

Research Article

An Object Detection Method Using Wavelet Optical Flow and Hybrid Linear-Nonlinear Classifier

Pengcheng Han, Junping Du, Jun Zhou, and Suguo Zhu

Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Junping Du; junpingdu@126.com

Received 5 June 2013; Accepted 29 July 2013

Academic Editor: Daniel Zaldivar

Copyright © 2013 Pengcheng Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a new computational intelligence method using wavelet optical flow and hybrid linear-nonlinear classifier for object detection. With the existing optical flow methods, it is difficult to accurately estimate moving objects with diverse speeds. We propose a wavelet-based optical flow method, which uses wavelet decomposition in optical flow motion estimation. The algorithm can accurately detect moving objects with variable speeds in a scene. In addition, we use the hybrid linear-nonlinear classifier (HLNLC) to classify moving objects and static background. HLNLC transforms a nonoptimal scalar variable into its likelihood ratio and uses a scalar quantity as the decision variable. This approach is appropriate for the classification of optical flow feature vectors with unequal variance matrices. The experimental results confirm that our proposed object detection method has an improved accuracy and computation efficiency over other state-of-the-art methods.

1. Introduction

In modern engineering, the requests for research and design are increasingly achieved with the help of intelligent models. Computational intelligence (CI) has emerged as powerful tools for information processing, decision making, and knowledge management [1]. CI is a set of nature-inspired computational methodologies and approaches to address complex real-world problems to which traditional approaches [2, 3]. In this paper, we propose a new computational intelligence method using wavelet optical flow and hybrid linear-nonlinear classifier (HLNLC) for object detection. Object detection can be subcategorized as either the detection that has similar characteristics [4, 5] or the detection of a specific object in a video sequence [6, 7]. Our paper is focused on the second category.

One important task in object detection is motion estimation. Optical flow is one commonly used approach to estimate the object motion. Starting with the original algorithms by Lucas and Kanade (LK) [8] as well as Horn and Schunck (HS) [9], gradient-based methods have led to other improved optical flow estimation methods. However, when the image background is cluttered or the detected object is moving

at high speed, the accuracy of gradient-based methods will be significantly decreased [10]. Another important task in object detection is classification. Classifier techniques such as pulse-coupled neural network (PCNN) [11], fuzzy neural network (FNN) [12], Gaussian SVM (GSVM) [13], and linear discriminant analysis (LDA) [14] have been applied to different object detection situations. In PCNN and FNN schemes, detection accuracy may be coupled with small training errors because each image pixel is associated with a unique neuron and vice versa. In SVM classifier approach, the classifier is characterized by excessive complexity when it comes to the binary classification task of identifying moving objects versus static background. The LDA classifier is a more robust linear classifier which has proven itself as the ideal observer for input feature vectors with equal covariance matrices [15], but it is not the optimal choice for data that contain multivariate optical flow vectors with unequal covariance matrices.

Our work will consider both factors. We propose a new object detection method using wavelet-based optical flow and hybrid linear-nonlinear classifier. Some wavelet-based optical flow estimation approaches have been proposed; Wu et al. [16] used wavelets to model and reconstruct flow vectors; at each iteration, the estimation will be repeated, which is to

reduce efficiency. In [17], Bernard assumed that the optical flow was locally constant. In [18], Srinivasan and Chellappa proposed a similar method which modeled optical field using a set of overlapping basis functions. In [17, 18], optical flow computation has been simplified, and the disadvantage is to compromise the accuracy, especially when several objects with different speeds existed in one scene. In this paper, we use wavelet calculus to compute derivatives of the functions in terms of the scaling expansion coefficients. Our proposed method could achieve an accelerated optical flow computation and accurately estimate the motions of different speed moving objects in the same scenes.

In the binary classification about distinguishing moving objects versus static background, a linear classifier is a more robust choice than existing methods [19]. Linear discriminant analysis (LDA) is the most popular linear classifier. LDA produces accurate classifications for two types of input feature matrices: those with normal distributions and those with equal covariance. However, it has trouble with vector data that have an unequal covariance. Thus, we use the novel hybrid linear-nonlinear classifier (HLNLC). HLNLC was proposed by Chen et al. in 2010 [20]. It has been proved to be a more robust linear classifier than LDA and other existing classifiers.

This paper is organized as follows. Section 2 describes wavelet-based optical flow motion estimation method. Section 3 introduces the hybrid linear-nonlinear classifier. Section 4 describes rectangle window scan algorithm. Section 5 describes experimental results. Our conclusions are presented in Section 6.

2. Wavelet Based Optical Flow

2.1. Gradient-Based Optical Flow. Gradient-based optical flow algorithms are based on the assumption of constant brightness [8, 9], where it is assumed that the gradient value of a pixel will not vary due to displacement [21]. It can be described as

$$E = \iint \left[(I_x u_x + I_y v_y + I_t)^2 + a(|\Delta u|^2 + |\Delta v|^2) \right] dx dy. \quad (1)$$

Here, $I(x, y, t)$ represents the brightness value of pixel (x, y) at time t . I_x , I_y , and I_t are the partial derivatives of $I(x, y, t)$ with respect to x , y , and t . The variable (v_x, v_y) is the velocity vector of optical flow estimation about $I(x, y, t)$, and (u, v) is the gradient operator of $I(x, y, t)$. The brightness constancy assumption assumes that the motion vectors are constant within small windows and that the image sequence $I(x, y, t)$ will not change significantly during a short period of time. This assumption can be expressed as

$$\Delta I(x, y, t) = \Delta I(x + u, y + v, t), \quad |\Delta u|^2 + |\Delta v|^2 = 0. \quad (2)$$

Based on (1) and (2), the flow algorithm produces two simultaneous equations for the velocity vector v_x and v_y :

$$\begin{aligned} p_x v_x + p_{xy} v_y + p_{xt} &= 0, \\ p_{xy} v_x + p_y v_y + p_{yt} &= 0, \end{aligned} \quad (3)$$

p_x , p_{xy} , p_y , p_{xt} , and p_{yt} are the product of the partial derivatives I_x, I_y in a time range I_t . The detailed representations may be written as

$$\begin{aligned} p_x &= \sum_t I_x I_x, & p_{xy} &= \sum_t I_x I_y, \\ p_y &= \sum_t I_y I_y, & p_{xt} &= \sum_t I_x I_t, & p_{yt} &= \sum_t I_y I_t. \end{aligned} \quad (4)$$

In gradient-based optical flow algorithms, object displacement between successive frames determines the accuracy of optical flow estimation because the assumption that there will be no major changes between successive frames will break down when the displacement between frames is significant. In order to improve the accuracy of optical flow estimation, displacements between video frames should be projected and recalculated in different frame rates, which means that the algorithm should be able to adaptively adjust the velocity components for different objects moving with different speeds.

2.2. Wavelet Based Optical Flow Estimation. We try to apply the wavelet transform into optical flow estimation. Wavelet transform is an important tool for signal processing, and its magnitude will not oscillate around singularities as the transform magnitude is locally nearly shift invariant [22]. To apply the wavelet decomposition to optical flow estimation, we transform the optical flow equation into the following expression [23]:

$$I = \iint \left[(I_x^2 u^2 + I_y^2 v^2 + I_t^2 + 2I_x I_y uv + 2I_x I_t u + 2I_y I_t v) + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) \right] dx dy. \quad (5)$$

Suppose the image size is $M \times M$ pixels. Then the optical flow vector $[u(x, y), v(x, y)]$ can be expressed as

$$\begin{aligned} u(x, y) &= \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} u \cdot \delta(x - m, y - n), \\ v(x, y) &= \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} v \cdot \delta(x - m, y - n) \\ \delta(X, Y) &= \frac{p(X) = (m, n | Y)}{|D_{m,n}^Y|} \left(\sum_{d \in D_{m,n}^{L(y)}} \ell_d \left(\frac{s_u(y - x)}{s} \right) \right). \end{aligned} \quad (6)$$

The variables $u(x, y)$, $v(x, y)$ are the weighted coefficients of the optical flow. Once $u_{m,n}$ and $v_{m,n}$ can be determined, the optical flow estimation will be accomplished [24]. Thus, we transform the optical flow estimation into a calculation that includes $2M^2$ node variables, where $u_{m,n}$ and $v_{m,n}$ can minimize the object function (5).

I_x^2 , I_y^2 , and $I_x I_y$ denote the products of spatial derivatives:

$$I_x^2(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} a_{m,n} \cdot \delta(x - m, y - n), \quad (7)$$

$$I_y^2(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} b_{m,n} \cdot \delta(x - m, y - n), \quad (8)$$

$$I_x I_y(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} c_{m,n} \cdot \delta(x - m, y - n). \quad (9)$$

I_t^2 , $I_x I_t$, and $I_y I_t$ are products of spatial and temporal partial derivatives, where the time variable t is taken from the variable frame rate:

$$\begin{aligned} I_t^2(x, y) &= \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} d_{m,n} \cdot \delta(x - m, y - n), \\ I_x I_t(x, y) &= \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} e_{m,n} \cdot \delta(x - m, y - n), \\ I_y I_t(x, y) &= \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} f_{m,n} \cdot \delta(x - m, y - n). \end{aligned} \quad (10)$$

The shortest frame interval is σ , and the interval of multiplications is set to $n\sigma$, so that the product of $p_x(t)$ and $p_y(t)$ in an interval $n\sigma$ is given by

$$\begin{aligned} p_x(t) &= \iint \left(\sum_{m,n} I_x u_{m,n} v_{m,n} \cdot \varphi(x, y | t + n\sigma) \right) dx dy, \\ p_y(t) &= \iint \left(\sum_{m,n} I_y u_{m,n} v_{m,n} \cdot \varphi(x, y | t + n\sigma) \right) dx dy, \\ \varphi(x, y | t) &= \sum_L w_L \cdot K(L - L'), \end{aligned} \quad (11)$$

where $w_L = \sum_{t=1}^T \sum_{d \in D_{m,n}^Y} (p(L_t(y)) / T | D_{m,n}^Y |) \ell_d(s_u(y - x) / s)$.

The partial derivatives of $I(x, y, t)$ can be obtained via two difference equations:

$$\begin{aligned} I_x &= I(x + 1, y, t) - I(x, y, t), \\ I_y &= I(x, y + 1, t) - I(x, y, t). \end{aligned} \quad (12)$$

In the time interval $n\sigma$, we compute the optical flow partial derivatives using (7). In our algorithm, $p_x(t)$ and $p_y(t)$ are replaced by the approximate products $\sim p_x(t)$ and $\sim p_y(t)$ in order to improve computation accuracy. The specific formulas are shown in the following expressions:

$$\begin{aligned} p_x(t) &\approx \sum_{i=0}^{n-1} \frac{1}{n} \left(\iint I_x u_{m,n} v_{m,n} \cdot \varphi'(x, y | t + i) dx dy \right) \\ &\equiv \sim p_x(t), \\ p_y(t) &\approx \sum_{i=0}^{n-1} \frac{1}{n} \left(\iint I_y u_{m,n} v_{m,n} \cdot \varphi'(x, y | t + i) dx dy \right) \\ &\equiv \sim p_y(t), \end{aligned}$$

$$\begin{aligned} \varphi'(x, y | t) &= \frac{1}{|D_{m,n}^Y|} \left(\ell_d \left(\frac{s_u(y - x)}{s} \right) \cdot p(c(y) = p_{m,n} | L(y)) \right). \end{aligned} \quad (13)$$

In this work, continuous displacements between image frames are considered to be very small. Specifically, I_x and I_y will be considered as constants in the time interval $t \sim t + n\sigma$. With this assumption, we can replace $p_x(t)$ and $p_y(t)$ with $\sim p_x(t)$ and $\sim p_y(t)$. The amplitude of optical flow is estimated by considering the previous frame $t - \sigma$. The adaptive parameter n is determined by the optical flow estimation ($V_x(t - \sigma), V_y(t - \sigma)$) from the previous frame. It can be used to estimate the pseudovariation frame rate (v_x, v_y). When computing the product in frame interval $n\sigma$, n is adjusted automatically according to the speed of detected object. For objects moving at high speed, n is small, while, for slow-moving objects, it is larger. Furthermore, $n(v_x, v_y)$ may take on different values at any pixel location because (v_x, v_y) may exhibit spatial variability in the optical flow estimation algorithm. By using a wavelet transform, we can rewrite (5):

$$\begin{aligned} E &= \iint \left\{ a_{m,n} \delta_{m,n}(x, y) \cdot \left(\sum_{m,n} u_{m,n} \delta_{m,n}(x, y) \right)^2 \right. \\ &\quad + b_{m,n} \delta_{m,n}(x, y) \cdot \left(\sum_{m,n} v_{m,n} \delta_{m,n}(x, y) \right)^2 \\ &\quad + 2c_{m,n} \delta_{m,n}(x, y) \cdot \left(\sum_{m,n} u_{m,n} \delta_{m,n}(x, y) \right) \\ &\quad \cdot \left(\sum_{m,n} v_{m,n} \delta_{m,n}(x, y) \right) + d_{m,n} \delta_{m,n}(x, y) \\ &\quad + 2 \sum_{m,n} e_{m,n} \delta_{m,n}(x, y) \cdot \sum_{m,n} u_{m,n} \delta_{m,n}(x, y) \\ &\quad + 2 \sum_{m,n} f_{m,n} \delta_{m,n}(x, y) \cdot \sum_{m,n} v_{m,n} \delta_{m,n}(x, y) \\ &\quad + \lambda \left[\left(\sum_{m,n} u_{m,n} \delta_{m,n}(x, y) \right)^2 \right. \\ &\quad \left. + \left(\sum_{m,n} v_{m,n} \delta_{m,n}(x, y) \right)^2 \right] \Big\} dx dy. \end{aligned} \quad (14)$$

Figure 1 depicts the application of wavelet into optical flow estimation.

The optimal coefficient of $(u_{m,n}, v_{m,n})$ can be determined by the following expression:

$$\text{Coeff}(m, n) = (1 - E(m, n)) \exp \left(- \frac{|u(x, y) - v(x, y)|^2}{\sigma^2} \right), \quad (15)$$

where $m, n = 0, 1, \dots, M - 1$.

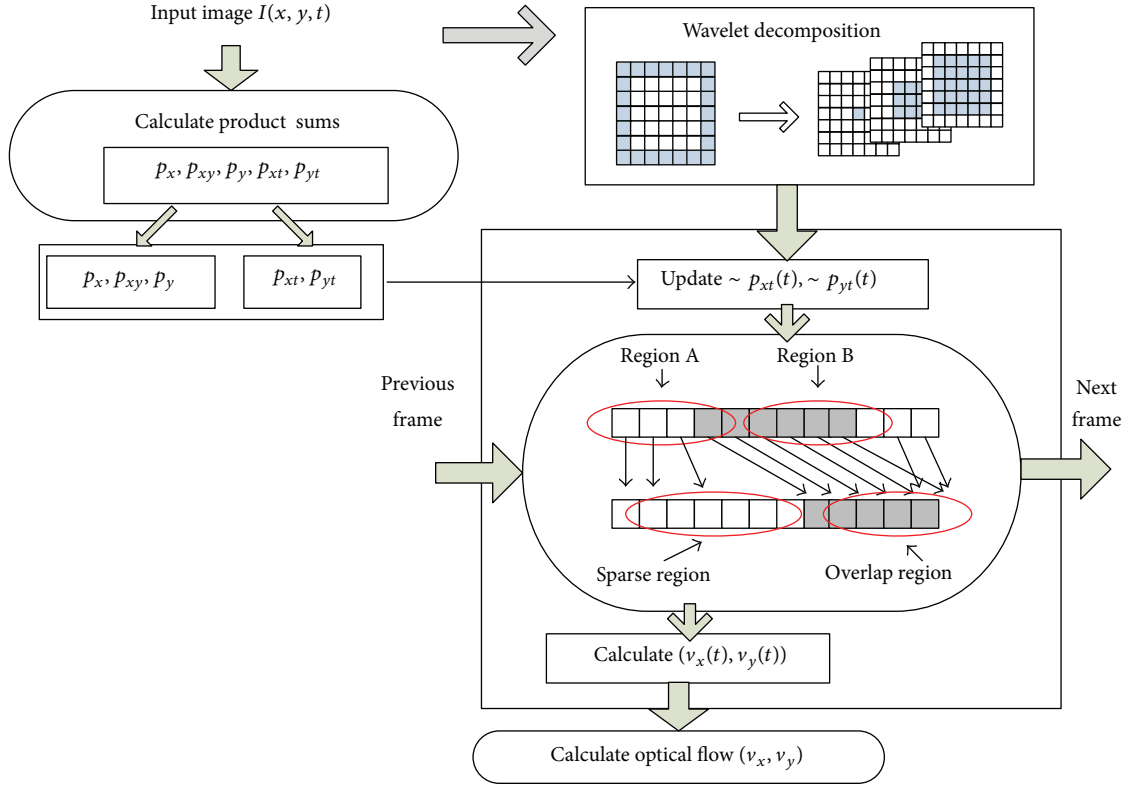


FIGURE 1: Application of wavelet calculations into optical flow estimation.

The sparse representation is $Eu = b$, and the matrix obtained by the wavelet transform is $E \in R^{M^2 \times M^2}$. Elements of matrix E are written as

$$E(m, n) = \arccos \left(\frac{u_m u_n + v_m v_n + 1}{\sqrt{u_m^2 + v_m^2 + 1} \sqrt{u_n^2 + v_n^2 + 1}} \right). \quad (16)$$

The combination of wavelet transform and optical flow defines moving objects via a sparse linear representation in the defined structure. The wavelet algorithm has collected all the information from the optical flow estimation and stored it in the matrix E . Once the computation $Eu = b$ is complete, an optimal and sparse $(u_{m,n}, v_{m,n})$ will have been determined. Our method transforms the optical flow estimation into the problem of minimizing an energy function. The coefficients determination yields accurate optical flow vectors.

3. Classification

3.1. Hybrid Linear-Nonlinear Classifier. The novel hybrid linear-nonlinear classifier (HLNLC) divides the traditional binary classification into two stages: in the first stage, a linear function combines the input feature vector and a scalar variable, and in the second stage, the scalar variable is transformed into a decision variable [25].

In a two-class sorting approach, a particular data set is divided into positive and negative parts. Suppose the feature vector is given by $x = (x_1, x_2, \dots, x_N)^T$, where x represents

the joint outcome of some random variables, where there are N of such variables. The corresponding distributions are described by a multivariable positive normal distribution (where the mean is $u_p = (u_{p1}, u_{p2}, \dots, u_{pN})^T$ and the $N \times N$ covariance matrix is $[C_p]$) and by a multivariable negative normal distribution (where the mean is $u_n = (u_{n1}, u_{n2}, \dots, u_{nN})^T$ and the $N \times N$ covariance matrix is $[C_n]$). The probability density function (PDF) for positive and negative components is $(x | p) \sim N(u_p; [C_p])$ and $(x | n) \sim N(u_n; [C_n])$.

In first stage of the HLNLC, the feature vector x is mapped into a scalar vector y . Because x follows the multivariable normal distribution and y is a linear combination of x , we see that y follows a two-variable normal distribution:

$$\begin{aligned} (yp) &\sim N(u_p; \sigma_p^2), \\ (yn) &\sim N(u_n; \sigma_n^2). \end{aligned} \quad (17)$$

These parameters can be represented by the input vector v and the other related input parameters. They may be written as

$$\begin{aligned} u_p &= v^T u_p; & \sigma_p^2 &= v^T [C_p] v, \\ u_n &= v^T u_n; & \sigma_n^2 &= v^T [C_n] v. \end{aligned} \quad (18)$$

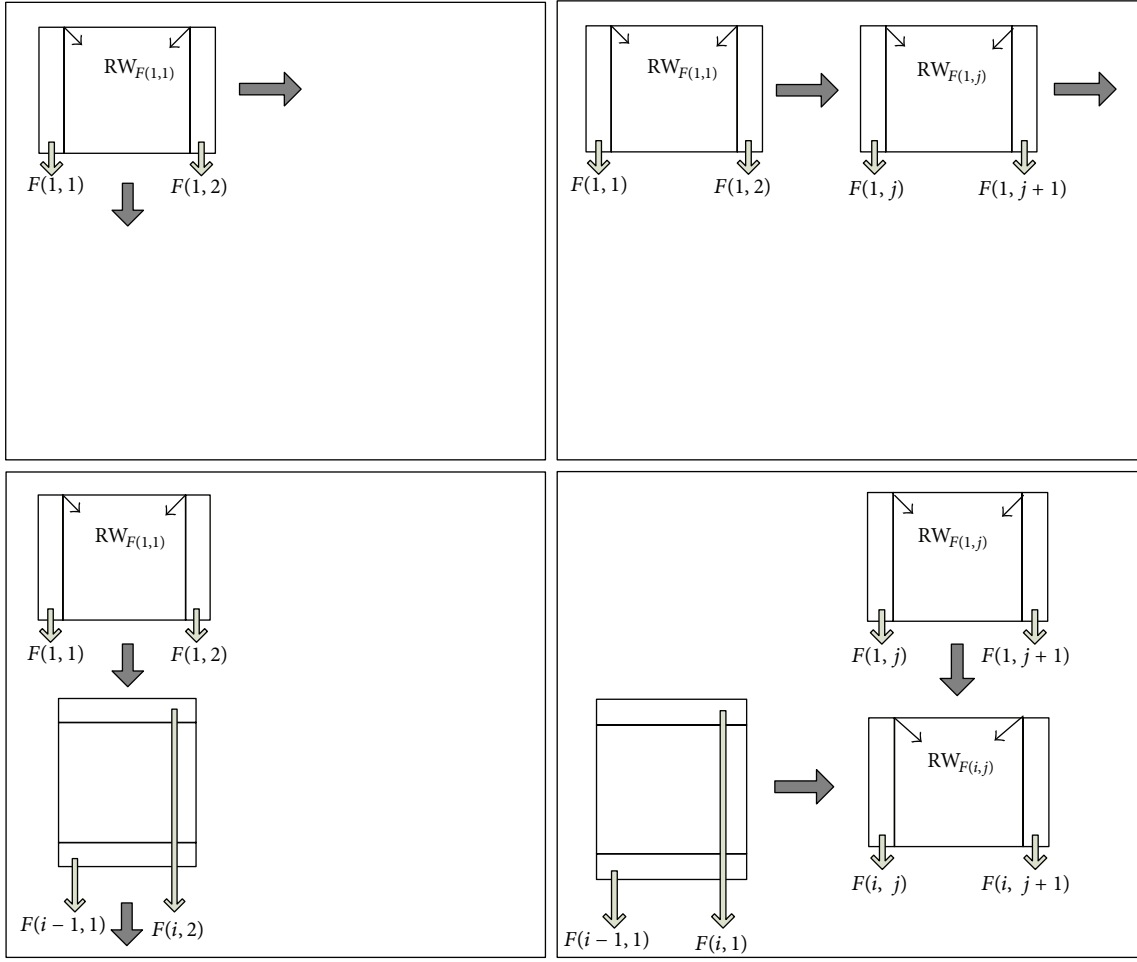


FIGURE 2: Rectangle window scan method.

In [26], the classifier is improved by projecting the multi-variable classification algorithm into a two-density distribution interval, with the projection method described by

$$D(x) = \frac{1}{2}(x - u_p)^T [C_p]^{-1} (x - u_p) - \frac{1}{2}(x - u_n)^T [C_n]^{-1} (x - u_n) + \frac{1}{2} \ln \frac{|[C_p]|}{|[C_n]|}. \quad (19)$$

In a second stage of the HLNLC algorithm, the likelihood ratio of y is used as a decision variable:

$$LR_y = \frac{f(y | p)}{f(y | n)} = \frac{\sigma_p}{\sigma_n} \exp \left\{ -\frac{1}{2} [(by - a)^2 - y^2] \right\}. \quad (20)$$

Based on binormal ROC theory, the corresponding P_{AUC} can be expressed as

$$P_{AUC} = \Phi \left(\frac{d_a}{\sqrt{2}} \right) + 2F \left(-\frac{d_a}{\sqrt{2}}, 0; -\frac{1 - c^2}{1 + c^2} \right), \quad (21)$$

where the d_a and c are

$$d_a = \frac{|\sqrt{2}(u_p - u_n)|}{\sqrt{\sigma_p^2 + \sigma_n^2}}, \quad c = \frac{\sigma_p - \sigma_n}{\sigma_p + \sigma_n}. \quad (22)$$

$F(X, Y; \rho)$ is the cumulative distribution function (CDF) of the standard normal distribution. Using the parameters (u_p, C_p, u_n, C_n) , we may express the AUC as a function of the linear coefficient vector v :

$$AUC_{HLNLC} = \Phi \left(\frac{d_a(v)}{\sqrt{2}} \right) + 2F \left(-\frac{d_a(v)}{\sqrt{2}}, 0; -\frac{1 - c(v)^2}{1 + c(v)^2} \right), \quad (23)$$

$$d_a(\vec{v}) = \frac{|\sqrt{2}v^T(u_p - u_n)|}{\sqrt{v^T([C_p] + [C_n])v}}, \quad (24)$$

$$c(\vec{v}) = \frac{\sqrt{v^T([C_p])v} - \sqrt{v^T([C_n])v}}{\sqrt{v^T([C_p])v} + \sqrt{v^T([C_n])v}}. \quad (25)$$

In order to find the optimal linear function of the HLNLC, which mainly focuses on v and AUC_{HLNLC} ,

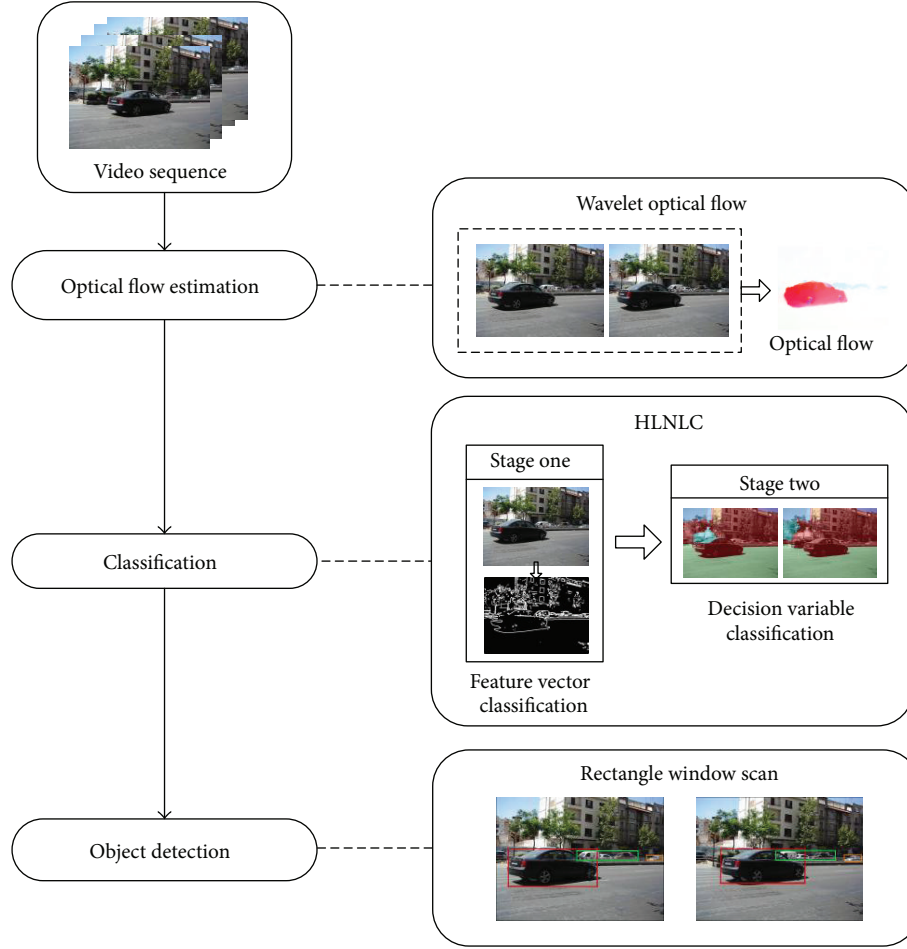


FIGURE 3: The whole experiment flowchart.

the HLNLC algorithm classifies the optimization problem as follows:

$$\vec{v}_{\text{HLNLC}} = \arg \max_{\vec{v}} \text{AUC}_{\text{HLNLC}}(\vec{v}). \quad (26)$$

The optimization problem is given by $\partial \text{AUC}_{\text{HLNLC}} / \partial \vec{v} = 0$, which can be solved with gradient-based mathematical methods. The specific process is shown in the following expressions:

$$\begin{aligned} \frac{\partial \text{AUC}_{\text{HLNLC}}}{\partial v_i} &= \frac{\partial \text{AUC}_{\text{HLNLC}}}{\partial d_a} \frac{\partial d_a}{\partial v_i} + \frac{\partial \text{AUC}_{\text{HLNLC}}}{\partial c} \frac{\partial c}{\partial v_i}, \\ \frac{\partial \text{AUC}_{\text{HLNLC}}}{\partial d_a} &= \begin{cases} \frac{\exp(-d_a^2/4)}{2\sqrt{\pi}}, & (c \neq 0), \\ \frac{\exp(-d_a^2/4)}{2\sqrt{\pi}} \left\{ 1 - 2\Phi\left(\frac{\rho d_a}{\sqrt{2(1-\rho^2)}}\right) \right\}, & (c = 0), \end{cases} \\ \frac{\partial d_a}{\partial v_i} &= \frac{2(\vec{v} \cdot \Delta u)(v^T([C_p] + [C_n])v) - 2(\vec{v} \cdot \Delta u) \sum_{i=1}^n C_{ij} v_j}{d_a(v^T([C])v)^2}, \end{aligned}$$

$$\begin{aligned} \frac{\partial \text{AUC}_{\text{HLNLC}}}{\partial c} &= \begin{cases} 0, & (c \neq 0), \\ \frac{2 \operatorname{sign}(c)}{\pi(1+c^2)} \exp\left(-\frac{d_a^2}{4(1-\rho^2)}\right), & (c = 0), \end{cases} \\ \frac{\partial c}{\partial v_i} &= \frac{2}{(\sigma_p + \sigma_n)^2} \left[\frac{\sigma_p \sum_{i=1}^n C_{ij} v_i - \sigma_n \sum_{i=1}^n C_{ij}^p v_i}{\sigma_p - \sigma_n} \right]. \end{aligned} \quad (27)$$

3.2. Classification of Optical Flow Vectors. In this paper, the linear coefficient vector v is determined by the parameter θ for a 2D optical flow vector $(u_{m,n}, v_{m,n})$, which can be understood to be the angle between vector coordinate x and vector coordinate y . Any 2D optical flow vector can be normalized as

$$y = (\vec{v})^T \vec{x} = u_{m,n} \cos \theta + v_{m,n} \sin \theta. \quad (28)$$

The operating assumption in the HLNLC algorithm is that two types of feature data follow a pair of multivariable normal distributions. However, the optical flow distributions contain some small differences. We can implement a more robust method. The pair of optical flow vector variables

uses bivariate normal distributions to normalize the data. Specifically,

$$(u_{m,n}, v_{m,n} | \rho) \sim N \left(\begin{bmatrix} \frac{1}{u_{m,n}^2} & \frac{\rho_{u,v}}{(u \cdot v)_{m,n}} \\ \frac{\rho_{u,v}}{(u \cdot v)_{m,n}} & \frac{1}{v_{m,n}^2} \end{bmatrix} \right). \quad (29)$$

Here, $\rho_{u,v}$ represents feature covariance of the optical flow $(u_{m,n}, v_{m,n})$. Using this method, $(u_{m,n}, v_{m,n})$ is normalized into the normal distribution. The parameters $(u_p, \sigma_p, u_n, \sigma_n)$ that concern the HLNLC algorithm can be obtained with (18), in which positive values indicate motion areas and negative values represent background. AUC_{HLNLC} can be calculated by using (23). The related parameter $(u(\theta_{\text{HLNLC}}), v(\theta_{\text{HLNLC}}))$ is used to produce an ROC curve and calculate AUC_{HLNLC} .

4. Rectangle Window Scan

In order to detect moving objects, an $M \times N$ rectangle window should be determined. We propose rectangle window scan algorithm; in scanning process, the classified optical flow vectors which are attributed to moving object are the input variables, and a rectangle window will shift n pixel locations in each direction per unit time. The operation is repeated until the rectangle window size is less than the presupposed threshold value. The detection area marked by the rectangle window will be the final output. Our proposed method could detect multiple moving objects in one scene. However, the iterative process may be time consuming. Figure 2 illustrates the details of rectangle window scan algorithm.

In the method, the i th focused area at the j th scan line is denoted by $F(i, j)$, and the related rectangle window is recorded as $RW_{F(i,j)}$. Initially, the scan area $RW_{F(1,1)}$ is obtained by a normal adaptive modification calculation method. Thereafter, the rectangle window shifts to the right by x pixels, where $x = 10$ was used in this paper, and $RW_{F(1,1)}$ is shifted to $RW_{F(1,2)}$. There is no need to recalculate the overlap region of an integral scan because the two adjacent regions $RW_{F(1,1)}$ and $RW_{F(1,2)}$ share the same overlap region $(RW_{1-x}) \times N$. For the remaining horizontal scan lines, $F(i, x)$ describes the region of concern which is below $F(i-1, x)$. The upper and lower boundaries of the nonoverlapping region are defined as $H(i, x)$ and $L(i, x)$.

Based on the distribution principle for motion vectors, less than 50% of the motion vectors may be zero vectors. We use the sum of absolute gradient difference (SAGD) to judge motion vector value. SAGD can be calculated in the following expression:

$$\begin{aligned} & \text{SAGD}(u, v) \\ &= \max \left\{ \begin{aligned} & \sum_{x=0}^M \sum_{y=0}^N \{|u(i+x, j+y) - u(i+x+a, j+y+a)|\} \\ & \sum_{x=0}^M \sum_{y=0}^N \{|v(i+x, j+y) - v(i+x+a, j+y+a)|\} \end{aligned} \right\}. \end{aligned} \quad (30)$$

Depending on the current location of rectangle window, some adjacent windows may not exist. We use windows

TABLE 1: Optical flow computation time comparison.

Video	LK	HS	OAOF	Proposed
Cars	20.52 sec	21.73 sec	25.36 sec	15.49 sec
Coastguard	15.66 sec	16.21 sec	24.55 sec	10.24 sec
Satellite	13.25 sec	15.44 sec	20.15 sec	8.32 sec
Running	14.37 sec	16.15 sec	22.32 sec	8.53 sec

adjacent to block combination to estimate the object motion, using the formula

$$H_{F(i,1)} = H_{F(i-1,1)} - \text{SAG}_{H(i-1,1)} + \text{SAG}_{L(i-1,1)}. \quad (31)$$

For the remaining areas, $F(i, j)$, with $i, j = 2, 3, \dots, H_{F(i,j)}$, the moving object region is identified as

$$\text{RW} = \sum_{i=1}^M \sum_{j=1}^N (H_{F(i,j-1)} - H_{F(i,j)} + L_{F(i,j)}). \quad (32)$$

5. Experimental Results

In this section, we validate the performance of our proposed algorithm on four different videos. All of the source videos were presented in a consistent format (MPEG-2 standard, 25.68 frames per second (fps)). Video (a) was produced with cameras where high-speed moving cars appear in both close and distant scenes, it has 996 frames of 768×576 resolution. Video (b) is a standard video compression sequence known as the coastguard sequence, where a video camera is fixed on a moving boat so it appears that the background is moving; it has 876 frames of 768×576 resolution. Video (c) is a spatial satellite video sequence, and the satellite is the detected object; it has 1025 frames of 720×480 resolution. Video (d) is a human-motion video in which the motion is a human running; it has 825 frames of 720×480 resolution. These four videos are the most tested videos, in which the pedestrians, vehicles, and satellites are typical detected targets. Comparison experiments with other state-of-the-art object detection algorithms could demonstrate the efficiency of our proposed method. Specific steps are shown in Figure 3.

5.1. Optical Flow Experiments. We use a constant value a to determine the time interval between video frames, and the parameter value $a = 0.64$ pixel/ms was found to be suitable. Each frame was processed with subsampling and rescaled under the wavelet estimation. Two important steps are involved: (1) normalization of optical flow vectors and computation of the matrix E and (2) computation of the spare representation of $Eu = b$. Our results show that cars and human bodies can be detected in (a) and (d). Different frequency components in unstable regions and jitters have been removed in sequences (c). Compared with other video sequences, detected objects in sequence (b) have different characteristics. The optical flow is detected for background area. The experimental results are shown in Figure 4.

Computation time comparisons results are shown in Table 1. We compare our proposed method with LK, HS,

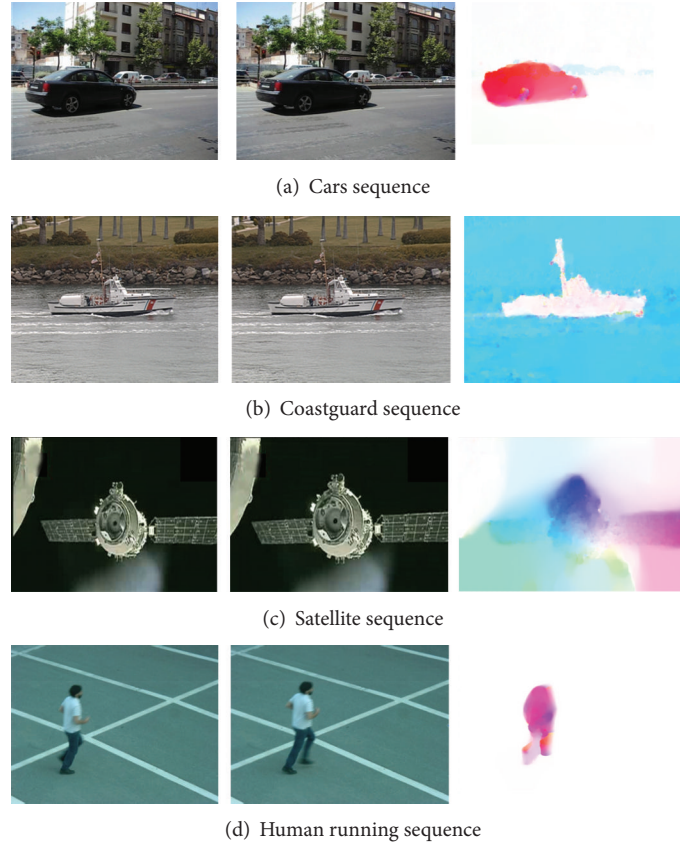


FIGURE 4: Optical flow estimation for different sequences.

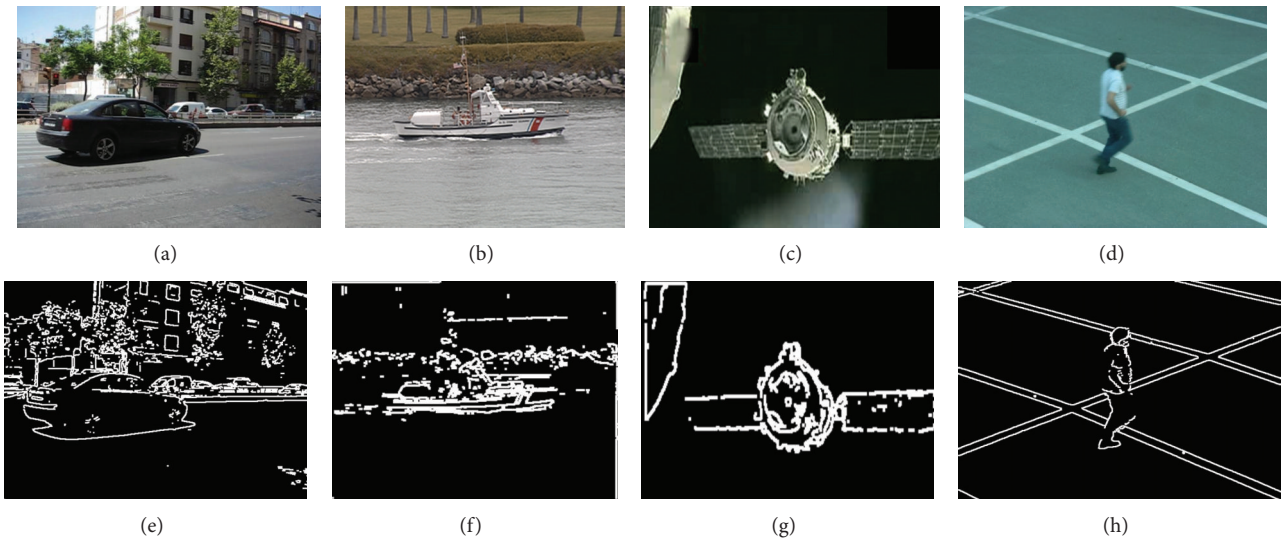


FIGURE 5: Initial classification results: (a), (b), (c), and (d) single frame in test video sequences; (e), (f), (g), and (h) corresponding classification result.

and occlusion-aware optical flow (OAOF) [27]. LK and HS calculations take almost the same time because they involve similar computation methods. OAOF takes a little longer time. Our proposed wavelet optical flow method has an improved computation time, with a reduction of nearly 5~6 sec over LK and HS and 10~14 sec over OAOF. It clearly

demonstrates the efficiency of using wavelet transform in optical flow estimation.

5.2. Classification Experiments. In the first stage of HLNL algorithm, structure optimal parameters are chosen as $u = 0.64$, and the classifier undergoes 20 iterations. In order to get

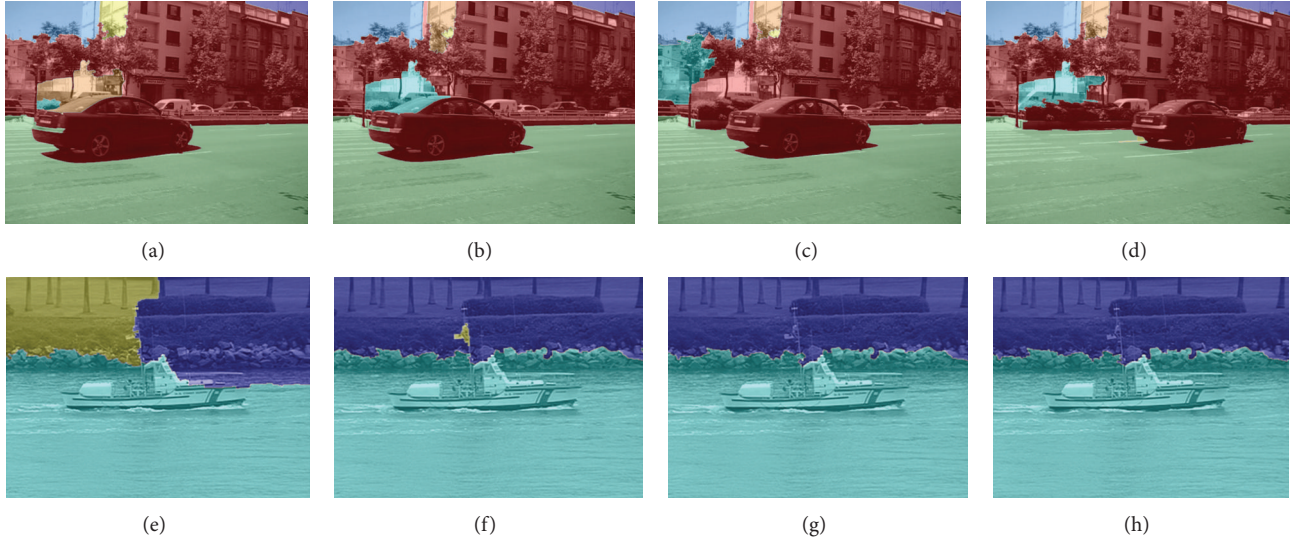


FIGURE 6: Second stage classification results: (a), (b), (c), and (d) frames 7, 9, 11, and 25 of cars sequences (e), (f), (g), and (h) frames 95, 97, 98, and 100 of coastguard sequence.

TABLE 2: Classification performance of PCNN, FNN, GSVM, LDA, and HLNL.

Video	PCNN	FNN	GSVM	LDA	HLNL
Cars					
M and SD	57.2 ± 16.7	55.4 ± 15.5	41.3 ± 28.5	37.8 ± 11.9	36.2 ± 10.4
AUC	0.65	0.53	0.77	0.78	0.81
DR	72.33%	71.96%	84.81%	88.27%	89.32%
Coastguard					
M and SD	44.3 ± 13.2	42.9 ± 12.4	34.7 ± 25.6	32.6 ± 12.5	31.4 ± 11.7
AUC	0.72	0.69	0.81	0.80	0.85
DR	73.43%	72.16%	84.54%	87.29%	89.52%
Satellite					
M and SD	32.3 ± 10.3	30.7 ± 10.7	25.8 ± 21.6	31.3 ± 13.8	30.2 ± 13.4
AUC	0.71	0.61	0.83	0.80	0.82
DR	72.33%	73.48%	84.49%	86.25%	88.64%
Running					
M and SD	35.7 ± 11.4	33.6 ± 10.9	25.5 ± 22.6	33.4 ± 16.2	32.7 ± 15.9
AUC	0.76	0.64	0.85	0.83	0.84
DR	72.31%	73.14%	84.39%	88.17%	89.09%

a dense optical flow field, our method uses a global smoothness constraint. Each optical flow vector has a spatial connection with respect to high-speed moving object. Figure 5 shows the initial classification results. As background information and moving object features have similar characteristics, motion regions adhere and empty holes emerged. Obviously, classification result is not very satisfactory.

In the second stage, the classifier uses a decision variable to deal with topology changes in detection region. Figure 6 shows the classification results about video sequences (a) and (b). Motion areas in video (a) contain queues of cars, which belong to close scene and distant scene, respectively. Because the cars in distant scene are too wide, some background information has also been classified as motion region. Optical

flow vectors generated from video (b) reflect background information. We employ a trust-region Newton-Raphson method to solve this problem [28]. This method separates moving objects from background by making judgments about the most meaningful trust-region in sequences based on the maximum likelihood ratio.

We are interested in comparing the classification performance of HLNL and other classifiers, including PCNN, FNN, GSVM, and LDA. We specified the population parameters of a pair of normal distributions (μ_p, C_p, μ_n, C_n) and drew samples from the specified distributions. Then, different classifiers are applied to each optical flow feature vector in the sampled data. The comparison results are shown in Table 2. M&SD are the mean and standard deviation of the motion

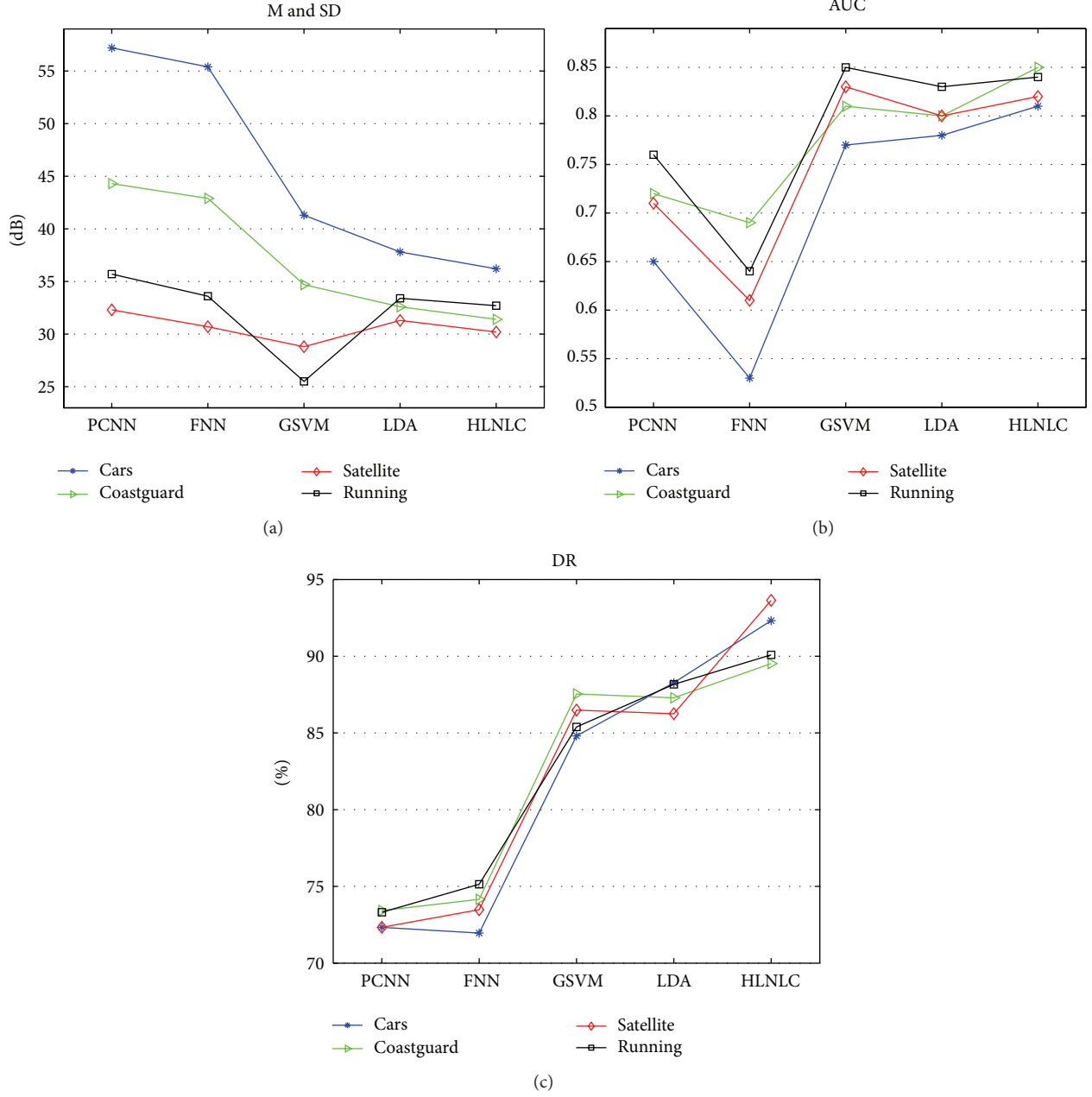


FIGURE 7: M&SD, AUC, and DR curves obtained by different methods.

region (positive region), and AUC is calculated based on M&SD:

$$\text{AUC} = \Phi \left(\frac{|u_p - u_n|}{\sqrt{\sigma_p^2 + \sigma_n^2}} \right), \quad (33)$$

$$\text{DR} = \prod_{i=1}^N \left(\frac{b_i}{\sqrt{2\pi}} \right) \exp \left\{ -\sum_{i=1}^N \frac{(\text{AUC}_i - \sigma)^2}{N} \right\}.$$

The function Φ is the cumulative distribution function of the standard normal distribution. Positive values indicate motion areas, and negative values represent background.

AUC, as a decision variable of normal distribution, equals the probability that outcomes from the actual positive class. A greater AUC means higher classification accuracy. Based on AUC, we were able to calculate the detection rate (DR), which is defined as the ratio between total moving object region and background area.

Our experiments use the mean AUC for each sequence which has different frame numbers, respectively. Neural network algorithms (PCNN\FNN) give rise to proliferation errors in the edges of moving objects and have larger deviations in their classification results. GSVM can directly classify data without using PCA, so the result is better than what is produced by NN. About LDA and HLNLC, we calculate AUC

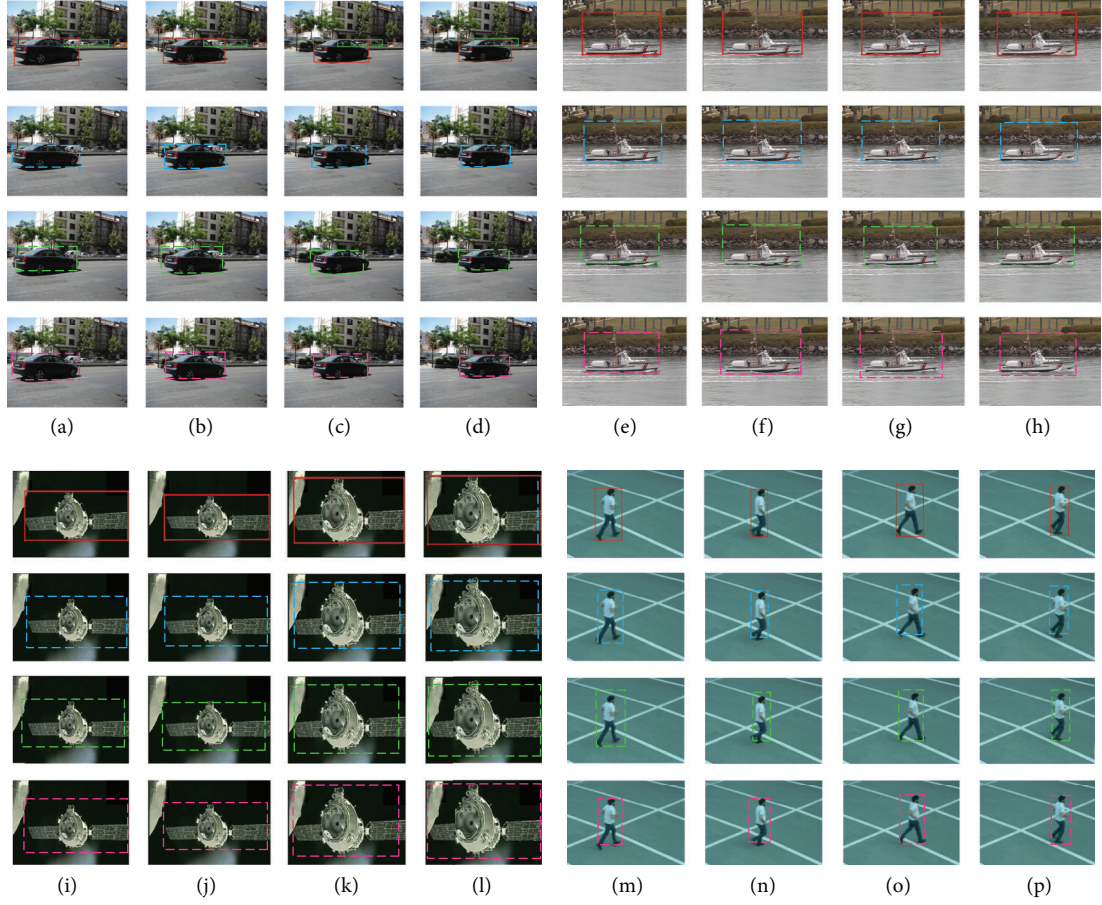


FIGURE 8: Object detection results: (a)–(d) frames 7, 10, 14, and 17 of cars sequence, (e)–(h) frames 17, 26, 35, and 45 of coastguard sequence, (i)–(l) frames 27, 35, 54, and 69 of human running sequence, and (m)–(p) frames 29, 33, 40, and 44 of human walking sequence.

by inserting trained LDA vectors, and AUC_{HLNLC} is found to be superior to AUC_{LDA} . In all the methods, We observe that the HLNLC can substantially improve classification performance over other classifiers. M&SD, AUC, and DR curves obtained by different methods are shown in Figure 7.

5.3. Rectangle Window Scan Experiments. In this part, we used three other object detection methods for comparison. These included SIFT, background subtraction (BS), and Hough forest method (HF). The SIFT method is based on SIFT features, which is invariant to image scale and rotation [29, 30]. The background subtraction method uses temporal differencing pixels from a Laplacian model and completes the object detection task via a threshold value [31]. Hough forests can be regarded as a task-adapted codebook, in which different locations, scales, and motions are stored [32]. Figure 8 shows the experimental results; the object detection results obtained by our proposed method are shown with red solid boxes; the results in SIFT, BS, and HF are shown in blue, green, and pink dotted boxes. Our proposed method could detect moving objects in different distant scenes in the same sequence; in video sequence (a), moving cars in the distant scene were detected in green and orange solid

rectangle windows. In addition, our proposed method could adaptively adjust the rectangle window that will be suitable for the detected object, such as satellite and human body in video sequences (c) and (d).

We use three measures to compare object detection accuracy [33]:

$$\begin{aligned}
 \text{OverLapRatio} &= \sum_{i=1}^{N_{\text{frame}}} \frac{|G_i^t \cap O_i^t|}{|G_i^t \cup O_i^t|}, \\
 \text{FDA}(t) &= \frac{\text{OverLapRatio}}{|(N_G^t + N_O^t)/2|}, \\
 \text{SFDA} &= \sum_{i=1}^{N_{\text{frame}}} \frac{\text{FDA}(t)}{\exists(N_G^t \text{ OR } N_O^t)}, \\
 \text{ODA} &= 1 - \frac{m_t + f_{pt}}{N_G^t}, \\
 \text{ODP} &= \frac{\text{OverLapRatio}}{N_{\text{mapped}}^t}.
 \end{aligned} \tag{34}$$

Here, G_i^t and O_i^t represent i th ground-truth and detected object in frame t . N_G^t and N_O^t represent the number of

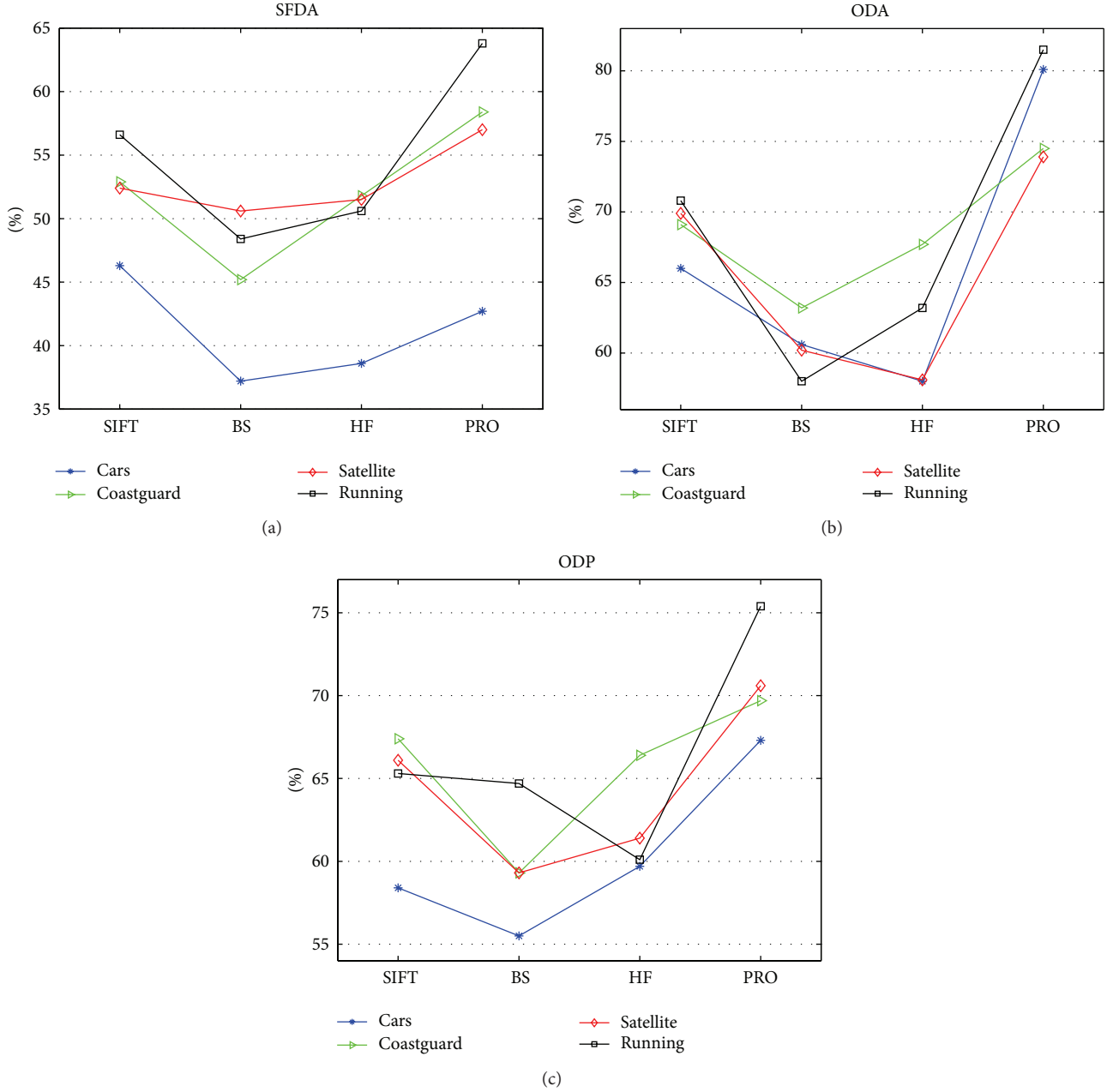


FIGURE 9: SFDA, ODA, and ODP curves obtained by different methods.

“ground-truth” and detected object in frame t . f_{pt} is the number of classification errors. m_t is the missed detection count. N_{mapped}^t is the pair number of ground-truth and detected object in frame t . OverLapRatio is the quality of alignment between the detected objects and the ground-truth. SFDA is the detection accuracy for a video sequence, which is essentially the average of FDA over all of the relevant frames in the sequence; ODA is the object detection accuracy, which utilized the missed detection and classification error counts. ODP is the object detection precision, which gave us the precision of detection by taking into account the spatial overlap information between ground-truth and system output.

Experimental results are shown in Table 3. Compared with SIFT, our proposed algorithm increases SFDA, ODP, and ODA by about 3%~7%, 5%~10%, and >10%. Compared with BS and HF, the SFDA, ODP, and ODA are increased by about 5%~15%, 8%~15%, and 10%, respectively. It implies that our proposed algorithm has a higher detection accuracy and lower false detection rate. SFDA, ODP, and ODA curves obtained by different methods are shown in Figure 9.

6. Conclusions

In this paper, we propose a new computational intelligence method for object detection. We apply the wavelet transform

TABLE 3: Detection results of different methods.

Video	SIFT	BS	HF	PRO
Cars				
SFDA	46.3%	37.2%	38.6%	42.7%
ODA	71.0%	65.6%	63.0%	85.1%
ODP	58.4%	55.5%	59.7%	67.3%
Coastguard				
SFDA	52.9%	45.2%	51.8%	58.4%
ODA	74.1%	68.2%	72.7%	79.5%
ODP	67.4%	59.3%	66.4%	69.7%
Satellite				
SFDA	52.4%	50.6%	51.5%	57.0%
ODA	74.9%	65.2%	63.1%	78.9%
ODP	66.1%	59.3%	61.4%	70.6%
Running				
SFDA	56.6%	48.4%	50.6%	63.8%
ODA	75.8%	63.0%	68.2%	86.5%
ODP	65.3%	64.7%	60.1%	75.4%

into optical flow estimation. Our proposed method is able to estimate the motions of different speed moving objects in the same scenes and achieve an accelerated optical flow computation. In the classification stage, HLNLC method is used to classify optical flow vectors, and it yields higher accuracy than other existing classifiers. Experimental results demonstrate that the combination of wavelet optical flow estimation and HLNLC classification can achieve more accurate and precise object detection.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) (2012CB821206), the National Natural Science Foundation of China (no. 91024001, no. 61070142) and the Beijing Natural Science Foundation (no. 4111002).

References

- [1] P. J. Werbos, "Computational intelligence for the smart grid-history, challenges, and opportunities," *IEEE Computational Intelligence Magazine*, vol. 6, no. 3, pp. 14–21, 2011.
- [2] A. Kusiak and F. A. Salustri, "Computational intelligence in product design engineering: review and trends," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 37, no. 5, pp. 766–778, 2007.
- [3] M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "Solution of fractional order system of Bagley-Torvik equation using evolutionary computational intelligence," *Mathematical Problems in Engineering*, vol. 2011, Article ID 675075, 18 pages, 2011.
- [4] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. 900–903, September 2002.
- [5] X. Tan, J. Li, and C. Liu, "A video-based real-time vehicle detection method by classified background learning," *World Transactions on Engineering and Technology Education*, vol. 6, no. 1, pp. 189–192, 2007.
- [6] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs for salient object detection in images," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3232–3242, 2010.
- [7] K. Schindler and D. Suter, "Object detection by global contour shape," *Pattern Recognition*, vol. 41, no. 12, pp. 3736–3748, 2008.
- [8] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conferences on Artificial Intelligence*, pp. 674–679, 1981.
- [9] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [10] S. V.-U. Ha and J. W. Jeon, "Readjusting unstable regions to improve the quality of high accuracy optical flow," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 4, pp. 540–547, 2010.
- [11] H. S. Ranganath and G. Kuntimad, "Object detection using pulse coupled neural networks," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 615–620, 1999.
- [12] C.-F. Juang and C.-M. Chang, "Human body posture classification by a neural fuzzy network and home care system application," *IEEE Transactions on Systems, Man, and Cybernetics Part A*, vol. 37, no. 6, pp. 984–994, 2007.
- [13] C.-F. Juang, S.-H. Chiu, and S.-W. Chang, "A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 5, pp. 998–1008, 2007.
- [14] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 862–873, 2009.
- [15] S. Pang, S. Ozawa, and N. Kasabov, "Incremental linear discriminant analysis for classification of data streams," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 5, pp. 905–914, 2005.
- [16] Y.-T. Wu, T. Kanade, J. Cohn, and C.-C. Li, "Optical flow estimation using wavelet motion model," in *Proceedings of the IEEE 6th International Conference on Computer Vision*, pp. 992–998, January 1998.
- [17] C. P. Bernard, "Discrete wavelet analysis: a new framework for fast optic flow computation," in *Proceedings of the European Conference on Computer Vision*, pp. 354–368, Freiburg, Germany, June 1998.
- [18] S. Srinivasan and R. Chellappa, "Optical flow using overlapped basis functions for solving global motion problems," in *Proceedings of the European Conference on Computer Vision*, pp. 288–304, Freiburg, Germany, June 1998.
- [19] K. Fukunaga and R. R. Hayes, "Effects of sample size in classifier design," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 873–885, 1989.
- [20] W. Chen, C. E. Metz, M. L. Giger, and K. Drukker, "A novel hybrid linear/nonlinear classifier for two-class classification: theory, algorithm, and applications," *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 428–441, 2010.
- [21] S.-H. Lai and B. C. Vemuri, "Reliable and efficient computation of optical flow," *International Journal of Computer Vision*, vol. 29, no. 2, pp. 87–105, 1998.
- [22] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 123–151, 2005.

- [23] L.-F. Chen, H.-Y. M. Liao, and J.-C. Lin, "Wavelet-based optical flow estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 1, pp. 1–12, 2002.
- [24] S. G. Mallat, "Theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
- [25] W. Chen, C. E. Metz, and M. L. Giger, "Hybrid linear classifier for jointly normal data: theory," in *Computer-Aided Diagnosis*, vol. 6915 of *Proceedings of SPIE*, pp. 1–4, February 2008.
- [26] C.-F. Juang and G.-C. Chen, "A TS fuzzy system learned through a support vector machine in principal component space for real-time object detection," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3309–3320, 2012.
- [27] S. Ince and J. Konrad, "Occlusion-aware optical flow estimation," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1443–1451, 2008.
- [28] D. M. Gay, "Subroutines for unconstrained minimization using a model/trust-region approach," *Association for Computing Machinery. Transactions on Mathematical Software*, vol. 9, no. 4, pp. 503–524, 1983.
- [29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [30] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [31] F.-C. Cheng, S.-C. Huang, and S.-J. Ruan, "Scene analysis for object detection in advanced surveillance systems using laplacian distribution model," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 41, no. 5, pp. 589–598, 2011.
- [32] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.
- [33] R. Kasturi, D. Goldgof, P. Soundararajan et al., "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, 2009.

