

Short Paper

Shape-based Pedestrian Detection in Infrared Images

SHU LIU, YUPIN LUO AND SHIYUAN YANG

*Department of Automation
Tsinghua University
Beijing, P.R. China*

With the increase of requirement for improving the safety of night driving, automatic pedestrian detection has received more and more attraction. This paper mainly introduces a pedestrian detection method in infrared images. Base on the properties of infrared images, we present a two-step pedestrian detection method including pedestrian candidate selection and validation. The first step is localization of pedestrian candidate, which is to detect warm objects with specific size and aspect ratio. Then, the validation process is based on template matching, which uses multiscale representation and Dynamic Programming for matching deformed and possible occluded contour. The superiority of the proposed shape matching algorithm to the conventional methods is due to the use of hierarchy of segmented representation. It can adjust automatically while the quantity of noise and deformation changes, which improves the accuracy of pedestrian detection. Experimental results have confirmed the effectiveness of the proposed method.

Keywords: infrared image, pedestrian detection, curvature scale space, dynamic programming, shape matching

1. INTRODUCTION

Safety of night driving has gradually attracted more and more attentions, especially for older drivers or drivers with visual limitations. Based on the technical report from Nebraska Highway Safety Program and the Lincoln-Lancaster County Health Department (July 2001), there are three to four times more deaths occur while driving at night than driving during the day, and a 50-year-old driver needs twice as much light to see as a 30-year-old [1].

The use of video sensors and image processing methods dedicated to detect and classify pedestrians provides a promising approach for the development of driver assistance systems [2]. A significant amount of progress has been made in the area of pedestrian detection. The performance of pedestrian detection depends heavily on the choices of feature extraction. The most popular methods use 2D templates for pedestrians at different poses [3, 4]. One or a series of typical templates are defined to capture pedestrians at unexpected poses. Pedestrian locations are determined by searching for regions with minimum shape distance measures to template patches. For example, a hierarchy for pe-

Received November 8, 2004; revised March 16 & June 7, 2005; accepted August 18, 2005.
Communicated by Kuo-Chin Fan.

pedestrian shapes is proposed in [5]. And a template probabilistic model is defined to describe the possibility of different poses [6, 7]. The detection of human part, such as skin hue, face, head location, *etc.*, provides more flexibility when dealing with pedestrians at different poses [8]. However it is possible that components, such as faces, head, *etc.*, are not visible in some situation. In several applications, pedestrians can be identified by special human motion model [9], such as leg motion [10] or other recurrent motion. The combination of a pedestrian model and walking rhythm of pedestrians [11] is also used to detect and track a pedestrian. Normally this feature is only useful in limited scenario, such as video surveillance monitoring situations, and is not suitable for outdoor driving situations. Symmetry property [3, 12] is often used to locate potential pedestrian, which is invariant under nodding movements and under changes of object size. However, it might not work very well for objects with arbitrary poses. A two-layer custom-designed neural network is designed to classify objects in [13]. Support Vector Machine (SVM) is also used to classify pedestrian candidates in [14] and [15]. For a camera installed on a fast moving vehicle, new difficulties arise: the relative movement between pedestrian and background is insignificant and pedestrian vary enormously in scales. So, an active background subtraction in sequential images taking from a moving camera is used to get the edge of moving object, and a method of contour matching is used in detection and location process [16].

Recently, pedestrian detection systems based on infrared images attract more attention [3, 8, 17] because of its advantages especially at night. The brightness intensities of infrared images are representative of temperature of object surface points. In general, infrared images have the following properties that differ from visible images [1].

- Pedestrians usually emit more heat than static background objects, such as trees, road, *etc.* Therefore image regions containing pedestrians or other “hot” objects will be brighter than background.
- Because the temperature of all pedestrians’ bodies is similar, the brightness of different pedestrians in different infrared images should be similar in spite of different color and texture of their clothing.
- The intensities of whole pedestrian regions should be approximately uniform since the temperature of different body parts is similar.
- The infrared cameras are independent of light conditions: they can be used in day-time or night-time with no or little difference, extending vision beyond the usual limitations of day-light cameras. Moreover, the absence of colors or textures eases the processing towards interpretation.

The above properties are the advantages of infrared images over visible images for pedestrian detection. Thus, infrared images can facilitate the pedestrian recognition especially at night.

Our night vision system deploys an infrared camera mounted on a moving vehicle to detect heat-emitting objects in the dark within the view of headlamps, and equips a separated screen to show infrared images. The system analyzes the infrared images and determine which of them contain pedestrians and where they are located in the image. When a pedestrian enters the driver’s visual scope, the system will give out warning information automatically. This information will aid the driver to avoid the collision be-

tween the vehicle and a pedestrian.

The following section introduces process of pedestrian candidate selection. Section 3 presents the pedestrian candidate validation method, and describes a shape matching algorithm in multi-scale space. Section 4 is the experimental results. The last section is discussions of the experimental results and concludes the paper with some final consideration.

2. THE PEDESTRIAN DETECTION METHOD

This pedestrian detection system includes two phases: pedestrian candidate detection and validation. In pedestrian candidate detection phase, regions of interest (ROIs) are segmented based on feature criteria (details in section 2.1). Then validation phase decides whether ROIs contain pedestrians or not through pedestrian shape feature matching. The following figure (Fig. 1) shows an infrared image and detected pedestrian bounded by a red box.



Fig. 1. An infrared image and detected pedestrian bounded by a red box.

2.1 ROIs Selection

Night-scene infrared image appears to have constant contrast, and a human body always appears as a hotspot due to its higher temperature compared with that of the environment. Our system begins with the detection of hotspots with a dynamic threshold.

$$Threshold = \alpha \cdot mean + (1 - \alpha) \cdot white \quad (1)$$

Where *mean* is the mean intensity of image and *white* is the highest intensity. In this paper, the parameter α is 0.8. We focus on those hotspots with a specific size and aspect ratio, because the size of a pedestrian often varies in a special range. The size of pedestrian is fixed to the following values: height: 180 cm \pm 15% and width: 60 cm \pm 10%.

The presence of a ROI is checked in bounding boxes with different size placed in different positions in the image. But not all possible bounding boxes need to be checked, mainly due to computational time and detail content. In fact, very small bounding boxes

enclosing faraway pedestrians show very low information. In such situations, it is easy to obtain false positives since many road infrastructures and other objects (for example, traffic signs, barriers, trees, buildings, and road markings) may present morphological characteristics similar to a human shape. And a great distance makes it not urgent to determine whether it is a pedestrian or not and warn the driver. So, the size of bounding boxes is limited from 12×30 pixels to 42×90 pixels.

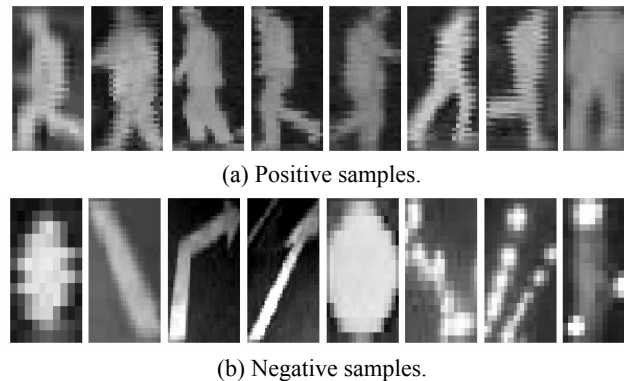


Fig. 2. The candidates clipped from infrared images. (a) is candidates containing pedestrian and (b) is candidates containing other objects which is similar to pedestrian.

Some of pedestrian candidates clipped from the infrared images are shown in Fig. 2. From the figure, it can be seen that the pedestrian candidates include not only pedestrian but also other objects. In fact, objects which have passive heat radiation behavior (shown in Fig. 1) may be strongly heated by the light, making the scene more complicated or even causing heat radiations or reflections. In addition, in the case of strong external heat radiation, clothes that people wear can have different thermal behavior depending on their type and color, thus adding texture to the image. It is therefore imperative to validate the pedestrian candidate.

2.2 Contours Extraction

There are many algorithms to extract contour from gray image. Here we detect the edge of ROIs got from previous section by using gradient based edge detection algorithm. Since the detected edges may have a multi-pixel width, a thinning algorithm is applied to remove extra pixel without losing connectivity. Combining with the information of binary copy, edges are linked along the binary shape. And some outer edges caused by noise are discarded. Line or curve along binary shape is used to connect broken edges. Finally, the contours are expressed by a simple closed curve which is sampled in a counter clockwise direction. The starting points of contours are selected randomly because it would not affect the matching result. These contours will be used as input of candidate validation process.

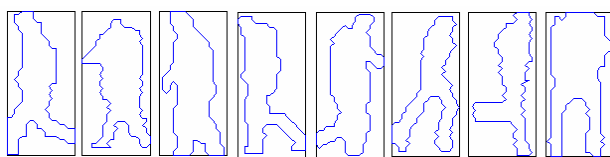


Fig. 3. Contours extracted from ROIs.

Some of pedestrian contours are shown in Fig. 3. It can be seen that the contours are deformed heavily due to low resolution and body pose, which strongly affect the accuracy of recognition. In the next section, we will introduce a stable matching algorithm based on multiscale representation and Dynamic Programming to classify the candidates. The proposed algorithm treats noise and shape distortions by consideration each segment of contour in different scale level. And the algorithm is invariant to translation, scale, and orientation.

3. PEDESTRIAN VALIDATION

The pedestrian validation is based on template matching. We compare the candidates with the pedestrian contours in template set. The minimum dissimilarity between candidates and templates is used to determine whether it is a pedestrian. The candidate with minimum dissimilarity lower than a threshold is pedestrian.

Considering the heavy deformation and noise of extracted contours, representation using local feature is more moderate than that using global feature. The major problem with conventional segmented representation is that small perturbations to the shape can yield large changes in the segmentation. And in many cases, it is hardly to find a suitable scale to segment all kinds of shapes with noise and deformation. Multiscale methods are considered as the most promising ones for shape matching because they are resistant to moderate amounts of deformation and noise [18-20]. The proposed method gets the hierarchy of segmented representation by tracking the Curvature Scale Space (CSS) representation [21], and sequences of curve segmentation are compared from coarser to finer scales. The process of contour segmentation can be adjusted automatically while the amounts of noise and deformation change. The more serious the noise and deformation are, the higher scale the final matched curve segmentation will be in. The corresponding points used to segment contours is often important points (for example, head or feet in pedestrian contour), so the correspondence of similar parts of shapes helps to analyze the structure of pedestrian contour. In this paper, the curvature extremes are used to avoid major structural changes in the contours of CSS image caused by small shape changes, which may lead to matching errors.

3.1 Template Construction

To improve the efficiency and accuracy of pedestrian recognition, a series of templates are used to express a variety of pedestrian contours. The pedestrian template set is generated with k-means algorithm because of its simplicity and efficiency.

Assume that there is a sequence of N rectangle images which are known to contain

pedestrians in different poses and orientation. After contour extraction process is performed, we can obtain a set of pedestrian contours. The pedestrian template set is constructed as follows:

Algorithm

Input: N pedestrian contours.

Output: template set containing k pedestrian contours.

Step 1: Select K pedestrian contours from the given samples at random, as initial cluster centers $(M_1^{(0)}, M_2^{(0)}, \dots, M_k^{(0)})$.

Step 2: Assign each sample to one of the clusters according to the minimum distance criterion (namely, the sample is classified to the nearest cluster). The cluster centers are $M_1^{(l)}, M_2^{(l)}, \dots, M_k^{(l)}$ at the l th iteration.

Step 3: Compute the mean within each cluster and let the samples closest to mean be the new cluster centers $(M_1^{(l+1)}, M_2^{(l+1)}, \dots, M_k^{(l+1)})$.

Step 4: Repeat from step 2, until the new cluster centers don't change any more.

Step 5: Choose the samples closest to the cluster centers to compose template set.

Template set is composed of cluster centers. Smaller template set will reduce the calculation time, while the accuracy is also reduced. When K (the number of clusters.) ranges between 50 and 70, the system has a better performance, which will be discussed in section 5. In this algorithm, the distance of two contours is computed with the matching algorithm described in the following subsection.

3.2 Multiscale Representation

Curvature extremes of contours with different level of noise and deformation may cause fake landmarks. As shown in Fig. 4, the front of the pedestrian contour is segmented by a few fake landmarks. It will lead curve 'a' to match with part of curve 'b' in a fixed scale, while this part of contour may be matched correctly in different scale.

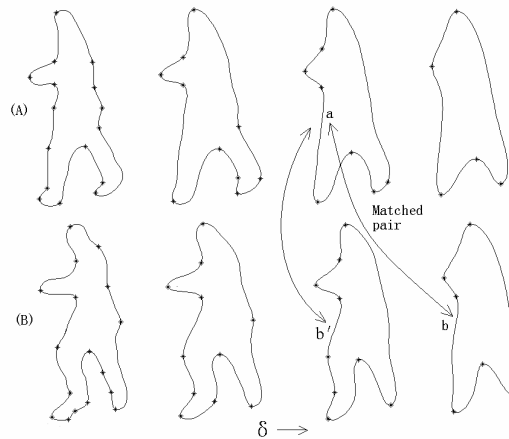
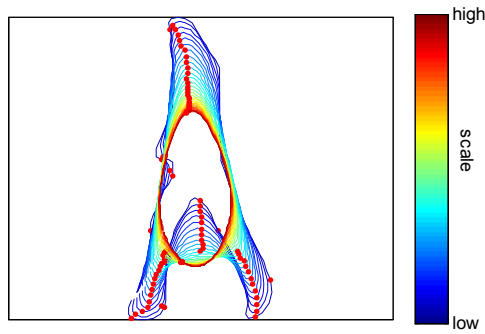


Fig. 4. Partial matching result with (a-b) and without (a-b') multiscale representation.

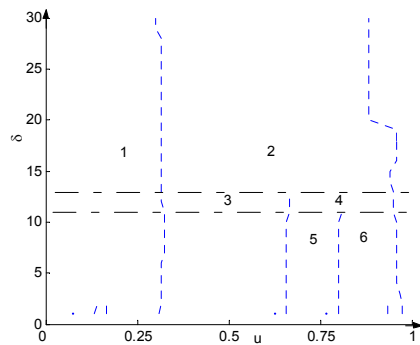
In this section, a series of smoothed shape contours and their curvature extremes are obtained by using the curvature scale space filtering technique, as shown in Fig. 5 (a). The dots along the contour represent the curvature extreme points, which are used as landmarks to segment contour.

All points on a shape contour can be expressed in terms of two periodic functions, $\Gamma(t) = (x(t), y(t))$, where variable t is measured along the contour curve from a specified starting point. $X(t, \delta)$ is defined as the convolution of $x(t)$ and a 1-D Gaussian kernel $g(t, \delta)$ with a scale factor δ . $Y(t, \delta)$ can be computed in a similar manner.

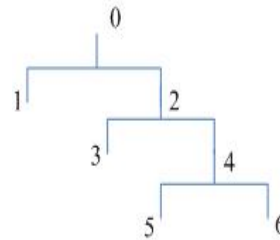
$$g(t, \delta) = \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{t^2}{2\delta^2}\right) \quad (2)$$



(a) The origin contour and their smoothed contour, the dot on the contour is curvature extreme points.



(b) The corresponding CSS images.



(c) The segmentation in multiscale space.

Fig. 5. The contours in different scale, their corresponding CSS image and segmentation.

$$X(t, \delta) = x(t) \oplus g(t, \delta)$$

$$Y(t, \delta) = y(t) \oplus g(t, \delta)$$

(3)

Different levels of detailed curvature of the smoothed curve are expressed as

$$K(t, \delta) = \frac{d\theta}{ds} = \frac{XY'' - X''Y'}{(X'^2 + Y'^2)^{3/2}}. \quad (4)$$

Where $\theta = \arctg(Y'(t, \delta)/X'(t, \delta))$, and s represents curve length. X' , X'' , Y' and Y'' stand for $\partial X/\partial \alpha$, $\partial^2 X/\partial \alpha^2$, $\partial Y/\partial \alpha$ and $\partial^2 Y/\partial \alpha^2$ separately. To simplify calculation, the derivatives (X' , X'' , Y' , Y'') can be computed using finite differences to approximate. As δ increases, the contour becomes smoother and the number of curvature extremes on it decreases. When δ becomes sufficiently high, the contour will be a convex curve with no more than two curvature extremes. The curvature extremes in different scale can be mapped to the (u, δ) plane as showed in Fig. 5 (b). The horizontal axis in this image represents the normalized arc length u , and the vertical axis represents δ , the width of the Gaussian kernel. The intersection of every horizontal line with the curves in this image indicates the location of curvature extremes on the corresponding evolved curve $\Gamma(u, \delta)$. The trajectories of curvature extremes reveal their relative importance on the shape. The important curvature extremes will exist in higher scale, while those caused by noise and local deformation will vanish or merge with other extremes as δ increases. We can obtain segmented structures at different scales from (u, δ) image, as shown in Fig. 5 (c).

The scale space image can be obtained by using continuous scales. Continuous scale space filtering needs excessive processing time and memory allocation. A simple method to avoid this problem is to use discrete scales. This induces the creation of incomplete scale space image curves, which make it difficult to perform tracking when inflection points are used. In this paper using the trajectory of extreme points makes it easier to track because the position of curvature extremes changes slightly when δ increasing, which can be seen in Fig. 5 (b).

3.3 Matching Algorithm

Assume that T and S are the two shapes to be matched. Points sequence used to segment the two contours are denoted by $(p_1^{(h)}, p_2^{(h)}, \dots, p_N^{(h)})$ and $(q_1^{(k)}, q_2^{(k)}, \dots, q_M^{(k)})$. $T^{(h)}$ and $S^{(k)}$ denote the two segment sequences at scales $\delta^{(h)}$ and $\delta^{(k)}$ corresponding to shapes T and S , that is

$$\begin{aligned} T^{(h)} &= t_1^{(h)}, t_2^{(h)}, \dots, t_N^{(h)}, h = 0, 1, \dots, H \\ S^{(k)} &= s_1^{(k)}, s_2^{(k)}, \dots, s_M^{(k)}, k = 0, 1, \dots, K. \end{aligned} \quad (5)$$

$N^{(h)}$ and $M^{(k)}$ are the number of segments, and $t_i^{(h)}$ stands for the segment between the two consecutive extreme points $p_i^{(h)}$ and $p_{i+1}^{(h)}$. $S_j^{(k)}$ is similar. $h = k = 0$ corresponds to the finest scale, and $h = H$, $k = K$ correspond to the coarsest scales. Usually, H equals K . Considering that the contour is closed, for the segment sequences at any scale, we have $t_{i+N^{(h)}}^{(h)} = t_i^{(h)}$, $s_{j+M^{(k)}}^{(k)} = s_j^{(k)}$.

The goal of the segment matching algorithm is to find the best association of segments in shape A to segments in shape B, and the segmentation resulting in minimum value of total dissimilarity cost. The proposed segment matching algorithm searches for the corresponding segment pairs from coarser scale to finer scale to find the most similar

parts of the two shapes. For a sequence of scale space representation, firstly, we choose curvature extremes in a moderate scale to segment the contour and calculate the dissimilarity of the two shapes. Then, add the curvature extreme with a higher resolution to the segment points sequence $p_i^{(h)}$ (or $q_j^{(k)}$), and corresponding segment split into two (or a group of) segments. If the total dissimilarity cost decreases with the new segment sequence, the procedure is repeated with the two shapes alternately until the total dissimilarity does not decrease anymore.

3.4 Dynamic Programming

The problem of finding the best correspondences between the two shapes is formulated as a minimization problem which is solved efficiently by dynamic programming. A table of partial costs is built and the optimal matching is searched in the form of a path in the DP table that minimizes the total dissimilarity cost.

The rows and the columns of the DP table correspond to segments of shape A and shape B respectively. Each cell $cell(i, j)$ in the DP table denotes the dissimilarity between the two segments. A path is a linked sequence of cells in the DP table along diagonal, which starts from the first row, and ends at the last row. The total dissimilarity cost $D(T, S)$ of shape T and S is defined as

$$D(T, S) = \min_p \{g(i_p, j_p)\} \quad (6)$$

where $g(i_p, j_p)$ is the total cost along a possible path. In turn, $g(i_p, j_p)$ is defined as follows:

$$g(i_p, j_p) = \sum_p \psi(t_i^{(h)}, s_j^{(k)}). \quad (7)$$

Function $\psi(t_i^{(h)}, s_j^{(k)})$ represents the dissimilarity cost of two segments, which consists of three components:

$$\begin{aligned} \psi(t_i^{(h)}, s_j^{(k)}) = & \text{DissimilarityCost}(t_i^{(h)}, s_j^{(k)}) + \lambda \text{SplitCost}(t_i^{(h)} | t_i^{(h-1)}) \\ & + \lambda \text{SplitCost}(s_j^{(k)} | s_j^{(k-1)}). \end{aligned} \quad (8)$$

The first term in Eq. (8) is the distance of feature vector of two segments. The following two terms represent the cost when the segment split into a group of segments. The greater the number of segments splitting from their parent segment, the higher the cost should be. Constant λ represents the relative importance of the splitting and dissimilarity costs. Low values of λ encourage splitting and, conversely, high values of λ inhibit splitting. SplitCost is to measure the importance of the splitted segment relative to the whole matched contour. It is defined as:

$$\text{SplitCost}(t_i^{(h)} | t_i^{(h-1)}) = \text{Length}(t_i^{(h)}) / \sum_{i=1}^{N(h)} \text{Length}(t_i^{(h)}) \quad (9)$$

$SplitCost(s_j^{(k)} | s_j^{(k-1)})$ is calculated similarly.

The feature vector of segment required in $DissimilarityCost(t_i^{(h)}, s_j^{(k)})$ is composed of one or several features (*i.e.*, curve length, area, or Fourier descriptors), which should be invariant to scale, translation, orientation. Here we use the Euclidean distance between their Fourier Descriptors vectors as the dissimilarity cost of two segments.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We tested the pedestrian classification algorithm on 400 positive samples and 200 negative samples randomly chosen from the infrared image database. Two examples of contour retrieval result using the proposed shape matching algorithm in section 3 are shown in Fig. 6. The left contour is inquired shape and followed by four closest shapes. Contours shown in the figure are smoothed contours which have less noise than original ones do. Experimental results have shown that the algorithm is robust to noise and shape deformation. Its superiority to the conventional methods lies in the use of hierarchy of segmented representation.



Fig. 6. Similar shapes using the proposed algorithm.

We also tested the proposed pedestrian detection system. The pedestrian contour database got from infrared images is divided into training set and test set. The training set containing 180 positive samples with different poses and orientations is used to generate template set. A MATLAB implementation of pedestrian detection using the proposed algorithm takes between 10 and 40 seconds on a PC (Pentium IV 1.8 GHz with 512M of RAM), depending on the size of template set. Table 1 displays the overall detection result of pedestrians. “Detected” in Table 1 means the ratio of correct classified samples

Table 1. Detection ratio.

Template size	Detected (%)	Missed (%)	False alarm (%)
50	83.97	3.14	12.89
70	86.26	1.57	12.17

over total test samples; “missed” means pedestrian which does not detected; “false alarm” means wrongly classified non-pedestrian. It is shown that with more templates the synthetic performance of pedestrian detection system will be better. On the other hand, the detected ratio and false alarm are also related to the training samples used to generate template set. With the size of training sample increasing, the template set will be more appropriate. And the size of template set will increase until a stable value. Moderate templates will increase the detected ratio with a decrease in missed ratio and false alarm ratio.

In this example, although the missed ratio is not very low, the detected frames are evenly distributed in time, which means that the pedestrian could always be detected in a short period of time. For the issue of false alarms, most of false candidates can be successfully discarded by using multiple classifier and more features.

5. CONCLUSION

This paper presents a method for pedestrian detection in night view system, and describes a new algorithm for shape matching, which is robust to noise and shape deformation. The pedestrian detection method can be readily extended to other traffic analysis, and the proposed multiscale matching algorithm can be used as a simple method for shape matching and similarity retrievals in image databases. As a result of the algorithm, the correspondence of similar parts of shapes also helps to analyze the object structure and can be used as prior knowledge to learn shape model.

In future work we will try to improve the system performance by optimizing the template set. Considering the pedestrian structure information, which can be got from the process of shape matching, some of templates can be synthesized to reduce the size of template set. Smaller size of template set and storage of CSS information will reduce the time complexity. And the detected ratio and false alarm ratio will be improved by combining other recognition algorithm and considering the sequence of images, which will also solve some of shape occlusion.

REFERENCES

1. Y. Fang, K. Yamada, Y. Ninomiya, B. Horn, and I. Masaki, “Comparison between infrared-image-based and visible-image-based approaches for pedestrian detection,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2003, pp. 505-510.
2. L. Fletcher, N. Apostoloff, L. Petersson, and A. Zelinsky, “Vision in and out of vehicles,” *Intelligent Systems, IEEE*, 2003, pp. 12-17.
3. M. Bertozzi, A. Broggi, P. Grisleri, T. Graf, and M. Meinecke, “Pedestrian detection

- in infrared images,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2003, pp. 662-667.
4. Z. Qui, D. Yao, Y. Zhang, D. Ma, and X. Liu, “The study of the detection of pedestrian and bicycle using image processing,” *IEEE Proceedings of the Intelligent Transportation Systems*, Vol. 1, 2003, pp. 340-345.
 5. D. M. Gavrilu, “Pedestrian detection from a moving vehicle,” in *Proceedings of the European Conference on Computer Vision*, 2000, pp. 37-49.
 6. H. Nanda and L. Davis, “Probabilistic template based pedestrian detection in infrared videos,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2002, pp. 15-20.
 7. C. Stauffer and E. Grimson, “Similarity templates for detection and recognition,” in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, Vol. 1, 2001, pp. 1221-1227.
 8. T. Tsuji, H. Hattori, M. Watanabe, and N. Nagaoka, “Development of night-vision system,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, 2002, pp. 203-209.
 9. P. Viola, M. J. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 734-741.
 10. L. Lee, G. Dalley, and K. Tieu, “Learning pedestrian models for silhouette refinement,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 663-670.
 11. C. J. Pai, H. R. Tyan, Y. M. Liang, H. Y. M. Liao, and S. W. Chen, “Pedestrian detection and tracking at crossroads,” *Pattern Recognition*, Vol. 37, 2004, pp. 1025-1034.
 12. M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi, “Shape-based pedestrian detection and localization,” in *Proceedings of the IEEE Intelligent Transportation Systems*, Vol. 1, 2003, pp. 328-333.
 13. H. Nanda, C. Benabdelkedar, and L. Davis, “Modelling pedestrian shapes for outlier detection: a neural net based approach,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2003, pp. 428-433.
 14. F. Xu and K. Fujimura, “Pedestrian detection and tracking with night vision,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2002, pp. 21-30.
 15. X. Yuan, Z. Sun, Y. Varol, and G. Bebis, “A distributed visual surveillance system,” in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, 2003, pp. 199-204.
 16. T. Hashiyama, D. Mochizuki, Y. Yano, and S. Okurna, “Active frame subtraction for pedestrian detection from images of moving camera,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, 2003, pp. 480-485.
 17. F. Xu and K. Fujimura, “Human detection using depth and gray images,” in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2003, pp. 115-121.
 18. C. H. Wang, H. C. Lin, C. C. Shih, H. R. Tyan, C. F. Lin, and H. Y. M. Liao, “Querying image database by video content,” *Journal of Information Science and Engineering*, Vol. 19, 2003, pp. 967-987.
 19. F. Mokhtarian, S. Abbasi, and J. Kittler, “Efficient and robust retrieval by shape con-

- tent through curvature scale space,” in *Proceedings of the International Workshop on Image DataBases and MultiMedia Search*, 1996, pp. 35-42.
20. E. G. M. Petrakis, A. Diplaros, and E. Milios, “Matching and retrieval of distorted and occluded shapes using dynamic programming,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, 2002, pp. 1501-1516.
 21. S. Liu, “Shape matching using dynamic programming in scale-space,” *IASTED International Conference on Visualization, Imaging and Image Processing*, 2004, pp. 787-791.

Shu Liu (刘曙) is a Ph.D. Candidate in the Department of Automation at Tsinghua University. She received the B.S. degree from Wuhan University of Science and Technology, in 1991, and the M.S. degree from The University of Science and Technology Beijing, in 1995. Her research interests include image processing and computer vision.

Yupin Luo (罗予频) is a Professor in the Department of Automation at Tsinghua University. He received B.S. degree from Hunan University in 1982, E.M. and Ph.D. from Nagoya Institute of Technology, Japan in 1987 and 1990 respectively. His current research interest is in the area of computational geometry, graph theory and artificial intelligence.

Shiyuan Yang (杨士元) was born in Shanghai, China. He received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 1970 and 1981, respectively. Since 1993, he is a Professor in Automation Department at Tsinghua University, Beijing, China. He is an Associate Director of the FTC committee, China. His main research interests are home automation network, test technology, electronic technology application, system fault diagnosing.