Geodesic Propagation for Semantic Labeling

Qing Li, Xiaowu Chen*, Yafei Song, Yu Zhang, Xin Jin, and Qinping Zhao

***Abstract*—This paper presents a semantic labeling framework with geodesic propagation. Under the same framework, three algorithms are proposed, including geodesic propagation (GP), supervised geodesic propagation (SGP) for image and hybrid geodesic propagation (HGP) for video. In these algorithms, we resort to the recognition proposal map and select confident pixels with maximum probability as the initial propagation seeds. From these seeds, the GP algorithm iteratively updates the weights of geodesic distances until the semantic labels are propagated to all pixels. On the contrary, the SGP algorithm further exploits the contextual information to guide the direction of propagation, leading to better performance but higher computational complexity than GP. For video labeling, we further propose the HGP algorithm, in which the geodesic metric is used in both spatial and temporal spaces. Experiments on four public datasets show that our algorithms outperform several state-of-the-art methods. With the geodesic propagation framework, convincing results for both image and video semantic labeling can be obtained.**

***Index Terms*—Semantic Labeling, Geodesic Propagation, Label Transfer, Indicator, Video Labeling**

I. Introduction

SEMANTIC labeling, or semantic segmentation, is a basic topic in computer vision and image understanding. The main objective of semantic labeling can be described as assigning a specific category label to each pixel in an image or a video frame. In recent years, many semantic labeling approaches (e.g., [1]–[11]) have been proposed due to the rapid development of recognition [12]–[17] and segmentation [18]–[22] algorithms. Among these approaches, some of them proposed to solve this problem with learned generative or discriminative models [9], [10], [23], [24]. These approaches often require training datasets containing fixed categories. Typically, most learning-based approaches construct conditional random fields (CRFs) over pixels (or superpixels, which are small coherent regions) and their major differences lie in the energy functions and inference algorithms. Since such learning-based models often need to be trained over dataset with fixed categories, they may fail when encountering objects in unknown categories. In this case, we need to re-train the whole model with additional categories so as to adapt to the new objects.

With the increasing availability of large-scale image collections (e.g, LabelMe [25], Flickr database [26]), the data-driven methods have been proposed for nonparametric semantic segmentation [27]–[29]. Among these methods, the concept of *label transfer* is proposed, which aims to transfer the labels from the annotated similar images to an input image. Usually, these methods have to address two key issues. The first issue is how to retrieve similar images from a large-scale database for a given input image. The second issue is how to parse the input image with the annotated similar images. In these two issues, the former has been well studied in previous works (e.g., [30], [31]), while the latter needs a precise matching between the retrieved images and the input image, which remains a challenging task. Typically, these data-driven approaches require complicated matching process between all pixels (or superpixels) from the retrieved images and the input image. Such matching process can be extremely time-consuming and often require certain kinds of post-smoothing through CRF or MRF (markov random field) optimization.

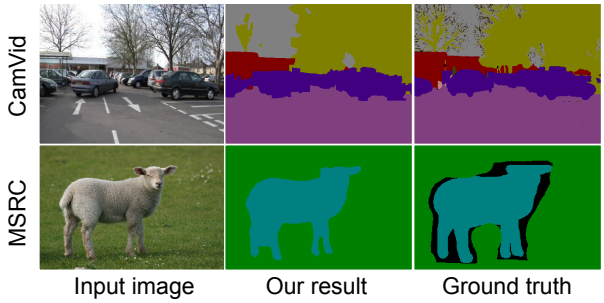

Fig. 1. Our objective is to get the semantic labeling of the input image. This figure shows our results on CamVid dataset and MSRC dataset. Best viewed in color.

In most cases, the CRFs are slow in solving many vision applications and can obtain optimal solution only for limited functions [8]. Besides, some foreground/background segmentation approaches [3], [18], [32], [33] work robustly by employing the geodesic distance metric. These methods avoid constructing CRFs and have no energy minimization step. Their solutions are obtained by labeling the pixel with a specific category, which has the smallest geodesic distance from known seeds to the pixel. In order to avoid the disadvantages of CRFs, we also resort to geodesic distance metric. To the best of our knowledge, geodesic metric is well studied in object segmentation or matting, while less studied in semantic labeling. The user-aided scribbles provide sufficient information of foreground/background objects [33]–[35], and are commonly taken as the initial seeds. However, these scribbles lack category information. Hence, to apply geodesic distance, we need to address the problem of generating appropriate category seeds first. Here, we benefit from recent recognition methods and propose an approach to select seeds in an automatic manner.

To make our semantic labeling more adaptive to the number of category, we follow the idea of *label transfer*. Instead of using the precise matching in typical label transfer methods, we

Qing Li, Xiaowu Chen, Yafei Song, Yu Zhang, Xin Jin and Qinping Zhao are with State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, 100191,

Xiaowu Chen is the corresponding author. E-mail: chen@buaa.edu.cn

take partly the matching of a little number of superpixels. In addition, we find that the contextual information is important in revealing the relationship of different categories [23], [36]–[39]. For example, the *cow* usually appears in concurrence with the *grass* in the natural scenes. It is observed that similar images have similar objects and contexts. Therefore, we take advantage of contextual information in similar images to guide the labeling process of an input image, which is one of the key differences between our approach and previous label transfer methods.

In this paper, we propose a framework for the semantic labeling of both image and video, including two algorithms for image and one algorithm for video. The two algorithms for image are denoted as geodesic propagation (GP) and supervised geodesic propagation (SGP). The major difference between these two is that SGP follows the label transfer idea and utilizes contextual information, while GP does not. In our framework, GP performs faster than SGP, while SGP obtains more accurate results. We propose these two algorithms to give users choices according to their needs. These two algorithms have similar pipeline: given the recognition proposal map of the image, confident pixels with maximum probability are selected as the initial seeds. The geodesic distance is then defined on a manifold. Based on the geodesic distance, the semantic labels are simultaneously propagated from the initial seeds to the rest of image pixels. In our SGP, an indicator implying the contextual information of similar images is exploited to guide the propagation. For video labeling, a hybrid geodesic propagation algorithm is proposed as an extension. The geodesic metric is used in both spatial and temporal spaces. Results of experiments on four public datasets show that our algorithms outperform several learning-based methods as well as label transfer models. In addition, a comprehensive analysis is conduct to reveal the roles of different parts in our framework (e.g., the performance of different features).

Our main contributions include: (1) We propose a semantic labeling framework based on geodesic distance metric, consisting of three propagation algorithms. (2) SGP algorithm exploits the contextual information of similar image to guide the propagation. (3) Geodesic propagation is extended to video semantic labeling.

The rest of this paper is organized as follows. Related works are reviewed in Sect. II. The system framework is briefly described in Sect. III. Our GP and SGP algorithms are described in Sects. IV and V, respectively. The extension of geodesic propagation to video labeling is presented in Sect. VI and experiments are conducted in Section VII. Finally, we conclude the whole article in Sect. VIII.

II. Related works

We will briefly review related work in two aspects: semantic labeling and geodesic segmentation.

Semantic labeling. Shotton *et al.* [7] detail a novel approach for learning a discriminative model of object classes, which incorporating texture, layout, and context information efficiently in a CRF model. To make efficient training, they exploit both random feature selection and piecewise training methods. Xiao and Quan [40] propose a powerful multi-view semantic segmentation framework for street scene images with MRF model. In their work, a multi-view semantic segmentation method is developed to recognize and segment each image into semantically meaningful regions. Similar to [40], [41], the GIST feature and boosting classifier are adopted in our system, however, we use geodesic propagation to get the deterministic solution while [40], [41] use CRF or MRF to get the optimal solution. In addition, we utilize the contextual information of both the similar images and the test image itself, while [40], [41] utilize only the contextual information of the test image or test sequence.

Liu *et al.* [27] address the semantic labeling in a nonparametric scene parsing approach, which is denoted as *label transfer*. They utilize a coarse-to-fine SIFT flow algorithm to build correspondence between similar images and the test image. Based on the dense scene correspondence, their system warps the existing annotations, and integrates multiple cues in a MRF framework to label the test image. Though it is a promising work, there still exists much room for improvement in terms of efficiency. Another nonparametric image parsing work is proposed by Tighe and Lazebnik [28]. It works by scene-level matching with global features, superpixel-level matching with local features, and MRF optimization for incorporating neighborhood context. These two nonparametric methods both require precise matching in an existing large-scale database. Moreover, without retrieved proper images, the parsing performance will be severely affected.

Based on the image parsing idea of these two methods [27], [28], a certain scene parsing method has been proposed to address the proper matching in [29]. To assure that the retrieved images are proper for label transfer, Zhang *et al.* [29] firstly obtain multiple image sets to cover all semantic categories in the input image. Then a KNN-MRF matching scheme is proposed to build dense correspondence between the input image and each retrieved image sets. A MRF optimization is used based on those matching correspondences. Compared with these label transfer methods, our framework utilize the similarity of retrieved images without precise pixel-level or superpixel-level matching.

Geodesic segmentation. Geodesic distance is the shortest path between two points in a feature space. It is used as a metric to classify pixels by Bai and Sapiro [18]. They obtain the weighted geodesic distance on the base of spatial and temporal gradients, thus can compute the distance and segment the image in linear complexity. The approach proposed by Price *et al.* [3] is also based on the idea of geodesic distance. To avoid the bias of seed placement and the lack of edge constraint in geodesic, Price *et al.* [3] combine geodesic distance information with edge information in a graph cut optimization framework. Gulshan *et al.* [32] introduce Geodesic Forests, which exploits the structure of shortest paths in implementing the star-convexity constraints. The star-convexity prior is used in an interactive setting. Inspired by these works, we adopt the geodesic distance metric in our framework.

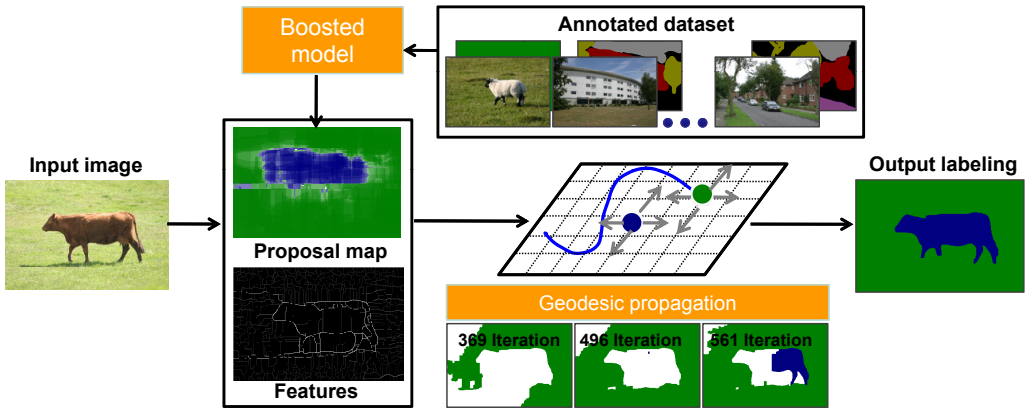

Fig. 2. The pipeline of our framework. Given the input image, we infer its proposal map using the boosted model learned on the annotated dataset. The initial seeds for geodesic propagation are selected based on the proposal map. The initial geodesic distance map is defined as: the higher probability, the smaller distance. With the computed weights based on features extracted from the image, the geodesic distance is updated during propagation. Finally the labels of seeds are propagated to the rest image pixels.

III. Framework

In our framework, we propose three algorithms based on geodesic metric. The basic pipeline of the two algorithms for image segmentation are shown in Fig. 2. The HGP is an extension by applying geodesic propagation in spatial-temporal space. All the three algorithms in our framework follow the basic idea of geodesic propagation. In this section, we will describe how we introduce the geodesic distance into our multi-class labeling work.

Inspired by Bai and Sapiro [18], we take the definition that geodesic distance is the smallest integral of a weight function over all possible paths from the seeds to a pixel [18]. Our objective is to assign each pixel a category label which has the smallest geodesic distance. We use geodesic distance to measure the weight between points in the feature space. Let the image be represented by a graph G, where each pixel is a vertex $v \in V$ and each edge connects two neighboring pixels. The weight W of edge E on G denotes the smoothness relationship between neighboring pixels. Many features can be embedded into the weights. To propagate correct labels, we exploit intra contextual information of the input image by using various features, such as color, texture and boundary features. To take account for spatial adjacency, an important factor in image labeling, we define a path C between any two vertices v',v as:

$$C(v', v) = (v' = v_0, v_1, \ldots, v_n = v), \tag{1}$$

with v_i,v_{i+1} being adjacent neighbors on G.

For any vertex v on G, the geodesic distance of category l is defined as the minimum weighted distance $d_l(v, s_l|C)$ from v to a closest seed $s_l \in \Omega_l$. Ω_l is the seed set of category l. The formulation is given by

$$D_l(v) = \min_{s_l \in \Omega_l} \min_{C(v,s_l)} d_l(v, s_l|C). \tag{2}$$

Based on the definition of geodesic distance, the vertex v is labeled as:

$$L(v) = l^* = \arg\min_{l \in L} D_l(v). \tag{3}$$

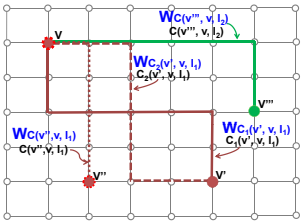

Fig. 3. Illustration of geodesic distance. The red vertex v' and v'' belong to the same class l_1, and the green vertex v''' belongs to the class l_2. The vertex v is an unlabeled point and needed to be assigned a class label. Suppose there are two paths from v' to v (one solid red line and one dashed red line), and one path from v'' to v (one dotted red line). The distance of v to l_1 is the shortest one between $C(v', v, l_1)$ and $C(v'', v, l_1)$. Without loss of generality, we suppose $W_{C(v'',v,l_1)} < W_{C_2(v',v,l_1)} < W_{C_1(v',v,l_1)}$. Then $D_l(v)$ is $W_{C(v'',v,l_1)}$. Meanwhile, the distance of v to l_2 is $W_{C(v''',v,l_2)}$. Suppose $W_{C(v'',v,l_1)} < W_{C(v''',v,l_2)}$, then according to equation 3, v is assigned the label of v'' (l_1, visualized in red).

As illustrated in Fig. 3, the vertex v' and v'' visualized as red points belong to the same class l_1, and the vertex v''' visualized as a green point belongs to the class l_2. The vertex v is an unlabeled point and needed to be assigned a class label. Suppose there are two paths from v' to v and one path from v'' to v. The geodesic distance of each path is denoted as W, which is computed as the accumulation of $W(i, j)$ on the path. According to equation 2, the distance from v to the class l_1 is the shortest one between $C(v', v, l_1)$ and $C(v'', v, l_1)$. Without loss of generality, we suppose it is $C(v'', v, l_1)$. Meanwhile, the distance from v to the class l_2 is $C(v''', v, l_2)$. Then according to equation 3, the vertex v is assigned the class l_1(in red) by comparing $C(v'', v, l_1)$ with $C(v''', v, l_2)$.

Based on the above definition, we apply geodesic distance to semantic labeling. Since geodesic framework is sensitive to the initial seeds, seed localization is important. Different from the interactive way used in previous works [3], [18], we localize robust seeds automatically inspired by recent work on recognition [7]. The geodesic weight is set in a color feature space for GP algorithm, and in hybrid feature space of color and boundary for SGP algorithm. An obvious differ-

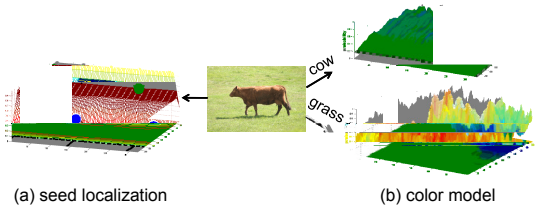

Fig. 4. (a) Robust seeds are localized by Mean-shift estimation on semantic proposal map. (b) Probability density estimation for horse class.

ence between GP and SGP is that an indicator implying the contextual information is exploited to guide the propagation in SGP. Additionally, we extend geodesic propagation to video semantic labeling. The geodesic metric is used in both spatial and temporal spaces.

IV. Geodesic propagation

Given an input image I, the semantic labeling is to assign each pixel $x \in I$ with a certain label $l \in L$, where $L = \{1, \cdots, N\}$. We build up a 4-neighbor-connected graph $G = \langle V, E \rangle$ on the image lattice. We then select robust seeds to propagate labels throughout the whole image. Parts of this section have previously appeared in [42].

A. Seed localization

Previous works [3], [18], [32] utilize manual scribbles, which take subjective prior into the segmentation task, as the initial seeds for each category or object. Instead, we take a dynamic seed selection strategy for geodesic propagation. To exploit the inherent possibility in training image, we learn a recognition model based on the training set, and predict the recognition proposal for each input image with this recognition model. The 17 dimension raw texton features and Joint Boost algorithm [7] are used in the learning of recognition model.

The recognition proposal map of an input image gives a rough distribution of objects. Based on this proposal map, we localize robust vertices, which have the maximum probabilities for each category, as the initial seeds. Each selected seed $s \in S$ is assigned with the corresponding label l_s, and the initial geodesic distance $d(s)$ is derived from the estimated probability $p^{ms}(s)$ using Mean-shift:

$$d(s) = logp^{ms}(s), s \in S. \tag{4}$$

B. Edge weight

In order to propagate confident labels to appropriate pixels, various image features should be elaborately designed. These features can indicate real objects distribution on the input image, and guide geodesic propagation approaching to the inherent labeling. Based on the recognition proposal, we incorporate global color features to measure the weight of edges on G.

Usually, the color distribution is diverse across all objects of a category. However, in the input image, the estimated color distribution across a few instances is compact. Such estimated color distribution captures more precise image-specific appearance feature than former class-specific recognition system. Moreover, it can capture several clusters in feature space, demonstrating a capacity to handle inner-class variety.

Here we calculate the expected color histogram $H(x|l_i)$ of the recognition proposal $p(l|I)$ for each category. Under the assumption of Gaussian Mixture Model in HSV color space, we estimate the color likelihood $p(x|l)$ through EM algorithm. By Bayesian theorem, we further calculate the posterior probability $p(l|x) = p(x|l)p(l)/\sum_i p(x|l_i)$ which indicates how likely the pixel x belongs to the label l. Fig. 4(b) illustrates probability estimation for *cow* and *grass* in HSV color space. With the definition above, the edge weight is defined as:

$$w^c(x, x'|l) = \frac{\|p(l|x) - p(l|x')\|}{p(l|x) + p(l|x')}. \tag{5}$$

C. Propagation

According to Section III, the solution to semantic labeling is formulated as how to compute the shortest geodesic path from each pixel to the initial seeds. We propose a geodesic propagation algorithm generalized from Fast Marching Algorithm [43] to simultaneously propagate the geodesic distance of all classes efficiently. For a pixel x, if one label l_i is propagated to x earlier than other labels, then the corresponding geodesic distance $D_l(x)$ to l_i is shorter than others. We propagate all labels simultaneously to the entire image, and once the geodesic path of label l_i reaches pixel x, its shortest geodesic distance $min_{l \in L} D_l(x)$ is determined. The time complexity of the algorithm is independent of the number of categories, thus our algorithm is more efficient than other labeling algorithms especially for multi-class labeling, as shown in Fig. 5(a).

Algorithm 1 Geodesic propagation algorithm

Require: candidate seed set $S = \{v_i : (x_i, y_i, p_i, l_i), i \in [1, S], l_i \in [1, N]\}$.
Ensure: label set $L = \{l_i\}$, where $l_i \in \{1, \cdots, N\}$

1: Put all nodes v_i into unlabeled set Q;
2: Push S into reachable queue Q_R;
$v_i \leftarrow$ head of Q_R, and put v_i into labeled set L;
3: choose any neighbor node $\{v_j\}$ of v_i;
4: push v_j in Q into Q_R;
5: Update v_j in Q_R
if $D_i + W_{ij} < D_j$ **then**
update D_j with $D_i + W_{ij}$;
assign l_i to l_j;
else
D_j and l_j remain;
end if
6: Repeat (3) to (5) until Q_R is empty.

During geodesic propagation, each vertex has three statuses: labeled, reachable and unlabeled. The labeled vertex is assigned label determinately, as well as its minimal geodesic distance. The set of reachable vertices includes the neighbors around the labeled vertices. The reachable vertices are sorted according to their geodesic distance and put into the ordered

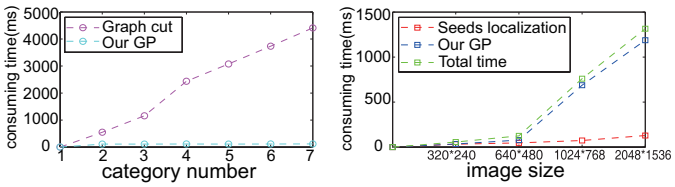

Fig. 5. The efficiency of our algorithm. The complexity of GP algorithm is (a) independent of the number of categories, and (b) increases linearly with the image size.

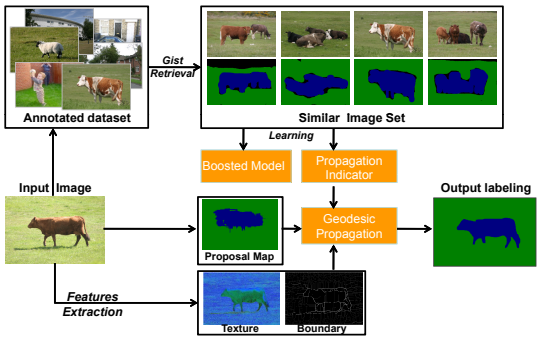

Fig. 6. The workflow of SGP algorithm. Given the input image, we get its similar image set from the annotated dataset using Gist retrieval (Section V-A). The boosted model is trained on this similar image set. Besides, the propagation indicator implying the contextual information is trained on the similar image set as well (Section V-C).

queue Q_R. Other vertices are marked as unlabeled to indicate that the geodesic propagation has not reached them yet.

Our algorithm iteratively selects the vertex v_i of the minimum distance in the reachable queue Q_R, sets v_i as labeled, and propagates labels to its neighboring vertices, until the reachable queue is empty.

Similar to the Dijkstra algorithm, our algorithm can efficiently compute the best labeling with minimal distance from each pixel to the initial seeds. The time complexity of the proposed algorithm only depends on the image resolution. Moreover, we implemente an untidy queue introduced in [44]. The computational time increases linearly with the image size, as shown in Fig. 5(b).

V. Supervised Geodesic Propagation

The GP algorithm encodes less contextual relationship between objects. Therefore, we propose a supervised geodesic propagation algorithm encoding context constraints. Since SGP utilizes the contextual information of similar images, it is similar to semantic label transfer method [27], in that it transfers the labels from the annotated similar images to the input image. In this section, we describe our SGP algorithm, as well as the difference between SGP and GP. Parts of this section have previously appeared in [45].

The pipeline of our SGP algorithm is illustrated in Fig. 6. Given an input image, we first obtain the similar image set from the annotated dataset using Gist matching [31]. We infer the proposal map of the input image for seed selection. Then the proposal map, the texture and boundary features of the input image, and the contextual similarity of the similar images are integrated into geodesic propagation to get semantic labeling result.

In this section, each superpixel sp_i is a vertex v in graph G, and is assigned a specific label l contained in the dataset through geodesic propagation procedure. The edge set E consists of the edges between neighboring vertices. We define the weight of edge $W(sp_i, sp_j)$ on a hybrid manifold, incorporating texture and boundary features. The edge weight indicates the smoothness between neighboring vertices sp_i and sp_j.

A. Similar image matching

In order to transfer the labels of annotated images to the input image, first we need to select proper images. These proper images have similar semantic categories and contextual information to those of the input image. How to retrieve proper similar images is not the main focus of this paper. Thus we use gist matching which is commonly used in recent label transfer methods [27]–[29]. The gist descriptor [31] is employed to retrieve the K-Nearest Neighbors from the dataset, and similar image set is formed with these neighbors. After gist matching, the K-Nearest neighbors in the similar image set are re-ranked in the following way. We over-segment the input image and each of its similar image $R \in \{R\}$ using the algorithm described by Arbelaez *et al.* [19]. Then each superpixel $sp_i \in I$ is matched to a proper superpixel $r(sp_i) \in R$ which has the smallest matching distance to sp_i. The following distance metric is used to compute the matching distance in the re-ranking procedure. Given two images I and R, the matching distance $D_r(I, R)$ is scored as:

$$D_r(I,R) = \sum_{sp_i \in I, r(sp_i) \in R} \left\| (fv_{sp_i} - fv_{r(sp_i)}) \right\|^2, \quad (6)$$

where fv_{sp_i} is a 22 dimension descriptor of sp_i, including average HSV colors, coordinates and 17 dimension filter responses [7] of sp_i (other filter responses can also be used, e.g., [46]). The Euclidean distance metric is used in our implementation. After re-ranking the gist similar images according to their matching scores, we get the top K similar images, which are denoted as $\{R_K\}$. In our experiments, we use $\{R_K\}$ as the similar image set instead of $\{R\}$.

B. Proposal map for seed selection

The images in $\{R_K\}$ imply the possible categories in the input image. We assume that categories $l \in \{R_K\}$ cover all the categories in I. To exploit the inherent possibilities, the similar image set R_K is used as the training set for the input image I to learn the recognition model. The training process is identical to that of Section IV-A.

When we get the recognition proposal map of I, the initial geodesic distances of all classes for each superpixel are defined according to this proposal map. Each superpixel is temporarily assigned the initial label which has the maximal probability $p_l(sp_i)$. According to equation 7, we get the initial geodesic distance of each superpixel with their temporary class. Then a distance map of the input image can be obtained:

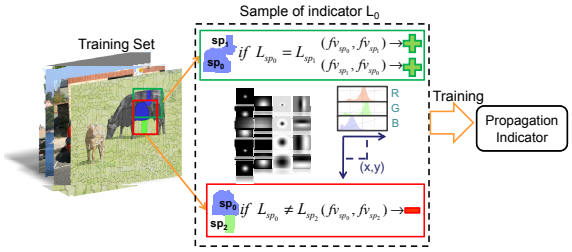

Fig. 7. Sample of indicator. Green cross means positive sample while red minus means negative sample.

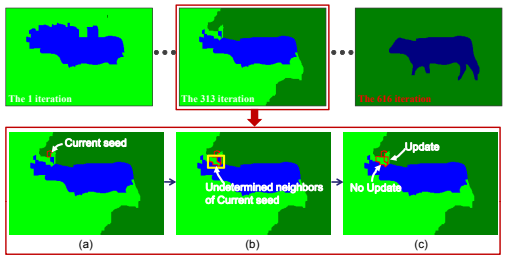

Fig. 8. Illustration of supervised propagation. Dark green means superpixels determinately labeled as the *grass* and dark blue determinately the *cow*. Light green means superpixels temporarily labeled as the *grass* and light blue temporarily the *cow*. Take 313 iteration for details. (a) In the first status, the current seed is selected and its label is determined as *grass*. (b) In the second status, we get the undetermined neighboring superpixels of current seed. We only show parts of its neighbors for clarity. The intra features of the input image and our indicator jointly decide that whether to update the current label and distance of these neighhors. (c) In the third status, one neighbor is updated while the other is not. Then it is ready for the next iteration.

superpixels with higher probabilities will have smaller initial geodesic distance. In each propagation step, the undetermined superpixel with the smallest distance of all classes is selected as the current seed (see Section V-E for more details).

$$Dis_{initial}(sp_i) = 1 - p_l(sp_i). \tag{7}$$

C. Propagation indicator

Each image $R \in \{R_K\}$ is similar to the input image in some aspects, such as the appearance and the contextual information. Thus we assume that the contextual similarity between the similar image set and the input image can provide useful information for parsing the input image. Based on this assumption, we take a supervised indicator for label propagation. A set of classifiers is learned on the similar image set to guide the propagation, and each semantic category has its corresponding classifier. We denote these classifiers as the propagation indicators. In this section, we introduce how to get the indicator of each category. More details about how these indicators work in the propagation procedure will be introduced in Section V-E.

Our indicator is used to classify whether to propagate label from superpixel sp_i to its neighbor sp_j in the input image. For neighboring superpixels which are classified as the same category, we propagate current label; otherwise we do not. Our propagation indicators for each category are trained using random forests [47], [48], a competitive non-linear model that predicts by averaging over multiple regression trees. The random forests implementation available online [49] is used with default parameters in our algorithm.

To generate training samples for the indicator of each category l in $\{R_K\}$, we get all neighboring superpixel pairs (sp_i, sp_j) as well as their category labels l_{sp_i} and l_{sp_j}, according to the annotation. Note that pair (sp_i, sp_j) is different from pair (sp_j, sp_i). For each pair (sp_i, sp_j), we denote $fv(sp_i, sp_j) = \langle fv_{sp_i}, fv_{sp_j} \rangle$ as a 44 dimension feature vector, which includes 22 dimensional features of both fv_{sp_i} and fv_{sp_j}. If l_{sp_j} is consistent with l_{sp_i}, then $fv(sp_i, sp_j)$ is taken as a positive sample of the indicator of label l_{sp_i}, otherwise a negative sample. All the features in fv are normalized in the range of $[0, 1]$. In the testing procedure, we extract the $fv(sp_i, sp_j)$ feature vector of neighboring superpixels, and then get the confidence produced by the trained classifier as an indicator value for propagation. As shown in equation 8, $T_l(sp_i, sp_j)$ is the indicator function, $con_l(sp_i, sp_j)$ is the confidence and ϕ is the threshold for indicator. The illustration of the indicator sample is shown in Fig. 7.

$$T_l(sp_i, sp_j) = 1[con_l(sp_i, sp_j) > \phi]. \tag{8}$$

D. Edge weight

To measure the weight of edge W_{sp_i,sp_j} on graph G, here we integrate two components: the texture component and the boundary component. The weight function W between neighboring vertices is demonstrated in equation 9. Regions of different categories can commonly present apparent texture disparities. Thus we use a texture descriptor to measure the $W_{texture}(sp_i, sp_j)$ with Euclidean distance metric. This texture descriptor consists of average HSV colors and 17 filter responses features [23].

The boundary, as significant local changes, carries strong information for object distinction. In this section, we apply reliable Berkeley edge detector [19] combining color, brightness and texture cues to capture the boundary confidence. The weight function for boundary component W_{bdry} is defined in equation 10, in which θ is the threshold for boundary confidence $P_b(\cdot)$. We detect the boundaries at pixel level and then convert these boundary confidences into superpixel level.

$$W(sp_i, sp_j) = \lambda_1 W_{texture}(sp_i, sp_j) + \lambda_2 W_{bdry}(sp_i, sp_j), \tag{9}$$

where $W_{texture}$ and W_{bdry} are obtained by the texture component and the boundary component, respectively. λ_1 and λ_2 are tuning parameters.

$$W_{bdry}(sp_i, sp_j) = P_b(sp_i, sp_j, \theta). \tag{10}$$

E. Supervised propagation

Our supervised propagation algorithm starts with the initial geodesic distance and initial labels for all the vertices. Similar to algorithm 1, undetermined vertices will be put into the unlabeled set Q and sorted for current seed selection, which has minimum geodesic distance. Once a vertex is selected as seed in a step, it is removed out of the set Q with its semantic label being determined. The difference from algorithm 1 is that

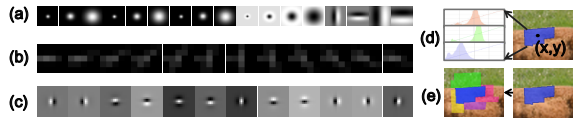

Fig. 9. Multiple features. (a) The 17 filter responses [23], including texture pattern and layout. (b) The 12 dimension DOOG filters responses [50]. (c) The 12 dimension texton responses [50]. (d) The color and the position of each superpixel. (e) The contextual texture of neighboring superpixels.

the weight of edge $W(sp_i, sp_j)$ and propagation indicator are integrated in the propagation iteration to decide how to update geodesic distance. In each step of geodesic propagation, the geodesic distance between a labeled seed and its neighboring undetermined superpixels have to be updated according to corresponding indicator. Suppose sp_i is labeled as l_{sp_i} and sp_j has undetermined label, then employ the propagation indicator of category l_{sp_i} to get the indicator confidence value $T_l(sp_i, sp_j)$. Thus the contextual similarity between the input image and its similar image set can guide its semantic labeling. Our supervised propagation algorithm is summarized in Algorithm 2. More details are illustrated in Fig. 8.

Algorithm 2 Supervised geodesic propagation algorithm

Require: Vertices with initial geodesic distance and label.
Ensure: Label set $L = \{l_{sp_i}\}$, where $sp_i \in superpixel\{I\}$, $l_{sp_i} \in \{1, \cdots, N\}$

1: Put all vertices into unlabeled set Q;
2: $v_i = min_Q(Dis(Q))$, set $l_{v_i} = current(l_{v_i})$ put v_i to labeled set L, remove v_i from Q.;
3: Get neighbor set $\{v_j\}$ of v_i,$\{v_j\} \subset Q$;
4: Update geodesic distance;
for each $v_j \in \{v_j\}$ **do**
  if $(W(v_i, v_j) < \theta_e)$ and $T(v_i, v_j)$ is true **then**
    update $Dis(v_j)$ with $Dis(v_i) + \kappa W(v_i, v_j)$;
    assign l_{v_i} to l_{v_j};
  else
    $Dis(v_j)$ and l_{v_j} remain;
  end if
end for
5: Repeat (2) to (4) until Q is empty.

F. Variations on parts and features

Our SGP framework consists of the boosted model, the indicator, the edge weight and the propagation algorithm. Each part is designed intuitively, so that our algorithm can work efficiently. We figure out the effect of each part in the experiments (Section VII-C). The algorithm implemented in each part can also be replaced by other promising algorithms if necessary.

The simple 22 dimension features we use in our SGP framework exploit the texture layout, the texture pattern, the color and the position of each superpixel (as shown in Fig. 9). We try to exploit more features on both texture layout and pattern.

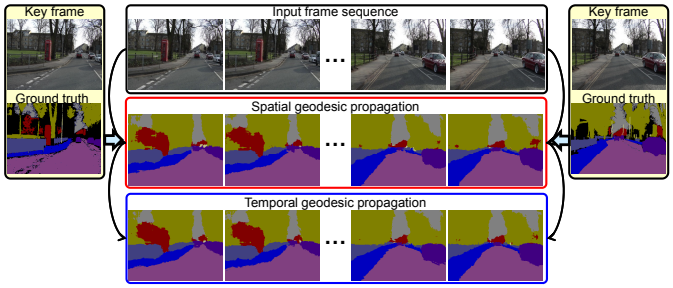

Fig. 10. The pipeline of our hybrid geodesic propagation for video. Given the key frames as well as their label annotations, we propagate the semantic labels throughout the whole video. In both spatial and temporal propagation, geodesic distance is exploited to propagate accurate label.

Considering that we only take the features of each individual superpixel, we take the contextual texture of neighboring superpixels into the feature descriptor to test the performance of our framework. However, these contextual textures make no noticeably better results. Our simple 22 dimension features have exploited the texture layout well. We also try the texture pattern features used in [50]. The experimental results reveal that, these texture pattern features can improve the performance of our framework. The visualization of our features is illustrated in Fig. 9. More details about this experiment will be described in Section VII-C.

VI. Hybrid Geodesic Propagation

Many studies pay attention to video segmentation, such as video object segmentation [33], [51], [52] and video labeling [53]–[55]. However, these methods do not apply geodesic metric for video labeling. We extend our framework to video semantic labeling with a hybrid geodesic propagation. Since object in neighboring frames are temporally consistent, we propagate labels with geodesic distance in spatial-temporal space. Our hybrid geodesic propagation includes two folds of propagation: spatial geodesic propagation and temporal geodesic propagation. The pipeline is shown in Fig. 10. The annotated key frames are used as the training images. The semantic labels of these annotated key frames are propagated throughout the whole video. In the spatial geodesic propagation, we propagate label for each frame independently. In the temporal geodesic propagation, we propagate label throughout the whole spatial-temporal space. We first predict the labels for each frame using the supervised geodesic propagation as described in Section V. The geodesic propagation in Section IV can also be used. Then to smooth the temporal inconsistency between neighboring frames, we establish a geodesic-based MRF model.

In this section, we apply a pixel-level SGP framework in the spatial propagation step. The predicted labels for each single frame are used as the initial labels for the temporal propagation step. Since the spatial-temporal consistency is not considered in the SGP algorithm, the individual label results are not consistent in the spatial-temporal space. It is observed that the inconsistencies commonly appear around the boundaries between semantic objects, see Fig. 11 for example. Besides, computation on the whole video data is consuming. Hence, in

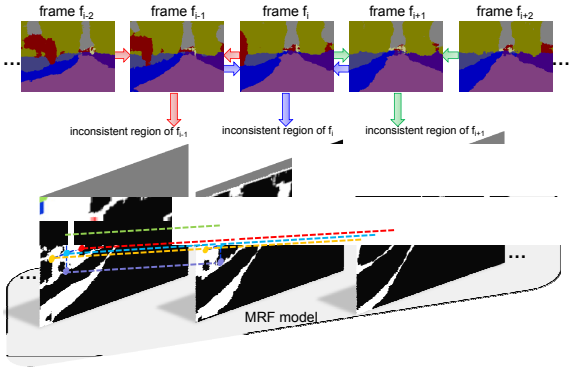

Fig. 11. Illustration of our MRF model. For frame f_i, comparing its label result of SGP with that of neighboring frame f_{i-1} and f_{i+1}, we obtain the inconsistent region of f_i. White pixels indicate the points which have different predicted labels in these frames. We establish MRF model based on the inconsistent region. Nodes in same color have temporal correspondences, and nodes in different color of a inconsistent region have spatial correspondences.

video propagation we focus on these inconsistent regions. To identify these regions, we match any two neighboring frames using the optical flow available at [56], and register them into a unified coordinate system. A point, which is predicted with different labels in neighboring frames, is identified as the inconsistent point. For a single frame, its inconsistent region consists of the projected pixels of the inconsistent points, compared with its forward and backward neighboring frames.

MRF model. In the spatial-temporal space, we establish a pixel-based MRF model, where each inconsistent pixel is a node in this model, and each edge connects two neighboring pixels. We denote neighboring pixels which have spatial relationship as spatial neighbors, and those have temporal correspondence as temporal neighbors. For a pixel, it has at most four spatial neighbors and two temporal neighbors. Since the inconsistent regions of each frame are not identical, the number of neighbors for each node is not fixed, ranging from one to six. We encode geodesic distance as well as the texture features into the MRF model. Fig. 11 illustrates our MRF model.

$$E(L|I) = \sum_i \psi_i(l_i) + \lambda \sum_{ij} \varphi_{ij}(l_i, l_j). \tag{11}$$

Geodesic distance as unary term. Based on SGP framework, we have the probability map for each frame. We localize the pixel which has the highest probability as the robust seed for each category in each frame. These seeds are excluded out of the inconsistent region. According to our definition of geodesic distance (Section III), we compute the geodesic distance of each node and convert the distance to the node potential. For a node i, its potential for category l is denoted as $1 - dis_i(l)$, where $dis_i(l)$ is its geodesic distance. We normalize these potentials and use them as unary term $\psi_i(l)$.

Pair-wise term. For an edge which connects node i and j in the MRF model, we compute the feature distance D_{ij} between i and j, and encode this D_{ij} into the pair-wise term φ_{ij}. We formulate φ_{ij} in equation 12, where n_c is the number of category. The feature we used in this section is a 23 dimensional feature vector, which is a combination of previous 22 dimensional feature and a frame index.

$$\varphi_{ij} = \begin{cases} (1 - D_{ij})/n_c, label_i = label_j \\ D_{ij}/({n_c}^2 - n_c), otherwise \end{cases}. \tag{12}$$

We adopt loopy belief propagation algorithm available at [57] to obtain the smoothed label prediction.

VII. Experiments

In this section we test the performance of our GP and SGP on several challenging datasets as well as the comparison between these two algorithms, and compare our results with several state-of-the-art works. The experiment of video propagation is demonstrated subsequently.

A. Dataset

In our experiment, we use four public datasets:

CamVid dataset. The Cambridge-driving Labeled Video database (CamVid, Brostow *et al.* [58]) is the first collection of videos with object class semantic labels. It provides 701 still images taken under different lighting conditions (day and dusk). The images in the original dataset are at the size of 960× 720 and cover 32 object classes. To make comparison with others, we group the dataset into 11 major categories and resize the images to 480×360 pixels as Zhang *et al.* [29]. The 11 categories are building, tree, sky, car, sign-symbol, road, pedestrian, fence, column-pole, sidewalk, and bicyclist. Besides, we use *void* label to indicate pixels not belonging to the 11 categories. In this dataset, 50% images are randomly split into the training set, and the left are used for testing.

MSRC dataset. The Microsoft Research Cambridge dataset (MSRC, shotton *et al.* [7]) is composed of 591 images of 21 object classes. We randomly split this dataset, 55% for training and 45% for testing, as suggested by Shotton *et al.* [7]. Each class contributes approximate proportion. The *void* label is used to cope with pixels not belonging to any class in the dataset, and the manual labeling is not aligned exactly with boundaries. Image in this dataset is at 320×213 resolution.

CBCL dataset. the CBCL StreetScenes dataset (CBCL, Bileschi [59]) contains 3547 still images of street scenes, which includes nine categories: car, pedestrian, bicycle, building, tree, sky, road, sidewalk, and store. The pedestrian, bicycle, and store are not included in our testing, which is same with the setting of Zhang *et al.* [29]. To compare with Zhang *et al.* [29], we resize the original images to 320×240. 50% images are randomly selected for training, and the left are used for testing.

LHI dataset. We use a subset of LHI dataset (Yao *et al.* [60]), including 400 images of 17 categories. We randomly split the dataset into 235 training images and the rest the testing images. Image resolution of this dataset is also 320×213.

B. GP performance

We test the performance of our geodesic propagation on the four datasets. The semantic labeling results are denoted as GP results for clarity. Fig. 12 shows the GP results of the four datasets.

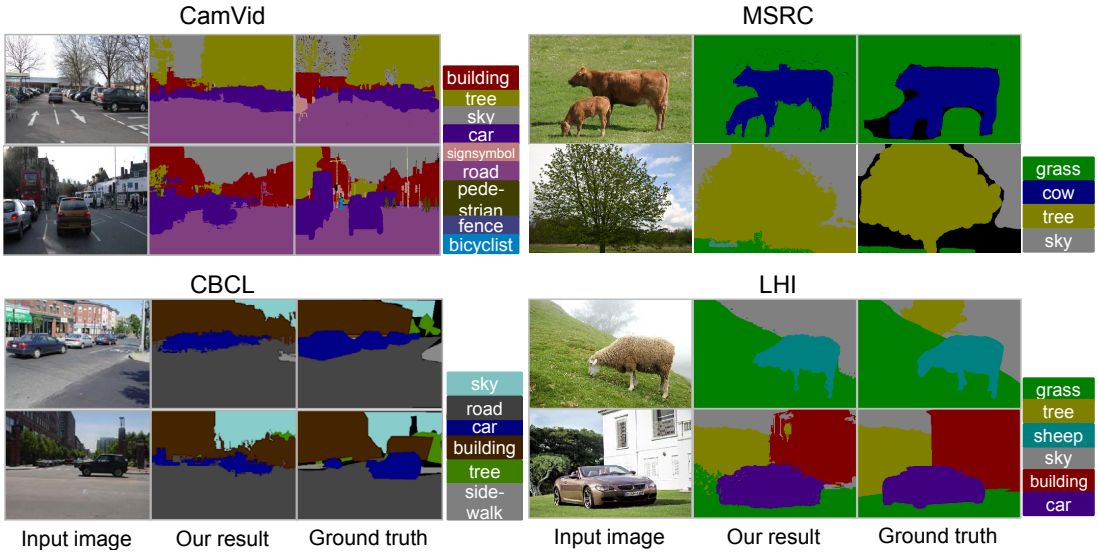

Fig. 12. Our GP results on the four datasets.

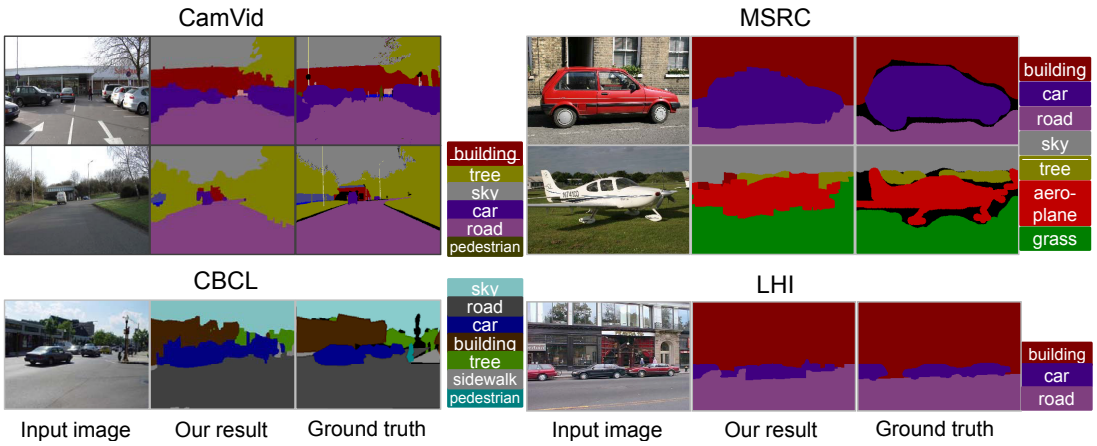

Fig. 13. Our SGP results on the four datasets.

Label accuracy is computed as percentage of image pixels assigned to the correct category label. Some classes such as grass, sky, road and book have high labeling accuracy, while some complex objects like body and some classes with limited training data such as bird, boat are not good enough. The comparison of labeling accuracy is listed in Table I. The global average accuracy on CamVid, MSRC, CBCL and LHI datasets are 82.37%, 73.41%, 75.31% and 80.4%. Our GP algorithm performs better than Shotton *et al.* [7] on MSRC dataset and better than Zhang *et al.* [29] and Shotton *et al.* [11] on CBCL dataset. The accuracy of Shotton *et al.* [11] on CBCL dataset is obtained from Zhang *et al.* [29].

In GP algorithm, the labeling is processed pixel by pixel, and the minimal distance of each single pixel only depends on its nearest neighbor along the geodesic path. The spatial constraint is relatively weak, therefore the algorithm prefers to label pixels which are similar in the feature space, not in the physical space. With this property, the algorithm does well in the interior region of object than graph cuts algorithm. A good probabilistic color model will give a good performance. Note that our GP result is weak at localization of object boundary.

TABLE I
COMPARISON OF SEMANTIC LABELING ACCURACY OVER FOUR DATASETS.

Method	CamVid	MSRC	CBCL	LHI
Zhang *et al.* [29]	84.4%	-	72.8%	-
Shotton *et al.* [7]	-	72.2%	-	-
Shotton *et al.* [11]	-	-	61.9%	-
Our GP	82.37%	73.41%	**75.31%**	80.4%
Our SGP	**87.76%**	**77.13%**	71.7%	**81.29%**

This is due to the lack of explicit spatial constraint.

C. SGP performance

We test the performance of our supervised geodesic propagation on the four datasets. In our experiments, the training procedure of Joint Boost model takes about 40 seconds per image and the training of propagation indicator for each image is also about 40 seconds. The propagation takes about 5 seconds. On each dataset, we set 500 rounds to train the Joint Boost model for each test image. Some results on the four datasets are shown in Fig. 13.

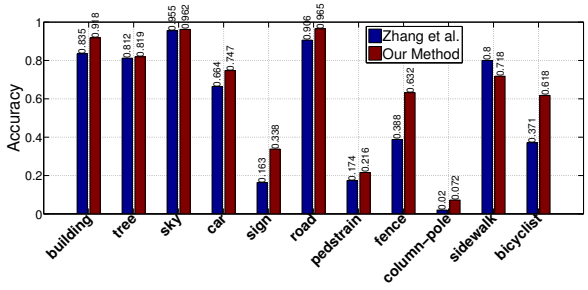

Fig. 14. Category accuracy comparison on CamVid dataset (SGP). The accuracy is located at the top of corresponding bar. Our algorithm is better than Zhang *et al.* [29] expect for the category of 'sidewalk'.

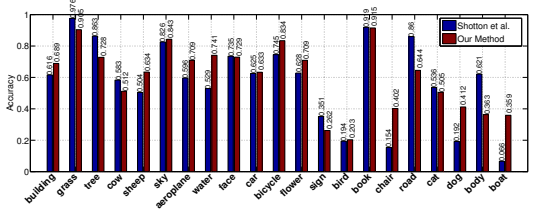

Fig. 15. Category accuracy comparison on MSRC dataset (SGP).

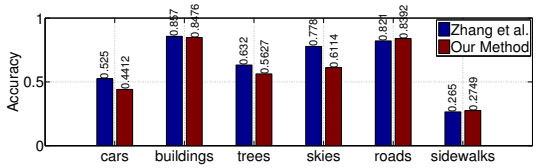

Fig. 16. Category accuracy comparison on CBCL dataset (SGP).

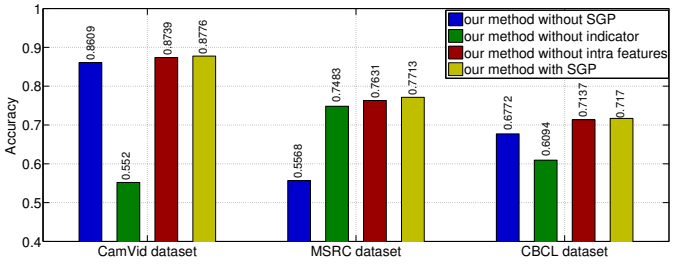

Fig. 17. The effects of different parts in our supervised geodesic propagation.

CamVid dataset. We use 5 similar images for each test image to train the propagation indicator. The threshold θ_e and ϕ are set to be 0.8 and 0.75, respectively. The segmentation accuracy of our SGP is 87.76% as shown in Table I.

The comparison of category accuracy with Zhang *et al.* [29] is shown in Fig. 14. Our algorithm obtains higher category accuracy than theirs except for the 'sidewalk' category. In this dataset, 'sidewalk' is similar with 'road' in appearance while our algorithm gets higher accuracy of 'road'. To address the problem, we may consider more details of appearance in future work.

MSRC dataset. We use 10 similar images for each test image to train the propagation indicator. The threshold θ_e and ϕ are set to be 0.5 and 0.4, respectively. The accuracy of our SGP algorithm on this dataset is 77.13%.

Our algorithm performs best on this dataset. The comparison of category accuracy with Shotton *et al.* [23] is shown in Fig. 15. Our algorithm performs better for major categories except for a few classes which are 'grass', 'tree', 'cow' ,'sign' and 'body'. The categories 'sign' and 'body' exhibit various appearance in color and texture, thus make the retrieved similar image set imprecise. The similar images of 'grass' consist of some sheep and cow images thus make the accuracy of 'grass' low.

CBCL dataset. We use 5 similar images for each test image to train the propagation indicator. The threshold θ_e and ϕ are set to be 0.3 and 0.6, respectively. The accuracy on this dataset is 71.7%. Although our SGP is not the best on this dataset (about 1 percent lower than Zhang *et al.* [29]), we perform better than Shotton *et al.* [11] which has accuracy of 61.9%. The comparison of category accuracy with Zhang *et al.* [29] is shown in Fig. 16. Our algorithm performs similar or slightly better than [29] on 'buildings', 'roads' and 'sidewalks'.

LHI dataset. We use 10 similar images for each test image. The threshold θ_e and ϕ are set to be 0.8 and 0.7, respectively. The accuracy on this dataset is 81.29%.

We test different parts, different parameter values and different features on three datasets(CamVid, MSRC, CBCL) to explore their importance and influence in the SGP algorithm.

Different parts. To analyse how our indicator and geodesic weight influence the performance, we test our method with 4 different configurations: *our method without SGP*, *our method without indicator*, *our method without intra features* and *our SGP*. *Our method without SGP* produces initial labeling results of boosted model without supervised propagation. This shows the influence of the Joint Boost model trained on the similar image set. *Our method without indicator* means the indicator is not involved in the propagation. In another word, the $T(vi, vj)$ is not considered into the *if* conditional judgement of Algorithm 2. This configuration shows the effect of our indicator. *Our method without intra features* indicates that $W(vi, vj)$ is not considered into the *if* conditional judgement of Algorithm 2. The performance of *our SGP* is demonstrated in Table I. Here we illustrate it again for clear comparison with other three configurations.

From Fig. 17, we can see that the configuration without indicator gets poor accuracy, thus proving that our indicator is useful and important. As *our method without indicator* involves only the weight of intra features and gets poor result, we test our method without intra feature and find that the influence of intra features is weaker than that of the indicator. However, comparing *our method without intra features* with *our SGP*, we can see that the intra features can slightly improve the final results. The configuration of *our SGP* gets the best performance.

Different parameters. In the configuration of *our SGP*, two thresholds are involved: θ_e and ϕ. θ_e implies the importance of geodesic distance weight while ϕ denotes the effect of our indicator. To analyse the effect of ϕ, we fix θ_e; Also, we fix ϕ when we analyse θ_e. In the configuration of *Our method without indicator*, we test different values of θ_e. Fig. 18 shows

TABLE II
ACCURACY COMPARISON OF DIFFERENT FEATURES

	CamVid	MSRC	CBCL
SGP with our features	87.76%	77.13%	71.7%
SGP with texture pattern features	87.81%	77.28%	71.98%

the effects of different values on CamVid dataset, MSRC dataset and CBCL dataset. In these three figures, the blue line, which denotes the initial labeling accuracy of boosted model without our supervised geodesic propagation, is shown for clearer comparison. The red line shows the accuracy of different ϕ with fixed θ_e while the green line is the accuracy of different θ_e with fixed ϕ. The magenta line denotes the accuracy of different θ_e without indicator. To avoid the noises of parameters, we get the values of θ_e and ϕ for each dataset, and set them these values when they are fixed. These values are: $\theta_e = 0.2$ and $\phi = 0.75$ on CamVid dataset, $\theta_e = 0.5$ and $\phi = 0.4$ on MSRC dataset, $\theta_e = 0.8$ and $\phi = 0.7$ on CBCL dataset.

In the subfigure of CamVid dataset, according to the red line, we can see that global accuracy increases significantly when the value of ϕ changes from 0.1 to 0.7. When ϕ is higher than 0.6, indicator plays an important positive role in the propagation (this part of red line is above the blue line). There is slightly accuracy variation when the value of θ_e changes as shown in the green line, proving that current features for the geodesic distance weight make little effect on this dataset. The magenta line shows that when the indicator is not taken into the propagation, the edge weight can not improve the accuracy compared with the initial accuracy.

In the subfigure of MSRC dataset, the indicator plays an important positive role in the propagation when the value of ϕ is lower than 0.7 (this part of red line is above the blue line). When the indicator with an appropriate ϕ is involved, as shown in green line, the final accuracy is always higher than the initial accuracy regardless of the value of θ_e (more than two percentages). Without indicator, the accuracy decreases with the increment of θ_e. According to the magenta line, the final accuracy is higher than the initial accuracy when θ_e is lower than 0.4.

In the subfigure of CBCL dataset, comparing the red line with the blue line, we can see that indicator improves the final accuracy when ϕ is higher than 0.5. The green line is above the blue line which shows that a appropriate ϕ can make good effect on the final accuracy. In the green line, the accuracy changes slightly when the value of θ_e changes. In the magenta line, the accuracy decreases apparently without indicator, proving that our indicator takes an important role in the propagation.

Different features. We test other features to figure out the scalability of our framework. We employ the 29 dimension texture features of [50] in the learning of the propagation indicator. Our original 22 dimension descriptor is expanded to 51 dimension. Compared with our features, the 51 dimension features can improve the final accuracy of the three datasets as shown in Table II. It shows that if the features are selected appropriately, our framework can get better performance.

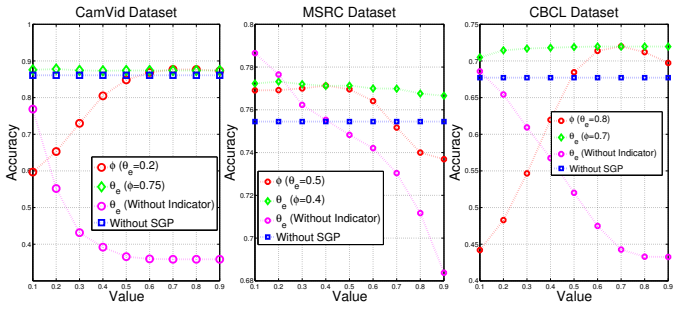

Fig. 18. The influence of different parameter values on CamVid, MSRC and CBCL datasets. The blue line denotes the initial label accuracy of boosted model. The red line shows the accuracy of different ϕ with fixed θ_e while the green line is the accuracy of different θ_e with fixed ϕ. The magenta line denotes the accuracy of different θ_e without indicator.

TABLE III
TIME COMPARISON BETWEEN GP AND SGP

Time complexity	CamVid	MSRC	CBCL	LHI
GP	1.26s	0.34s	0.33s	0.31s
SGP	8.29s	4.32s	2.71s	2.88s

Feature selection attracts attentions of many works, and is beyond the scope of this paper.

D. Comparison between GP and SGP

We compare our GP and SGP in terms of accuracy and time complexity. The accuracy comparison is shown in Table I. Our SGP performs the best on the CamVid, MSRC and LHI datasets, our GP performs the best on the CBCL dataset. With the supervision of contextual information, SGP algorithm performs better than GP algorithm. Compared to other datasets, the retrieved similar images in CBCL dataset have weaker similarity to the test image. Thus the indicators trained on the similar images are not so efficient as they are in other datasets. In other words, the contextual information is not well exploited. In this case, the GP gives an encouraging performance to propagate labels in feature space.

Since the GP algorithm does not need supervision in propagation, it performs faster than the SGP algorithm. We record the average consuming time to label an image in the four datasets. All the experiments are implemented on the same computers. Table III shows the consuming time comparison. According to this table, the GP algorithm performs much faster than the SGP algorithm.

According to the comparisons, users can choose appropriate algorithm in different cases. When users have a requirement of accuracy, they can resort to SGP algorithm. When they need a result in a limited time, they can resort to GP algorithm. It just depends on the situation.

E. Video geodesic propagation

We test our hybrid geodesic propagation on two video sequences taken from CamVid dataset: (1) Camseq01, 101 frames from seq16E5 15Hz sequence. (2) Camvid seq05, first 3000 frames from 0005VD sequence (original seq05VD sequence). Both sequences depict driving scenes and include

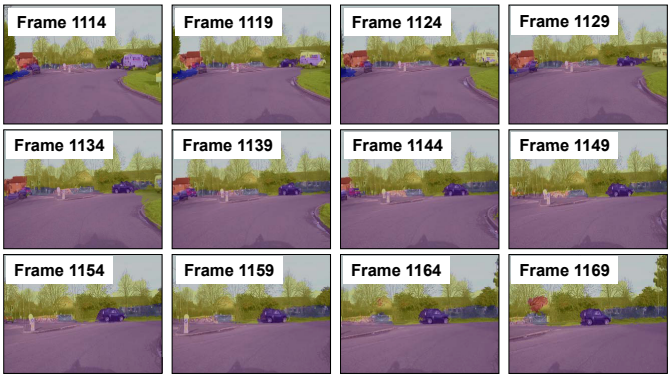

Fig. 19. The result of video propagation. These frames are from CamVid sequence Seq06R0. Semantic labels are overlaid on the image.

TABLE IV
VIDEO PROPAGATION COMPARISON

R_K frames	Camseq01				Camvid seq05		
	1	5	10	15	1	15	30
DP-MRF [55]	305	120	84	75	1017	342	201
Our method	243	112	91	88	618	284	257

32 classes. For comparison, we resize images to 398x530 and choose the top 10 frequently appeared classes from 32 classes for evaluation as [55] did. Each frame in Camseq01 is labeled manually while one frame in every 30 frames is labeled manually in Camvid seq05. We only use labeled frames for training and evaluation.

The quantitative comparison with [55] is shown in Table IV. The metric is the average number of incorrect pixels over all frames in hundreds of pixels (the standard metric in previous work [53]–[55]). We evaluate our algorithm with variant values of the similar images number R_K (number of labeled frames).

For sequence Camseq01, we use 1, 5, 10 and 15 similar images to train the recognition model. Our algorithm performs better than [55] with 1 and 5 similar images. Since this sequence is obtained by moving straight along the street, the R_K images have high similarity. Thus for computational efficiency, we select two frames for value 5, four frames for values 10 and 15 to train the indicator.

For sequence seq05, our algorithm performs much better than [55] with 1 and 15 similar images. In the values of 15, 30, 45 and 60, the recognition model is trained on the corresponding number of similar images. However, due to the limited physical memory, the training of indicator is based on ten similar images. With the increase of R_K value, [55] obtains decreasing average errors obviously while ours are stable. The average error of value 45 and 60 are listed in supplementary materials. Fig. 19 demonstrates our video propagation results of other sequence. For more video results, please refer to the supplementary materials.

VIII. CONCLUSION AND DISCUSSION

In this paper we propose a semantic labeling framework with geodesic propagation. Under the same framework, three algorithms are proposed, i.e. GP, SGP and HGP. GP and SGP are provided for image semantic labeling, and HGP is extended for video labeling. We generate appropriate propagation seeds based on recognition proposal map. Confident pixels with maximum probability are selected as the initial seeds. From these seeds, the GP algorithm iteratively updates the weights of geodesic distances until the semantic labels are propagated to all pixels. SGP exploits more contextual information than GP to guide the propagation direction. In HGP algorithm, the geodesic metric is used in both spatial and temporal spaces. Experiments on four public datasets show that our algorithms outperform the traditional learning-based methods and the previous label transfer methods. Each part in our framework, such as the recognition model, the color model, the features or the boost algorithms can be replaced by other promising ones. Our contribution focuses on the geodesic propagation framework.

Limitations and future work. As we suppose the similar image set covers all categories in the input image, our algorithm is sensitive to the retrieved similar images. If the retrieved images have little similarity with the test image, or the retrieved images do not have the category in testing stage at all, our algorithm will fail. In the future, we will pay attention on how to retrieve similar image set of high quality. Meanwhile, we will test our framework on more public databases. Since the feature design is not the focus of our work, we feed a set of simple features into the geodesic propagation framework. In the future, we will consider more appropriate visual features as well as other methods for generating initial seeds (e.g., [61]).

Note that we do not embed smoothness term in our propagation explicitly, thus gives rise to some un-smoothed segmentation results. We could use some optimization algorithm, such as graph-cut, to smooth the results. However, a better solution, which we are attempting to do, would be to embed the smooth term into the geodesic propagation more efficiently.

In terms of video labeling, HGP performs in spatial and temporal space separately. A better solution may be a simultaneous propagation in the whole spatial-temporal space instead of the inconsistent regions. Thus the MRF optimization for HGP can be replaced by simultaneous propagation.

ACKNOWLEDGMENT

In addition to the anonymous reviewers who provided insightful suggestions, the authors would like to thank Dongyue Zhao, Changqun Xia, Yi Liu, Liang Lin, Yibiao Zhao and Jia Li for their invaluable help. This work was partially supported by NSFC (61325011), 863 Program (2012AA011504), ITER (2012GB102008) and SRFDP (20131102130002).

REFERENCES

[1] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation," in *Proc. CVPR*, 2010.
[2] J. Carreira, F. Li, and C. Sminchisescu, "Object recognition by sequential figure-ground ranking," *Int. J. Comput. Vis.*, vol. 98, no. 3, pp. 243–262, 2012.
[3] B. L. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," in *Proc. CVPR, 2010*.
[4] J. Winn and N. Jojic, "Locus: learning object classes with unsupervised segmentation," in *Proc. ICCV, 2005*.
[5] D. Pinar, B. Kobus, de Freitas J. F. G., and F. D. A, "Object recognition as machine translation: learning a lexicon for a fixed image vocabulary," in *Proc. ECCV, 2002*.

[6] M. P. Kumar, P. Torr, and A. Zisserman, "Obj cut," in *Proc. CVPR, 2005*.

[7] 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," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, 2009.

[8] Z. Tu, "Auto-context and its application to high-level vision tasks," in *Proc. CVPR, 2008*.

[9] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. ICCV, 2009*.

[10] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proc. CVPR, 2004*.

[11] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. CVPR, 2008*.

[12] L. Lin, T. Wu, J. Porway, and Z. Xu, "A stochastic graph grammar for compositional object representation and recognition," *Pattern Recognit.*, vol. 42, no. 7, pp. 1297–1307, 2009.

[13] L. Lin, X. Liu, S. Peng, H. Chao, Y. Wang, and B. Jiang, "Object categorization with sketch representation and generalized samples," *Pattern Recognit.*, vol. 45, no. 10, pp. 3648–3660, 2012.

[14] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. CVPR*, 2008.

[15] Z. Tu, "Probabilistic boosting tree: learning discriminative models for classification, recognition, and clustering," in *Proc. ICCV*, 2005.

[16] R. Fergus, D. P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. CVPR*, 2003.

[17] L. Lin, Y. Lu, Y. Pan, and X. Chen, "Integrating graph partitioning and matching for trajectory analysis in video surveillance," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4844–4857, 2012.

[18] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *Proc. ICCV*, 2007.

[19] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[20] X. Ren, C. C. Fowlkes, and J. Malik, "Figure/ground assignment in natural images," in *Proc. ECCV, 2006*.

[21] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, 2004.

[22] Y. Tian, J. Li, S. Yu, and T. Huang, "Learning complementary saliency priors for foreground object segmentation in complex scenes," *Int. J. Comput. Vis.*, 2014.

[23] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: joint appearance, shape and context modeling for mulit-class object recognition and segmentation," in *Proc.ECCV*, 2006.

[24] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, "Associative hierarchical crfs for object class image segmentation," in *Proc. ICCV, 2009*.

[25] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, no. 1-3, pp. 157–173, 2008.

[26] J. Li, Y. Tian, and T. Huang, "Visual saliency with statistical priors," *Int. J. Comput. Vis.*, vol. 107, no. 3, pp. 239–253, 2014.

[27] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: label transfer via dense scene alignment," in *Proc.CVPR*, 2009.

[28] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," in *Proc.ECCV, 2010*.

[29] H. Zhang, J. Xiao, and L. Quan, "Supervised label transfer for semantic segmentation of street scenes," in *Proc.ECCV, 2010*.

[30] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: a large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 1958–1970, 2008.

[31] A. Oliva and A. Torralba, "Building the gist of a scene: the role of global image features in recognition," *Prog. Brain Res.*, vol. 155, pp. 23–36, 2006.

[32] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *Proc. CVPR*, 2010.

[33] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 113–132, 2009.

[34] D. Wang, C. Yan, S. Shan, and X. Chen, "Active learning for interactive segmentation with expected confidence change," in *Proc. ACCV*, 2012, pp. 790–802.

[35] Z. Liu, D. Hu, H. Shen, and G. Feng, "Graph-based image segmentation using directional nearest neighbor graph," *SCIENCE CHINA Information Sciences.*, vol. 56, no. 11, pp. 1–10, 2013.

[36] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *Proc. ICCV, 2007*.

[37] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. J. Comput. Vis.*, vol. 80, no. 3, pp. 300–316, 2008.

[38] J. J. Lim, P. Arbelaez, C. Gu, and J. Malik, "Context by region ancestry," in *Proc. ICCV, 2009*.

[39] X. Liu, J. Feng, S. Yan, L. Lin, and H. Jin, "Segment an image by looking into an image corpus," in *Proc. CVPR, 2011*.

[40] J. Xiao and L. Quan, "Multiple view semantic segmentation for street view images," in *Proc. ICCV, 2009*.

[41] J. Xiao, T. Fang, P. Zhao, M. Lhuillier, and L. Quan, "Image-based street-side city modeling," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–12, 2009.

[42] X. Chen, D. Zhao, Y. Zhao, and L. Lin, "Accurate semantic image labeling by fast geodesic propagation," in *Proc. ICIP, 2009*.

[43] S. J. A., *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics*. Cambridge Univ. Press, 1996.

[44] L. Yatziv, A. Bartesaghi, and G. Sapiro, "On implementation of the fast marching algorithm," *J. Comput. Phys.*, vol. 212, pp. 393–399, 2005.

[45] X. Chen, Q. Li, Y. Song, X. Jin, and Q. Zhao, "Supervised geodesic propagation for semantic label transfer," in *Proc. ECCV, 2012*.

[46] X. Chen, H. Wu, X. Jin, and Q. Zhao, "Face illumination manipulation using a single reference image by adaptive layer decomposition," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4249–4259, 2013.

[47] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[48] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, pp. 18–22, 2002.

[49] A. Jaiantilal, "Classification and regression by randomforest-matlab," *http://code.google.com/p/randomforest-matlab*, 2009.

[50] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. ICCV*, 2005.

[51] X. Di, H. Chang, and X. Chen, "Multi-layer spectral clustering for video segmentation," in *Proc. ACCV*, 2012.

[52] Q. Zhu, Z. Song, Y. Xie, and L. Wang, "A novel recursive bayesian learning-based method for the efficient and accurate segmentation of video with dynamic background," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3865–3876, 2012.

[53] V. Badrinarayanan, F. Galasso, and R. Cipolla, "Label propagation in video sequences," in *Proc. CVPR*, 2010.

[54] I. Budvytis, V. Badrinarayanan, and R. Cipolla, "Label propagation in complex video sequences using semi-supervised learning," in *Proc. BMVC*, 2010, pp. 1–12.

[55] S. Vijayanarasimhan and K. Grauman, "Active frame selection for label propagation in videos," in *Proc. ECCV*, 2012.

[56] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[57] M. Schmidt, "Ugm: Matlab code for undirected graphical models," http://www.di.ens.fr/~mschmidt/Software/UGM.html, 2013.

[58] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: a high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.

[59] S. Bileschi, "Cbcl streetscenes challenge framework," *http://cbcl.mit.edu/software-datasets/streetscenes/*, 2007.

[60] B. Yao, X. Yang, and S. C. Zhu, "Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks," in *EMMCVPR*, 2007, pp. 169–183.

[61] J. Li, D. Xu, and W. Gao, "Removing label ambiguity in learning-based visual saliency estimation," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1513–1525, 2012.

Qing Li is presently a Ph.D. candidate in the State Key Laboratory of Virtual Reality Technology and Systems, the School of Computer Science and Engineering at Beihang University. Her research interests are computer vision and visual computing.

Xiaowu Chen received the Ph.D. degree at Beihang University in 2001. He is a professor in the State Key Laboratory of Virtual Reality Technology and Systems, also in the School of Computer Science and Engineering, Beihang University. His research interests include computer vision, computer graphics, virtual reality and augmented reality.

Yafei Song is presently a Ph.D. candidate in the State Key Laboratory of Virtual Reality Technology and Systems, also in the School of Computer Science and Engineering at Beihang University. His research interests are computer vision and image processing.

Yu Zhang is presently a Ph.D. candidate in the State Key Laboratory of Virtual Reality Technology and Systems, also in the School of Computer Science and Engineering at Beihang University. His research interests are computer vision and pattern recognition.

Xin Jin is presently a Ph.D. candidate in the State Key Laboratory of Virtual Reality Technology and Systems, also in the School of Computer Science and Engineering at Beihang University. His research interests are visual computing and virtual reality.

Qinping Zhao is a professor in the School of Computer Science and Engineering, Beihang University. He is the founder and director of the State Key Laboratory of Virtual Reality Technology and Systems. His research interests include virtual reality and artificial intelligence.