Semantic Segmentation of Urban Scenes Using Dense Depth Maps

Chenxi Zhang, Liang Wang, and Ruigang Yang

Center for Visualization and Virtual Environments, University of Kentucky, USA

Abstract. In this paper we present a framework for semantic scene parsing and object recognition based on dense depth maps. Five view-independent 3D features that vary with object class are extracted from dense depth maps at a superpixel level for training a classifier using randomized decision forest technique. Our formulation integrates multiple features in a Markov Random Field (MRF) framework to segment and recognize different object classes in query street scene images. We evaluate our method both quantitatively and qualitatively on the challenging Cambridge-driving Labeled Video Database (CamVid). The result shows that only using dense depth information, we can achieve overall better accurate segmentation and recognition than that from sparse 3D features or appearance, or even the combination of sparse 3D features and appearance, advancing state-of-the-art performance. Furthermore, by aligning 3D dense depth based features into a unified coordinate frame, our algorithm can handle the special case of view changes between training and testing scenarios. Preliminary evaluation in cross training and testing shows promising results.

1 Introduction

Scene parsing, which refers to the process of simultaneously classifying and segmenting objects in an image, is one of the fundamental problems of computer vision. A successful scene parsing system is of great benefit to a variety of vision applications, such as object recognition, automatic driver assistance and 3D urban modeling. In this work, we focus on semantic segmentation of urban scenes from a monocular video sequence filmed at street level and propose an effective algorithm to address this particular problem. The most distinct feature that differentiates our approach from existing solutions lies in the use of dense depth maps recovered via multi-view stereo matching techniques as cues to achieve accurate scene parsing.

While the task of segmentation traditionally relies on color information alone, using depth information has some obvious advantages. Firstly it is invariant to lighting and/or texture variation; secondly it is invariant to camera pose and perspective change. Therefore using depth can potentially enable successful segmentation independent of illumination or view, significantly expanding the range of operation conditions. Recently, advances in structure from motion (SFM) techniques make it easier to obtain depth ques from video sequences. As a result there

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 708–721, 2010.

is notable progress in performing semantic segmentation using 3D cues. A pioneer work in using depth for urban scene segmentation is [9], in which the authors demonstrated that semantic segmentation is possible based solely on *sparse* 3D point clouds obtained from structure from motion techniques. Given the success of [9], a natural question raised is whether *dense* 3D information can perform equally well, or, even better on this challenging task. Our experiments indicate that this is true for street scene segmentation and recognition.

1.1 Related Work

Recently, many efforts have been made to achieve accurate semantic segmentation and classification. Traditional approaches employ 2D appearance information, such as color, texture, shape [10,3,12] and have achieved impressive results. However, a drawback of appearance based features is that they may change dramatically under different imaging conditions. For example, in day time and night, summer and winter, a scene may have different appearances. With recent advances in 3D imaging, 3D structure information has been exploited for semantic segmentation and recognition [9,11]. Specifically, when the input is a video sequence rather than a still image, the available motion based cues contain a large amount of information that can be used for segmentation and recognition. They are invariant to appearance changes. In [3], Liu et al. proposed a novel nonparametric approach for scene parsing using dense scene alignment. Their method is based on the alignment of testing image and its best matching image in the database by SIFT flow. However their formulation requires a large amount of training data of around 2700 fully annotated images [2].

The approaches most related to ours are [9] and [11]. Our method is inspired by the work of [9] with the following distinctions: First, we propose to use *dense (per-pixel)* depth map information for street scene segmentation, while in [9] they proposed to utilize sparse structure from motion point clouds; and in [11] they combined structure from motion and appearance descriptors together. Second, although both of the two previous approaches have achieved impressive results, some of the features, such as height above camera, is defined as the relative height between object and camera, therefore dependent on camera pose. Note there are two types of common camera configuration: front-view camera (as used in [9]) and side-view camera (as used in [11]). Training using one of the datasets and testing on the other is likely to lead to fail in both approaches. In our approach, because we only use per-pixel depth information without any dependence on camera pose or appearance, our segmentation algorithm can be easily formulated in a *view-independent* fashion. Applying our approach we can get satisfactory segmentation results from a video sequence using training data captured under a different configuration from testing data. Finally, while it is shown in [9] that motion-derived information leads to results comparable to existing state-of-the-art appearance based techniques, we demonstrate in this paper that semantic segmentation using only dense depth information outperforms both sparse structure from motion based method and appearance based method, or even the combination of sparse depth with appearance.

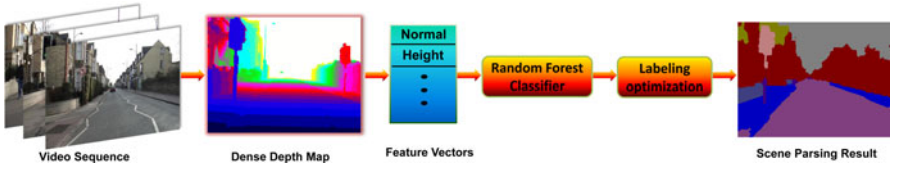


Fig. 1. Overview of our framework

1.2 Overview and Contributions

Figure 1 shows an overview of our scene parsing pipeline. The proposed algorithm starts from generating dense depth maps by plane swiping method. An over-segmentation is applied to video sequences and 3D information is obtained using dense depth maps. Five discriminative 3D features are extracted from dense depth maps and combined together to build a randomized decision forest classifier to obtain accurate semantic scene segmentation. The primary contribution is that we demonstrate a scene parsing algorithm that uses only dense 3D depth information to *outperform the combination of sparse 3D features and appearance*. Moreover, we demonstrate that by transferring those 3D features to a common coordinate system that is independent on camera pose, view-independent semantic scene segmentation can be achieved. Training in one type of camera view and application in a changed camera view is now possible.

In summary, our dense depth based segmentation algorithm lends itself well for real-world applications in which the viewpoints, lightings and object textures are likely to be significantly different from those captured in the training database.

2 Depth Maps Recovery

Stereo reconstruction of dense depth maps from a video sequence has long been a research topic in computer vision. Recently, there have been great advances that are based on high-quality stereo matching algorithms and effective 3D modeling pipelines [19,15]. As depth recovery is not the primary focus of this work, we simply modify existing techniques to compute the scene depth information from video.

Given an input video sequence $\{I^t\}$ captured at street level, we first employ the SFM software released by Zhang et.al. [20] to estimate camera parameters. Then, our stereo matching module takes as input camera poses and the corresponding frames from the video and produces a dense depth map D^t for each frame. The stereo matching pipeline used in our paper consists of a depth initialization step followed by a depth refinement process.

For depth initialization, we model stereo matching as an energy minimization problem. The global energy function contains three terms, i.e., a data term, a smoothness term and a segmentation term. The data term measures how well the depth map agrees with the input images under the color consistency assumption. In this paper, we use the standard plane-sweep approach as described in [4,17] to compute data costs. In our implementation, the plane-sweep stereo

is solved on 17 consecutive images where the middle one is the reference view. The smoothness term incorporates the assumption that the scene is piecewise smooth and penalizes assigning different depth values to neighboring pixels. We use the truncated linear model [6] to define our smoothness cost. In order to better handle textureless areas, we incorporate the segmentation information into our MRF stereo framework as a soft constraint. We first segment each reference frame using mean-shift algorithm [5]. Each color segment is treated as a nonfronto-parallel plane in 3D and a robust plane fitting method [7] is applied to estimate the plane parameters. Similar to [16], the segmentation term is modeled to penalize depth assignment that departs from that given by the pixel's corresponding plane parameter. For each frame I^t, we use belief propagation (BP) [6] to estimate an initial depth map $\widetilde{D^t}$ by minimizing our energy function.

In depth initialization step, we compute the disparity map $\widetilde{D^t}$ for each frame without considering the temporal consistency among depth maps. To address this issue, during the depth refinement step a multi-view fusion algorithm [13] is applied to refine the depth maps returned by BP. Depth maps after refinement are smooth both spatially and temporally and contain less visual artifacts.

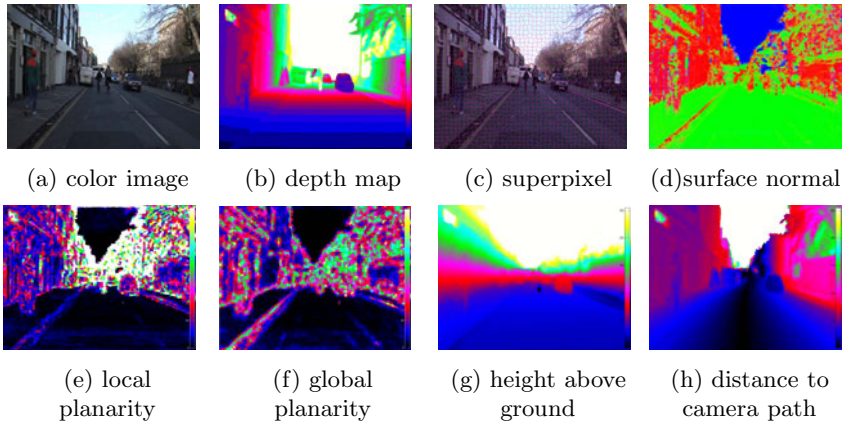
3 Semantic Segmentation from Dense Depth Maps

After recovering dense depth maps from videos, our algorithm starts by over-segment each image into homogeneous pixel clusters, i.e. *superpixels*, then extract feature vectors. These vectors are used for training and classifications. Raw classified outputs are combined via a pairwise Markov Random Field (MRF) for final segmentation. Details are presented below.

3.1 Image Over-Segmentation

Over-segmentation of image into superpixels is a common preprocessing step for image parsing algorithms. We choose to use over-segmentation as one of the preprocessing steps in semantic segmentation due to the following reasons: First, each superpixel in 2D images can be approximately viewed as a patch in 3D. Some features we employed in this work can be well defined over a patch, *e.g.*, surface normal, surface planarity etc. Second, over-segmentation can increase the chances that the boundaries of different object classes are extracted. In this regard, pixel-wise classification may result in less consistent boundaries. Finally, using over-segmentation can reduce the computational complexity of the system, since by counting each superpixel as one sample, the number of total samples are largely reduced as compared to pixel-wise training and testing.

We applied a geometric-flow based algorithm named "TurboPixels" [1] to achieve dense image over-segmentation. This recent technique can produce superpixels with uniform size and shape, maintain connectivity and compactness, and preserve original image edges. The choice for superpixel size is a key issue. On one hand, using large superpixels may bring in the risk of a superpixel spanning across multiple semantic objects. On the other hand, a small superpixel



(a) color image (b) depth map (c) superpixel (d)surface normal

(e) local planarity (f) global planarity (g) height above ground (h) distance to camera path

Fig. 2. Visualization of different steps and features in our algorithm

may contain insufficient points to precisely define a good feature. In our experiment, we set the initial number of superpixel as 6800 for image of size 960×720, roughly 100 pixels per superpixel.

3.2 Features from Dense Depth Map

For each superpixel, we extract five features in 3D space to train our classifier. These features are computed based on the 3D points within each superpixel. All five features are invariant of appearance changes. The five features are surface normal, height above ground, surface local planarity, surface neighboring planarity, and distance to camera path, which are denoted as $F_n^i, F_h^i, F_l^i, F_g^i, F_d^i$ for each superpixel i. Brostow et.al. [9] defined some similar features based on sparse 3D points and projected the features from 3D point clouds to 2D image plane. The way we compute our 3D features is different from theirs, and dense depth maps allow us to estimate these features in a more principled way.

Surface normal (F_n^i): We compute surface normal F_n^i of a 3D patch by fitting a least square plane to the set of 3D points within a superpixel. Similar method has been used in [11].

Height above ground (F_h^i): An object's height information is normally fixed and invariant to change of driving direction, thus can be used as a good feature for classification. [9] used height above camera center as one of the structure features. However, this feature is not invariant to the car on which the camera is mounted, or, large camera position changes. We instead use height above ground as our feature. Our algorithm requires a process to estimate ground plane parameters from depth map. In our implementation we use an iterative RANSAC method to estimate ground plane. At the beginning only 3D points whose normal are close to the up direction in camera coordinate system are used as samples. To avoid including points from sidewalk into plane fitiing, after

each RANSAC procedure we decrease the error threshold T_h used by RANSAC by half and use inliers from previous iteration to fit a new plane. Here T_h is simply a value that controls the point to plane distance. RANSAC will treat a point as an inlier if the distance from the point to plane is smaller than T_h. After a few iterations (6-8 in our implementation) the algorithm terminates and the final plane parameters are treated as ground plane parameters. We find this method works fairly well for our data, where nearly half of the pixels in images are dominated by ground scene. For a superpixel i, height above ground F_h^i is computed as the average distance to ground within the corresponding 3D patch.

Surface local and neighboring planarities: We in work define two types of surface planarity. One is the local planarity of a 3D patch (F_l^i), which corresponds to a superpixel in the image. The other is the neighboring planarity (F_g^i), that is, measuring the variance of a 3D patch orientation with respect to its neighboring patches. The local planarity is computed by using RANSAC based least square plane fitting and calculating the sum of square distances from points to the plane. The neighboring planarity is defined as the average difference of a 3D patch's surface normal with respect to its neighbors' surface normals. These features are useful for splitting planar and non planar objects, for example, building facades and plants.

Distance to camera path(F_d^i): Inspired by [9], we can take advantage of the distance to camera path to separate objects which are horizontally distanced from the camera. We compute the minimum distance from the centroid of the 3D patch to the camera path. The camera path is estimated by fitting a quadratic curve to the camera trajectory. This approach is more accurate and robust for computing objects' distances to camera path, and works well on various scenarios.

3.3 Randomized Decision Forest

Randomized decision forest is a well-known machine learning technique that has been employed in many computer vision tasks [14]. We use this technique to train our classifier based on features derived from the depth maps. In our implementation, we choose the proportion of training data used at each split node to be 0.66. A total number of 80 random decision trees are built for training. We experimentally find these parameters work well for achieving satisfactory recognition rate. We also apply the idea in [10] to balance the number of classes used for training, thus achieve a better class average performance.

3.4 Graph-Cut Based Optimization

We construct a pairwise Markov Random Field (MRF) for each image I by building a graph $G = \langle V, E \rangle$, where each node $v_i \in V$ in the graph represents a superpixel and each edge $e_{ij} \in E$ denotes the neighboring relationship between superpixels. The labeling problem is equal to assign a label $l_i \in L$ to each node

$v_i \in V$. The optimal assignment L_{assign} can be achieved by minimizing the energy:

$$E(L_{assign}) = \sum_{v_i \in V} \psi_i(l_i) + \lambda \sum_{e_{ij} \in E} \phi_{ij}(l_i, l_j) \tag{1}$$

Data term $\psi_i(l_i)$ and smoothness term $\phi_{ij}(l_i, l_j)$ are defined in the following paragraphs. They are computed from the feature responses and the randomized decision forest based classifier. After the costs are computed, a graph-cut optimization [18] is applied to obtain the global optimal labeling configuration.

Data term. The feature responses $F_n^i, F_h^i, F_l^i, F_g^i, F_d^i$ of each superpixel i are collected and applied rank normalization before passed to randomized decision forest for training. Specifically, given the samples for a feature F for all the superpixels as $F^1, F^2, ..., F^n$, where n is the number of superpixels, we first find the low-to-high order statistics $F^{(1)}, F^{(2)}, ..., F^{(n)}$ and then replace each image's feature value by its corresponding normalized rank, as:

$$\widetilde{F^i} = \frac{rank(x_i) - 1}{n - 1} \tag{2}$$

where F^i is the feature value for the i'th sample. The procedure uniformly maps all the features to the range of $[0, 1]$. When there are multiple samples with the same feature value, they are assigned the average rank of that value. We applied this data processing approach based on the fact that there are some scale factors between the features' measurements of different scenes. In addition, normalized rank is effective to compensate some inaccuracies induced by depth map generation and 3D features computation. After rank normalization, the feature responses of each superpixel are passed to randomized decision forest with corresponding ground-truth labels to build the classifier. When testing, the feature responses of each testing sample are passed to the classifier and a posterior probability distribution $P_i(l_i|F_n^i, F_h^i, F_l^i, F_g^i, F_d^i)$ which represents the probabilities the testing sample belongs to each category $l_i \in L$ is returned. We define the data term in MRF as :

$$\psi_i(l_i) = -logP_i(l_i|F_n^i, F_h^i, F_l^i, F_g^i, F_d^i) \tag{3}$$

Smoothness term. For a superpixel v_i and each of its neighbor superpixel v_j, the smoothness cost is defined as:

$$\psi_{ij}(l_i, l_j) = [l_i \neq l_j] \cdot \frac{1}{\delta \|c_i - c_j\|_2 + 1} \tag{4}$$

where $\|c_i - c_j\|_2$ is the L2-Norm of RGB color difference between neighbor superpixels. The penalty is inversely proportional to the color difference between neighbor superpixels. The more similar the colors of two neighbor superpixels are, the less likely they belong to different categories. In our experiment, the value of λ is set to be 1.9 and δ to be 0.1.

3.5 Temporal Multi-view Fusion

The temporally redundant information in video can be used to enhance the accuracy of scene parsing. In Xiao et al.'s work [11], they utilized multi-view information by defining a Markov Random Field for the entire sequence and imposing smoothness terms on superpixels in different images. The way we take advantage of multi-view segmentation consistency is different from theirs. The output of randomized decision forest classifier is a posterior probability distribution which represents the probabilities that a certain testing superpixel belongs to each class. Although we prefer to deem each superpixel belonging to one class, there are some thin structures, such as a column pole, which are far from filling the whole superpixel. In this case, a pixel-wise refinement is needed to achieve more accurate results. Moreover, for those superpixels which are misclassified, temporal fusion is able to increase the chance of making refinement based on neighbor frames' classification results. The fused probability distribution of each pixel can be represented as: $p_r(c) = \sum_{j \in N} w_j p_j(c), c = 1, 2, ..., k$ where $p_r(c)$ represents the probability of a pixel in reference view belonging to class c, $p_j(c)$ is the probability of correspondence pixel in neighbor view j belonging to class c. N represents the set of neighbor frames and k is the number of classes. The fused probability distribution is computed as weighted average of neighbor frames' probability distributions, where weights w_j are determined by how far the neighbor frame j is from reference view r. We define w_j as a Gaussian function:

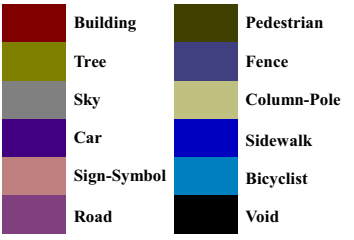
$$w_j = \exp\left(-\frac{(j-r)^2}{\sigma}\right) \tag{5}$$

where σ is set to 10 in our implementation.

After pixel-wise temporal fusion, we applied a superpixel level refinement by aggregating the refined probability distributions of all pixels within a superpixel. In this way we incorporate segmentation consistency across adjacent views.

3.6 Cross Training and Testing

We propose the idea of cross training and testing based on the fact that using per-pixel depth information is independent on camera pose and appearance. In our five 3D features, only surface normal is dependent on camera pose because we compute each 3D patch(a superpixel in 2D)'s normal in camera coordinate system. For cross view training and testing, surface normals in training and testing datasets should be transformed to a common coordinate system. We define the common coordinate system to be as the following: taking the surface normal of ground and camera moving direction(vertical to ground surface normal) as y axis and z axis, x axis can be obtained by the cross product of them. Note that since we only need to transform surface normals, the origin of the coordinate system does not matter. Any normal calculated in the camera coordinate can be rotated to the common coordinate frame, enabling cross training and testing.



Sequence	# of Frames	Dataset
0016E5	204	Day Train
0006R0	101	Day Train
0001TP1	62	Dusk Train
Seq05VD	171	Day Test
0001TP2	62	Dusk Test

Fig. 3. (a)Labeled colors for 11 object classes. (b)Split of sequence as training or testing data.

4 Experiments

We use the challenging CamVid database [8] to evaluate our algorithm's performance. The database includes four high quality video sequences at 30 fps with total duration about 10 minutes. The labeled ground truth images are extracted from the four original video sequences at a rate of 1 fps, with a part of one of the sequences at 15 fps. The image resolution is 960 × 720. Camera extrinsic and intrinsic parameters are also provided in the database. 32 semantic object classes are defined which include fixed objects, types of road surface, moving objects (such as vehicles and people) and ceilings (such as sky, tunnel, archway). Same as in [9], we use 11 dominant categories: Building, Tree, Sky, Car, Sign-Symbol, Road, Pedestrian, Fence, Column-Pole, Sidewalk and Bicyclist. Labeled colors for each object class are shown in Figure 3(a). A quantitative comparison to the state of art is provided. In addition, in order to show our algorithm's compatibility of view-independent training and testing, we carried out another test on images captured by a side-view camera provided by Google. Even using our classifier trained by the CamVid database which is mostly composed of front-view video sequences, we still get decent classification results.

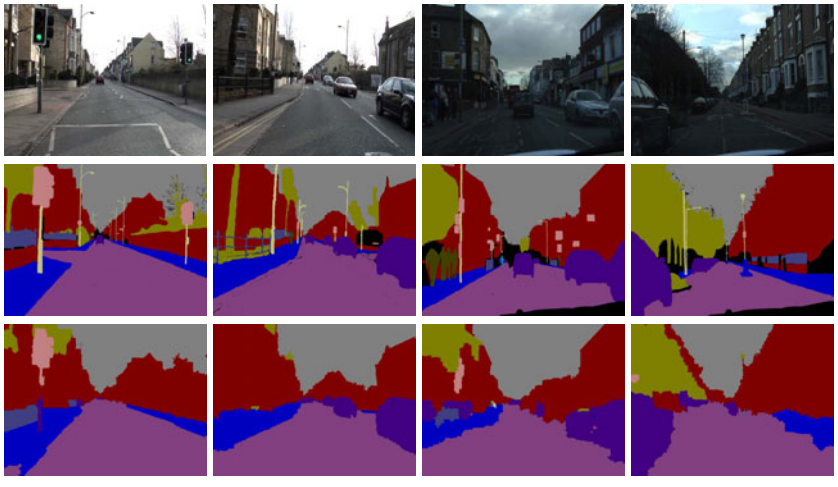
4.1 Evaluation Using the CamVid Database

For comparison with the results from [9], we split the training labeled frames into two groups in the same way as in [9], shown in Figure 3(b). Two groups of the labeled training data, 0016E5 and 0006R0, are used for day sequence training data, and another group 0005VD are used for day sequence testing. The first half of the dusk sequence (0001TP) are used for training, and the second half are used for testing.

We train our classifier using randomized decision forest based on the five dense depth based 3D features and carry out the same set of experiments as in [9]. Table 1 shows the quantitative testing result. In terms of global accuracy(i.e. pixel-wise percentage accuracy), we achieve 82.1% in comparison to 69.1% of combined structure from motion and appearance based approach and 61.8% of solely structure from motion based approach in [9]. Our algorithm also

Table 1. Comparison of Pixel-wise percentage accuracy with [9]. Our dense depth map based approach gives best result on 7 classes. 'Global' is the percentage of pixels correctly classified. 'Average' is the average value of per-class accuracies.

Alg	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Sidewalk	Bicyclist	Average	Global
Motion [9]	43.9	46.2	79.5	44.6	19.5	82.5	24.4	**58.5**	0.1	**61.8**	18	43.6	61.8
Appearance [9]	38.7	60.7	90.1	**71.1**	**51.4**	88.6	**54.6**	40.1	1.1	55.5	23.6	52.3	66.5
Combined [9]	46.2	**61.9**	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5	53.0	69.1
Depth	**85.3**	57.3	**95.4**	69.2	46.5	**98.5**	23.8	44.3	**22.0**	38.1	**28.7**	**55.4**	**82.1**



(Seq05VD #390) (Seq05VD #270) (0001TP #525) (0001TP #1770)

Fig. 4. Scene Parsing result samples. From top to bottom: test image, ground truth, dense depth map inferred segmentation. Note that our algorithm solely using dense depth map is able to achieve accurate segmentation and recognition of street scene.

performs well on most of per-class accuracies and outperforms combined motion and appearance based approach in 7 classes out of 11 classes, which are Building, Sky,Car, Sign-symbol, Road,Column-Pole and Bicyclist. There are two categories (Sidewalk, Pedestrian) that our approach's performances are not as good as in [9]. Both can be attributed to the lack of high-quality depth maps, in which the depth difference between the ground and sidewalk is small. Pedestrian is moving so its depth map is usually wrong. In addition, there are not sufficient samples for the Pedestrian class. Applying superpixel based training on such small classes may face the problem of insufficient samples compared with pixel-wise training approach. Figure 4 shows the qualitative results achieved by our approach.

Table 2. Comparison of pixel-wise percentage accuracy with [9] in illumination variation test

Algorithm	Day Train - Dusk Test	Dusk Train - Day Test
Mot&Struct	45.5%	59.4%
Appearance	21.7%	50.5%
Depth	**63.4**%	**69.2**%

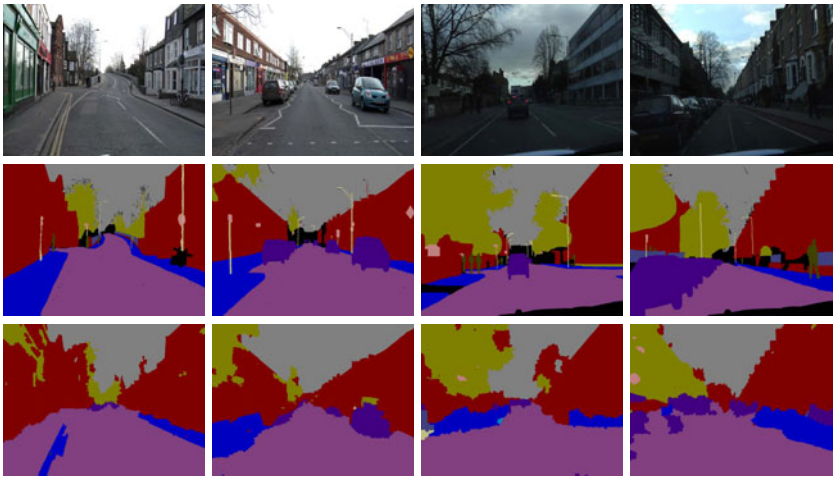


Fig. 5. Robustness to lighting condition changes. In the left figure, from top to bottom: test image, ground truth, segmentation results. The left two columns are examples of training by a dusk sequence and testing on day sequence. The right two columns show results when the training and the test sequence are swapped. Table 2 shows the overall accuracy and comparison to numbers reported in [9].

Testing on illumination variation. One expected advantage of our approach is its independence of appearance or illumination changes. We carried out a test training in one lighting condition(day/dusk) and testing on the other(dusk/day). [9] also carried out the same test using sparse 3D features and appearance features. We compare our results with theirs in Table 2 and Figure 5. As expected, our method has significantly improved the global classification accuracy. The improvement from dusk-training-day-testing is less than that from day-training-dusk-testing, probably due to the poor depth maps generated at dusk. Using active sensors, such as LiDAR scanners can alleviate this problem.

Multi-View Temporal Fusion Test. We test our multi-view temporal fusion algorithm using test sequence Seq05VD. A neighborhood of 40 frames of the reference frame are used for fusion. Some results are shown in Figure 6. It can be seen that fusion improves the overall consistency of the segmentation, which is particularly pronounced when viewing the sequence as a video.

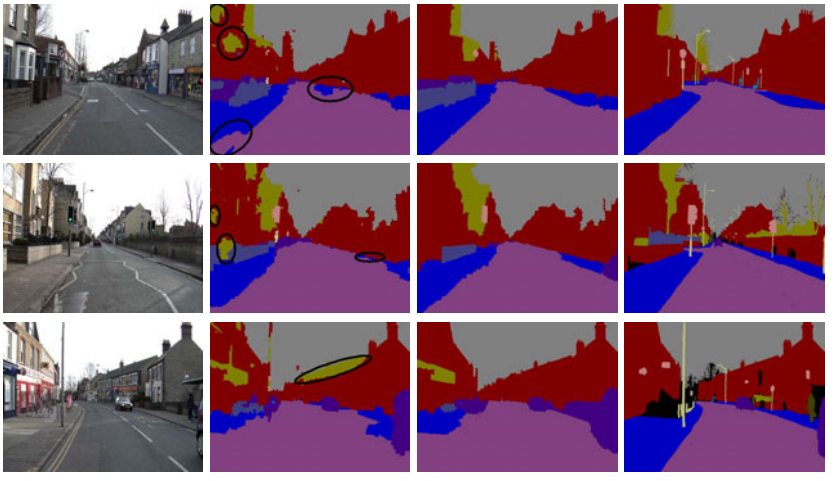


Fig. 6. Refinement of segmentation by multi-view fusion. From left to right: test image, result before fusion, result after fusion, ground truth. Based on temporal consistency of video sequence, some inaccurate classifications in reference view are refined by multi-view fusion, as circled in black.

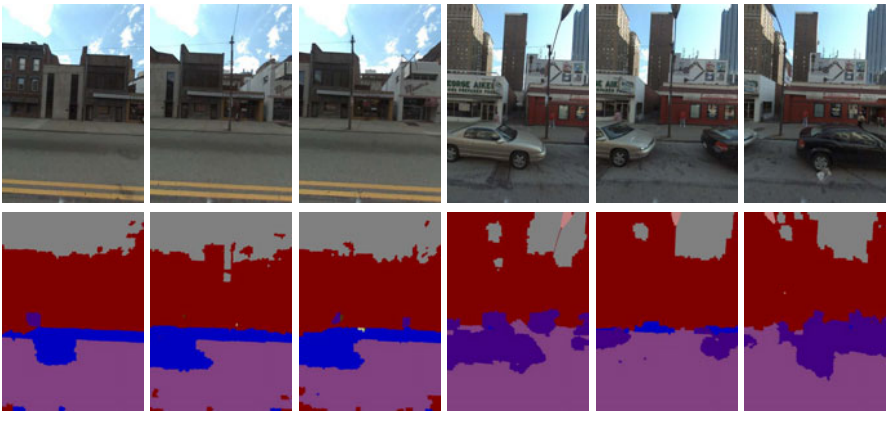


Fig. 7. Preliminary result of changed-view training and testing. From the side-view video sequence,we can see that our algorithm trained based on front-view sequences still obtained acceptable results.

Quantitatively, however, we found that the overall accuracy just improved by 1%. Further investigation in this topic is needed.

4.2 Cross Training Test

As mentioned above, we use per-pixel depth information which is independent of camera pose. It is reasonable to carry out a crossing training and testing

experiment between two types of camera configuration: front-view(used in [9]) and side-view cameras (used in [11]). We use CamVid(front-view) database as training images and Google Street View images(side-view) for testing. All the 3D features in both training and testing database are transferred to the unified coordinate system. In practice, it is necessary to apply a scale change to the testing data due to the scale ambiguity in structure from motion. We use a person's height as the reference. Our results are shown in Figure 7. It is visually comparable to the results from [11] in which both training and testing are performed with Google Street side-view images. We currently do not have access to ground truth labeled dataset, so a quantitative comparison is not available.

5 Conclusion

In this paper, we presented a novel framework for semantic segmentation and recognition based solely on dense depth maps. Our main contribution is that we have shown that dense depth maps generated by stereo contain plenty 3D information for scene parsing and can outperform segmentation using sparse 3D features or appearance features, or even the combination of both. Our method does not rely on any appearance information, making it robust against lighting changes. In addition our full 3D metric representation is independent of camera configurations, therefore we can use one set of front-view or side-view video for training and the other set for testing. This is not possible using appearance information dependent on camera poses and illuminations.

The accuracy of our approach depends on the quality of depth map. While we have applied state-of-the-art algorithms to calculate depth map, the quality of depth map is still quite fragile. Our next step is to apply laser range scan data that have significantly better accuracy and consistency. We expect to see large performance improvement with better input data. In addition, one future area of work is the bias against less frequently appearing objects, such as thin column poles. This is mainly due to lack of sufficient training examples, which naturally lead to a less statistically significant labeling for objects in these classes. One possibility to preserve the classification of less frequent object classes could be to include context information that may boost the significance of objects in certain cases.

Acknowledgements. The authors thank the anonymous reviewers and area chair for their constructive feedbacks. Thanks to University of Cambridge and Google for providing street view datasets. This project is sponsored in part by NSF grant HCC-0448185, CPA-0811647, and CNS-0923131.

References

1. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J.: Turbopixels: Fast superpixels using geometric flows. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(12), 2290–2297 (2009)
2. Russell, B.C., Torralba, A.: Labelme: a database and web-based tool for image. Int. J. of Computer Vision 77(1)

3. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2009)
4. Collins, R.T.: A space-sweep approach to true multi-image matching. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 358–365 (1996)
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(5), 603–619 (2002)
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. of Computer Vision 70(1) (October 2006)
7. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
8. Brostow, G., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition letters 20(2), 88–97 (2009)
9. Browtow, G., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
10. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2008)
11. Xiao, J., Quan, L.: Multiple view semantic segmentation for street view images. In: Proc. of Intl. Conf. on Computer Vision (2009)
12. Li, L., Li, F.: What, where and who? classifying events by scene and object recognition. In: Proc. of Intl. Conf. on Computer Vision (2007)
13. Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.M., Yang, R., Nister, D., Pollefeys, M.: Real-time visibility-based fusion of depth maps. In: Proc. of Intl. Conf. on Computer Vision (2007)
14. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning 36(1), 3–42 (2006)
15. Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. Int. J. of Computer Vision 78(2), 143–167 (2008)
16. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2005)
17. Yang, R., Pollefeys, M.: Multi-resolution real-time stereo on commodity graphics hardware. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2003)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
19. Zhang, G., Jia, J., Wang, T.T., Bao, H.: Recovering consistent video depth maps via bundle optimization. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2008)
20. Zhang, G., Qin, X., Hua, W., Wang, T.T., Heng, P.A., Bao, H.: Robust metric reconstruction from challenging video sequences. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2007),
`http://www.zjucvg.net/acts/acts.html`