Enhancing scene parsing by transferring structures via efficient low-rank graph matching

Tianshu Yu
University of Calgary
2500 Univ. Dr NW
Calgary, Canada
yut@ucalgary.ca

Ruisheng Wang
University of Calgary
2500 Univ. Dr NW
Calgary, Canada
ruiswang@ucalgary.ca

ABSTRACT

Scene parsing has attracted significant attention for its practical and theoretical value in computer vision. A typical scene parsing algorithm seeks to densely label pixels or 3-dimensional points from a scene. Traditionally, this procedure relies on a pre-trained classifier to identify the label information, and a smoothing step via Markov Random Field to enhance the consistency. LabelTranfer is a category of scene parsing algorithms to enhance traditional scene parsing framework, by finding dense correspondence and transferring labels across scenes. In this paper, we present a novel scene parsing algorithm which matches maximal similar structures between scenes via efficient low-rank graph matching. The inputs of the algorithm are images, and well-aligned point clouds if available. The images and the point clouds are processed in separate pipelines. The pipeline of images is to learn a reliable classifier and to match local structures via graph matching. The pipeline of point clouds is to conduct preliminary segmentation and to generate feasible label sets. The two pipelines are merged at inference step, in which we elaborate effective and efficient potential functions. We propose a new graph matching model incorporating low-rank and Frobenius regularization, which not only guarantees an accurate solution, but also provides high optimization efficiency via an eigen-decomposition strategy. Several challenging experiments are conducted, showing competitive performance of the proposed method compared to state-of-the-art LabelTransfer algorithm. Further, with point clouds, the performance can be significantly enhanced.

CCS Concepts

•**Computing methodologies** → **Scene understanding;** *Computer vision;*

Keywords

scene parsing; graph matching; semantic segmentation

SIGSPATIAL'16, October 31-November 03, 2016, Burlingame, CA, USA

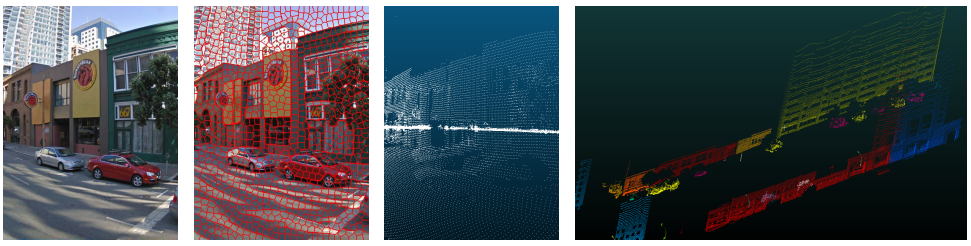
DOI: http://dx.doi.org/10.1145/2996913.2996956

1. INTRODUCTION

In the computer vision field, scene parsing amounts to simultaneously segmenting and labeling an image pixel-by-pixel with the categories which the pixel belongs to. As a challenging task, scene parsing requires to solve a detection, a segmentation and a multi-category recognition in a uniform framework. A reliable scene parsing is very powerful in varying areas, such as image editing, autonomous vehicle and video surveillance. For example, scene parsing is an essential function in autonomous driving to understand the traffic environment and to help to guide the action in next step. To handle general scenarios, scene parsing always needs a large amount of manually labeled data for training, which gives rise to the development of corresponding databases and labelling tools, such as LabelMe [21].

It is straightforward to consider labelling as a classification problem, since the massive manually labelled data provides sufficient training samples. Particularly, with the advent of deep learning, many relevant approaches have been proposed based on convolutional neural networks (CNN) or recursive neural networks (RNN) [19, 33, 6, 7, 1], achieving better performance than traditional classifiers [25, 31]. By combining a smoothing procedure via MRF, classification-based methods take into account the label consistency to some extent. Higher order potentials can also be integrated into MRF to represent more complex relations. However, the optimization of higher order MRF is so time-consuming that it's not applicable in real world data. Label-transfer is another strategy that takes into account the semantic relations of the objects [14, 17, 16]. Since scenes always show several specific patterns, it's anticipated that the semantic relations are more reliable than a single feature. With the development of Lidar sensors, 3D information of a scene can be directly obtained, which can provide more cues for the parsing task.

In this paper, we are interested in the case when images and corresponding point clouds are both available. However, our method can also be applied when only images are provided. Specifically, similar to the existing methods [14, 25, 17], we first retrieve a subset from training scenes, but holistic appearance of both images and point clouds is utilized. Since the density of point clouds is sparse and not uniform, we cannot obtain deterministic features. Instead, an initial segmentation is implemented, to exclude impossible categories and to calculate a prior for further inference. Our purpose is to transfer co-occurred structures from the retrieved scenes to the query scene. This is realized by over-segmenting scenes into superpixels, and finding maximal similar subgraphs across scenes via graph matching. To ease



(a) (b) (c) (d)

Figure 1: (a) The street-level image; (b) The over-segmented image with superpixels; (c) The corresponding point cloud in camera view; (d) The corresponding segmented point cloud. The outliers and ground points have been removed.

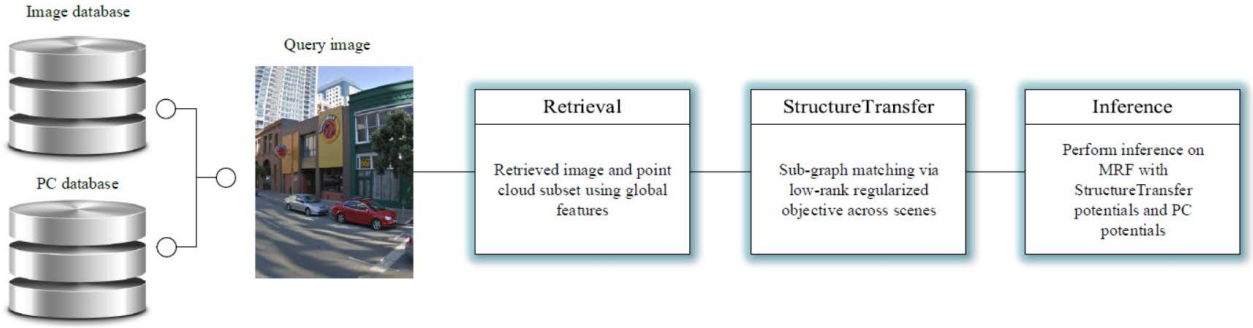


Figure 2: The workflow of the proposed parsing framework. "PC" refers to point cloud.

the optimization, we propose a new graph matching model, which can be optimized by using an eigen-decomposition strategy. Finally, along with the prior and transferred structures, the inference is performed on an enhanced MRF. A schematic diagram of the workflow is depicted in Fig 2.

There are three contributions of this paper including: (1) We present a structure-transfer framework which can be applied with many graph matching algorithms; (2) A new low-rank regularized graph matching model with high computational efficiency is introduced; (3) Segmentation of sparse point clouds is utilized as a prior for inference, and an efficient scene retrieval scheme on both images and point clouds is proposed. If using images alone, the performance of the proposed method is very close to state-of-the-art algorithms. However, with point clouds, the proposed method significantly improves the parsing accuracy.

2. RELATED WORK

There are generally two strategies of scene parsing algorithms, classification-smoothing-based [19, 33, 7, 1, 25] and label-transferring-based [14, 17, 16]. There are also some alternatives incorporating both strategies. The classification-smoothing-based method mainly consists of two phases as its name shows. In the classification phase, a classifier is trained to label regions by using the local appearance, which is abstracted as local descriptors. Varying classifiers can be introduced, such as SVM [25], random forest [31] and learning-based features [19, 33]. The selection of descriptors and classification schemes can remarkably influence the performance. In the smoothing phase, an MRF/CRF inference procedure is employed which can smooth the labeling by taking into account the neighboring label consistency. In this procedure, isolated pixels with irregular labels are smoothed by its neighboring labels, which is based on the fact that in natural scenes, single pixel object barely exists. Very recently, the relation of classification and smoothing has been revealed, and the two phases can be unified into a single RNN framework [33], achieving state-of-the-art performance. This model takes advantages of CRF with Gaussian potentials which can be efficiently optimized via Mean Field inference [11]. Since the training and pixel-wise classification are time-consuming, some alternative methods are proposed working on superpixels [25, 31]. Superpixels are also used as cues for pixel-level segmentation [1]. The main drawback of classification/smoothing frameworks is that, the training of classifier needs massive correctly labeled data, while it's not always possible in real-world applications. It's natural that human can identify a scene given only limited training samples. In other words, human parses a scene by capturing structural features and classifying local features, since structural features can represent semantics of the objects more comprehensively. This observation motivates the second type of parsing algorithms: label-transferring [14, 17, 16, 32]. Label-transferring methods focus on finding regional correspondences between training and testing samples. Once the correspondence is obtained, the algorithms transfer the label in training pixel to the corresponding testing pixel with a high belief. The correspondence can be holis-

tic [14] or partial [17, 16, 32, 10], which means that the matching can be from a whole image to another whole image, or a partial structure to another. In [14], the pixel-wise dense correspondence is established via SIFT-FLOW [15]. However, as a holistic strategy, this method cannot handle large displacement. Unlike holistic label-transfer, partial methods take into account only a small collection of pixels, with the constraint of label consistency. The constraint can be second-order [16, 10] or higher-order [17] reflecting the semantic relations of the objects. Further, similar regions containing much higher evidence can be identified to boost the recognition of co-occurence [26, 30]. The method in [26] finds the most similar match contained in bounding boxes, while [30] introduces subgraph matching to handle more complex situations. In this paper, we focus on partial-matching-based transfer, which is assumed to be more reliable since two scenes cannot be totally similar. By considering scenes as graphs, the partial matching finds the maximal similar sub-graph across scenes.

Due to the combinatorial nature, it's NP-hard to solve graph matching exactly. One effective relaxation is to assume that the solutions lie in an interval rather than just integers, yielding a convex problem. This relaxed problem can be solved using spectral method [13, 5] or random walk [4]. Another common relaxation to graph matching is semi-definite programming (SDP) [23]. Very recently, a fast algorithm solving graph matching in SDP framework has been proposed by exploiting the low-rank property and introducing Frobenius norm [27]. In this paper, we reformulate the sub-graph matching problem by introducing multiple regularizations, and optimize it via [27].

Rather than using images alone, some alternative methods are proposed by introducing 3D information [31, 2, 24]. Different from images, 3D information can provide abundant 3D geometric information, such as depth, height and reflection intensity. In [31], depth information is estimated from multiple views by using the structure-from-motion method. Then a classification/smoothing framework is utilized regarding depth and image-based descriptors as training inputs. The estimation of 3D information is computationally intensive and imprecise. On the other hand, the application of point clouds directly captured from Lidar sensors is more convenient and exact [2, 24]. In this paper, we also present a novel parsing algorithm incorporating images and point clouds.

2.1 Preliminaries and Point Cloud Segmentation

2.2 Preliminaries

The data from four cities (i.e. New York, Rome, San Francisco and Paris), including images and point clouds at street level, is provided by Google. The images are taken by shuttering cameras, and the point clouds are collected by SICK LiDAR sensors.

Images and point clouds should be aligned first. If there's no position information, the alignment can be conducted by using the appearance of both images and point clouds [28], which is out of the scope of this paper. With the given information, we transform point clouds into camera view taking into account the perspective model and camera distortions. All training and query images are over-segmented into superpixels using [20]. An example showing the superpixels and corresponding point cloud is shown in Fig 1 (b) and (c), respectively. We also manually label the training images using LabelMe [21]. To label a superpixel, the category with the largest proportion of pixel numbers in this superpixel is assigned.

2.3 Point Cloud Segmentation

The step is to segment point clouds into separate objects, which can provide more evidence on how the elements of the scenes are related. We employ the same segmentation strategy as in [30], which first identifies the ground points using piece-wise RANSAC [9], then separates the objects by means of Euclidean distance clustering [22]. After the above procedure, the objects that lie away from each other can be effectively separated.

Next we calculate feasible label set for each superpixel based on the strategy in [30]. Rather than providing precise classification or a probability distribution, this procedure assigns a superpixel with an initial feasible label set, which provides to which category a superpixel likely belong. This function is realized via a rule-based exclusion. Since our dataset with point clouds is at street-level, we can first find the ground points via height and vertical normals. The building facades are then identified by recognizing the horizontal normals and the density of the projected sample points onto the ground plane. Sometimes trees and building facades may attach, however, we still can separate them to some extent by performing the PCA, as facades generally have two principle directions, while trees have three. It can be difficult to distinguish pedestrians and cars, because they can be similar in point clouds. We exclude the impossible categories for such objects, and assign a uniform distribution to the rest labels. Fig 1 (d) shows the initial segmentation of the point cloud corresponding to Fig 1 (a). After assigning the feasible label set, one can obtain an initial label distribution for each superpixel. Assume there are n superpixels in a scene, and $l_i \in \{1, ..., R\}$ is the label of the superpixel i, where R is the number of labels. Then the feasible label set for the superpixel i is defined as:

$$\mathcal{F}_i = \{\text{the feasible labels of } l_i\} \tag{1}$$

We further denote $B(l_i)$ the initial label distribution vector of the superpixel i. However, on some nodes, the probabilities of some specific categories can be 0, which may not be a good guess. To assign each category with some possibility, we adjust $B(l_i)$ by $B(l_i) = B(l_i) + \Delta$, where $\Delta > 0$ is a small value. We then normalize $B(l_i)$ to make it a probability distribution. The value of Δ is according to the size of the label set, to ensure that the probability of each label is above 0. In all the experiments in this paper, $\Delta = 0.01$.

3. STRUCTURE TRANSFER VIA GRAPH MATCHING

3.1 Problem Formulation

We aim at finding the maximal similar regions across scenes, which is with both local appearance similarity and higher order label consistency. Typically, this can be realized via any graph matching procedures. To exploit the low-rank feature and to accelerate the computation, we present a novel model. To this end, we first present the graph matching problem over superpixels in the following.

Suppose that there are R labels in total. Given training image I^1 with full annotation and query image I^2, Let S^1 and S^2 be two sets of superpixels generated from the two images, and n^1 and n^2 be the number of the superpixels in I^1 and I^2, respectively. We construct two graphs $\mathcal{G}^1 = \left(\mathcal{N}^1, \mathcal{E}^1, \mathcal{A}^1\right)$ and $\mathcal{G}^2 = \left(\mathcal{N}^2, \mathcal{E}^2, \mathcal{A}^2\right)$ by connecting edges between neighboring superpixels, where $\mathcal{N}_i$ denotes the node i of the graph, $\mathcal{E}_{ij}$ denotes an edge connecting nodes i and j, and $\mathcal{A}_{ij}$ represents the attributes assigned to edge $\mathcal{E}_{ij}$. Note that for a node $\mathcal{N}_i$, the corresponding attribute can be expressed as $\mathcal{A}_i = \mathcal{A}_{ii}$, which represents the feature of the superpixel i. The attribute can be any features for the superpixel and the edge. By using the attribute, one can calculate the edge similarity $W_{ai:bj}$, which measures the edge consistency of $\mathcal{E}_{ab}$ from $\mathcal{G}^1$ and $\mathcal{E}_{ij}$ from $\mathcal{G}^2$. For the integrity, we also use $W_{ai:ai}$ to represent the similarity of node a from $\mathcal{G}^1$ and node i from $\mathcal{G}^2$.

Since structure transfer is to find similar partial regions across scenes, we denote $x \in \{0,1\}^{n^1 n^2}$ representing the matchings between the two graphs (images) over nodes (superpixels). Identifying maximal similar regions can be expressed as a graph matching problem:

$$\begin{aligned} &\min_x x^T W x \\ \text{s.t.} \quad &x \in \{0,1\}^{n^1 n^2} \\ &X' \mathbf{1}_{n^1 \times 1} \leq \mathbf{1}_{n^2 \times 1} \\ &X'^T \mathbf{1}_{n^2 \times 1} \leq \mathbf{1}_{n^1 \times 1} \end{aligned} \tag{2}$$

where $X' \in \{0,1\}^{n^1 \times n^2}$ denotes the matrix form of x, and $\mathbf{1}_n$ denotes a length n vector with all 1 element. It's NP-hard to solve the problem (2) due to its combinatorial nature. In this paper, instead, we use the SDP relaxation to obtain an approximate solution as follows:

$$\begin{aligned} &\min_X \operatorname{tr}(XW) \\ \text{s.t.} \quad &X \geq 0 \\ &\operatorname{tr}(XA_i) = h_i \\ &\operatorname{tr}(XB_j) \leq g_j \end{aligned} \tag{3}$$

where $X \geq 0$ means that $X \in [0,1]^{n^1 n^2 \times n^1 n^2}$ is positive semidefinite. tr refers to the trace of a matrix. We also tacitly assume that X is symmetric. W is the similarity matrix, and A_i and B_j correspond to the equality and inequality constraints, respectively. In this paper, we introduce all the constraints in [29], which reflect the relation between X and x.

3.2 Objective Function

Though the previous reformulation (3) of graph matching relaxes the problem into a convex form, a main drawback is that SDP is computationally expensive as it squares the variable number. Another drawback is that, though X is supposed to be a rank one matrix according to its construction, the solution generated by interior point methods tends to be high rank according to existence of the barrier function. This is because that interior point methods are likely to produce solutions away from boundary of the feasible set, whereas a low-rank solution is likely to lie on the boundary.

Some methods have been proposed to approximate a low-rank solution or to accelerate the computation [27, 29, 12] by integrating new constraints or combining regularization terms. Frobenius norm, such like L^2-regularization in solving linear system, is reported to be helpful to approximate a low-rank solution, as well as to reduce the computational complexity [27]. On the other hand, nuclear norm or trace norm is proved more intuitive to generate lower rank solutions, especially for graph matching [29]. In this paper, we regard both Frobenius norm and trace norm as regularization, yielding the following objective function:

$$\min_X \operatorname{tr}(XW) + \lambda \|X\|_* + \gamma \|X\|_F^2 \tag{4}$$

where $\|\cdot\|_*$ and $\|\cdot\|_F$ refer to the trace norm and Frobenius norm, respectively. The second term is for seeking a low-rank and sparse solution, while the third term can avoid severe vibration, as well as provide a fast solution. It is easily found that $\|X\|_* = \operatorname{tr}(X) = \operatorname{tr}(XI)$, where I is the identity matrix. Thus we can rewrite the objective (4) as:

$$\min_X \operatorname{tr}(X(W + \lambda I)) + \gamma \|X\|_F^2 \tag{5}$$

A similar regularization in linear systems is elastic net [34], which takes into account both L^1 and L^2 regularization and outperforms any single regularization model. The optimization (5) can be solved using an eigen-decomposition strategy [27]. The complexity of the this approach is $\mathcal{O}(kn^3)$, where k and n are the iteration time and number of the rows of matrix X, respectively. It saves significant computation comparing to an interior point method $\mathcal{O}(n^{6.5})$.

Some implementation details are presented here. The local descriptors for a single superpixel includes color histogram, SIFT histogram and texton histogram [30]. We also introduce the STE measurement to calculate similarity W as in [30]. Further, since the solution of (5) is not 0-1 value, a sampling strategy is necessary to obtain the matchings. In our paper, we follow the sampling scheme in [29] to obtain the final matchings.

4. SEMANTIC SCENE PARSING

4.1 Scene Retrieval

Since we aim at finding the maximal similar regions across scenes, given a query scene, it's essential to retrieve highly related ones from all training scenes. This can lead to higher possibility to locate objects of the same label. Traditional methods based on images adopt global features for the retrieval, such as GIST [18] and Spatiogram [3]. By concatenating these two features, our approach also introduce L^2-distance as measurement to retrieve most similar K images from the dataset. Define the candidate scene set $\mathcal{I}^Q$.

Aside from images, it's natural for our system to integrate spatial features collected from the point clouds into the retrieval. To the best of our knowledge, there is no off-the-shelf scene retrieval algorithm using point clouds. In this paper, rather than developing a complicated system, we introduce an easy but effective strategy. For each superpixel, we calculate its depth to the camera and height to the ground, generating a depth map and a height map. The depth of a superpixel is defined by the a 3D point with the shortest distance to camera, among all 3D points within this superpixel. Similarly, the height of a superpixel is identified by

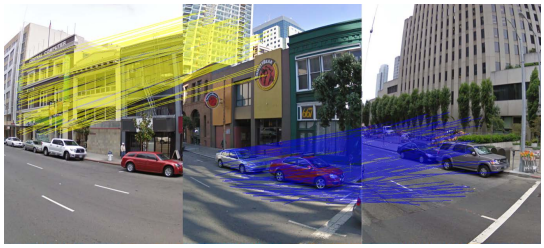


Figure 3: Structure-transfer via sub-graph matching. The image in the center is the query image, while the images on two sides are training samples. Structure-transfer seeks to transfer label across scene with higher confidence.

the height of the nearest point to the camera within the superpixel. For infinitely distant points, such as sky or regions without 3D points, we simply omit them. After centralizing the points, we calculate two histograms, one for depth and one for height, and concatenate them into one spatial feature. We use this feature to retrieve L similar scenes in terms of L^2-distance. Denote $\mathcal{I}^P$ the candidate set with respect to the spatial information.

The final candidate set is defined as: $\mathcal{I} = \mathcal{I}^Q \cup \mathcal{I}^P$. And the number of the candidates is $M \leq K + L$. These candidates are selected as the structure transfer subjects with respect to the query scene. Union is adopted here rather than intersection because we find that intersection sometimes leads to very small candidate set $\mathcal{I}$. This is because, similar images do not necessarily share similar point cloud structures.

4.2 Inference

After obtaining M candidate scenes, we perform the graph matching as described in section 4 to transfer regions from training scenes to the query scene, which is to find the latent label for each query superpixel with high belief. The retrieval step is necessary because the graph matching procedure can still be performed even if two scenes are irrelevant, which possibly leads to incorrect partial matching. For a given candidate $k \in \{1, ..., M\}$, suppose C_k is the optimal objective value of (4) for the kth candidate, where $k \in \{1, ..., M\}$. Also suppose x^{k*} is the kth optima. According to x^{k*}, the query scene partially inherits the (first-order) labels and (second-order) edge relations from the kth training scene. For a superpixel i in the query scene, we define a vector $\mathbf{L}_k(i)$. If the label of the corresponding node in the training scene belongs to the fth category, the fth element of $\mathbf{L}_k(i)$ is set to be 0, otherwise 1.

To exploit the information gathered from point clouds and images, we build up a pairwise MRF model over the superpixels. Aside from conventional unary and smoothing terms, this model also includes structure transferring information and segment information from point clouds. Concretely, the energy function of the proposed model can be expressed as:

$$E = E^I + E^P \tag{6}$$

where E is the energy of the pairwise MRF, E^I and E^P represent the potentials for image field and point cloud field, respectively. In the optimization we aim to minimize energy E. E^I can be further defined as:

$$\begin{aligned} E^I(\{l_i\}) = &\sum_i^{n^2} [1 - B(l_i)] E^u(l_i) + \sum_{(i,j)\in\mathcal{E}^2} E^s(l_i, l_j) \\ &+ \sum_{k=1}^{M} \left\{ \sum_{i\in\mathcal{N}_k^t} E_k^t(l_i) + \sum_{(i,j)\in\mathcal{E}_k^t} E_k^t(l_i, l_j) \right\} \end{aligned} \tag{7}$$

where $l_i \in \{1, ..., R\}$ is the label for superpixel i. B is the initial distribution according to the feasible label set from point cloud segmentation. Here we subtract B from 1 to assign a larger weight to the label with smaller initial probability. $\mathcal{N}_k^t$ and $\mathcal{E}_k^t$ are the node subset and the edge subset in the query scene that matches the kth candidate, respectively. Given a local feature of a superpixel i, $E^u(l_i)$ is a common classifier output, and $E^s(l_i, l_j)$ is the pairwise potential. We employ the same data term E^u and smoothness term E^s as in [31]. E_k^t is defined based on the structure transferring result. This term is to collect label information from similar structures in the candidates, and to evaluate the label by the optimal energy. If we denote $\mathbf{E}_k^t \in \Re^R$ the vector form of $E_k^t(l_i)$, it can be expressed as:

$$\mathbf{E}_k^t(i) = \frac{1}{Z} e^{-C_k} \mathbf{L}_k(i) \tag{8}$$

where $Z = \sum_k e^{-C_k}$ is a normalization parameter. This term incorporates the first-order information that the query scene inherits from a candidate scene. For an edge (i, j) in the query scene, if we assume that the matched edge in the kth candidate is (a_k, b_k), then the second-order form of E_k^t is defined as:

$$E_k^t(l_i, l_j) = \frac{1}{Z} e^{-C_k} \{1 - [l_i = l_{a_k}] \times [l_j = l_{b_k}]\} \tag{9}$$

This term penalizes the case when the labels of an edge in the query scene do not agree with the matched ones in the candidate k. In summary, E_k^t is essential to integrate structure transfer result into the inference. According to the definition of graph matching, low energy implies that the matched regions are more similar. Thus we amplify the belief that one superpixel belongs to label l_i, if the corresponding energy is low. Fig 3 shows the structure-transfer across images, where partial matchings are established via sub-graph matching procedure. Typically, sub-graph matching finds correspondence across scenes, and the correspondence between two images can be on a collection of objects. The final transferring potential is accumulated from all the matched candidates.

On the other hand, E^P reflects the relation obtained from point cloud segmentation. Superpixels belonging to the same point cloud segment, tends to share the same label. This is a natural observation, since point cloud segmentation identifies objects if they are spatially separate. Hence a single cluster of points can be highly possible in the same category. E^P is defined as:

$$E^P(l_i, l_j) = \begin{cases} [l_i \neq l_j]_{(i,j)\in\mathcal{E}^2} \times e^{-\delta \|\mathcal{A}_i^2 - \mathcal{A}_j^2\|_2^2} & \text{if } \mathfrak{C} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where condition $\mathfrak{C}$ refers to $G_i = G_j$ and $l_i, l_j \in \mathcal{F}_i \cap \mathcal{F}_j$, and G_i refers to the segment that superpixel i belongs to.

This term encourages two superpixels in the same segment to share the same label when their labels are in the feasible set; otherwise we do not consider the additional label consistency. When the labels are different, the cost is inversely proportional to the feature difference. In all the experiment, we set $\delta = 0.4$.

In general, the new energy function E takes into account the maximal co-occurrence by finding similar partial scenes via graph matching, but doesn't introduce any higher-order cliques. In our settings, pairwise MRF is sufficient to model the inference framework, resulting in simpler optimization. We employ max-product belief propagation to minimize the energy function E [8].

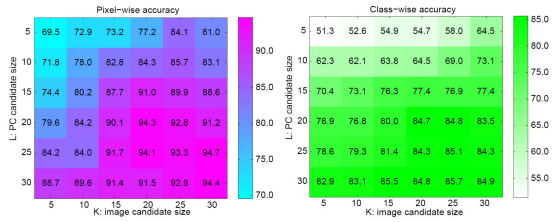


Figure 5: The pixel-wise and class-wise accuracy with varying K and L. "PC" refers to point cloud. Best viewed in color.

5. EXPERIMENTS

In this section, we carry out two experiments. The first experiment is on the Google street-view dataset, where both images and point clouds are available. We validate if the existence of point clouds and the structure-transfer can improve the parsing performance. The second experiment are performed when only images are available. This experiment includes comparison tests on SIFT-FLOW dataset [14] and Jain dataset [10], against several competitive algorithms.

Evaluation metric. The overall performance is measured using pixel-wise accuracy, which indicates the rate of correctly labelled pixels. Class-wise accuracy is also adopted in our experiments to analyze the performance on each category, which refers to the correctly labelled proportion of pixels in each category.

5.1 Parsing on Point Clouds and Images

Google street-view dataset. This dataset is provided by Google, which includes scenes from four cities. The images and point clouds are collected at street-level by using the sensors mounted on a vehicle. As the vehicle moves, the cameras and Lidar keep capturing images and point clouds, respectively. There are over 20,000 images in this dataset, with the resolution 1936×2592. 400 images are randomly selected and manually labelled into 9 categories by using LabelMe [21]. We split the selected scenes into 320 training scenes and 80 test scenes. In order to reduce the computational cost, we downsample the selected images into 484×648. For a given image, we consider its corresponding portion of the point clouds that is in the camera view.

We set $K = 20$ and $L = 20$ for both the retrieval sets of images and point clouds. After normalizing the similarity matrix W, we set $\lambda = 0.1$ and $\gamma = 0.01$ in equation (5). δ in equation (10) is set to be 0.05. To evaluate the impact of point clouds and the structure-transferring scheme, we also conduct the experiments without introducing the point cloud term and structure-transferring term. To eliminate the influence of point clouds, we remove the multiplier $[1 - B(l_i)]$ from the first term in (7), and E^P from (6). On the other hand, for testing the influence of structure-transferring, we remove the third term from (7). Further, when evaluating the performance without point clouds, we set $K = 15$ and $L = 0$, which implies that the point cloud retrieval is not active at this moment. The performance of a baseline method with only the standard data and smoothness terms is also presented.

The performance can be found in Table 1. Some example results are in Fig 4. We report the performance on four settings: a baseline algorithm without point clouds and structure transferring; one with only point clouds; one with only structure transferring; and a combined algorithm with point clouds and structure transferring. The baseline algorithm is to test what performance a standard classification and smoothing framework can reach. It's obvious that, the existence of point clouds can significantly enhance the parsing accuracy. Especially, some small objects that are difficult to separate from the background in the images, can be effectively recognized by using point clouds. The appearance of windows can remarkably vary, but they always correspond to "holes" on the building facade point clouds. Similar situation also occurs on glass doors. This situation, however, can be easily found in point clouds. On the other hand, if the appearance of an object is stable (e.g., vehicle and tree), the structure-transferring strategy can provide more semantic and organized information from the images, yielding enhancement of accuracy. Without point clouds and structure-transferring, the accuracy of the baseline method drops drastically. In general, by incorporating point clouds and structure-transferring, the overall accuracy of the proposed method reaches 94.3%, and is stable in finding objects in various categories.

We further adjust the values of K and L to evaluate the influence of the candidate sets. The experimental results can be found in Fig 5, which is designed as heatmap for better visualization. It can be observed that, along with the increment of the candidate sizes, the pixel-wise and the class-wise accuracy also increases. The performance becomes generally stable when the candidate sizes are sufficiently large. Besides, when both K and L are small, the performance is more sensitive to L.

5.2 Parsing on Images

In this section, we evaluate the parsing performance of the proposed method on two datasets: SIFT-FLOW [14] and Jain [10]. Five competitive counterparts are selected for the comparison, including LabelTransfer [14], Tighe and Lazebnik [25], Myeong et al. [17], Jain et al. [10] and CollageParsing [26]. For [17], we introduce the "sum" clique potential setting, which is the summation of the confidence score. The parsing performance in terms of pixel-wise and class-wise accuracy is summarized in Table 2.

As described in the previous experiment, since only images are available, we use the energy function as "without PC".

SIFT-FLOW dataset. This dataset consists of 2688 images (size 256×256), together with the manually labelled 33 categories. The dataset includes a large variety of natural scenes. We use the same split as in [14], in which there are 2,488 training images and 200 test images. The candidate

Table 1: The class-wise parsing performance in terms of pixel-wise and superpixel-wise accuracy. Nine categories are included in this dataset. "without PC" and "without ST" refer to without point clouds and without structure-transferring, respectively. "overall P" and "overall C" refer to the overall pixel-wise and overall class-wise accuracy, respectively.

	sky	building	window	door	vehicle	pedestrian	road	tree	sign	overall P	overall C
proposed	97.4	92.8	62.2	71.0	93.9	78.5	97.2	93.6	78.8	94.3	84.7
without PC	90.4	84.1	46.9	61.7	81.6	69.4	92.0	84.3	54.6	82.2	73.9
without ST	97.3	90.5	59.9	70.3	90.4	75.1	94.1	83.6	73.6	88.4	81.6
baseline	82.9	70.5	39.4	46.9	62.2	44.6	72.0	62.4	31.7	64.8	57.0

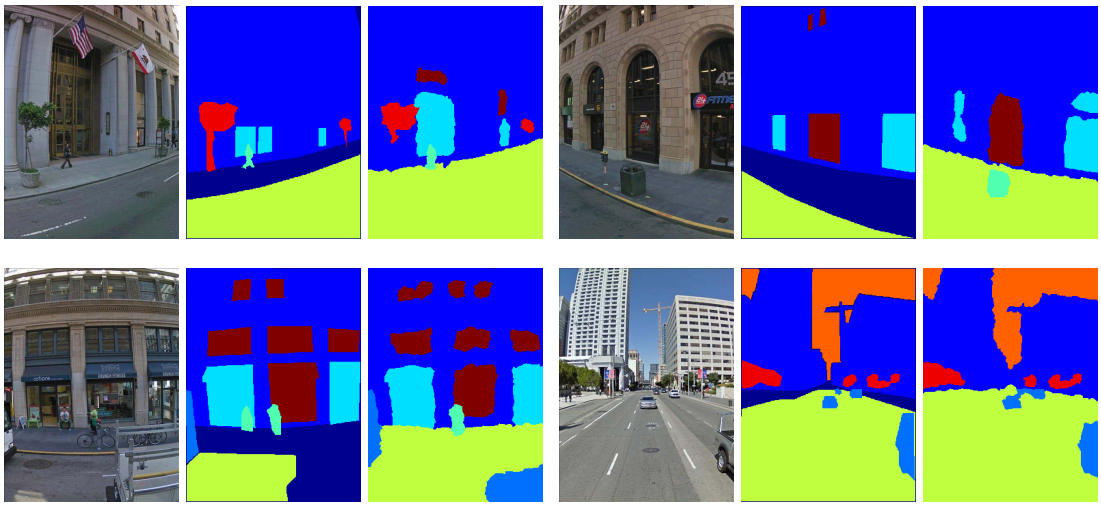


Figure 4: Parsing samples on Google street-view dataset. Three images as one group. The first and the second images correspond to the scene image and ground-truth segmentation, respectively. The third image is the parsing result with the proposed approach.

number M is set to be 30. We also set $\lambda = 0.13$, $\gamma = 0.02$ and $\delta = 0.08$.

This dataset is complex, consisting of a large variety of natural images with more categories. As reported by the experiment, the pixel-wise performance of the proposed method is very close to state-of-the-art algorithms. It can be concluded that, though without the existence of point clouds, the structure-transferring mechanism can still ensure a relatively high parsing accuracy. The class-wise accuracy is lower than [26], which is likely due to the large retrieval set utilized in [26] (400 candidates are selected in this method). Besides, since the appearance of some objects varies drastically, it brings trouble to the structure-transferring to identify similar partial regions. In general, our method achieves a promising performance on a small candidate set, reaching 78.4% and 42.3% in terms of pixel-wise and class-wise accuracy, respectively.

Jain dataset. Following the settings in [10], 350 images with totally 19 labels are randomly selected into this dataset. The image size is 640×480, and the images are split into 250 training samples and 100 test samples. In this test, the candidate size M is 20, and we set $\lambda = 0.1$, $\gamma = 0.01$ and $\delta = 0.08$.

The image size of this dataset is larger than that of SIFT-FLOW. This fact results in unstable object sizes. As the structure-transferring can be relatively sensitive to the scaling of partial scene, it cannot performance perfect matching on each candidate. The overall pixel-wise accuracy is till close to state-of-the-art algorithms. However, the proposed method achieves the best class-wise parsing performance. We think this is because, though large objects (e.g., sky, building and road) are more likely to scale drastically, the sizes of small objects are generally not. Thus our approach can work well on small objects, yielding significant class-wise accuracy by 59.3%.

6. CONCLUSION

In this paper we present a novel scene parsing approach that incorporates images and aligned point clouds. The point clouds are segmented and coarsely assigned an initial distribution, which is utilized as some kind of prior in the inference step. We also introduce structure-transferring strat-

Table 2: Comparison on two datasets against the selected algorithms. Pixel-wise performance and class-wise performance in the brackets are presented.

dataset / method	SIFT-FLOW	Jain
Liu et al. [14]	74.8(-)	-
Tighe and Lazebnik [25]	76.8(29.4)	-
Myeong and Lee [17]	76.2(29.6)	**81.8**(54.4)
Jain et al. [10]	-	59.0(-)
CollageParsing [26]	**79.9**(**49.3**)	-
proposed	78.4(42.3)	78.1(**59.3**)

egy across scenes, by casting the partial matching problem as a low-rank SDP. The structure-transferring is powerful to localize structurally similar regions between two images. The framework of the proposed approach treats images and point clouds in two pipelines, and combines them during the inference step, which makes the method flexible. In addition to the significant accuracy enhancement by structure-transferring, this approach can further improve the performance using the information from point clouds. As the point clouds are becoming more popular and available, this approach shows promising prospect in real-world applications.

7. ACKNOWLEDGMENTS

This work was funded in part by Google Research Award, 2014. Google also provided the street view dataset from four cities.

8. REFERENCES

[1] A. Arnab, S. Jayasumana, S. Zheng, and P. Torr. Higher order potentials in end-to-end trainable conditional random fields. *arXiv preprint arXiv:1511.08119*, 2015.

[2] P. Babahajiani, L. Fan, and M. Gabbouj. Semantic parsing of street scene images using 3d lidar point cloud. In *ICCVW*, pages 714–721, 2013.

[3] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *CVPR*, volume 2, pages 1158–1163, 2005.

[4] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *ECCV*, pages 492–505, 2010.

[5] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, pages 313–320, 2007.

[6] C. Farabet, C. Couprie, L. Najman, and Y. Lecun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*, pages 575–582, 2012.

[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 35(8):1915–1929, 2013.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[9] M. A. Fischler and O. Firschein. *Readings in Computer Vision: Issues, Problem, Principles, and Paradigms*. Morgan Kaufmann, 2014.

[10] A. Jain, A. Gupta, and L. S. Davis. Learning what and how of contextual models for scene labeling. In *ECCV*, pages 199–212. 2010.

[11] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[12] B. Kulis, A. C. Surendran, and J. C. Platt. Fast low-rank semidefinite programming for embedding and clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 235–242, 2007.

[13] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489, 2005.

[14] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 33(12):2368–2382, 2011.

[15] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011.

[16] H. Myeong, J. Y. Chang, and K. M. Lee. Learning object relationships via graph-based context model. In *CVPR*, pages 2727–2734, 2012.

[17] H. Myeong and K. M. Lee. Tensor-based high-order semantic relation transfer for semantic scene segmentation. In *CVPR*, pages 3073–3080, 2013.

[18] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[19] P. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, pages 82–90, 2014.

[20] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, pages 10–17, 2003.

[21] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[22] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In *IROS*, pages 1–6, 2009.

[23] C. Schellewald and C. Schnörr. Probabilistic subgraph matching based on convex relaxation. In *EMMCVPR*, pages 171–186, 2005.

[24] C. J. Taylor and A. Cowley. Fast scene analysis using image and range data. In *ICRA*, pages 3562–3567, 2011.

[25] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, pages 352–365. 2010.

[26] F. Tung and J. J. Little. Scene parsing by nonparametric label transfer of content-adaptive windows. *CVIU*, 143:191–200, 2016.

[27] P. Wang, C. Shen, and A. van den Hengel. A fast semidefinite approach to solving binary quadratic problems. In *CVPR*, pages 1312–1319, 2013.

[28] R. Wang, F. P. Ferrie, and J. Macfarlane. Automatic registration of mobile lidar and spherical panoramas. In *CVPRW*, pages 33–40, 2012.

[29] T. Yu and R. Wang. Graph matching with low-rank regularization. In *WACV*, 2016.

[30] T. Yu and R. Wang. Scene parsing using graph

matching on street-level data. *doi:10.1016/j.cviu.2016.01.004*, 2016.

[31] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In *ECCV*, pages 708–721, 2010.

[32] H. Zhang, T. Fang, X. Che, Q. Zhao, and L. Quan. Partial similarity based nonparametric scene parsing in certain environment. In *CVPR*, pages 2241–2248, 2011.

[33] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[34] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.