoor scene analysis. In: NASA Workshop on Space Operations, Automation, and Robotics. (1988)
6. Baseski, E., Pugeault, N., Kalkan, S., Kraft, D., Worgotter, F., Kruge, N.: Indoor scene segmentation using a structured light sensor. In: 3DRR Workshop. (2007)
7. Koppula, H.S., Anand, A., Joachims, T., Saxena, A.: Semantic labeling of 3d point clouds for indoor scenes. In: NIPS. (2011)
8. Brostow, G., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: ECCV. (2008)
9. Gould, S., Baumstarck, P., Quigley, M., Ng, A.Y., Koller, D.: Integrating visual and range data for robotic object detection. In: M2SFA2 Workshop. (2008)
10. Xiao, J., Quan, L.: Multiple view semantic segmentation for street view images. In: ICCV. (2009)
11. Zhang, C., Wang, L., Yang, R.: Semantic segmentation of urban scenes using dense depth maps. In: ECCV. (2010)
12. Collet, A., Srinivasa, S., Hebert, M.: Structure discovery in multi-modal data: a region-based approach. In: ICRA. (2011)
13. Tombari, F., Stefano, L.D.: 3d data segmentation by local classification and markov random fields. In: 3DIMPVT. (2011)
14. Douillard, B., Fox, D., Ramos, F., Durrant-Whyte, H.: Classification and semantic mapping of urban environments. IJRR **30** (2011)
15. Lai, K., Bo, L., Ren, X., Fox, D.: Detection-based object labeling in 3d scenes. In: ICRA. (2012)
16. Munoz, D., Bagnell, J.A., Hebert, M.: Stacked hierarchical labeling. In: ECCV. (2010)
17. Xiong, X., Munoz, D., Bagnell, J.A., Hebert, M.: 3-d scene analysis via sequenced predictions over points and regions. In: ICRA. (2011)
18. Wolpert, D.H.: Stacked generalization. Neural Networks **5** (1992)
19. Russell, B., Torralba, A., Murphy, K., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. IJCV **77** (2007)
20. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV **59** (2004)
21. Medioni, G., Lee, M.S., Tang, C.K.: A Computational Framework for Segmentation and Grouping. Elsevier (2000)
22. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: AISTATS. (2011)
23. Ladicky, L.: Global Structured Models towards Scene Understanding. PhD thesis, Oxford Brookes University (2011)
24. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. IJCV **80** (2008)