Joint Segmentation of Images and Scanned Point Cloud in Large-Scale Street Scenes With Low-Annotation Cost

Article *in* IEEE Transactions on Image Processing · August 2014

DOI: 10.1109/TIP.2014.2348795 · Source: PubMed

Honghui Zhang
The Hong Kong University of Science and Technology
11 PUBLICATIONS **114** CITATIONS
SEE PROFILE

Jinglu Wang
City University of Hong Kong
9 PUBLICATIONS **39** CITATIONS
SEE PROFILE

Tian Fang
43 PUBLICATIONS **686** CITATIONS
SEE PROFILE

Long Quan
The Hong Kong University of Science and Technology
194 PUBLICATIONS **5,156** CITATIONS
SEE PROFILE

Joint Segmentation of Images and Scanned Point Cloud in Large-Scale Street Scenes with Low Annotation Cost

Honghui Zhang, Jinglu Wang, Tian Fang, Long Quan

***Abstract*—We propose a novel method for the parsing of images and scanned point cloud in large-scale street environment. The proposed method significantly reduces the intensive labeling cost in previous works by automatically generating training data from the input data. The automatic generation of training data begins with the initialization of training data with weak priors in the street environment, followed by a filtering scheme to remove mislabeled training samples. We formulate the filtering as a binary labeling optimization problem over a conditional random filed that we call object graph, simultaneously integrating spatial smoothness preference and label consistency between 2D and 3D. Toward the final parsing, with the automatically generated training data, a CRF-based parsing method that integrates the coordination of image appearance and 3D geometry is adopted to perform the parsing of large-scale street scenes. The proposed approach is evaluated on city-scale Google Street View data, with encouraging parsing performance demonstrated.**

***Index Terms*—Segmentation, Street Scene, Image, Point Cloud**

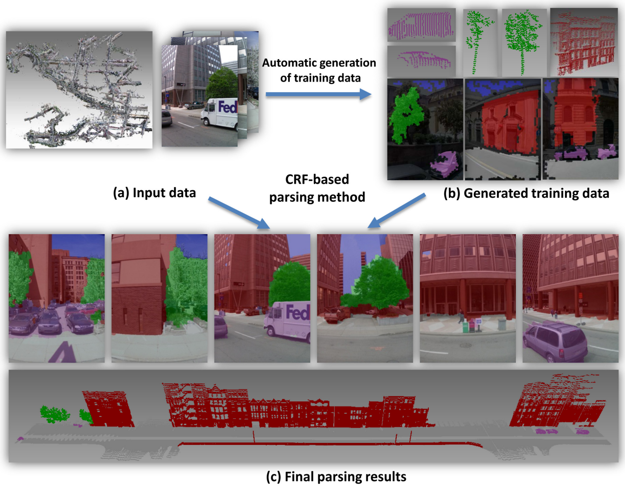

Automatic generation of training data
(a) Input data
CRF-based parsing method
(b) Generated training data
(c) Final parsing results

Fig. 1. Overview of the proposed method

I. Introduction

The parsing of images and scanned point cloud in street scenes has received significant attention recently because of its fundamental impact on scene understanding, content-based retrieval, and 3D reconstruction in the street environment. Moreover, with the dramatically boosting volumes of street view data on the Internet and the urgent requirement for virtual urban applications, parsing methods applicable to large-scale street scenes is in great demand. The parsing of street view images has been extensively studied in previous works [1], [2], [3], [4], [5], with impressive results demonstrated. In parallel, great effort has been devoted to the segmentation of 3D point clouds acquired by 3D laser range sensors in many previous works [6], [7], [8], [9], [10].

Besides images or scanned point cloud individually, modern street view data [11] includes both color images and 3D scanned points captured simultaneously with the calibrated cameras and laser scanners are widely applicable, as shown in Figure 2. It has been demonstrated in the previous works [1], [2], [3], [9] that the fusion of 2D appearance information and 3D geometry information can significantly improve the accuracy of the parsing of street scenes. However, to apply the traditional methods [1], [2], [3], [6], [7], [8], [9], [10] to the large-scale parsing of street scenes, a large amount of training data that can account for the vast visual and structural variance of street environment is necessary. Unfortunately, such training data is mostly obtained by tedious and time-consuming manual labeling in the previous approaches, which inevitably becomes an obstacle to applying these traditional parsing methods to the large-scale parsing of street scene. Though there exist some databases with annotations for the street scene, like the CBCL Street Scenes database [12], they are still limited in scale and variance of data sources.

To reduce the cost of manually annotating training data for the parsing of large scale street scenes, we propose a large scale parsing system that can automatically generate training data from input data. Given the coordination of the input images and point cloud in street scenes, the automatic proposal of training data is achieved by fully utilizing the prevailing knowledge of street environment. Intuitively, some simple priors can be easily used to distinguish instances of different categories in the street environment. For example, the average height of a building above the ground should be greater than a certain value, and the shape of the ground is usually a planar surface. These priors may not be valid for every instance of the categories, but valid for most of them, and thus we call them *weak priors*. The weak priors are treated as “weak classifiers” and are combined to recognize instances

All authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, e-mail: (zhhsmail@gmail.com, {jwangae, tianft, quan}@cse.ust.hk).

of different categories from the input data. The recognized instances are likely to be misclassified since the weak priors are solely from simple observations. We regard the recognized result by the weak priors as initial training data with a certain degree of noise. In the next step, a filtering scheme to remove the mislabeled training samples in the initial training data is introduced by formulating it as a binary labeling problem over a CRF. The unary confidence for the initial labeling is estimated by a cross-validation inspired algorithm. The interaction term imposes the geometric spatial smoothness and label consistency which characterizes the correspondences between images and scanned point cloud and is encoded in a carefully designed joint 2D-3D object graph.

Finally, with the automatically generated training data, we use a CRF-based joint 2D-3D method to simultaneously segment the scanned point clouds and street view images into five most common categories in the street environment: building, car, tree, ground and sky as the previous works [13], [4] did. An overview of the proposed method is given in Figure 1.

In summary, the contributions of our approach are threefold. First, the utilization of weak priors in both street view images and scanned point cloud automates the generation of training data, significantly reducing the intensive manual labeling in previous works. To our best knowledge, this is the very first exploration of this idea for scene parsing. Second, the novel joint 2D-3D object graph significantly purifies the automatically generated training samples. Last but not least, integrated with the state-of-the-art CRF-based parsing techniques, we demonstrate the potential of fully automatic large-scale parsing of street scene with comparative performance to that achieved by using manually labeled training data. The rest of this paper is organized as follows: In Section II, some related works are reviewed. Then, we introduce the automatic generation of training data In Section III and the CRF-based parsing module In Section IV; Last, we present the experiment evaluation in Section V, and conclude in Section VI.

II. Related work

For the parsing of street view images, different methods have been proposed [2], [3], [13], [4], [14], which usually formulate the parsing problem with graphical models, such as the CRF. In [2], a multi-view parsing method for image sequences captured by a camera mounted on a car driving along streets is proposed, with SfM(Structure from Motion algorithm) [15] used to reconstruct the scene geometry. Similar works were introduced in [3], using dense depth maps recovered via multi-view stereo matching techniques as cues to achieve accurate scene paring. In [5], the temporal consistency between consecutive frames and 3D scene geometry recovered by stereo techniques are explored to improve the parsing accuracy of street view images. In [16], the authors jointly address the semantic segmentation and dense 3D scene reconstruction by learning appearance-based cues and 3D surface orientations and performing class-specific regularization. For the parsing of still images, a hierarchical two-stage CRF model is proposed in [4] to segment images of street scenes.

The semantic segmentation of scanned point cloud in street environment is closely related to the parsing of street view images, and well studied in the previous works [6], [8], [9]. In [6], [7], learning-based methods for segmenting 3D scan data into objects or object classes are proposed, with only 3D information used. In [9], the authors introduced a probabilistic, two-stage classification framework for the semantic segmentation of urban maps as provided by a mobile robot, using both appearance information from color images and geometric information from scan data. A similar work was introduced in [8], incorporating visual appearance by projecting the laser returns into images collected by a calibrated camera mounted on the vehicle. Without exception, the training data in all these methods [2], [3], [4], [6], [8], [9] is obtained by manually labeling.

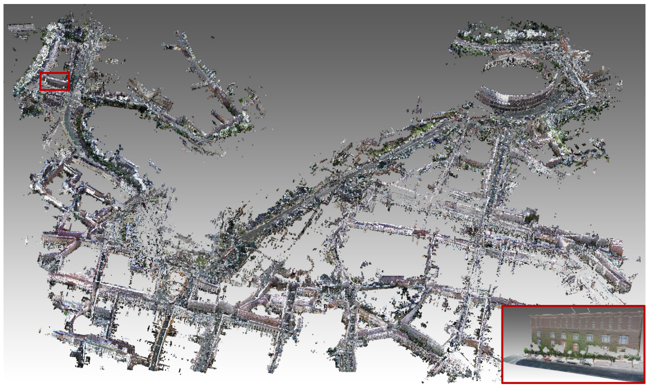


Fig. 2. A 3D view of the street view data. The scene is rendered by fusing the point cloud captured by laser scanners and the images captured by color cameras registered to the laser scanners.

III. Automatic Generation of Training Data

The automatic generation of training data includes two successive steps: 1) labeling initialization in the input data which includes both scanned point cloud and images; 2) filtering of mislabeled training samples. For the labeling initialization, objects of different categories are first segmented in the scanned point cloud, and recognized with weak priors about these categories in 3D space. Then, we transfer the labeling of the recognized objects from 3D space to image space to initialize the training data for the street view image parsing. For the category *sky* that does not exist in the scanned point cloud, we directly initialize the training data for it in images with proper weak priors in image space. As the weak priors used to initialize the training data are just based on some simple observations, there are inevitably some mislabeled training samples, which could affect the performance of the street view image parsing significantly. To remove those mislabeled samples in the initialized training data, we propose a filtering algorithm in Section III-C. First, we introduce the labeling initialization in the scanned point cloud and images.

Data preprocessing Using the mobile platform (usually a car) which the data capturing equipments mount on as a reference, the height and depth information of each point in the scanned point cloud are estimated. The normal directions of points in the scanned point cloud are estimated by the tensor voting algorithm [17], with isolated points removed.

A. Labeling Initialization in Scanned Point Cloud

The labeling initialization starts with extracting objects of different categories from the scanned point cloud with the proposed object-model-based extraction algorithm, as described in Algorithm 1. Objects of different categories are extracted and recognized sequentially, in the order: *ground*, *building*, *car* and *tree*. As laser rays cannot reach the sky, the category *sky* is excluded here.

Algorithm 1 Object-Model-Based Extraction

```
Input : point cloud $\mathcal{P}$
for all $C \in \{ground, building, car, tree\}$ do
    Construct a KNN(K-nearest-neighbor) graph for points, $\mathcal{G}_C = \langle \mathcal{P}, \mathcal{E}_C \rangle$, with the length of each edge smaller than $\kappa$;
    # Recognize components of different categories
    Extract connected components $\{S_i\}$ from $\mathcal{G}_C$ and set $T_C = \emptyset$
    for all $S \in \{S_i\}$ do
        if $S$ fits the object model for $C$ then
            Insert $S$ into $T_C$ and remove $S$ from $\{S_i\}$
        end if
    end for
    # Merge components of different categories
    for all $T \in T_C$ do
        for all $S \in \{S_i\}$ do
            if The shortest distance between $T$ and $S$, $D < \epsilon$ & $T \cup S$ fits the object model for $C$ then
                $T = T \cup S$, $\{S_i\} = \{S_i\} - S$
            end if
        end for
        Label $T$ as category $C$, and remove it from $\mathcal{P}$
    end for
end for
```

Based on some weak priors about each category, the object model for the category specifies several discriminative properties for recognizing objects of the category, which includes properties of the following several aspects:

(1) *Width* f_w, length of an object along the scan direction.

(2) *Average height above the ground*f_h, the average height of all points in an object.

(3) *Shape*, the saliency feature [18] can distinguish between three basic shapes of objects: line, surface and scatter cloud. Suppose the saliency feature for an object is $(\lambda_1, \lambda_2, \lambda_3)$, sorted in descending order, which are the eigenvalues of the covariance matrix for all points in an object. If $\lambda_1 \gg \lambda_2 \approx \lambda_3$, the shape of the object is a line; If $\lambda_1 \approx \lambda_2 \approx \lambda_3$, the shape of the object is a scatter cloud; If $\lambda_1 \approx \lambda_2 \gg \lambda_3$, the shape of the object is a surface.

(4) *Ratio of points whose dominant normal direction are vertical and horizontal*R_v, the ratio of points in an object with $N_z \geq N_x, N_z \geq N_y$; R_h, the ratio of points in an object with $N_z \leq N_x, N_z \leq N_y$. (N_x, N_y, N_z) is the normal vector for a point, and z is the vertical direction.

The object models for different categories are presented in Table I. A typical labeling initialization result obtained by Algorithm 1 is shown in Figure 3. The model parameters for these categories are estimated with very few instances (¡ 10), since the the raw initialization will be purified in the next stage.

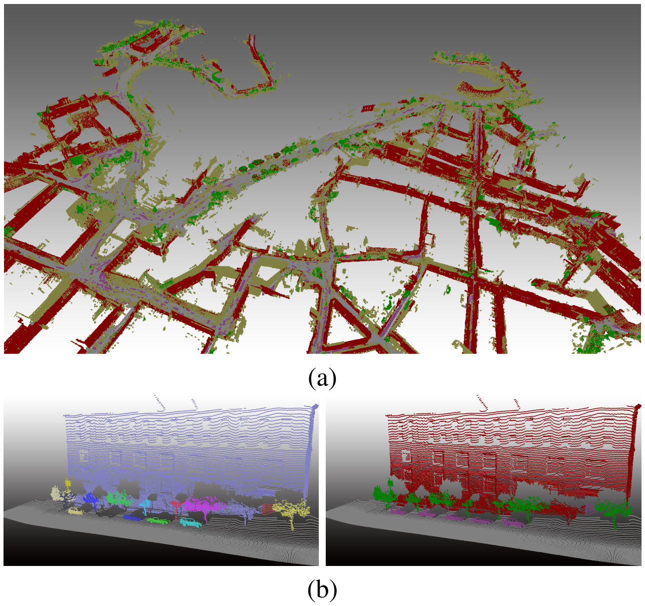


Fig. 3. Labeling initialization in the scanned point cloud with Algorithm 1. (a) the initial labeling of city-scale scanned point cloud. Different categories are denoted with different colors: red for *building*, green for *tree*, gray for *ground*, purple for *car* and yellow for unlabeled points. (b) a local initial labeling, with the extracted objects shown in different colors in the left image and the initial labeling shown in the right image.

Object Model	Properties			
	f_w	Shape	R_v & R_h	f_h
ground	> 12m	surface	$R_v > 0.5$	< 0.5m
building	> 12m	not a line	$R_h > 0.5$	> 5m
car	$1 \sim 10$m	scatter cloud	-	$0.5 \sim 2.5$m
tree	$1 \sim 10$m	scatter cloud	-	$1 \sim 10$m

TABLE I
Object models for different categories

B. Labeling Initialization in Images

With the initialized labeling of the categories: *ground*, *building*, *car*, *tree* in the scanned point cloud and the projections of 3D points to images, the labeling initialization for these categories in image space is carried out by transferring the initialized labels of 3D points to image space. For the category *sky* not initialized in the scan data, initial guess is carried out in the image space with the following priors:

(1) *Existence of 3D projections*: As the scan laser ray cannot reach the sky, the existence of 3D projections in images is a strong indicator of non-sky region.

(2) *Region Color Variance*: The color variance of sky regions are usually small.

(3) *Position in images*: As the images are taken on ground level, the *sky* regions in images mostly appear in the upper part.

Before applying these weak priors, we first create super pixel partition of images with the method [19]. Then, candidate superpixels that satisfy the following requirements are selected: 1) no 3D projections in the superpixel; 2) color variance $s < s_t$ ($s_t = 25$), which is measured by the max color (RGB space) difference of any two pixels within the superpixel; 3) average height $h < h_t$ (the height is scaled to $[0, 1]$ against the image height, $h_t = 0.25$). Last, some pixels are randomly sampled from these candidate superpixels and assigned the label *sky*.

C. Filtering of Mislabeled Training Samples

As the automatically initialized training samples are generated with only weak priors, some of them are probably mislabeled. Meanwhile, the registration error between images and scanned point could also cause the mislabeling. As well known, noise in training data could severely degrade the performance of the trained classification models. To remove the mislabeled training samples in the initialized training data, we propose a mislabeling filtering scheme based on the flexible CRF formulation [20], [21]. The confidence for the initial labeling served as the unary potential in CRF-based formulation is estimated jointly with the appearance information from 2D images and geometric information from 3D scanned point cloud. However, more than the estimated confidence in the initial labeling that was used to identifies the mislabeled training samples in the previous work [22], we integrate the spatial smoothness and label consistency between images and scanned point cloud as well. All these cues are integrated into a CRF model that we call object graph to robustly identify and remove the mislabeled training samples.

1) Estimation of confidence in the initial labeling: Our confidence estimation for the initial labeling, as described in Algorithm 2, is inspired by the previous work [22]. Please note that in Algorithm 2, the confidence estimation for the initial labeling of different categories is performed independently, in both images and scanned point cloud. The estimation process for each category follows the standard Leave-one-out cross-validation of multiple rounds with random data partitions. In each round of the cross-validation, the initial training data is randomly partitioned into two sets, training set and testing set. A binary classifier is trained by the data from training set. Suppose the binary classifier performs better than random guess, if the initial label of a sample from testing set agrees with that predicted by the trained classifier, then the probability that the initial label is correct is large than 1/2. As the testing in each round of the cross-validation is based on random data partition and thus can be treated as independent testing, the more times a sample's initial label agrees with the predicted label, the more likely its initial label is correct.

In the following, we use $\mathbf{P}(y, k)$ to denote the probability that a sample is classified as a positive sample k times during the N iterations in the algorithm 2, where $y \in \{-1, +1\}$ denotes the true label of the sample. Suppose the classification accuracy of the trained classifiers during the N iterations in the algorithm 2 is q, then we have:

$$\mathbf{P}(k|y = -1) = C_N^k (1 - q)^k q^{N-k} \tag{1}$$

$$\mathbf{P}(k|y = +1) = C_N^k q^k (1 - q)^{N-k} \tag{2}$$

Algorithm 2 Confidence estimation for the initial labeling

```
1:  Input: the initial training samples $\mathrm{S} = \{s_i\}$ for the target category $C$, and initial training samples $\{Q_i\}$ for other categories
2:  Initialize: $k_s = 0$ for all $s \in \mathrm{S}$
3:  for $i = 1, 2, ..., N$ do
4:      Randomly split $\mathrm{S} = \mathrm{S}_{train} \cup \mathrm{S}_{test}$ with $50\%/50\%$
5:      Train a binary random forest classifier $\mathcal{R}$ with positive samples from $\mathrm{S}_{train}$ and negative samples of the same number that are randomly sampled from $\{Q_i\}$
6:      for $s \in \mathrm{S}_{test}$ do
7:          if $s$ is classified as a positive sample by $\mathcal{R}$ then
8:              $k_s = k_s + 1$
9:          end if
10:     end for
11:     Exchange $\mathrm{S}_{train}$ and $\mathrm{S}_{test}$, repeat step 5 -10
12: end for
13: For all $s \in \mathrm{S}$, set $k = k_s$ and compute the confidence in its initial label with equation (4)
```

For a sample classified as a positive sample k times during the N iterations, the probability that its initial label is correct is:

$$\begin{aligned} \mathbf{P}(y = +1|k) &= \frac{\mathbf{P}(y = +1, k)}{\sum_{y \in \{-1,+1\}} \mathbf{P}(y, k)} \\ &= \frac{q^k (1 - q)^{N-k}}{q^k (1 - q)^{N-k} + \frac{\mathbf{P}(y=-1)}{\mathbf{P}(y=+1)} (1 - q)^k q^{N-k}} \end{aligned} \tag{3}$$

As $\mathbf{P}(y)$is unknown, we make an assumption that the ratio of mislabeled training samples is under 50%, so that we can properly approximate (3). This assumption gives $\mathbf{P}(y = -1) \leq \mathbf{P}(y = +1)$, so we have $\mathbf{P}(y = +1|k) \geq f(q, k, N)$, where

$$f(q, k, N) = \frac{q^k (1 - q)^{N-k}}{q^k (1 - q)^{N-k} + (1 - q)^k q^{N-k}} \tag{4}$$

With this assumption which is verified in the following experiment, we can approximate $\mathbf{P}(y = +1|k)$with (4) in the algorithm 2 ($N = 10$, $q = 0.6$ in our implementation), and safely assume that the trained classifier in each iteration of the algorithm 2 is better than random guess. For the training/testing procedure (step 5 and 7) in the algorithm 2, the following features are used:

a) Features for confidence estimation in the scanned point cloud: For each initialized object, we extract three bag-of-word features built with the normal, height, and depth of points in the object respectively, with the same way that the bag-of-word features were built in [23].

b) Features for confidence estimation in images: For each initialized pixel with projection of 3D points in images, we extract patch level appearance features: Texton and SIFT, and combine them with the normal, height, and depth of the corresponding 3D point that projects to the position. For the samples of the category *sky* without projection of 3D points, we add default normal, height, and depth.

2) Object graph for the filtering of mislabeled training samples: In this section, we introduce the object graph which integrates multiple cues for the filtering of mislabeled training samples. For each initialized training sample in the point cloud, a recognized 3D object O obtained by Algorithm 1, we define a graph $\mathcal{G} = \langle \mathcal{V} + \mathcal{T}, \mathcal{E}_\mathcal{V} + \mathcal{E}_\mathcal{T} \rangle$ that we call object graph, as shown in Figure 4. The global node T denotes the object O. The nodes in $\mathcal{V}$ denote the initialized training samples in the images, pixels with projections of 3D points in O. As the 3D objects extracted by Algorithm 1 are connected components, the links between the points in O are transfered to the graph $\mathcal{G}$. We denote these links with $\mathcal{E}_\mathcal{V}$, and the links between nodes in $\mathcal{V}$ and nodes in $\mathcal{T}$ with $\mathcal{E}_\mathcal{T}$. Then, the energy function associated with $\mathcal{G}$ is defined as:

$$\begin{aligned} E(\mathbf{x}, y) &= \phi_\mathcal{T}(y) + \sum_{(i,j)\in\mathcal{E}_\mathcal{T}} \phi_{ij}(x_i, y) + \sum_{i\in\mathcal{V}} \varphi_i(x_i) \\ &+ \sum_{(i,j)\in\mathcal{E}_\mathcal{V}} \varphi_{ij}(x_i, x_j) \end{aligned} \quad (5)$$

The random variable $x_i \in \mathbf{x}$ and y associated with each node take values from the label set $L = \{l_{good}, l_{bad}\}$ that distinguish the correctly labeled training samples and the mislabeled training samples. The data terms $\varphi_i(x_i)$ and $\phi_\mathcal{T}(y)$ that encode the estimated confidence in the initial labeling are defined as:

$$\phi_\mathcal{T}(y) = \begin{cases} |\mathcal{V}|(1 - P_\mathcal{T}) & y = l_{good}; \\ |\mathcal{V}|P_\mathcal{T} & y = l_{bad}. \end{cases} \quad (6)$$

$$\varphi_i(x_i) = \begin{cases} 1 - P_i & x_i = l_{good}; \\ P_i & x_i = l_{bad}. \end{cases} \quad (7)$$

$P_\mathcal{T}$ and P_i are the estimated confidence in the initial labeling for different nodes. $|\mathcal{V}|$ are the number of points in the 3D object. To encourage spatial smoothness, the smooth term $\varphi_{ij}(x_i, x_j)$ takes the following form:

$$\varphi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j; \\ \lambda & \text{otherwise.} \end{cases} \quad (8)$$

To encode label consistency between the images and scanned point cloud, the term $\phi_{ij}(x_i, y)$ takes the following form:

$$\phi_{ij}(x_i, y) = \begin{cases} \infty & \text{if } y = l_{bad}, x_i \neq y; \\ \frac{1-2P_\mathcal{T}}{0.1|\mathcal{V}|} & \text{if } y = l_{good}, x_i \neq y; \\ 0 & \text{if } y = l_{good}, x_i = y \end{cases} \quad (9)$$

This term (9) forces that each node in the images can be considered as correctly labeled, only when the global node in the point cloud is correctly labeled. It also allows no more than 10% of the nodes in images can take the label l_{bad} when of the 3D object takes the label l_{good}, in order to tolerate certain degree of registration error between the scanned point cloud and images.

For the initialized training samples of the category *sky* in each image, we define a similar graph as the object graph, without the global node. For each node that denotes one pixel in an image, it is linked to its $K(K = 5)$ nearest neighbors. The corresponding energy function takes the form of (5), excluding terms involving the global node. Some results of the filtering of mislabeled training samples are shown in Figure 5. To the end, we solve the optimization problem (5) by the α-expansion algorithm [24].

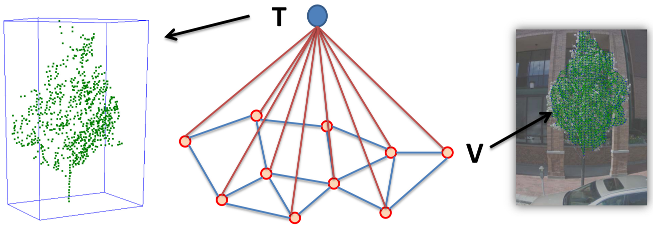


Fig. 4. Object graph: the global node $\mathcal{T}$ denotes the 3D object, and nodes in $\mathcal{V}$ denote pixels in the images with projections of 3D points in the 3D object. Links between the nodes are denoted by lines of different colors.

IV. Joint Segmentation of Images and Scanned Point Cloud

The final segmentation of scan data and images will be obtained by the CRF-based joint segmentation module. Details are introduced in the following.

A. Object extraction in scanned point cloud

As the first step of the joint segmentation, objects of different categories are extracted by Algorithm 1, using new object models for different categories. The original object models in Algorithm 1 are replaced by the new object models, multiple binary random forest classifiers. These classifiers determines whether a 3D object belongs to a specific category, and are trained with the automatically obtained training data. Specifically, we replace the object model fitting test (line 7 and 15) in Algorithm 1 with the following test: if the estimated probability of a 3D object belonging to the category is large than 0.5, we consider it passing the test.

B. Associative Hierarchical CRF for joint optimization

With the extracted objects in the scanned point cloud, we use the associative Hierarchical CRF [25] to formulate the joint segmentation problem of images and scanned point cloud. We define a hierarchical graph $\mathcal{G}' = \langle \mathcal{V}' + \mathcal{T}', \mathcal{E}_{\mathcal{V}'} + \mathcal{E}_{\mathcal{T}'} + \mathcal{E}_{\mathcal{N}'} \rangle$. The global node set $\mathcal{T}'$ denotes the extracted objects in the point cloud, and the nodes in $\mathcal{V}'$ denote the pixels in images. For each extracted object, we build an object graph defined in section III-C2, as a part of $\mathcal{G}'$. For each pixel in images, we add the four neighborhood links, denoted by $\mathcal{E}_{\mathcal{N}'}$. $\mathcal{E}_{\mathcal{V}'}$ denotes the links between nodes in $\mathcal{V}'$ associated with the object graphs, and $\mathcal{E}_{\mathcal{T}'}$ denotes the links between nodes in $\mathcal{V}'$ and $\mathcal{T}'$. The energy function associated with $\mathcal{G}'$ is defined as:

$$\begin{aligned} E'(\mathbf{X}, \mathbf{Y}) = \sum_{i\in\mathcal{T}'} \phi'_i(y_i) + \alpha \sum_{(i,j)\in\mathcal{E}_{\mathcal{T}'}} \phi'_{ij}(y_i, x_j) + \\ \beta \sum_{i\in\mathcal{V}'} \varphi'_i(x_i) + \gamma \sum_{(i,j)\in\mathcal{E}_{\mathcal{V}'}+\mathcal{E}_{\mathcal{N}'}} \varphi'_{ij}(x_i, x_j) \end{aligned} \quad (10)$$

This energy function integrates the image appearance information, geometry information and object level information obtained from the scanned point cloud together. Different cost terms are explained in the following:

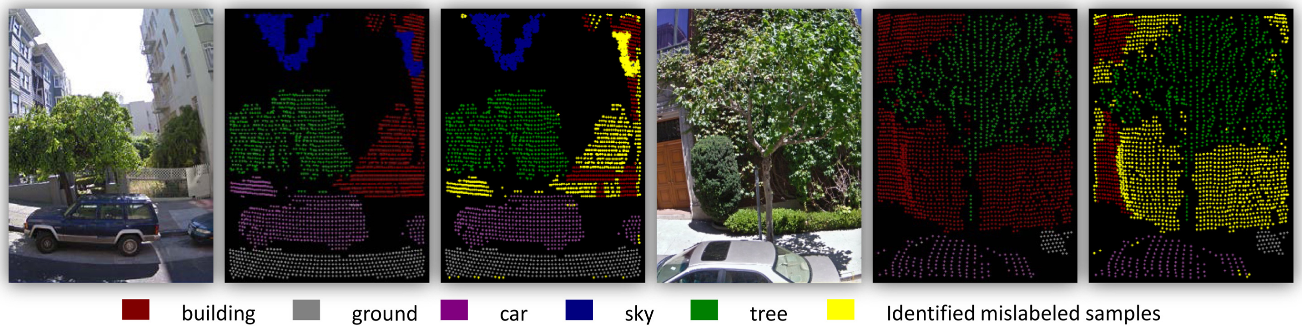


Fig. 5. Examples of the initialized labeling and cleaned labeling obtained by the proposed filtering of mislabeled training samples. column 1 and 4: the original images; column 2 and 5: the corresponding initialized labeling; column 3 and 6: the cleaned labeling.

1) Energy cost for 2D images: The random variable $x_i \in \mathbf{x}$ associated with one node in $\mathcal{V}'$ takes a value from the label set $L = \{$*building*, *car*, *tree*, *ground*, *sky*$\}$. The data term $\varphi'_i(x_i)$ is a unary potential term to encode the probability of a node taking a certain label estimated by the two random forest classifiers R$_{2D}$ and R$_{2D3D}$, similar to the TextonBoost framework [20]. R$_{2D}$ is trained with patch level appearance features: Texton and SIFT, applied to the pixels without projections of 3D points. R$_{2D3D}$ is trained with the patch level appearance features for R$_{2D}$ combined with the normal, height, and depth of the 3D points that project to the corresponding pixels, applied to the pixels with projections of 3D points. As the projections of 3D points are strong indicators of non-sky region, the nodes in $\mathcal{V}'$ with projections of 3D points will never take the label *sky*. The smooth term $\varphi'_{ij}(x_i, x_j)$ is similar in form to the smooth term (8).

2) Energy cost for 3D objects: Random variables $y_i \in Y$ associated with nodes in $\mathcal{T}'$ take a value from the label set $L = \{l_{good}, l_{bad}\}$ that denotes whether the initial label of the extracted object is correct or wrong. For an object with initial label L_i, the data term $\phi'_i(y_i)$ takes the following form:

$$\phi'_i(y_i) = \left\{ \begin{array}{ll} |c|(1 - P_{y_i}) & y_i = l_{good} \\ |c|P_{y_i} & y_i = l_{bad} \end{array} \right. \qquad (11)$$

Here, the estimated confidence P_{y_i} is obtained by testing with the trained binary Random Forest classifiers for the category L_i, the updated object model for the category L_i. $|c|$ is the number of points contained in the object. The smooth term $\phi'_{ij}(y_i, x_j)$ are used to encode label consistency between the 3D point cloud and 2D images, which are defined as:

$$\phi'_{ij}(y_i, x_j) = \left\{ \begin{array}{ll} (1 - 2P_{y_i})/0.1|c| & y_i = l_{good}, x_i \neq L_i \\ 0 & y_i = l_{bad} \text{ or } x_i = L_i \end{array} \right. \qquad (12)$$

The weights α, β, γ in (10) can be estimated by the cross validation on a hold-out set, and we solve the optimization problem (10) with the α-expansion algorithm [24].

V. Experiments

We evaluate the proposed method on the Google Street View data of two cites: San Francisco (SF) and Rome, captured by the R5 system of Google [11]. The images are captured by a ring of interline-transfer, CCD sensors with wide-angle lenses. Three laser scanners is also included in the R5 system, thereby enabling the capture of sparse 3D data alongside the imagery. As a part of the street view processing pipeline, image pixels have been corresponded to 3D rays in the presence of a rolling shutter. More details about the data used in our experiment are described in Table II. To handle the huge data of such a large scale, we split the entire dataset into about fifty segments, and each segment is processed independently. To build the object models described in Section III-A, we roughly estimate them with five manually labeled objects of each category, which is the only annotation needed in our method. In terms of the ground truth data for quantitative evaluation, we uniformly sampled one thousand images from the input dataset and labeled them into five categories: *ground*, *building*, *car*, *tree*, *sky*, as well as points in the scanned point cloud associated with their projections in these selected images.

As the focus of this paper is the parsing of street scenes with automatically generated training data, our evaluation consists of two parts: 1) evaluation of the automatically generated training data; 2) evaluation of the parsing performance achieved with the automatically generated training data. For the training data generated by the proposed algorithm in Section III, we refer to the training data generated by the labeling initialization as the initialized training data, and the training data obtained by performing the filtering of mislabeled training samples on the initialized training data as the cleaned training data in the following description. The processing time spent on different components of our method is given in Table II, based an unoptimized C++ implementation running on a pc (i7 2.8G, 16G RAM).

A. Evaluation of the automatically generated training data

Similar to the evaluation on the ground truth annotation obtained by manual labeling in [26], we measure the quality of the automatically generated training data in two aspects: the amount of the training data and the degree of noise in the training data which reflects the accuracy of the automatic generation of training data.

For the first aspect, the proportion of the generated training data (the proportion of pixels with labels among all pixels in all images and points with labels in the entire scanned point cloud) in the input dataset is shown in Figure 6, with the

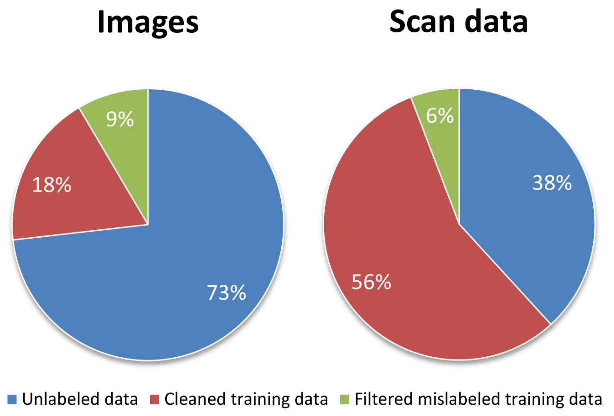


Fig. 6. The proportion of the generated training data in the input dataset. The initialized training data consists of two parts: the cleaned training data and the removed training data by the filtering of mislabeled training samples.

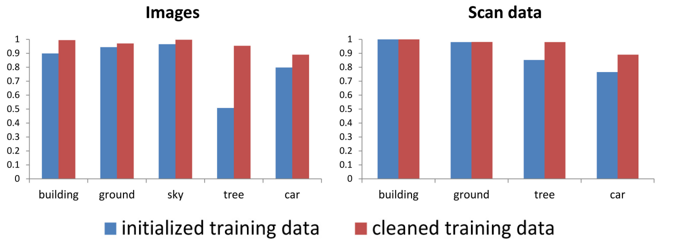


Fig. 7. The purity of the initialized training data and the cleaned training data in images and scanned point clouds.

Dataset	#Points	#Images	LI	FT	TR	TE
Rome	42,111 k	192,861	5h	63h	38h	42h
SF	19,423 k	19,107	1h	7h	4h	4h

TABLE II

The description of the dataset. #Points: the number of 3D points; #Images: the number of images; LI: total time of labeling initialization; FT: total time of filtering the mislabeled training samples; TR: total time of training the CRF model with the automatically generated training data; TE: total time of testing all the images in the dataset. The size of all the images in the dataset is 387×518.

label associated with each pixel diffused within superpixels as described in Section IV. From Figure 6 which illustrates the composition of the input dataset, we note that about 18% of pixels in all images are annotated, with 9% of pixels removed in the filtering of mislabeled training samples. By contrast, about 56% of points in the scanned point cloud are annotated, with 6% of points removed in the filtering of mislabeled training samples.

For the second aspect, the *purity* of the initialized training data and the cleaned training data evaluated on the ground truth dataset is shown in Figure 7, which counts the ratio of the correctly labeled training samples in the training data. From the comparison in Figure 7, we find that the purities of all categories in the initialized training data are larger than $1/2$, which verifies the assumption that the ratio of mislabeled training samples is under $1/2$ made in Section III-C1. We also find that the purities of most categories in the cleaned training data are improved by the proposed filtering of mislabeled training samples, especially the categories *tree* and *car*. Several images with the automatically generated annotation for them are shown in Figure 8.

B. Evaluation of the parsing performance

In this section, we evaluate the parsing performance achieved by our method. For the evaluation of segmentation performance, we use the CAA (category average accuracy, the average proportion of pixels/points correctly labeled in each category) and the GA (global accuracy, total proportion of pixels/points correctly labeled), as previous work [27], [13].

The segmentation performance achieved by our method on the images and scanned point cloud is shown in table III, using the cleaned training data for the training process. To evaluate the influence of the joint segmentation of images and scanned point cloud together, we first test both the joint segmentation and non-joint segmentation, under the formulation (10). Here, the non-joint segmentation refers to treating the segmentation of images and scanned point cloud separately. For the non-joint segmentation of images, we remove all cost terms involving scanned point cloud, treating it as a still image segmentation like [4], [28]. For the non-joint segmentation of scanned point cloud, we remove all cost terms involving images, treating it as a pure scanned point cloud segmentation problem like [6]. The global accuracy, category average accuracy and segmentation accuracy of different categories achieved by the joint segmentation and non-joint segmentation are compared in Table III. As expected, we can find the joint segmentation achieves better global accuracy and category average accuracy. However, we also note that not all categories benefit from the joint 2D-3D framework. On one hand, for man-made objects (all classes except trees), they bear strong regularity in the scanned point cloud, therefore, they can well segmented with 3D geometry information merely. The joint segmentation of point-clouds and images did not provide much improvement on these man-made categories. On the other hand, the registration errors between the 2D images and 3D point cloud also lead to the degradation of the segmentation performance on some categories. Some segmentation results obtained by the joint segmentation are shown in Figure 9. More results are provided in the supplementary video associated with the manuscript.

To further evaluate the quality of the automatically generated training data, we randomly split the manually labeled ground truth dataset in our experiments into two parts with ratio 30%/70%, one part used as training data, and another part for performance evaluation. We repeated the random split five times, and averaged the achieved global accuracy and category average accuracy for each random split. the achieved segmentation accuracy with the automatically generated training data and manually labeled training data is compared in Table IV. The average global accuracy and category average accuracy achieved on the images are 84.3% and 82.2%, slightly better than the accuracy achieved with the automatically generated training data. The average global accuracy and category average accuracy achieved on the scanned point cloud are 89.1% and 80.5%, slightly worse than the accuracy achieved with

Data set	Image segmentation		Point clouds segmentation	
	Joint	Non-joint	Joint	Non-joint
Building	0.879	0.887	0.843	0.845
ground	0.843	0.861	0.991	0.986
sky	0.887	0.912	-	-
tree	0.819	0.754	0.654	0.540
car	0.670	0.421	0.843	0.837

TABLE III
The segmentation accuracy of different categories, achieved by the joint segmentation and non-joint segmentation respectively. With the cleaned training data, the global accuracy and category average accuracy achieved on the images are 83.9%/82% (joint segmentation) and 81.8%/76.7% (non-joint segmentation). The global accuracy and category average accuracy achieved on the scanned point cloud are 91.1%/83.3% (joint segmentation) and 90.4%/80.2% (non-joint segmentation).

Training data	Image segmentation		Point clouds segmentation	
	GA	CAA	GA	CAA
automatically generated	0.839	0.82	0.911	0.832
manually labeled	0.843	0.822	0.891	0.805

TABLE IV
The segmentation accuracy achieved with the automatically generated training data and manually labeled training data.

the automatically generated training data. In general, in this comparison, the segmentation performance achieved by using the automatically generated training data is comparable to that achieved by using the manually labeled training data.

Last, we compare our method with previous works [25], [13], [6], [29]. For the competitors, we use the same training/testing setting as the comparison given in Table IV. [25], [13] are trained with annotated images and applied to image segmentation, with the same parameters as that in [25], [13]. [6], [29] are trained with annotated point cloud and applied to point cloud segmentation), with the parameters of [6] well tuned on the dataset used in our experiment. The comparison result is given in Table V. From the comparison, we can find our method outperforms the competitors to varying degrees, which shows the advantage of the joint segmentation.

VI. Conclusion and Discussion

In this paper, we propose a novel method for the parsing of images and scanned point cloud captured in large-scale street scenes, which can automatically generate training data from the given input data with weak priors. First, utilizing the weak priors about the most five common categories in the street environment, the initialized training data is generated. Then, a filtering algorithm is proposed to remove those mislabeled training samples in the initialized training data. Finally, with the generated training data, a CRF-based parsing module is proposed for the parsing of large scale street scenes, which uses both the image appearance information and geometry information.

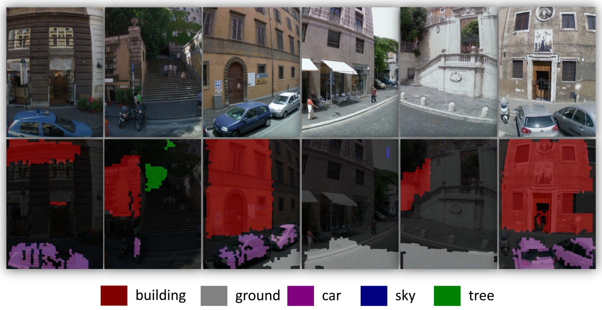


Fig. 8. Examples of the parsing results obtained by our method. The first row shows the input images, and the second row shows the parsing results. The third row shows the automatically generated annotation for these images by our method.

Data set	Image segmentation		Point clouds segmentation	
	GA	CAA	GA	CAA
our method	0.839	0.82	0.911	0.832
[25]	0.821	0.809	-	-
[13]	0.8	0.792	-	-
[6]	-	-	0.89	0.75
[29]	-	-	0.91	0.787

TABLE V
The segmentation accuracy achieved by our method and previous methods [25], [13], [6], [29].

With scanned point cloud of low resolution, like the Google Street View data we used in our experiment, it is difficult to extract small objects as the points on them are quite few, which hinders the automatic generation of training data for these categories. Therefore, currently, the targeted categories in the proposed method are constrained to the major categories in the street view. Given scanned point cloud captured by laser sensors with higher resolution, it is reasonable to expect that the proposed method can be generalized to more categories.

Acknowledgment

This work is supported by RGC-GRF 618711, RGC/NSFC N_HKUST607/11, ITC-PSKL12EG02, National Basic Research Program of China (2012CB316300) and Google University Award GOOG13EG03.

References

[1] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, “Segmentation and recognition using structure from motion point clouds,” *European Conference on Computer Vision*, 2008.

[2] J. Xiao and L. Quan, “Multiple view semantic segmentation for street view images,” *International Conference on Computer Vision*, 2009.

[3] C. Zhang, L. Wang, and R. Yang, “Semantic segmentation of urban scenes using dense depth maps,” *European Conference on Computer Vision*, 2010.

[4] Q. Huang, M. Han, B. Wu, and S. Ioffe, “A hierarchical conditional random field model for labeling and segmenting images of street scenes,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[5] G. Floros and B. Leibe, “Joint 2d-3d temporally consistent segmentation of street scenes,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[6] D. Anguelov, B. Taskarf, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, “Discriminative learning of markov random fields for segmentation of 3d scan data,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[7] A. Golovinskiy, V. G. Kim, and T. Funkhouser, “Shape-based recognition of 3d point clouds in urban environments,” *International Conference on Computer Vision*, 2009.

[8] B. Douillard, D. Fox, and F. Ramos, “Laser and vision based outdoor object mapping,” *Proceedings of the Robotics Science and Systems*, 2008.

[9] I. Posner, M. Cummins, and P. Newman, “Fast probabilistic labeling of city maps,” *Proceedings of Robotics Science and Systems*, 2008.

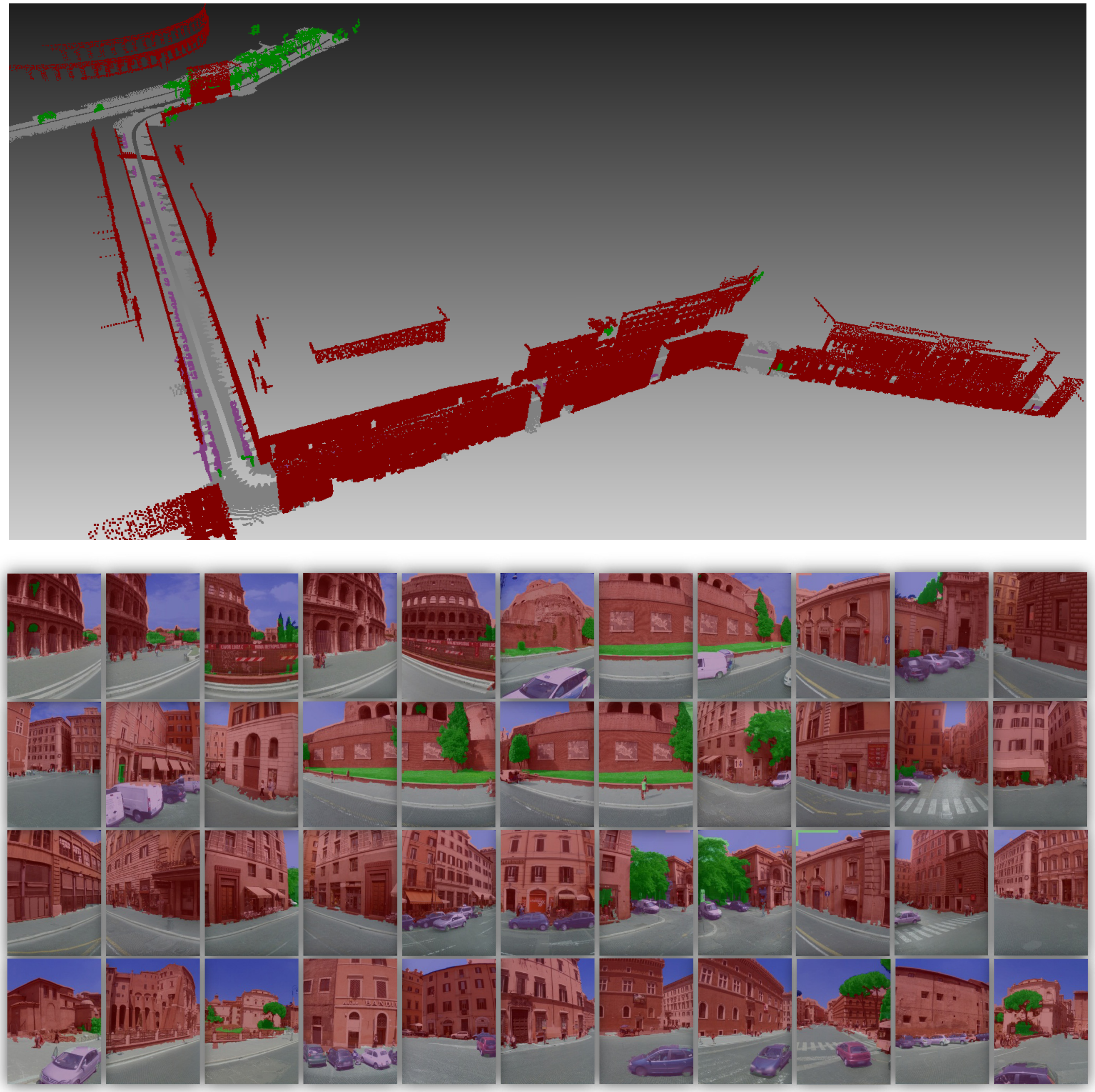


Fig. 9. Examples of the parsing results obtained by our method.

[10] D. Munoz, N. Vandapel, and M. Hebert, “Onboard contextual classification of 3-d point clouds with learned high-order markov random fields,” *International Conference on Robotics and Automation*, 2009.

[11] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, “Google street view: Capturing the world at street level,” *Computer*, vol. 43(6), pp. 32–38, 2010.

[12] S. Bileschi, “Streetscenes: Towards scene understanding in still images,” Ph.D. dissertation, MIT, 2007.

[13] H. Zhang, J. Xiao, and L. Quan, “Supervised label transfer for semantic segmentation of street scenes,” *European Conference on Computer Vision*, 2010.

[14] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr, “Urban 3d semantic modelling using stereo vision,” *IEEE International Conference on Robotics and Automation*, 2013.

[15] M. Lhuillier and L. Quan, “Quasi-dense reconstruction from image sequence,” *European Conference on Computer Vision*, 2002.

[16] C. Haene, C. Zach, R. A. Cohen, and M. P. Angst, “Joint 3d scene reconstruction and class segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[17] G. Medioni, M. S. Lee, and C. K. Tang, *A Computational Framework for Feature Extraction and Segmentation*. Elsevier Science, 2000.

[18] J.-F. Lalonde, N. V, D. F. Huber, and M. Hebert, “Natural terrain classification using three-dimensional ladar data for ground robot mobility,” *Journal of Field Robotics*, vol. 23, pp. 839–861, 2006.

[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “Slic superpixels,” *EPFL Technical Report*, 2010.

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context,” *European Conference on Computer Vision*, 2006.

[21] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, “Associative hierarchical crfs for object class image segmentation,” *International Conference on Computer Vision*, 2009.

[22] C. E. Brodley and M. A. Friedl, “Identifying mislabeled training data,”

Journal of artificial intelligence research, vol. 11, pp. 131–167, 1999.

[23] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[24] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

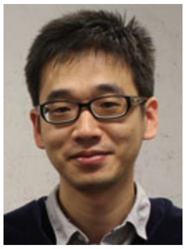
[25] L. Ladicky, C. Russell, P. Kohli, and P. Torr, "Associative hierarchical crfs for object class image segmentation," *International Conference on Computer Vision*, 2009.

[26] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Comptuer Vision*, vol. 88(2), pp. 303–338, 2010.

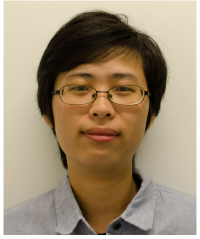
[27] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[28] H. Zhang, T. Fang, X. Chen, Q. Zhao, and L. Quan, "Partial similarity based nonparametric scene parsing in certain environment," *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[29] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

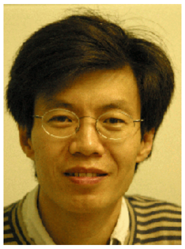


Honghui Zhang received his BS degree in Electronic and Information Engineering from Tongji University in 2005, and MA degree in Signal and Information processing form University of Science and Technology of China in 2008. In 2012, he received the PhD degree in Computer Science from HKUST (the Hong Kong University of Science and Technology). His research interests include scene parsing, object recognition and tracking, image-based modeling.



Jinglu Wang is an PhD student in the Hong Kong University of Science and technology. She received her BS degree on Computer Sience and Technology from Fudan University in 2011. Her research interests include 3D reconstruction, image-based modeling and scene parsing.

Tian Fang received the bachelor and master degree in Computer Science and Engineering from the South China University of Technology, China, in 2003 and 2006, respectively, and received the Ph.D. degree in Computer Science and Engineering from the Hong Kong University of Science and Technology (HKUST) in 2011. He is now a postdoctoral researcher in HKUST. His research interests include image-based modeling, image segmentation, recognition, and photo-realistic rendering. He now works on projects related to real-time 3D reconstruction and recognition.



Long Quan is a Professor of the Department of Computer Science and Engineering and the Director of Center for Visual Computing and Image Science at the Hong Kong University of Science and Technology. He received his Ph.D. in 1989 in Computer Science from INPL, France. He entered into the CNRS (Centre National de la Recherche Scientifique) in 1990 and was appointed at the INRIA (Institut National de Recherche en Informatique et Automatique) in Grenoble, France. He joined the HKUST in 2001. He works on vision geometry, 3D reconstruction and image-based modeling. He supervised the first Best French Ph.D. Dissertation in Computer Science of the Year 1998 (le prix de thèse SPECIF), the Piero Zamperoni Best Student Paper Award of the ICPR 2000, and the Best Student Poster Paper of IEEE CVPR 2008. He co-authored one of the six highlight papers of the SIGGRAPH 2007. He was elected as the HKUST Best Ten Lecturers in 2004 and 2009. He has served as an Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) and a Regional Editor of Image and Vision Computing Journal (IVC). He is on the editorial board of the International Journal of Computer Vision (IJCV), the Electronic Letters on Computer Vision and Image Analysis (ELCVIA), the Machine Vision and Applications (MVA), and the Foundations and Trends in Computer Graphics and Vision. He was a Program Chair of IAPR International Conference on Pattern Recognition (ICPR) 2006 Computer Vision and Image Analysis, is a Program Chair of ICPR 2012 Computer and Robot Vision, and is a General Chair of the IEEE International Conference on Computer Vision (ICCV) 2011. He is a Fellow of the IEEE Computer Society.