Investigating 3D Model and Part Information For Improving Content-based Vehicle Retrieval

Yen-Liang Lin, Ming-Kuang Tsai, Winston H. Hsu, Chih-Wei Chen

***Abstract*—Content-based vehicle retrieval in unconstrained environment plays an important role in surveillance systems. However, due to large variations in viewing angle/position, illumination, occlusion, and background, traditional vehicle retrieval is extremely challenging. We approach this problem in a different way by rectifying vehicles from disparate views into the same reference view and searching the vehicles based on informative parts such as grille, lamp, and wheel. To extract those parts, we fit 3D vehicle models to a 2D image using active shape model (ASM). In the experiments, we compare different 3D model fitting approaches and verify that the impact of part rectification on the content-based vehicle retrieval performance is significant. We propose a model fitting approach with weighted jacobian system which leverages the prior knowledge of part information and shows promising improvements. Then, we use pyramid histogram of orientation (PHOG) feature to describe rectified parts (e.g., grille, wheel, lamp). We compute mean average precision of vehicle retrieval with L1 distance on *NetCarShow300* dataset, a new challenging dataset we construct. We conclude that it benefits more from the fusion of informative rectified parts (e.g., grille, lamp, wheel) than a whole vehicle image described by SIFT feature for content-based vehicle retrieval.**

***Index Terms*—3D model construction, 3D model fitting, vehicle retrieval, part rectification**

I. Introduction

VEHICLES are one of the most important subjects in surveillance environment when surveillance cameras become ubiquitous and more and more surveillance video data is available. However, millions of surveillance videos are so large-scaled that it is impossible for human to deal with. Therefore, effective vehicle retrieval is becoming increasingly significant. Moreover, people may be interested in some scenarios, for example, "Where and when did white vehicles pass through here yesterday?," "Can I find hatchbacks in these videos?," "Can I find vehicles made by Honda or whose models are BMW X6 in these videos?," and more possible descriptions about vehicles. As a result, we propose an effective content-based vehicle retrieval approach to satisfy the needs. See Figure 1.

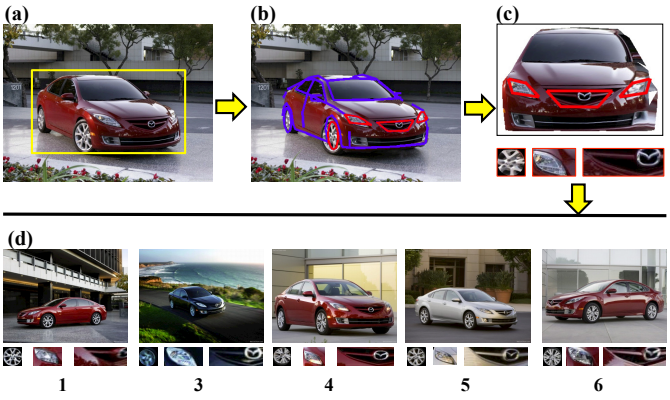


Fig. 1. An overview of the proposed system. (a) Input image with bounding box. (b) Aligning 3D model to 2D image. (c) Rectifying vehicles to the same reference view points and extracting vehicle parts (e.g., grille, wheel, and lamp). (d) Top 5 searching results by fusing three parts. Best seen in color.

There is no doubt that vehicle retrieval and recognition are very challenging. For surveillance videos with time information or surveillance images, there are several problems which cannot be ignored. Besides reflective surface and semi-transparent media, shadow or illumination possibly makes edge detectors or feature detectors incur more noises. Occlusions in crowded scenarios are common, and occluded parts cause discontinuity and incompleteness in shape. Also, background noise and shape variations are potential difficulties when detecting or segmenting vehicles. What is more, vehicles are observed from a variety of viewpoints, so it is hard to retrieve similar vehicles without constructing correspondence.

To address diversity of viewpoints and shape variation, it comes to our minds that using rectified parts extracted from fitted 3D vehicle models are more promising than a whole image for content-based vehicle retrieval. In other domains, such as face [1] and people [2] search, salient attributes or parts have been utilized to identify targets. However, the utilization of parts for vehicles has not achieved similar successes. In addition, unlike people or face recognition, using 2D models to extract parts of a vehicle within the bounding box generally fails due to dramatic variations in viewing angles. Employing 3D models is more suitable for our work. In fact, using 3D vehicle model is one of the major line of research in the fields of vehicle detection [3], pose estimation [4][5], classification [6][7][8][9], etc.

This work was supported in part by grants from the National Science Council of Taiwan, under Contracts NSC 100-2631-H-002-003, and Industrial Technology Research Institute (ITRI), Taiwan.

Yen-Liang Lin is with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei City 10617, Taiwan (e-mail: d98944010@ntu.edu.tw).

Ming-Kuang Tsai is with Garmin Corporation, New Taipei City 221, Taiwan (e-mail: mingkuang.tsai@gmail.com).

W. H. Hsu is with the Graduate Institute of Networking and Multimedia and the Department of Computer Science and Information Engineering, National Taiwan University, Taipei City 10617, Taiwan (e-mail: winston@csie.ntu.edu.tw).

Chih-Wei Chen is with Industrial Technology Research Institute, Hsin-Chu City 31004, Taiwan (e-mail: piny@itri.org.tw).

We propose to augment content-based vehicle retrieval by aligned 3D vehicle model and fusing informative parts (cf. Figure 1). First, we establish consistent shape representation between several 3D vehicle models and align 3D model to natural images (cf. Figure 1(b)). Second, informative parts (e.g., grille, lamp, and wheel) are extracted and rectified into one reference view, and the parts are represented by several features for retrieval (cf. Figure 1(c)). Third, we evaluate our approach under different viewpoints, illuminations, and situations. The result shows that we improve the retrieval performance significantly even in diversity of viewpoints.

The main contributions of this work include:

- We implement and compare current state-of-the-art 3D model fitting algorithms and evaluate on a challenging dataset.
- We argue to improve 3D model fitting precision by leveraging the prior knowledge of those informative parts.
- We investigate the impacts of rectified parts on the content-based retrieval performance.
- To our best knowledge, this is the first content-based vehicle retrieval approach that uses informative parts and analyzes the detailed parameterizing components for the framework.

II. Related Works

Vehicles have been a subject of interest. Most approaches or systems either detect vehicles from background or classify vehicle types such as cars, buses, trucks. Some people further use 3D models to improve their works. Arie-Nachmison et al. [4] construct a general 3D implicit shape model by factorization, and they apply RANSAC procedure to estimate vehicle poses. [6] embeds rendered 3D vehicle prototype to deal with vehicle classification. [5] builds 3D representations of vehicle classes to handle viewpoint changes. In the past, simple 3D polyhedral vehicle models have been used for vehicle recognition based on the assumption that matching the edges in the images with the edges of the polyhedron is sufficient [7][8]. However, it is clear that few edges of these rough models, even in low-resolution images, are limited in gaining acceptable accuracy. To address the lack of details in simple polyhedral models, some approaches adopt more delicate 3D vehicle models which provide rich constraints to match vehicles reliably. In [9], based on a set of labeled salient points on the detailed 3D model and metadata information, HOG features are extracted and compared between the rendered model and the real scene to classify vehicle types. Leotta et al. [10] and Tsin et al. [11] use 3D CAD models and refine 3D-to-2D alignment until convergence.

Vehicle make and model recognition or content-based vehicle retrieval is a relatively new research problem. The basic idea is to extract suitable features from the images of a vehicle, which can be used to not only retrieve vehicle images having similar appearances but also retrieve its make and model.

The objective of content-based image retrieval (CBIR) is to efficiently analyze the contents of the image and retrieve similar images from the database when metadata such as keywords and tags are insufficient. To bridge the semantic gaps, how to efficiently use available features such as color, texture, interest points of images and spatial information is the key. VisualSEEk system [12] developed a joint color/spatial images query strategy. To acquire region-based signature for retrieval, Malik et al. [13] applied image segmentation by using cues of color and texture. Scale and affine-invariant interest points have been used to deal with significant affine transforms and illumination changes and shown effective for image retrieval [14]. To improve CBIR performance, other works provide effective indexing, refine results with feedback from the user, select more powerful features from a pool of features, etc.

Searching for vehicles in surveillance videos, Feris et al. [15] build a surveillance system capable of vehicle retrieval based on semantic attributes. They train motionlet detectors which cover 12 viewpoints and are learned in shape-free appearance space from a set of city surveillance cameras. They define several attributes as possible descriptions, such as dominant color, length, height, and direction. Those attributes are extracted from detected motion blobs. To estimate vehicle dimensions in world coordinates, they manually do camera calibration and use a simple 3D model fitting approach on the basis of several assumptions (i.e., a vehicle's location on the ground plane, orientation of heading direction, and the scale of the model). Different from this system based on attributes and applied on surveillance videos, our work focuses on extracting informative parts from fitted vehicle models and investigating part information for improving content-based vehicle image retrieval.

Vehicle make and model recognition (MMR) is mostly applied on frontal vehicle images. Petrovic and Cootes [16] present an investigation in a rigid structure approach for vehicle make and model recognition. They show that gradient representations are reliable for recognition of vehicles from frontal views under a variety of conditions. Similarly, Negri et al. [17] propose an oriented-contour points based voting algorithm. They use LPREditors license plate recognition method to get the corners of the vehicle license plate. They empirically regard the area extended from the corners as the region of interest and do vehicle type classification based on oriented contours. Instead of using edge maps as features, Kazemi et al. [18] apply curvelet transforms to represent image edges and curved singularities. Rahati et al. [19] show contourlet transforms are more suitable with smooth contours in all directions. Zafar et al. [20] propose to use additional local texture in the contourlet decomposition across scales of directional resolutions and get increased accuracy on vehicle make and model recognition. They have shown that structural features are distinguishable on frontal vehicle images.

Different from those prior works, we conduct content-based retrieval on vehicle images using more rigorous 3D model fitting methods to deal with a variety of viewpoints. Our approach is applicable not only in absolutely frontal view but also in a wider range than previous vehicle make and model recognition. Moreover, we present an investigation for content-based vehicle retrieval based on several semantic parts and their fusion. The experimental results show that vehicle make and model retrieval based on those informative parts

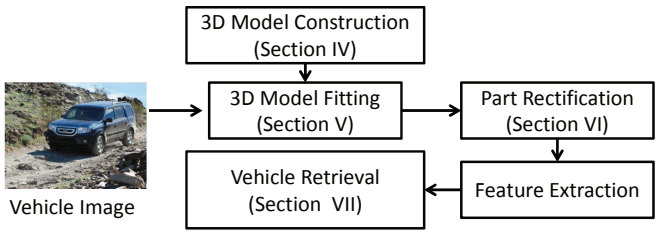


Fig. 2. Our framework of content-based vehicle retrieval. In the offline process, we use 3D vehicle models to build an active shape model (ASM). Then we are able to do 3D model fitting on the input vehicle image. We crop and rectify informative parts into the same reference view. After feature extraction, we conduct part-based vehicle retrieval on NetCarShow300, a challenging dataset.

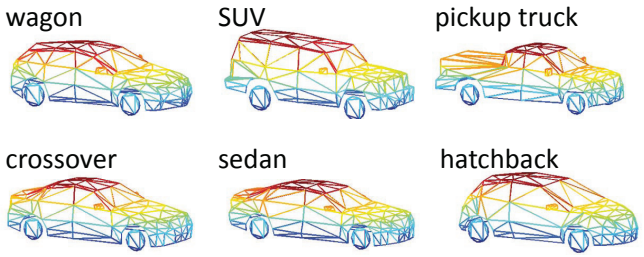


Fig. 3. 3D vehicle training models, including sedan, wagon, pickup truck, crossover, hatchback, and SUV.

are more representative than a whole vehicle, and it is also possible to enhance more information or attributes for further improvement.

III. Overview

Our approach focuses on improving content-based vehicle retrieval performance. In order to deal with a variety of viewpoints and shapes, we propose to analyse the vehicle in an image by 3D vehicle model fitting and rectified parts. With prior part information, we can further enhance model fitting results.

The overall framework is as illustrated in Figure 2. First, to deal with shape variation of models, we want to build an active shape model (ASM). Therefore, we use a CAD tool to construct 3D vehicle models, and we can enforce point consistency between all models manually. Second, we investigate different 3D model fitting methods (e.g., fitting by point registration, jacobian system, weighted jacobian system). We are able to get the fitted vehicle in an image and reconstruct its 3D vehicle models after the model fitting step. Third, we rectify informative parts into the same reference view. There is no doubt that recognizing vehicles or parts in different viewpoints is difficult. As a result, we apply image warping to informative parts, flip them into the same side, and remove partial distorted regions. Finally, we extract features from those rectified parts and conduct detailed content-based vehicle retrieval experiments. Moreover, we also compare state-of-the-art 3D model fitting approaches and our improved method involving part information.

The rest of this paper is organized as follows. 3D model construction and fitting are defined in Section IV and Section V. Part rectification method is described in Section VI. Experiment results and comparisons are provided in Section VII, followed by our conclusion and future work in Section IX.

IV. 3D Vehicle Model Construction

Considering shape variation of vehicles, we have to build a deformable 3D vehicle model before 3D model fitting process. As a result, we construct an active shape model (ASM) for vehicles. An ASM represents an object by a set of 3D points $\mathbf{P}^k = [\mathbf{p}_1^k, \mathbf{p}_2^k, \cdots, \mathbf{p}_N^k]^T$. A 3D point $\mathbf{p}_j$ in each instance represents the same corner or semantic location, such as a left-top corner of a windshield, a right-front corner of a roof, or one of points consisting of a front wheel. To make sure the correspondence of the same physical shape, we manually select 128 points for a half vehicle template model. The other half can be obtained by mirroring. Then, we adjust locations of points from the template model to corresponding locations in each training models. It is inevitable that there exists difference on rotation, translation, and scale factors between 3D vehicle models. To eliminate the influences of these factors and only analyze shape variation by principal component analysis (PCA), we conduct generalized Procrustes analysis (GPA) [21]. GPA is an approach to align shapes of each instance. The algorithm is outlined as following.

1) Choose a reference shape or compute mean shape from all instances.
2) Superimpose all instances to current reference shape.
3) Compute mean shape of these superimposed shapes.
4) Compute Procrustes distance between the mean shape and the reference. If the distance is above a threshold, set reference to mean shape and continue to step 2.

After finishing the GPA process, we apply PCA. That is, we compute the mean shape m from K training vehicle models:

$$\mathbf{m} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{P}^k. \tag{1}$$

and we can compute the matrix $C = \sum_{k=1}^{K}(\mathbf{P}^k - \mathbf{m})^T(\mathbf{P}^k - \mathbf{m})$ and the eigenvectors $\mathbf{\Omega} = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ..., \boldsymbol{\omega}_M]$ which correspond to the M largest eigenvalues of C and define the "vehicle space." By projecting a 3D vehicle model to the "vehicle space," we get a M-dimensional vehicle shape parameter $\mathbf{U}$ which controls the variability of the shape of a vehicle model. Then, we can reconstruct a 3D vehicle model $\mathbf{P}'$:

$$\mathbf{P}' \approx \mathbf{m} + \mathbf{\Omega} \cdot \mathbf{U}. \tag{2}$$

In our experiment, we use 11 3D vehicle models as training instances, including 3 sedans, 2 wagons, 1 pickup truck, 1 crossover, 2 hatchbacks, and 2 SUVs (cf. Figure 3), with totally 256 salient points for two sides and 342 triangular faces to describe wheels, radiator grille, doors, lamps, windows, rear, and other semantic parts (cf. Figure 4).

Then, we estimate the reconstruction error. According to the mean reconstruction error estimated by the ratio between average distance error and the vehicle length, we find that less than 0.4% reconstruction error results from 8 eigenvectors; that is, the error is only about 4 pixels if the length of a

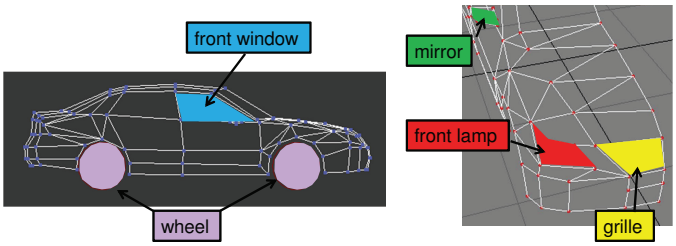


Fig. 4. 3D vehicle model representation. Several vertices form one or more faces which represent a semantic part, such as wheel, front window, or grille.

projected vehicle is 1000 pixels, which is relatively low. It can be expected that no more than 10 eigenvectors is enough, even there are more models. In our experiment, we use 10 eigenvectors to reconstruct the models for better quality.

V. 3D Vehicle Model Fitting Approach

In the model fitting step, we assume that initial position and pose of a vehicle in an image can be estimated. It is reasonable since detecting the direction and location of the moving vehicles or some multi-view object detection can be derived by many promising approaches (e.g., [4][3]). Content-based vehicle retrieval is based on the information about the target vehicle. That is, we must detect and estimate the pose of our target before we deal with this object.

In order to extract parts of vehicles, model fitting is essential. Therefore, we investigate and compare two different state-of-the-art approaches in [10] and [11]. One depends on point registration and the other solves a Jacobian system. The two approaches have not been compared before our work, so we try to implement them and do several sensitivity tests to see their capabilities in different configurations. Moreover, we propose to leverage the prior knowledge of semantic parts (e.g., grille, lamp, and wheel) and further improve the challenging 3D alignment problem. The results are shown in Section VII-D.

The overall model fitting steps are illustrated in Figure 6. Given initial pose and position parameters, the 3D model is projected into 2D image and the hidden lines are removed by using depth map rendered from the 3D mesh. Then, a set of hypothetic edges is generated. For each projected edge point, we find all corresponding observed edge pixels in a range of 20 pixels along the normal direction of the projected edges. After iterative updates for the correspondences, the shape and pose will converge. Both state-of-the-art methods attempt to minimize the distance error between the observed and projected edge points. The illustration of 3D model fitting process is depicted in Figure 5. The main difference between the methods lie in finding point correspondences and shape and pose optimization stages. In [11], they apply point registration (PR) to find more correct point correspondences and use least square approach to iteratively optimize the shape and pose parameters until convergence. In [10], they iteratively optimize shape and pose parameters by solving a Jacobian system.

A. Model Fitting Methods

1) Fitting by Point Registration: Given initial orientation and location of a vehicle in an image, we want to build a 3D vehicle model with our deformable model by fitting the landmarks to the corresponding salient edge points in the image. Here we apply point registration (PR) algorithms to find corresponding points, solve equations, and obtain projected weights and translations as [11].

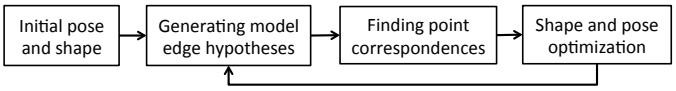
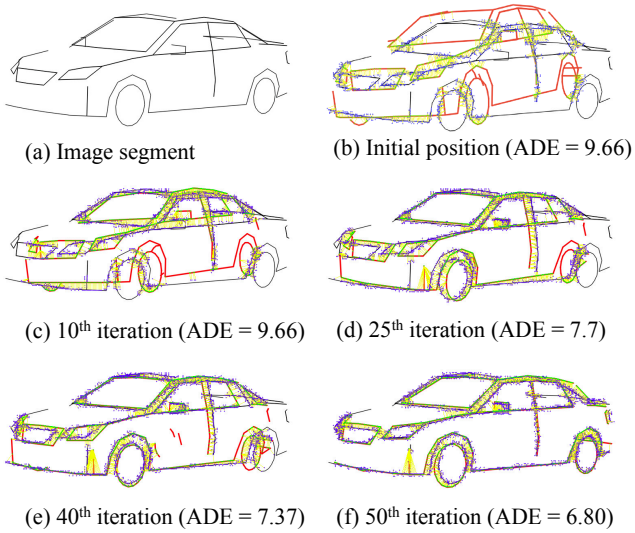


Fig. 5. 3D vehicle model fitting procedure. Given the initial pose and shape, we generate the edge hypotheses by projecting the 3D model into a 2D image and remove hidden lines by using depth map rendered from the 3D mesh. For each projected edge point, the corresponding points are found along the normal direction of the projected edges. Then, a 3D model fitting method is performed to optimized pose and shape parameters. The above procedure is repeated several times until convergence.



(a) Image segment (b) Initial position (ADE = 9.66)

(c) 10th iteration (ADE = 9.66) (d) 25th iteration (ADE = 7.7)

(e) 40th iteration (ADE = 7.37) (f) 50th iteration (ADE = 6.80)

Fig. 6. Illustration of 3D vehicle model fitting process. Given an image segment (a), we iteratively optimize the shape and pose parameters to minimize the distance error between the observed and projected edge points (b) $\sim$ (f). Red line segments are hypothetic edges of current vehicle pose. Green points are the points on the hypothetic edges and blue points are the corresponding points on the observed edges. Each correspondence is linked by a yellow line which represents the error measurement. The average distance error (ADE) under each image indicates the average (pixel) distances between the observed and projected edge points. Best seen in color.

In the beginning, given an initial pose, each landmark point is reconstructed from the mean shape Eq. 1 and projected according to the general camera equation Eq. 3.

$$\begin{bmatrix} u_{(j)} \\ v_{(j)} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & u_0 \\ 0 & s_y f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{T}_{3\times 1} \end{bmatrix} \mathbf{p}_j. \quad (3)$$

$u_{(j)}$ and $v_{(j)}$ represent one 2D point with respect to an origin in the top left corner of the image plane, and $\mathbf{p}_j^k$ is the corresponding 3D point reconstructed according to Eq. 2. f is the focal length or a constant. s_x and s_y are scale factors on x and y axes respectively. $\mathbf{R}$ is a rotation matrix and $\mathbf{T}$ is a translation matrix. Eq. 3 can be reformulated:

$$\begin{cases} \widetilde{m}_{(j)x} - \widetilde{u}_{(j)}\widetilde{m}_{(j)z} = (\widetilde{u}_{(j)}\widetilde{\omega}_{(j)z} - \widetilde{\omega}_{(j)x})\mathbf{U} + [-1, 0, \widetilde{u}_{(j)}]\mathbf{T} \\ \widetilde{m}_{(j)y} - \widetilde{v}_{(j)}\widetilde{m}_{(j)z} = (\widetilde{v}_{(j)}\widetilde{\omega}_{(j)z} - \widetilde{\omega}_{(j)y})\mathbf{U} + [-1, 0, \widetilde{v}_{(j)}]\mathbf{T}, \end{cases} \quad (4)$$

$$\begin{aligned} \text{where} \quad \widetilde{\mathbf{m}}_{(j)} &= \mathbf{R}\mathbf{m}_{(j)} = (\widetilde{m}_{(j)x}, \widetilde{m}_{(j)y}, \widetilde{m}_{(j)z}), \\ \widetilde{\boldsymbol{\omega}}_{(j)} &= \mathbf{R}\boldsymbol{\omega}_{(j)} = (\widetilde{\omega}_{(j)x}, \widetilde{\omega}_{(j)y}, \widetilde{\omega}_{(j)z}), \\ \widetilde{u}_{(j)} &= \frac{u_{(j)} - u_0}{s_x f}, \widetilde{v}_{(j)} = \frac{v_{(j)} - v_0}{s_y f}. \end{aligned} \tag{5}$$

f U represents the vehicle shape parameter as the definition in Eq. 2, $\widetilde{\mathbf{m}}_{(j)}$ and $\widetilde{\boldsymbol{\omega}}_{(j)}$ are rotated mean model and eigenvectors respectively. $\widetilde{u}_{(j)}$ and $\widetilde{v}_{(j)}$ are adjusted coordinates from observed points in an image. When we assume only $\mathbf{U}$ and $\mathbf{T}$ are unknown, Eq. 4 is a least square equation [11]. As a result, the following is the steps for solving this problem. First of all, according to initial position and pose, we project 3D vehicle model to a 2D image and sample points on edges as a projected point set. Second, for each projected salient edge point, we find all nearby points in the normal direction as a candidate point set. Third, we apply a point registration approach, Kernel Correlation (KC) [22] or Coherent Point Drift (CPD) [23]. The step finds a rigid or non-rigid transformation which maximizes the correlated distribution between two point sets. The transformation is applied to the projected point set and outputs possible corresponding points. Finally, the model fitting problem is now formulated as a least square problem based on the correspondence and other known factors, and it can be solved by repeating the steps until it converges.

2) Model Fitting by Jacobian System: As mentioned earlier, we assume that there are initial parameters (e.g., position, pose). Given a collection of correspondences between observed and projected edges, each corresponding edge point produces one error measurement e_i.

$$e_i = E_i(\mathbf{q}) = E_i^1(E_i^2(E_i^3(\mathbf{q}))). \tag{6}$$

Each error measurement e_i is defined by an error function E_i which can be expressed by 3 subsequent functions. E_i^3 generates the ith 3D points from the 3D model and parameters, $\mathbf{q}$. E_i^2 projects the ith 3D points into 2D image coordinates. E_i^1 computes the perpendicular signed distance from the correspondence. In a word, e_i represents the signed perpendicular distance from ith projected point to the corresponding observed edge point (cf. Figure 7(c)). See more explantions in [10]. The fitting problem can be formulated as a Jacobian system (JS):

$$\mathbf{J}\Delta\mathbf{q} = \mathbf{e}, \tag{7}$$

where $\mathbf{e}$ is the vector of signed errors, $\Delta\mathbf{q}$ is the vector of parameter displacement updated at each iteration, and $\mathbf{J}$ is the Jacobian matrix with current parameters. The solution is derived by a least square method and iteratively optimizing the parameters until convergence.

B. Model Fitting with Part Information

d Most of the model-fitting algorithms find corresponding points by local search of the projected edges depending only on some low-level features, such as edge intensity and edge orientation, which are likely to fail and converge to local maxima in common cases due to cluttered background or complex edges on the surface of vehicles. We are interested in whether it is possible to improve the fitting algorithm with some prior knowledge of parts (e.g., grille, lamp, wheel). That is, we can give different weights to different correspondences and lead to better fitting results. To validate our assumption, we generate synthetic weight maps of parts by using annotated ground truth data (cf. Figure 7(b)), and formulate this problem into a weighted Jacobian system (WJS):

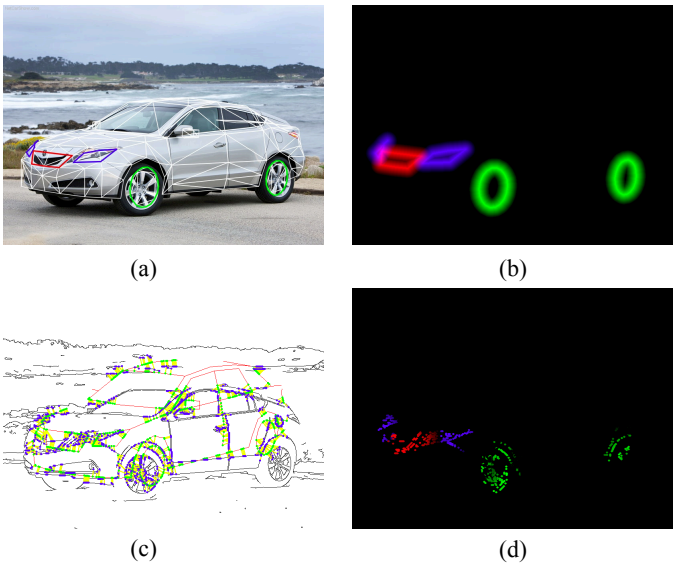


Fig. 7. Illustration of 3D model fitting process with part information. (a) The input image with superjacent 3D ground truth data. (b) The synthetic weight map of three parts, grille, lamp and wheel drawn in different colors. For each part, the color strength represents strength of the weights. (c) Intermediate result in 3D model fitting process. (d) Intermediate weight value of each observed points. Higher weight values imply higher probability of the observed point belonging to the correct part. Best seen in color.

$$\mathbf{W}\mathbf{J}\Delta\mathbf{q} = \mathbf{W}\mathbf{e}, \tag{8}$$

where $\mathbf{W}$ is a diagonal weight matrix with each diagonal element w_{ii} representing the weight of each correspondence. We take two important weights into consideration, distance weight w_{dist} and part weight w_{part}. w_{ii} is computed by a linear combination with λ:

$$w_{ii} = \lambda \cdot w_{dist} + (1 - \lambda) \cdot w_{part}. \tag{9}$$

The distance weight w_{dist} is based on the Beaton-Tukey biweight [24]. For each projected point, the edges far from the point will not be taken into computation. The part weight w_{part} is determined by the value of the location of observed edge point in the part weight map. Higher weight values in the part weight map imply where the part is with higher probability (cf. Figure 7(d)). In other words, we know that the projected point belongs to which part in the 3D model, and the part weight will be higher if the observed edge point belongs to the correct part or near the location of the correct part according to the part weight map. Our experiments show that the 3D model fitting precision is improved with the aid of the prior weight map (cf. Section VII-D).

VI. Part Rectification

Depending on the estimated pose of each vehicle (cf. Figure 1(b)), we extract the parts after the state-of-the-art 3D model

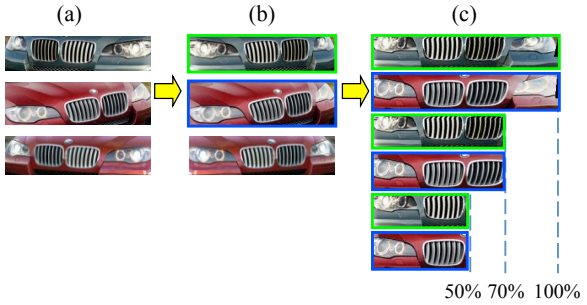


Fig. 8. Illustration of part rectification. (a) The original extracted frontal regions after 3D fitting. (b) The frontal parts flipped to the same side. (c) The flipped parts rectified to specific pose and retaining 70% and 50% width respectively.

Fig. 9. Two warping approaches. (a) An input vehicle image. (b) For some specific grille and lamp meshes, applying warping by barycentric coordinates. (c) Computing a projective matrix by solving a least square equation and applying the matrix to warp the vehicle image. (The license plate is whiten for the privacy issue.)

fitting approach. Next, we rectify the parts by projecting them into specific angles, such as the frontal view or the side view, before feature extraction and feature comparison. We try to use two image warping methods, that is, barycentric coordinates (cf. Figure 9(b)) and projective matrix (cf. Figure 9(c)). We find that applying one global projective matrix is not suitable to rectify a vehicle from one view to the other. So we adopt barycentric coordinates to do image warping in our work.

Barycentric coordinates are triples of numbers corresponding to masses placed at the vertices of a reference triangle. That is, a triangle can be defined by three vertices $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$. A point $\mathbf{t}$ in this triangle is uniquely represented as

$$\mathbf{t} = \alpha \cdot \mathbf{a} + \beta \cdot \mathbf{b} + \gamma \cdot \mathbf{c}, \tag{10}$$

where $\alpha + \beta + \gamma = 1$. If we get the barycentric coordinates of $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$, it is able to calculate α, β, and γ. Hence, by bilinear interpolation and inverted mapping, each point in the projected view can find the corresponding point in the original image and get the mapped pixel value. Furthermore, we can remove background pixels which are out of projected triangles. By this way, we can obtain rectified semantic parts. (cf. Figure 8(a))

Due to viewpoint variation, parts may be contrary in different vehicle images. Therefore, utilizing symmetry of vehicle shape, we flip the visible parts into the same side before applying feature extractionx (cf. Figure 8(b)). The step turns out to be essential for good performance. Moreover, the visible parts are under different poses, so there are some distorted regions after rectification. In other words, warping may enlarge originally small regions and cause distortion. As a result, trimming the non-informative regions is shown effective to improve the performance. Empirically, we retain 70% and 50% ratio of width to investigate the influence. (cf. Figure 8(c))

Fig. 10. Some examples in *NetCarShow300* dataset.

VII. Experiments

In the following, we conduct several experiments on a challenging dataset to show the performance of our content-based vehicle retrieval approach. We also do the comparison between different 3D model fitting methods. It is obvious that fitting precision may influence the part extraction. To prove our idea and remove uncertain factors in content-based vehicle retrieval for leveraging informative parts, our retrieval experiments are initially based on extracted parts from the ground truth. We will further show the retrieval results based on those parts extracted by model fitting in Section VII-D. Similarly, in order to evaluate the influence of knowledge of part information for 3D vehicle model fitting, weight maps are generated from the ground truth.

A. NetCarShow300 Dataset

To investigate our approach on retrieving vehicle images under different conditions and in various viewpoints, we collect 300 images from *NetCarShow.com*[1], the *NetCarShow300*[2] dataset, where the size is comparable to commonly used vehicle type recognition datasets [17][20] which are composed of only frontal cropped grayscale vehicle images. *NetCarShow300* dataset comprises 30 vehicle instances, such as *Acura ZDX*, *Honda Odyssey*, *Honda Pilot*, *Opel Corsa*, *Volvo V70*. Each instance has 10 images respectively. All images are 800×600 color images. Each image contains one main vehicle of which the frontal part is visible. The vehicles are presented in different environments, including noisy background, little occlusion, different illumination, and shadows. Moreover, a vehicle may be extremely projective, and the surface has

[1] http://www.netcarshow.com
[2] We will make the dataset public.

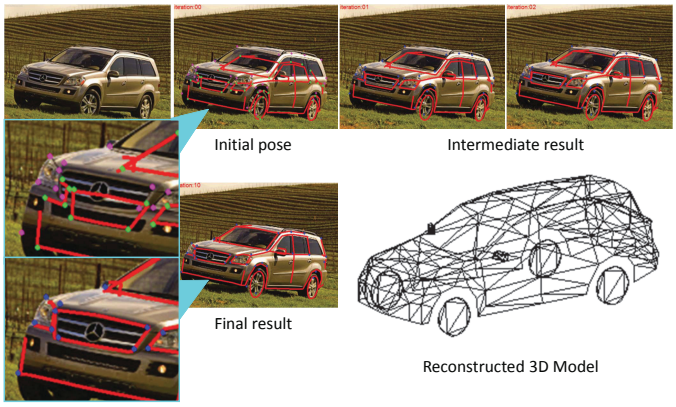


Fig. 11. Illustration of ground truth generation. Red lines represent a projected vehicle model. Green dots are hard-constrained projected points and purple dots are corresponding points. After several iterations, we get a well fitted model as the final result which can be reconstructed as a 3D model. Best seen in color.

reflection. The pose variation is not only on the up-axis but also on the other axes. Some examples are shown in Figure 10. No doubt the diversity challenges the model fitting and the recognition. Also, vehicle instances made by the same manufacturer, for example, *Honda Odyssey*, *Honda Pilot*, and *Honda CR-V* shown in the second row of Figure 10, may influence the performance if we focus on retrieving vehicle images which belong to the same instance as the input image. We have generated ground truth for this dataset. The ground truth of each image includes reconstructed 3D vehicle model, 2D vehicle mask in the image, and parameters of perspective matrix and shape. The grouth truth generation process is described in the next section.

B. Ground Truth Generation

The ground truth of *NetCarShow300* is obtained by aligning the projected models manually as Figure 11. In other words, we implement an annotation tool which lets annotators be able to adjust the location and pose of each 3D model, and we can set several hard-constrained corresponding points between a projected model and a vehicle image. According to our experience, the points on the corners, silhouette, and especially rear parts of a vehicle are mostly needed to be selected. Given those hard-constrained correspondence, we can update the shape distribution iteratively by point registration approach. Finally, we get a good fitting result which can be used to reconstruct an approximate 3D vehicle model as the ground truth.

C. Vehicle Retrieval Performance

In this experiment, we apply several descriptors on extracted parts which are resized to the same number of pixels while keeping the ratio between height and width. Those retrieved vehicle instances which have the same label as the query instance are correct. We compare the mean average precision (MAP) performance on different sources including a whole vehicle image and three parts, grilles, lamps, and the most visible wheel. Then, we do sensitivity tests to select the late fusion weights and obtain the best parameters. We test on three state-of-the-art feature descriptors. Firstly, Difference of Gaussian (DoG) detector and SIFT descriptor are used [25]. For constructing the visual word vocabulary, we start by applying SIFT descriptor on the images of three informative parts. Each descriptor contains 4×4 cells with 8 orientation bins, resulting in 128-dimensional feature vectors. The average number of feature descriptors for grille, wheel and lamp parts are 44.8, 62.43 and 56.75 respectively. We then group these visually similar descriptors by hierarchical clustering method to create a general 512-visual word vocabulary of prototypical local appearances. This visual word vocabulary will be used for all different sources (each row in TABLE I represents a different source) in the following experiments. In Section VIII, we further discuss the effect of using different vocabulary set and different vocabulary sizes for each source. Secondly, we use Pyramid Histogram of Oriented Gradients (PHOG) [26] which computes the histogram of oriented gradients in a region with several levels. Here we let the number of level be 3 and concatenate the vectors into a 168-dimension descriptor. Thirdly, we adopt the rotation-invariant feature, Local Binary Pattern Histogram Fourier (LBPHF) [27]. the descriptor applies to a whole region and is computed from discrete Fourier transforms of local binary pattern (LBP) histograms. In other words, the descriptor describes the appearance locally based on the signs of differences of neighboring pixels. We use three different radiuses of the circular neighborhoods and obtain 478-dimension descriptors.

We collect the leave-one-out results with these descriptors combined with L1 or cosine distance (cf. Table I). Distance metric is complementary to our part-based model; other distance metrics are also adoptable in this framework. To compare our part-based method, the baseline is set as the state-of-the-art approach in content-based image retrieval (please refer to #1 row: "Original Body Original Side" in Table I), which is based on the un-rectified and segmented whole vehicle image described by several state-of-the-art feature descriptors. To be fair, the whole vehicles of our baseline are segmented from background (cf. segmented vehicle image in Fig. 1(a)). Comparing these local or global descriptors combined with two measurements, we find that L1 distance is superior to cosine similarity, SIFT feature is better on the whole image or original parts, PHOG feature is more competitive on rectified parts, and LBPHF feature has moderate precision. Obviously, PHOG maintains the structural consistency which is important to distinguish different vehicle instances so it is benefited a lot by part rectification. LBPHF feature considers neighboring pixels and local distribution so it gains a little with rectification.

Intuitively, flipping alignment which mirrors the opposite structures makes vehicle images which are primarily in different viewpoints become similar. In Table I, the pair of the 3rd and 4th rows indicates that flipping alignment from "Original Side" to "Same Side" improves the performance by about 2–10%.

After we rectify parts into the frontal view, the pair of the 4th and 5th rows, the pair of the 7th and 8th rows, and the pair of the 9th and 10th rows illustrate that this step from

TABLE I
THE PERFORMANCE (IN MAP) FOR VEHICLE RETRIEVAL EXPERIMENTS."FUSION OF 70% GRILLE, LAMP, AND WHEEL" OBTAINS THE BEST MAP OF 63.08% AND IS MUCH BETTER THAN THE BASELINE "ORIGINAL BODY ORIGINAL SID" (18.77% WITH SIFT+L1) REFERRING TO THE VEHICLE IMAGE WITHIN THE BOUNDING BOX IN FIGURE. 1(A)."RECTIFIED BODY SAME SIDE" REFERS TO FIGURE 1(C),"ORIGINAL FRONT ORIGINAL SIDE" AND "ORIGINAL FRONT SAME SIDE" REFER TO THE FIGURE 8(A) AND (B) RESPECTIVELY."RECTIFIED SAME SIDE" WITH "100% FRONT", "70% FRONT" AND "50% FRONT" REFER TO FIGURE 8(C). THE MEANINGS OF THE REST ROWS ARE SIMILAR. THE SIFT FEATURES IN THIS TABLE IS BASED ON A GENERAL 512-VISUAL VOCABULARY CONSTRUCTED FROM THREE INFORMATIVE PARTS.

#	Descriptor+Distance Measure	SIFT+L1	SIFT+COS	LBPHF+L1	LBPHF+COS	PHOG+L1	PHOG+COS	WJS+PHOG+L1
1	Original Body Original Side (baseline)	18.77%	18.21%	11.06%	9.71%	10.44%	9.30%	10.81%
2	Original Front Original Side	36.96%	32.98%	20.17%	17.54%	22.87%	20.02%	18.14%
3	Rectified Body Same Side	32.01%	29.30%	21.10%	17.47%	31.61%	23.76%	25.26%
4	Original Front Same Side	39.42%	34.63%	20.08%	17.40%	35.63%	29.87%	29.22%
5	Rectified Front Same Side	37.95%	34.00%	25.27%	21.23%	48.99%	38.87%	37.80%
6	Rectified 70% Front Same Side	38.95%	32.99%	26.89%	22.77%	51.76%	41.16%	41.88%
7	Rectified 50% Front Same Side	29.27%	26.48%	25.58%	22.66%	**54.84%**	45.05%	44.58%
8	Original 50% Front Same Side	30.98%	26.88%	20.83%	18.39%	44.96%	35.19%	36.93%
9	Original Grille Same Side	36.26%	27.74%	24.10%	21.49%	38.01%	29.50%	28.90%
10	Rectified Grille Same Side	31.85%	27.17%	35.57%	31.64%	45.13%	34.53%	34.40%
11	Rectified 70% Grille Same Side	30.72%	27.55%	34.30%	30.99%	47.38%	36.87%	31.66%
12	Rectified 50% Grille Same Side	22.36%	20.61%	35.09%	32.69%	47.26%	37.91%	28.79%
13	Rectified Lamp Same Side	13.17%	12.24%	25.77%	23.78%	47.31%	42.21%	28.93%
14	Rectified Wheel Same Side	13.78%	11.87%	10.80%	9.62%	14.00%	12.13%	9.86%
15	Fusion of 70% Grille, Lamp, Wheel	34.26%	30.23%	43.24%	38.54	**63.08%**	53.79%	42.89%

"Original" to "Rectified" improves the performance by about 5–13%. We see that rectifying parts into a specific view deals with various viewpoints and makes parts more comparable physically.

Refer to the 6th or 7th row, "Rectified 70% Front Same Side" or "Rectified 50% Front Same Side," in Table I, retaining 70% or 50% part regions which are more undistorted increases MAP by around 2–6% compared with the 5th row, "Rectified Front Same Side." It shows that if we retain only informative regions, the performance will be improved.

The 7th row, "Rectified 50% Front Same Side," in Table I reveals that PHOG descriptor with L1 measure (PHOG+L1) outperforms other descriptors and achieves an MAP of 54.84% with 50% frontal parts composed of half grille and a lamp. The MAP is higher than our baselines (32.01% and 36.96%). Considering each parts, the grille and lamp are more discriminative than the wheel. It is shown in the 11th and 13th rows of Table I that the grille part has an MAP of 47% when retaining 70% rectified region, and the lamp part also achieves 47% with PHOG. Clearly, the composition of grilles and lamps is distinct between vehicle instances, but wheels are not very helpful when distinguishing the vehicle instances. One explanation is that the wheel structure may be not consistent in one vehicle instance. The other reason is that the internal structure of wheel parts may be blurred and unidentifiable when the vehicle is in motion.

Furthermore, the last row in Table I shows the result when we combine the three parts, grille, lamp, and wheel, to do late fusion:

$$S_{fusion} = w_{grille} \cdot S_{grille} + w_{lamp} \cdot S_{lamp} + max(1 - w_{grille} - w_{lamp}, 0) \cdot S_{wheel}, \quad (11)$$

where S means the similarity score, and w_{grille} and w_{lamp} are the weights of the grille and lamp respectively. For determining the weights, we exhaustively search different weight combinations over the range $0 \sim 1$ and evaluate them by leave-one-out cross validation. We then select the weight combination that leads the best MAP. The achieved MAP revealed in the last row is 63.08% on w_{grille} = 0.4 and w_{lamp} = 0.5, and it significantly outperforms the previous unfused results and our baselines. Some retrieved results are shown in Figure 12. Besides, the discussion about the fusion of fitted parts is described in the next section.

TABLE II
FITTING PRECISION.

Method	APD	STD
Initial Location	47.15	6.06
PR(KC)	39.26	9.90
PR(Rigid CPD)	29.59	6.91
PR(Non-rigid CPD)	26.53	6.84
JS	34.19	6.31

TABLE III
SENSITIVITY TEST WITH WEIGHTED JACOBIAN SYSTEM (WJS).

λ	APD	STD
0	20.29	5.97
0.1	19.75	5.09
0.2	19.22	4.84
0.3	**18.73**	**4.66**
0.4	18.98	4.71
0.5	19.25	4.79
0.6	19.86	4.94
0.7	22.83	5.54
0.8	26.59	6.38
0.9	30.60	7.40
1	34.19	7.91

D. Model Fitting Comparison

To compare the differences between model fitting approaches, we generate a testing data with noisy initial position

Fig. 12. Content-based vehicle retrieval results by "Fusion of 70% Grille, Lamp, Wheel" and "PHOG+L1." The number represents its retrieval ranking. The images with red border are wrong. The results show that proposed vehicle retrieval approach using part information and 3D model can retrieve the same instances under different environments. Best seen in color.

by adding random noises to ground truth. Then, we measure average pixel distance (APD) and standard deviation (STD) of visible vertices between fitted models and ground truth.

For CPD parameters, we have done sensitivity test to select the parameters. The parameters we used are noise weight = 0.9, width of Gaussian Kernel = 5, regularization weight = 10. In Table II, Rigid CPD (29.59 in APD) is better than other approaches (39.2 and 34.19), and non-rigid transformation improves the performance (26.53) because deformation possibility is considered even the model does not actually change the shape. In fact, we find that translating to good location is an important key for good fitting performance because worse translation may increase overall distance. Rotation and shape deformation then adjust the position of each vertex locally and lead to minor improvement. Furthermore, with the knowledge of the salient parts, we can utilize these weight maps to facilitate the 3D model fitting precision. In the sensitivity test shown in Table III, λ = 0.3 has the lowest error 18.73 which surpasses the non-rigid point registration result. Figure 13 depicts the some fitting results. Comparing these 3D model fitting approaches, weighted Jacobian System (WJS) approach shows the best performance.

The retrieval performance corresponding to the fitting result "WJS+PHOG+L1" is shown in the last column of Table I. It gets lower MAP (42.89%) than the ideal case (63.08%), but it still achieves relatively better performance than our baselines which are based on the segmented vehicle images, "Original Body Original Side" (18.77%) and "Original Front Original Side," (36.96%) and validates the impact of vehicle retrieval using part information and 3D model.

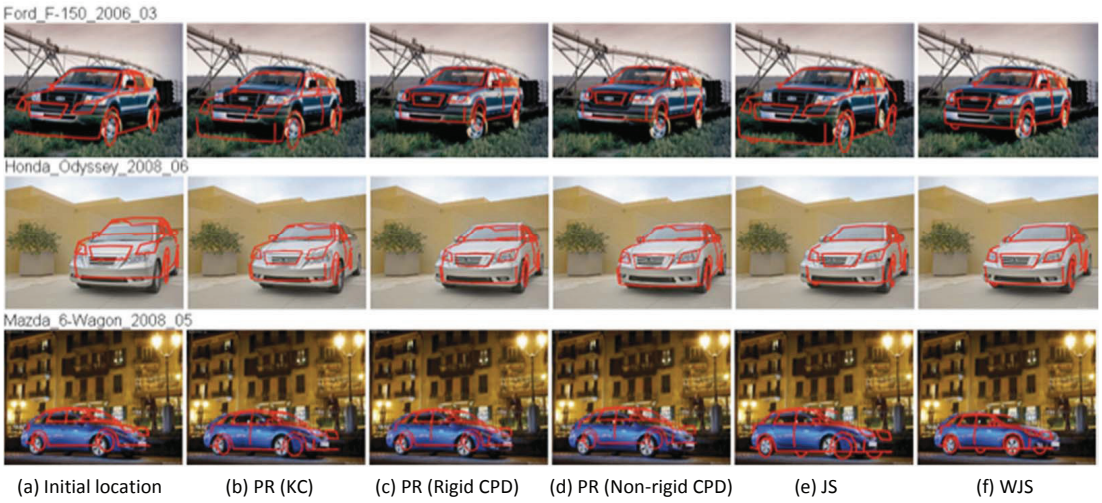


Fig. 13. 3D model fitting results. Proposed weighted Jacobian system (WJS) approach leveraging weight maps of informative parts outperforms other fitting approaches. Best seen in color.

VIII. Discussions

A. Visual Word

To examine the effect of a specific visual vocabulary and different vocabulary sizes for each source (each row in Table I represents a different source), we experimented on some representative sources selected from Table I. The results are shown in Table V. The performance is improved as using a different visual vocabulary for each source compared to a general visual vocabulary for all sources (comparing "512" column in Table V and "SIFT+L1" column in Table I). In addition, as the vocabulary size increases, the retrieval performance gets better, but saturates at certain point, as the similar observations in [28]. The proposed weighted Jacobian System (WJS) (MAP = 42.89%) still outperforms the baseline method, whose best MAP = 40.15% is achieved on 8192 vocabulary size.

We also find that "Rectified Body Same Side" achieves considerably good result (MAP = 55.18%) on 4096 vocabulary size, which demonstrates the effectiveness on rectification of the whole vehicle body (cf. Table V). This result also inspires us to take the advantages for both local informative parts with PHOG and global rectified body with SIFT. We conduct late fusion over the results from detailed part model: "Fusion of 70% Grille, Lamp, Wheel + PHOG" and global model: "Rectified Body Same Side + SIFT" in an average way to further boost the final performance to MAP 72.85% (cf. Table VI) by marrying the global context and local informative parts. The improvement is significant comparing with the prior CBIR methods, which generally ignore 3D alignment or informative parts.

B. Part Weight

In the section VII-C, we have demonstrated the effectiveness of rectifying parts into a specific view. However, the distortion areas of parts maybe increased as we rectify the parts from a large angle (e.g., rectifying the wheel region from side to frontal view). To further boost the performance and investigate the effects of view-dependent fusion, we categorize each input vehicle image into different view categories according to the pose parameters obtained after 3D model fitting step and apply different part weights on each view category. Because of the aid of the rectification technique, we only need a small set of view categories and thus speed up the computation.

We conduct an experiment to analyze the effect of part weights for different view categories. We categorize the training samples into two different categories according to their rotating angles of up axis (e.g., y axis) relative to frontal pose. Figure 14 shows the pose distribution of our dataset images. Due to the symmetric property of vehicles and the aid of the rectification technique, two categories are enough on our dataset for training the part weights. The ranges of angles on the two categories are $\{[-30, 30]\}$ and $\{[-60, 30), (30, 60]\}$ respectively. For each category, we select the weight combination that leads the best MAP as the same approach described in section VII-C. Table IV shows the performance results on two categories using the same experimental settings as our best part model: "Fusion of 70% Grille, Lamp, Wheel + PHOG + L1." We further boost the performance (MAP = 68.54% and 65.06%) compared to the best MAP (63.08%) in Table I. Grille and lamp are still more discriminative than wheel on both categories due to the range of viewing angles of our dataset. However, wheel gets higher weight as the rotating angles increase. Since it is common to have occlusions or certain rectification errors in the parts, we leverage multiple parts for ensuring more robust retrieval accuracy. That is why the view-dependent weights further boost the fusion performance. We believe more improvement will be gained as deriving the view-dependent fusion weights in more fine granularities.

IX. Conclusions and Future Work

In this paper, we effectively utilize 3D vehicle models for novel image-based vehicle retrieval. When robust 3D model fitting approaches are applied, it is possible to extract some discriminative parts. After part rectification, we demonstrate

TABLE IV

LEARNING PART WEIGHTS ON DIFFERENT VIEW CATEGORIES.THE EXPERIMENTAL SETTINGS ARE THE SAME AS OUR BEST PART MODEL: "FUSION OF 70% GRILLE, LAMP, WHEEL + PHOG + L1." BY VIEW-DEPENDENT FUSION, WE CAN FURTHER BOOST PERFORMANCE (MAP = 68.53% AND 65.06%) COMPARED TO THE BEST MAP 63.8% IN TABLE I. ALTHOUGH GRILLE AND LAMP ARE STILL MORE DISCRIMINATIVE THAN WHEEL ON BOTH CATEGORIES, WHEEL GETS THE HIGHER WEIGHT AS THE VIEWING ANGLES GET CLOSER TO SIDE VIEW.

Category	Part weights	MAP
$\{[-30, 30]\}$	$\omega_{grille} = 0.5, \omega_{lamp} = 0.8, \omega_{wheel} = 0$	68.54%
$\{[-60, 30), (30, 60]\}$	$\omega_{grille} = 0.4, \omega_{lamp} = 0.4, \omega_{wheel} = 0.2$	65.06%

TABLE V

THE PERFORMANCE (IN MAP) OF USING AN INDIVIDUAL VOCABULARY FOR EACH SOURCE (EACH ROW REPRESENTS A DIFFERENT SOURCE) VARIED WITH DIFFERENT VOCABULARY SIZE. "RECTIFIED BODY SAME SIDE" ACHIEVES CONSIDERABLE GOOD RESULT (MAP = 55.18%) ON 4096 VOCABULARY SIZE DEMONSTRATING THE EFFECTIVENESS ON RECTIFICATION OF WHOLE VEHICLE BODY IMAGE.AS IT PROVIDES GLOBAL CONTEXT FOR THE VEHICLE. MORE VOCABULARY DIMENSION IN SIFT VISUAL WORD DOES HELP BUT SATURATES AT 8192 AND BEYOND.

#	Vocabulary Size	256	512	1024	2048	4096	8192
1	Original Body Original Side(baseline)	19.92%	23.96%	28.45%	33.88%	37.21%	40.15%
3	Rectified Body Same Side	35.27%	39.80%	47.26%	52.78%	55.18%	54.78%
11	Rectified 70% Grille Same Side	32.94%	36.21%	39.09%	41.53%	40.47%	34.38%
13	Rectified Lamp Same Side	15.82%	17.36%	17.91%	19.76%	22.78%	21.66%
14	Rectified Wheel Same Side	16.75%	20.01%	21.76%	24.53%	25.11%	24.03%

TABLE VI

THE PERFORMANCE IS BOOSTED TO MAP 72.85 % BY LATE FUSION OVER THE RESULTS OF "FUSION OF 70% GRILLE, LAMP, WHEE" AND "RECTIFIED BODY SAME SIDE" BY MARRYING THE GLOBAL CONTEXT AND LOCAL INFORMATIVE PARTS.

Experimental Setting	MAP
Fusion of 70% Grille, Lamp, Wheel + PHOG + L1 + general 512-vocabulary	63.08%
Rectified Body Same Side + SIFT + L1 + individual 4096-vocabulary	55.18%
Fusion (average)	72.85%

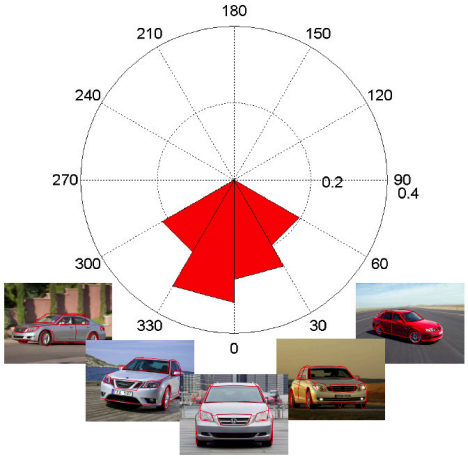


Fig. 14. The circular histogram shows the distribution of vehicle poses in our dataset. We compute the rotation angle on up axis for each image with respect to the frontal pose.

remarkable performance on a challenging dataset. The precision is surely notable and supports our idea on vehicle part information fusion. Besides, finding corresponding points during model fitting is such a challenging problem that a lot of researchers are investigating it. Here our investigation shows that the prior knowledge regarding certain parts has noteworthy impacts on 3D model fitting. While our current application is based on given initial pose and location, we are undergoing an approach to automatically generate the information. The computational cost of our method depends on two major components, 3D model alignment and image retrieval. For 3D model alignment, we currently implement two state-of-the-art approaches [10][11] on a laptop with 1.7 GHz Intel Core i5 CPU and 4G 1333 MHz memory, it takes about 1.5 seconds on average to align a 3D model for an image. In the future, we will try to improve the speed by using some parallel processing techniques (e.g., GPU). For image retrieval efficiency, two state-of-the-art techniques can be used: locality sensitive hashing [29] and inverted index [25]. When the feature vector is dense, locality sensitive hashing might be more efficient than inverted index. We conduct an experiment on our best part model: "Fusion of 70% Grille, Lamp, Wheel + PHOG + L1" to measure the required time for the retrieval step. PHOG is a 168-dimensional dense feature, we leverage Random Project (RP) [30] to speed up the image retrieval task since RP had been shown very effective in retrieving high-dimensional data [31][32][33][34]. We project the dense features to 500 bits and then compare the query and database images in the hamming space. In our experiment, the average response time is about 0.01 second on our dataset. Also, from our past experiences on CBIR systems [35][36], we believe these two techniques can still achieve real-time performance in the million-scale dataset. We expect to include vehicle detection and pose estimation steps, and we can build a structural content-based vehicle retrieval system on more difficult natural images without human annotation and leverage more informative parts to improve the performance.

REFERENCES

[1] N. Kumar, P. Belhumeur, and S. Nayar, "Facetracer: A search engine for large collections of images with faces," in *Proceedings of the 10th European Conference on Computer Vision: Part IV*, 2008, pp. 340–353.

[2] D. A. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk, "Attribute-based people search in surveillance environments," in *IEEE Workshop on Applications of Computer Vision*, 2009, pp. 1–8.

[3] M. Stark, M. Goesele, and B. Schiele, "Back to the future: Learning shape models from 3d cad data," in *BMVC*, 2010, pp. 106.1–106.11.

[4] M. Arie-Nachmison and R. Basri, "Constructing implicit 3d shape models for pose estimation," in *ICCV*, 2009, pp. 1341–1348.

[5] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-independent object class detection using 3d feature maps," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[6] Y. Guo, Y. Shan, H. S. Sawhney, and R. Kumar, "Peet: Prototype embedding and embedding transition for matching vehicles over disparate viewpoints," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[7] J. M. Ferryman, A. Worrall, G. D. Sullivan, and K. Baker, "A generic deformable model for vehicle recognition," in *Proceedings of the 1995 British Machine Vision Conference (Vol. 1)*, 1995, pp. 127–136.

[8] D. Koller, K. Danilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," in *Int. J. Comput. Vision*, 1993, pp. 257–281.

[9] S. M. Khan, H. Cheng, D. Matthies, and H. S. Sawhney, "3d model based vehicle classification in aerial imagery," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010, pp. 1681–1687.

[10] M. J. Leotta and J. L. Mundy, "Predicting high resolution image edges with a generic, adaptive, 3-d vehicle model." in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009, pp. 1311–1318.

[11] Y. Tsin, Y. Genc, and V. Ramesh, "Explicit 3d modeling for vehicle monitoring in non-overlapping cameras," in *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009, pp. 110–115.

[12] J. R. Smith and S. fu Chang, "Visualseek: a fully automated content-based image query system," in *Proceedings of the fourth ACM international conference on Multimedia*, 1996, pp. 87–98.

[13] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1026–1038, 2002.

[14] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, pp. 63–86, 2004.

[15] R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, and S. Pankanti, "Attribute-based vehicle search in crowded surveillance videos," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, 2011, pp. 18:1–18:8.

[16] V. Petrovic and T. F. Cootes, "Analysis of features for rigid structure vehicle type recognition," in *In British Machine Vision Conference*, 2004, pp. 587–596.

[17] P. Negri, X. Clady, M. Milgram, U. Pierre, and M. Curie-paris, "An oriented-contour point based voting algorithm for vehicle type classification," in *Proc. International Conference on Pattern Recognition*, 2006, pp. 574–577.

[18] F. M. Kazemi, S. Samadi, H. R. Poorreza, and M.-R. Akbarzadeh-T, "Vehicle recognition based on fourier, wavelet and curvelet transforms - a comparative study," *Information Technology: New Generations, Third International Conference on*, pp. 939–940, 2007.

[19] S. Rahati, R. Moravejian, E. M. Kazemi, and F. M. Kazemi, "Vehicle recognition using contourlet transform and svm," in *Proceedings of the Fifth International Conference on Information Technology: New Generations*, 2008, pp. 894–898.

[20] I. Zafar, E. A. Edirisinghe, and B. S. Acar, "Localized contourlet features in vehicle make and model recognition," *Proceedings of SPIE*, pp. 725 105–725 105–9, 2009.

[21] J. Gower, "Generalized procrustes analysis," *Psychometrika*, pp. 33–51, 1975.

[22] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *ECCV*, 2004, pp. 558–569.

[23] A. Myronenko and X. B. Song, "Point set registration coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 2262–2275, 2009.

[24] A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," in *Technometrics*, 1974, pp. 147–185.

[25] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," *ICCV*, 2003.

[26] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[27] T. Ahon, J. Matas, C. He, and M. Pietikainen, "Rotation invariant image description with local binary pattern histogram fourier features," in *Proceedings of the 16th Scandinavian Conference on Image Analysis*, 2009, pp. 61–70.

[28] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, 2007.

[29] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," *VLDB*, 1999.

[30] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," *KDD*, 2001.

[31] M. Casey and M. Slaney, "Fast recognition of remixed music audio," in *ICASSP*, 2007.

[32] R. Cai, C. Zhang, L. Zhang, and W.-Y. Ma, "Scalable music recommendation by search," *ACM Multimedia*, 2007.

[33] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," *ACM Multimedia*, 2004.

[34] W. Dong, Z. Wang, M. Charikar, and K. Li, "Efficiently matching sets of features with random histograms," in *ACM Multimedia*, 2008.

[35] Y.-H. Lei, Y.-Y. Chen, L. Iida, B. chu Chen, H.-H. Su, and W. H. Hsu, "Photo search by face positions and facial attributes on touch devices," in *ACM Multimedia*, 2011.

[36] Y.-H. Kuo, K.-T. Chen, C.-H. Chiang, and W. H. Hsu, "Query expansion for hash-based image object retrieval, acm multimedia," in *ACM Multimedia*, 2009.

Yen-Liang Lin received the BS degree in computer science and information engineering from the Chang Gung University in 2007 and received the MS degree in the Graduate Institute of Networking and Multimedia from National Taiwan University in 2009. He is currently a Ph.D. student in Graduate Institute of Networking and Multimedia at National Taiwan University. His research interests fall in multimedia content analysis, computer vision and mobile multimedia applications.

Ming-Kuang Tsai received the B.S. and M.S. degree from the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, in 2009 and 2011. He currently works for Garmin Taiwan. His research focuses on multimedia content analysis and retrieval.

Winston H. Hsu received the Ph.D. degree from the Department of Electrical Engineering, Columbia University, New York. He has been an Associate Professor in the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan, since September 2011. Prior to this, he was in the multi- media software industry for years. He is also with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. His research interests include multimedia content analysis, image/video indexing and retrieval, machine learning and mining over largescale databases.

Chih-Wei Chen received the B.S. degree in electronic engineering from National Taiwan University of Science and Technology, Taiwan, R.O.C., in 2007, and the M.S. degree in computer and communication engineering from National Cheng Kung University, Taiwan, R.O.C., in 2009. He is currently an associate engineer in Service Systems Technology Center, Industrial Technology Research Institute, Hsin-Chu, Taiwan. His research interests include image processing, image recognition, and cloud computing.