Three-Dimensional Deformable-Model-Based Localization and Recognition of Road Vehicles

Zhaoxiang Zhang, *Member, IEEE*, Tieniu Tan, *Fellow, IEEE*, Kaiqi Huang, *Senior Member, IEEE*, and Yunhong Wang, *Member, IEEE*

***Abstract*—We address the problem of model-based object recognition. Our aim is to localize and recognize road vehicles from monocular images or videos in calibrated traffic scenes. A 3-D deformable vehicle model with 12 shape parameters is set up as prior information, and its pose is determined by three parameters, which are its position on the ground plane and its orientation about the vertical axis under ground-plane constraints. An efficient local gradient-based method is proposed to evaluate the fitness between the projection of the vehicle model and image data, which is combined into a novel evolutionary computing framework to estimate the 12 shape parameters and three pose parameters by iterative evolution. The recovery of pose parameters achieves vehicle localization, whereas the shape parameters are used for vehicle recognition. Numerous experiments are conducted in this paper to demonstrate the performance of our approach. It is shown that the local gradient-based method can evaluate accurately and efficiently the fitness between the projection of the vehicle model and the image data. The evolutionary computing framework is effective for vehicles of different types and poses is robust to all kinds of occlusion.**

***Index Terms*—Evolutionary computing, fitness evaluation, model-based vision, vehicle localization, vehicle recognition, visual surveillance.**

I. Introduction

MODEL-BASED object localization and recognition is a classical issue in computer vision over many years with great potential for many real applications. Particularly in the traffic-scene surveillance domain, model-based vehicle localization and recognition from monocular images or videos plays important roles not only in accurate vehicle detection, tracking, and recognition but also in supplying intermediate information for high-level trajectory analysis and semantic interpretation.

Since the pioneering work of Roberts [1] in the early 1960s, prominent progress has been achieved from then on. In the early days, most of the algorithms are based on fixed models and adopt a typical matching and recognition framework. Two-dimensional geometric primitive features such as edge points, edge lines, vertices, and conic sections are first extracted from images as symbolic descriptions. Recognition is then realized by establishing the matching between the fixed model and extracted features. The tree search [2], the attributed graph search [3], the generalized Hough Transform [4], the viewpoint consistency constraints [5], and so on are adopted to achieve the 2-D–3-D correspondence and pose determination. However, the extraction and matching of 2-D geometric primitives such as lines and curves might be time-consuming and error-prone. As a result, these algorithms can only deal with simple objects such as polyhedrons and are not suitable for real applications.

Instead of direct 3-D–2-D matching, there is another strategy to achieve model-based localization. With an initialized pose confirmed, the 3-D model is projected first into an image plane so that the fitness between projection and image data can be evaluated directly in the image plane. This kind of method avoids a bottom-up procedure and simplifies the problem into an optimization framework. The common practice for fitness evaluation is based on the distance error of 2-D geometric primitives in the form of point-to-point [6], [7], point-to-line [8]–[10], and line-to-line [11], [12]. Lou *et al.* proposed the PLS method [9], which extracted edge points as primitive features and evaluated fitness scores based on the distance between each edge point and each projected line segment of a wire-frame model. However, primitive extraction and distance calculation are time-consuming and not robust to noise and occlusion. Beveridge *et al.* [10] made use of key features for matching between the 2-D line-segment model and image data. The ICONIC method in [13] sampled pixels uniformly around projected lines to estimate the directional derivatives of image intensities and combined them into fitness evaluation in a statistical framework. However, the pixel-intensity-based method is not stable and sensitive to noise. Liu *et al.* [14] proposed the Bayesian classification error (BCE) method, which adopted local region information and made use of a BCE to measure the dissimilarity of local regions for fitness evaluation. However, the assumption that image intensities satisfy Gaussian distribution is not always true in small regions.

Vehicle localization and recognition in traffic-scene surveillance is a good platform to apply model-based vision methods. The shapes of vehicles are approximately polyhedral, and much prior information in traffic scenes is useful to simplify the problem. Ground-plane constraints (GPC) [15] can decrease the number of pose parameters from six to three, whereas static cameras and calibrated scenes can make the projection much easier. In the previous years, most work in this field

Manuscript received December 16, 2009; revised July 19, 2010 and March 16, 2011; accepted June 07, 2011. Date of publication June 30, 2011; date of current version December 16, 2011. This work was supported by the National Basic Research Program of China under Grant 2010CB327902, by the National Natural Science Foundation of China under Grant 61005016, Grant 60736018, and Grant 61061130560; and the Fundamental Research Funds for the Central Universities. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sabine Susstrunk.

Z. Zhang and Y. Wang are with the Laboratory of Intelligent Recognition and Image Processing, Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: zxzhang@buaa.edu.cn; yhwang@buaa.edu.cn).

T. Tan and K. Huang are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: tnt@nlpr.ia.ac.cn; kqhuang@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TIP.2011.2160954

was based on fixed vehicle models (e.g., in [9] and [14]–[16]). In this case, vehicle localization is realized by determining the pose parameters that can make the projection of the fixed model best fit the image data, and vehicle recognition is based on comparing evaluation scores of different vehicle models. Evidently, this strategy has inherent disadvantages. Success of the strategy depends strongly on how accurately the 3-D model captures the geometry of real vehicles. Since there are so many different makes of vehicles in reality, a large set of fixed models are needed to capture accurately their geometries, respectively. As we know, building so many different vehicle models is a formidable task. Even worse, the processing time of fixed-model-based methods is linearly proportional to the number of vehicle models. As a result, fixed-model-based methods are not efficient and limit the accuracy of model-based recognition.

A deformable vehicle model seems to be a better choice to overcome the disadvantages mentioned above. On one hand, it can exert the advantages of model-based methods. On the other hand, it can adapt itself to deal with most vehicles in reality. The deformable vehicle model was mentioned first in [11]. However, their focus was on vehicle tracking instead of localization and recognition. Ferryman *et al.* [17] presented a deformable model with 29 parameters and made use of PCA coefficients as parameters for recognition. However, they needed to collect a large sample of 3-D models from images interactively for training and cannot recover accurate shape parameters. In addition, there are studies focusing on 3-D triangle-based vehicle modeling [18] and image-level representation [19], but they cannot deal with vehicles of different types and poses effectively. A recent progress has been proposed in [20] to achieve vehicle alignment based on a 2-D shape model, which can deal effectively with different views, noise, and occlusion. The method made use of redundant examples to learn a landmark-based 2-D shape model and achieved vehicle alignment based on a hypothesis-test strategy. Our proposed approach has some common features with this method. Both of them adopt prior information for fitness evaluation and solve the problem in an optimization framework. The differences are that we make use of 3-D models as prior information instead of redundant examples and we adopt evolutionary computing instead of hypothesis-test-based strategy. In the following, we will describe our approach in detail.

In this paper, we propose a novel method for model-based localization and recognition of road vehicles. A deformable vehicle model containing 12 shape parameters is set up as prior information and will be described in detail in Section II. As vehicles move on the ground plane almost all the time, only three parameters are needed to determine the vehicle pose, which are position (X, Y) on the ground plane and orientation θ about the vertical axis. The three pose parameters and two shape parameters form a combined 15-D parameter space. Each point in the combined 15-D space can be imagined to be an individual, which can determine completely its projection on the image plane since we assume to use only one static calibrated camera. An efficient fitness evaluation method is then proposed to evaluate the fitness between the projection and image data. Further, the fitness evaluation is combined into an evolutionary computing framework to select better individuals from the combined parameter space based on an iterative population selection strategy. Here, better individuals correspond to those whose projections on the image plane have higher fitness evaluation scores (FESs). In this way, the population selection can select iteratively better individuals and finally output the best individual whose projection best fits image data. The pose parameters of the best individual achieve vehicle localization, whereas the 12 shape parameters are used for vehicle recognition. The flowchart of our algorithm is shown in Fig. 1. As we can see, evolutionary computing is a stochastic searching-based method, which has already had good applications in human body-motion analysis [21].

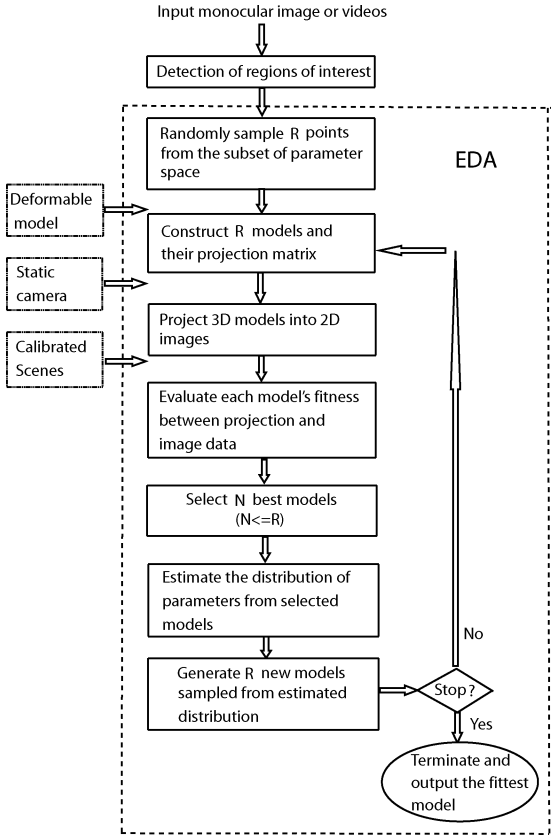


Fig. 1. Flowchart of the algorithm.

From one monocular image of calibrated scenes containing vehicles, we can recover both pose and shape parameters to realize the localization and recognition of vehicles. The contributions of this paper include a simple but distinctive deformable model setup, an efficient local gradient-based method to evaluate the fitness between the projection of the vehicle model and image data, and a novel evolutionary computing framework to recover both pose and shape parameters.

The remainder of this paper is organized as follows. We present a deformable 3-D vehicle model in detail in Section II. Then, we address the pose parameters in Section III. In Section IV, an efficient method based on local gradient image features is proposed to evaluate fitness between the projection of the 3-D model and image data. In Section V, we describe our evolutionary computing framework to recover both pose and

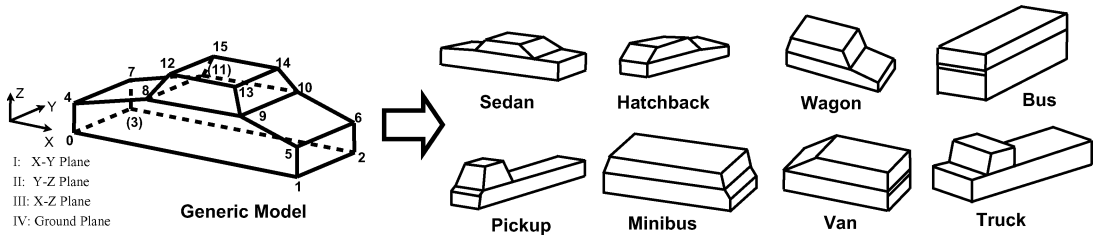


Fig. 2. Generic 3-D vehicle model that can be deformed to fit with different vehicles.

TABLE I
SHAPE PARAMETERS OF THE DEFORMABLE VEHICLE MODEL (IN MILLIMETERS)

Parameters	Descriptions	Value Range
$W1$	Distance from 1 to 2	$[1500, 2100]$
$H1$	Distance from 1 to 5	$[400, 800]$
$H2$	Distance from 0 to 4	$[400, 800]$
L	Distance from 0 to 1	$[3200, 4000]$
$H3$	Distance from 8 to I	$[\max(H1, H2), 900]$
$X1$	Distance from 8 to II	$[0, L/2]$
$X2$	Distance from 9 to II	$[\max(X1, L/2), L - 200]$
$X3$	Distance from 12 to II	$[X1, \min(X2 - 1300, L/2)]$
$X4$	Distance from 13 to II	$[X3 + 1000, X2 - 200]$
$W2$	Distance from 13 to 14	$[1000, W1]$
$H4$	Distance from 13 to I	$[1.5 \times H3, 1400]$
Δ	Distance from I to IV	$[150, 250]$

shape parameters. Then, the experimental results and analysis are presented in Section VI. Finally, we draw our conclusions in Section VII.

II. SHAPE PARAMETERS OF THE DEFORMABLE VEHICLE MODEL

In this section, we present our deformable vehicle model with 12 shape parameters, which is shown in Fig. 2.

It is a 3-D wire-frame model whose 12 shape parameters are listed in Table I. As we know, most vehicles in reality should belong to a very small subset of the 12-D shape parameter space. To enhance the efficiency of the following shape recovery, we have collected a set of fixed 3-D models, including most types of vehicles in reality and mined rules from the set to reduce the value range of each shape parameter. The mined rules are shown in Table I, which are not unique and perfect but comply with all the fixed 3-D vehicle models in the set. We can see that all the shape parameters have respective value ranges and are not self-independent. After recovering the 12 parameters, we can determine totally the shape due to the symmetric property so that we can use them for vehicle recognition.

Compared with previous vehicle models, our model has several advantages, which are listed as follows.

1) *Accurate*. As we have described before, there are different kinds of vehicles in reality, such as sedan, truck, wagon, van, and hatchback. Even if the type is restricted to one, the makes are still different from each other in detail. As a result, fixed-model-based methods may need a large number of models and cannot capture accurately the geometry of most vehicles in reality. In contrast, one deformable model is adaptable to fit with most vehicles accurately.
2) *Efficient*. Fixed-model-based methods need a large number of models. Since the computational time is linearly proportional to the number of models, the fixed-model-based method is not efficient to deal with large vehicle variations. In contrast, deformable-model-based methods need only one model with shape parameters to be more efficient.
3) *Simple*. Compared with previously used deformable models, our model has only 12 shape parameters. As vehicles somehow lack in texture and are rich in edge lines, our model is wire-framed and composed of only 26 3-D line segments. It is robust to light and view changes and also convenient for graphic operation. Furthermore, we ignore some unstable structures such as wheels, side windows, and lights since they are not discriminative for localization and recognition.
4) *General*. As shown in Fig. 2, our deformable model can capture accurately the geometry of most vehicles in reality by variation of the 12 shape parameters.

III. POSE PARAMETERS UNDER GPC

With the GPCs adopted [15], the pose of a vehicle can be determined only by its position (X, Y) on the ground plane and its orientation θ about the vertical axis of the world coordinate system (WCS). With these extra three pose parameters, we can project the model into images. The projection relation between the WCS and the model coordinate system (MCS) is shown in (1), where M is the projection matrix from the WCS to the image coordinate system and known by camera calibration. Points are represented by homogeneous coordinates, i.e.,

$$\begin{bmatrix} \mu \\ \nu \\ z \end{bmatrix} = M \begin{bmatrix} \cos\theta & -\sin\theta & 0 & X \\ \sin\theta & \cos\theta & 0 & Y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix}. \quad (1)$$

In practice, for model-based vehicle localization and recognition, we need to confirm first an initialized pose to project the 3-D model into the image plane for matching and optimization. In this paper, the pose initialization of the 3-D model is achieved from image-plane-based vehicle detection. For moving vehicles, we can extract regions of interest by motion detection and obtain vehicle regions by motion and shape information, which is well illustrated in [22]. For static vehicles, many statistical models can be used to detect vehicles, which have been well summarized in [23]. The details of detection is not the emphasis of this paper, and we assume that the bounding boxes of vehicle regions have already been obtained, as illustrated in Fig. 3. With the camera calibrated and the projection matrix M known, we can obtain simply the homography correspondence between

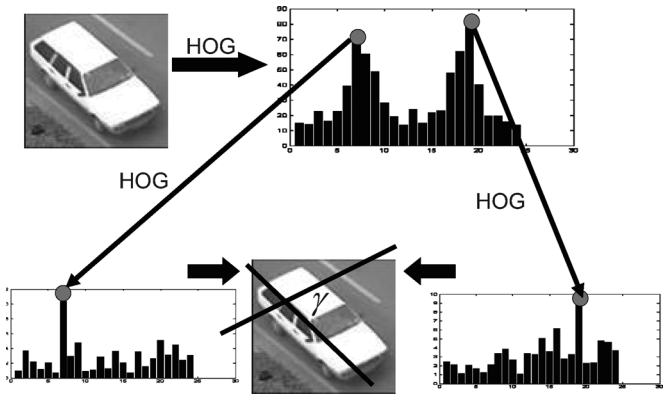


Fig. 3. Illustration of bounding-box extraction and pose initialization.

the image plane and the ground plane. The coordinates of the point in the ground plane, which corresponds to the center of the bounding box in the image plane, are taken as the initialized value of X and Y.

Compared with the translation parameters (X, Y), the initialization of θ is more complicated. As we know, vehicles are artificial objects of prior shape information. Projections of a vehicle in the image plane are rich in lines of the three orientations $(\overrightarrow{O_1}, \overrightarrow{O_2}, \overrightarrow{O_3})$, which correspond to three orthogonal directions $\overrightarrow{(1,0,0)}$, $\overrightarrow{(0,1,0)}$, and $\overrightarrow{(0,0,1)}$ in MCS. We can estimate these three orientations in the image plane based on image gradients and use angles among them as constraints to estimate the initialization value of θ. Since the z axes of MCS and WCS are coincident to be perpendicular to the ground plane, the projection $(\overrightarrow{O_3})$ of $\overrightarrow{(0,0,1)}$ can be estimated in the image plane based on the calibrated camera. In this case, we can distinguish this orientation from the others. One example is shown in Fig. 3. There are only two peaks in the histogram of oriented gradients (HOG). By comparing these two peaks to the calculated $\overrightarrow{O_3}$, we can confirm that they are corresponding to $\overrightarrow{O_1}$ and $\overrightarrow{O_2}$. With a second-step HOG in the neighborhood of each peak, we can obtain further more accurate estimation of $\overrightarrow{O_1}$ and $\overrightarrow{O_2}$. Then, angles γ between $\overrightarrow{O_1}$ and $\overrightarrow{O_2}$ can be calculated simply, and the initialization of θ is achieved by

$$\cos\gamma = \frac{\overrightarrow{O_1} \cdot \overrightarrow{O_2}}{|\overrightarrow{O_1}||\overrightarrow{O_2}|}. \tag{2}$$

With the pose parameters initialized, the 3-D vehicle model can be projected into the image plane to match with image data. An accurate and efficient method is required for fitness evaluation between the projection of the 3-D model and image data, which will be described in detail in the next section.

IV. Fitness Evaluation

Fitness evaluation between the projected vehicle model and image data is proposed first in our previous work for fixed-model-based vehicle localization [24] and tracking [25]. Owing to its outstanding performance, we try to adopt it to deformable-based approaches, which will be described in detail as follows.

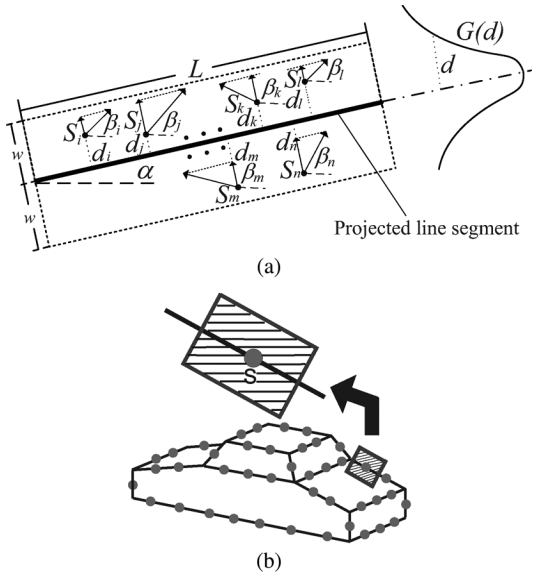


Fig. 4. Principle of fitness evaluation. (a) Regions around a visible projected line. (b) Regions around sampled points.

With shape and pose parameters initialized, we can project the wire-frame vehicle model onto the image plane to form a series of visible projected line segments.

For every visible projected line segment whose direction is assigned as α with length of L in the image plane, we form a $L \times 2w$ virtual rectangle, as shown in Fig. 4(a). If the line fits the image data well, the gradient directions of pixels with large gradient magnitude values in the rectangle should focus on the perpendicular direction of the projected line. As a result, we can estimate the fitness score from the gradient information of all pixels within the rectangle. For pixel S_i within the rectangle, we can calculate simply its gradient magnitude $m(x, y)$ and orientation $\beta(x, y)$ from pixel differences as follows:

$$\begin{cases} A(x,y) = I(x+1,y) - I(x-1,y) \\ B(x,y) = I(x,y+1) - I(x,y-1) \\ m(x,y) = \sqrt{A(x,y)^2 + B(x,y)^2} \\ \beta(x,y) = \tan^{-1}\left(B(x,y)/A(x,y)\right). \end{cases} \tag{3}$$

Then, the fitness score E_{S_i} contributed by S_i is measured by the vertical component of its gradient magnitude along the line direction as follows:

$$E_{S_i} = \left| m(x,y) \cdot \sin\left(\beta(x,y) - \alpha\right) \right|. \tag{4}$$

It is evident that not all pixels in the rectangle have the same weight for fitness evaluation. For those closer to the projected line segment, the pixels should give more contributions to the FES. As a result, we give every pixel S_i a weight value that equals to $G_{\mu,\sigma}(d_i)$. Here, d_i is the distance between S_i and the projected line segment, and $G_{\mu,\sigma}$ is a standard Gaussian distribution with $\mu = 0$ and $\sigma = w$.

In this way, the fitness score of the projected line segment l is measured by the weighted sum of E_{S_i} as

$$E_l = \sum_{S_i} \left[E_{S_i} \cdot G_{0,w}(d_i) \right] \tag{5}$$

and we calculate the whole FES between the projection of the vehicle model and image data from all visible projected line segments as

$$E = \sum_{l} [\log(E_l)]. \tag{6}$$

In addition to the above descriptions, our fitness evaluation can be sped up in the following way. First, instead of taking one $L \times 2w$ local region around every projected line, we can sample points along the projected line to generate many subregions, as shown in Fig. 4(b), for respective calculation. This adjustment can still improve its ability to deal with static occlusion. In realization, the number of sampling points of each project line with a length of L is $n = 6\times((L-L_s)/(L_l-L_s))+1$. Here, L_l and L_s are the length of the longest and shortest projected lines, respectively. Second, instead of calculating the contributions of every pixel, we can set threshold T and only focus on those with gradient magnitude above T. The performance of these two strategies will be analyzed in Section VI.

Our approach performs efficiently and accurately for fitness evaluation, which is demonstrated in Section VI. In fact, our approach is not isolated but has many relations to other pioneering methods. Compared with ICONIC [13], we use gradient information from every pixel in the local regions instead of estimating directional derivatives from sampled image points. This strategy keeps efficiency but adds robustness to noise and occlusion. Compared with PLS [9], we avoid time-consuming edge detection and distance calculation. Both BCE [14] and our method make use of local region information, but our method is free of inaccurate assumptions of data distributions. In Section VI, we will compare our approach to other pioneering methods to demonstrate its advantages.

V. Parameter Recovery Based on Evolutionary Computing

In this section, we combine our methods of model projection and fitness evaluation with the estimation of distribution algorithms (EDAs) to form a novel evolutionary computation framework. This framework can select iteratively shape and pose parameters from the combined parameter space to make the projection fit the image data better and better.

EDAs were introduced first in the field of genetic computing in [26]. Unlike traditional genetic algorithms, neither crossover nor mutation operators are necessary in EDAs. Instead, it samples a new population of individuals from a probability distribution, which is estimated from a database containing selected individuals of the previous generation. The interrelations between variables are expressed explicitly through a joint probability distribution.

There are several EDAs ([26], [27]) for optimization applications in continuous domains. One kind of the algorithms, which is called a generic continuous EDA approach, is shown in algorithm 1. Here, D denotes a data set, g denotes the generation, and $f(x)$ denotes the joint probability distribution of the data set. EDAs have great advantages for optimization in high-dimensional space but have been used hardly in the computer-vision domain before.

Algorithm 1 Continuous EDA Approach

1: BEGIN

2: $D_0 \longleftarrow$ Generate R individuals randomly; $g \longleftarrow 1$

3: **while** The final stopping criterion is not met **do**

4: $D_{g-1}^N \longleftarrow$ Select $N \leq R$ individuals from D_{g-1} according to a selection method

5: $f_g(x) \longleftarrow$ Estimate the density function using D_{g-1}^N

6: $D_g \longleftarrow$ Sample R individuals from $f_g(x)$

7: $g \longleftarrow g + 1$

8: **end while**

9: END

Now, we see the problem of recovering the pose and shape parameters of vehicle models. As we know, it is a very complicated optimization problem, which has 15 parameters to be optimized simultaneously. Furthermore, these parameters have certain value ranges and are not independent to each other. It is difficult to tackle this problem using conventional optimization algorithms. Here, we combine the projection-fitting-optimization strategy with EDAs to form our new framework, which has been attempted with BCE-based fitness evaluation in our previous work [28].

In this framework, each individual in the population represents a point in the combined 15-D parameter space. All shape parameters have their value ranges, as described in Section II. With pose parameters initialized as (X_i, Y_i, θ_i) by the method described in Section III, we set the value range of pose parameters as $(X_i \pm 1000 \text{ mm}, Y_i \pm 1000 \text{ mm}, \theta_i \pm 5^\circ)$, which is enough to tolerate the initialization inaccuracy and contain the correct pose. With the value range of pose and shape parameters confirmed, the individuals are in fact selected from a very small subset instead of the whole parameter space. Every individual of the population can determine a unique projection on the image plane, which is used to fit image data. Those individuals who have higher evaluation scores are chosen from them and applied to generate new individuals for the next generation.

As the parameter space is high dimensional and the parameters are not independent of each other, we choose a specific continuous EDA approach called the estimation of multivariate normal algorithm–global ($\text{EMNA}_{\text{global}}$) [27] to form our evolutionary computing framework. In this framework, the joint distribution of parameters is assumed to be a multivariate normal distribution, and the FES is used as a criterion for selection. The flow of the algorithm is illustrated in algorithm 2 as follows.

Algorithm 2 $\text{EMNA}_{\text{global}}$ for deformable models

1: BEGIN

2: $D_0 \longleftarrow$ Sample R individuals randomly from the 15-D parameter space; $g \longleftarrow 1$

(a) (b)

Fig. 5. Two frames of PETS 2000 Database. (a) Frame 0137. (b) Frame 0157.

3: **while** the difference between two generations of average fitness is less than a threshold **do**

4: Calculate the evaluation score of each individual using FES and sort them.

5: $D_{g-1}^N \longleftarrow$ Select $N \leq R$ individuals which have higher scores from D_{g-1}

6: $f_g(x) = f(x|D_{g-1}^N) = \mathcal{N}(x;\mu_g,\Sigma_g) \longleftarrow$ Estimate the multivariate normal density function using D_{g-1}^N

7: $D_g \longleftarrow$ Sample R individuals (the new populations) from $f_g(x)$

8: $g \longleftarrow g+1$

9: **end while**

10: END

In the described algorithm, we have to estimate the multivariate normal density function at generation g. The means $\mu_{i,g}$ and the elements of the variance–covariance matrix $\sigma_{ij,g}^2$ are estimated in the following way:

$$\begin{cases} \hat{\mu}_{i,g} = \frac{1}{N}\sum_{r=1}^{N} x_{i,r}^g, & i=1,2,\cdots,n \\ \hat{\sigma}_{i,g}^2 = \frac{1}{N}\sum_{r=1}^{N}\left(x_{i,r}^g - \hat{\mu}_{i,g}\right)^2 \\ \hat{\sigma}_{ij,g}^2 = \frac{1}{N}\sum_{r=1}^{N}\left(x_{i,r}^g - \hat{\mu}_{i,g}\right)\left(x_{j,r}^g - \hat{\mu}_{j,g}\right). \end{cases} \tag{7}$$

In this section, we have proposed an evolutionary computing framework based on $\text{EMNA}_{\text{global}}$ to recover iteratively the pose and shape parameters of deformable vehicle models. The experimental results and analysis of the framework for model-based vehicle localization and recognition will be described in detail in the next section.

VI. Experimental Results and Analysis

Numerous experiments have been conducted, and experimental results are presented in this section to demonstrate the performance of the proposed algorithm. Parts of the experiments are conducted using the videos of the PETS 2000 database [29]. All experiments are carried out on a personal computer with a P4 3.0G central processing unit and 512M DDR. Two frames shown in Fig. 5 illustrate the scene of PETS 2000 database.

A. Performance Evaluation of Fitness Evaluation

Fitness evaluation plays an important role in our evolutionary computing framework. How accurately the approach selects better individuals from the population determines the performance of vehicle localization and recognition. Furthermore, the fitness evaluation method also determines how fast the iterative algorithm converges.

A good fitness evaluation algorithm should have the following desirable properties.

1) *Accurate.* The approach should be consisted with the human's perception of fitness between the projection and image data. The point with the highest evaluation score should correspond to the ground truth in reality.
2) *Efficient.* The approach should be efficient with low computational cost. As we know, fitness evaluation is the most frequent and time-consuming step in our evolutionary computing framework. The efficiency of fitness evaluation can boost greatly the global performance of localization and recognition.
3) *Convenience for optimization*. The approach should be convenient for optimization. That means the optimization surface should be smooth with conspicuous peaks so that the algorithm can converge with a small number of iterations.

We assess the performance of our fitness evaluation method according to the above three considerations. Since the shape parameters are high dimensional and not independent of each other, it is difficult to test fitness evaluation performance in the entire shape parameter space. However, we can evaluate the performance of fitness evaluation on the 3-D pose parameter space by confirming manually the vehicle shape. The ICONIC [13], PLS [9], and BCE [14] approaches are chosen for comparison because all of them have already had successful applications in model-based vehicle localization.

1) Optimization-Surface Analysis: The optimization-surface property is an important metric to decide its convenience for optimization, which is calculated as follows. For a monocular image with regions of interest detected, we first adjust manually the pose parameters to obtain (X^*, Y^*, θ^*) as "ground truth" so that the projection can fit image data as closely as one can visually make. Then, the model is moved in the neighborhood of the "ground truth" to obtain the optimization surface $\{E(P_{i,j,k}), -30 \leq i,j,k \leq 30\}$, where $P_{i,j,k} = (X^* + i\lambda, Y^* + j\lambda, \theta^* + k\lambda')$. As we have discussed in Section III, θ can be determined quite accurately using motion and main gradient direction information. For data visualization, we fix $\theta = \theta^*$ to obtain the optimization surface. Almost all the frames in PETS 2000 Database [29] show similar results. Frames 0137 and 0157 are chosen from them as examples for analysis, whose optimization surfaces are shown in Figs. 6 and 7 using all the four methods.

The comparison of optimization surfaces in these two frames demonstrates very clearly the advantages of our approach. Our approach performs best to show the smoothest surface and the most conspicuous global extreme. In contrast, the surface of ICONIC [13] is not smooth at all and has many local extremes; the curve of the PLS [9] has no conspicuous global extreme,

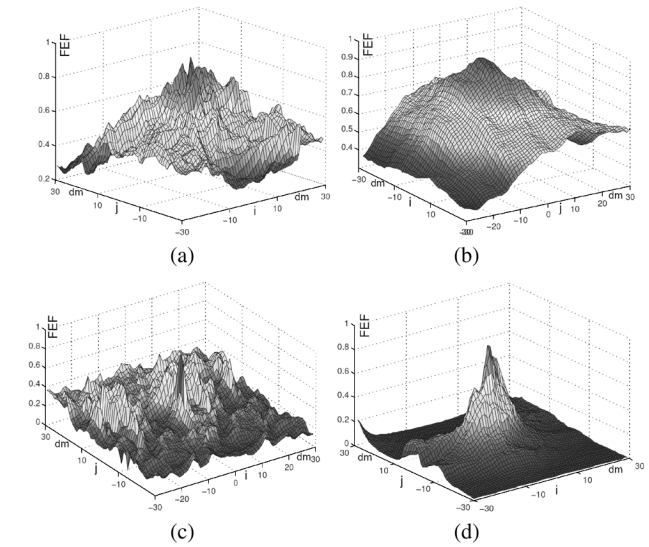


Fig. 6. Optimization-surface comparison of frame 0137. (a) Optimization surface of ICONIC [13]. (b) Optimization surface of PLS [9]. (c) Optimization surface of BCE [14]. (d) Optimization surface of our approach.

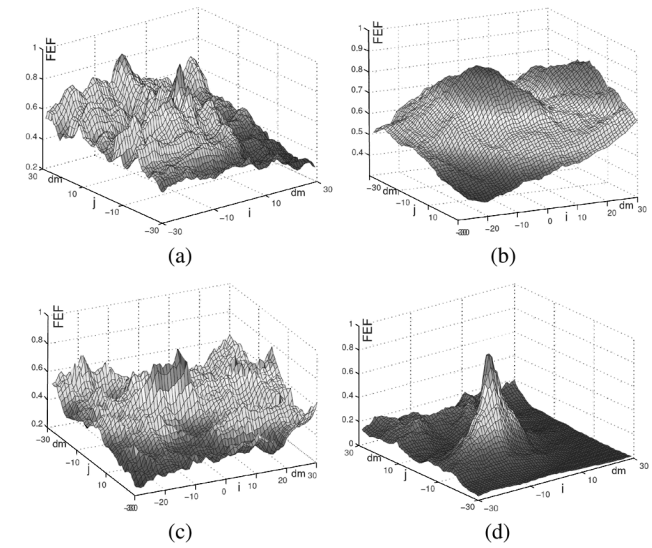


Fig. 7. Optimization-surface comparison of frame 0157. (a) Optimization surface of ICONIC [13]. (b) Optimization surface of PLS [9]. (c) Optimization surface of BCE [14]. (d) Optimization surface of our approach.

and BCE [14] can just perform well with a good initialized pose near the "ground truth." Compared with ICONIC, our approach tried to extract more abundant information with region-based organization, which lead to a more smooth optimization surface. PLS are based on the sum of edge points to projected lines. It cannot lead to sudden changes with a small parameter change. As a result, PLS cannot supply a conspicuous global extreme. BCE also makes use of region information. However, BCE just estimates Gaussian distribution from regions, which loses too much information. In summary, the redundant information, the region-based organization, and the mechanism to evaluate scores are the reasons why our approach supplies the best optimization surface.

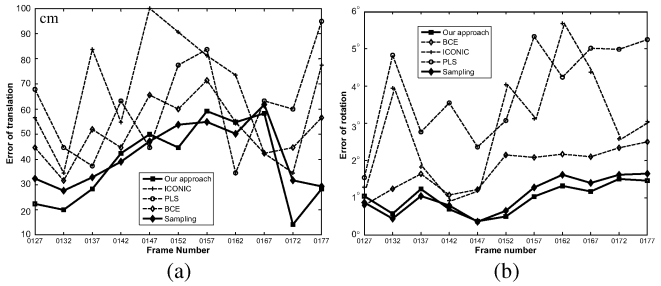


Fig. 8. Comparison of localization accuracy of different methods in selected frames. (a) Comparison of translation error. (b) Comparison of rotation error.

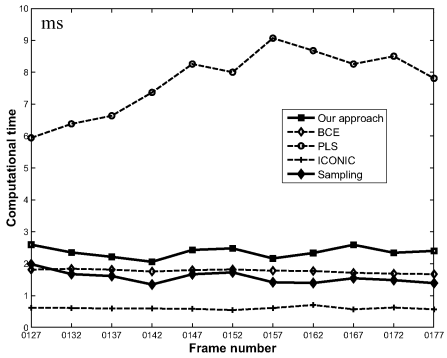


Fig. 9. Comparison of computational time of different methods in selected frames.

2) Accuracy: Accuracy is also an important metric for fitness evaluation, which can be evaluated by the distance between the global extreme of the surface and the "ground truth." We assign $P_E = (x_E, y_E, \theta_E)^T$ as the extreme of the FEF curve, and the localization accuracy is evaluated by these two metrics: the translation error $E_T = ||(X^*, Y^*)^T - (x_E, y_E)^T||$ and the rotation error $E_R = ||\theta^* - \theta_E||$. In this experiment, we adopt further the sampling-based strategy mentioned in Section IV for comparison. The video from frame 0127 to frame 0177 in PETS 2000 Database shows a vehicle running into a scene with different 3-D poses, which is processed to obtain the curve of translation and rotation errors, as shown in Fig. 8. As we can see, our approach also has good performance with better accuracy and stability in localization. The sampling-based strategy does not decrease significantly the accuracy of fitness evaluation.

3) Efficiency: Efficiency is tested by computational cost for fitness evaluation for video frames. For the same video processed with the four methods, the curve of computational time is shown in Fig. 9. Although our approach spends a slightly more time than [13] and [14], it is still quite efficient. The sampling-based strategy can enhance further the efficiency of our approach to be even better than [14] with acceptable accuracy.

4) Summary: A good fitness evaluation approach should be accurate for localization, which is efficient with low calculation cost and convenient for optimization with a smooth optimization surface and conspicuous global peaks. With these three factors in mind, we find that our approach has clear advantages in global performance. Good properties of fitness evaluation can boost greatly the performance of our evolutionary computing framework for model-based vehicle localization and recognition.

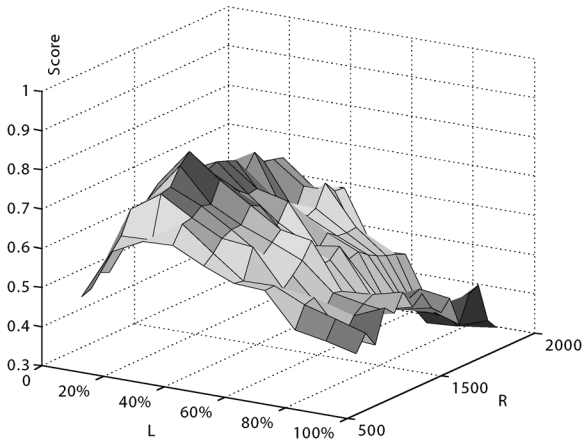


Fig. 10. $R-L-$Score curve (R: the number of population; L: livability; $Score$: the fitness score of the best individual selected by evolutionary computing with parameters R and L).

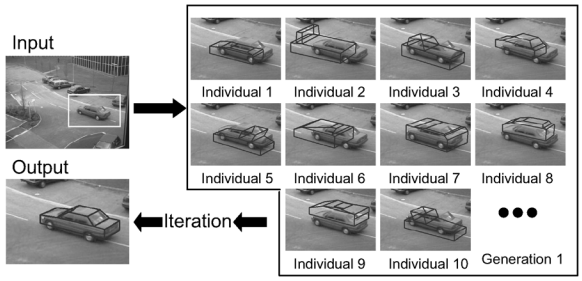


Fig. 11. Illustration of the iteration procedure of a sedan from all kinds of odd structures of the first generation.

B. Parameter Setting of Evolutionary Computing

There are two parameters to be set in our evolutionary computing framework. The first is the number of population R, and the other is livability L, which is calculated as the ratio between the number of selected individuals in every generation N and population R. As has been researched, appropriate R and L values are related to specific problems to be solved. The algorithm may converge to a local extremum with too small population but may increase the computation time if the population is too large. Additionally, appropriate smaller livability facilitates the evolutionary process. In practice, we first take some images as validated examples and sample (R, L) from the 2-D parameter space to choose the best combination. We find that different images give somehow similar results, and we illustrate one example in Fig. 10.

The analysis of the surface shows that good results are given when R is larger than 800, and L is between 10% and 30%. Considering the tradeoff between effectiveness and efficiency, we choose $R = 1000$ and $L = 10\%$.

With the parameters confirmed, the evolutionary computing framework can generate better and better individuals by iteration until convergence. An illustration of the iteration procedure is shown in Fig. 11. As we can see, all kinds of odd structures are generated in the first generation but, at last, converge to the best shape.

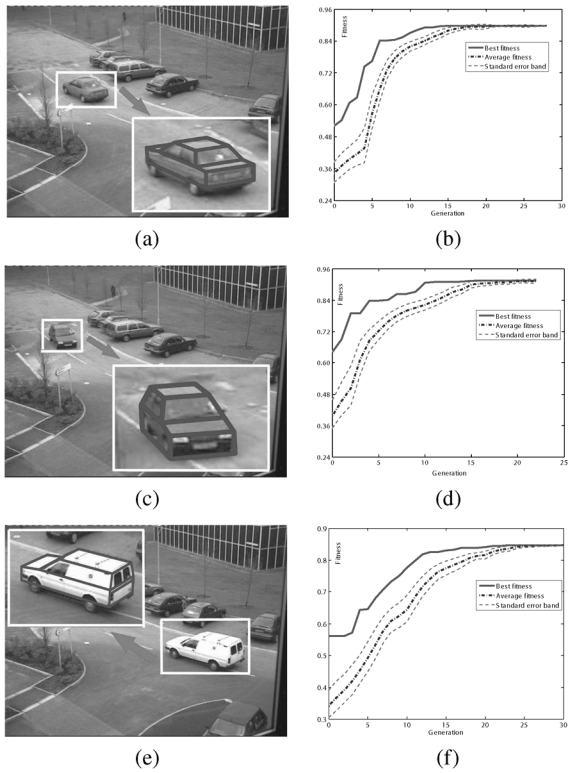


Fig. 12. Illustration of fitting results and evolutionary curves of three kinds of vehicles. (a) Fitting result of the red sedan. (b) Fitting result of the black hatchback. (c) Fitting result of the white SUV. (d) Evolutionary curve of the red sedan. (e) Evolutionary curve of the black hatchback. (f) Evolutionary curve of the white SUV.

C. Illustration of Different Vehicle Types

Experiments are conducted to test the performance of our approach using different kinds of vehicles. Three typical images are chosen first from the PETS 2000 database, [29] with their regions of interest initialized. Good performance has been achieved for all the vehicle, as shown in Fig. 12.

As we can see, the pose of each vehicle is estimated very accurately, and the parameterized 3-D model is deformed automatically to fit image data very well. Furthermore, the iterative curves of best fitness, average fitness, and the standard error band for every vehicle are also shown. We can see that the fitness scores increase very quickly and become stable after about 10–20 generations.

D. Illustration of the Evolutionary Procedure

Furthermore, we take two frames from the above experiment to illustrate the evolutionary procedure of recovering pose and shape parameters. One frame contains a red sedan, whereas the other frame contains a black hatchback. The regions of interest are detected, and the 3-D model fits image data better and better, as shown in Figs. 13 and 14.

E. Effectiveness to Vehicles of Different Poses

We also test the performance of our approach to the same vehicle of different poses. A video of a red sedan passing by the traffic scene is sampled as 12 frames with their fitting results

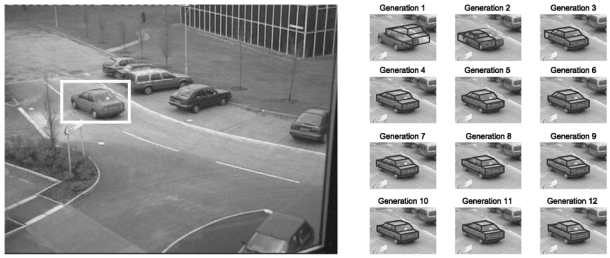


Fig. 13. Evolutionary procedure of the red sedan. (a) One frame of a red sedan. (b) Illustration of evolutionary procedure.

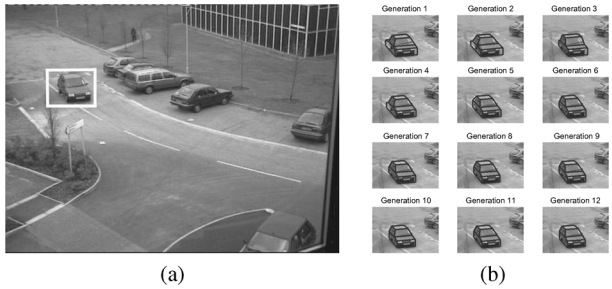


Fig. 14. Evolutionary procedure of the black hatchback. (a) One frame of a black hatchback. (b) Illustration of evolutionary procedure.

shown in Fig. 15, and the recovered shape parameters are listed in Table II.

As we can see from Table II, although all these frames are processed independently with temporal information discarded, the recovered shape parameters of different poses just vary in a less than 5% small range, which demonstrates the effectiveness and stability of our approach.

F. Effect of the Threshold-Based Strategy

As we have described in Section IV, the threshold setting of gradient magnitude has potential to speed up fitness evaluation, enhancing the efficiency of the whole algorithm. However, an inappropriate threshold may affect the accuracy of fitting results and weaken the robustness to noise. We have conducted an experiment to show the effect of threshold setting. As shown in Fig. 16, we mark those pixels whose gradient magnitude below T as black in the second row, and the corresponding fitting results are shown in the third row. It is shown that a small threshold setting is acceptable $(T = 2, 4)$, but a higher threshold setting has more significant effects on shape recovery.

We set $T = 4$ to conduct computation time analysis on PETS 2000 videos. As shown in Fig. 17, the computation time has an about 40% decrease based on the strategy of the threshold setting, which demonstrates the effectiveness of the threshold-based strategy. However, this strategy weaken the robustness of our algorithm to noise and occlusion. In the occlusion cases, we should set $T = 0$ to achieve better accuracy of robustness, which will be described as follows.

G. Occlusion Reasoning

1) Static Occlusion: Static occlusion is the simplest case with a part of objects out of the image border. The sampling strategy can deal with it effectively. After projection of the 3-D model into the image and the removal of hidden lines, the sampled points out of the image border are removed, and the iterations are done in the visible area with normalization. We conduct experiments in two situations. For the first case, vehicles are parked with a part of which being occluded in the visual field. The fitting results are shown in Fig. 18(a) and (b). For the second case, one vehicle is running gradually into the scene with fitting

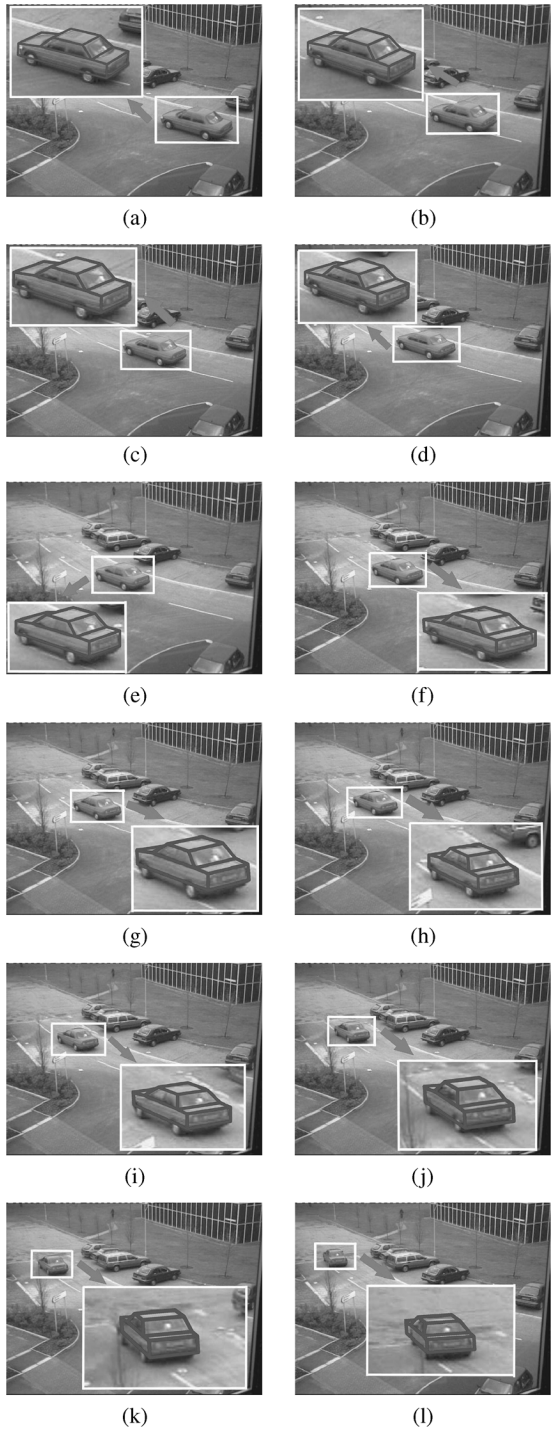


Fig. 15. Results of different poses. (a) Pose 1 (Frame 0127). (b) Pose 2 (Frame 0132). (c) Pose 3 (Frame 0137). (d) Pose 4 (Frame 0142). (e) Pose 5 (Frame 0147). (f) Pose 6 (Frame 0152). (g) Pose 7 (Frame 0157). (h) Pose 8 (Frame 0162). (i) Pose 9 (Frame 0167). (j) Pose 10 (Frame 0177). (k) Pose 11 (Frame 0187). (l) Pose 12 (Frame 0199).

TABLE II
RECOVERED SHAPE PARAMETERS OF DIFFERENT POSES (IN MILLIMETERS)

Shape	$W1$	$H1$	$H2$	L	$H3$	$X1$	$X2$	$X3$	$X4$	$W2$	$H4$	Δ
Pose 1	1624	482	442	3739	527	617	2811	1099	2252	1285	928	227
Pose 2	1607	505	446	3802	531	609	2781	998	2317	1305	918	229
Pose 3	1591	482	466	3679	537	619	2775	1017	2074	1268	916	190
Pose 4	1627	468	486	3695	529	607	2829	1047	2350	1286	933	212
Pose 5	1572	509	459	3592	556	612	2815	1014	2193	1277	911	174
Pose 6	1616	486	462	3733	577	575	2737	1047	2272	1301	914	209
Pose 7	1591	511	480	3669	542	621	2759	1062	2150	1274	929	212
Pose 8	1621	512	434	3761	567	615	2713	986	2221	1257	932	277
Pose 9	1529	504	433	3726	532	612	2906	973	2283	1175	911	231
Pose 10	1634	473	450	3731	576	599	2813	1005	2342	1214	917	242
Pose 11	1594	501	471	3652	544	609	2782	1053	2150	1244	923	189
Pose 12	1526	465	496	3588	597	607	3076	924	2383	1139	936	219
Average	1594	492	460	3697	551	609	2816	1018	2249	1252	922	218
Variance	2.26%	3.55%	4.42%	1.75%	4.18%	2.00%	3.39%	4.57%	4.18%	4.10%	0.97%	1.25%

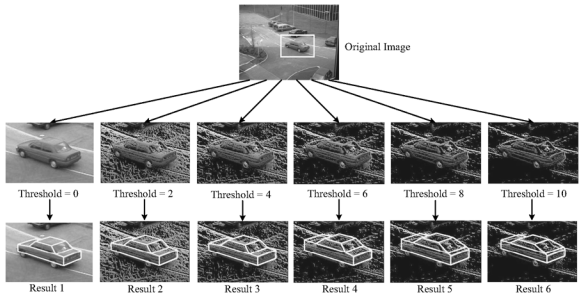


Fig. 16. Illustration of fitting in different threshold settings.

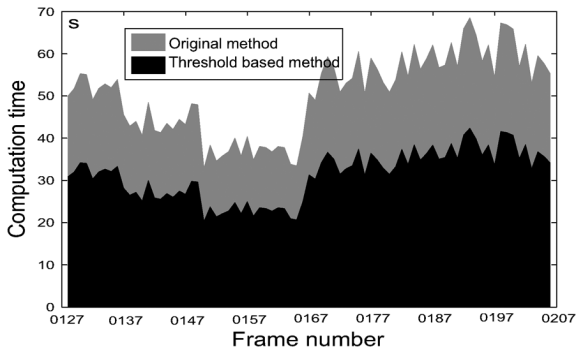


Fig. 17. Comparison of computation time of $T = 0$ and $T = 4$.

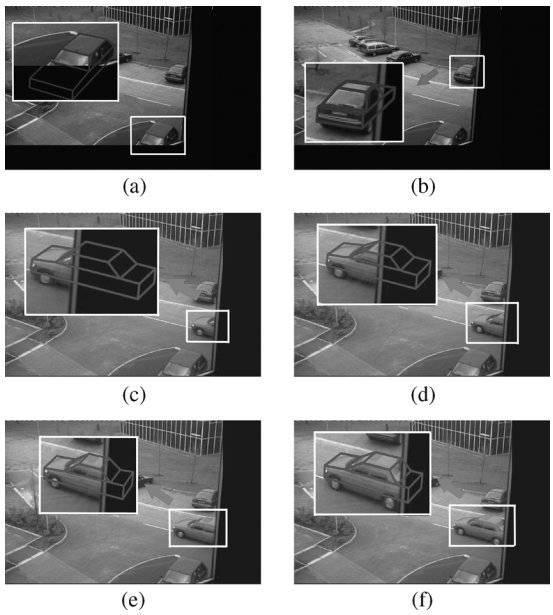


Fig. 18. Results of statistic occlusion. (a)Occluded Hatchback. (b)Occluded car. (c) Frame 0114. (d) Frame 0116. (e) Frame 0118. (f) Frame 0120.

results shown in Fig. 18(c)–(f). As we can see, the whole shape parameters of vehicles have been recovered, and the model fits image data very well in the visible scene. Of course, the parameters that have no evidence from images are assigned some random values within certain ranges evolved by evolutionary computing. Fig. 18(c)–(f) shows a series of frames in which a red sedan is driving into the scene. We can see that the model also fits the frames as expected, and the shape parameters are refined gradually with more and more image evidences.

2) Occlusion by Unrelated Structures: In addition to static occlusion, it is very common that a part of the vehicle is occluded by other objects in images. The case is easier if the structures of other objects are not similar to that of vehicles. The noise generated in the occluded parts would not affect significantly the determination of the evolutionary computing result. One example is a hatchback passing by an artificial object (i.e., a white pillar). The series of images and results are shown in Fig. 19. As we can see, the fitting results are overall acceptable, except some errors occurring when the car's rear end is occluded (Frame 0560). Except for frame 0560, the recovered shape parameters of the other frames are within less than an 8% variance, which demonstrates further the stability of our approach.

3) Occlusion by Related Structures: The occlusion of vehicles by related structures, particularly other vehicles, are the most complicated case. In this case, a part of image evidences are from the vehicle of interest, whereas other image evidences are from other vehicles, and the model may adapt itself inaccurately to fit the inconsistent image evidences. One example of this occlusion is shown in Fig. 20. With four different bounding-box set (1, 2, 3, and 4), different fitting results are obtained. As we can see, our algorithm can deal with the occlusion of related structures to some extent based on stochastic analysis (1, 2, and 3). However, it may fail in

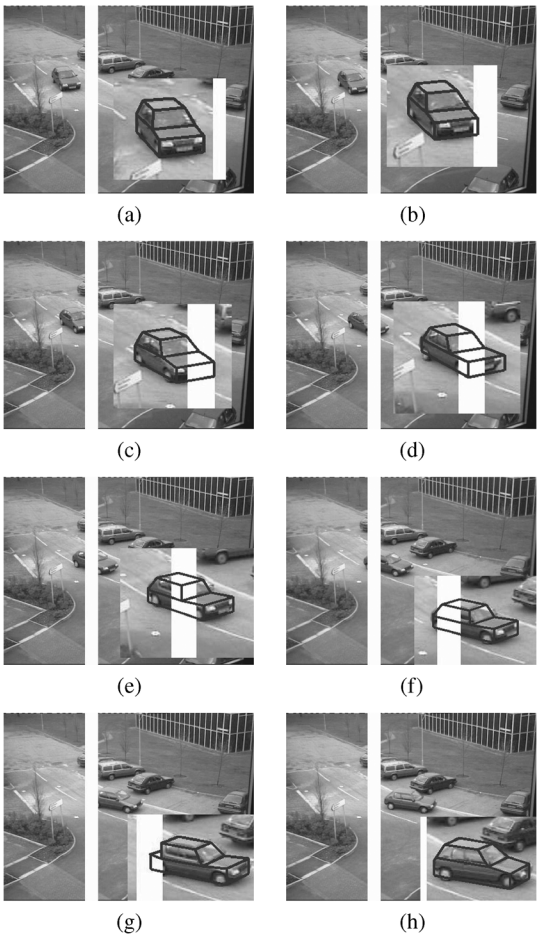


Fig. 19. Results of a black hatchback passing by an artificial object. (a) Frame 0500. (b) Frame 0510. (c) Frame 0520. (d) Frame 0530. (e) Frame 0540. (f) Frame 0550. (g) Frame 0560. (h) Frame 0570.

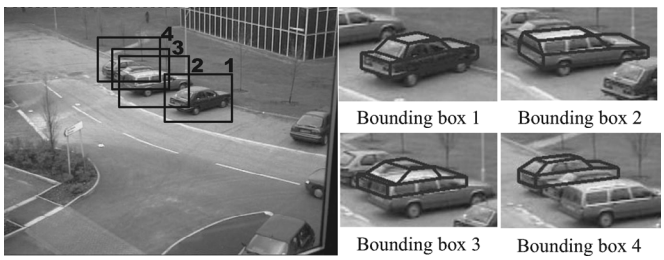


Fig. 20. Illustration of occlusion by related structures.

severe occlusion (4). This experiment also illustrates the effect of bounding-box acquisition to fitting results.

H. Localization and Recognition

From one monocular calibrated image, we can take prior information of the 3-D model to recover the vehicle's 3-D pose parameters based on its 2-D position. The 3-D localization of vehicles has great advantages in high-level computer-vision tasks such as trajectory analysis and semantic interpretation.

With the 12 shape parameters recovered, vehicle recognition becomes quite straightforward. A simple method is based on the distance comparison in the 12-D shape parameter space. In practice, we can collect a set of typical vehicles for each type

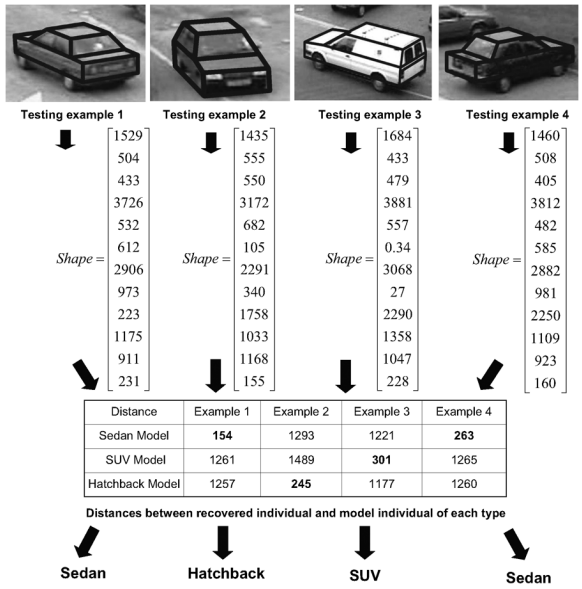


Fig. 21. Illustration of recognition by comparing distances to model individuals of different types.

and calculate their average shape corresponding to the model individual of that type in the shape parameter space. For vehicle recognition, we can recover first the shape parameters of the vehicle, which corresponds to an individual in the shape parameter space, and compare its distance to model individuals of all types. The vehicle is recognized as the type whose model individual is closest to the recovered shape parameters. The four testing examples with their shape parameters recovered are recognized using the Euclidian distance, as shown in Fig. 21.

In addition to the PETS database, we have tested our algorithm for model-based vehicle recognition in our captured image sets with part of which shown in Fig. 22. The set includes vehicles of different types, different view angles, and occlusion in the case of trees, bicycles, pedestrians, and other vehicles. The scenes are calibrated practically based on the motion and appearance information of moving objects in videos with camera height measured as the only one user input. The details of the calibration method refer to [30]. Compared with [28], our algorithm can deal with all these conditions much better to recover the shape and pose parameters of different kinds of vehicles.

I. Discussion

Our method can be extended to vehicle tracking. With the regions of interest detected in several frames, we can recover the shape parameters of the vehicle from every frame to form a more robust average shape, which can be used as a fixed 3-D model. Then, the 3-D model-based tracking can be applied. One of the methods was well illustrated in [9] and has advantages to be more accurate and robust to occlusion. Furthermore, tracking can then feedback to the 3-D modeling with temporal information to recover more accurate and robust 3-D shapes.

As a stochastic-analysis-based method, our algorithm may be affected by many factors. The acquisition of the bounding box is a prior step in our method. Experiments are conducted to test the performance of our algorithm to different bounding-box

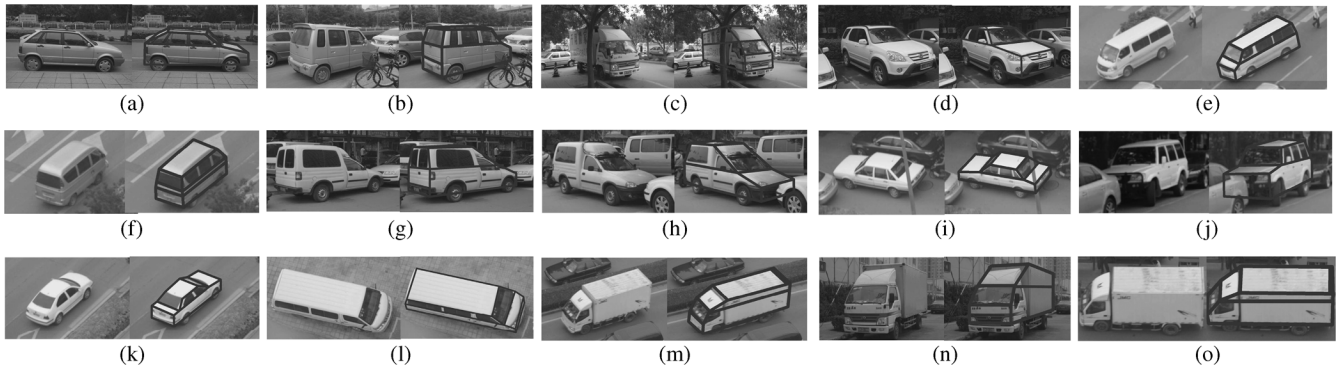


Fig. 22. Illustration of fitting and recognition in all kinds of cases in different views angles and occlusion. (a) Hatchback (sideview). (b) Minivan (occluded by bicycles). (c) Truck (occluded by trees). (d) SUV (occluded by vehicles). (e) Van. (f) Minivan. (g) Pick-up (occluded by vehicles). (h) Pick-up (occluded by vehicles). (i) Sedan (occluded by pillars). (j) SUV (occluded by vehicles). (k) Sedan. (l) Van (top–down view). (m) Truck (view 1). (n) Truck (view 2). (o) Truck (view 3).

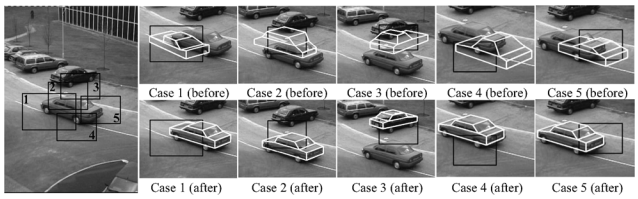


Fig. 23. Illustration of fitting results with different bounding-box setting before and after optimization.

settings with results shown in Fig. 23. For each bounding-box setting, we show the fitting result of the best individual before and after iteration for comparison. As we can see, our algorithm has tolerance to deal with the inaccuracy of vehicle detection in common situations (case 1, 2, 4, and 5) to achieve acceptable fitting results with recovered shape parameters varying in less than 3% ranges, which demonstrates further the stability of the approach. However, in some extreme cases (Case 3) or occlusion, different bounding-box setting may lead to different results or may even fail, which has also been illustrated in Fig. 20. Another factor that may affect the performance is the initialization of pose parameters. The algorithm described in Section III can achieve successful initialization of vehicles in most cases but may fail if the occlusion is too severe. Overall, our algorithm has good tolerance to the inaccuracy of bounding-box acquisition and pose initialization.

VII. Conclusion

A novel algorithm has been described for the model-based localization and recognition of vehicles from monocular images. A deformable model is set up with 12 parameters, and an efficient method based on the image gradient is proposed to evaluate fitness between the projection of the vehicle model and image data. An evolutionary framework is adopted to generate a large number of models based on the deformable model and to choose the best model and position by iterative evolution. The algorithm cannot only realize vehicle localization and recognition but also recover the real shape of different kinds of vehicles. Experimental results demonstrate the effectiveness, robustness, and stability of the algorithm. It can deal with different vehicles, different poses, and static occlusion. Although it is developed in the context of vehicle recognition, the algorithm can be used for many other vision problems.

References

[1] L. Roberts, "Machine perceptions of three-dimensional solids," in *Proc. Opt. Electro-Opt. Inf.*, 1965.
[2] W. Grimson, "The combinatorics of heuristic search termination for object recognition in cluttered environment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 920–935, Sep. 1991.
[3] T. Fan and A. Jain, "Recognizing 3D objects using surface descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 11, pp. 1140–1157, Nov. 1989.
[4] W. Grimson and D. Huttenlocher, "On the sensitivity of Hough transform for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 3, pp. 255–274, Mar. 1990.
[5] D. Lowe, "The viewpoint consistency constraint," *Int. J. Comput. Vis.*, vol. 1, no. 1, pp. 57–72, Mar. 1987.
[6] T. S. Caetano, T. Caelli, D. Schuurmans, and D. A. Barone, "Graphical models and point pattern matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1646–1663, Oct. 2006.
[7] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
[8] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 932–946, Jul. 2002.
[9] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank, "3-D model-based vehicle tracking," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1561–1569, Oct. 2005.
[10] J. R. Beveridge, C. R. Graves, and J. Steinborn, "Comparing random starts local search with key feature matching," in *Proc. 15th IJCAI*, San Francisco, CA, 1997, pp. 1476–1481.
[11] D. Koller, K. Daniilidis, and H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, Jun. 1993.
[12] D. Lowe, "Three dimensional object recognition from two-dimensional images," *Artif. Intell.*, vol. 31, no. 3, pp. 355–395, Mar. 1987.
[13] K. S. Brisdon, "Hypothesis verification using iconic matching," Ph.D. thesis, Univ. Reading, Reading, U.K., 1990.
[14] Q. F. Liu, J. G. Lou, W. M. Hu, and T. N. Tan, "Pose evaluation based on Bayesian classification error," in *Proc. 14th Brit. Mach. Vis. Conf.*, 2003, pp. 409–418.
[15] T. N. Tan, G. D. Sullivan, and K. D. Baker, "Model-based localisation and recognition of road vehicles," *Int. J. Comput. Vis.*, vol. 27, no. 1, pp. 5–25, Mar. 1998.
[16] T. N. Tan and K. D. Baker, "Efficient image gradient based vehicle localization," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1343–1356, Aug. 2000.
[17] J. Ferryman, A. Worrall, G. Sullivan, and K. Backer, "A generic deformable model for vehicle recognition," in *Proc. Brit. Conf. Mach. Vis.*, 1995, pp. 127–136.
[18] R. Fraile and S. J. Maybank, "Comparing random starts local search with key feature matching," in *Proc. Vis. Inf. Inf. Syst. Conf.*, 1999, pp. 697–702.

[19] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 224–237.
[20] Y. Li, L. Gu, and T. Kanade, "A robust shape model for multi-view car alignment," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognition*, 2009, pp. 2466–2473.
[21] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," *Int. J. Comput. Vis.*, vol. 61, no. 2, pp. 185–205, Feb. 2005.
[22] Z. X. Zhang, Y. Cai, K. Huang, and T. Tan, "Real-time moving object classification with automatic scene division," in *Proc. IEEE Conf. Image Process.*, 2007, pp. V-149–V-152.
[23] M. Everingham, A. Zisserman, C. K. I. Williams, L. V. Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, W. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang, "The 2005 PASCAL visual object classes challenge," *Lecture Notes in Computer Science*, vol. 3944, pp. 117–176, Jan. 2006.
[24] Z. Zhang, K. Huang, and T. Tan, "3D model based vehicle localization by optimizing local gradient based fitness evaluation," in *Proc. Int. Conf. Pattern Recog.*, 2008, pp. 1–4.
[25] Z. Zhang, K. Huang, T. Tan, and Y. Wang, "3D model based vehicle tracking using gradient based fitness evaluation under particle filter framework," in *Proc. Int. Conf. Pattern Recog.*, 2010, pp. 1771–1774.
[26] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Proc. 4th Int. Conf. Parallel Probl. Solving Nature*, 1996, pp. 178–187.
[27] P. Larrauaga and J. A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Norwell, MA: Kluwer, 2001.
[28] Z. Zhang, W. Dong, K. Huang, and T. Tan, "EDA approach for model based localization and recognition of vehicles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Minneapolis, MN, 2007, pp. 1–8.
[29] Pets 2000 Data Base [Online]. Available: http://ftp.pets.rdg.ac.uk/PETS2000/
[30] Z. Zhang, M. Li, K. Huang, and T. Tan, "Practical camera auto-calibration based on object appearance and motion for traffic scene surveillance," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

Zhaoxiang Zhang (M'08) received the B.S. degree in electronic science and technology from the University of Science and Technology of China, Hefei, China, in 2004 and the Ph.D. degree in pattern recognition and intelligent systems from the Chinese Academy of Sciences, Beijing, China, in 2009, respectively.

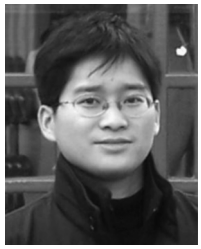
In October 2009, he joined the Laboratory of Intelligent Recognition and Image Processing, Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing, as a Lecturer.

His research interests include computer vision, pattern recognition, image processing, and machine learning.

Tieniu Tan (F'03) received the B.Sc. degree in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 1984 and the M.Sc. and Ph.D. degrees in electronic engineering from Imperial College London, London, U.K., in 1986 and 1989, respectively.

In October 1989, he joined the Computational Vision Group, Department of Computer Science, University of Reading, Reading, U.K., where he worked as a Research Fellow, a Senior Research Fellow, and a Lecturer. In January 1998, he returned to China to join the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China. From 2000 to 2007, he was the Director General with the Institute of Automation, CAS, and since 1998, he has been a Professor and the Director of NLPR. He is also the Deputy Secretary-General (for cyberinfrastructure and international affairs) with CAS. He is the author of more than 350 research papers in refereed journals and conferences in the areas of image processing, computer vision, and pattern recognition. He is also the author or editor of nine books. He is a holder of more than 50 patents. His current research interests include biometrics, image and video understanding, and information forensics and security.

Dr. Tan is a Fellow of the International Association of Pattern Recognition (IAPR). He has served as a Chair or a Program Committee member for many major national and international conferences. He was the Founding Chair of the IAPR Technical Committee on Biometrics, the Founding Chair of the IAPR/IEEE International Conference on Biometrics, and the IEEE International Workshop on Visual Surveillance. He was the Deputy President of the China Computer Federation and the Chinese Automation Association. He is currently the Vice President of the IAPR, the Executive Vice President of the Chinese Society of Image and Graphics, and the Deputy President of the Chinese Association for Artificial Intelligence. He has served as an Associate Editor or member of the editorial boards of many leading international journals, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Pattern Recognition, Pattern Recognition Letters, Image and Vision Computing*, etc. He is the Editor-in-Chief of the *International Journal of Automation and Computing*. He has given invited talks and keynotes at many universities and international conferences and has received numerous national and international awards and recognitions.



Kaiqi Huang (M'09) received the B.Sc. and M.Sc. degrees from Nanjing University of Science Technology, Nanjing, China, and the Ph.D. degree from Southeast University, Nanjing.

Since 2004, he has been with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science, Beijing, China, and in 2005, became an Associate Professor. He is the author of over 70 papers in the international journals and conferences such as T-PAMI, TIP, T-CSVT, T-SMCB, Pattern Recognition, ECCV, CVPR, ICIP, and ICPR. He is in charge of several fundings, including National Science Funding and collaboration with some enterprises. He is also involved in several national research projects (e.g., 863, 973, and NSFC). His current research interests include visual-surveillance digital image processing, pattern recognition, biological-based vision, and so on.

Dr. Huang is the Vice Secretary of IEEE Beijing Section and a member of the IEEE SMC. He also serves as a local Chair and a program committee member of over 40 international conferences and workshops. Technical Committee on Cognitive Computing. He is an Associate Editor of *International Journal of Image and Graphics* and a Guest Editor of the *Signal Processing* special issue.

Yunhong Wang (M'98) received the B.S. degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 1989 and the M.S. and Ph.D. degrees in electronic engineering from Nanjing University of Science and Technology, Nanjing, China, in 1995 and 1998, respectively.

From 1998 to 2004, she was with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Since 2004, she has been a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, where she is also the Director of the Laboratory of Intelligent Recognition and Image Processing, Beijing Key Laboratory of Digital Media. Her research interests include biometrics, pattern recognition, computer vision, data fusion, and image processing.

Dr. Wang is a member of the IEEE Computer Society.