中国科技论文在线 http://www.paper.edu.cn

A Low-Cost Pedestrian-Detection System With a Single Optical Camera

Xian-Bin Cao, Hong Qiao, *Senior Member, IEEE*, and John Keane

***Abstract*—The ultimate purpose of a pedestrian-detection system (PDS) is to reduce pedestrian-vehicle-related injury. Most such systems tend to adopt expensive sensors, such as infrared devices, in expectation of better performance. In comparison, a low-cost optical-camera-based system has much potential practical value, including a greater detection range, and can easily be trained to detect other objects. However, such low-cost systems are difficult to design (e.g., little original information can be collected, and the scene is very complex). To address these problems, an effective and reliable classifier is needed. The classifier should have a proper structure, its features need to be well selected, and a large number of high-quality samples are necessary for training. In this paper, we present a low-cost PDS which only uses a single optical camera. We design a cascade classifier to achieve an effective and reliable detection. First, our system scans two sequential frames at each zoom scale with a sliding window. Second, with each window, both appearance and motion features are extracted. A well-trained cascade classifier, combining statistical learning with a decomposed support-vector-machine classifier, then determines whether the window contains a human body. At the same time, to provide as much information as possible about the pedestrian, a small-scale weighted template tree trained by a coevolutionary algorithm is adopted to identify each pedestrian's direction, and the distance of each from the vehicle is also provided using an estimation algorithm. During the training procedure, we select key features by using the AdaBoost algorithm and a large number of high-quality samples. Experimental results demonstrate that the system is suitable for pedestrian detection in city traffic: The detection speed is more than 10 ft/s, the detection rate reaches 80%, and the false positive rate is no more than 0.3‰.**

***Index Terms*—Cascade classifier, coevolutionary algorithm, decomposed support vector machines (SVMs), pedestrian-detection system (PDS), small-scale weighted template tree, statistical learning.**

Manuscript received February 19, 2006; revised October 18, 2006, March 12, 2007, and May 20, 2007. This work was supported in part by the Chinese Ministry of Science and Technology under Grant 2006AA04Z217 and Grant 2007AA11Z200, by the Selective Support of 100 Outstanding Professors Scholarship, and by the Chinese Academy of Sciences, Chinese National Sciences Foundation, under Grant 60675039. The Associate Editor for this paper was H. Takahashi.

X.-B. Cao is with the Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China, and also with the Anhui Province Key Laboratory of Software in Computing and Communication, Hefei 230026, China (e-mail: xbcao@ustc.edu.cn).

H. Qiao is with the Laboratory of Complex Systems and Intelligent Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China, and also with the School of Informatics, University of Manchester, M60 1QD Manchester, U.K. (e-mail: hong.qiao@mail.ia.ac.cn).

J. Keane is with the University of Manchester, M13 9PL Manchester, U.K. (e-mail: John.keane@manchester.ac.uk).

Digital Object Identifier 10.1109/TITS.2007.909239

I. Introduction

THIS PAPER addresses the challenge of how both to guarantee road safety and to reduce pedestrian-vehicle-related injury. A pedestrian-detection system (PDS) is one approach to this challenge. Most PDSs tend to be based on complicated expensive devices such as an infrared camera [1]–[6] or even radar [7] in order to sense as much information as possible to enable detection. For example, Xu *et al.* [1] proposed a typical PDS with an infrared camera to detect pedestrians in a night scene. Bertozzi *et al.* [2], [3] proposed a system equipped with a pair of infrared cameras which can detect pedestrians in stereo images. The PROTECTOR system (currently SAVE-U) is arguably the most complicated and practicable PDS so far [7]. Not only can it detect pedestrians in real time, but it also traces them and makes risk assessment. However, its performance appears, to a certain extent, to be a consequence of its high-cost hardware (an optical camera, an infrared camera, and five custom-built radars).

On the other hand, a simple and low-cost PDS with a single optical camera is still worth investigating due to its potential practical value. Compared with the expensive systems, the low-cost system with normal cameras has remarkable advantages: 1) It can obtain fast and reliable performance if well designed; 2) it has a greater detection range; and 3) it can easily be trained to detect other objects, regardless of whether they have temperature characteristics.

However, such a low-cost system is difficult to design. The most difficult issue is the scene complexity, and also, little practicable information can be gathered with only one optical camera. In general, a PDS needs at least the techniques of feature extraction and classification. Therefore, an effective and reliable classifier is needed, and enough features must be extracted. First, the classifier should have a proper structure, its features need to be well selected, and a large number of high-quality samples are necessary for training. If the classifier is poorly designed or trained, it will have a low positive rate or high false positive rate when it has a high detection speed. Conversely, the requirement of high positive rate and low false positive rate will lead to a low-speed classifier and makes real-time detection impossible. Second, features such as appearance and motion need to be extracted (most systems are based either only on appearance [8], [9] or motion [10]–[14], and very few use both appearance and motion features [15]–[17]). Therefore, it is hard to design a single-optical-camera-based PDS for practical use in which both the detection rate and the detection speed are accepta

转载

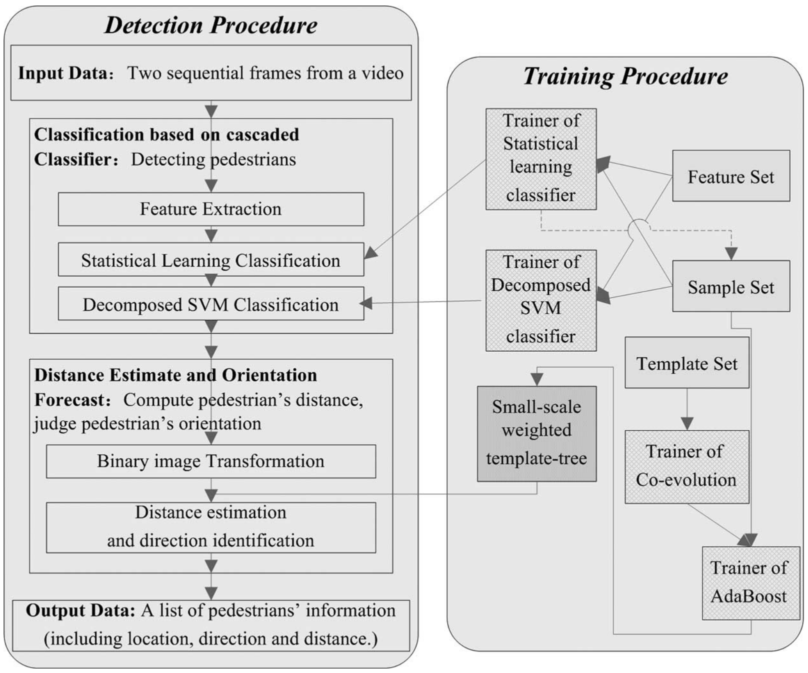

Detection Procedure
Input Data: Two sequential frames from a video
Classification based on cascaded Classifier: Detecting pedestrians
Feature Extraction
Statistical Learning Classification
Decomposed SVM Classification
Distance Estimate and Orientation Forecast: Compute pedestrian's distance, judge pedestrian's orientation
Binary image Transformation
Distance estimation and direction identification
Output Data: A list of pedestrians' information (including location, direction and distance.)
Training Procedure
Trainer of Statistical learning classifier
Feature Set
Trainer of Decomposed SVM classifier
Sample Set
Template Set
Small-scale weighted template-tree
Trainer of Co-evolution
Trainer of AdaBoost

Fig. 1. Architecture of our PDS.

There are several PDSs which are based on a single optical camera. For example, Gavrila [18] proposed a car-mounted system. He adopted a chamfer system to select candidates using edge features and then applied a pattern classification such as radial basis functions with a richer set of intensity features to verify the candidates. However, he only managed to locate a pedestrian with a low speed and a high false positive rate. The reason for this may be that the features are not properly selected and that the system performance is not good for both the detection rate and the detection speed. Moreover, Viola *et al.* [19] proposed a PDS for static scenes which was the first to use both the appearance and motion information in the detection phase. They also proposed the integrated image algorithm, which is an extremely efficient method to represent image motion. The AdaBoost algorithm is also introduced in [19] to train a statistical-learning classifier with huge samples. The system could even operate on low-resolution images under difficult conditions (such as in rainy or snowy weather). However, using a single statistical-learning classifier can only achieve a high detection rate. The detection speed of the system proposed in [19] is 4 ft/s, which cannot meet the real-time detection demands; therefore, new methods are needed. Further, this is not a vehicular-based system.

Pedestrian-detection methods can not only be used for driving assistance but also for other areas such as automatic pedestrian-counting security systems, traffic/pedestrian control systems, and automatic switching systems. The main difference is that for driving assistance, a PDS is based on the information from a moving camera, and for other applications, most PDSs are based on the information from a static camera. This causes technical differences between the PDS for driving assistance and for other applications.

For other applications using the static cameras, the feature used in PDS is also mainly based on the shape or motion information. Furthermore, there are many other features adopted to assist detection. For example, Shashua *et al.* [20] proposed a PDS which took nine main parts of a human body and their position relations as key features to detect pedestrians, Havasi *et al.* [21] designed a special PDS mainly using the symmetry of human legs to detect walking, and Andrade *et al.* [22] provided an optical flow-based PDS for automatic pedestrian counting.

In this paper, we design a vehicular- and single-optical-camera-based PDS. The system detects pedestrians using a cascade classifier using both the appearance and motion features. The detection procedure is as follows: 1) Obtain the appearance and motion information from sequential frames (only the region of interest (ROI) is dealt with in the original frame pair); 2) recognize a human body in each sliding window using a combination of a statistical-learning classifier and a decomposed support-vector-machine (SVM) classifier; and 3) estimate the distance from the vehicle of each detected pedestrian, and identify his/her direction. The system is suitable for pedestrian detection within 25 m from the camera with a vehicle speed that is not faster than 60 km/h in the daytime.

The remainder of this paper is arranged as follows. Section II describes in detail the architecture and the detection procedure of our system. Section III introduces the training procedure. Section IV shows the experimental design and results, and finally, this paper is concluded in Section V.

II. Dectection Procedure of the System

The architecture of our system is shown in Fig. 1. As described in the left half of the figure, the detection procedure contains a cascaded classifier module that takes charge of the pedestrian detection and another module that deals with

中国科技论文在线 http://www.paper.edu.cn

distance estimation and direction identification. The detection procedure is described as follows.

```
while (the vehicle is moving)
{
    intercept the next pair of sequential frames from the video flow;
    while (not all the sliding windows have been detected)
    detect the pedestrian in the next sliding window;
    /* perform an exhaustive scan on the ROI at every zoom scale */
    estimate the detected pedestrians' distance, and identify his/her direction;
    output the detection result;
    /* pedestrian's location, distance, and orientation */
};
```

Furthermore, in order to accelerate the detection process, we only deal with a particular region of the original frame pair. This ROI is a rectangular region in which the pedestrians might cause a collision with the vehicle.

A. *Classification Based on a Cascaded Classifier*

The general detection stage is described as follows.

Step 1) From the recorded video, intercept the ROI from two sequential frames. The following process only deals with the ROI.

Step 2) Select a constant $Z (0 < Z < 1)$, set the initial value of N to be zero, and it is increased by one for each iteration. The process ends when $N = 7$. The zoom factor is Z^N (e.g., when $Z = 0.8$, the zoom factor serials are 1, 0.8, 0.64, 0.51, 0.41, 0.33, 0.26, and 0.21).

Step 3) Move the sliding window on the zoomed ROI, and then, use a cascaded classifier to identify the pedestrian figures. Go to Step 2).

The details of Step 3) are described as follows.

Step 3.1) Intercept two images in the sliding window from the zoomed ROI; the sliding window moves from the left top to the right bottom of the ROI.

Step 3.2) Extract both the appearance and motion features from the two images using the shifting and subtracting image techniques [9].

Step 3.3) Use these features and the cascaded classifier to detect a pedestrian figure.

Step 3.4) If a pedestrian figure is detected, output his/her information for further direction identification and distance estimation.

In this module, Step 3.3) is the most important. The cascaded classifier judges whether there is a pedestrian in each sliding window. The cascaded classifier is a combination of two different classifiers: The first part is a statistical-learning classifier, which aims to select preliminary candidates; the second part is a decomposed SVM classifier [1], [23]–[25], which carries out an accurate classification. The reason to have such a cascade is that, during the tests, we found that the statistical-learning classifier was very fast and had a low false negative rate, but its false positive rate was high. Comparatively, the decomposed SVM classifier had a very high positive rate and a low false positive rate, but it was much slower; hence, we cascade the two classifiers. The statistical-learning classifier acts as the front of the cascade to achieve very quick scanning in order to reduce the amount of the target area for consideration by the decomposed SVM classifier, which then performs an accurate classification, thus improving both detection rate and speed.

The statistical-learning classifier works as follows.

1) Compute the key features of a sliding window as the input.
2) Calculate each classification function of each filter, and then, get ten values $f_i(x)$; each filter has a weight ω_i.
3) Calculate $f(x) = \sum_{i=1 \text{ to } 10} \omega_i f_i(x)$. If $f(x) \geq \theta$ (θ is a threshold), then it is a pedestrian; otherwise, it is not.
4) If it is a pedestrian, add it to the output chain; else, do nothing.

As to ω_i and θ, they are all trained by the AdaBoost algorithm [26], [27]. At the same time, the proper subset of features has also been selected.

After the statistical-learning classifier phase, there remain about 100 candidates to be considered further in the ROI. Then, the decomposed SVM classifier will do a precise classification. It works as follows.

1) Compute the key features of a candidate as the input vector.
2) Calculate $f(x) = \text{sgn}(\sum_{\text{Support Vector}} y_i \alpha_i k(x_i, x) - b)$; here, $k(x, y) = (x \cdot y + 1)^2$.
3) If $f(x) = 1$, then it is a pedestrian; otherwise, it is not;
4) If it is not a pedestrian, remove it from the output chain; else, do nothing.

B. *Distance Estimation and Direction Identification*

Usually, a PDS with a single optical camera can obtain relatively little information from the images. Until now, only few systems have accessory functions to estimate a pedestrian's distance from the vehicle and to identify his/her direction. However, in the system presented here, we both estimate the distance of a pedestrian according to the zoom scale and develop a distance-transform (DT) algorithm [18], [28], [29] to identify direction.

Gavrila [30] has summarized several methods for direction identification. All these methods are complex and time-consuming; hence, we propose a simple method instead.

1) We design a small-scale weighted template tree. A template is a typical human shape obtained from real traffic. Obviously, we need to select several templates for each direction. Different templates have different characteristics; therefore, each template needs to be weighted.
2) To forecast a pedestrian's direction, we match the detected pedestrian with each template in the template tree, and the pedestrian's orientation is then considered to be the same as the most similar templates. In the matching process, we apply a DT algorithm to scale the similarity of two binary images (gray-scale images must be converted to binary images for the DT algorithm first).

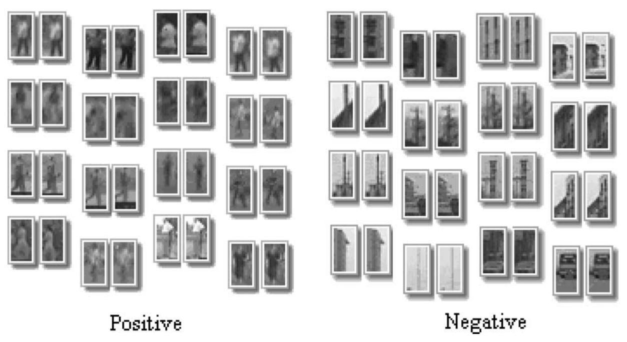


Fig. 2. Samples with a dimension of 32 × 16; a pair of images comprises a single example for training. The ones in 24 × 12 are similar.

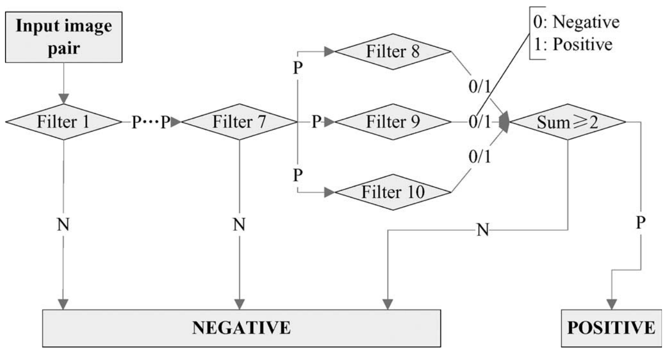


Fig. 3. Architecture of the statistical-learning classifier.

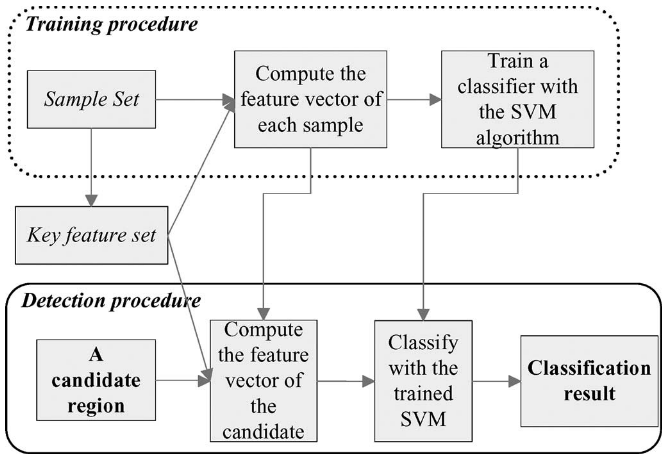

Training procedure
Sample Set
Compute the feature vector of each sample
Train a classifier with the SVM algorithm
Key feature set
Detection procedure
A candidate region
Compute the feature vector of the candidate
Classify with the trained SVM
Classification result

Fig. 4. Training and detection procedures of the decomposed SVM classifier.

III. Training of the System

A. Feature Extraction

The performance of a classifier strongly depends on the features adopted. To extract both the appearance and motion features, Viola and Jones [26] proposed an extremely effective method, and in this paper, we use a similar method to extract the features for each classifier.

B. Sample Formulization

We cascade the two classifiers to accelerate detection speed. In order to improve the detection rate, the classifiers must first be well trained.

For a classifier, large numbers of high-quality samples are very important. In the training process, first, we obtain a large number of various pedestrian/nonpedestrian (negative/positive) samples. Due to the need for motion information, we use two grayscale images extracted from the same place in consecutive frames to form a training sample. As we use a fixed-size sliding window to detect pedestrians, all samples should be scaled to the same size as the sliding window. We prepared samples in both 32 × 16 and 24 × 12 for the comparisons of experiments and determine which is better.

For each size, we have produced 3600 positive sample pairs and 3000 high-quality negative sample pairs from large amounts of video of real city traffic. Here, high-quality means that the negative samples are very similar to a human body, such as trees, etc. We also automatically produced a large number of negative sample pairs (about 1 000 000 pairs). All of these samples were used to train the statistical-learning classifier, whereas only the high-quality negative sample pairs and the false positive sample pairs from the statistical-learning classifier were used to train the decomposed SVM classifier.

Examples of some training samples in our system are shown in Fig. 2 (the training sample and the test-video database are available at http://nical.ustc.edu.cn/PDS/).

C. Statistical-Learning Classifier

The statistical-learning classifier is itself a cascaded system composed of ten filters, as shown in Fig. 3. Appearance features are used in the first seven cascades, which are organized in series. In order to increase the positive rate, we add motion features to the classifier because motion is an attribute of the pedestrian. The motion features are adopted in the other three filters, which are parallel connected. We trained each filter with a positive rate no less than 0.95, whereas the false positive rate of filter i must be no more than $0.1 + (i - 2) \times 0.05$ (‰).

In the first seven filters, only when an object is estimated as a positive one by the present filter that it can be sent to the next one; otherwise, it will be estimated as a negative one by the classifier. After the object passes through all the first seven filters, it is separately assessed by the other three filters. The object will be considered as a positive one only when at least two of the posterior filters estimate it as positive.

In our system, the statistical-learning classifier is used to select candidates for the decomposed SVM classifier; therefore, it must be fast with a very low false negative rate, and the false positive rate needs to be as low as possible.

D. Decomposed SVM Classifier

The training and detection procedures of the decomposed SVM classifier are shown in Fig. 4.

The training procedure can be described as follows.

Step 1) Select a feature subset.

For each sample, after the original feature set is obtained, we apply the AdaBoost algorithm to select a subset from it. Suppose that the size of the subset is n and that each feature c_k corresponds to a feature

中国科技论文在线 http://www.paper.edu.cn

function $f_k(A, B)$, $k = 1, \ldots, n$, where A and B are the bitmap matrices representing a sample.

Step 2) Compute the feature vector of each sample.

Before the information of a sample is used by the decomposed SVM, it must be converted to a feature vector. We can use the feature functions to compute it. For example, we use a pair of bitmap matrices A_i and B_i to represent the sample S_i, where $i = 1, \ldots, l$, and l is the number of training samples.

Vector $x_i = (f_1(A_i, B_i), f_2(A_i, B_i), \ldots, f_n(A_i, B_i))$ is computed and treated as the feature vector of S_i.

Step 3) Train a classifier using a decomposed SVM.

Here, we adopt a decomposition SVM algorithm proposed by Joachims [25]. Using Joachims' decomposed method requires the size of working set and the kernel function to be determined beforehand. In our training, we choose 20 as the size of the working set, and the kernel function is $k(x, y) = (x \cdot y + 1)^2$.

E. Training a Small-Scale Weighted Template Tree

In our system, in order to identify a pedestrian's direction of movement, we design a small-scale weighted template tree. By matching a pedestrian with all the templates in the tree, we obtain the probable direction of the pedestrian.

The tree is well organized with typical templates. A template is the edge of a typical human shape obtained from the real traffic, and its size is the same as the sliding window. In the tree, when a detected pedestrian is matched with a template, we obtain only his/her preliminary direction, and it may be inaccurate. In order to increase the correctness of identification, it is necessary to select various templates; however, this needs to be a small enough set of templates to ensure processing speed. Therefore, we select 30 representative templates. As different templates have different characteristics, their ability to describe a pedestrian's orientation is also different. Therefore, each template needs to be accurately weighted.

During the identification of the pedestrian's direction, we apply a DT algorithm [18], [28], [29] to match a pedestrian with the templates; for the DT algorithm, we must efficiently scale the similarity between two binary images. The DT algorithm determines a DT value of two images of the same size, and if the value is small, then the two images are similar.

In the identification procedure, we match the pedestrian with all the templates in each direction and then determine the summation of the weighted DT values in each direction. We regard the pedestrian's moving orientation as the direction which has the smallest DT summation.

Our system is able to distinguish pedestrians in three directions: left, right, and middle. Several typical templates in three directions are separately shown in Fig. 5.

Populating the weighted template tree to obtain typical templates and associated weights is important as there are about 4000 types of template, and typically, less than 50 will be chosen. Optimization methods can solve this problem; however, in contrast to other approaches, a new coevolutionary algorithm,

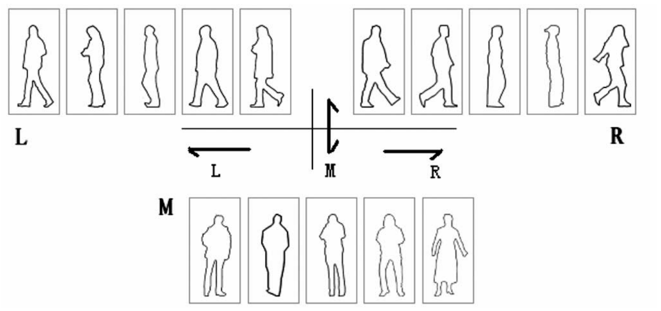


Fig. 5. Templates (with a dimension of 32 × 16) in three directions.

which has been mentioned in our previous work [29], is used to efficiently build the template tree in our system. This algorithm has the following advantages.

1) The algorithm is evolution-based and is suitable for the cases where the search space is large. As the amount of templates is quite large, the algorithm is relatively effective.
2) Compared with the traditional evolution-based algorithms, the coevolutionary algorithm integrates the information at individual level for generations; thus, it has good specialization, generalization, and efficiency.
3) As it is known, population size is a key factor in an evolution-based algorithm because it determines the population state space and then affects the algorithm's convergent performance. If the population size is too small, the algorithm may suffer from premature convergence and lose the global optimum; if the population size is too big, it results in excess computation. The coevolutionary algorithm used here introduces a strategy to self-adaptively and globally asymptotically adjust the subpopulation size. Thus, the algorithm possesses the ability to maintain the proper subpopulation size.

The generating procedure of the template-searching tree can be described as follows.

With the coevolutionary algorithm, the template tree will be built by a clustering method. For the current template set, templates which are highly similar (and together have a low DT value) will be clustered into one group. In each clustered template group, a representative template will be selected and act as the father node of the group. The set of father nodes then becomes the template set for the next iteration. This procedure will be repeated until the size of the current template set reduces to one, and then, the template tree is built bottom-to-top. The following algorithm is used to find a high-quality group division in each run of the procedure.

1) Because the templates are divided into three types, the template tree can be divided into three subtrees. We use one subpopulation for the tree construction of each template type, and then, three subpopulations are used in the coevolutionary algorithm. We initialize three subpopulations for the algorithm, where an individual is a solution of the group division.
2) The fitness value of an individual is computed by the sum of DT values in all groups, which is denoted as $f(x)$. Individuals with lower fitness value are better solutions.

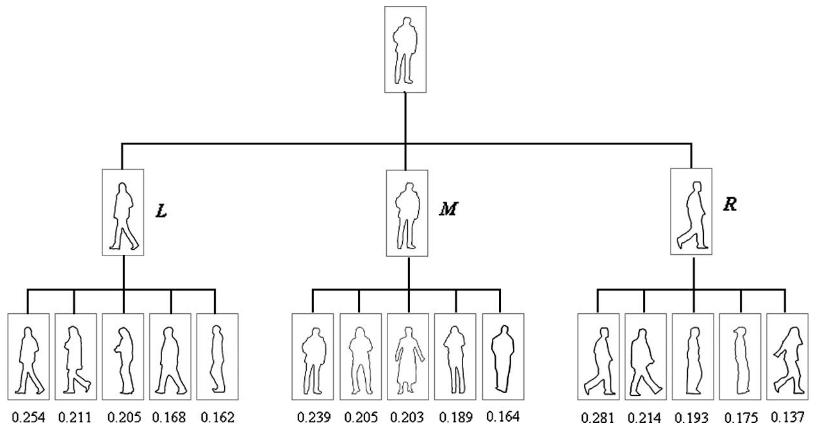


Fig. 6. Small-scale weighted template tree with only three levels.

3) For each pair of individuals (a, b), $a \neq b$ in the same subpopulation, the worst one may be eliminated with the probability of α, and the better one has a chance (denoted as β) to produce a bonus offspring. To prevent a subpopulation from increasing indefinitely, β should be less than α.
4) Between different subpopulations, for each pair of individuals (a, b), let coefficient w_{ab} denote the influence of b upon a. We define $w_{ab} = (f(b) - f(a))/(N(f^{\max} - f^{\min}))$, where N is the total number of individuals in all subpopulations, and $f^{\max}$ and $f^{\min}$ are the maximum and minimum of fitness values in all subpopulations of the current generation.

   Obviously, w_{ab} is positive when b is inferior to a and negative when b is superior to a. Similarly to internal evolution, a positive w_{ab} will give a additional chances, which are equal to w_{ab}, to produce a bonus offspring, whereas a negative w_{ab} will cause a to be eliminated with the probability of $-w_{ab}$.
5) For each individual $a \in P_i$, it has a chance (denoted as ρ) to produce an offspring.
6) Repeat steps 3) to 5).

After the template-searching tree is built, we select several top levels as the small-scale template tree and then weigh the templates in each direction. We use the AdaBoost algorithm to train the template tree with the positive samples, obtaining the weight of each template. The training procedure is similar to that of the statistical-learning classifier.

An example of the weighted template tree is shown in Fig. 6.

IV. Experiments

A. Preparation

We have designed three separate experiments to assess the performance of the proposed PDS. The first experiment is to test the detection performance of our system, the second experiment is to verify the effectiveness of the motion features, and the performance of an individual classifier was also tested; the final experiment is to assess the performance of the distance estimation and the direction identification. All experiments were carried out on a Pentium IV 2.8-GHz computer with 512-MB double data rate RAM, and most test materials were captured with a digital vidicon (Sony DCR-HC21E).

To prepare for the experiments, in order to train the classifiers and test the performance of the proposed system, we captured more than 30 h of video in an urban area (Hefei, China) from a moving vehicle. The average speed of the vehicle was 45 km/h, and the digital vidicon was fixed in the car.

By using part of the video records, we manually made 3600 positive sample pairs and 3000 negative sample pairs. In addition, 1 000 000 negative sample pairs are automatically generated. These samples are used for classifier training.

The rest of the video records were used to evaluate the performance of an individual classifier. There were 476 video clips used; each had 450 frames (30 ft/s and was 15 s long). The detection performance mainly depends on the traffic condition; hence, we divided the video clips into seven types, including one type that has no pedestrian in the video record.

The key parameters were set as follows. The verification videos were in 320 × 240 resolution, and the size of the ROI was 240 × 120 (pixel × pixel); with a zoom factor of 0.8, the ROI needed to be zoomed seven times for 32 × 16 sliding window or eight times for 24 × 12 window.

B. Experiment 1: Detection-Performance Verification

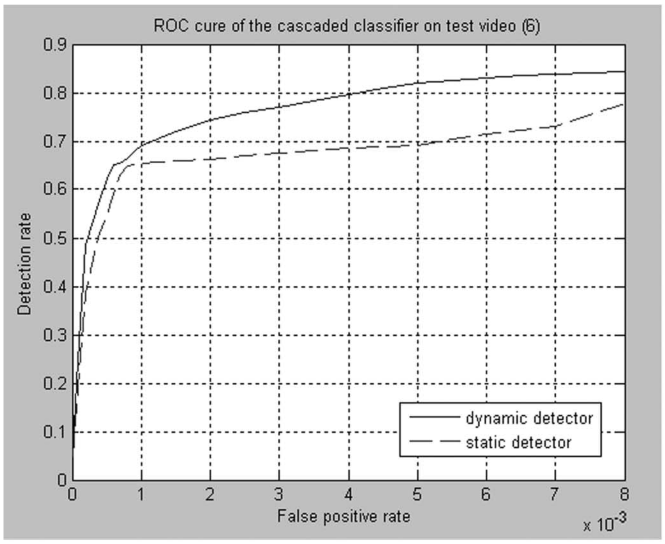
This experiment was designed to test the main function of our system, and the detection performance was evaluated by three key parameters: detection rate, false positive rate, and detection speed. In order to thoroughly test the system, both the training samples and the verification videos of urban traffic were chosen. The overall results of pedestrian detection with both sliding window sizes (24 × 12 and 32 × 16) for 1000 positive samples and 3000 negative samples, as well as seven typical type test videos, are listed in Table I.

The experimental results in Table I indicate the following.

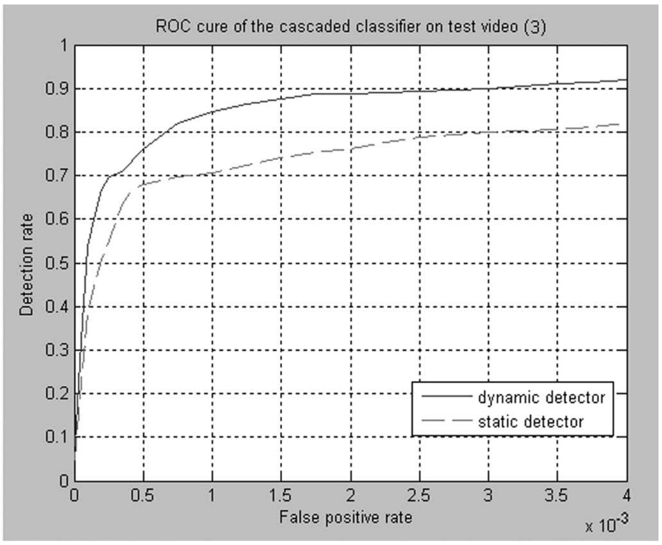
1) The system achieves a good performance with both sample sizes: 24 × 12 and 32 × 16. With videos of real city traffic, the detection speed is more than 10 ft/s, the detection rate reaches about 80%, and the false positive rate is no more than 0.3‰.

TABLE I
SYSTEM DETECTION PERFORMANCE WITH DIFFERENT SAMPLES IN SIZE

Size of sample \ Resources		Samples from training set		Videos of real city traffic							
		positive samples (1000 pairs)	**negative samples (3000 pairs)**	Type 0	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	**Average**
24×12	Detection rate (%)	**99.5**	—	—	84.4	78.6	80.8	83.6	76.6	75.4	**79.9**
	False positive rate(‰)	—	**1.0**	2.00	0.96	1.92	1.42	0.98	1.80	2.25	**1.90**
	Detection speed(fps)	—	—	11.5	10.2	11.3	10.9	11.2	11.3	11.6	**11.0**
32×16	Detection rate (%)	**99.9**	—	—	92.4	82.6	80.8	93.6	90.6	81.0	**86.8**
	False positive rate(‰)	—	**0.3**	0.15	0.05	0.14	0.10	0.06	0.20	0.25	**0.13**
	Detection speed(fps)	—	—	13.0	12.8	11.7	12.4	12.1	11.6	11.8	**12.2**



(a)



(b)

Fig. 7. ROCs of the cascaded classifier with different test videos and samples in different sizes. (a) ROC with test video (6) and samples in size of 24 × 12. (b) ROC with test video (3) and samples in size of 32 × 16.

2) Compared with the 24 × 12 samples, which have a detection range of 35 m (the detection range is the maximum distance that the pedestrians can be away from the car), the detection speed of the 32 × 16 ones is even faster (the classifier works less because it zooms a few times), and the detection rate is obviously increased, although the detection range reduces to approximately 25 m.

The system finally selected the 32 × 16 templates because 25 m is already enough to assist driving in real traffic when the vehicle speed is less than 60 km/h.

C. Experiment 2: Effectiveness of Motion Features and Performance of Each Classifier

An important issue is whether the motion filer was efficient. Hence, we tested the system performance with/without motion features, and the comparative results shown the effectiveness of the motion features.

With two kinds of sample [24 × 12 and 32 × 16 (pixel × pixel)], the rate of change (ROC) cures of the cascaded classifier with some test videos is shown in Fig. 7. Fig. 7 showed that the detection rate significantly increases with the dynamic detection (using both the appearance and motion features) than with the static detection (using only the appearance features). Hence, the motion features enhance the detection ability of the system.

Further, in order to compare the performance of the combination of two classifiers with the individual classifiers, we separately present the performance of the individual stages, i.e., the statistical-learning stage and the decomposed SVM stage, as shown in Fig. 8. The corresponding average performances are listed in Table II.

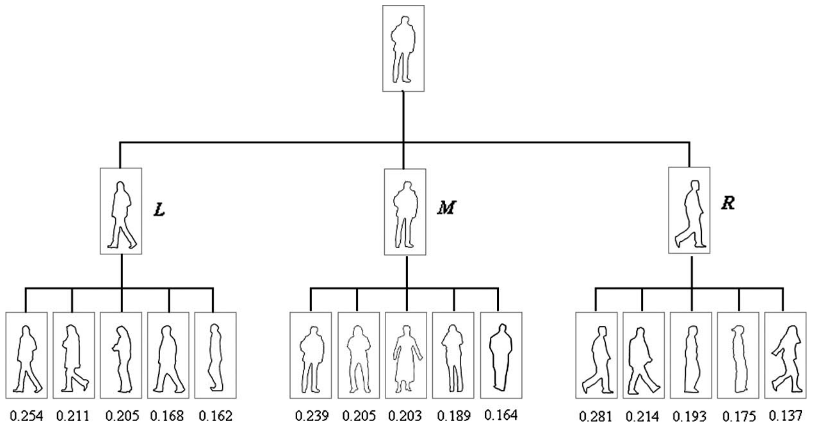


Fig. 6. Small-scale weighted template tree with only three levels.

3) For each pair of individuals (a, b), $a \neq b$ in the same subpopulation, the worst one may be eliminated with the probability of α, and the better one has a chance (denoted as β) to produce a bonus offspring. To prevent a subpopulation from increasing indefinitely, β should be less than α.
4) Between different subpopulations, for each pair of individuals (a, b), let coefficient w_{ab} denote the influence of b upon a. We define $w_{ab} = (f(b) - f(a))/(N(f^{\max} - f^{\min}))$, where N is the total number of individuals in all subpopulations, and $f^{\max}$ and $f^{\min}$ are the maximum and minimum of fitness values in all subpopulations of the current generation.

   Obviously, w_{ab} is positive when b is inferior to a and negative when b is superior to a. Similarly to internal evolution, a positive w_{ab} will give a additional chances, which are equal to w_{ab}, to produce a bonus offspring, whereas a negative w_{ab} will cause a to be eliminated with the probability of $-w_{ab}$.
5) For each individual $a \in P_i$, it has a chance (denoted as ρ) to produce an offspring.
6) Repeat steps 3) to 5).

After the template-searching tree is built, we select several top levels as the small-scale template tree and then weigh the templates in each direction. We use the AdaBoost algorithm to train the template tree with the positive samples, obtaining the weight of each template. The training procedure is similar to that of the statistical-learning classifier.

An example of the weighted template tree is shown in Fig. 6.

IV. EXPERIMENTS

A. Preparation

We have designed three separate experiments to assess the performance of the proposed PDS. The first experiment is to test the detection performance of our system, the second experiment is to verify the effectiveness of the motion features, and the performance of an individual classifier was also tested; the final experiment is to assess the performance of the distance estimation and the direction identification. All experiments were carried out on a Pentium IV 2.8-GHz computer with 512-MB double data rate RAM, and most test materials were captured with a digital vidicon (Sony DCR-HC21E).

To prepare for the experiments, in order to train the classifiers and test the performance of the proposed system, we captured more than 30 h of video in an urban area (Hefei, China) from a moving vehicle. The average speed of the vehicle was 45 km/h, and the digital vidicon was fixed in the car.

By using part of the video records, we manually made 3600 positive sample pairs and 3000 negative sample pairs. In addition, 1 000 000 negative sample pairs are automatically generated. These samples are used for classifier training.

The rest of the video records were used to evaluate the performance of an individual classifier. There were 476 video clips used; each had 450 frames (30 ft/s and was 15 s long). The detection performance mainly depends on the traffic condition; hence, we divided the video clips into seven types, including one type that has no pedestrian in the video record.

The key parameters were set as follows. The verification videos were in 320 × 240 resolution, and the size of the ROI was 240 × 120 (pixel × pixel); with a zoom factor of 0.8, the ROI needed to be zoomed seven times for 32 × 16 sliding window or eight times for 24 × 12 window.

B. Experiment 1: Detection-Performance Verification

This experiment was designed to test the main function of our system, and the detection performance was evaluated by three key parameters: detection rate, false positive rate, and detection speed. In order to thoroughly test the system, both the training samples and the verification videos of urban traffic were chosen. The overall results of pedestrian detection with both sliding window sizes (24 × 12 and 32 × 16) for 1000 positive samples and 3000 negative samples, as well as seven typical type test videos, are listed in Table I.

The experimental results in Table I indicate the following.

1) The system achieves a good performance with both sample sizes: 24 × 12 and 32 × 16. With videos of real city traffic, the detection speed is more than 10 ft/s, the detection rate reaches about 80%, and the false positive rate is no more than 0.3‰.

TABLE IV
PERFORMANCES OF DIFFERENT PDSs

Main Designer Of the PDS	System Setting			Kernel Classifier Performance		Computer Setting		System Performance		
	Vehicular or Not	Number of Camera	Optical or Infrared Camera	Positive rate with positive samples (%)	False positive rate with negative samples (%)	CPU	RAM	Detection rate (%)	False positive rate (%)	Detection speed (fps)
Cao XB	Y	One	Optical	99.9	0.03	2.8G	512M	86.8	0.013	12.2
Paul Viola [19]	N	One	Optical	—	—	2.8G	—	80	0.00025	4
DM Gavrila [18]	Y	One	Optical	90	15	450M	—	—	—	—
Fengliang Xu [1]	Y	One	Infrared	42 - 90	10	500M	256M	26– 94*	2.6*	0.025*
M.Bertozzi [2]	Y	Two	Infrared	—	—	2.8G	1G	83**	0.1– 0.5**	12**

* – Data is calculated from the original data in the paper.
** – Data is estimated from the figures presented in the paper.

the shelter ratio has more influence on the correct identification rate than the distance. Nevertheless, the proposed direction-identification method was shown to be efficient in normal situations.

To improve the performances of these two accessory functions, we plan to use a recognition-based tracking technique in the future.

E. Summarization and Comparison With Other Typical PDSs

To assess the contribution of this paper, we compared our system with four typical PDSs with similar/different settings.

Table IV lists the performances of typical PDSs. It indicates that our system has an acceptable performance compared with other PDSs; at the same time, our system is low cost.

V. CONCLUSION

This paper has proposed a vehicular PDS using one optical camera. The system adopts cascaded classification using both the statistical learning and the decomposed SVM classification. In contrast to most other PDSs with a single optical camera, this system uses both the appearance and motion features for detection. Furthermore, by using a new algorithm (proposed in our previous work [29]), the system can quickly and precisely identify the direction of a pedestrian and estimate his/her distance from the vehicle.

In general, compared with the infrared-camera-based PDS, the single-optical-camera-based PDS is low cost, has longer detection distance, and is not influenced by temperature. In addition, the technology developed for the single-optical-camera-based PDS can also be used for other integrated systems.

The experiments show that the single-optical-camera-based PDS algorithm in this paper has a very good performance in both the pedestrian detection and the direction identification.

Future work includes the following:

1) hardware and algorithm design of a PDS, which is suitable for high-speed vehicle assistance. To achieve this, we envisage needing a dynamic shift in gathering motion information;
2) hardware and algorithm design of PDS to further increase the detection performance;
3) hardware and algorithm design of PDS to handle special situations, such as pedestrians that are suddenly rushing out.

REFERENCES

[1] F. Xu, X. Liu, and K. Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 63–71, Mar. 2005.

[2] M. Bertozzi, A. Broggi, A. Fascioli, T. Graf, and M.-M. Meinecke, "Pedestrian detection for driver assistance using multiresolution infrared vision," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1666–1678, Nov. 2004.

[3] M. Bertozzi, A. Broggi, A. Lasagni, and M. Del Rose, "Infrared stereo vision-based pedestrian detection," in *Proc. IEEE Int. Conf. Intell. Veh. Symp.*, Jun. 2005, pp. 24–29.

[4] Y. Fang, K. Yamada, Y. Ninomiya, B. K. P. Horn, and I. Masaki, "A shape-independent method for pedestrian detection with far-infrared images," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1679–1697, Nov. 2004.

[5] T. Tsuji, H. Hattori, M. Watanabe, and N. Nagaoka, "Development of night-vision system," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 203–209, Sep. 2002.

[6] X. Liu and K. Fujimura, "Pedestrian detection using stereo night vision," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1657–1665, Nov. 2004.

[7] D. M. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: The PROTECTOR system," in *Proc. IEEE Int. Conf. Intell. Veh. Symp.*, Jun. 2004, pp. 13–18.

[8] D. M. Gavrila and J. Giebel, "Shape-based pedestrian detection and tracking," in *Proc. IEEE Int. Conf. Intell. Veh. Symp.*, Jun. 2002, vol. 1, pp. 8–14.

[9] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi, "Shape-based pedestrian detection and localisation," in *Proc. IEEE Intell. Transp. Syst.*, Oct. 2003, vol. 1&2, pp. 328–333.

[10] B. Heisele and C. Woehler, "Motion-based recognition of pedestrians," in *Proc. IEEE 14th Int. Conf. Pattern Recog.*, Aug. 1998, vol. 2, pp. 1325–1330.

[11] Y. Ricquebourg and P. Bouthemy, "Real-time tracking of moving persons by exploiting spatio-temporal image slices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 797–808, Aug. 2000.

[12] C. Wohler, U. Kressel, and J. K. Anlaur, "Pedestrian recognition by classification of image sequences global approaches vs. local spatio-temporal processing," in *Proc. IEEE 15th Int. Conf. Pattern Recog.*, Sep. 2000, vol. 2, pp. 540–544.

[13] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen, "Walking pedestrian recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 3, pp. 155–163, Sep. 2000.

[14] U. Franke and S. Heinrich, "Fast obstacle detection for urban traffic situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 173–181, Sep. 2002.

[15] H. Elzein, S. Lakshmanan, and P. Watta, "A motion and shape-based pedestrian detection algorithm," in *Proc. IEEE Int. Conf. Intell. Veh. Symp.*, Jun. 2003, pp. 500–504.

[16] Y. W. Xu, X. B. Cao, and H. Qiao, "A low-cost pedestrian detection system with a single optical camera," in *Proc. 6th World Congr. Control Autom.*, 2006, vol. 10, pp. 8759–8763.

[17] Y. W. Xu, X. B. Cao, H. Qiao, and F. Y. Wang, "A cascaded classifier for pedestrian detection," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 13–15, 2006, pp. 336–343.

[18] D. M. Gavrila, "Pedestrian detection from a moving vehicle," in *Proc. Eur. Conf. Comput. Vis.*, 2000, pp. 37–49.

[19] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, Jul. 2005.

[20] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems—Single-frame classification and system level performance," in *Proc. IEEE Int. Conf. Intell. Veh. Symp.*, Jun. 2004, pp. 1–6.

[21] L. Havasi, Z. Szlávik, and T. Sziranyi, "Higher order symmetry for nonlinear classification of human walk detection," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 822–829, May 2006.

[22] L. Andrade, S. Blunsden, and B. Fisher, "Characterization of optical flow anomalies in pedestrian traffic," in *Proc. IEE Int. Symp. Imag. Crime Detection Prevention*, 2005, pp. 73–78.

[23] H. Qiao, F. Y. Wang, and X. B. Cao, "Application of a decomposed support vector machine algorithm in pedestrian detection from a moving vehicle," in *Proc. IEEE Int. Conf. Intell. Security Informatics*, 2005, pp. 662–663.

[24] C. J. C. Bugres, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[25] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1998.

[26] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[27] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," in *Proc. EuroCOLT*, 1995, pp. 23–37.

[28] M. H. Alsuwaiyel and D. M. Gavrila, "On the distance transform of binary images," in *Proc. IEEE Int. Conf. Imag. Sci., Syst., Technol.*, 2000, vol. I/II, pp. 83–86.

[29] X. B. Cao, H. Qiao, F. Y. Wang, and X. Z. Zhang, "Application of cooperative co-evolution in pedestrian detection systems," in *Proc. IEEE Int. Conf. Intell. Security Informatics*, 2005, pp. 664–665.

[30] D. M. Gavrila, "The visual analysis of human movement: A survey," *Comput. Vis. Image Underst.*, vol. 73, no. 1, pp. 82–98, Jan. 1999.

Xian-Bin Cao received the B.S. degree in computer science and the M.S. degree in information and system from Anhui University, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree in intelligent information processing from the University of Science and Technology of China (USTC), Hefei, in 1996.

He has been with the USTC since 1996 and became an Associate Professor with the Department of Computer Science and Technology from 1999 to 2005, where he is currently a Professor. He is also the Vice-Director of the department and of the Artificial Intelligence Research Center. Since 2005, he has been the Administrative Director of the Anhui Province Key Laboratory of Software in Computing and Communication, Hefei. His current research interests include natural computation, intelligent transportation systems, and information security. He has been publishing more than 90 books, book chapters, and papers in these areas since 1993.

Hong Qiao (SM'06) received the B.Eng. degree in hydraulics and control and the M.Eng. degree in robotics from Xi'an Jiaotong University, Xi'an, China, the M.Phil. degree in robotics control from the Industrial Control Center, University of Strathclyde, Strathclyde, U.K., and the Ph.D. degree in robotics and artificial intelligence from De Montfort University, Leicester, U.K., in 1995.

She was a University Research Fellow with De Montfort University from 1995 to 1997. She was a Research Assistant Professor from 1997 to 2000 and an Assistant Professor from 2000 to 2002 with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China. Since January 2002, she has been a Lecturer with the School of Informatics, University of Manchester, Manchester, U.K. Currently, she is also a Professor with the Laboratory of Complex Systems and Intelligent Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China (on leave from the University of Manchester). She first proposed the concept of "the attractive region in strategy investigation," which has successfully been applied by herself in robot assembly, robot grasping, and part recognition. The work has been reported in *Advanced Manufacturing Alert* (Wiley, 1999). Her current research interests include information-based strategy investigation, robotics and intelligent agents, animation, machine learning (neural networks and support vector machine), and pattern recognition.

Dr. Qiao is a member of the Program Committee of the IEEE International Conference on Robotics and Automation from 2001 to 2004. She is currently an Associate Editor of the IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS, PART B.

John Keane received the B.Sc. degree in computation and computer science from the University of Manchester Institute of Science and Technology, Manchester, U.K., in 1985 and the M.Sc. degree in computation and computer science from the University of Manchester in 1989.

He is the MG Singh Chair of Computing Science and the Deputy Director of the U.K. National Centre for Text Mining, University of Manchester. He has had industrial experience in the U.K. with Trustees Savings Bank, International Computer Ltd., and Philips Data Systems. His research interests are in the area of data engineering, particularly data integration, data-intensive systems, data mining, and decision support.

Prof. Keane is currently an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C and an Area Editor of *Simulation Modeling*.