Scene transformation for detector adaptation

Liwei Liu a,⇑, Junliang Xing b, Genquan Duan a, Haizhou Ai a

aComputer Science and Technology Department, Tsinghua University, Beijing, China
bInstitute of Automation, Chinese Academy of Sciences, Beijing, China

A R T I C L E   I N F O

Article history:
Received 17 January 2013
Available online 16 October 2013

Communicated by A. Shokoufandeh

Keywords:
Vehicle detection
Detector adaptation
Scene transformation
Automatic parameter estimation

A B S T R A C T

This paper focuses on detecting vehicles in different target scenes with the same pre-trained detector which is very challenging due to view variations. To address this problem, we propose a novel approach for detection adaptation based on scene transformation, which contributes in both view transformation and automatic parameter estimation. Instead of modifying the pre-trained detectors, we transform scenes into frontal/rear view handling with pitch and yaw view variations. Without human interactions but only some general prior knowledge, the transformation parameters are automatically initialized, and then online optimized with spatial–temporal voting, which guarantees that the transformation matches the pre-trained detector. Since there is no need of labeling new samples and manual camera calibration, our approach can considerably reduce manual interactions. Experiments on challenging real-world videos demonstrate that our approach achieves significant improvements over the pre-trained detector, and it is even comparable to the performance of the detector trained on fully labeled sequences.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Vehicle detection in video sequences is of fundamental importance which provides strong observation models for many high level traffic surveillance applications such as traffic analysis, abnormal trajectory detection and collision avoiding. The difficulties behind vehicle detection, however, are also pronounced because of variations in views, resolutions, illuminations and backgrounds. In practical application, a pre-trained vehicle detector often performs worse in general scenes than in the training scene, and furthermore, it doesn’t work at all. The major reason that results in performance decrease is the view variation between training and testing scenes.

In recent years, the fast development of object detection techniques has resulted in many promising methods for detecting particular object classes, e.g., faces (Viola and Jones, 2004; Huang et al., 2007), pedestrians (Wu and Nevatia, 2007; Dalal and Triggs, 2005) and vehicles (Kuo and Nevatia, 2009; Song et al., 2008; Liu et al., 2012). Compared with techniques based on background subtraction (Kamijo et al., 2000), detection based techniques are more robust to lighting variations. But robust solutions of object detection for piratical applications need further research due to view variations in diverse scenes. Aiming at modeling the characteristics of samples from target scenes with few manual interactions, a popular trend is to design a labeler to select positive and negative samples from a target scene to retrain a scene specific detector (Ali et al., 2011; Kalal et al., 2010; Wang and Wang, 2011) or modify a general detector (Jain and Learned-Miller, 2011).

Besides the above approaches, there are a few attempts based on sample transformation to adapt the pre-trained detectors. In Li et al. (2008), the proposed 3D search approach significantly improved detection performance. At each grid point, a rectified sub-image is generated to approximate the orthogonal projection of the samples in which the pre-trained detector can be applied. As for most surveillance videos, the camera parameters are unknown which does not meet its requirement. Our work is inspired by the approach of Li et al. (2008). However, we focus on rigid targets such as vehicles in general scenes without camera parameters, and hence there are more difficulties should be handled with: (1) beside the pitch view variation like pedestrians, vehicles also varies a lot in different yaw views; (2) vehicle detection is potentially applied in general surveillance scenes of which the camera parameters might be unknown or imprecise.

In order to address the above-mentioned problems, we propose a novel approach based on scene transformation, rather than training a scene specific detector or modifying a pre-trained detector. The system overview can be summarized as follows. First, preliminary camera calibration is performed by exploiting scene information. With the camera parameters, the scene transformation framework is initialized. And then, the spatial–temporal voting guides the procedure of parameter optimization iteratively. Finally, we can obtain the optimal parameters and their scene transformation framework for the pre-trained detector. Accordingly, our system consists of two key components: scene transformation modeling and camera parameter initialization and optimization.
The main contributions of this paper include: (1) Scene transformation is carried out for detector adaptation in general scenes, which is both robust to pitch and yaw view variations (some training and testing samples shown in Fig. 1). (2) Automatic parameter estimation fuses parameter initialization and optimization without manual interactions, which guarantees the best utilization of the pre-trained detector in our transformation framework.

The remainder of the paper is organized as follows. Section 2 will provide the details of scene transformation modeling. And then we will introduce the camera parameter initialization and optimization in Section 3. Experiments are carried out in Section 4 and finally the conclusion is given in the last section.

2. Scene transformation modeling

View transformation in an unknown scene is really a challenging task due to the following reasons: (1) Even in the same scene the view variations of vehicle are different in different positions. (2) In most cases, a vehicle has more than one view variations all of which can be decomposed into pitch and yaw variations. Our objective is to transform the current views of target scenes to the referential views of training scenes which the pre-trained detector can capture.

To address these problems and follow the idea, we propose the scene transformation modeling of which an overview is shown in Fig. 1(d). We actually transform sub-images in searching grids with different parameters. Given searching points, referential views will be selected in training scenes. And then corresponding pitch and yaw view transformation are performed on source images to approximate to the views of training samples. In the next step, the pre-trained detector can work well in the transformed sub-images. Finally, the detection results are projected back to the source images.

Before introducing the scene transformation, we will first list the notations. Given the camera position \( P_s \), at each point of the searching grid, we generate a transformed sub-image \( Q \) from the source image \( I \). \( W \) is their corresponding 3D coordinate, which is also the bridge between \( Q \) and \( I \). Homogeneous coordinates are employed to denote points in \( Q, I \) and \( W \) by \( q = (u_0, v_0, 1)^T \), \( p = (u, v, 1)^T \) and \( P = (x, y, z, 1)^T \). In our problem, their relations can be denoted as

\[
p = HP = HMq = HM_{q_p}, \tag{1}
\]

where \( H \) is the camera matrix, \( H = A[R||T] \). \( M \) is the transformation matrix which contains the pitch view transformation matrix \( M_p \) and yaw view transformation matrix \( M_y \). In our framework, all the transformed sub-images are size-fixed. We enumerate all the pixels of the transformed image, use Eq. (1) to get their corresponding positions of the source image and obtain the color values, finally we can get the transformed image. In the following, we will mainly introduce the pitch view transformation and the yaw view transformation.

**Pitch view transformation** is carried out to approximate the orthogonal projection of samples. With the camera position \( P_s = (x_s, y_s, z_s)^T \), the search grid point \( P_q = (x_0, y_0, 0)^T \) and its desired projected position \( q_p = (u_x, v_x, 1)^T \) in \( Q \), the pitch view transformation matrix \( M_p \) in the search grid can be written as

\[
M_p = \begin{pmatrix}
cos \theta & 0 & -u_x \cos \theta + x_0 / x \\
\sin \theta & 0 & -u_x \sin \theta + y_0 / x \\
0 & -1 & v_0 \\
0 & 0 & 1 / x
\end{pmatrix}, \tag{2}
\]

where \( x \) is the size ratio between the real world object and the transformed image patch size of height (after normalization). In our experiments, we set \( x \) to be 1700 mm / 50 pixel since vehicles can be normalized to about 50 pixels (higher than 1700 mm has more than 50 pixels and vice versa) which facilitate the detection procedure for appropriate search range. Derived from Eq. (1) and Eq. (2), we can obtain the relation between the 3D coordinate and the transformed image coordinate,

\[
P = M_p q' = \begin{pmatrix} x(u' - u_0) \cos \theta + x_0 \\
y(u' - u_0) \sin \theta + y_0 \\
z(v_0 - v') \\
1 \end{pmatrix}.
\]

The scene transformation modeling of which an overview is shown in Fig. 1(d). We actually transform sub-images in searching grids with different parameters. Given searching points, referential views will be selected in training scenes. And then corresponding pitch and yaw view transformation are performed on source images to approximate to the views of training samples. In the next step, the pre-trained detector can work well in the transformed sub-images. Finally, the detection results are projected back to the source images.

Fig. 1. (a) training samples; (b) testing samples; (c) detection in target scenes with large view variations. (green: results of the pre-trained detector; red: results of our approach with the same detector); (d) the scene transformation framework: transform the testing samples’ views to the training samples’ in search grids, then detect vehicles with the pre-trained detector, finally back-project to source images. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
\[ \theta = \arccos \left( \frac{y_u - y_c}{\sqrt{(x_u - x_c)^2 + (y_u - y_c)^2}} \right). \] (3)

**Yaw view transformation** fills the gaps between the current views and training samples’ (Fig. 1(a)) caused by yaw variations through out-of-plane rotation. Therefore, we rotate the transformed sub-image from pitch view transformation out-of-plane based on the angle of difference to the corresponding sample in the training scene (one referential training scene will be selected for the target scene). Projective transformation is employed to get the corresponding point in training scene

\[ p_{ref} = Ep_{new}, \] (4)

where \( p_{ref} \) and \( p_{new} \) are corresponding points in training scene and target scene and \( E \) is the projective matrix which can be recovered by Least Square Method. And then calculate the yaw difference angle \( \Delta \theta \) with Eq. (3). Finally, we can denote the yaw view transformation matrix \( M_y \) as

\[ M_y = \begin{pmatrix} c/\cos \Delta \theta & 0 & -cu_0/\cos \Delta \theta + u_0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \] (5)

where \( c \) is a constant and set to be 1 for rear and 2 for body of vehicle respectively in our experiments. We don’t need to set the width and length of vehicle while \( c \) is a rough proportion but can approximate most types of the vehicles. With Eq. (1) and Eq. (5), we can get

\[ q' = (u', v', 1) \] from \( q = (u, v, 1) \)

\[ q' = M_y q = \begin{pmatrix} c(u - u_0)/\cos \Delta \theta - u_0 \\ v \\ 1 \end{pmatrix}. \]

We out-of-plane rotate the x direction coordinate of \( q \) to get the x of \( q' \) according to \( \Delta \theta \) while keeping y direction coordinate the same.

With a transformed image, if a vehicle is detected in the image, we will project it back to the source image with Eq. (1). Some transformed results will be shown in the experiment section. The performance of our transformation framework depends on the camera parameters of which the estimation method is detailed in the next section.

### 3. Camera parameter initialization and optimization

As for most surveillance videos, land marks are inaccurate and mixed with noises, which conventional camera calibration methods can barely handle with in automatic ways. Besides, the methods have poor error correcting capability when there is a large gap between estimated parameters and the actual ones caused by marks’ inaccuracy and lens distortion. To address these problems, we first perform automatic preliminary camera calibration by exploiting scene information, and then use spatial–temporal voting to guide the online optimization of the camera parameters, with which the pre-trained detector can achieve the best performance (shown in Fig. 2).

#### 3.1. Automatic initialization from scene information

In this section, scene information extractor is devised to exploit useful data and implement preliminary camera calibration automatically. At present, we mainly aim to implement our approach in one direction traffic surveillance. First, an algorithm is proposed to estimate vanishing points in images of traffic environments. Given the points, we can recover the camera parameters. In the following, we will give the details.

**Vanishing points estimation.** In traffic surveillance scenes, the primary semantic objects include roads and buildings. Their extending directions will intersect in vanishing points. Based on these observations, we propose an automatic vanishing point estimation algorithm by extracting the extending directions. As for most typical surveillance scene videos, our approach can extract sufficient lines paralleling to the extending directions (such as the land marks, the edge of buildings and the edges of vehicles paralleling to roads which provide both of horizontal and vertical lines). Therefore, we exploit these lines from multiple key frames. The process can be summed up as follows (one frame sample is shown in Fig. 3).

1. **Pre-processing.** After Canny edge detection, Hough transform is used to recover straight lines on the edge map (Fig. 3(b)). With a constraint that only lines with similar tilt angles can generate vanishing point candidates, 1000 points are randomly generated from the line set (Fig. 3(c)).

2. **Candidate point clustering.** K-means clusters the candidate points into \( pc \) modes through self-adaptation (Fig. 3(d)) based on their positions and tilt angles of the intersected lines \( \{x_u, y_u, \Delta \theta_0\} (x_\) is a constant).

3. **Line clustering and refining.** With these point clusters, we can obtain the mean positions as their cluster centers. And then cluster the lines into \( lc \) modes with their distances to the cluster centers \( \{x_c, y_c, \Delta \theta_0\} \) (Fig. 3(e)). In the refining stage, line clusters belonging to a same point cluster will be merged into a new line cluster (Fig. 3(f)).

4. **Vanishing point selection.** The line clusters in accordance with the Manhattan directions are dense and ordered which are different from the noise clusters, therefore they can be easily selected (Fig. 3(g)). Finally, the points of x, y and z directions with minimal distances to their corresponding line clusters will be selected as the final vanishing points (Fig. 3(h)).

**Preliminary camera calibration.** With vanishing points \( (u_1, v_1), (u_2, v_2), (u_3, v_3) \), we recover intrinsic matrix \( A \) and rotation matrix \( R \) based on He and Li (2008). Since translation vector \( T = [t_x, t_y, t_z]^T \) cannot be calculated from vanishing points, we will estimate \( T \) in the following. We set a point \( p_{\text{origin}} \) at ground plane from target scene as the origin of world coordinate system:

\[ [R\ T]P_{\text{origin}} = A^{-1}p_{\text{origin}}, \] (6)

where \( P_{\text{origin}} = [0, 0, 0, 1]^T \), hence

\[ T = A^{-1}p_{\text{origin}}. \] (7)

The camera location relative to the origin can be represented as:
\[ P_t = -R^T T. \]  

Assuming the height of camera as \( h \) (about 5 m above ground plane in practice), we can obtain \( T \) from the solution of Eq. (7) and Eq. (8).

3.2. Online optimization by spatial–temporal voting

To further improve the robustness and accuracy of our approach, Spatial–Temporal Voting is adopted as feedback to online optimize the preliminary camera parameters. The Spatial–Temporal Voting measures the effectiveness, smoothness and reliability metrics which reflect the utilization performance of the pre-trained detector in the transformation framework. These three metrics are computed based on the tracklets (object trajectories) generated after a pre-trained detector and a tracker run parallelly in the transformation framework of a sub-sequence from target scene.

Specifically, the objective of this section is searching for a set of camera parameters which maximize the S–T voting probability, given the tracklets \( x \)

\[ \omega^* = \arg\max_s (p_{st}(\omega)), \]  

where S–T voting \( p_{st}(\omega) \) contains three metrics: effectiveness, smoothness and reliability terms,

\[ p_{st}(\omega) = p_e(\omega) \times p_x(\omega) \times p_r(\omega). \]  

Effectiveness: this metric measures the detection rate of the pre-trained detector in transformed scenes. It is reasonable with intuition that if the view is more similar to the training samples’ view, then more targets will be detected and tracklets will last longer. So the probability of effectiveness can be formulated as:

\[ p_e(\omega) = C_1 \exp(\lambda_1 T) \times C_2 \exp(\lambda_2 T/|\omega|), \]  

where \( T \) is the number of detected targets and \( T/|\omega| \) is the mean length per tracklet. In order to reduce noises, we remove short tracklets. False alarms will be suppressed by the reliability term.

Smoothness: the smoothness of motion and appearance are fused to measure the consistency of tracklets which reflect the validity of camera parameters from another perspective. The motion vector \( \dot{e}_t \) for the target of kth tracklet in time frame \( t \) we concern is \( \{v_x, v_y, s\} \) where \( (v_x, v_y) \) is the velocity and \( s \) is the scale. We assume the distribution of motion vector as Gaussian process. So the motion probability can be denoted as

\[ L_M = C_3 \sum_{k=1}^{\text{tracklets}} \sum_{t=k+t_k}^{\text{span}} \frac{1}{(2\pi)^{3/2} \det(V_t)^{1/2}} \exp \left( -\frac{1}{2} e_t^T V_t^{-1} e_t \right), \]  

where \( V_t \) is the variance matrix of the motions from \( t_k \) to \( t \). We ignore first two motions for each tracklet because of unknown variance matrix. As for appearance, we use the similarity of gray scale histograms of the same target in consecutive frames to depict the smoothness of its appearance,

\[ L_A = C_4 \sum_{k=1}^{\text{tracklets}} \sum_{t=k+t_k}^{\text{span}} \exp \left( \lambda_1 B(t_k, \tau_k(t-1)) \right), \]  

where \( B(\cdot) \) is Battacharyya distance and \( \tau_k(t) \) is the bounding box of the kth tracklet in time frame \( t \). So the smoothness probability is

\[ p_x(\omega) = L_M \times L_A. \]  

Reliability: this metric encodes how similar the samples after transformation are to the training samples. In our problem, the reliability is derived from the confidence output of the pre-trained vehicle detector. With confidence output \( h_k(\cdot) \), the reliability metric is

\[ p_r(\omega) = C_5 \sum_{k=1}^{\text{tracklets}} \sum_{t=k+t_k}^{\text{span}} \exp \left( \lambda_2 h_k(\tau_k(t)) \right) \]  

With S–T voting to measure parameters, the optimization procedure is tantamount to search for the best state \( s \) of camera parameters, in which the search path is only from it to its neighbor states \( N(s) \), \( A \) is in close proximity to actual value (assuming the optic center is the center of image), while \( T \) can be calculated from Eq. (7) and Eq. (8) given \( A \) and \( R \). Consequently, we can get the neighbor states \( N(s) \) by slightly perturbing \( R \) and then calculating the corresponding \( T \). We can get the new rotation matrix through rotating world coordinate system in 3D space with \( V \) in advance:

\[ R' = R \times V \]  

where the 3D rotation matrix \( V \) can be denoted as (angle \( \alpha, \beta, \gamma \) in \( x, y \) and \( z \) directions respectively)

\[ V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \]  

In our experiments, \( R' \) of each neighbor state only has slight rotations (different step sizes) in one direction for covering the range of variations since the rotation in 3D is superimposed.

Gibbs Sampler (Kamijo et al., 2000) is employed to perform iterative parameter optimization. It evaluates the S–T voting of the neighbor states using Eq. (10), and then determines which state
to transit to (with larger S–T voting). The iterative optimization is carried on until there is no state transition in a loop process. For clarity, we summarize the procedure of online parameter optimization in Algorithm 1 of our Supplementary material.

4. Experiments

Experiments are carried out on five video sequences collected from traffic surveillance camera and some real-world video data collected with a hand-held camera. These sequences contain 2120 frames from which 15083 ground truth are manually labeled for evaluation. In the following, we first give a description of the pre-trained detector as well as the implementation details. And then we will introduce the baselines. Finally, the experimental results are provided and the corresponding discussions are made.

4.1. Pre-trained detector

The offline vehicle detector is trained in the boosting framework using the Joint Sparse Granular Features (JSGF) (Ai et al., 2007) which has been proven to be effective for vehicle detection (Liu et al., 2012). The positive samples are labeled from several scenes with different weathers such as raining, snowing and shadow, in the hope of handling variations of illuminations. The negative samples are randomly extracted from the images with highways, cross-roads and buildings of which the vehicles are removed manually.

4.2. Implementation details

In the optimization stage, the sub-sequences for our parameter optimization contain 100 frames to 200 frames in which the vehicle motion regions cover most of the road planes. Therefore, we can guarantee that our result makes sense for the whole target scene rather than local minimum caused by lens distortion and achieve optimal result or suboptimal maximal result. Each type of V used to get neighbor states has plus–minus and different rotation angles (0.25°, 0.5°, 1°, 2°, 5°).

4.3. Baselines and our approach

In our experiments, we compare the performance of our approach with four baseline methods. The first baseline (Pre-trained Detector) uses the pre-trained detector to detect vehicle in the target scene without view transformation. The second one (Retrained Detector) completely retrains the detector with fully labeled data directly from the target scenes. Since the training data and the testing data are from the same types of video data, the performance of a retrained detector actually is the ultimate goal for a general detector to be adapted to the target scenes. The third one (Scene Transformation) incorporates the pre-trained detector with scene transformation, but without the stage of online parameter optimization. The fourth one (Li et al., 2008) is the approach for pedestrian 3D search framework implemented by ourselves with the same optimized parameters and the same detector as our approach. To make more fare comparisons, all the above detectors are offline trained in the same algorithm (Adaboost) with the same feature (JSGF).

4.4. Results

Experiments are run on the vehicle sequences from target scenes (15083 vehicles in 2120 frames). Some transformed results and detection responses from the same pre-trained detector are shown in Fig. 4. From the figure, we can see that our approach achieves view transformation from different views (1st row) to frontal/rear view (2nd row), through which the transformed vehicles (4th row) have much stronger detection responses than the

Fig. 4. Transformed results and detection responses (1st row: cropped source images; 2nd row: transformed results; 3rd row: detection responses of source vehicles; 4th: detection responses of transformed vehicles. A green box is a detection response.) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
source ones (3rd row). Even in some cases, the source views have no detection responses, but the pre-trained detector can quite catch the transformed views, which demonstrates the effectiveness of our transformation framework.

The ROC curves and some comparison results are given in Fig. 5. Compared with the first baseline (Pre-trained Detector) which uses the same detector but without view transformation (top left in Fig. 5), our performance is much better than its. This comparison accounts for that the detection responses of transformed images are much stronger than the conventional 2D search when the pre-trained detector loses effectiveness, because our scene transformation framework is capable of detecting vehicles with different views. Actually, the performance of the retrained detectors is the ultimate goal for our approach since the testing and training samples of the retrained detector are similar. From the ROC curves at top right in Fig. 5, we can see that our approach is comparable with the second baseline (Retrained Detector). Our detection rate is 87.27% compared to 85.61% of the retained detectors for

![ROC curves and some comparison results](image)

**Fig. 5.** Performance of our approach compared with baselines (ROC curves shown at the top and some comparison results at the bottom).

![Some typical comparison results](image)

**Fig. 6.** Some typical comparison results (the first column: the pre-trained detector without transformation; the second column: retrained detectors with fully labeled data; the third column: our approach with optimized parameters).
practical application (FPPI = 0.1). Because our approach can handle some types of partial occlusions while the second baseline method cannot just as shown at bottom right in Fig. 5. We attribute this to our scene transformation framework, since the occluded vehicle will have the best transformation in its search grid but the others won’t. Our processing time is 200 ms–400 ms for full detection of a 640 × 360 image while the retrained detectors’ are 180 ms–250 ms. In contrast to the third baseline (Scene Transformation), our approach benefits from the stage of online parameter optimization. The optimization stage is designed for the cases in which the estimations of camera parameters from vanishing points are inaccurate. With optimized camera parameters, our approach achieves higher performance than the third baseline. Focusing on vehicle transformation, our approach outperforms the fourth baseline (Li et al., 2008), and the detailed reasons will be discuss in the following. More typical comparison results are shown in Fig. 6.

For the clarity of comparison, we compare our approach with Li et al. (2008) in Table 1, which can facilitate the discussion of our scene transformation framework for vehicle. First, Li et al. (2008) requires camera parameters which are unknown in most surveillance videos. In order to obtain fair comparison results, we parameterize their 3D search framework with our optimized camera parameters and detect vehicle in the transformed images with the same detectors. From Table 1, we can see that our approach improve the detection rate by 15% than Li et al. (2008) in average, and both of them take huge steps from baseline 1 (Pre-trained Detector). The comparison results demonstrate that view transformation is effective in general scenarios. Furthermore, vehicle has more difficulties than pedestrian should be handled with: beside the pitch view variation like pedestrians, vehicles also vary a lot in different yaw views, which we consider in our framework. Therefore, the effectiveness of our approach for vehicle detector adaptation in general scenarios is demonstrated.

Table 1

| Comparison with the state-of-the-art approaches. Bold values are the best performance. |
|------------------|----------------|----------------|----------------|----------------|
|                  | FPPI           | Ours           | Li et al. (2008) | Pre-trained Detector |
|                  | 0.05(%)        | 79.09          | 54.60           | 42.27           |
|                  | 0.1(%)         | 87.27          | 68.45           | 58.07           |
|                  | 0.15(%)        | 90.00          | 76.00           | 65.91           |
|                  | 0.2(%)         | 91.82          | 78.33           | 69.55           |

5. Conclusions and discussions

To handle with the challenging problem of view variation for pre-trained detectors, we propose a novel vehicle detection adaptation approach based on scene transformation. Our approach not only significantly improves the performance of pre-trained detectors through scene transformation framework, but also implements automatic camera parameter estimation without manual interactions. Extensive experiments and comparisons demonstrate the effectiveness and robustness of our approach.

Although our approach achieves much better performance than pre-trained detectors and is comparable to the detectors retrained on fully labeled data from target scenes, it can barely handle with large view variations. Our future work will apply our approach to more scenes (such as crossroads) to put it into practical applications. Besides, we will generalize the algorithm to the body view of vehicle so that we can handle larger range of view variations.

Acknowledgements

This work is supported in part by National Science Foundation of China under Grant No.61075026, National Basic Research Program of China under Grant No.2011CB302203.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.patrec.2013.10.004.

References