Computer Vision and Image Understanding 145 (2016) 70–80

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Scene parsing using graph matching on street-view data☆

Tianshu Yu[a,*], Ruisheng Wang[a]

Department of Geomatics Engineering, University of Calgary, 2500 University Dr NW Calgary, Alberta T2N1N4, Canada

ARTICLE INFO

Article history:
Received 28 April 2015
Accepted 11 January 2016
Available online 21 January 2016

Keywords:
Scene parsing
Graph matching
Markov random field
Street view

ABSTRACT

Scene parsing, using both images and range data, is one of the key problems in computer vision and robotics. In this paper, a street scene parsing scheme that takes advantages of images from perspective cameras and range data from LiDAR is presented. First, pre-processing on the image set is performed and the corresponding point cloud is segmented according to semantics and transformed into an image pose. A graph matching approach is introduced into our parsing framework, in order to identify similar sub-regions from training and test images in terms of both local appearance and spatial structure. By using the sub-graphs inherited from training images, as well as the cues obtained from point clouds, this approach can effectively interpret the street scene via a guided MRF inference. Experimental results show a promising performance of our approach.

1. Introduction

Scene parsing, which is the segmentation and classification of regions in an image with different semantics, is of great importance in the computer vision community. For decades, various approaches have been developed to parse scenes on image sets, so that algorithms can learn and infer results from the significant amount of information provided by images. However, there are difficulties when only images are used. First, it is difficult to train a sufficiently effective classifier to label the regions due to the diversity of the categories. Second, shadows heavily influence the labeling in most of the images, such that regions on two sides of the shadow edge are labeled with different categories although they represent the same object. In contrast, point clouds can provide extra cues that images cannot convey. For example, given depth and height measurement, the 3D shape of an object can be estimated precisely; shadows will never affect the segmentation of the point clouds. Methods integrating both images and LiDAR data have been explored over the past few years. In this paper, we focus our attention on the urban street view with data from both images and point clouds collected at street level; the data were provided by Google.

Many efforts have been made to accurately parse images into a variety of categories. These methods are mostly based on the 2D global or local features. Liu et al. [1] proposed a scene parsing framework based on dense image alignment over dense scale-invariant feature transform (SIFT) images, which has been a successful technique and performs very well. However, this method works on a pixel-wise level and the belief propagation optimization over the graph, with respect to pixels, has a high computational complexity. In addition, a large image database is needed [2], which makes this method difficult to use in practice. Farabet et al. [3,4] employed a multi-scale convolutional network to compute dense feature vectors centered on each pixel. After a max-pooling stage, a classifier is trained to estimate the histograms of all object categories. Another method is a scene segmentation approach that matches object boundaries or edges across scenes [5], which does not need to extract features from images explicitly. Similarly, Russell et al. [6] also proposed a scene segmentation method by matching image composites across scenes. Thus different visual clues can be collected to help enhance the parsing accuracy. Compared to the algorithms that work on pixel level, on the other hand, some attempts over superpixels have been made [7–9]. These methods are based on the fact that representation over superpixel can reduce the computational complexity remarkably, and also can naturally form an aggregate of pixels with a similar local appearance. Since the performance of the superpixel segmentation significantly influences the parsing results, state-of-the-art segmentation approaches [10,11] are always employed. The approach proposed by Tighe and Lazebnik [7] is a typical non-parametric parsing scheme over superpixels using the segmentation method in [11], in which a training stage is not necessary.

More accurate parsing performance can be achieved by integrating images and 3D information than when images alone are used. Some approaches [12–14] use structure-from-motion

☆ This paper has been recommended for acceptance by Stephen Gould.

* Corresponding author.
E-mail addresses: yut@ucalgary.ca, shuitx@gmail.com (T. Yu), ruiswang@ucalgary.ca (R. Wang).

http://dx.doi.org/10.1016/j.cviu.2016.01.004
1077-3142/

method [15], which allows 3D information in different scenes to be effectively estimated from image sequences (e.g., stereo, video). Especially, Xiao and Quan [14] developed a method in which a refined dense depth map was computed, providing extra information such as surface normal, planarity, height and distance to camera path. However, methods in this category heavily rely on the estimation accuracy of the 3D information and have a high computational complexity. An alternative to estimate the 3D information is to gather data using LiDAR sensors [16–19]. Zhao et al. [16,17] used Velodyne LiDAR sensors and cameras mounted on vehicles to capture images and LiDAR data. They detected a large range of objects as "obstacle" (e.g., car, tree, pedestrian and any other objects limited in a bounding box); Fuzzy logic inference was employed to classify them in various categories. Instead of using 3D information, Ardeshir et al. [20] proposed a method to conduct scene understanding with the help of location and address. Wang et al. [21] introduced a holistic scene understanding framework by integrating object detection, pose estimation, depth reconstruction and semantic segmentation. Since the main objects in street scenes are buildings, it is anticipated that the parsing scheme can also work on building facades. Unlike regular image parsing tasks with several labels, facade parsing aims at identification of targets with common shape and symmetry. Some researchers propose to implement facade parsing by using only images [22,23]. Other methods [24,25] try to interpret building facades using range data.

Almost all the aforementioned methods use the information in pixel or superpixel, either to train a classifier or to minimize the energy function. The information in pixel or superpixel can be considered as lower level vision features, while these features include SIFT, HOG or Color Histogram, which tend to be isolated without spatial relationship. However, in pictures of real world, pixels or superpixels appear in a more organized structure which can be referred to as higher level features, namely "objects". One object always includes a collection of typical features and their spatial relationship, and this observation inspires many computer vision applications by representing an object using a graph (i.e., undirected graph with attributes on both nodes and edges). In street view images, similar objects appear repeatedly across scenes, such as buildings, cars and pedestrians. One can anticipate that matching similar objects in terms of structured graphs regardless of locations where they appear in the pictures, will make the parsing more effective and reliable. Compared with the methods that employ information at pixel or superpixel level, matching at object level can be considered as employing "higher" level visual cues from images. Moreover, identifying similar objects is a more natural way as how human beings recognize a scene. To the best of our knowledge, extensive research has not been conducted on scene parsing based on graph matching, though this has been shown to be successful in object matching, recognition and retrieval [26–29]. A similar method was proposed by Gould and Zhang [30], in which a correspondence graph is established via patch similarity to achieve the annotation transferring. Our work is inspired by [6] which collects low-level visual features across scenes.

A parsing algorithm via Markov Random Field (MRF) optimization on a guided graph derived from a graph matching of superpixels is proposed. In this scheme, the structure of the guided graph has advantages of both data fidelity and inherited sub-graphs from training images. The following contributions are made in this paper:

1. A graph matching scheme [31] is employed to interpret the semantic structure of the query scene with respect to the guidance scene. Taking into account the fact that similar images share a similar semantic structure, this step involves the calculation of sub-graphs in the query scene, which correspond to matched structures in the training scenes, providing more plausible evidence for the labeling stage.
2. Apart from normal MRF optimization, this approach takes both the query scene itself and the structures inherited from training scenes as input; consequently, the optimization is implemented on an enhanced graph structure where the label distribution is constrained and the inherited adjacency relationship is considered.

The rest of the paper is organized as follows. In Section 2, we introduce and elaborate on the proposed approach. Specifically, details how to find correspondence via graph matching and how to use the matching results as guidance to infer the labels of the query images are presented in in Sections 2.3 and 2.4, respectively. In Section 3, we present the experiments that was conducted on datasets from several cities with different street view styles. Experimental results show that approach achieves the promising performance. The conclusion and future research are presented in Section 4.

2. Methodology

In this section the workflow of the approach is detailed. First, preliminaries and pre-processing of the image data are introduced. A segmentation scheme is applied to the point clouds for partitioning structures with different semantics and constraining the probability distributions of the labels. By finding correspondences using graph matching between query images and training images, the regions that share similar structures and local appearances can be found. Finally, label inference is implemented using MRF on a guided graph, which is the combination of image fidelity and inherited structures. The potentials of the nodes in the graph follow distribution constraints derived from the point clouds.

2.1. Preliminaries and pre-processing of image sets

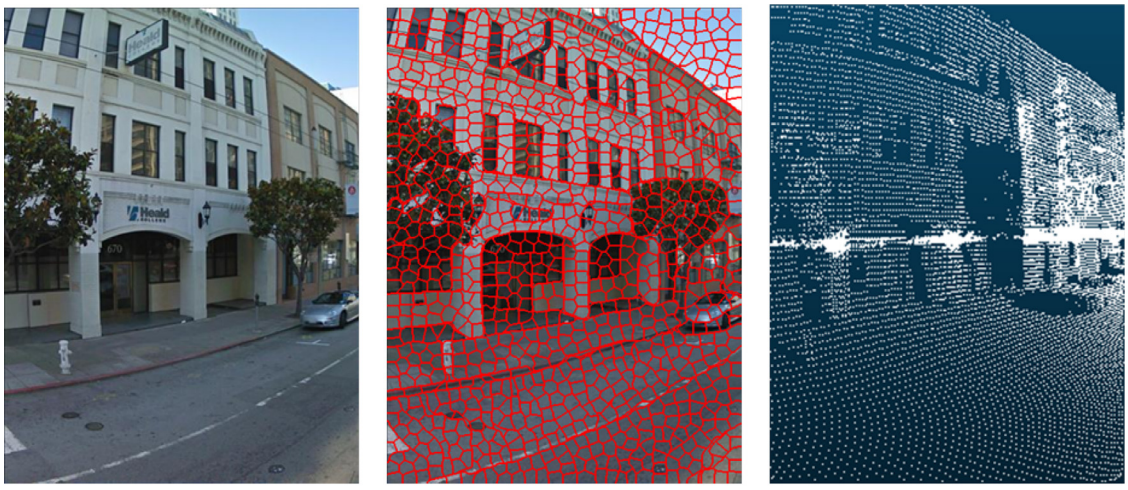
The images are collected using perspective shuttering cameras and the range data is scanned using SICK LiDAR sensors. Examples of the data are shown in Fig. 1 (a) and (c). According to the figure, the camera intrinsics distort the image to some degree.

The images in the training set and the test set are labelled manually using the tool "labelMe" [2]. The images are segmented into the label set sky, building, window, door, vehicle, pedestrian, road, tree, and sign. Different categories can be specified using our method on other problems, depending on what is required to address. The regions that cannot be categorized into the above labels, are classified as "undefined". One may observe that "window" and "door" appear on the building facades, therefore they also belong to the "building" category. This setting is used in order to examine the parsing ability of our algorithm on building facades.

All the training and test images are then over-segmented into superpixels using the approach in [11]. This method is capable of giving appropriate and sufficient information on image regions, and reaching a good balance between computational complexity and informational fidelity, as compared to a pixel-level approach. An example of superpixel segmentation is shown in Fig. 1 (b). The approach is effective under the assumption that images and range data are well aligned; otherwise an alignment procedure for the two types of data would need to be implemented first.

2.2. Point cloud segmentation

In Fig. 1 (c), the corresponding point cloud is shown from the same viewpoint as Fig. 1 (a). The point cloud is transformed into the camera pose using the corresponding camera intrinsics and projective transformation. To segment the objects in the point



(a) (b) (c)

Fig. 1. Example of the data. (a) The image from the perspective camera, (b) Superpixel segmentation of (a), (c) The corresponding LiDAR data.

cloud, a framework is employed as follows. First, the ground points are separated and labeled using piece-wise random sample consensus (RANSAC) [32] plane fitting near the ground height with respect to the vehicle, assuming that the ground is locally planar. All points corresponding to the ground are then removed and the above-ground structures are retained. Since all other objects are originally attached to the ground, it is simple to partition and separate each single object using Euclidean clustering [33]. Thus the ground points and all the above-ground points without labels are obtained.

Since the captured point cloud is not sufficiently dense and information is missing, as shown in Fig. 1 (c), it is difficult to classify each structure perfectly. Rather than identifying the exact label of each isolated structure, our method recognizes the subset of labels to which the structure does not belong. For some specific structures where the features are apparent, such as “sky” and “facade”, the exact classification can be obtained. However, ambiguity occurs on small structures; for example, it can be unclear as to whether an object is a “pedestrian” or small “tree” due to the lack of contextual information from the sparse point cloud. Thus, only the feasible label set pedestrian, tree can be applied, and the exact classification is left until the following stages using image information. For now a uniform probability distribution summing to 1 on each feasible label set is used.

To calculate the feasible label set, the building facade is first identified by taking into account the size of structure and the normal on each point, since most of the normal vectors are parallel to the ground. Due to the fact that there are small irregularities on the plane of the facade, the difference of normal [34] is used to estimate the normal vector. This method not only identifies the planes of facades, but also calculates the edges of the windows and doors to some extent if they are made of glass (i.e., no reflection of laser from the area). The lack of precision is due to the fact that normals on a small scale, and near the edges of windows and doors, are not stable. Therefore, by finding “holes” and the “edges”, the possible positions of windows and doors can be located. However, given sparsity of the point cloud, the locations of windows and doors are still imprecise. This will be discussed further in Section 3.2.

The rest of the objects are interpreted using their sizes and 3D shapes. First, we construct a bounding box for each object. The objects can be assigned feasible label sets by taking into account the size of bounding boxes and the ratio of height, width and length. For the objects with low height, we exclude “sign” and “tree” from the feasible label set, since signs on road are mostly attached to a high pole (trees are also high). We calculate the ratio of the length to the height of the bounding box, and use this ratio to identify if an object does not belong to “vehicle”. This is because the shape of the bounding box of an vehicle is always a cuboid, thus the ratio tends to be above a threshold. For a high object, we also try to identify if it belongs to “tree ” or “sign” by using the shape of the object. Since the points of a tree diverse more than a sign or pole in the horizontal direction, we introduce the PCA to calculate the principle direction of the point cluster, and use overall variance along the principle direction to identify to which category an object belongs. As discussed above, sometimes it is difficult to identify the object into a single label, because of occlusion, partial scan or movement of the objects. In this case, an initial uniform distribution over undecided labels is assigned to the object. The regions with no points that are not labeled into aforementioned labels are considered to be in the category of “sky”. It should be noted again that the segmentation procedure does not provide a deterministic classification for each single object, rather it calculates the feasible set that an object may belong to with a relatively high belief. Besides, by narrowing the feasible label set for each object, the corresponding searching space for each node (superpixel) in MRF can further be reduced.

After segmenting the point clouds, the feasible label sets of the objects are obtained; and each single object in point clouds is assigned a uniform probability distribution. This information is used to assist the MRF inference.

2.3. Finding correspondence via graph matching

Feature or region correspondence via graph matching is one of the fundamental problems in computer vision, which consists of forming undirected graphs on features for both reference and query images, and finding similarities between both node pairs and edge pairs using graph theory. Since the relative positions of the camera and the LiDAR sensor are fixed, it can be observed that, although there are many different images, their structures can be similar. It does not mean that the structures of the images are the same, instead, it’s more likely to find out some training samples that share a similar image structure as the query image. For example, many images taken by the front camera show a structure in which buildings are on both sides of the road and a portion of sky appears between them, as well as vehicles and pedestrians appearing on the road. This observation leads to the representation

of image structures using graphs. In our work, the approach [31] to match sub-graphs between training images and query images is utilized.

The structures of the images are represented using adjacency graphs with respect to superpixels. An adjacency graph on an image is denoted as $G = \langle N, E, A\rangle$ where N is the node set of the graph while each element in N corresponds to a superpixel, an edge E_{ij} belongs to the edge set E if and only if the corresponding two superpixels i and j are adjacent. A is the attribute assigned to the node or the edge. The attribute A_i assigned to the node i is a vectorized feature on the corresponding superpixel; and A_{ij} is the attribute assigned to the edge E_{ij}. Several features are adopted at this stage, including color histogram, normalized visual words in terms of SIFT features and texton histogram. The selected features are listed in Table 1.

Specifically, the normalized visual words are computed by first finding clustering centers over dense SIFT descriptors on all the training images, and, a histogram is then formed by calculating to which cluster each pixel on a superpixel belongs. Determination of the features is presented as follows. Assuming that the centroids C_k of the SIFT clusters have already been calculated using the Bag of Words (BOW) model, and the number of the centroids is K, then for a given superpixel P_i the corresponding normalized feature H_i is calculated as follows:

$$B_{k,i} = \sum_{h \in S_i} Nearest(SIFT(h), C_k) \quad d = 1, \ldots, K$$
$$H_i = \frac{1}{Z}[B_{1,i}, ..., B_{K,i}] \tag{1}$$

where S_i is the ith superpixel and h represents a pixel in P_i, $SIFT(h)$ calculates the SIFT feature of pixel h, C_k is the kth centroid of BOW and $Nearest(\cdot)$ is the nearest classifier. Z is a parameter to guarantee H_i is normalized.

The original adjacency graph G represents a pairwise MRF which is easy for labeling inference. However, adjacency relationship alone cannot provide sufficient information for graph matching since affinity should also be considered in finding feature correspondence. Additionally, matching over all superpixels is a waste of computational resources because correspondence can be found only on a sub-set of the superpixels. Therefore, an "active graph" [31] should be constructed in order to reduce the candidate node and edge set and preserve the affinity between node pairs. To achieve this, we propose to use l^λ-distance as the metric to represent the similarity between superpixels:

$$S(i, j) = \|F_i - F_j\|_\lambda \tag{2}$$

where i and j represent two superpixels from the query image and training image, respectively. F_i and F_j represent the corresponding feature vectors, respectively. For superpixel i, all the values of j that satisfy $S(i, j) < \sigma$ are regarded as candidate nodes, and vice versa. σ is a threshold. To measure the similarity between edge pair, we follow the setting from the approach [31] by using Symmetric Transfer Error (STE). The difference in this research is that we normalized and stacked the three descriptors on superpixels (color histogram, normalized visual words and texton histogram) into a single feature vector. Further, the position of a superpixel is defined by the center of mass. We define the obtained "active graphs" as G^p and G^q for the query and training images, respectively.

For a query image, the global features are first computed and a subset is found from the training images using a K-nearest approach in terms of a l^2-distance. This subset leads to fewer images to compare, saving a significant amount of computational time. Two types of global features are adopted: GIST [35] and Spatiogram [36], as shown in Table 1 (b).

A graph matching approach proposed by Cho and Lee [31] is utilized as the basic scheme to find region correspondence in our implementation. In this approach, correspondences between node pairs are found by maximizing an energy function defined on a reweighted affinity-preserving graph. Let us regard both the query image and one training image as active graphs with attributes with respect to the superpixels, hence $G^p = (V^p, E^p, A^p)$ for query image and $G^q = (V^q, E^q, A^q)$ for a training image. A compatibility matrix $W_{ia:jb} = f(A_i^p, A_j^p, A_{ij}^p, A_a^q, A_b^q, A_{ab}^q)$ measures the mutual consistency of the attributes between pairs of correspondences (V_i^p, V_a^q) and (V_j^p, V_b^q), where A is the attribute assigned to both nodes and edges. i and j are nodes from graph G^p, and a and b are nodes from graph G^q. The elements of W are defined by using the concatenated feature descriptor on the superpixels and the STE measurement. Concretely, for a pairwise edge similarity $W_{ia:jb}$, an affine region feature i on the superpixel x_i^p (concatenated feature), centered at the center of mass of the superpixel, is represented by an elliptic region, and the orientation is estimated using gradient histogram [37]. With this notation, the affine homography transformation $\mathcal{T}_{ia}(\cdot)$ is defined from a feature i in p to another feature a in q, so that the neighborhood points x^p and x^q of x_i^p and x_a^q are related by $x^q = \mathcal{T}_{ia}(x^p)$ [31,38]. Then the transfer error of (j, b) with respect to (i, a) is denoted as $d_{jb|ia} = \|x_b^q - \mathcal{T}_{ia}(x_j^p)\|$. Thus the STE measurement W can further be defined as $W_{ia:jb} = max(0, \alpha - (d_{jb|ia} + d_{bj|ai} + d_{ia|jb} + d_{ai|bj})/4)$.

We denote $x \in \{0,1\}^{n_p n_q}$ representing correspondence between the two graphs which is a vectorized replica of $X \in \{0,1\}^{n_p \times n_q}$, where n_p and n_q are the numbers of superpixels in the query and training images respectively. Using the pre-defined W, the graph matching problem can then be formulated as finding vector x^* that maximizes the energy function:

$$x^* = \arg\max(x^T W x) \quad s.t.$$
$$x \in \{0,1\}^{n_p, n_q}$$
$$X 1_{n^q \times 1} \le 1_{n^p \times 1}, \quad X^T 1_{n^p \times 1} \le 1_{n^q \times 1} \tag{3}$$

where the constraints represent the one-to-one matching from G^p to G^q, which makes X an assignment matrix. $1_{n \times 1}$ denotes an all-ones vector with size n.

For more information on the optimization approach, refer to the work [31]. This approach is significantly faster than traditional methods in which graph nodes are image pixels, because superpixel segmentation guarantees the searching space to be small.

We define the matched sub-graph from the query image and the training image as $G_M^p = (V_M^p, E_M^p)$ and $G_M^q = (V_M^q, E_M^q)$, respectively. Moreover, a node correspondence set is also established as $\{(V_i^p, V_a^q)\}$. Once the correspondence between the query image and the training image is computed, it can be supposed that the adjacency or consistency relationship among superpixels is inherited from the graph of the training image. This inheritance can be partial, which means that only a part or a region of the two images holds the similarity. In other words, we find a common subgraph from the training image and the query image, and this subgraph represents the similar image structure from the two scenes. This relationship will guide the label inference in the next step.

2.4. Label inference using guided MRF

Before the label inference, the data and smoothness terms in the MRF formula should be defined. Similar to the process outlined in the paper [16], given all labeled training sample superpixels, a classifier is trained by using a randomized decision forest. The input of the training stage includes the stacked descriptors and the corresponding labels. Unlike the method outlined in the paper [16], the stacked features are used without normalization, because each

Table 1
Selected local and global features.

(a)
Superpixel features

Type	Description	Dimension
Color histogram	3-channel RGB, 11 bins per channel	33
SIFT histogram	100 visual words, normalized histogram	100
Texton histogram	Dilated texton histogram	100

(b)
Global features

Type	Description	Dimension
GIST	3-channel RGB, 3 scales, 8, 8 and 4 orientations	960
Spatiogram	3-channel RGB, 8 bins, 8^3 histogram	512

single feature in the stacked feature is a type of normalized histogram, that has already been uniformly scaled. After training, the classifier takes features of the test superpixel as input, and outputs a probability distribution vector P_i which represents the probabilities of the test sample i belonging to each category. This distribution is considered the data term. The same definition as in [16] is used to determine the smoothness term. Potentials of energy function are thus obtained.

To utilize the obtained regions that match the training images, the structures of the adjacency graphs of the test images have to be modified. For a set of node pairs with correspondence $\{(V_i^p, V_a^q)\}$, where $n_i \in V^p$ and $n_a \in V^q$, we adjust the graphs as follows. If $E_{ab}^q \in E_M^q$ is an edge in the matched sub-graph corresponding to the training image, we add a new edge E_{ij}^p into the graph G^p where (V_j, V_b) is in the correspondence set and $V_{ab}^q \in E^q$. This means that the adjacency relationship between superpixels on the query image is enhance or established by identifying the matched superpixels and adjacency in the test image. This procedure is implemented over all candidate training images, and the enhanced graphs are defined as $G_{guided}^p =< N_{guided}^p, E_{guided}^p >$.

A guided MRF is introduced to handle the refined labeling problem. In this scheme, both the label space and the adjacency relationship are constrained using the derived adjacency relationship and the distribution from the point clouds.

In general, the aim is the minimization of the following energy function:

$$E(L) = \sum_{i \in N^p, L_i \in D(i)} \psi_i(L_i) + \rho_1 \sum_{(i,j) \in E^p} \psi_{i,j}(L_i, L_j) + \rho_2 \times \sum_{(f,g) \in E^H} \psi_{f,g}(L_f, L_g) \quad (4)$$

where $D(i)$ represents the category to which superpixel i belongs. E^p is the edge collection in original graph and E^H is the edge collection in the inherited graph. ψ_i is calculated by taking the minus log form of the output of the random decision forest. In other words, $\psi_i(L_i) = -logP_i(L_i)$, and P_i is the output of random forest on superpixel i. We employ same way to construct $\psi_{i,j}$ as [16]. The first term in this formula represents the data fidelity according to the superpixel appearance, while the possible distribution of the labels on a specified superpixel is constrained under the cues from the point clouds and the subgraph inherited from training images. In Eq. (4), the first two terms are traditional in MRF inference, which measure data fidelity and neighboring smoothness, respectively. The third term, which is derived from the graph matching, is added based on the following fact. A superpixel does not appear in an image as a single element, instead a group of superpixels formulate an "object" in a way of cooccurrence. The objects then form an natural image in a semantic manner. In street view, although there exist deformation or displacement, similar objects frequently appear across scenes. Graph matching is a natural choice to find the object correspondence against deformation or displacement. In general, graph matching is utilized to find partial correspondence from query images and training images. Then, the matching from a region in the query image to a region in the training image gives more belief to which category the region belongs, since the cooccurrence is a reliable evidence. In this case, we should enhance the interaction between the nodes that belong to the same category, which is reflected by adding the third term.

An obstacle that lies apart from the building facade can be easily found from the point clouds, therefore the corresponding possible distribution can only be tree, car, pedestrian, vehicle, sign. In the MRF inference we assign the probabilities of these label states on a specific node summing to 1 subject to the output of randomized decision forest, and the rest probabilities of other states to be 0. Normally, if the number of the category is n, and a feasible distribution from point cloud is defined as $\mathscr{D}_i = \{0,1\}^n$, then the probability distribution can be adjusted using:

$$P_i^* = \frac{1}{Z} P_i \bullet \mathscr{D}_i \quad (5)$$

where P_i is the vector output using the trained classifier, $\bullet$ refers to the dot product of two vectors and Z is a parameter that guarantees P_i^* to be a probability distribution. After adjusting the distribution, P_i^* is employed as the new data term. A simple example is shown in Fig. 2 (a), where it is assumed that there are only three categories.

The second term is the smoothness term, which is defined on the edges of the graph. The third term, which is newly employed, measures the smoothness of the relationship that the query image inherits from the training image. This term adds a strong constraint, yielding an extra relationship on the nodes that ensures a similar inference on a similar pair of scenes. Alternatively, it can be said that a new weighted edge is added to the graph if the two end nodes inherit a relationship from the training images. Normally, an new edge is added if and only if:

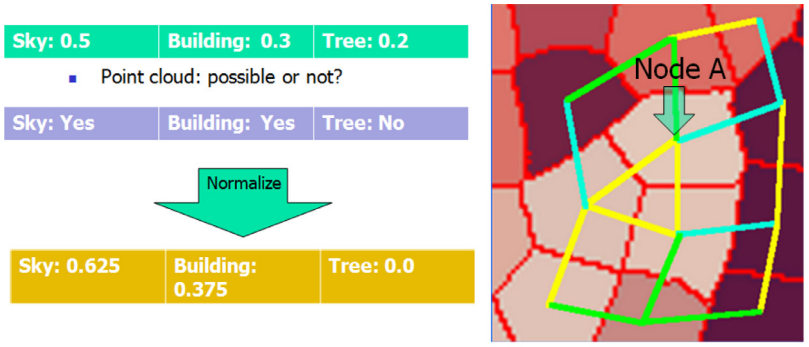
1. This edge does not exist in the current graph;
2. After graph matching, a corresponding edge exists in the reference image;
3. The nodes of the reference edge belong to the same category. A simple example is depicted in Fig. 2 (b).

Formula (4) can be integrated as one in the pairwise MRF framework. To realize this integration, we simply add some inherited edges onto the graph and provide every node with a different distribution. The energy function is then transformed into:

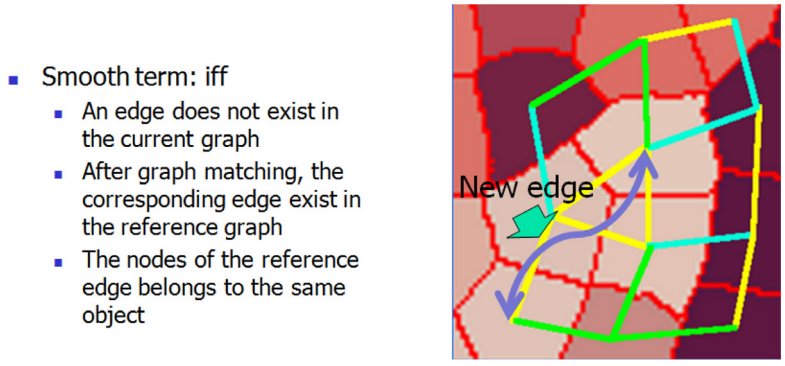
$$E(L) = \sum_{i \in N_{guided}^p, L_i \in D(i)} \psi_i(L_i) + \sum_{(i,j) \in E_{guided}^p} \psi_{i,j}(L_i, L_j) \quad (6)$$

The optimization of the energy on this newly constructed graph can be achieved using Max-product Belief Propagation [33]. The message at time t from node i to node j is defined as:

$$m_{i \to j}^t = \min_{L_i} (\psi_i^*(L_i) + \varphi_{i,j}(L_i, L_j) + \sum_{(k,i) \in N_{guided}(i) j} m_{k \to i}^{t-1}) \quad (7)$$



(a)



(b)

Fig. 2. Illustration of the construction of guided MRF. In (a) a new distribution is constructed to adjust the energy term, while in (b) new edge is added to MRF to enhance the intraclass relationship.

Under the constraints from the point clouds, the value of ψ_i^* is obtained by adjusting ψ_i. After iterations, the image can be classified by taking the label with the max belief value. The street view, combined with range data, can be parsed effectively under this process.

3. Experiments

To test the performance of the proposed parsing method, we conducted experiments on the datasets from New York, Paris, Rome and San Francisco. All the data is in the format as in the example shown in Fig. 1. In the experiments, the inference results, using graph matching with both images and point clouds, were compared with the results of using only images.

3.1. Dataset and parameter set-up

There is no public database containing well-aligned images and point clouds. In our comparison experiment, we test two sets of data.

The first database, including images and well-aligned point clouds of the street views of the aforementioned four cities (New York, Paris, Rome and San Francisco), was collected by Google. In this database, the images were taken with eight vehicle-mounted perspective rolling shutter cameras, each of which had its own camera intrinsics and distortion parameters. There are also three LiDAR scanners mounted on the front, left and right sides of the vehicle. The images from the same camera were taken at specific time interval and range data were captured. The coordinate system was UTM (Universal Transverse Mercator). The number of images in the datasets for the four cities were in tens of thousands. However, there were many overlaps since the time intervals were short. Parameters ρ_1 and ρ_2 in formula (4) and λ in formula (2) were not defined. They were set to 0.9, 0.3 and 1.3, respectively, as determined from the experiments, in order to achieve the best performance. The candidate size is set to 30.

The second database is LM+SUN[15], which is adopted to test the parsing performance when only images are available, though our algorithm is designed to deal with images and point clouds. This database contains 45,676 images, while 24,494 images are outdoor and the rest images are indoor. We only test our algorithm using the outdoor images, which are split into 23,994 training images and 500 testing images. ρ_1, ρ_2 and λ are set to 0.9, 0.2 and 1.6, respectively. Since the database is large, the size of the candidate set is chosen to be 400.

To present a quantitative evaluation of our method, we employ three criteria from [16].

$$\begin{aligned} \text{Precision} &= \frac{|GT \cap DR|}{|DR|} \\ \text{Recall} &= \frac{|GT \cap DR|}{|GT|} \\ \text{F-measure} &= \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \end{aligned} \tag{8}$$

where GT represents the set of pixels that are classified to a specific category by the proposed approach, and DR represents the set of pixels that is a manual label to a specific category (i.e., ground truth). F-measure is the weighted harmonic mean of GT and DR, which is used to quantify the overall performance of the parsing scheme.

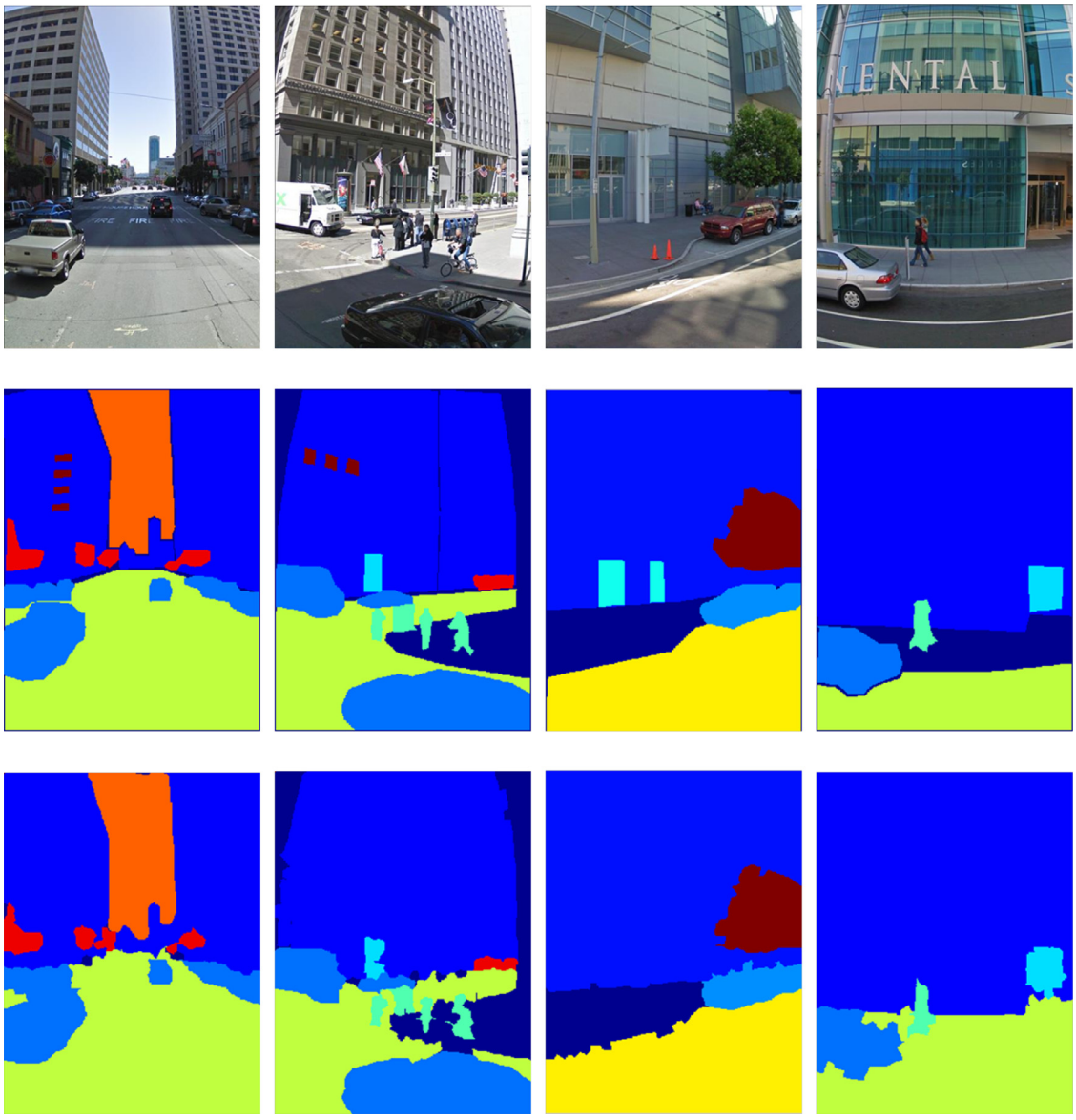


Fig. 3. Examples of parsing result. The first row lists the images. The second row represents the ground truth segmentation by manually labeling. In the third row, we present the parsing result using our approach. It can be seen that though with highly accurate overall performance, problems still occur in the categories of windows and doors.

3.2. Evaluation on images and range data

The performance on both images and range data was evaluated using the proposed method. Fig. 3 shows several parsing examples, where the first row contains the sample images from the dataset, the second and third rows correspond to the ground truth labeling and the parsing result, respectively. In this test, four sub-tests were conducted on each city. There are many images and overlaps in each sub-dataset, therefore for each city, we randomly choose 300 scenes as the training data and another 200 scenes as test samples, in order to avoid the overlaps and bias introduced by manual selection. The experiments for each city were conducted separately rather than combining all scenes together since the building and street styles can differ between cities. However, the building and street styles are similar in most cities, resulting in ease of identification in a specific city for both natives or travelers. This interesting phenomenon was discussed in [39].

A series of tests were performed with different sizes of the candidate image set. Experimental results are demonstrated in upper left of Fig. 4. As shown, if the size of the candidate set was properly chosen, scenes can be effectively parsed using the proposed approach under complex environmental distortions. We found that 30 was a good choice in most cases. We also present the parsing performance with a candidate set size 30 on each category in Table 2. The proposed algorithm worked well on structures with large areas such as sky, buildings and roads, where the superpixel feature is easier to discriminate and the corresponding portion of the point cloud is easier to classify. However, errors occurred when parsing windows and doors, as shown in the first and third columns of Fig. 3. Several factors led to these errors. First, when manually labeling the images, we did not label each window and door on the building facades since there were many items in these two categories; instead, we just label several apparent ones. This way of labeling made it difficult to train a good classifier for windows and doors. Second, because the point clouds collected are sparse, it's relatively difficult to locate the windows and doors in the point clouds. Therefore, the corresponding probability distribution of the categories could not be effectively constrained. Third,

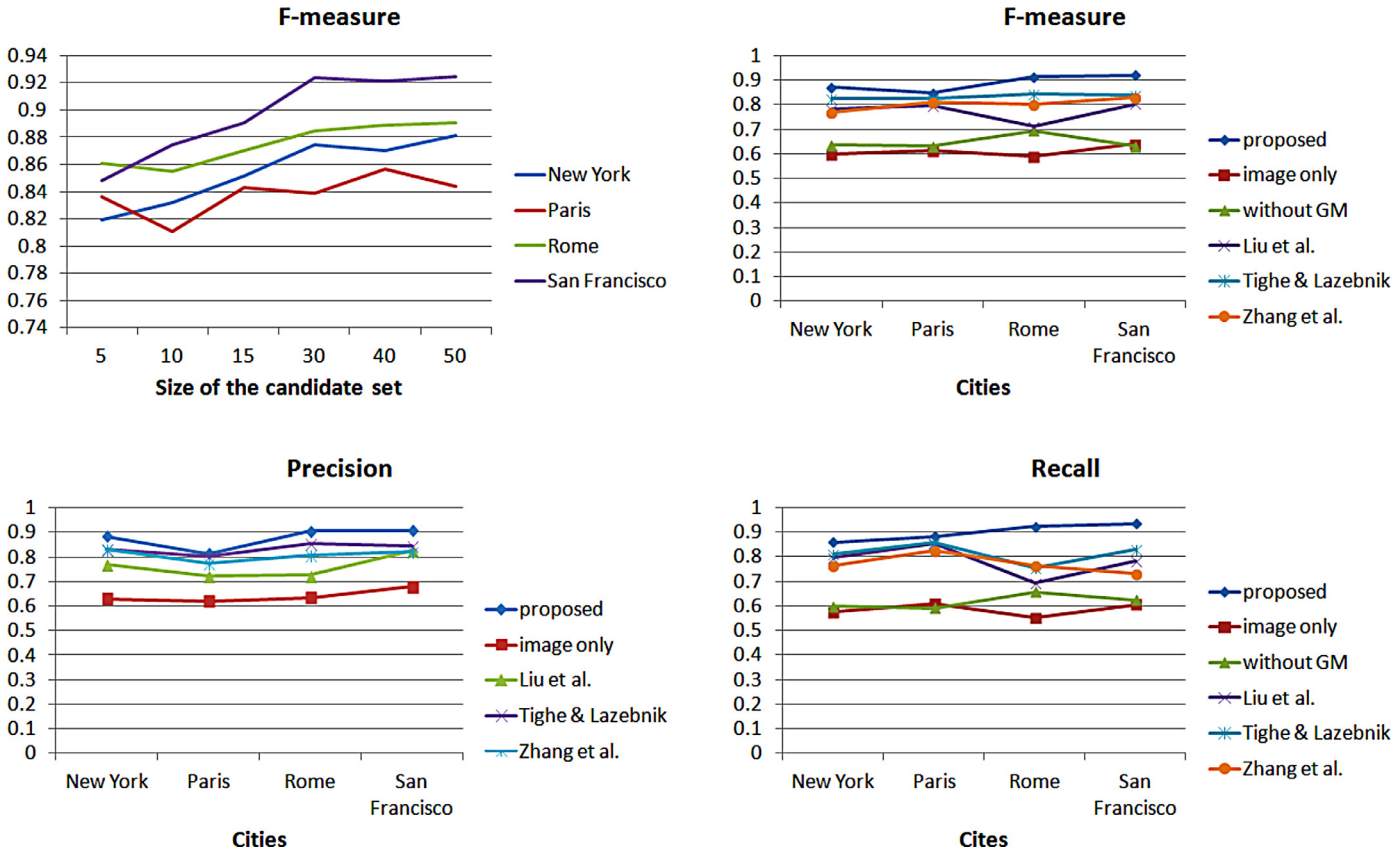


Fig. 4. Results of performance evaluation on Google dataset. The upper left shows the performance under different sizes of candidate set. The remaining 3 sub-images show the comparison in terms of F-measure, Precision and Recall. For either image, we compare our method to the MRF inference without guidance and constraints, as well as the approaches from [1], [7] and [16].

Table 2
Parsing performance of the proposed method on each category.

	Sky (%)	Building (%)	Window (%)	Door (%)	Vehicle (%)	Pedestrian (%)	Road (%)	Tree (%)	Sign (%)
Precision	96.6	93.5	52.2	47.8	90.7	74.3	94.9	87.2	59.0
Recall	89.3	87.5	55.4	56.5	92.7	70.7	91.3	90.6	49.8
F-measure	92.8	90.4	53.8	51.8	91.7	72.5	93.1	88.9	54.0

it was also difficult to identify doors in point clouds, because they can be made of different materials, such as glass, metal, wood or plastic. Moreover, obstacles behind the glass windows or doors can affect the parsing results. Fortunately, most of the regions with other labels can be parsed with a high precision.

In addition to the difficulty in recognizing windows and doors, the proposed method also had trouble in finding small or tiny objects. This is because some tiny structures are smaller than the sizes of superpixels. In this case, the tiny objects either are amplified in the parsing results, or simply disappeared. This problem can be decreased by choosing smaller size of superpixels, but leading to lower computational efficiency.

3.3. Comparison test on Google data

A comparison test was conducted for the performance of the proposed approach and the traditional MRF inference without graph matching guidance, which uses images alone (image only). This is to test the tradition Classification+MRF framework on our dataset. Further, we test the performance obtained from point clouds and images, but without graph matching (without GM). We anticipate that this test can evaluate how much improvement the graph matching guidance procedure can provide.

We also compared the performance with the approaches [1], [7] and [16] using our datasets. Liu et al., [1] is chosen because it is a typical method using information from pixels, while [7] uses the clues from superpixels. Zhang et al. [16] computes the depth map first, then uses the 3D information to help the inference. The dataset set-up was the same as that in the Section 3.2. The upper right, lower left and lower right of Fig. 4 show the comparative results of the overall performance. Some typical results are shown in Fig. 5.

Compared to the traditional MRF inference, the parsing performance was remarkably enhanced when information from both images and range data were used. The existence of the range data significantly reduced the ambiguity in inference using image alone. Besides, according to our experiment, the graph matching procedure can indeed improve the parsing performance by introducing more stringent constraints on MRF edges. In most cases, the method integrated with range data increased the accuracy in terms of F-measure, Precision and Recall by 20–30% over the basic MRF inference, and surpassing the method without GM by 15–20%. The approach proposed in [1] did not perform well on our datasets,

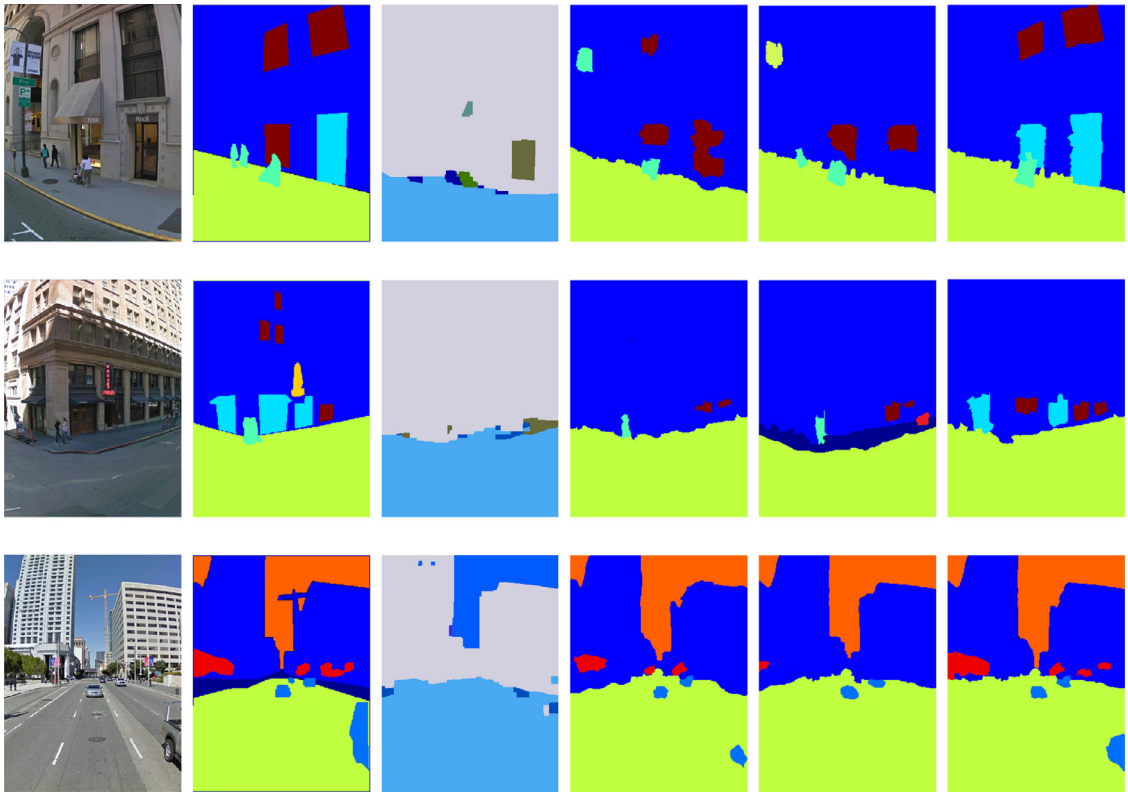


Fig. 5. Some typical parsing results. The first column corresponds to the original images. The second column contains the ground-truth (manual) segmentation. From the third column to the sixth column, there list the parsing results using the methods from [1], [7], [16] and ours.

reaching accuracies of only around 70–80% in the four datasets, due to the lack of information from the point clouds and the fact that the method in [1] required a very large set of training samples for parsing. In the experiments conducted in [1], thousands of images were selected and labeled as the training set to ensure that similar scene as query image exists. Although the scenes in our database are similar, the displacement of similar objects across scenes is very large, this is the case that [1] cannot handle. However, by handling the distortions via graph matching, our method required only hundreds of images for training, yet achieved better performance than approaches using images alone. As shown in Fig. 5, the method in [1] is able to find large structures such as sky, buildings and roads. However, small ones such as trees and pedestrians cannot be effectively interpreted since the dataset is small and sufficiently similar scenes cannot be found. Specifically, windows and doors can hardly be identified using the method from [1].

The approach in [7] introduced superpixels and thus performs better than that of [1] with our datasets. The results show that accuracies of over 80% on New York and Paris and approximately 85% on the Rome and San Francisco were obtained. However, our method still outperform the approaches in [7] by 5–10%. While [7] employed many local and global features (over 1000 dimensions on local feature and over 5000 dimensions of global features), the proposed method needed only 233 and 2472 dimensions, respectively. However, Fig. 5 demonstrates that the method from [7] still cannot handle the areas of windows and doors since a perfect classifier cannot be trained using our datasets.

Since the method proposed in [16] tried to construct the depth maps for each scene, we follow this framework by providing each scene with several images taken sequentially using the same camera. However, because the time interval between neighboring time stamps was not sufficiently dense, it was difficult to obtain the precise 3D information. As a result, the estimated depth maps merely provided limited information, or even wrong 3D information. Once the precise 3D information is unavailable, the method proposed in [16] will degenerate into a normal MRF inference scheme. In general, [16] reached accuracy between 75–83%. In Fig. 5, the parsing performance of the method from [16] heavily relies on the estimation of the depth maps. As in the third row and the fifth column of Fig. 5, the method from [16] has a poor performance when objects are too far from the view point. In this case, the estimated depth maps are almost impossible to provide reliable 3D information. This result demonstrate that using range data directly is more reliable than using 3D information estimated from structure from motion algorithms. It is supported by the experiments that by introducing graph matching on both images and range data, our method has more reliable parsing performance using less features from smaller training sets.

3.4. Comparison test on LM+SUN

In this part, we follow the selection of the algorithms as in previous section. The scenes in this database is much more complex than in the Google database, since it not only contains the street view images, but also a large variety of outdoor scenes. The number of categories is much larger and compositions of the image components are various.

The parsing performance in terms of Precision, Recall and F-measure is shown in Fig. 7. Some typical parsing results are shown in Fig. 6. The algorithm in [7] holds the best overall performance on Precision, reaching over 81.6%. The overall Precision of the proposed framework is lower than [7] with 76.6%. However, our method has a better Recall performance than [7]. This is due to the selection of different superpixel segmentation strategies. Since [7] adopts a coarser superpixel segmentation method,

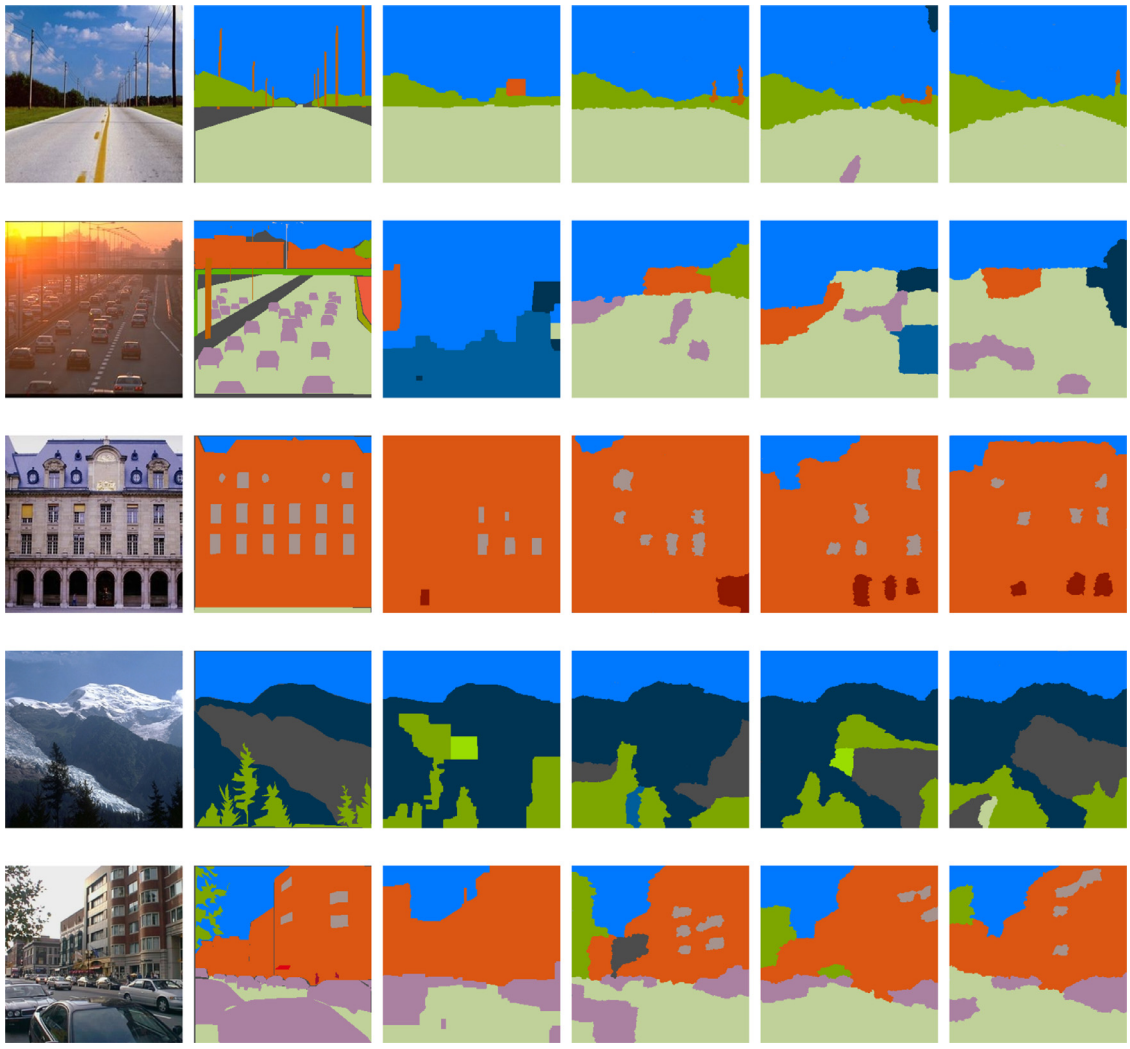


Fig. 6. Some typical parsing results on LM+SUN dataset. The first column corresponds to the original images. The second column contains the ground-truth (manual) segmentation. From the third column to the sixth column, there list the parsing results using the methods from [1], [7], [16] and ours, respectively.

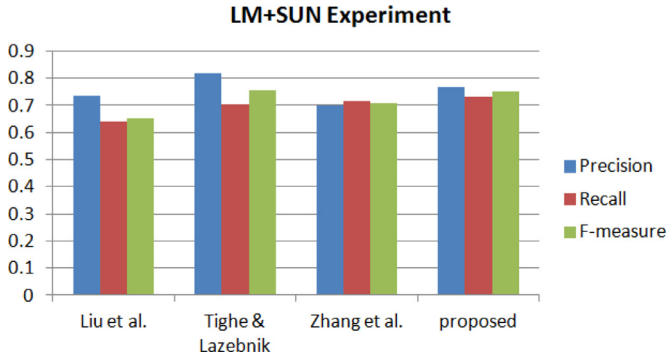


Fig. 7. Parsing performance on LM+SUN dataset.

it cannot effectively separate small objects or connected regions between two objects from two different categories. This leads to some misclassification on such regions. Instead, by utilizing a finer superpixel segmentation method, our algorithm can distinguish the boundary regions to some extent, though sacrificing some Precision performance. Liu et al. [1] has a similar Precision as the proposed method, since it parses the scenes by matching images at pixel level. Zhang et al. [16] degenerates to traditional Classification+Smoothing framework when depth information is unavailable. Note that in the second row of Fig. 6, no method achieves a good parsing performance. For [1], it's difficult to find a similar scene as the query image, so the SIFT-flow procedure will try to collect the label information from some unreliable training sample. Since all the other methods are based on training and classification, the performance can heavily rely on the precision of the classifier. Unfortunately, this query image does not follow the category distribution in the training sample space, as one can observe that the overall color of the image tends to be red. In this case, a graph matching procedure can correct the classification to some extent by matching images partially.

4. Conclusion

In this paper, we present a novel approach for street scene parsing based on fusion of images and range data and graph matching. To the best of our knowledge, this researc\ h is the first study that has employed graph matching to generate a guidance in scene parsing problem, which helps find similar higher level visual features across scenes. Our method does not need a large training set and can significantly enhance the parsing performance by integrating 3D information from the point cloud data.

The next step in our research will be the realization of the recognition of windows and doors on the building facade by

integrating a building structure interpretation scheme into the framework. Another area for future research will be the collection of cues from different scenes and development of a learning method to combine them together, in order to overcome the challenge that only part of two scenes can be accomplished with graph matching. More weight information from several training scenes will be utilized through a learning strategy.

Acknowledgments

This research is supported by Google Faculty Research Award program 2014–2015.

References

[1] C. Liu, J. Yuen, A. Torralba, Nonparametric scene parsing: Label transfer via dense scene alignment, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, IEEE, 2009, pp. 1972–1979.
[2] B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, Labelme: a database and web-based tool for image annotation, Int. J. Comput. Vis. 77 (1-3) (2008) 157–173.
[3] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Scene parsing with multiscale feature learning, purity trees, and optimal covers, ICML, arXiv preprint arXiv:1202.2160 (2012).
[4] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1915–1929.
[5] B. Douillard, A. Brooks, F. Ramos, A 3d laser and vision based classifier, in: Proceedings of 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009, IEEE, 2009, pp. 295–300.
[6] B. Russell, A. Efros, J. Sivic, B. Freeman, A. Zisserman, Segmenting scenes by matching image composites, in: Advances in Neural Information Processing Systems, 2009, pp. 1580–1588.
[7] J. Tighe, S. Lazebnik, Superparsing: scalable nonparametric image parsing with superpixels, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 352–365.
[8] D. Eigen, R. Fergus, Nonparametric image parsing using adaptive neighbor sets, in: Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)„ IEEE, 2012, pp. 2799–2806.
[9] X. He, R.S. Zemel, D. Ray, Learning and incorporating top-down cues in image segmentation, in: Computer Vision–ECCV 2006, Springer, 2006, pp. 338–351.
[10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.
[11] X. Ren, J. Malik, Learning a classification model for segmentation, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003, IEEE, 2003, pp. 10–17.
[12] J. Weng, T.S. Huang, N. Ahuja, Motion and Structure from Image Sequences, Springer Publishing Company, Incorporated, 2012.
[13] G.J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and recognition using structure from motion point clouds, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 44–57.
[14] J. Xiao, L. Quan, Multiple view semantic segmentation for street view images, in: Proceedings of IEEE 12th International Conference on Computer Vision, 2009, IEEE, 2009, pp. 686–693.
[15] J. Tighe, S. Lazebnik, Finding things: Image parsing with regions and per-exemplar detectors, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3001–3008.
[16] C. Zhang, L. Wang, R. Yang, Semantic segmentation of urban scenes using dense depth maps, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 708–721.
[17] G. Zhao, X. Xiao, J. Yuan, Fusion of Velodyne and camera data for scene parsing, in: Proceedings of the15th International Conference on Information Fusion (FUSION), 2012, IEEE, 2012, pp. 1172–1179.
[18] G. Zhao, X. Xiao, J. Yuan, G.W. Ng, Fusion of 3d-lidar and camera data for scene parsing, J. Vis. Commun. Image Represent. 25 (1) (2014) 165–183.
[19] C.J. Taylor, A. Cowley, Fast scene analysis using image and range data, in: Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 3562–3567.
[20] S. Ardeshir, K.M. Collins-Sibley, M. Shah, Geo-semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2792–2799.
[21] S. Wang, S. Fidler, R. Urtasun, Holistic 3d scene understanding from a single geo-tagged image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3964–3972.
[22] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, L. Quan, Image-based façade modeling, in: ACM Transactions on Graphics (TOG), 27, ACM, 2008, p. 161.
[23] F. Alegre, F. Dellaert, A probabilistic approach to the semantic interpretation of building facades, in: Proceedings of CIPA International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres, 2004, pp. 25–27.
[24] A. Toshev, P. Mordohai, B. Taskar, Detecting and parsing architecture at city scale from range data, in: Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 398–405.
[25] R. Wang, J. Bach, F.P. Ferrie, Window detection from mobile lidar data, in: Proceedings of 2011 IEEE Workshop on Applications of Computer Vision (WACV), IEEE, 2011, pp. 58–65.
[26] T.S. Caetano, J.J. McAuley, L. Cheng, Q.V. Le, A.J. Smola, Learning graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 31 (6) (2009) 1048–1058.
[27] L. Torresani, V. Kolmogorov, C. Rother, Feature correspondence via graph matching: models and global optimization, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 596–609.
[28] O. Duchenne, F. Bach, I.-S. Kweon, J. Ponce, A tensor-based algorithm for high-order graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 33 (12) (2011a) 2383–2395.
[29] O. Duchenne, A. Joulin, J. Ponce, A graph-matching kernel for object categorization, in: Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011b, pp. 1792–1799.
[30] S. Gould, Y. Zhang, Patchmatchgraph: Building a graph of dense patch correspondences for label transfer, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 439–452.
[31] M. Cho, K.M. Lee, Progressive graph matching: making a move of graphs via probabilistic voting, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 398–405.
[32] M.A. Fischler, O. Firschein, Readings in Computer Vision: Issues, Problem, Principles, and Paradigms, Morgan Kaufmann, 2014.
[33] R.B. Rusu, N. Blodow, Z.C. Marton, M. Beetz, Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009, IEEE, 2009, pp. 1–6.
[34] Y. Ioannou, B. Taati, R. Harrap, M. Greenspan, Difference of normals as a multi-scale operator in unorganized point clouds, in: Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), IEEE, 2012, pp. 501–508.
[35] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[36] S.T. Birchfield, S. Rangarajan, Spatiograms versus histograms for region-based tracking, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, 2, IEEE, 2005, pp. 1158–1163.
[37] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the Computer vision, IEEE International Conference on, 2, IEEE, 1999, pp. 1150–1157.
[38] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Int. J. Comput. Vis. 60 (1) (2004) 63–86.
[39] C. Doersch, S. Singh, A. Gupta, J. Sivic, A. Efros, What makes paris look like paris? ACM Trans. Gr. 31 (4) (2012).