Pattern Recognition 62 (2017) 189–201

Contents lists available at ScienceDirect

Pattern Recognition

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

Semantic annotation for complex video street views based on 2D–3D multi-feature fusion and aggregated boosting decision forests

Xun Wang [a], Guoli Yan [a], Huiyan Wang [a,*], Jianhai Fu [a], Jing Hua [b], Jingqi Wang [c], Yutao Yang [a], Guofeng Zhang [d], Hujun Bao [d]

[a] *School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[b] *Department of Computer Science, Wayne State University, Detroit, MI 48202, United States*
[c] *School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*
[d] *The State Key Lab of CAD and CG, College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China*

ARTICLE INFO

Article history:
Received 1 December 2015
Received in revised form
8 July 2016
Accepted 24 August 2016
Available online 30 August 2016

Keywords:
Semantic annotation
Superpixel segmentation
2D–3D feature fusion
ABDF model

ABSTRACT

Accurate and efficient semantic annotation is an important but difficult step in large-scale video interpretation. This paper presents a novel framework based on 2D–3D multi-feature fusion and aggregated boosting decision forest (ABDF) for semantic annotation of video street views. We first integrate the 3D and 2D features to define the appearance model for characterizing the different types of superpixels and the similarities between two adjacent superpixel blocks. We then propose the ABDF algorithm to build the week classifier by using a modified integrated splitting strategy for decision trees. And a Markov random field is then adopted to perform global superpixel block optimization to correct the minor errors and make the boundary for semantic annotation smoother. Finally, a boosting strategy is used to aggregate the different week decision trees into one final strong classification decision tree. The superpixel block instead of the pixel is used as the basic processing unit, thus only a small amount of features are required to build an accurate and efficient model. The experimental results demonstrate the advantages of the proposed method in terms of classification accuracy and computation efficiency over those of existing semantic segmentation methods. The proposed framework can be used in real-time video processing applications.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Image semantic annotation [1] is a challenging problem in applications such as semantic image retrieval, information visualization and web resource reasoning. In the domain of computer vision, unambiguous semantic knowledge representation for images is essential to eliminate the semantic gap. For image and video retrieval based on the semantic annotation, correct and reasonable semantic annotation is an important step of the entire process, and its accuracy directly affects the ultimate retrieval results. The image information is associated with the semantics by using semantic annotation, thus the video image retrieval can be achieved by retrieving semantics. The semantic annotation of a video image is generally based on the low-level features of the image and the image semantics is acquired through knowledge reasoning. For big video data, exploring automatic and real-time semantic annotation methods becomes an important but difficult task. For the real-time automatic semantic annotation, the semantics are acquired with less manual interference and the effective management on massive video images is also needed. Therefore, using the images with annotations as the input, the inference system could determine the candidate semantic annotations by calculation or reasoning according to the basic features of the image, such as color, texture and shape. This is also known as semantic annotation or semantic segmentation [2,3] in computer vision.

Current automatic annotation methods can be roughly divided into two categories: learning-based methods and search-based methods. The learning-based automatic image annotation methods usually build a statistical model for the joint distribution of the components based on the image annotations and its visual features, which can be further divided into supervised-learning-based and unsupervised-learning-based methods [4]. For

[*] Corresponding author.
E-mail addresses: wx@zjgsu.edu.cn (X. Wang), 1327237426@qq.com (G. Yan), cederic@zjgsu.edu.cn (H. Wang), 15138469880@163.com (J. Fu), jinghua@wayne.edu (J. Hua), wangjingqi555@163.com (J. Wang), 1121082097@163.com (Y. Yang), zhangguofeng@cad.zju.edu.cn (G. Zhang), bao@cad.zju.edu.cn (H. Bao).

http://dx.doi.org/10.1016/j.patcog.2016.08.030
0031-3203/© 2016 Elsevier Ltd. All rights reserved.

supervised semantic annotation methods, the computer system could improve the performance of the classifier by using the category information of samples, and, after training, it can recognize a certain category of images and distinguish the different objects in the image to annotate new video images more reasonably, which reduces the workload and improves the classification accuracy. For unsupervised methods, semantic annotation is performed by only using unlabeled samples, namely, without any priori knowledge of the objects to be annotated. Generally, the accuracy of unsupervised methods is inferior to supervised learning methods.

In this paper, a supervised-learning-based method of semantic annotation is adopted. A semantic segmentation framework based on 2D–3D multi-feature fusion and aggregated boosting decision forest (ABDF) is proposed to automatically annotate video street views. The 2D visual features and 3D geometric features of image sequences are extracted and combined with the information of the depth map to form a hybrid multi-type feature vector, which can effectively avoid the semantic ambiguity. We choose Random Forest (RF) [5–7] as the basic classifier in view of its wide applications and good performance of in semantic segmentation [8], object recognition [9] and data packets [10]. The ABDF uses a modified splitting strategy to improve the classification accuracy of RF. It constructs Markov random field for each decision tree and then performs global superpixel block optimization to make the boundary of semantic annotation smoother. Finally, a boosting strategy is used to aggregate the different week classifiers into one final strong classification model to improve the overall performance.

2. Related work

In recent years, a lot of effort have been spent on supervised-learning-based [11] annotation methods. Chang et al. [12] proposed a content-based soft annotation method (CBSA), which trains the Bayes classifier with a manually annotated image set. Multiple classifiers categorize the images and set reliability coefficients to various tags regarding each image, and each coefficient represents the possibility of the specified tag. Reliability coefficients are determined by the classification results, and then, through comparing the reliability coefficient of each tag, a final decision is made to assign a certain tag to annotate the image. Zhang et al. [13] proposed an annotation method using group sparsity, which mainly selects the low-level features of the training set through sparsity and clustering under prior knowledge. The classification of positive and negative training sets is obtained through iteration, and the annotation for similar images is transmitted to realize semantic annotation on the test set. Carneiro et al. [14] proposed a supervised multiclass labeling method (SML), which calculates the similarity between the image and the category as the reference of new annotations and clusters the image's Gauss mixture model into a concept class. The main advantage of the method is that annotation will automatically generate the sorting information and its lexicon is scalable, and the shortcoming is that the computation consumes too much time. Socher et al. [15] put forward a semi-supervised method, which segments the image, extracts the low-level visual features such as color and texture, generates the visual vocabulary tree using the *K*-means clustering method and associates the text label with the image to generate the lexicon by a small amount of annotated images. This method maps the text and visual vocabulary to low-dimensional spatial features through the typical Kernel Canonical Correlation Analysis (KCCA) method. For semi-supervised learning, only a small amount of annotated samples is required, but image segmentation increases the time complexity of the algorithm and results in slower processing speed.

According to the ways of image feature extraction, semantic annotation can be roughly divided into two types: supervised learning methods based on the 2D features [16] and supervised learning methods based on 3D geometric features [17]. Conventional 2D features refer to the color, texture, shape and so on. However, the impacts due to the time, weather and illumination conditions would be a great disturbance to the prediction accuracy of supervised learning. Therefore, to reduce the influence of these factors, object motion and structure feature should be considered, which is particularly important in the processing of video sequences. 3D geometric features refer to the features of the information regarding the objects' spatial position, such as the cameras' relative height to the 3D point, the smallest distance between the 3D point and the camera trajectory, the angle between a 3D vector and 2D plane, etc.

Liu et al. [18] used a non-parametric method to process scene analysis and 2D feature information of the image selected. Joseph andSvetlana. [19] presented a non-parametric Super-Parsing semantic segmentation method (SP-SS) based on lazy learning. This method used scene-level matching, superpixel-level matching and Markov Random Field (MRF) optimization, which outperformed the state-of-the-art non-parametric method based on SIFT. Heesoo and Kyoung [20] proposed a non-parameter approach for semantic segmentation using high-order semantic relations transfer method (SRT-SS). The high-order semantic relations were transferred from annotated images to unlabeled images. Browtow et al. [21] proposed an ego-motion-based 3D point cloud to predict the video sequence annotation, which requires no descriptors and can execute sound semantic annotation for sparse and noisy point clouds. Zhang et al. [17] proposed using a dense depth map to identify and annotate objects, which extracts 3D geometric features for training and constructs a MRF to optimize the control. Xiao and Quan [22] proposed a simple but powerful multiple-view semantic segmentation framework, which takes the 2D features and 3D features into consideration simultaneously and is applicable for large-scale scene semantic annotation. In this paper, both the 2D features and 3D features are used, and the Markov Random Field (MRF) is adopted to perform global superpixel block optimization to improve the semantic segmentation results.

From the learning mechanisms of classifiers, semantic annotation can be roughly divided into two categories: traditional classifiers and new classifiers based on a deep learning approach. The traditional classifier includes the Bayes classifier, decision tree classifier (DF), support vector machine (SVM) classifier, neural network classifier and so on. The deep learning-based classifiers do not need to select features manually and automatically extract the features that can be comprehended for later learning and recognition while replacing the color histogram, texture features, SIFT features, HOG, etc.

Automatic semantic annotation is essentially a problem of classification. RF has been widely used in solving non-parametric and highly non-linear structured problems and has achieved good results in classification and logistic regression problems. Schulter et al. [23] proposed a boosting RF classifier, which converts the training of RF into the process of minimizing a global energy function. The training process obtains the minimized energy function through adaptively adjusting the sample weights. Schwing et al. [24] proposed a confidence interval-based statistical mechanism and a binomial conjugate relationships-based adaptive RF method. In this paper, according to the features of street view segmentation, RF is adopted as the most fundamental component of a classifier.

In the last two years, deep learning (DL) has led to a breakthrough in many visual applications. Brust et al. [25] presented convolutional patch networks for semantic segmentation and road detection and achieved state-of-the-art result. Ross et al. [26]

proposed a simple and scalable algorithm by combining region proposals with convolutional neural networks (CNN) for accurate object detection and semantic segmentation. Saurabh et al. [27] proposed deep classification nets for semantic segmentation based on depth CNN features and RGB CNN features. Noh et al. [28] proposed a semantic segmentation algorithm by learning a deconvolution network. Long et al. [29] built a fully convolutional network (FCN) to be trained end-to-end and pixels-to-pixels. The proposed FCN outperformed the state-of-the-art methods in semantic segmentation. These DL-based approaches have three disadvantages. First, they cost more than several hours or several days to train or fine-tune a network and spend much more online time than the traditional methods. Second, the massive number of training samples is required to construct a robust model in these methods. Third, the high-end graphics cards are needed, such as GTX Titan X GPU, which put high demanding requirements on hardware. To show the effectiveness of the proposed method, we compared it with several state-of-the-art DL-based methods.

Unsupervised learning-based automatic annotation is also widely studied. Lu et al. [30] proposed a context-based multi-label annotation, which mainly uses the context to transfer keywords and also simultaneously transmits several keywords to the test image. The effect is good, but it is time-consuming. Jamieson et al. [31] proposed a method for learning the appearance of target models based on the visual mode and language tips, which combines the typical appearance of target models with the corresponding names into a name mark and annotates the similar objects of the test images.

For searching-based automatic annotation methods, the main idea is to mine the related semantic description of similar images. This type of method requires no training sample set and is not restricted by the predefined vocabulary, so the process is simple and generally contains only two steps, namely searching and mining. Research studies of this category of methods are relatively few in number and implementations are less used in applications. Wang et al. [32] proposed a model-free-based image annotation method, which mines the search results with visual and semantic similarity to realize the ultimate image semantic annotation. It has good robustness to exception handling. In this paper, the main contribution is two-fold. First, the 2D and 3D features are extracted based on superpixels and aggregated to a representative feature vector with a high discriminative power and the feature aggregation can improve the robustness of the appearance model and outperform each individual one. Second, we propose a novel aggregated boosting decision forest to build the classifier, we call it ABDF algorithm, in which an aggregated splitting strategy is used and the breadth-first strategy is adopted instead of the depth-first strategy. To obtain more accurate segmentation results, GraphCuts are then adopted to tune and correct some minor errors. The proposed methodology achieves better performance in segmentation accuracy, robustness and comparable computation efficiency than existing state-of-the-art semantic annotation methods.

3. Architecture of the proposed method

The architecture of the proposed method is illustrated in Fig. 1. The main steps are summarized as follows:

Step 1: Segment the superpixels. SLIC is used to segment the superpixels of the training video sequence.

Step 2: The camera motion and 3D scene structure are recovered by using the automatic tracking system and then the depth maps are recovered based on a bundle optimization framework [33]. The 2D and 3D features of the superpixels are then extracted. Although the segmented superpixels have different sizes, their features have the same dimension. The 2D and 3D Features are normalized and fused based on a continuous feature fusion strategy.

Step 3: Semantic annotation. The superpixels are classified according to ABDF modeling, which is described in Algorithm 1. We train 100 week classifiers in parallel in our model.

Algorithm 1. ABDF algorithm.

```
Input: Training dataset $F_1 = \{(X_i, Y_i)\}$, the feature set $X_i$, the class label $Y_i$, $Y_i \in \{1, \dots, J\}$, $i = 1, 2, \dots, N_t$, the maximum tree depth $N_m$, the number of trees $N_r$
Input: Feature set $X_k$ of the testing dataset $F_2$, $k = 1, 2, \dots, N_k$
Output: Class label set $Y_k$ of $F_2$
Initialize the root nodes and the weight $w_i^1 = 1/N_t$;
for $i=1$ to $N_m$ do
    Check stopping criteria for all nodes in depth $i$
    for all $j=1$ to $N_r$ do in parallel
        Route the sample sets $S_1$ and $S_2$ to the left and right child nodes according to Eq. (8).
        Compute the local score $A(S1)$ and $A(S_2)$ according to Eq. (10).
        Compute the probability distribution $p(j|F_1)$ according to Eq. (14);
        Compute $\eta_1(F_1)$, $\eta_2(F_1)$, $\eta_3(F_1)$ according to Eqs. (11), (12), (13).
        Determine the best splitting function according to Eq. (9) and learn a week classifier $\varphi_i(X; D^i)$.
    end for
    Update the model $\Gamma$ according to Eq. (16).
    Update the weight $w_i^{i+1}$ according to Eq. (15).
end for
$Y_k = \text{argmax}_{Y_k^* \in \{1, \dots, J\}} \frac{1}{N_r} p(Y_k^*|X_k)$, $p(Y_k^*|X_k)$ is the class probability distribution returned by $\Gamma$.
Set $m=2$, update $Y_k$ according to Eq. (17).
```

4. Appearance feature model construction based on 2D–3D multi-feature fusion

Features are used to describe the most important attributes of the image. For image segmentation and recognition, using only 2D or 3D features to annotate target objects would result in semantic ambiguity. Considering that street view images usually contain complex objects, objects may partially occluded by each other. To address this problem, we construct the appearance model based on the combination of 2D and 3D features and depth information. The proposed appearance model is built based on superpixels [34]. For each superpixel, clues about object motion, color and texture features are used to extract the 3D and 2D features. Suppose the appearance model is presented as

$$\boldsymbol{A} = (\boldsymbol{T}, \boldsymbol{L}), \tag{1}$$

where $\boldsymbol{T}$ denotes the 3D feature vector, and $\boldsymbol{L}$ denotes the 2D feature vector, $\boldsymbol{A}$ denotes the feature vector after the concatenation of these two sub-vectors.

To extract 3D features, we use our automatic tracking system to recover camera motion as well as 3D scene structure from videos or image sequences [33]. For a given video sequence, we first use the SFM method to recover the camera parameters. Then, the disparity map for each frame is initialized independently. After initialization, bundle optimization is performed to refine the disparity maps.

4.1. *Extraction of superpixels*

After recovering dense maps, we extract the features based on superpixels. Superpixel techniques are one of oversegmentation methods. The major advantage of segmentation using superpixel is computational efficiency. A superpixel is usually defined as a perceptually uniform region in the image and a superpixel representation greatly reduce the dimension of image features compared to pixel representation. We adopt the linear iterative clustering (SLIC) [35,36] image superpixel segmentation method. The SLIC method is performed by using CIELAB color space and the 2D position information. It uses a new distance measure and controls the number of superpixels through parameter adjustment. By experiments, we found that SLIC achieves good performance in terms of computational complexity and the control over the size and number of superpixels. In this paper, the experimental images have resolutions of 960×540 and 690×720, and the number of superpixels is set to 1000 and 1500.

4.2. *Extraction of 3D features*

For the superpixel p, we extract five motion and structure features to form a 3D feature vector $\boldsymbol{T}^{\mathbf{p}} = \left\{ T_h^p, T_d^p, T_n^p, T_r^p, T_g^p \right\}$, where $T_h{}^p$ is the features of relative height to the camera, $T_d{}^p$ is the nearest distance of the camera, $T_n{}^p$ is the surface normal vector, $T_r{}^p$ is the reprojection error and $T_g{}^p$ is the relative height to the ground. $\boldsymbol{T}^{\mathbf{p}}$ is robust to the changes of appearance, efficient to compute, intuitive, and general but object-category covariant.

4.2.1. *The relative height to the camera*

For the image sequence of a motion scene, the relative height of the camera to the road remains constant in the real world. And it is the only rather fixed relationship between the 3D coordinate frames of the camera and the world, thus can be selected as a good feature for classification. Because the height parameter is susceptible to the road conditions in the plane coordinate system, 3D coordinate positioning is used. Suppose the direction of the y-axis is upward, then the relative height of a pixel m to the camera is defined as

$$t_m^h = y_m - y_c, \tag{2}$$

where y_c and y_m are the y-axis coordinates of the camera and the point m in 3D coordinate system. Therefore, the relative height of the superpixel p to the camera c is defined as

$$T_h^p = \frac{1}{K} \sum_{m=1}^{K} t_m^h, \tag{3}$$

where K is the number of pixels in a superpixel block and $T_h{}^p$ is the mean height of all pixels in p to the camera c.

4.2.2. *The nearest distance of the camera*

The distance to camera path can be used to separate objects which are horizontally distanced from the camera [17]. To compute the relative distance between one target and the camera in the real world, the camera center can be used. Let $\tilde{p}$ denotes the mean coordinate of the superpixel p in 3D coordinate system, $c(t)$ is the 3D coordinate of the camera center of the tth video frame. We define the nearest distance between the superpixel p and the camera c as the minimum value of the distance between $\tilde{p}$ and c (t), which provides a more accurate calculation of the distance. It is defined as

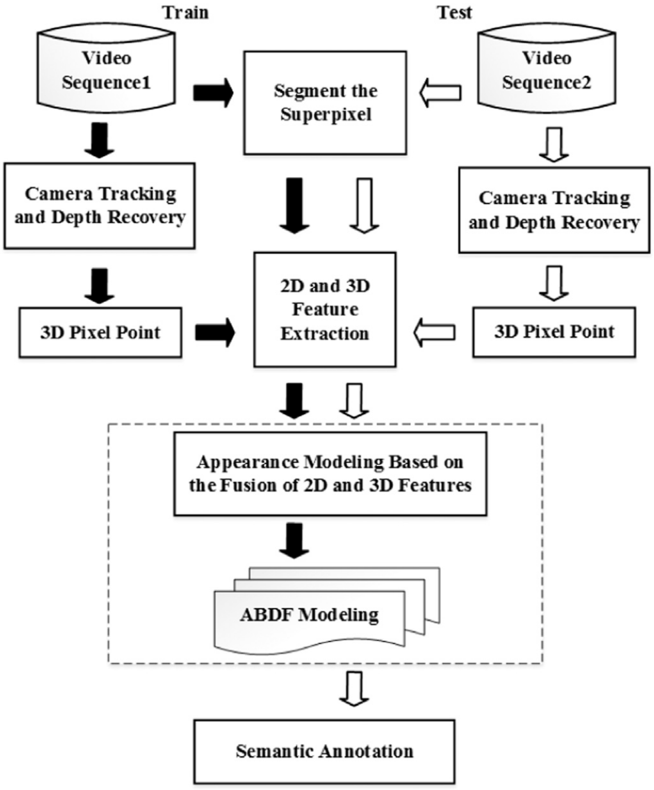

Train
Test
Video Sequence1
Segment the Superpixel
Video Sequence2
Camera Tracking and Depth Recovery
Camera Tracking and Depth Recovery
2D and 3D Feature Extraction
3D Pixel Point
3D Pixel Point
Appearance Modeling Based on the Fusion of 2D and 3D Features
ABDF Modeling
Semantic Annotation

Fig. 1. The flow chart of our method.

$$T_d^p = \min_t \| \tilde{p} - c\left(t\right) \| . \tag{4}$$

4.2.3. *Surface normal vector*

The surface normal vector $T_n{}^p$ of the superpixel p is computed by fitting a least square plane s_p to the set of 3D points in p. Then the vector normal $\mathrm{T}_n{}^p$ of s_p is calculated by a symmetric 3×3 positive semidefinite matrix $\sum_{c\in\tau_p}(c - m_p) \otimes (c - m_p)$, where τ_p denote all tracks that have projections in p, m_p denote the medians of three components of all 3D points in τ_p [22]. The eigenvectors, v_1, v_2 and v_3, correspond to the eigenvalues, $\lambda_1 \geq \lambda_2 \geq \lambda_3$, respectively. $T_n{}^p$ is chosen to be v_3 or $-v_3$. The sign is decided by having a greater than 180° angle between the camera orientation and $T_n{}^p$.

4.2.4. *Reprojection error*

In general, if an object moves fast, e.g., a moving vehicle or person, it will produce a sparser feature trajectory when compared with static objects. For each tracked point (u_i, v_i), its corresponding 3D position $\mathbf{P}_i$ is calculated, and then the 3D point is reprojected back to 2D space. The reprojection error $e(\mathbf{P})$ is calculated to test the accuracy of the proposed hypothesis. To prevent apparent corners and tracking errors on remote objects from dominating the residuals caused by real moving objects [21], a logarithmic scaling is used. The feature $T_r{}^p$ can be defined as

$$T_r^p = \log(1 + e(\mathbf{P})). \tag{5}$$

4.2.5. *The relative height to the ground*

Height information of an object is usually considered as fixed because its relative height is invariant, and this can be used as a feature for classification. Here, the sum of the heights of all pixels in the superpixel p to the ground plane is defined as the feature

$$T_g^p = \sum_{m=1}^{K} t_g^m, \tag{6}$$

where ${t_g}^m$ represents the distance between the pixel m and the ground and K is the size of superpixel blocks.

4.3. Extraction of 2D features

The 2D feature vector $\boldsymbol{L} = \left\{ L_H^p, L_T^p \right\}$ includes the color histogram features ${L_H}^p$ and Filter–Banks texture features ${L_T}^p$.

4.3.1. Color histogram

Color histogram is an effective and common global feature due to being robust to the changes in views and poses. The superpixel color histogram is also one of important features for classification based on superpixel [37]. Color histogram can be constructed from various color space such as *RGB*, *HSV*, and *LAB*. Because *HSV* space is closer to the human subjective understanding of color, the color histogram in the HSV space is extracted in our work, which is denoted as H_p. The dimension of H_p is 64. In addition, considering that many superpixels have nearly uniform colors, thus the mean of the RGB color space over the superpixel [22] is also used, which is denoted as R_p. Therefore, the color feature ${L_H}^p$ is defined as

$$L_H^p = \{H_p, R_p\}. \tag{7}$$

4.3.2. Texture

The texture feature is also one of the important features for image segmentation and recognition. Because physical surfaces have different features, the difference of the brightness and color of the image is extracted as texture features ${L_T}^p$. In this paper, the hybrid filter group is composed by three Gaussian filters, four Laplacian of Gaussian (LoG) filters and four first-order Gaussian filters. The kernel widths of three Gaussian filters are set to 1, 2 and 4 respectively, which are used for every channel of the CIELab with nine outputs. The kernel width of four Laplacian of Gaussian filters is set to 1, 2, 4 and 8, respectively. They are used only for the L channel to produce four filter outputs. Four first-order Gaussian are derived separately in the X direction and the Y direction with two and four different scales, and the Gaussian derivatives also affect only the L channel and generate four outputs. Thus, the mean value of all pixels in the superpixel block p is taken as the ${L_T}^p$.

5. Semantic annotation based on ABDF

To build an accurate semantic segmentation model, a strategy similar to Boosting [38,39] is adopted to convert the RF training process into a problem of minimizing the global energy function. The core idea is to transform the combination of several weak classifiers into a strong one, and the most likely category is selected as the predicted label through balloting. In general, for given labeled training samples $\{x_i, y_i\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^M$ and $y_i \in \mathbb{R}^K$, M and K are the dimensions of the input and output variables. RF typically describes a non-linear mapping $\mathcal{M}: \mathbb{R}^M \to \mathbb{R}^K$. During training of a RF, we train the decision trees with a random subset of the training data and the decision trees (DTs) are independently trained from each other. The mapping $\mathcal{M}$ is learned by an ensemble of DTs. During testing, for a given x, each decision tree (DT) returns a category probability distribution $p(y|x)$. The training of a DT is to recursively split the given training data into two partitions, the tree is grown until some stopping criterion is reached. Here, we use the maximum tree depth as the stopping criterion.

Suppose the maximum layer of the training tree is N_m. Tangent (tan) is chosen for the loss function, and ε is the current layer of the training tree ($\varepsilon = 1, 2, \ldots, N_m$). The splitting function is defined as follows:

$$\phi(X;\, D) = \begin{cases} 0 & \text{if } x(D_1) < D_2 \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

where D defines the splitting parameters, $D_1 \in \left\{ 1, 2, \ldots, M \right\}$ is the feature dimension and $D_2 \in \mathbb{R}$ is the threshold, $x(D_1)$ denotes the $D1$-th dimension in x.

Each node in a tree randomly samples a set of splitting functions $\phi(X;\, D^i)$, each routing the data into two disjoint subsets. All nodes will choose the optimal splitting function $\phi(X;\, D^*)$ according to a optimization function ν, which is defined as follows:

$$\nu = \frac{N_1}{N_1 + N_2} A(S_1) + \frac{N_1}{N_1 + N_2} A(S_2) \tag{9}$$

where S_1 and S_2 are sample sets routing to the left and right child nodes, the allocation to the left or right child node is decided by $\phi(X;\, D^i)$. N_1 is the size of sample S_1 and N_2 is the size of sample S_2. $A(S)$ is the local score of a set S of data samples (S_1 or S_2). The score can be distinct metrics, such as information entropy, the Gini index, and a loss function or classification error rate. Some of metrics are complementary to each other. The splitting strategy of the RF function can greatly influence the performance of classifier. To exploit useful information as much as possible, we propose a novel aggregated splitting strategy. Suppose $\left\{ F_i \| 1 \le i \le N_f \right\}$ are a set of N_f different splitting functions, we aggregate these functions to improve the efficiency and accuracy of the classifier, then the aggregated metric of the cost function $A(F)$ can be defined as:

$$A(F) = P(A_1(F), A_2(F), \ldots, A_{N_f}(F)) \propto \frac{1}{\rho} \sum_{i=1}^{N_f} \eta_i(F), \tag{10}$$

where $A_i(F)$ represents the value of the ith cost function, ρ is a constant, N_f is the number of aggregation models and $\eta_i(F)$ denotes a local score metric. In our method, we aggregate the information entropy function, the Gini index and special loss function to construct the cost function $A(F)$. The information entropy is defined as

$$\eta_1(F) = -\sum_{j=1}^{J} \left[p(j|F) \right], \tag{11}$$

where J is the number of categories, $p(j|F)$ is the probability belonging to category j, estimated from the set F. The Gini index is defined as

$$\eta_2(F) = \sum_{j=1}^{J} \left[p(j|F) \cdot (1 - p(j|F)) \right], \tag{12}$$

and the special loss function can be presented as

$$\eta_3(F) = \tan(p(j|F)), \tag{13}$$

To make the splitting function consider the sample weights, the cost function Eq. (10) is modified to a weighted metric by changing the estimation of the category distributions $p(j|F)$, namely

$$p(j|F) = \frac{\sum_{i=1}^{N_t} \chi_i \cdot w_i^{\varepsilon}}{\sum_{i=1}^{N_t} w_i^{\varepsilon}} \tag{14}$$

where χ_i is the value returned by the predictive function. If the label y_i of a sample $X_i \in F$ equals j, then χ_i is set to 1; otherwise, it is set to 0. The weighting formula is defined as follows:

$$w_i^{\varepsilon} = \left| \frac{\partial \xi(y_i, \Gamma_{1:\varepsilon-1}(X_i; \bar{D}))}{\partial \Gamma(X)} \right| \tag{15}$$

where X_i is the training sample, ε is the number of iterations, $\xi(\cdot)$ is a differentiable loss function, and w_i^{ε} is the updated weight of sample i for the ε th iteration

$$\Gamma_{1:\varepsilon-1}(X; \bar{D}) = \sum_{i=1}^{\varepsilon-1} \beta_i \cdot \varphi_i(X; D^i) \tag{16}$$

where $\varphi_i(X; D^i)$ is the classifier of the i th iteration, $\bar{D}$ denotes the parameters of the trained weak classifier, Γ_i represents the training parameters of the current i th iteration layer and β_i is the shrinkage factor. The process of splitting the node in stage ε and updating the weight $w_i^{\varepsilon+1}$ for the next iteration is repeated until the maximum layer is reached.

Next, we further optimize the semantic annotation results based on a Markov conditional random field (*CRF*) model.

As a mathematical model of semantic segmentation, the *CRF* [40] is a hot research topic recently, and many algorithms have been studied. Boykov et al. [41] proposed the construction of a Markov *CRF* with data and a smooth term, mainly for image segmentation. A diagram $G = \langle V, E \rangle$ is constructed for each corresponding image, where each node $v_i = V$ in the diagram represents one superpixel and each edge $e_{ij} \subset E$ corresponds to the similarity between adjacent superpixel blocks. The energy function $E(f)$ is composed of two parts, the data item $E_{data}(f)$, and the smooth term $E_{smooth}(f)$, and thus the problem of semantic annotation can be equivalently transformed into the problem of vertex label assignment.

The energy function can be defined as

$$E(f) = E_{data}(f) + E_{smooth}(f) \tag{17}$$

where $E_{data}(f) = -\log P_i(p_i | T_n^p, T_d^p, T_h^p, T_r^p, T_g^p, L_H^p, L_T^p)$ and $E_{smooth}(f) = \left[p_i \neq p_j \right] \cdot \exp\left(-\frac{\| r_i - r_j \|}{2 * m^2} \right)$, the smooth term of the energy function is determined by every superpixel r_i and its neighboring superpixel r_j, $\| r_i - r_j \|$ is the mean difference of the *RGB* color of two adjacent two superpixels, and m is set to 2 in the experiment.

The ABDF algorithm is outlined in Algorithm 1.

6. Experiment

In this section, we adopt the challenging CamVid database [42] to evaluate the performance of the proposed algorithm. This dataset consists of four longer clips of driving sequences, with a total duration of approximately 10 min. The video are captured by the camera mounted on a fast-moving platform. The image resolution is 960 × 720 and camera intrinsic and extrinsic parameters are also provided. The database contains residential, suburban and street views. Annotations are sparsely provided at 1 Hz for 32 semantic classes. Labeled colors for each object class are shown in Fig. 2. In our experiments, we train and test on 13 classes shown in Fig. 2, while most of the previous work [17,21] only processed 11 categories as common practice when evaluating on CamVid database. In our framework, we consider cars, lane, pedestrian, column-pole, trafficlight, bicyclist, building, tree, sky, road, fence, wall, and sidewalk as the object classes and others as the background. Quantitative comparisons with state-of-the-art approaches are provided.

To further validate the reliability, robustness and efficiency of the proposed method, we also use four video sequences of our own. Each sequence consists of two videos: the school video and the street video. For each sequence, one video is used for training and the other is used for testing. The image resolution is 960 × 540. The school video contains eight categories: the sky, buildings, cars, roads, pedestrians, bicycles, flowerbeds and trees.

All algorithm results are achieved by Visual Studio 2010 and OpenCV using an Intel(R) core(TM) i5-3470 processor with 3.2 GHz frequency and 4 GB memory.

To evaluate the performance of the algorithms, we use pixel-wise percentage accuracy rate (PW-AR) as the evaluation metric. Let n_{ij} denote the number of pixels of class i predicted to belong to class j, $m_i = \sum_j n_{ii}$ denote the total number of pixels of class i. The PW-AR is defined as $\sum_i n_{ii} / \sum_i m_i$.

6.1. Evaluate the performance with our own dataset

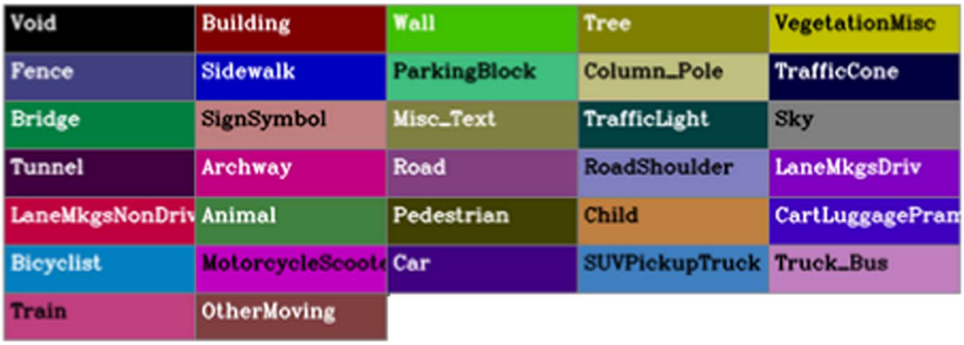
To validate the performance of the overall framework, our own datasets are also used. Four video sequences were taken by high-definition camera. According to the video content, they are divided into two sequences, the school sequence (SchoolVideo01, SchoolVideo02) and the street sequence (StreetVideo01, StreetVideo02). The SchoolVideo01 video sequence is used for training, and SchoolVideo02 is used for testing. For street view, StreetVideo01 is used for training, and StreetVideo02 is used for testing. The playback speed for the video sequences is 25 frames per second.

Fig. 3 shows the comparison results of our method with some existing methods. The superpixel segmentation result and the depth map are also given. We conduct two experiments on this sequence. In the first experiment, we use only StreetVideo01 to validate the efficiency of the proposed method. Under identical conditions (same feature dimensions and same dataset), we compare the proposed ABDF method with the RF Boosting algorithm [43] and the CN algorithm [25]. We use 10 frames of StreetVideo01 to build the ABDF model and the RF Boosting based model, and the remaining frames are used as the test set. Because the CN method cannot build an effective model using only a small amount of samples, we use 50 samples to build its model. The experiment shows that our method obtains more accurate and smoother edge details and achieves much higher PW-AR.

In the second experiment, both StreetVideo01 and StreetVideo02 are used to validate the proposed method. We use 10 frame and 50 frames of StreetVideo01 to build the ABDF model and the CN model respectively, and StreetVideo02 is used as the test set. In StreetVideo01 and StreetVideo02, the image resolution is 960 × 540. Fig. 4 illustrates the predicted results of the StreetVideo02 sequences.

Fig. 5 gives two frames of SchoolVideo01. We use 10 frames and 50 frames of SchoolVideo01 to build ABDF model and CN model respectively. The image has a resolution of 960 × 540, and 87 dimensions of features are extracted from each superpixel, including 82 dimensions of 2D features (65-dimensional color histogram and 17-dimensional texture feature) and five types of 3D features, including the sky, buildings, cars, roads, flowerbeds, pedestrians, bicycles, and trees. For each category, a different color is used for annotation. The SchoolVideo02 is used as the test set. This sequence includes a total of 150 frames. The comparison of the annotated results using CN method and the proposed method is shown in Fig. 8. From these experiments, it can be seen that the ABDF model obtains more accurate results, especially in the annotation of some complex objects, such as trees (Fig. 6) .

To show the efficiency of the proposed method, we compare our method with the CN method [25] and the CN method combing with spatial prior (CN+spatial prior) [25] under different settings, which are DL-based methods. As shown in Table 2, the average runtimes (ATs) of CN method and CN+spatial prior are 31 s and 41 s, respectively. The proposed method consumes only 2 s. To explore the influence of the number of training samples on the



Void	Building	Wall	Tree	VegetationMisc
Fence	Sidewalk	ParkingBlock	Column_Pole	TrafficCone
Bridge	SignSymbol	Misc_Text	TrafficLight	Sky
Tunnel	Archway	Road	RoadShoulder	LaneMkgsDriv
LaneMkgsNonDriv	Animal	Pedestrian	Child	CartLuggagePram
Bicyclist	MotorcycleScoot	Car	SUVPickupTruck	Truck_Bus
Train	OtherMoving			

Fig. 2. List of class labels and corresponding colors. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

segmentation results, we use 10, 20, 50, 100 samples of StreetVideo01 to build the models respectively, and use StreetVideo02 for validation. The result is presented in Table 2. From experiments, we found that, to build an effective model, the CN method requires at least 50 samples. However, our method can build an accurate model by using only few samples. It can be seen that the PW-AR increases with the augmentation of samples for all three methods. Moreover, our method achieves much higher PW-AR than other two methods. When only 10 samples are used to build the model, we obtain PW-AR of 84.20%. When the number of samples increases to 50, the PW-AR increases to 88.21%, while CN method achieves a PW-AR of 71.12%.

Finally, we illustrate the comparison results of PW-AR on different datasets in Table 3. The proposed ABDF method achieves PW-ARs of 85.79%, 84.20%, 80.90% on the three video datasets, respectively. However, CN method achieves PW-ARs of 72.91%, 79.66%, 72.54% respectively. Our method can build an efficient and accurate model using fewer samples, and it achieves much higher PW-AR and is much more efficient than CN-method (Table 4).

6.2. *Evaluate the performance with the Camvid database*

In the Camvid database, two groups of the labeled training data, 0016E5 and 0006R0, are used for day sequence training data, and another group 0016E5_15Hz are used for testing.

6.2.1. *Overall performance*

We compared our method with the method based on ego-motion-based 3D point clouds (EMD) [21] and the method based on dense depth map (DDM) [17], the SP-SS method [19] and SRT-SS method [20]. Table 1 shows the quantitative evaluation results. Our method achieves PW-AR 84.7%, SP-SS and SRT-SS achieve PW-ARs of 76.9% and 77.4%, EED and DDM achieve PW-ARs 69.1% and 82.1%. Our method achieves the highest PW-AR and is much better than other methods. Figs. 7 and 8 give two examples of segmentation results achieved by SP-SS, SRT-SS and our method. It can be seen that our method achieves much better results than SP-SS and SRT-SS for the annotation of main objects, especially for the complicated objects, such as car, tree and pedestrian. Our model contains 100 trees trained to a maximum depth of 15. The learning takes only about 18 min and testing takes about 1.5 s per frame. For the SRT-SS method, the total runtime is about 35 h and testing takes about 78 min per frame. For the SP-SS method, the total runtime is about 3 h and testing takes about 0.55 min per frame. Note that, the computation of our method is much more efficient than the other two methods. The experiments validate the accuracy and efficiency of our method. The experiments also show that the proposed algorithm can accurately segment more complex scenes, which has 13 categories.

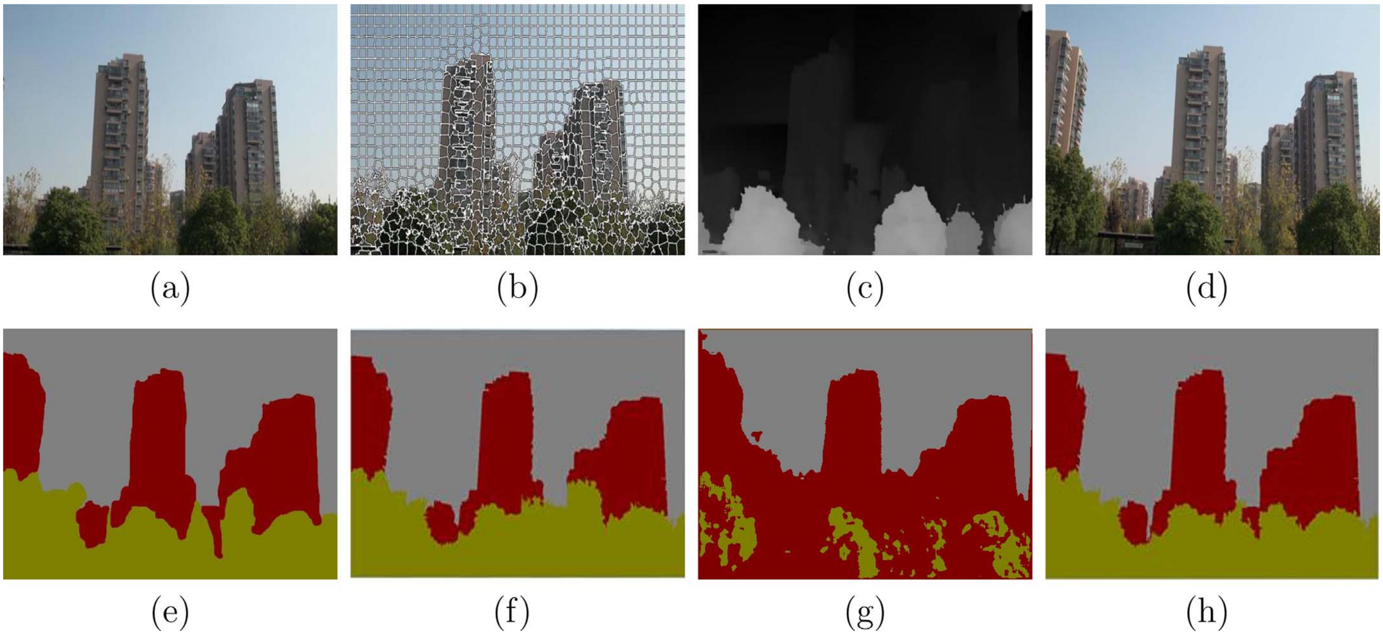

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 3. Comparison of annotation results based on different methods using StreetVideo01. (a) A train video frame, (b) the superpixel segmented result, (c) the depth map of the original image, (d) a test video frame, (e) manually annotated result, (f) RF Boosting method, (g) CN method and (h) ABDF method.

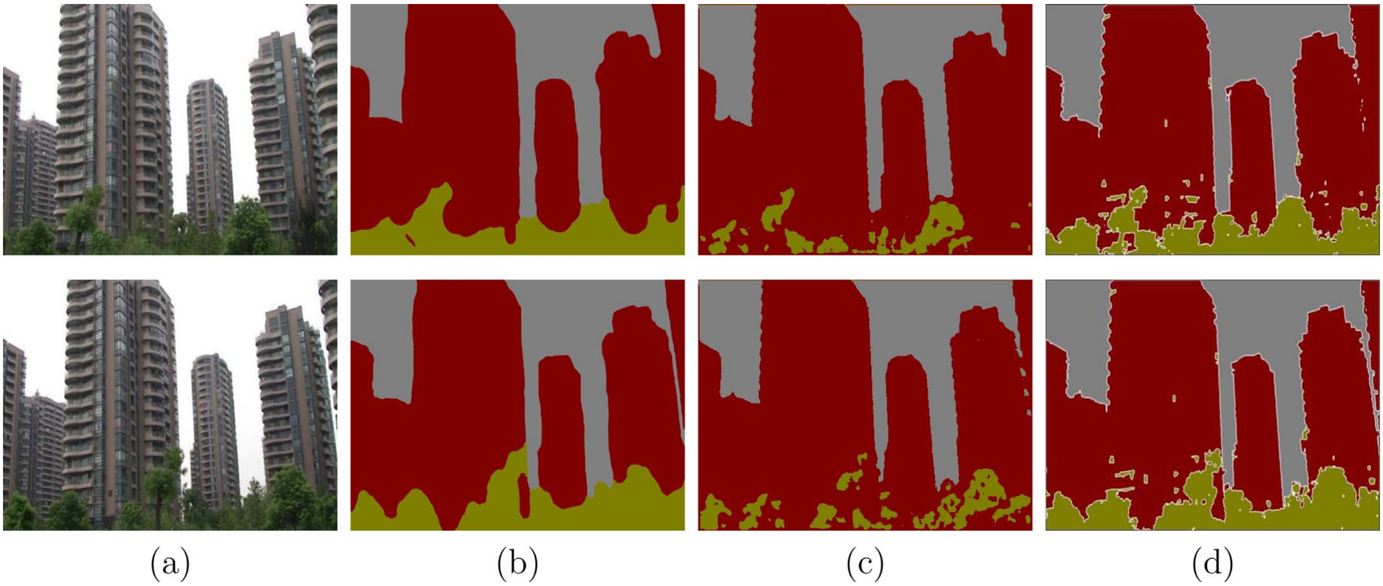


Fig. 4. Comparison of annotation results based on different methods using StreetVideo02. (a) Two test video frames, (b) manually annotated results, (c) CN method and (d) ABDF method.

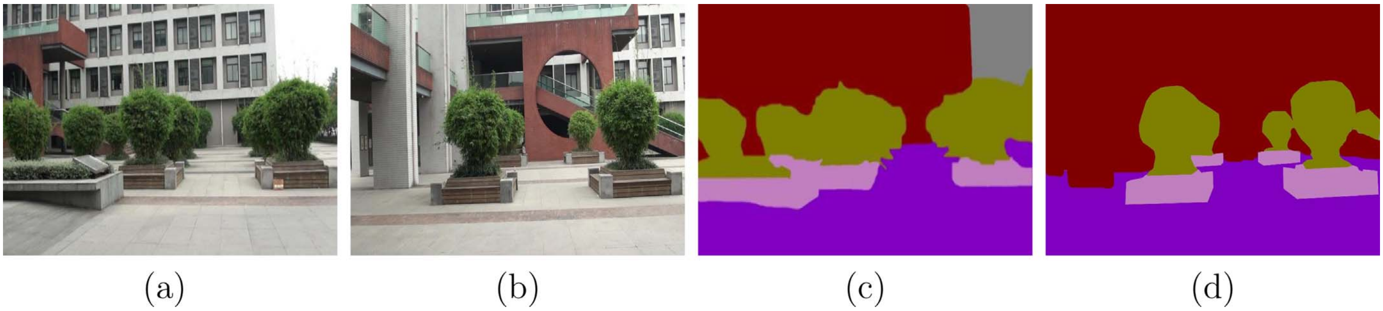


Fig. 5. Two frames of SchoolVideo01 and their annotated benchmark. (a and b) Two frames; (c and d) the manually annotated results.

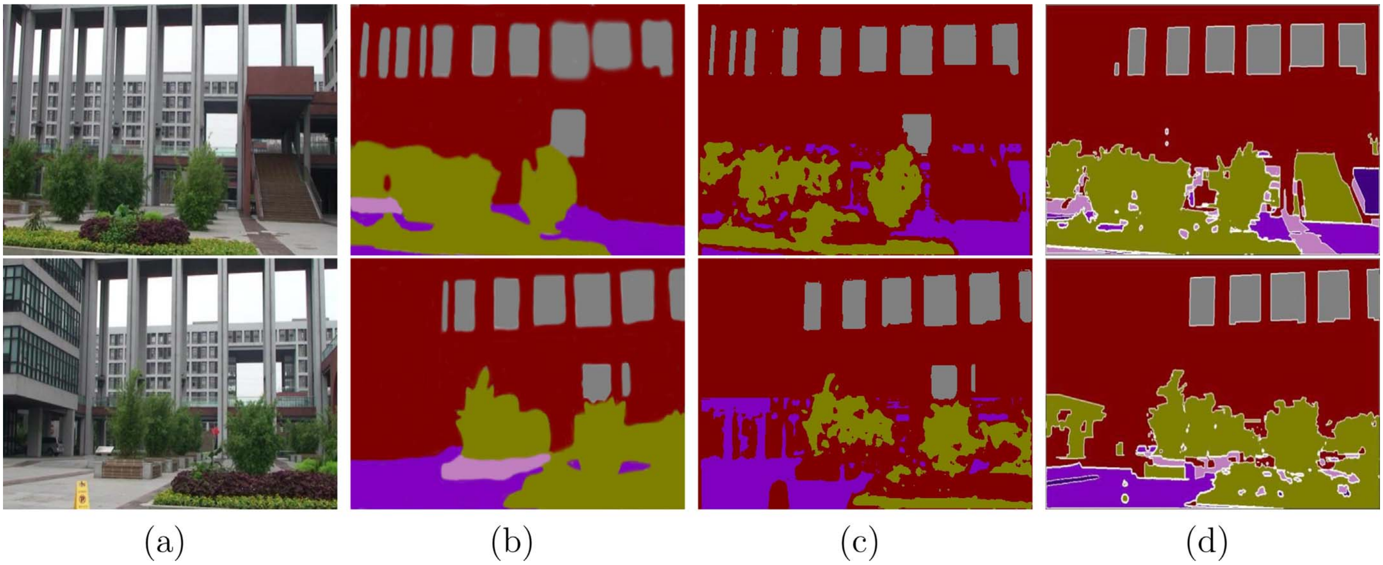


Fig. 6. Comparison of annotation results based on different methods using SchoolVideo02. (a) Two test video frames, (b) manually annotated results, (c) CN method and (d) ABDF method.

6.2.2. Per class performance

To further evaluate the effectiveness and the efficiency of the proposed method, we also use the other four metrics from common semantic segmentation and scene parsing evaluations that are variations of region intersection over union (IU) [29]. Let n_c denote the number of different classes. For class i, the per class accuracy rate (PC-AR) is defined as n_{ii}/m_i, then the mean PC-AR is defined as $(1/n_c)/\sum_i n_{ii}/m_i$. The per class IU (PC-IU) is defined as $n_{ii}/(m_i + \sum_j n_{ji} - n_{ii})$, then the mean PC-IU is defined as

Table 1
The average runtime of processing an sample using different methods.

Method	Image size	Run time (s)
CN method [25]	960 × 540	31
CN+spatial prior [25]	960 × 540	41
ABDF	960 × 540	**2**

Table 2
Comparison of PW-AR (%) based on different methods using different number of samples.

Method	10	20	50	100
CN method	–	–	70.12	78.42
CN+spatial prior	–	–	79.66	84.81
ABDF	**84.20**	**86.37**	**88.21**	**90.11**

Table 4
Comparison of PW-AR (%) with different methods.

Method	PW-AR
SP-SS [19]	76.9
SRT-SS [20]	77.4
EMD [21]	69.1
DDM [17]	82.1
ABDF	**84.7**

$(1/n_c)/\sum_i n_{ii}/(m_i + \sum_j n_{ji} - n_{ii})$.

In this section, the experiments are implemented with two objectives. First, we compare the per class performance of the new algorithm with other three learning-based method. The SP-SS method [19] is a semantic segmentation method based on lazy learning, the SRT-SS method [20] is based on transfer learning, and the FCN method [29] is based on deep learning. Second, we assess the effectiveness of our appearance feature model. To achieve the second objective, we also build an appearance model based on only 2D features, which is denoted as 2D-only. We compute the PW-AR, PC-AR, PC-IU, the mean PC-AR and the mean PC-IU, and record the runtime of training and testing. The experimental results are listed in Tables 5 and 6, respectively. The PW-ARs of SP-SS, SRT-SS, FCN and our method are 76.41%, 79.86%, 49.98% and 84.7%, respectively. The mean PC-ARs of SP-SS, SRT-SS, FCN and our method are 47.49%, 44.95%, 22.45% and 53.16%, respectively. The mean PC-IUs of SP-SS, SRT-SS, FCN and our method are 37.74%, 38.01%, 14.04% and 45.78%, respectively. It demonstrates that our method achieves much better results than other methods. The FCN method achieves the worst results on the Camvid database, and it does not work well on dataset with a small sample size. The runtime of training a classifier and testing for SP-SS, SRT-SS, FCN and our method are listed in Table 7. It shows that the proposed method is much more efficient than other three methods.

The experiments also evaluate the effectiveness of the proposed appearance model. The fusion of 2D features and 3D features achieves much higher PC-AR and PC-IU than the model based on only 2D features.

6.2.3. Evaluate the performance of ABDF

To show the efficiency of ABDF method, we also compared it with several most related competing methods, i.e., the ADF algorithm [23], Boosted algorithm (BT) [44] and RF algorithm [6] by using five standard machine learning databases. The five datasets are shown in Table 8. For a fair comparison of all methods, we set the common parameters to the same values. The number of trees T is set to 100 (for BT, this is equivalent to the number of weak learners), the maximum depth *Dmax* of the trees is set to either 10, 15, or 25 (depending on the size of the training data), the number of random thresholds is set to 10 per node and the minimum number of samples for further splitting is set to 5. For ADF and BT, we also evaluate three different loss functions that can be integrated in the Gradient Boosting formulation, i.e. Exponential, Savage and tan. The results are illustrated in Table 9. We also investigate the influence of the loss function on the annotation results. It can be seen that the proposed ABDF model achieves the best results on all five datasets. For G50c, BTs with Savage loss function obtains the second best results. For ADF method, tan loss function achieves better results than that of the other two loss functions and for BT, Savage function achieves better results on most of datasets.

Table 3
Comparison of PW-AR (%) based on different methods using different datasets.

Method	StreetVideo01	StreetVideo02	SchoolVideo02
CN method	67.58	70.12	61.46
CN+spatial prior	72.91	79.66	72.54
ABDF	**85.79**	**84.20**	**80.98**

We further evaluate the computational efficiency of these four classifiers. The results are shown in Table 10. The runtime of the proposed method, ADF and RF are roughly the same. Our method consumes slightly more time (approximately 0.1–0.5 s) than ADF, however, it achieves higher accuracy. Note that, our algorithm achieves significantly better results in semantic annotation for complex video street.

7. Conclusion

In this paper, we have presented a new methodology based on 2D and 3D features fusion and an aggregated boosting decision forest for semantic annotation of video street view. We adopt an appearance model based on the combination of 2D and 3D features and depth information. The multi-feature fusion can improve the robustness of the model. Besides feature fusion, we propose a ABDF algorithm to built the classifier by using a modified integrated splitting strategy for decision tree. The experiments have validated the efficiency of the proposed method. The results demonstrate that our method improves the performance of semantic segmentation based on three evaluation metrics, i.e. PW-AR, PC-AR and PC-IU, and simultaneously reduces the time and memory consumption. Compared with the state-of-the-art DL-based methods, the proposed method has three advantages. First, our method only needs a small amount of samples to build a robust model. Second, our method achieves higher PW-AR, PC-AR and PC-IU when only a small number of samples are available. Third, the computation of our method is much more efficient for both training and testing. In summary, the proposed methodology achieves over 84% PW-AR, 53% mean PC-AR and 45% mean PC-IU in the Camvid database by using 305 samples for training. The experimental results demonstrated that our proposed method is superior to that of existing semantic annotation methods in terms of accuracy and computation efficiency, and can be used in real-time video processing applications. However, some disadvantages still exist. For example, small objects, such as columns, contain only a few pixels after segmentation, and they will be merged with adjacent regions, resulting in relatively rough recognition results.

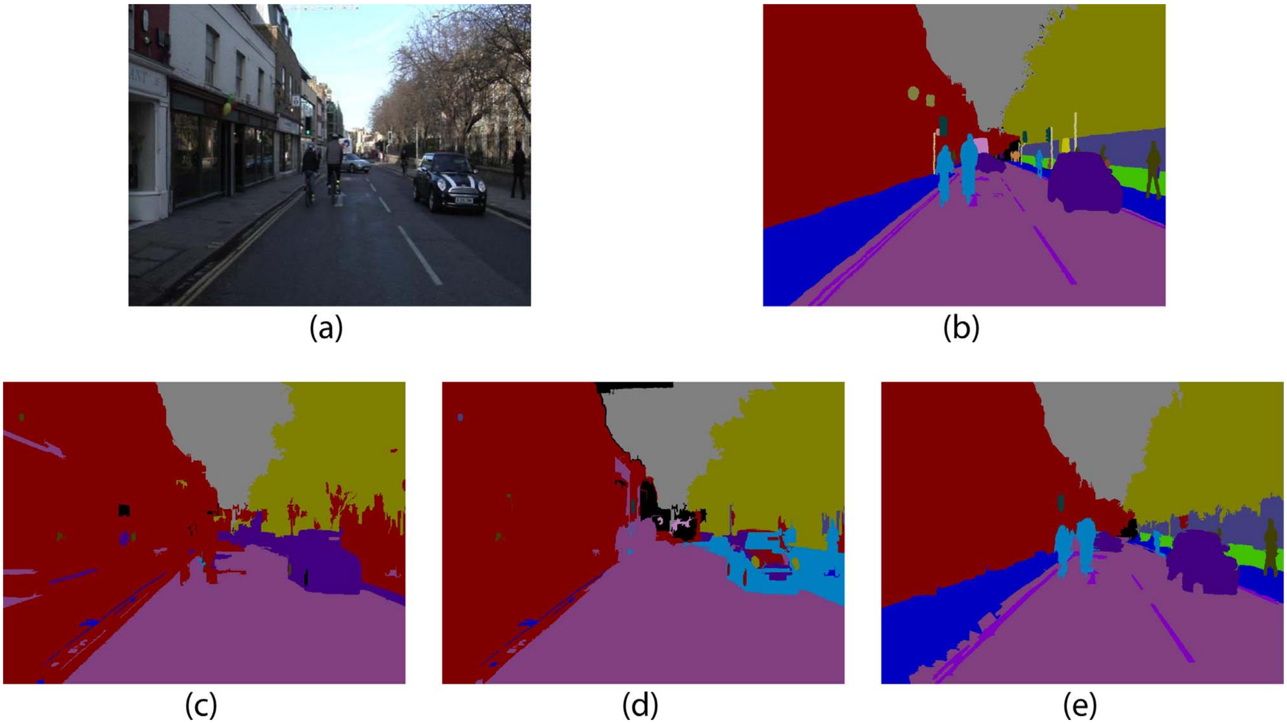


Fig. 7. Example 1 of semantic segmentation comparisons. (a) One video street view frame in Camvid database; (b) manually annotated results (ground truth); (c) SP-SS; (d) SRT-SS; (e) our method.

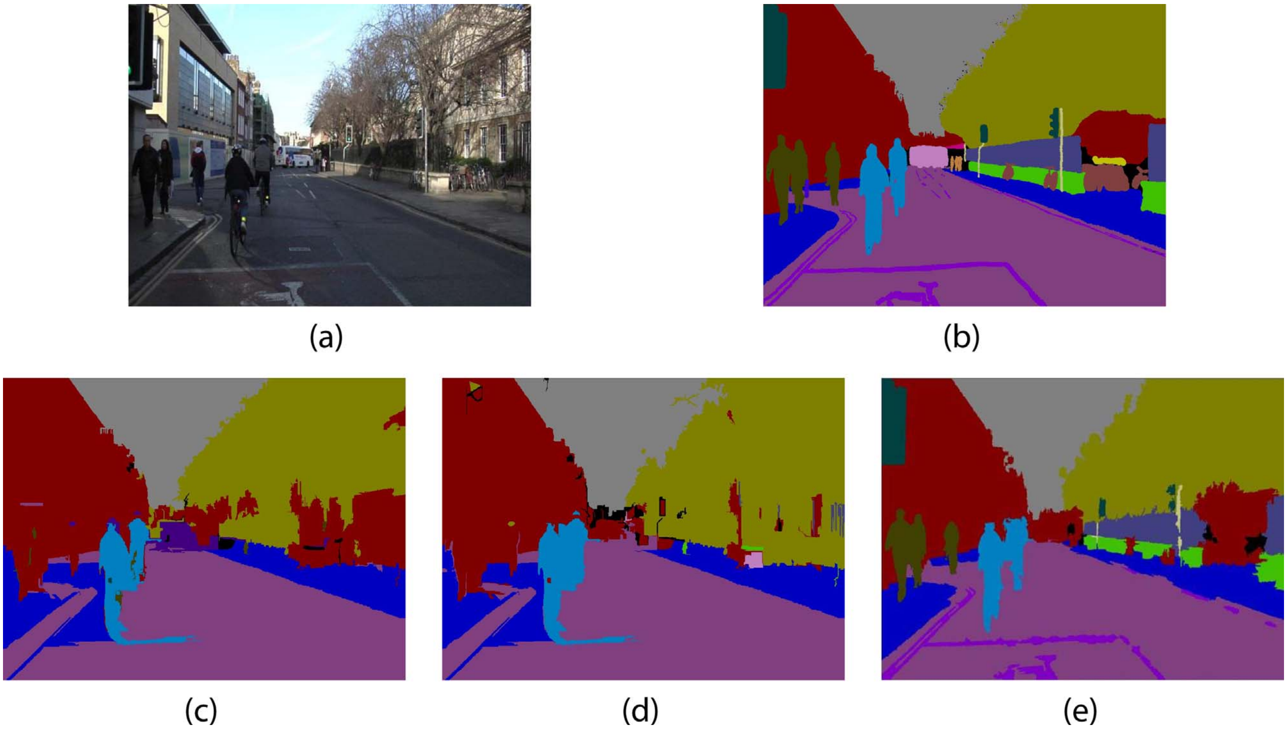


Fig. 8. Example 2 of semantic segmentation comparisons. (a) One video street view frame in Camvid database; (b) manually annotated results (ground truth); (c) SP-SS; (d) SRT-SS; (e) our method.

Table 5
Comparison of PC-AR (%) with different methods.

Method	Bicyclist	Building	Car	Pole	Fence	Lane	Pedestrian	Road	Sidewalk	Sky	TrafficLgt	Tree	Wall	Mean
SP-SS	45.42	80.2	**50.01**	0.00	0.10	46.87	**29.46**	92.49	**59.28**	**97.11**	**36.67**	77.79	1.91	47.49
SRT-SS	33.45	85.3	40.3	2.54	20.56	43.29	1.03	93.64	48.17	96.97	8.51	94.74	15.91	44.95
FCN	1.46	48.98	21.83	0.64	8.09	1.03	5.06	91.79	19.2	73.46	0.15	16.38	3.73	22.45
Ours	**55**	**91.55**	45.79	**7.36**	**23.47**	**58.90**	21.80	**95.14**	53.51	96.41	28.86	**95.23**	**18.10**	**53.16**
2D-only	42.99	86.70	26.80	2.41	7.68	49.10	19.76	88.97	36.86	90.59	23.21	87.00	13.27	44.26

Table 6
Comparison of PC-IU (%) with different methods.

Method	Bicyclist	Building	Car	Pole	Fence	Lane	Pedestrian	Road	Sidewalk	Sky	TrafficLgt	Tree	Wall	Mean
SP-SS	20.63	57.44	31.86	0.00	0.10	39.15	9.91	87.07	40.99	**92.42**	**36.29**	72.92	1.90	37.74
SRT-SS	24.89	71.86	29.86	2.51	18.10	35.96	0.98	**87.83**	36.87	91.01	8.50	74.56	11.16	38.01
FCN	1.26	30.78	5.60	0.47	5.54	0.86	2.71	68.08	15.03	36.24	0.14	12.76	3.09	14.04
Ours	**45.49**	**76.70**	**37.42**	**6.72**	**22.21**	**46.73**	**13.73**	81.07	**51.51**	91.41	24.01	**81.42**	**16.66**	**45.78**
2D-only	36.34	66.78	21.77	2.28	7.47	40.35	11.80	72.21	34.22	87.74	22.02	76.74	12.23	37.84

Table 7
The comparison of runtime (s) with different methods.

Method	Training	Testing (per frame)
SP-SS	12,320	33
SRT-SS	125,640	4692
FCN	49,200	510
ABDF	**1080**	**2**

Table 8
Five machine-learning databases.

Dataset	#Train	#Test	#Feature	#Classes
G50c	50	500	50	2
USPS	7291	2007	256	10
MNISTS	60,000	10,000	784	10
Letter	16,000	4000	16	26
Char74K	66,707	7400	64	62

Table 9
Comparison of error rate (%) by using different models.

Method	Loss	G50c	USPS	MNISTS	Letter	Char74K
ABDF	Tan	**18.57 ± 1.31**	**5.56 ± 0.10**	**2.68 ± 0.08**	**3.35 ± 0.15**	**16.59 ± 0.25**
	Tan [45]	18.71 ± 1.27	5.59 ± 0.16	2.71 ± 0.10	3.52 ± 0.12	16.67 ± 0.21
ADF [23]	Savage [46]	19.00 ± 1.32	5.76 ± 0.16	2.78 ± 0.09	3.94 ± 0.14	16.92 ± 0.15
	Exp	19.09 ± 1.17	6.03 ± 0.29	2.96 ± 0.05	4.27 ± 0.13	16.82 ± 0.15
	Tan [45]	18.90 ± 1.31	5.93 ± 0.27	3.15 ± 0.05	4.70 ± 0.18	17.59 ± 0.29
BT	Savage [46]	18.87 ± 1.31	5.92 ± 0.19	3.19 ± 0.07	4.65 ± 0.12	17.62 ± 0.25
	Exp	18.91 ± 1.30	5.83 ± 0.19	3.17 ± 0.07	4.78 ± 0.12	17.57 ± 0.21
RF [6,47]		18.91 ± 1.27	5.96 ± 0.21	3.21 ± 0.07	4.75 ± 0.10	17.76 ± 0.13

Table 10
Comparison of runtime (s) using different models.

Method	G50c	USPS	MNISTS	Letter	Char74K
ABDF	1.1	1.2	1.4	1.3	1.5
ADF	1.0	1.0	1.0	1.0	1.0
BT	3.99	6.55	7.05	6.55	7.09
RF	1.55	0.45	0.79	0.45	1.52

Acknowledgments

This work was supported in part by grants from National Key Technology Research and Development Program of the Ministry of Science and Technology of China (Grant no. 2014BAK14B01), National Science Foundation of China (Grant nos. 61472362, 61170098, 61379075), Zhejiang Provincial Natural Science Foundation of China (Grant no. LZ16F020002).

References

[1] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, Y. Kompatsiaris, S. Staab, M.G. Strintzis, Semantic annotation of images and videos for multimedia analysis, in: The Semantic Web: Research and Applications, Springer, Berlin, Heidelberg, 2005, pp. 592–607.

[2] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 891–898.

[3] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, J. Malik, Semantic segmentation using regions and parts, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3378–3385.

[4] F. Meng, H. Li, G. Liu, K.N. Ngan, From logo to object segmentation, IEEE Trans. Multimed. 15 (8) (2013) 2186–2197.

[5] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.

[6] Y. Amit, D. Geman, Shape quantization and recognition with randomized trees, Neural Comput. 9 (7) (1997) 1545–1588.

[7] A. Criminisi, J. Shotton, Decision Forests for Computer Vision and Medical Image Analysis, Springer, London, 2013.

[8] J. Shotton, M. Johnson, R. Cipolla, Semantic texton forests for image categorization and segmentation, in: 2008 IEEE Conference on Computer vision and pattern recognition, IEEE, 2008, pp. 1–8.

[9] J. Gall, V. Lempitsky, Class-specific hough forests for object detection, in: Decision Forests for Computer Vision and Medical Image Analysis, Springer, London, 2013, pp. 143–157.

[10] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, Adv. Neural Inf. Process. Syst. 19 (2007) 985–992.

[11] R. Caruana, N. Karampatziakis, A. Yessenalina, An empirical evaluation of supervised learning in high dimensions, in: Proceedings of the 25th International Conference on Machine Learning, ACM, New York, 2008, pp. 96–103.

[12] E. Chang, K. Goh, G. Sychay, G. Wu, Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines, IEEE Trans. Circuits Syst. Video Technol. 13 (1) (2003) 26–38.

[13] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, D.N. Metaxas, Automatic image annotation using group sparsity, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3312–3319.

[14] G. Carneiro, A.B. Chan, P.J. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 29 (3) (2007) 394–410.

[15] R. Socher, L. Fei-Fei, Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 966–973.

[16] H. Cheng, Z. Liu, L. Yang, X. Chen, Sparse representation and learning in visual recognition: theory and applications, Signal Process. 93 (6) (2013) 1408–1425.

[17] C. Zhang, L. Wang, R. Yang, Semantic segmentation of urban scenes using dense depth maps, in: Computer Vision—ECCV 2010, Springer, Berlin, Heidelberg, 2010, pp. 708–721.

[18] C. Liu, J. Yuen, A. Torralba, Nonparametric scene parsing: Label transfer via dense scene alignment, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1972–1979.

[19] T. Joseph, L. Svetlana, Superparsing: scalable nonparametric image parsing with superpixels, Int. J. Comput. Vis. 101 (2) (2012) 352–365.

[20] M. Heesoo, M.L. Kyoung, Tensor-based high-order semantic relation transfer for semantic scene segmentation, in: 2013 IEEE conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3073-3080.

[21] G.J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and recognition using structure from motion point clouds, in: Computer Vision—ECCV 2008, Springer, Berlin, Heidelberg, 2008, pp. 44–57.

[22] J. Xiao, L. Quan, Multiple view semantic segmentation for street view images, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 686–693.

[23] S. Schulter, P. Wohlhart, C. Leistner, A. Saffari, P.M. Roth, H. Bischof, Alternating decision forests, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 508–515.

[24] A.G. Schwing, C. Zach, Y. Zheng, M. Pollefeys, Adaptive random forest how many experts to ask before making a decision?, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1377–1384.

[25] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, J. Denzler, Convolutional patch networks with spatial prior for road detection and urban scene understanding, in: Proceedings of the 10th International Conference on Computer Vision Theory and Applications, 2015, pp. 510–517.

[26] G. Ross, D. Jeff, D. Trevor, M. Jitendra, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 580-587.

[27] G. Saurabh, G. Ross, A. Pablo, M. Jitendra, Learning rich features from rgb-d images for object detection and segmentation, in: Computer Vision – ECCV 2014, Springer International Publishing, Switzerland, 2014, pp. 345-360.

[28] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, 2015, pp. 1520-1528.

[29] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 3431-3440.

[30] Z. Lu, H.H. Ip, Q. He, Context-based multi-label image annotation, in: Proceedings of the ACM International Conference on Image and Video Retrieval, ACM, New York, 2009, pp. 1-7.

[31] M. Jamieson, A. Fazly, S. Stevenson, S. Dickinson, S. Wachsmuth, Using language to learn structured appearance models for image annotation, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2010) 148–164.

[32] X.-J. Wang, L. Zhang, M. Liu, Y. Li, W.-Y. Ma, Arista-image search to annotation on billions of web photos, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2987–2994.

[33] G. Zhang, J. Jia, T.-T. Wong, H. Bao, Consistent depth maps recovery from a video sequence, IEEE Trans. Pattern Anal. Mach. Intell. 31 (6) (2009) 974–988.

[34] G. Zeng, P. Wang, R. Gan, H. Zha, Structure-sensitive superpixels via geodesic distance, in: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 447-454.

[35] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.

[36] A. Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, K. Siddiqi, Turbopixels: fast superpixels using geometric flows, IEEE Trans. Pattern Anal. Mach. Intell. 31 (12) (2009) 2290–2297.

[37] M.V. den Bergh, X. Boix, G. Roig, L.J.V. Gool, Seeds: superpixels extracted via energy-driven sampling, Int. J. Comput. Vis. 111 (3) (2015) 298–314.

[38] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

[39] N.C. Oza, Online bagging and boosting, in: 2005 IEEE international conference on Systems, man and cybernetics, vol. 3, IEEE, 2005, pp. 2340–2345.

[40] X. He, R.S. Zemel, M. Carreira-Perpindn, Multiscale conditional random fields for image labeling, in: 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, IEEE, 2004, pp. II–695.

[41] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (11) (2001) 1222–1239.

[42] G.J. Brostow, J. Fauqueur, R. Cipolla, Semantic object classes in video: a high-definition ground truth database, Pattern Recognit. Lett. 30 (2) (2009) 88–97.

[43] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn. 37 (3) (1999) 297–336.

[44] H. Trevor, T. Robert, F. Jerome, The elements of statistical learning, Data Min. Inference Predict. (2000) 587–603.

[45] M.-s. Hamed, M. Vijay, V. Nuno, On the design of robust classifiers for computer vision, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 779–786.

[46] H. Masnadi-shirazi, N. Vasconcelos, On the design of loss functions for classification: theory, robustness to outliers, and savageboost, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, 2009, pp. 1049–1056.

[47] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

Xun Wang is currently a professor at the School of Computer Science and Information Engineering, Zhejiang Gongshang University, China. He received his B.Sc. in mechanics, M.Sc. and Ph.D. degrees in computer science, all from Zhejiang University, Hangzhou, China, in 1990, 1999 and 2006, respectively. His current research interests include mobile graphics computing, image/video processing, pattern recognition and intelligent information processing. In recent years, He has published over 80 papers in high-quality journals and conferences. He holds 9 authorized invention patents and 5 provincial and ministerial level scientific and technological progress awards. He is a member of the IEEE and ACM, and a senior member of CCF.

Guoli Yan was born in 1991. She is pursuing her Master's degree in computer science and technology in Zhejiang Gongshang University. Her interests include video/image processing and pattern recognition.

Huiyan Wang was born in Yantai, China. She received the M.S. degree in power engineering from Shandong University, Jinan, China and the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1999 and 2003, respectively. Then she conducted as a postdoctoral research fellow (2003–2005) in clinical medicine from pharmaceutical informatics institute, Zhejiang University, Hangzhou, China. She is currently a professor of Computer Science and Technology in the school of Computer Science and Information Engineering, Zhejiang Gongshang University, China. Her current interests are biomedical signal processing and pattern recognition. She also works on image and video processing.

Jianhai Fu was born in 1989. He is pursuing his Master's degree in computer science and technology in Zhejiang Gongshang University. His interests include video/image processing and pattern recognition.

Jing Hua is a Professor of Computer Science and the founding director of Computer Graphics and Imaging Lab (GIL) and Visualization Lab (VIS) at Computer Science at Wayne State University (WSU). Dr. Hua received his Ph.D. degree (2004) in Computer Science from the State University of New York at Stony Brook. He also received his M.S. degree (1999) in Pattern Recognition and Artificial Intelligence from the Institute of Automation, Chinese Academy of Sciences in Beijing, China and his B.S. degree (1996) in Electrical Engineering from the Huazhong University of Science & Technology in Wuhan, China. His research interests include Computer Graphics, Visualization, Image Analysis and Informatics, Computer Vision, etc. He has published over 100 peer-reviewed papers in the above research fields at top journals and conferences, such as IEEE Transactions on Visualization and Computer Graphics, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Visualization, MICCAI, CVPR, ICDM, etc.

Jingqi Wang was born in 1992. He is pursuing his Master's degree in computer science and technology in Zhejiang Gongshang University. Her interests include video/image processing and pattern recognition.

Yutao Yang was born in 1993. He is pursuing his Master's degree in computer science and technology in Zhejiang Gongshang University. His interests include video/image processing and pattern recognition.

Guofeng Zhang received the B.S. and Ph.D. degrees in Computer Science from Zhejiang University, in 2003 and 2009, respectively. He was a recipient of the National Excellent Doctoral Dissertation Award and the Excellent Doctoral Dissertation Award of the China Computer Federation. He is currently an associate professor at State Key Laboratory of CAD& CG, Zhejiang University. His research interests include structure-from-motion, 3D reconstruction, augmented reality, video segmentation and editing. He is a member of IEEE.

Hujun Bao is a Professor in the Computer Science Department of Zhejiang University, and the director of the state key laboratory of Computer Aided Design and Computer Graphics. He graduated from Zhejiang University in 1987 with a B.Sc. degree in mathematics, and obtained his Ph.D. degrees in applied mathematics from the same university in 1993. In August 1993, he joined the laboratory. He leads the virtual reality and visual analysis center in the lab, which mainly makes researches on geometry computing, 3D visual computing, real-time rendering, virtual reality and visual analysis. He has published a number of papers over the past few years. These techniques have been successfully integrated into our virtual reality system VisioniX, 3D structure recovery system from videos ACTS, the spatio-temporal information system uniVizal, and the 2D-to-3D video conversion system. His researches are supported by National Natural Science Foundation, the 973 program and the 863 program of China.