Li *et al. EURASIP Journal on Image and Video Processing* (2018) 2018:15
https://doi.org/10.1186/s13640-018-0254-1

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Hierarchical semantic segmentation of image scene with object labeling

Qing Li*, Aihua Liang and Hongzhe Liu

Abstract

Semantic segmentation of an image scene provides semantic information of image regions while less information of objects. In this paper, we propose a method of hierarchical semantic segmentation, including scene level and object level, which aims at labeling both scene regions and objects in an image. In the scene level, we use a feature-based MRF model to recognize the scene categories. The raw probability for each category is predicted via a one-vs-all classification mode. The features and raw probability of superpixels are embedded into the MRF model. With the graph-cut inference, we get the raw scene-level labeling result. In the object level, we use a constraint-based geodesic propagation to get object segmentation. The category and appearance features are utilized as the prior constraints to guide the direction of object label propagation. In this hierarchical model, the scene-level labeling and the object-level labeling have a mutual relationship, which regions and objects are optimized interactively. The experimental results on two datasets show the well performance of our method.

Keywords: Semantic labeling, Object labeling, Semantic segmentation

1 Introduction

Semantic segmentation is a fundamental task in computer vision, which is a basic work for many applications, such as image editing, image-based modeling and autonomous driving [1–3]. The typical approaches of semantic segmentation include the parametric ones [4–8] and the nonparametric ones [9–13], which both achieve promising performance. Previous works focus on assigning a unique category label to each pixel correctly, generating region segments with semantic information. However, these segments have little information of objects in the scene. Specifically, all the objects from the same category are considered as a whole *object*, thus making it difficult to distinguish different instances. For clarity, we use the same *object* definition as that used in [14, 15], in order to differentiate from *material*. The objects are better characterized by overall shape than local appearance, while *material* categories have no consistent shape but fairly consistent texture.

Besides the typical semantic segmentation approaches, many approaches aim at the accuracy improvement of object segment, such as the interactive segmentation [16–18] and the co-segmentation [19–21]. In these works, the prior of latent object is provided with either the user scribbles or the object coherency to generate exact segments, while these segments usually have no semantic information of objects.

In fact, the details of objects are useful for precise understanding of the image scene. For example, in Fig. 1, the scene-level semantic labeling tells us that this image shows person and horse in a natural scene. The object-level semantic labeling gives us more details about the scene, such as the numbers of person and horse, and the layouts of each person and horse. With these details, we can even infer that this scene may be a snapshot of a polo game. Therefore, object details are effective for precise scene parsing, and should be predicted accurately. Considering the diversity of objects in texture, shape and pose, object labeling is still a challenge problem, though there have been several approaches dealing with this problem [1, 15, 22, 23].

In this paper, we propose a hierarchical semantic segmentation method which aims at understanding both scene regions and objects. Under the assumption that the scene-level and the object-level labeling are stimulative for each other, we first get a scene-level labeling via a

*Correspondence: liqing10@buu.edu.cn
Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China

Fig. 1 Motivation. From left to right, the initial image, the motivation of scene-level labeling, and the motivation of object-level labeling. In the scene level, different colors indicate different categories. In the object level, different colors indicate different objects

feature-based MRF, and then utilize the category prior as well as the appearance features to improve the labeling of object. We give a definition of object labeling which is similar to that of semantic labeling, i.e., assigning a unique object label to each pixel. Then a constraint-based geodesic propagation algorithm is proposed to achieve object segments.

The main contributions of this paper include the following: (1) A multi-level semantic labeling framework is proposed for understanding of both regions and objects; (2) A constraint-based geodesic propagation algorithm is introduced for object labeling. The rest of this paper is organized as follows. A precise overview is described in Section 2.1. Then Sections 2.2 and 2.3 give the details of scene-level and object-level labeling respectively. We show our experimental results in Section 3 and give a brief conclusion in Section 4.

2 Methods

2.1 Hierarchical model overview

The framework of our hierarchical model includes scene-level labeling and object-level labeling. The overview is illustrated in Fig. 2.

In the scene-level labeling, we use a feature-based MRF model to recognize categories in a scene. To make the labeling more efficiently, we over-segment the image into a set of superpixels using turbopixel algorithm [24]. On the over-segmented image, pixel-wise features are mapped into a feature vector of the corresponding superpixel, including filter responses, boundary features, pyramids of HOG, and RGB colors. We utilize a one-vs-all classification mode to get the raw probability for each category. The raw probability and features are embedded into the MRF model as unary potential and binary potential respectively. With the graph-cut inference, we get the raw scene-level labeling result. Besides, we conduct object detection with SVM algorithm, predicting object candidates. The number of instances is identified based on the raw probability and object candidates. The final scene-level labeling is adjusted with the object-level labeling, generating a more precise scene-level result.

In the object-level labeling, we conduct saliency detection to get the saliency map. The region of interest (ROI) for objects is obtained based on the saliency map and the raw probability map. A graph model is formulated over this ROI, of which a node denotes one superpixel and an edge denotes the adjacency of superpixels. The weights on edges are computed from multi-dimension features of each superpixel, including the HOG descriptor, texture descriptor, Lab colors, and gradient features. These features are different from those used for scene-level labeling. The weights on nodes consist of the saliency confidence and the raw probability, which are mapped into geodesic distance. We conduct geodesic propagation on the graph model. In each step of propagation, a node with the smallest geodesic distance is selected as the seed node. We fix the label of this seed and update the status of its neighbors for next propagation step. When all the nodes are fixed, we get the object labeling result.

2.2 Scene labeling

2.2.1 Features for recognition

We utilize a one-vs-all mode to train the classifier for each category with dense samples from the training images. The feature vector $fv(i)$ of superpixel i consists of filter response features [4], boundary features [25], pyramids of HOG [26], and RGB color features. The filter response, boundary and color features are sampled for each pixel, and mapped to superpixel level by averaging pixels over each superpixel. The pyramids of HOG are computed over a patch region of each superpixel. In testing, the classifiers learned by the Joint Boosting algorithm [4] generate the initial raw probabilities of each category l for i, which is denoted as $P(i, l)$.

2.2.2 MRF model

The objective of scene-level labeling is to assign each superpixel i a category label l_i from fixed category label set L. The energy $E(L)$ of our MRF model over all superpixels is defined as follows:

$$E(L) = \sum_i \psi(l_i) + \lambda_\Phi \sum_{(i,j)} \phi_{ij}(l_i, l_j), \quad (1)$$

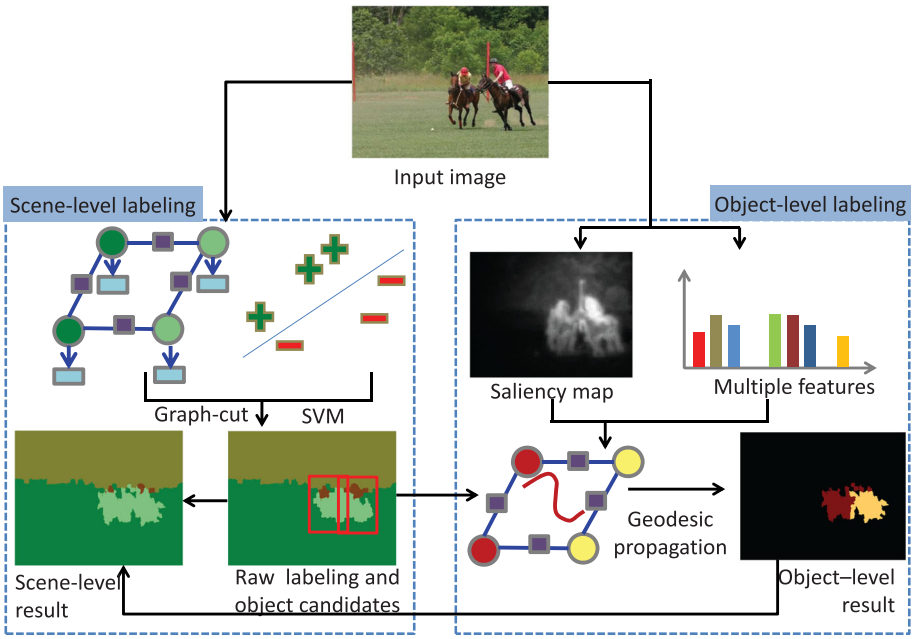

Fig. 2 Overview of our framework. See the text for more details

The unary term $\psi(l_i)$ measures the cost of assigning label l_i to superpixel i , and the pairwise term $\phi_{ij}(l_i, l_j)$ measures the penalty of different assignment between similar adjacent superpixels i and j. Our unary term is an exponential form of the normalized raw probabilities, i.e. $\psi(l_i) = exp(-P(i,l))$. Our binary term is computed as $exp(-z * (\|fv(i) - fv(j)\|)^2)$,where z is a normalization parameter.

We use graph-cut algorithm [27] to get the scene-level labeling result. The region of *object* category can be refined with the following object-level labeling.

2.2.3 Recognition for objects

It is a challenge to identify object number accurately in the scene. We first perform object detection to get a raw estimation of object candidates. The detectors are learned by SVM algorithm, proposing a set of object hypothesis $\{H\}$, in which each $h \in \{H\}$ has a bounding box and a score. Then, we rank these hypotheses according to their scores and prune the hypotheses whose scores are lower than threshold T_h. The rest hypotheses in the pruned $\{H\}$ are the object candidates that need to be labeled. In our implementation, T_h is learned on the training images with their bounding boxes. We estimate the score distribution with 95% confidence interval. Theoretically, T_h should be the value which meets over 95% true positive predictions, i.e., over 95% bounding boxes whose scores are higher than T_h are true positive. Considering the outliers of bounding box in testing, however, we broaden this restriction and actually select T_h as the value which meets over 85% true positive predictions, expecting to reduce the false negative predictions.

In practice, when T_h can not work well for a specific image, for example, if the pruned $\{H\}$ has no object candidates or much more object candidates than common amount, then we adjust the number of candidates experimentally.

2.3 Object labeling

The objective of object-level labeling is to assign each pixel a unique object label. The scene-level labeling gives a rough region of *object* category. We perform object labeling on such region instead of the whole image. In this section, we start with how our ROI region is identified.

2.3.1 ROI region

Objects in an image scene usually attract more attentions of a human being than materials; therefore, we assume that saliency detection may predict valid object region. Some approaches have utilized saliency information for object segmentation [28, 29]. In our implementation, we utilize the algorithm of Goferman et al. [28] to get a down-sampling output saliency map, and then up-sampling output to the original size of an image. An example of saliency map is shown in Fig. 3.

For a specific category C, the ROI region of its instances is supposed to include (1) superpixels whose probabilities of C are higher than other categories, (2) superpixels whose probabilities of C are higher than a threshold T_p, and (3) superpixels whose saliency values are higher than

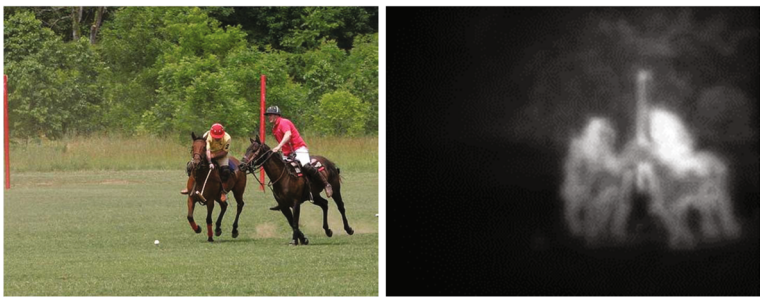

Fig. 3 Saliency map. The brighter the region is, the higher object probability it has

a threshold T_s. The thresholds T_p and T_s are estimated in the similar way to T_h. We estimate the distributions of raw probability and saliency respectively on training images. Then the values which meet over 85% superpixels are selected as T_p and T_s respectively.

2.3.2 Feature weights on graph

Once the ROI is identified, a graph model is then formulated, where each node denotes one superpixel in the ROI and each edge denotes the adjacency of superpixels. The weights on graph consist of appearance features, including the HOG descriptor, texture descriptor, Lab colors, and gradient features. The first three types of features are embedded in the weights of nodes, and the gradient features are embedded in the weights of edges. These features are different from the *fv* used for scene labeling. We observe experimentally that, for object labeling, color features in Lab space perform better than that in RGB space. We leverage these features as the prior constraints. Each kind of feature is described in a bag of words style, as did in [30]. A pixel-level HOG spatial pyramid is constructed with 8×8 blocks, 4 pixel step size, and 2 scales per octave. These HOG features are concatenated into a one-dimensional vector. We cluster these features to 1000 kmeans centers, resulting in a HOG descriptor. The pixel-level texture features are extracted with a Gaussian filterbank and quantized to nearest 256 kmeans centers. The histogram of 256 bins is used as the texture descriptor. In Lab color space, the color features are densely sampled and quantized to the nearest 128 kmeans centers. The gradient features which reflect the boundaries of objects are used as propagation constraint, including both horizontal and vertical gradients. All these pixel-level features are mapped to superpixel level. The HOG, texture and color features are encoded as the weight difference $D(i,j)$ in a linear combination, as shown in Eq. 2.

$$\begin{aligned} D(i,j) = {} & \lambda_1 * \|F_{hog}(i) - F_{hog}(j)\| \\ & + \lambda_2 * \|F_{tex}(i) - F_{tex}(j)\| \\ & + \lambda_3 * \|F_{color}(i) - F_{color}(j)\| \end{aligned} \tag{2}$$

where i and j denote the adjacent nodes, F_{hog}, F_{tex}, and F_{color} indicate the HOG, texture, and color features. In our implementation, we set λ_1 0.1, λ_2 0.3, and λ_3 0.6 experimentally.

2.3.3 Geodesic propagation

The geodesic propagation is considered to be valid for semantic labeling [12, 31]. We follow the definition of geodesic distance in [12, 31], while modify the details of implementation.

In the graph model for geodesic propagation, the weight on node is computed as the geodesic distance, and the weight on edge is the difference cost. The object labels propagate iteratively throughout all the superpixels in the ROI. If the raw probability of node i for category C is

Fig. 4 Annotations of Polo. **a** The initial image, **b** the groundtruth labeling of categories, **c** the groundtruth labeling of instances, and **d** the bounding boxes of instances. Best viewed in color

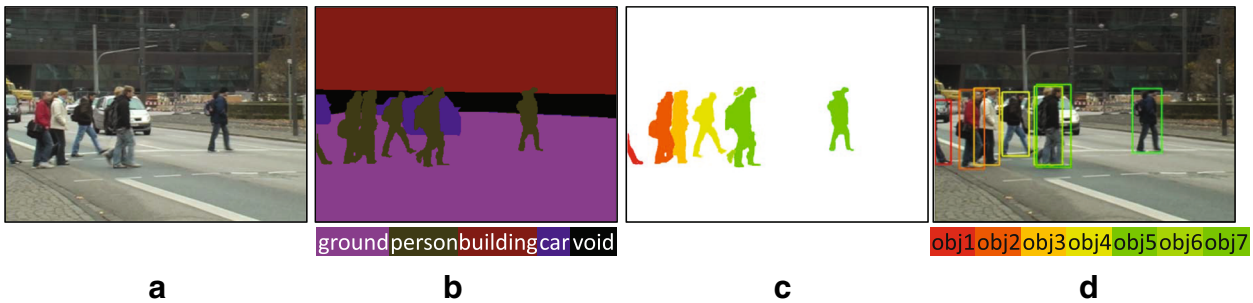

Fig. 5 Annotations of TUD. **a** The initial image, **b** the groundtruth labeling of categories, **c** the groundtruth labeling of instances, and **d** the bounding boxes of instances. Best viewed in color

higher than T_p and the saliency value of i is higher than T_s as well, The weight on node i consists of the probability, the saliency and the bounding box score. Otherwise, the weight on node i consists of the probability of other category and the negation value of saliency, as shown in Eq. 3.

$$Op(i,o) = \begin{cases} ib(i,o) * B(o) + P(i,C) + S(i), & \Delta_w \\ (1 - P(i,C)) + (1 - S(i)), & else \end{cases} \quad (3)$$

$$s.t.\ \Delta_w : P(i,C) > T_p \cap S(i) > T_s$$

where $Op(i,o)$ denotes the probability of each node i belong to the object o, $P(i,C)$ is raw probability of category C, $S(i)$ is the saliency value of i. $ib(i,o)$ indicates whether i is inside of the scope of o, $ib(i,o) \in \{0,1\}$. $B(o)$ is the detected bounding box score of o.

The weights on nodes are normalized and converted to initial geodesic distances, as shown in Eq. 4. The geodesic distances have inverse proportion to weights, i.e., a node with a higher weight has a shorter distance.

$$geoDis(i,o) = exp(1 - Op(i,o)) \quad (4)$$

At the beginning of propagation, all the nodes have the status *unlabeled*. In each step of propagation, the node with the shortest geodesic distance among all labels of all nodes is selected as the current seed s. The related object label l of this distance is identified as the final label of s, thus the object label of s is determined. Then the node s has an updated status *labeled*, and will not be considered in the following propagation. Next, the *unlabeled* neighbors of s are prepared for the update of their geodesic distances. As shown in Eq. 5, if $D(s,j)$ is lower than threshold T_1 and the gradient difference $bdry(s,j)$ between s and its neighbor j is lower than threshold T_2, then the weight on edge $W_e(s,j)$ is equal to $bdry(s,j)$, else it is a combination of $D(s,j)$ and $bdry(s,j)$. λ_d and λ_b are set to 0.2 and 0.8 experimentally.

$$W_e(s,j) = \begin{cases} bdry(s,j), & \Delta_d \\ \lambda_d * D(s,j) + \lambda_b * bdry(s,j), & else \end{cases} \quad (5)$$

$$s.t.\ \Delta_d : D(s,j) < T_1 \cap bdry(s,j) < T_2$$

If the sum of $geoDis(s,o)$ and $W_e(s,j)$ is shorter than the previous distance $geoDis(j,o)$, we update $geoDis(j,o)$ with the new distance, else we maintain the $geoDis(j,o)$ unchanged.

$$geoDis(j,o) = \begin{cases} geoDis(s,o) + W_e(s,j), & \Delta_c \\ geoDis(j,o), & else \end{cases} \quad (6)$$

$$s.t.\ \Delta_c : geoDis(s,o) + W_e(s,j) < geoDis(j,o)$$

3 Results and discussion

3.1 Dataset and experimental setup

To evaluate the performance of our method, we use the public datasets *Polo* [13, 15] and *TUD* [32, 33].

Polo dataset. This dataset contains 317 polo scene images, including 6 categories, i.e., sky, grass, person, horse, ground, and tree. We split these 317 images into 80 training images and 237 testing images, as did in [13, 15]. The horse and person category are the *object* categories, and the others are *material* categories. The 80 training images contain 208 horse instances and the 237 testing images contain over 500 instances of different poses, appearances, and scales. Each image in this dataset contains one or more than one object instances, some of which have occlusions. Therefore, this dataset is applicable to test the performance of object labeling.

Table 1 Performance comparison of scene-level labeling on Polo

Method	Total accuracy	Average accuracy
Shotton [4]	83.9	77.1
Ours	85.3	81.7

Table 2 Performance comparison of scene-level labeling on TUD

Method	Total accuracy	Average accuracy
Shotton [4]	95.1	91.9
Ours	95.5	92.8

Fig. 6 Examples of scene-level labeling result on Polo dataset. Best viewed in color

In this dataset, the category annotation map of each image is provided, while the annotations of object instances are not given. In order to evaluate our method quantitatively, we need to annotate the groundtruth of object instances. For an *object* category, to make our object annotation fit with the category region of provided annotation, we develop an annotation tool which tailors our annotation to its category boundary. In this way, pixels outside our object annotation are considered as *void*. In addition, we also annotate the bounding boxes for the training of object detection.

The label maps of category and object are shown in Fig. 4b, c respectively. In (b), different colors indicate different categories. In (c), different colors indicate different instances. Black indicates *void* in both category and object label maps. We order the instances in a scene by their layouts from left to right. The first instance is visualized in red, and the second in yellow, etc. Subfigure (d) shows the annotated bounding box of each instance.

TUD dataset. This dataset is priviously used for tracking by detection. It provides 201 images from a pedestrian sequence with 1216 tight bounding boxes and instance annotations of the pedestrians. Most of the pedestrians are side-view poses and many are partially occluded in the whole sequence. Pedestrians with at least 50% visibility are annotated in this dataset. We randomly split 100 images for training and the other 101 images for testing. In this dataset, the category annotation maps are not provided, thus we annotate the categories manually, including ground, person, building, and car. Besides, we make the annotation of person category fit with the given instance annotations. As illustrated in Fig. 5, the colors in (b) indicate the category labels and the colors in (c) and (d) indicate instances.

Fig. 7 Examples of scene-level labeling result on TUD dataset. Best viewed in color

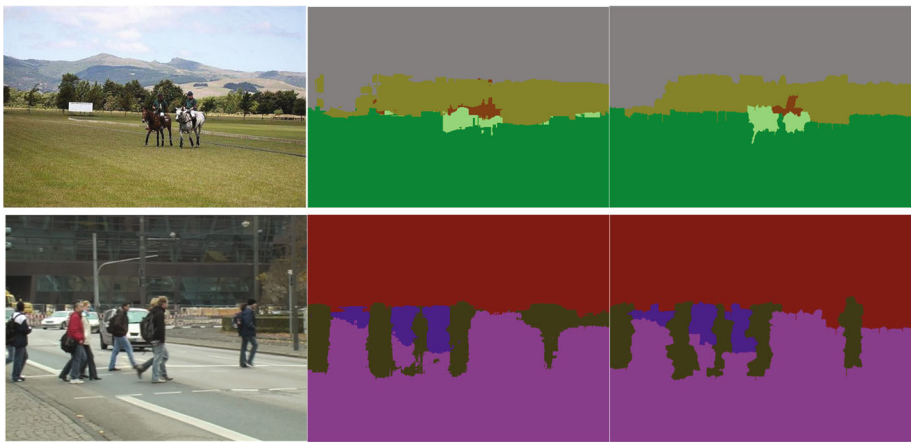

Fig. 8 Comparison of raw labeling and final labeling of our method. **a** The image. **b**, **c** The raw and final labeling respectively. Our method improves the boundaries of objects

Baseline methods. We adopt several baseline methods for quantitative comparison. For scene-level labeling comparison, the baseline is referred to that of Shotton et al. [4], which is the typical method in scene-level manner. For object-level labeling comparison, the baseline methods include E-SVM and HV+GC, similar to that of [15]. E-SVM generates segmentation by transferring template masks to the detected objects. HV+GC gets the instance segmentation by performing GrabCut on the voted hypotheses.

Running time. The average resolution per image is roughly 500*350 pixels for *Polo* dataset , and 640×480 pixels for *TUD*. Our implementation of Matlab code takes 5 min for learning per image, and less than 1 min for labeling of scene-level and object-level on a desktop with a 3.2-GHz Intel i5 CPU and a 12-Gb memory.

3.2 Scene labeling results

For the Polo dataset, we select 10 sample training images from the training set to learn our object detector. These sample images include object instances with variation in scale, pose, and occlusion. The other 70 images of training set are used as validation. The learned detector is performed on each test image, generating multiple bounding box predictions, which are denoted as $\{H\}$. We prune $\{H\}$ to include up to 7 objects, with T_h equal to 0.35 experimentally.

For the TUD dataset, we also select 10 sample images from the training set. Images in this dataset have some similarities since they are from the same tracking sequence. To avoid overfitting, we take a validation with a subset of training images instead of the whole training set. The $\{H\}$ is pruned to include up to 10 objects.

We modify the framework of TextonBoost [4] with 500 round traning times to generate the raw category probability. In the graph-cut optimization, we set the configuration of 1000 times iteration.

We use two metrics for scene-level labeling comparison, i.e., total accuracy and average accuracy. The total accuracy is the overall accuracy per pixel, and the average accuracy is the average accuracy per category. Tables 1 and 2 show the comparisons on Polo and TUD datasets. As we can see from these tables, our method performs better than that of Shotton et al. [4] in both total and average accuracy, especially the average one. The reason is that our method performs well for both material and object categories, while theirs is good at the material categories but poor at the object categories. Figures 6 and 7 show some examples of our scene-level labeling result on Polo and TUD datasets respectively.

Table 3 Performance comparison of object labeling on Polo

Method	Mi-AP	Mi-AR	Ma-AP	Ma-AR
E-SVM	38.5	33.6	43.9	38.3
HV+GC	44.6	38.7	61.7	49.4
He and Gould [15]	50.9	53.7	57.4	68.8
Ours	55.3	58.2	61.7	71.8

Table 4 Performance comparison of object labeling on TUD

Method	Mi-AP	Mi-AR	Ma-AP	Ma-AR
E-SVM	33.7	29.5	49.5	33.0
HV+GC	24.9	42.9	41.6	51.9
He and Gould [15]	62.6	56.9	64.8	64.5
Ours	62.9	59.8	65.6	64.7

Fig. 9 Examples of object labeling result on Polo dataset. Different colors indicate different instances. Other categories are visualized as *void*

Figure 8 shows the raw labeling and the final labeling of our method. The raw labeling is the initialization of probabilities. The final labeling is fine-tuned with our object labeling result. Comparing these two results, our method improves the overall accuracy as well as the object segmentation. See the figure for details.

3.3 Object labeling results

The accuracy of object labeling is different from that of semantic labeling. For example, assigning a wrong category label to a pixel will make an inaccurate understanding of scene; however, assigning a wrong object label to a pixel will not change the fact that it is an object of given category, as the purpose of object labeling is to partition the multiple instances of the same category.

Therefore, we use different criteria for object labeling. We calculate four evaluation metrics of (1) pixel-wise precision rate per object averaged over all object predictions (Mi-AP), (2) pixel-wise recall rate per object of groundtruth (Mi-AR), (3) pixel-wise precision rate over all

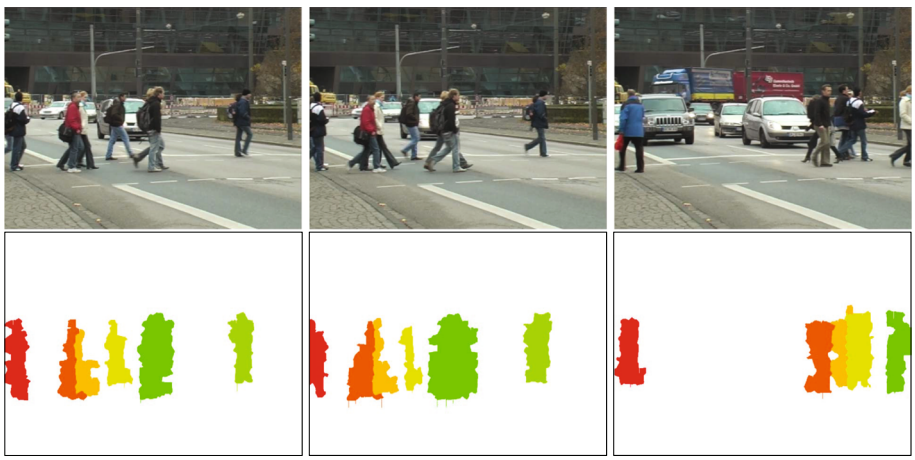

Fig. 10 Examples of object labeling result on TUD dataset. Different colors indicate different instances. Other categories are visualized as *void*

pixels (Ma-AP), and (4) pixel-wise recall rate over all pixels (Ma-AR).

To compare quantitatively with the groundtruth, we need to search the matching pairs between the prediction and the annotation. We reorder the segments of each instance from left to right, thus the matching pairs can be found efficiently. The comparisons of object labeling on Polo and TUD are listed in Tables 3 and 4. According to these tables, our method performs better than that of He and Gould [15] in the four metrics. Besides, we are better than the baselines of E-SVM and HV+GC, except in the Ma-AP of the Polo dataset.

In our experiment, the threshold T_p and T_s are 0.3 and 0.45, T_1 and T_2 are 0.85 and 0.5 for the Polo dataset. For TUD dataset, these parameters are 0.5, 0.2, 0.85, and 0.8 respectively. Some examples of object labeling result are shown in Figs. 9 and 10. These examples include multiple instances in different scales, poses, and even occlusions. Different instances are visualized in different colors and the region of non-object is visualized as *void*.

4 Conclusions

In this paper, we propose a hierarchical semantic segmentation method of both scene-level and object-level labeling. The two levels work together to give a more accurate understanding of an image scene. In the scene-level, we use a feature-based MRF model to recognize the categories. In the object-level, we use a constraint-based geodesic propagation to segment each instance. The experimental results show the good performance of our framework. However, a most important prior for object segmentation is not used explicitly in this work, i.e., shape information. Therefore, in future, we attempt to utilize the shape prior to improve the accuracy. We are going to set up a new dataset of our own for object labeling, which includes much more object instances than the two datasets we used in this work. In future, we will evaluate and improve our method and conduct many comparison experiments on our dataset.

Besides, we will utilize more discriminative features to segment object instances, such as the features captured by convolutional neural networks. Our method can be applied to autonomous driving systems, robots, etc. Considering the computational complexity, we may refer to some parallel methods [34, 35].

Acknowledgements
The authors would like to thank the editors and anonymous reviewers for their valuable comments.

Funding
This work is supported by National Natural Science Foundation of China (61502036), the General Project of Scientific Research Project of the Beijing Education Committee(KM201611417015), and Open Funding of Beijing Key Laboratory of Information Service Engineering (Zk20201502).

Availability of data and materials
Not applicable.

Authors' contributions
QL generated the idea of this work and discussed with the other two authors. QL carried out the main experiments and wrote the manuscript. AL and HL read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 13 June 2017 Accepted: 8 February 2018
Published online: 01 March 2018

References

1. Z Zhang, S Fidler, R Urtasun, in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. Instance-Level Segmentation for Autonomous Driving with Deep Densely Connected MRFs (IEEE Computer Society, Las Vegas, 2016)
2. C Yan, H Xie, D Yang, et al, Supervised hash coding with deep neural network for environment perception of intelligent vehicles. IEEE Trans. Intelligent Transportation Systems. **19**(1), 284–295 (2018)
3. C Yan, H Xie, S Liu, J Yin, Y Zhang, Q Dai, Effective uyghur language text detection in complex background images for traffic prompt identification. IEEE Trans. Intelligent Transportation Systems. **19**(1), 220–229 (2018)
4. J Shotton, J Winn, C Rother, et al, Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. Int. J. Comput. Vis. **81**(1), 2–23 (2009)
5. J Xiao, L Quan, in *IEEE Int. Conf. Computer Vision*. Multiple view semantic segmentation for street view images (IEEE Computer Society, Kyoto, 2009), pp. 686–693
6. J Yao, S Fidler, R Urtasun, in *The IEEE Conference on Computer Vision and Pattern Recognition*. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation (IEEE Computer Society, Providence, 2012), pp. 702–709
7. X Ren, L Bo, D Fox, in *the IEEE Conference on Computer Vision and Pattern Recognition*. RGB-(D) scene labeling: Features and algorithms (IEEE Computer Society, Providence, 2012), pp. 2759–2766
8. J Tighe, M Niethammer, S Lazebnik, in *The IEEE Conference on Computer Vision and Pattern Recognition*. Scene parsing with object instances and occlusion ordering (IEEE Computer Society, Columbus, 2014), pp. 3748–3755
9. C Liu, J Yuen, A Torralba, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Nonparametric scene parsing: Label transfer via dense scene alignment (IEEE, Miami, 2009), pp. 1972–1979
10. H Zhang, J Xiao, L Quan, in *Proceedings of European Conference on Computer Vision*. Supervised label transfer for semantic segmentation of street scenes (Springer, Crete, 2010), pp. 561–574
11. J Tighe, S Lazebnik, in *Proceedings of European Conference on Computer Vision*. Superparsing: Scalable nonparametric image parsing with superpixels (Springer, Crete, 2010), pp. 352–365
12. X Chen, Q Li, Y Song, et al, in *Proceedings of European Conference on Computer Vision*. Supervised geodesic propagation for semantic label transfer (Springer, Florence, 2012), pp. 553–565
13. H Zhang, T Fang, X Chen, et al, in *24th IEEE Conference on Computer Vision and Pattern Recognition*. Partial similarity based nonparametric scene parsing in certain environment (IEEE Computer Society, Colorado Springs, 2011), pp. 2241–2248
14. J Tighe, S Lazebnik, in *IEEE Conference on Computer Vision and Pattern Recognition*. Finding things: Image parsing with regions and per-exemplar detectors (IEEE Computer Society, Portland, 2013), pp. 3001–3008
15. X He, S Gould, in *IEEE Conference on Computer Vision and Pattern Recognition*. An exemplar-based CRF for multi-instance object segmentation (IEEE Computer Society, Columbus, 2014), pp. 296–303
16. BL Price, BS Morse, S Cohen, in *The IEEE Conference on Computer Vision and Pattern Recognition*. Geodesic graph cut for interactive image segmentation (IEEE Computer Society, San Francisco, 2010), pp. 3161–3168

17. D Batra, A Kowdle, D Parikh, et al, Interactively co-segmentating topically related images with intelligent scribble guidance. Int. J. Comput. Vis. **93**(3), 273–292 (2011)
18. J Wu, Y Zhao, J Zhu, et al, in *IEEE Conference on Computer Vision and Pattern Recognition*. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation (IEEE Computer Society, Columbus, 2014), pp. 256–263
19. C Rother, TP Minka, A Blake, et al, in *IEEE Conference on Computer Vision and Pattern Recognition*. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs (IEEE Computer Society, New York, 2006), pp. 993–1000
20. S Vicente, V Kolmogorov, C Rother, in *11th European Conference on Computer Vision*. Cosegmentation revisited: Models and optimization (Springer, Crete, 2010), pp. 465–479
21. S Vicente, C Rother, V Kolmogorov, in *The 24th IEEE Conference on Computer Vision and Pattern Recognition*. Object cosegmentation (IEEE Computer Society, Colorado Springs, 2011), pp. 2217–2224
22. X Liang, Y Wei, X Shen, et al, Proposal-free network for instance-level semantic object segmentation. IEEE Transaactions on Pattern Analysis and Machine Intelligence (2018). https://doi.org/10.1109/TPAMI.2017.2775623
23. Y Chen, X Liu, M Yang, in *CVPR*. Multi-instance object segmentation with occlusion handling (IEEE Computer Society, Boston, 2015), pp. 3470–3478
24. A Levinshtein, A Stere, KN Kutulakos, DJ Fleet, SJ Dickinson, K Siddiqi, Turbopixels: Fast superpixels using geometric flows. IEEE Trans. Pattern Anal. Mach. Intell. **31**(12), 2290–2297 (2009)
25. P Arbelaez, M Maire, CC Fowlkes, J Malik, Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 898–916 (2011)
26. PF Felzenszwalb, RB Girshick, DA McAllester, D Ramanan, Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
27. Y Boykov, O Veksler, R Zabih, Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
28. S Goferman, L Zelnik-Manor, A Tal, Context-aware saliency detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(10), 1915–1926 (2012)
29. M Cheng, NJ Mitra, X Huang, PHS Torr, S Hu, Global contrast based salient region detection. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 569–582 (2015)
30. A Farhadi, I Endres, D Hoiem, DA Forsyth, in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*. Describing objects by their attributes (IEEE Computer Society, Miami, 2009), pp. 1778–1785
31. Q Li, X Chen, Y Song, Y Zhang, X Jin, Q Zhao, Geodesic propagation for semantic labeling. IEEE Trans. Image Process. **23**(11), 4812–4825 (2014)
32. H Riemenschneider, S Sternig, M Donoser, et al, in *Proceedings of European Conference on Computer Vision*. Hough regions for joining instance localization and segmentation (Springer, Florence, 2012), pp. 258–271
33. M Andriluka, B Schiele, S Roth, in *the IEEE Conference on Computer Vision and Pattern Recognition*. People-tracking-by-detection and people-detection-by-tracking (IEEE Computer Society, Anchorage, 2008), pp. 2759–2766
34. C Yan, Y Zhang, J Xu, F Dai, L Li, Q Dai, F Wu, A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. IEEE Signal Process. Lett. **21**(5), 573–576 (2014)
35. CC Yan, Y Zhang, J Xu, F Dai, J Zhang, Q Dai, F Wu, Efficient parallel framework for HEVC motion estimation on many-core processors. IEEE Trans. Circ. Syst. Video Techn. **24**(12), 2077–2089 (2014)

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com