Data-Driven Street Scene Layout Estimation for Distant Object Detection

Donghao Zhang
CECS,ANU
zdhpeter1991@gmail.com

Xuming He
NICTA & ANU
Xuming.He@nicta.com.au

Hanxi Li
NICTA & ANU
Hanxi.Li@nicta.com.au

***Abstract*—We present a street scene layout estimation method based on transferring layout annotation from a (large) image database and its application for distant object detection. Inspired by nonparametric scene labeling approaches, we estimate a scene's geometric layout by matching global image descriptors and retrieving the most similar layout configuration. Our label transfer is done for each sub-region of an image and a tiered scene model is used to integrate all the local label information into a coherent scene layout prediction. Given the geometric layout, we use a super-resolution method to zoom in the distance region and refine the search in object detection. On KITTI dataset, we show that we can reliably generate scene layout and improve the detection of distant cars over the state of the art DPM detector.**

I. Introduction

Geometric scene parsing, in which we segment a single image into regions with pre-defined coherent geometric properties, has been an important task in scene understanding [9]. The parsing output provides a coarse-level description of the geometric layout of the scene depicted in the image. This layout information can be utilized to generate additional image cues for other recognition tasks, and has been shown beneficial effects on semantic scene labeling [6] and object detection [10].

However, most existing approaches require segmenting the input images into superpixels and training a superpixel classifier to predict the local layout property. Despite that a global model (i.e., random fields) is used to integrate the local information and enforce the contextual consistency in a later stage, the prediction solely from local regions is a difficult task due to the ambiguity in image appearance. In addition, this requires sufficient training data to cover different scenarios.

In this work, we propose a simple layout estimation method based on the tiered scene model [4] and label transfer. Our main idea is built on two insights: first, the local layout predication can become easier if we work on a larger region designed to reflect the overall scene structure; second, we can avoid training local classifiers by transferring layout labels from similar images [12], [14], [15].

Specifically, we focus on the outdoor street scene images taken by a facing-forward car-mounted camera in this paper, as shown in Figure 1. We make use of the regularity of street scene to divide the image into uniformly-spanned columns and matching those image columns based on image and (noisy) geometric context similarity. We build a one-dimensional CRF model on the layout boundary segments and use an efficient dynamic programming to search the best configuration of overall layout of the scene.

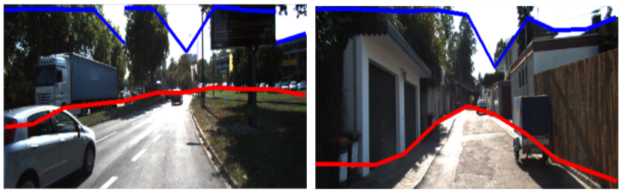

Fig. 1. Examples of street scene layout from KITTI dataset. The blue line is the boundary between 'sky' and 'vertical' regions, and the red line is the boundary between 'vertical' and 'ground' regions.

To demonstrate utility of the predicted scene layouts, we use the layout information to address the challenges in detecting distant objects, which are too small and lack details. We take the layout as a guidance to choose a distance region in the image and apply a super-resolution method [17] to recover the details of the region. We then enhance the process of object detection by re-applying an object detector to the zoomed-in regions. We evaluate our method on a subset of KITTI dataset [5], and the results show our method achieves the state of the art layout prediction performance and object detection results.

II. Related Work

Scene understanding and object detection have attracted much attention in recent years and there is a large number of literatures. We will focus on only the most related works here. We would refer the interested readers to the survey [11] and for more detailed discussion.

The geometric layout estimation from a single image is discussed in [9], where a superpixel based CRF model is used to infer the local orientation of object surfaces. More recently, Felzenszwalb et al. propose a tiered scene model and an efficient dynamic programming based inference algorithm [4]. Other approaches use more advanced geometric primitives, such as 3D blocks [8], or focus on building class [16], for layout reasoning but has to resort to more complex inference procedure. Unlike those methods, our method has a simpler representation and also an efficient inference procedure.

Our data driven method is inspired by the recent nonparametric label transfer approaches [15]. However, we do not use superpixels [3], nor do we deform local patches to match target images [12]. Our method is based on larger image regions for retrieving similar image elements, which exploits the tiered structure of street scenes.

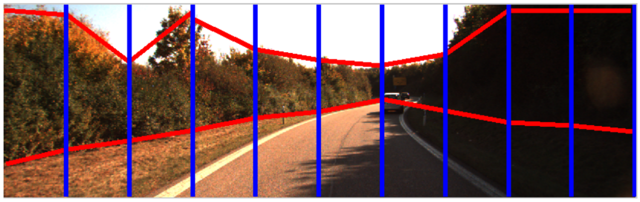

Fig. 2. The column-based tiered scene model. We use ten columns and within each column, the boundaries between different regions are linear segments.

Geometric information has been explored before for improving monocular object detection [10]. In most cases, the layout is used to constrain the search for a specific class based on the common geometric or support relationship. Our method, however, uses the geometric reasoning to identify the distant area and enhances those regions so that small-sized objects can be successfully detected.

III. Our approach

We will first introduce our scene model for geometric layout estimation, which is based on a one-dimensional CRF model. We then present our nonparametric label transfer for the local layout prediction and a joint inference to estimate the layout for the full scene. Given the scene layout, we propose a zoom-based method to improve the detection of objects at small scales.

A. Column-based tiered scene model

Given an image, we aim to infer a coarse layout of the underlying scene. Specifically, we have a set of three main geometric labels, including $\{sky, vertical, ground\}$, and our task is to assign these labels to every pixel in the image. To this end, we design a scene layout model with a tiered structure [4] and piece-wise linear boundaries between regions (See Figure 2). To be more specific, we partition the image plane into K uniformly spanned columns $\mathcal{P} = \{P_k\}$ and each column is segmented into three regions by two line segments L_k^t and L_k^b. We assign the top, middle and bottom region by the label sky, $vertical$ and $ground$, respectively.

Mathematically, we consider modeling the boundary segments between two neighboring labeled regions in the entire image. For clarity, we only describe the top boundary between sky and $vertical$ and the other boundaries are similar. As the x positions of two ends of each line segment L_k are fixed, we denote the line segment as $l_k = (y_{k,1}, y_{k,2})$. We build the following CRF on the boundary segments:

$$E(\mathbf{L}, I) = \sum_{k=1}^{K} \phi(l_k, I) + \sum_{k=1}^{K-1} \psi(l_k, l_{k+1}, I), \qquad (1)$$

where ϕ is the unary term to capture the data consistency, and ψ is the pairwise term for the smoothness of the boundary. The main novelty of our method is to use data-driven method to compute the unary term, which will be described in the following.

B. Data-driven layout estimation

We want to transfer labeling information from a dataset of images annotated with the coarse-level layout information. Our goal is to generate the unary term for the column-based tiered scene model by nearest neighbor search. We adopt a two-step strategy similar to [15]. We first describe how we find similar images in the dataset.

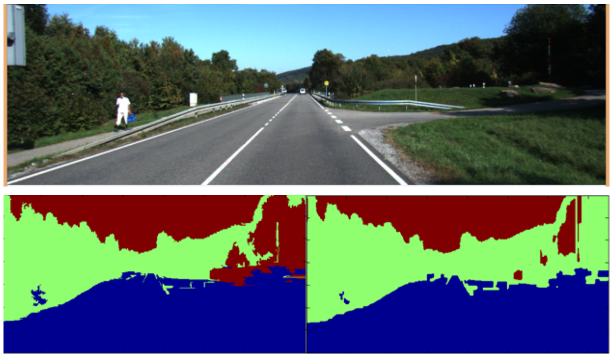

Fig. 3. Input image (Top), original (Bottom left) and improved geometric context (Bottom right) feature.

1) Image candidate retrieval: We follow the nonparametric scene parsing approach [15] and first retrieve a small set of images that are similar to the target image. We use two feature descriptors, one is the pyramid HOG (PHOG) [1] and the other the geometric context (GC) feature [9]. For PHOG, we use the Euclidean distance and for GC, we use the Hamming distance.

As the initial Geometric Context feature is noisy, we propose a smoothing procedure to obtain a cleaner version of the GC feature. Specifically, our procedure includes the following steps.

- If there is a patch whose area is less than 50, then this area should be reallocated.
- Ground region can only border vertical region; In other words, sky region and ground region are not neighbors
- Due to position of camera, the angle view of camera will be within a range. The ground region will take up at one-sixth at every y value.
- Most of discontinuities of segmentation lines are caused by the sudden change of trees and buildings. Other causes are rare phenomenon.

Figure 3 shows an example of original and improved Geometric Context. We can see that our version is less noisy. Figure 4 shows two top candidates from the PHOG feature matching and GC feature matching alone.

We explore each feature individually and find these two types of features are complementary in retrieving similar image candidates. The PHOG focuses more on the shape of the scene layout and the GC encodes regional properties. We combine both features to search the initial image candidates. The distance between the target image I^r and an annotated image I^d in the dataset is defined as

$$D(I^r, I^d) = \|f^r_{PHOG} - f^t_{PHOG}\|_{l_2} + w_d \|f^r_{GC} - f^t_{GC}\|_{l_{Hamm}} \qquad (2)$$

where w_d is the weight between two features.

Based on the distance in Equation (2), we retrieve M nearest neighbors from the annotated dataset for the target

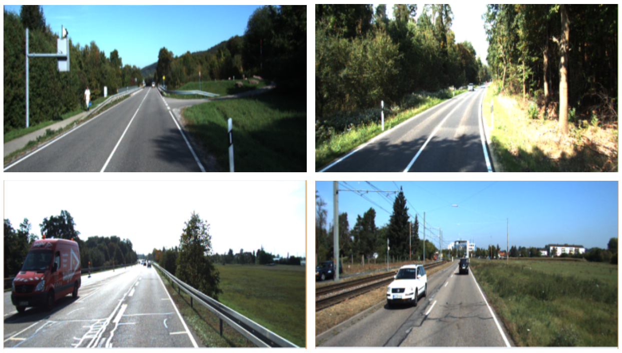

Fig. 4. Top two matches based on the PHOG feature (Top row) and improved geometric context features (Bottom row). The target image is the same as in Fig 3.

image, and denote them as $\mathcal{C}$, i.e., candidate set for next step. In this work, we use a linear search method while more advanced approaches based on approximate nearest neighbor (ANN) [13] can be used to improve the efficiency.

2) From similar image to similar column: Unlike [15], we exploit the tiered structure of the street scene and represent the entire scene as a set of columns instead of superpixels, as in Section III-A. In the second step, we retrieve the most similar columns from the candidate set $\mathcal{C}$ for each column in the target image. The allows us to use more fine-level annotation in the training set and accommodate variations in the data.

In this work, we set $K = 10$, which is small enough to find sufficiently similar neighbors in terms of annotation. We also use PHOG and GC features to retrieve similar columns from the candidate set. An example of a single column and its five nearest neighbors are shown in Figure 5. We can see that they are very similar in their layouts.

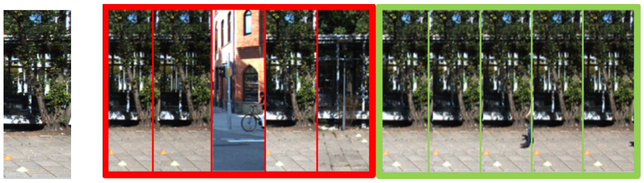

Fig. 5. Left: Image column for retrieval.Right: Top five matches of columns based on improved geometric context features (Red) and PHOG (Green).

To build the unary term in Equation (1), we find five nearest neighbors for each column P_k based on PHOG feature and another five based on GC feature. In total, we have 10 candidate columns for P_k and we use them as the state space for the unary term $\phi(l_k, I)$. In other words, we would like to find the best annotation from those ten configurations. We will use the global CRF model to find the most likely joint configuration for the whole image.

To define the unary term, we rank the ten candidates based on their matching scores. However, due to two different types of features we use, we need to adjust their original ranking. We use a small validation set (30 images) to find a global re-ranking for these 10 candidates. Specifically, we compute their performance in terms of the average error of the retrieved annotation $S = \frac{1}{30}\sum_i \|l_k^r - l_k^d\|^2$, where l_k^r is the target column and l_i^d is the retrieved column. Then we re-rank them according to their average performance. The detailed error cost and re-ranking is shown in Table I. Note that the original ranking is instance specific while the re-ranking is applied globally across the dataset. Given the ranking, we define the unary term as follows [7]

TABLE I. RE-RANKING OF TEN CANDIDATES

Avg error	1098.1	1278.8	1285.5	1327.1	1495.7
Re-rank	1	2	3	4	5
Orig rank	1	2	3	4	5
Avg error	1279.3	1619.6	1671.6	1801.8	1720.4
Re-rank	5	7	8	10	9
Orig rank	6	7	8	9	10

$$\phi(l_i = k, I) = \alpha \log(r_k), \tag{3}$$

where r_k is the ranking of the kth candidate.

C. Consistent global layout estimation

We define a pairwise potential function $\psi(l_i, l_j, I)$ to impose the smoothness constraint on two consecutive columns. Specifically, we prefer a smooth transition between the right end of l_i and the left end of l_j. In addition, the smoothing should be modulated by the edge strength between line segments as shown in Figure 6. Let the edge strength be e_{ij}, the pairwise term can be written as

$$\psi(l_i, l_j, I) = \beta(e_{ij})\|y_{i,2} - y_{j,1}\|^2, \tag{4}$$

where $\beta(e_{ij})$ is a function of the edge strength. We define $\beta(e_{ij}) = 1$ if $e_{ij} < \tau$, and $\beta(e_{ij}) = \exp(-e_{ij}) + \gamma$ otherwise. γ is a bias term.

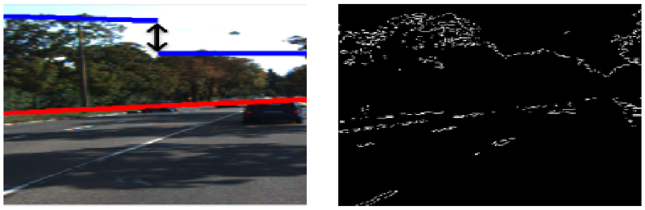

Fig. 6. Left: Discontinuity between two consecutive column is shown by the black arrow; Right: Sobel edge response of the left image and thresh value is set to 0.49.

To estimate the optimal joint configuration of $\mathbf{L} = \{l_k\}$ for all the columns $P_1, \cdots, P_K$, we compute the MAP estimation of the CRF in Equation (1):

$$\begin{aligned}\mathbf{L}^* &= \arg\min_{\mathbf{L}} E(\mathbf{L}, I) \\ &= \arg\min_{l_1,\cdots,l_K} \sum_{k=1}^{K} \phi(l_k, I) + \sum_{k=1}^{K-1} \psi(l_k, l_{k+1}, I)\end{aligned} \tag{5}$$

As our CRF has a chain structure, we use Dynamic Programming to compute the global minimum of the energy function.

D. Layout guided object detection

The coarse-level scene layout provides useful information for scene understanding and object detection. The region layout label can eliminate unreasonable false positive detection of a specific class, such as car, and guide more efficient and thorough search in certain regions. For example, cars and pedestrians normally are on the ground; therefore the locations

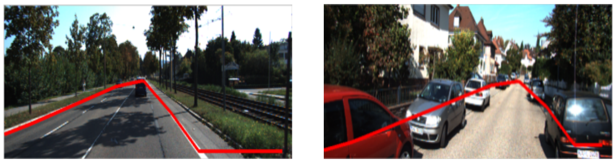

Fig. 7. The ground truth of the road surface from our annotation.

of cars and pedestrian should not be far from the ground region. This would also improve the efficiency of all kind of detections because classifiers can target on the particular region rather than whole image.

In this work, we use car as an example to show how the layout can be used to improve the detection of distant objects. To improve car detection, we further introduce road surface as another layer of layout information and implement a similar label transfer procedure as described in the previous sections. Figure 7 shows some examples of road annotation.

Once we obtain the layout and road labeling of an image, our method exploits the scene geometry to enhance the image region that is far away from the camera based on super-resolution before applying the object detectors. It would be very difficult to estimate the depth information of objects with a single image. We assume the road boundaries are roughly parallel and define a vanishing point in camera's optical axis direction. Given the location of vanishing point, we select a rectangular window centered at the vanishing point and with $1/3$ of the image height. The width of the window depends on the two boundaries of road surface.

Essentially, we recover partial depth information from the scene layout and selectively choose the distant region to expand our search for car objects. While simple interpolation can provide zoomed view of the region, we found it has certain artefacts and may blurs object features. Instead, we then apply a super-resolution method [17] to zoom this selected region to three times larger than its original size. Figure 8 shows an example of this pipeline. We can see that the super-resolution step removes some artefacts and the zoomed image has more fine-level gradient than the original. In the zoomed image, we apply a pre-trained DPM detector to search the object at a finer scale. Note that we also keep the detection results at the original resolution and merge them together for the detections on the entire image.

IV. Experiments

A. Dataset and Setup

To evaluate our method, we build a dataset with our own annotations from the KITTI dataset. The KITTI dataset consists of images captured from the two high resolution cameras installed on the car roof [5]. This dataset includes scenes from CBD, rural areas and highways. We use a simple random sampling method to get a representative subset from the whole KITTI dataset.

We choose 200 images and use a random subset of 100 images for training and the rest of 100 images for testing. To annotate the dataset, we divide each image into ten columns, and use line segments to separate the sky, vertical and ground, as well as the road within each column. We use grid search to determine all the parameters α, τ and γ in our models on a validation set. On the detection task, we compare our method with the DPM car detector on the original image. We use pre-trained DPM detector on the PASCAL dataset [2].

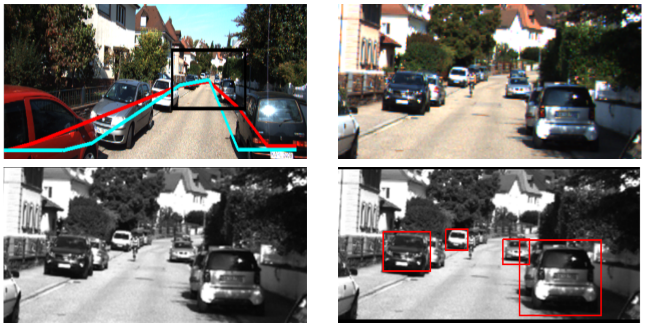

Fig. 8. Top Left: Comparison between the ground truth (red) and transferred annotation (cyan). The black rectangle is the selected region for zooming. Top Right: The image window extracted from the black rectangle before super-resolution. Bottom Left: The image window after super-resolution (better viewed on screen). Bottom Right: The result of car detection on image patches after super-resolution.

B. Layout estimation results

We evaluate the coarse-level scene layout performance on three classes, 'Sky', 'Vertical', and 'Ground'. Our method is compared with the Geometric Context [9] as a baseline. Table II shows the quantitative results on the test set. We compute the pixel-level accuracy of the three classes on each image and report the mean and standard deviation on the whole test set. We can see that our method achieves better performance on both measures. The potential reason is that we impose stronger constraints on the layout, which can remove many incorrect pixel labeling at superpixel level.

TABLE II. Comparison on the per-image average accuracy of the layout estimation.

Method	Region	Avg Accu	std
GC	Sky	51.4	31.0
	Vert	83.9	14.7
	Gnd	77.1	11.1
Our	Sky	76.9	14.9
	Vert	85.9	11.5
	Gnd	91.6	10.6

In addition, we show the confusion matrix of those two methods in Table III. Again, we can see from the table that our method can achieve much better pixel-level performance on the entire dataset. We should some examples of both scene layout and road estimation in Figure 9 and Figure 10. We find that our method can recover the layout with good accuracy in most cases, although it may oversmooth the boundaries between different classes. We can also see that the road and layout provide useful depth cues for identifying the distant regions relevant for car detection.

TABLE III. Comparison on the overall confusion matrix of the layout estimation.

GC	Sky	Vert	Gnd
Sky	64.7	35.3	0.0
Vert	13.0	84.1	2.9
Gnd	2.4	20.0	77.5

Our	Sky	Vert	Gnd
Sky	78.3	21.7	0.0
Vert	6.0	86.9	7.1
Gnd	0.0	7.5	92.5

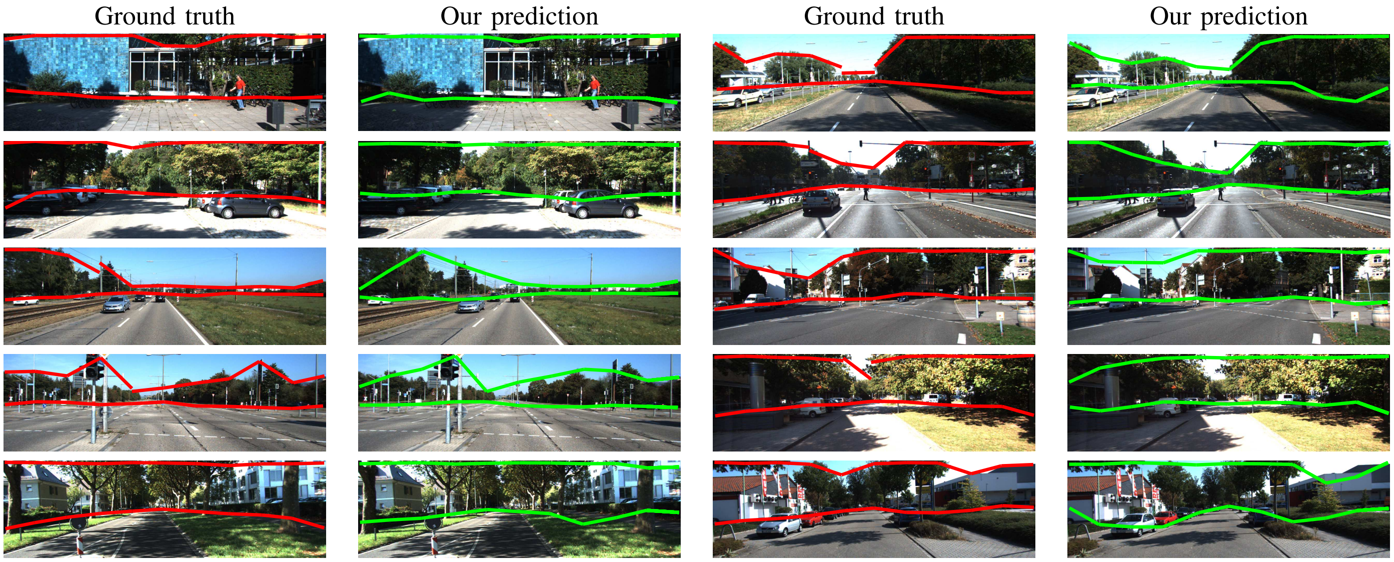

Fig. 9. Comparison between the ground truth and transferred layout. Red lines and green lines are the ground truth and transferred annotations respectively.

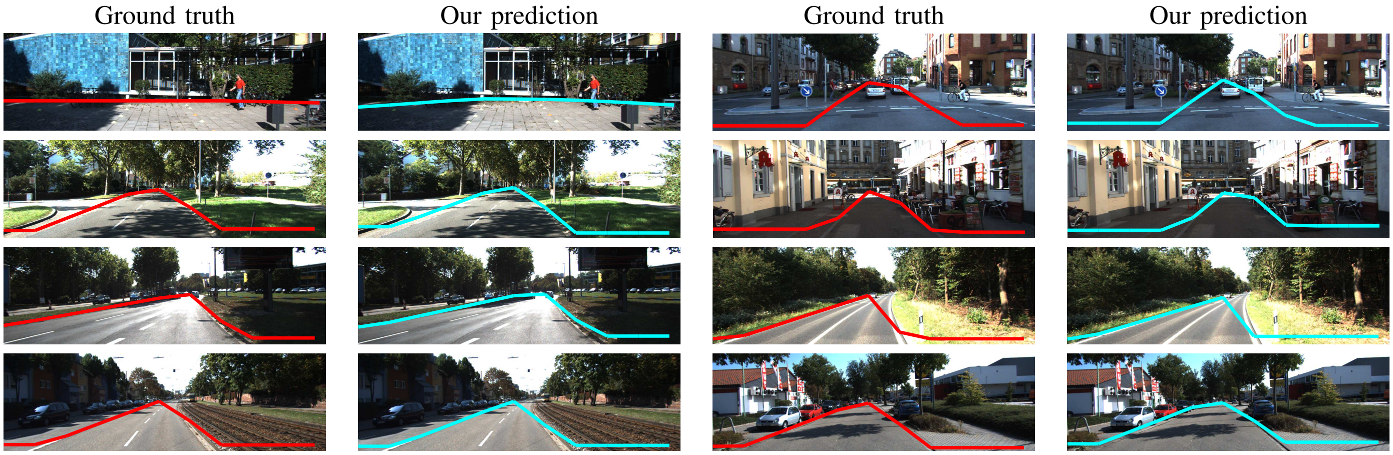

Fig. 10. Comparison between the ground truth and transferred road label. Red lines and cyan lines are the ground truth and transferred annotations respectively.

C. *Car detection results*

We use the pre-trained DPM detector and set the threshold to -0.62 to report the results in this work, which achieves the best F-measure. The image window has size of 1200x560 after super-resolution. We show the number of true positives (TP), false positives (FP), precision, recall and F-measure of the baseline method and our method in Table IV. We can see that our method achieves better performance overall. The true positive of detected cars is approximately 1.5 times as many as the original PASCAL model. The increase of F-score indicates the improvement compared to the original PASCAL model. While our precision slightly lower, our recall rate is much higher than the baseline.

We also display some qualitative results in Figure 11. As shown in the figure, the DPM classifier is able to detect cars close to the camera, but unable to find any instance that is far from the camera. The main reason is that it is hard to detect small-scale car from the original image due to lack of gradient information or constraints from the pre-trained model. The super-resolution step provide sufficient details to enable us to achieve better detection performance. The scene layout makes it possible to select the distant regions to analyze.

TABLE IV. COMPARISON OF CAR DETECTION PERFORMANCE.

Method	TP	FP	Precision	Recall	F-score
DPM	165	13	0.927	0.267	0.415
Ours	230	25	0.902	0.373	0.528

V. CONCLUSION

In this work, we present a data-driven approach to street scene layout estimation and its application in detection of distant objects. Our street layout model is based a simplified column-like tiered scene model and we use retrieval-based method to define the unary term of the corresponding CRF model. We show that our approach can achieve better performance than the state of the art. In addition, we infer the scene geometry based on our layout estimation and find the image region that is far away from the camera. We apply a super-resolution step to zoom into those regions so that the object detector can successfully find distant objects at small scales on

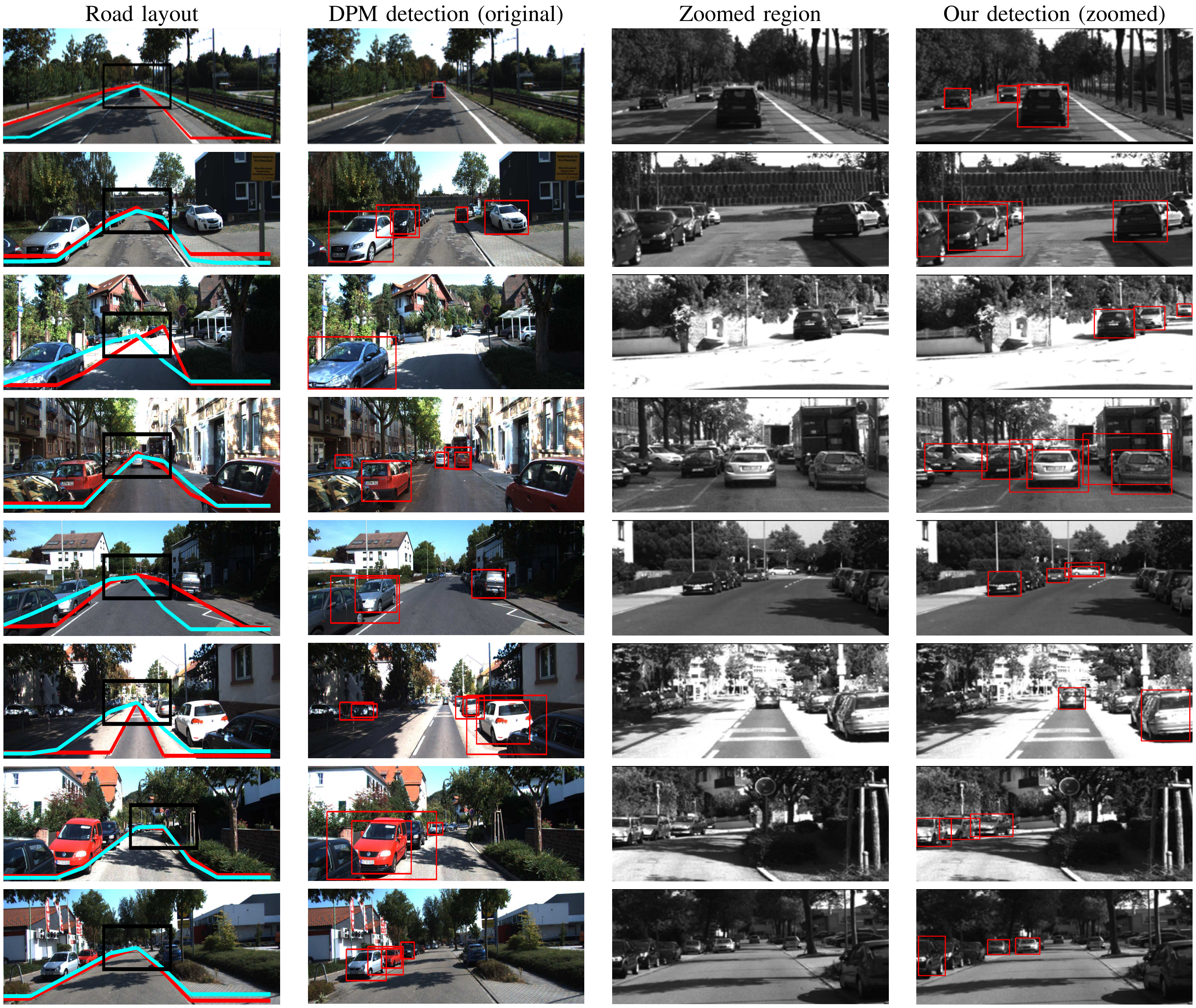

Fig. 11. Comparison between the detection from DPM and our method. First column: Road and distant region estimation. Second column: DPM detection results. Third column: distant region after super-resolution. Fourth column: our detection results.

the original image. We show that our improved detector has a precision-recall rate superior to the DPM.

Acknowledgment

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications, and the Digital Economy, and the Australian Research Council (ARC) through the ICT Centre of Excellence Program. This research was also supported in part by ARC through its Special Research Initiative (SRI) in Bionic Vision Science and Technology grant to Bionic Vision Australia (BVA).

Reference

[1] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *ACM ICIMR*, 2007.

[2] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[3] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 2004.

[4] Pedro F Felzenszwalb and Olga Veksler. Tiered scene labeling with dynamic programming. In *CVPR*, 2010.

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[6] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[7] Stephen Gould and Yuhang Zhang. Patchmatchgraph: Building a graph of dense patch correspondences for label transfer. In *ECCV*. 2012.

[8] Abhinav Gupta, Alexei A Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*. 2010.

[9] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, 2005.

[10] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 2008.

[11] Derek Hoiem and Silvio Savarese. *Representations and Techniques for*

3D Object Recognition and Scene Interpretation. Morgan & Claypool Publishers, 2011.

[12] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing via label transfer. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011.

[13] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2014.

[14] Bryan Russell, Alyosha Efros, Josef Sivic, Bill Freeman, and Andrew Zisserman. Segmenting scenes by matching image composites. In *Advances in Neural Information Processing Systems*, 2009.

[15] Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*. 2010.

[16] Rashmi Tonge, Subhransu Maji, and CV Jawahar. Parsing worlds skylines using shape-constrained mrfs. In *CVPR*, 2014.

[17] S. Villena, M. Vega, D. Babacan, J. Mateos R. Molina, and A. K. Katsaggelos. Superresolution software. http://decsai.ugr.es/pi/superresolution/software.html, 2011.