Robust Video Target Tracking Based on Multi-feature Fusion and H_∞ Filtering

Howard Wang*, Sing Kiong Nguang
The Department of Electrical and Computer Engineering, the University of Auckland, 1142, New Zealand.

* Corresponding author. Tel.: (+64)0220348016; email: hwan733@aucklanduni.ac.nz

Manuscript submitted November 3, 2014; accepted June 23, 2015.
doi: 10.17706/ijcce.2016.5.2.79-98

Abstract: In this paper, a new video tracking method based on multi-feature fusion and H_∞ filtering is proposed. We extract color spatial distribution feature, target contour feature and edge gradient histogram of video targets, and calculate the matching degree between the candidate feature and target feature. The color spatial distribution feature includes not only the color statistics but also the pixels position information, which can avoid the misjudgment caused by only using color statistics matching, and hence improve the color matching accuracy. An efficient contour extraction method based on wavelet transform combined with three consecutive frames difference and dynamic background updating is introduced as well. The normalized divisional contour mean value vector is employed to calculate the contour matching degree. Moreover, target edge gradient histogram is calculated to further improve the matching accuracy. We fuse these three features and calculate the final matching degree by using linear weighted fusion method. To boost the efficiency of features extraction and fusion, thread pool and multi-thread synchronization are adopted. A robust video target motion state estimation method based on H_∞ filtering is presented to recursively estimate and predict the target state which can narrow the scope of features searching and then greatly improve the efficiency and accuracy of features extraction. Compared with the Kalman filter, the H_∞ filter makes no assumptions in process and measurement noise but has the similar efficient recursive equations. Therefore when the noises are non-Gaussian distributed, the H_∞ filter-based visual tracking systems have better performance in robustness.

Key words: Video tracking, multi-feature fusion, color spatial distribution feature, contour extraction, edge gradient histogram, H_∞ filtering.

1. Introduction

Video tracking refers to the process of measuring the real-time locations of moving targets over time. Video sequence consists of consecutive frames which can be separated into foreground and background. The moving targets belong to the foreground and the rest of image belongs to the background. So in other words, video target tracking is to detect, associate and locate the foreground targets in real time. Video targets detection refers to utilizing image processing technology, combining with intra-frame and inter-frame analysis methods to discover the moving targets in video sequence. A high-performance video detection algorithm is beneficial to the robust and accurate video tracking. Likewise, video tracking calculates the real-time locations of moving targets, which helps to narrow the search range and boost the search speed for video detection. Therefore, detection and tracking can complement and promote each other [1]. Since the frame-rate of video sequence can reach up to 25 per second and the image signal is susceptible to all kinds of ambient factors, hence the video tracking systems have high requirements in accuracy, robustness and real-time performance. Current researches for video tracking mainly focus on the following aspects [2]: video tracking based on regional correlation; video tracking based on feature matching; optical flow calculation; motion estimation and prediction.

The principal idea of regional correlation is to search current frame to find out the regions which have the maximum correlation with the regions of interest in reference frame [3]. The specific approach is to move the target region of reference frame along the current frame and calculate the regional correlation; the position with maximum correlation is the offset of target relative to the reference frame. Regional correlation-based tracking method is insensitive to the slight change of illumination, but susceptible to the deformation and scaling. Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene [4]. Optical flow reflects the relationship between gray scale change and target motion. To represent gray scale instantaneous gradient of target edge, optical flow vector is employed. All optical flow vectors constitute the optical flow field. J. L. Barron *et al.* summarized current optical flow researches and divided them into four categories [5]: differential method, region-based block matching method, energy-based method and phase-based method. Usually, video tracking methods based on optical flow calculation outperform other methods in target detection accuracy, but fall behind in real time performance.

Features matching based tracking methods extract target features which can describe video target robustly, accurately and discriminatively, and then calculate the matching degree between the candidate target and reference target by using the efficient matching algorithms. Video target features can be classified as visual features, statistical features, transform coefficient features, geometric features, algebraic features and dynamic features, among which visual features such as the color histogram are the most frequently used features. Actually, feature matching-based tracking experiences a complex process of feature extraction, target modeling, target matching and model updating. The moving targets may be deformable and scaled, as well as, inevitably affected by ambient noises and interferences. Therefore, the target features can hardly remain unchanged. Single feature is insufficient to describe video target. It is preferred to select a group of features which are relatively stable and easily extracted to describe the moving targets efficiently and accurately. For rigid targets, the corner, scaling and normalized color histogram can be considered as stable features. The extracted feature group should be processed by an efficient fusion algorithm to realize real time matching and tracking. Multi-feature fusion strategy is more robust and preferred to describe moving targets, especially the detection and tracking for non-rigid targets.

Motion estimation and prediction methods can be divided into two types: intra-frame estimation and inter-frame motion estimation. Intra-frame estimation and prediction is based on the estimation of motion blocks, which helps to reduce the code rate [6]. However in practical applications, intra-frame estimation and prediction can lead to a large amount of calculation due to the intra-frame block matching and compensation operation. Inter-frame estimation mainly employs state filters to estimate and predict the motion state of moving targets. The objective of state filtering is to remove the noise from the measurements and produce more accurate estimation. The filter can predict the motion state of targets in the next frame via combining current measurements with the estimated value at previous frame.

Video tracking systems face lots of challenges such as target occlusion, deformation, scaling, and illumination variation. In this paper, we propose a robust video tracking algorithm based on multi-feature fusion and H_∞ filtering. In Section 2, the color spatial distribution feature, contour feature and edge gradient feature are efficiently extracted. In Section 3, the multi-feature fusion method is proposed and optimized to calculate the matching degree and update the target feature. The video target motion state estimation based on H_∞ filtering is discussed in Section 4. The performance evaluation of the video tracking algorithm is presented in Section 5 and the conclusion is given in Section 6.

2. Fast Extraction of Multiple Features for Video Targets

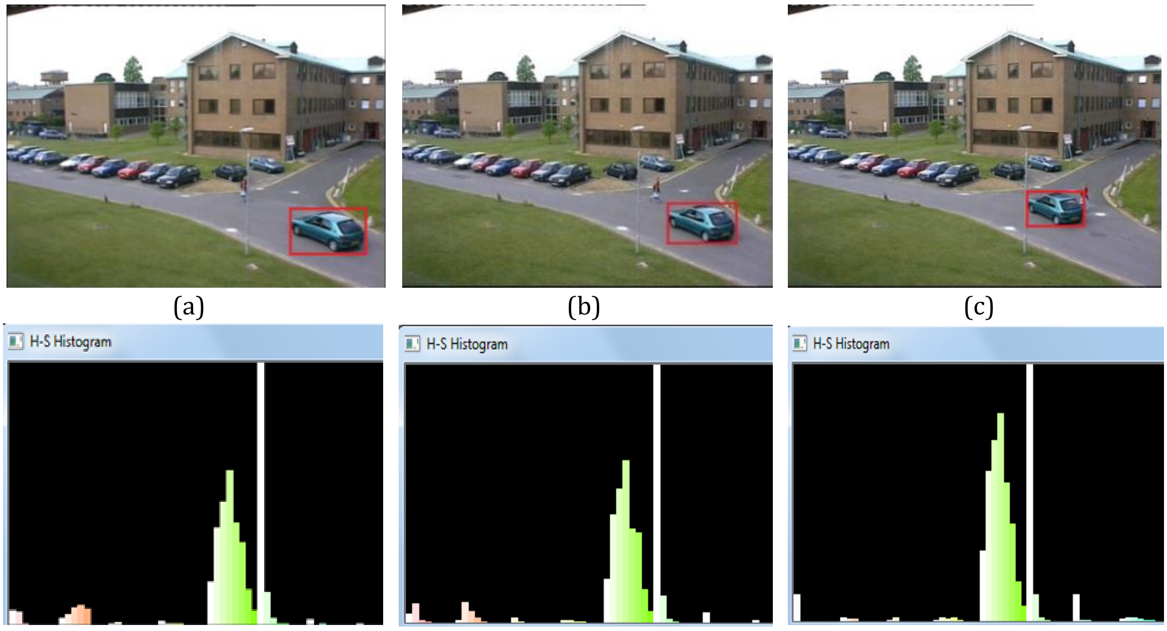
2.1. Color Spatial Distribution Feature Extraction and Matching

Usually, different moving targets have different color distributions and the color histogram statistics can be used to represent the color distribution. The color space is divided into several color ranges or called 'bins', each of which indicates the number of pixels with a particular color. The whole process is referred to as color quantization. Compared with RGB color histogram, HSV color histogram is more practical and can directly describe the color distribution. The conversion from RGB space to HSV space [7] can be given by

$$H = \begin{cases} 60^{\circ} \times (\frac{G' - B'}{\Delta C} \bmod 6), C_{\max} = R' \\ 60^{\circ} \times (\frac{B' - R'}{\Delta C} + 2), C_{\max} = G' \\ 60^{\circ} \times (\frac{R' - G'}{\Delta C} + 4), C_{\max} = B' \end{cases} \qquad H \in [0, 360^{\circ}) \tag{1}$$

where $\begin{cases} R' = R/255 \\ G' = G/255 \\ B' = B/255 \end{cases}$, $\begin{cases} C_{\max} = Max(R', G', B') \\ C_{\min} = Min(R', G', B') \\ \Delta C = C_{\max} - C_{\min} \end{cases}$

The Hue histograms of moving target at different time step are shown in Fig. 1(a)-Fig. 1(e).



(a) (b) (c)

(d) (e) (f)

Fig. 1. Hue histogram of the green car at different time steps.

Traditional color histogram does not consider the mapping relationship between the pixel color and pixel position, in other words, it can only represent the global color distribution. Therefore, the regions with different color distributions may have the similar color histogram. The color distribution in Fig. 2 and Fig. 3 are different, but their color histograms are the same, that is, only color histogram cannot differentiate two regions. Stanley proposed video tracking method based on spatial histogram [8], but the computational effort of the algorithm is large, especially the calculation of covariance matrices for the pixels at each bin of the histogram is time-consuming.

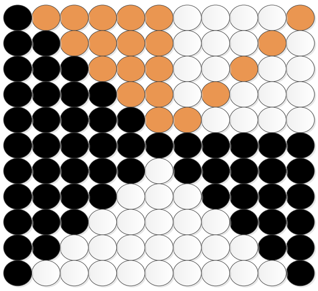


Fig. 2. Color distribution of target region.

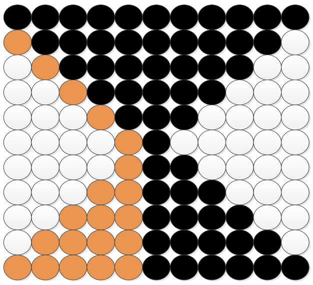


Fig. 3. Color distribution of candidate region.

In this paper, we propose an efficient and accurate color spatial distribution feature matching algorithm based on hue histogram and density centroid vector combined with pixel mean value vectors. The schematic of feature matching based on spatial color distribution histogram is shown in Fig. 4.

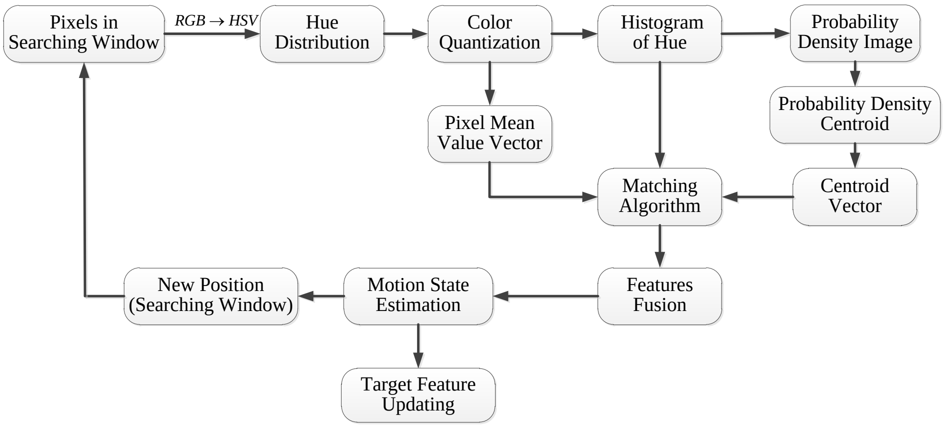


Fig. 4. Schematic of feature matching based on spatial color distribution histogram.

The color density image is also called back-project image which reapplies the modified histogram to the original image [9]. In other words, the pixels locate in the same bin are replaced by the relative histogram bin values which represent the probability that the pixels belong to the video target. In video tracking systems, the probability density centroid (mass center) can be calculated by the zero moment and first moment of the probability density image. Let $D(x, y)$ represents the probability density image, $h(x, y)$ represents the hue value of pixel (x, y), and Hist $(h(x, y)$ represents the histogram of hue. Suppose the width and height of the searching window is 'W' and 'H' respectively, then the zero moment can be given by

$$M_{00}=\sum_{x=0}^{W}\sum_{y=0}^{H}D(x,y)=\sum_{x=0}^{W}\sum_{y=0}^{H}Hist(h(x,y)) \tag{2}$$

The first moment

$$M_{01}=\sum_{x=0}^{W}\sum_{y=0}^{H}yD(x,y)=\sum_{x=0}^{W}\sum_{y=0}^{H}yHist(h(x,y)) \tag{3}$$

$$M_{10}=\sum_{x=0}^{W}\sum_{y=0}^{H}xD(x,y)=\sum_{x=0}^{W}\sum_{y=0}^{H}xHist(h(x,y)) \tag{4}$$

Suppose the probability density centroid coordinate of the searching window is $O(x_{DC}, y_{DC})$, the geometrical center coordinate is $O(x_{GC}, y_{GC})$, then

$$x_{DC} = M_{10} / M_{00}, y_{DC} = M_{01} / M_{00} \tag{5}$$

$$x_{GC} = W / 2, y_{GC} = H / 2 \tag{6}$$

The probability density centroid vector $\vec{P}_{GD}$ can be calculated by

$$\vec{P}_{GD} = \left\|\vec{P}_{GD}\right\| \angle \arccos((M_{10} / M_{00} - W / 2) / \left\|\vec{P}_{GD}\right\|) \tag{7}$$

$$\left\|\vec{P}_{GD}\right\| = \sqrt{(M_{10} / M_{00} - W / 2)^2 + (M_{01} / M_{00} - H / 2)^2} \tag{8}$$

Assume the hue histogram is divided into *N* equal bins, that is, the pixels in the searching window are divided into *N* groups.

Let $\vec{Q}(i)$ denote the pixel mean value vectors of the *N* group pixels, $B(i)$ represents the hue value of the ith bin, then

$$\vec{Q}(i) = \left\|\vec{Q}(i)\right\| \angle \arccos \frac{x - W / 2}{\left\|\vec{Q}(i)\right\|} \tag{9}$$

where $\overline{H}_k = \frac{1}{N}\sum_{i=1}^{N} H_k(i)$

The commonly used methods for histogram matching include Chi-squared distance, intersection distance and Bhattacharya distance [9]. Assume the target histogram and candidate histogram is $H_T(i)|_{i=1}^{N}$ and $H_C(i)|_{i=1}^{N}$ respectively, *N* is the number of histogram bins. The matching degree *M* (H_T, *HC*) can be calculated as follows:

Bhattacharyya distance:

$$M_{BD}(H_T, H_C) = \sqrt{1 - \frac{1}{\sqrt{\overline{H_T}\,\overline{H_C} N^2}} \sum_{i=1}^{N} \sqrt{H_T(i) H_C(i)}} \tag{10}$$

where $\overline{H}_k = \frac{1}{N}\sum_{i=1}^{N} H_k(i)$

Chi-Squared distance:

$$M_{CS}(H_T, H_C) = \sum_{i=1}^{N} \frac{(H_T(i) - H_C(i))^2}{H_T(i) + H_C(i)} \tag{11}$$

Intersection distance:

$$M_{ID}(H_T, H_C) = \sum_{i=1}^{N} \min(H_T(i), H_C(i)) \tag{12}$$

Intersection distance-based histogram matching is the most efficient method; Bhattacharyya distance matching has the best accuracy and the efficiency is close to that of intersection distance method. Therefore, the Bhattacharyya distance is a preferred method to measure the similarity between the target hue histogram and candidate hue histogram. The value of M_{BD} ranges from zero to one and the smaller the value, the better matching the two histograms. In other words, if M_{BD} equals to zero, the two histograms are completely matched.

As mentioned above, two well matched color histograms can hardly reach the conclusion that the target region and candidate region are similar. We need to further compare the pixel mean value vector and probability density centroid vector with the corresponding vectors derived from the target region at previous time step. The Bhattacharyya coefficient [10] is employed to measure the overlapping degree between two vectors. Let $\vec{V}_1 = (p_1, p_2, ..., p_n)^T$ and $\vec{V}_2 = (q_1, q_2, ..., q_n)^T$ represent two n-dimensional vectors respectively, the Bhattacharyya coefficient of these two vectors can be computed as following:

$$F = \cos\beta = \frac{\vec{V}_1^T \cdot \vec{V}_2}{\left\|\vec{V}_1\right\| \cdot \left\|\vec{V}_2\right\|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2 \cdot \sum_{i=1}^{n} q_i^2}} \tag{13}$$

Actually, β is the angle between vectors $\vec{V}_1$ and $\vec{V}_2$; the greater the Bhattacharyya coefficient, the closer the two vectors. Suppose the normalized probability density centroid vector and pixel mean value vectors at previous frame is $\vec{R}_{GD}$ and $\vec{W}(i)$ respectively. The Bhattacharyya coefficient between $\vec{P}_{GD}$ and $\vec{R}_{GD}$, and the coefficient between $\vec{Q}(i)$ and $\vec{W}(i)$ can be calculated as

$$F_{GD} = \frac{\vec{P}_{GD}^{\ T} \cdot \vec{R}_{GD}}{\left\|\vec{P}_{GD}\right\| \cdot \left\|\vec{R}_{GD}\right\|} \tag{14}$$

$$F_{PM}(i) = \frac{\vec{W}(i)^T \cdot \vec{Q}(i)}{\left\|\vec{W}(i)\right\| \cdot \left\|\vec{Q}(i)\right\|}, \quad 1 \le i \le N \tag{15}$$

The matching degree for color spatial distribution can be computed as

$$M_{\text{Spatial Color}} = \lambda_1(1 - M_{BD}) + \lambda_2 F_{GD} + \frac{\lambda_3}{N}\sum_{i=1}^{N} F_{PM}(i) \tag{16}$$

where $\lambda_1 = 0.2, \lambda_2 = \lambda_3 = 0.4$ are the weighted coefficients. The color feature of video target is vulnerable to the variation of illumination, so it should be updated over time. When the color spatial distribution histogram matching degree $M_{\text{Spatial Color}} > 0.9$, the target color spatial feature can be normalized and updated by

$$\begin{gathered} \vec{R}_{GD} = \frac{\vec{P}_{GD}}{\sqrt{\left\|\vec{P}_{GD}\right\|_2^2 + e^2}}, \vec{W}(i) = \frac{\vec{Q}(i)}{\sqrt{\left\|\vec{Q}(i)\right\|_2^2 + e^2}} \\ H_T(i)\,|_{i=1}^{N} = H_C(i)\,|_{i=1}^{N} \end{gathered} \tag{17}$$

where e is a small constant.

2.2. Contour Extraction and Matching

Geometrical features of video target mainly include target contour, minimum bounding rectangle, area and key points. Actually, video target contour is a curved line connecting the pixels whose gray value significantly change in a digital image. Target contour feature is quite important for video target detection and tracking. If the contour is obtained, the minimum bounding rectangle and centroid can be determined as well. Video tracking systems have high requirement in real-time performance; therefore, the contour extraction algorithm should guarantee both the accuracy and efficiency. M. Kass proposed energy-minimizing spline based active contour models which can lock onto video target nearby edges and localize them accurately [10]. However, the spline has to be guided by the external user-imposed constraint forces, in other words, the algorithm cannot locate the target contour automatically. Since the edge pixels of video target have the maximum orientation gradient, hence the edge detection can be viewed as the process of calculating the local maximum values and direction of orientation gradient. Traditional edge detection algorithms are mainly based on the first-order and second-order differential operators among which the Robert, Sobel and Prewitt operators are the first-order operators; the Laplacian, Wallis and Log operators are the second-order operators. To simplify the calculation, convolution operation is executed between the image and corresponding odd dimension template matrices which are used to replace complex differential operations. Literature [11] introduced the target contour extraction method based on Canny edge detection and adaptive background updating. However in some cases, the Gaussian smoothing filtering in Canny operator can result in the loss of slight edge. In addition, differential operators are sensitive to noises and easy to treat the noises as edge.

In this paper, we introduce an efficient and accurate contour extraction algorithm based on multi-scale wavelet transform combined with three consecutive frames difference and background dynamic updating. The schematic of target contour extraction is shown in Fig. 5.

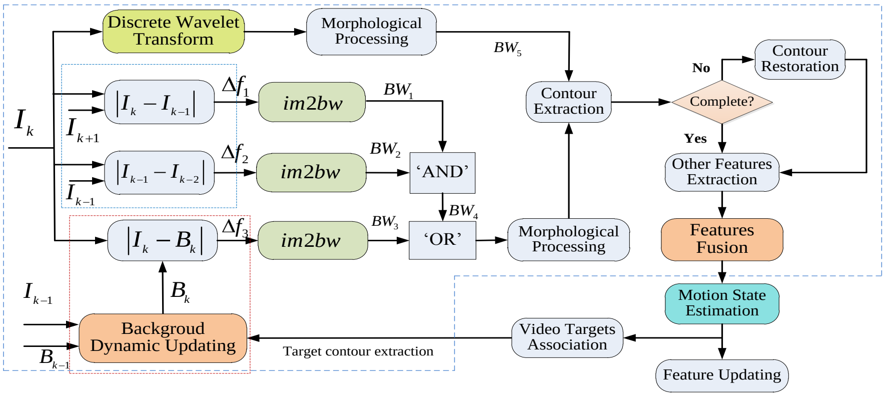


Fig. 5. Schematic of contour feature extraction for video target.

Wavelet transform has the intrinsic advantage in multi-scale analysis; it can decompose the signal into a series of wavelet functions which are derived from the scaled and shifted mother wavelet [12]. Actually, wavelet transform can be viewed as the linear combination of operators which represent the difference scales and extract edge information with different levels. Wavelet transform based edge detection has good performance in detection accuracy and noise immunity. Given a time domain signal $f(t) \in L^2(R)$, the wavelet transform of $f(t)$ can be given by

$$F(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) \psi^*(\frac{t-b}{a}) dt \tag{18}$$

where $\psi(t)$ is the wavelet basis (mother wavelet), $\psi^*(t)$ is the corresponding complex conjugate form, $\psi(t)$ holds the following conditions:

$$\begin{aligned} &\psi(t) \in L^2(R) \\ &\int_{-\infty}^{+\infty} |\psi(t)|^2 dt = 1 \\ &\int_{-\infty}^{+\infty} t|\psi(t)|^2 dt = 0 \end{aligned} \tag{19}$$

Parameters ‘*a*’ and ‘*b*’ represent the scaling factor and time translation respectively. To simplify computer programming, we employ discrete wavelet transform which is based on scaling factor ‘*a*’ and translation parameter ‘*b*’, instead of time variable. To simplify the calculation and boost the efficiency of wavelet transform, the scaling factor and time translation are both set as a multiple of $2^P\big|_{P=0}^{N}$, that is, the wavelet basis is shifted and scaled by powers of two. The discrete wavelet function is given by

$$W(p,q)=\frac{1}{\sqrt{2^p}}\psi(\frac{1}{2^p}-q) \tag{20}$$

The discrete wavelet transform coefficient

$$C(p,q)=\frac{1}{\sqrt{2^p}}\int_{-\infty}^{+\infty} f(t)\psi^*(\frac{1}{2^p}-q)dt \tag{21}$$

In the case of image edge detection, discrete wavelet transform is implemented by filter banks .The digital image is simultaneously decomposed into high-frequency and low-frequency coefficients by using high-pass and low-pass filters [13] which should be quadrature mirror filters. High-frequency parts appear sharp gray variation in small areas and represent the edge information. Conversely, low-frequency parts have gradual transition in gray value and represent the main frame of video targets. The core idea of multi-scale (multi-resolution) image edge detection is to use edge operators with different scales to detect the pixels which have the maximum modulus. Let $f(x,y)$ represent a digital image, the image can be decomposed via wavelet transform into four parts : horizontal low frequency and vertical low frequency $f_{LL}^p(x,y)$, horizontal low frequency and vertical high frequency $f_{LH}^p(x,y)$, horizontal high frequency and vertical low frequency $f_{HL}^p(x,y)$, horizontal high frequency and vertical high frequency $f_{HH}^p(x,y)$. The schematic of multi-scale decompose for digital image is shown in Fig. 6. To boost the decomposition efficiency and facilitate programming, we adopt fast wavelet multi-scale decompose method which is shown in Fig. 7.

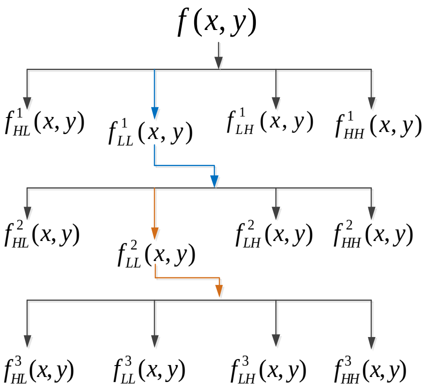


Fig. 6. Schematic of wavelet multi-scale decompose.

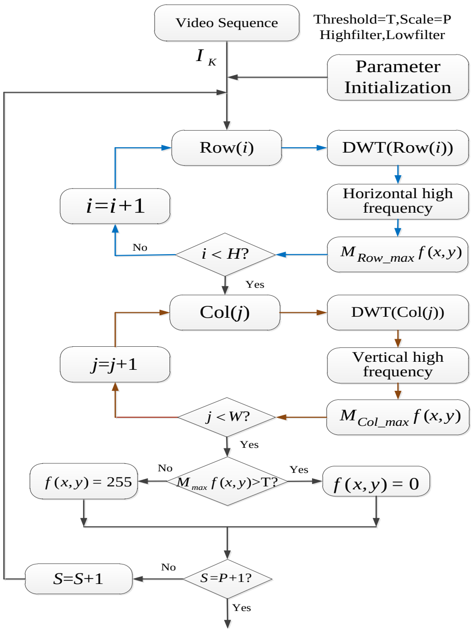
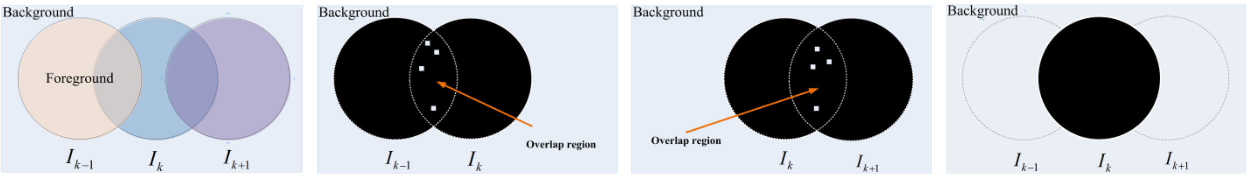


Fig. 7. Flow chart of fast wavelet multi-scale decompose.

where P is the predefined decomposition scale, T is the binary threshold, $M_{Row\_max}f(x, y)$ and $M_{Col\_max}f(x, y)$ is the maximum modulus in the horizontal and vertical directions respectively; W and H are the width and height of image. The time complexity of traditional wavelet transform is $O(N\log N)$, while in the fast wavelet transform, it can be reduced to $O(N)$. The determination of scaling and basis function for wavelet transform directly impacts the detection efficiency.

Consecutive frames difference method which calculates the difference value between two or more consecutive frames is a highly efficient video targets detection method. The commonly used consecutive frames difference method involves two consecutive frames difference and three consecutive frame difference. Two consecutive frames difference is equivalent to perform the 'OR' operation of two adjacent frames; it has a certain degree of robustness to the illumination variation. But meanwhile, two consecutive frames difference can expand the detection region, and produce holes in the overlapping regions due to the similar color distribution of two frames, as shown in Fig. 8(b)-Fig. 8(c).



(a) (b) (c) (d)

Fig. 8. Consecutive frames difference.

Three consecutive frames difference is insensitive to the illuminance variation and scene change and can avoid the expansion of detection region, but likewise, the detection result still contains holes which come from the two consecutive frames difference. With the help of image morphological processing, most holes can be removed. Suppose the three consecutive frames are $I_{k\text{-}2}$, $I_{k\text{-}1}$ and I_k, the two difference values are computed as

$$BW_1 = im2bw(|I_k - I_{k-1}|) \tag{22}$$

$$BW_2 = im2bw(|I_{k-1} - I_{k-2}|) \tag{23}$$

The final detection region BW_4 is the 'AND' operation of BW_1 and BW_2, that is,

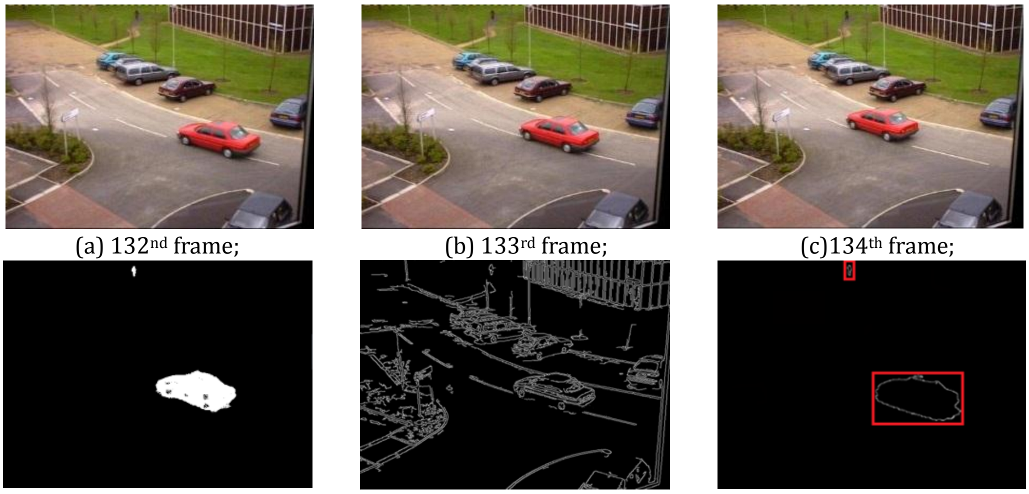
$$BW_4 = BW_1 \cap BW_2 \tag{24}$$

The notation *im*2*bm* (·) denotes the image binarization processing. The detection result is shown in Fig. 9(d) As a supplement, background difference method based on background dynamic updating can quickly detect and locate the video targets.

The background is susceptible to the ambient factors such as illumination variation and random noise; therefore it should be updated dynamically. Let B_k represent the background frame at time k; $P_{k-1}(i)|_{i=1}^{N}$ represent the minimum bounding rectangles of the detected targets at time k-1.The dynamically updated background can be given by

$$B_k = \alpha B_{k-1} + (1-\alpha)(I_{k\text{-}1} - \sum_{i=1}^{N} P_{k\text{-}1}[i]) \text{ , } B_0 = \frac{1}{20}\sum_{i=1}^{20} B_i \tag{25}$$

where α is the updating factor which ranges from 0.90 to 0.99; $I_{k\text{-}1}$ is the video frame at time k; the initial background frame can be obtained by calculating the average of the first twenty frames.



(a) 132nd frame; (b) 133rd frame; (c)134th frame;

(d) Three frames difference; (e) Wavelet transform edge detection; (f) Target contour;

Fig. 9. Contour extraction based on wavelet transform and three consecutive frames difference.

Let $f(x,y)$ represent the binary image which only contains video targets, W and H denote the width and height of the minimum bounding rectangle of target contour. The horizontal coordinate of target contour ranges from x_0 to x_0+W, the vertical coordinate ranges from y_0 to y_0+H. The centroid of the contour can be given by

$$x_c=\frac{\sum_{x=x_0}^{W+x_0}\sum_{y=y_0}^{H+y_0} xf(x,y)}{\sum_{x=x_0}^{W+x_0}\sum_{y=y_0}^{H+y_0} f(x,y)},\ y_c=\frac{\sum_{x=x_0}^{W+x_0}\sum_{y=y_0}^{H+y_0} yf(x,y)}{\sum_{x=x_0}^{W+x_0}\sum_{y=y_0}^{H+y_0} f(x,y)} \tag{26}$$

The centroid moment of target contour

$$C_m=\frac{\sum_{x=x_0}^{W+x_0}\sum_{y=y_0}^{H+y_0}(x-x_c)(y-y_c)f(x,y)}{\sum_{x=x_0}^{W+x_0}\sum_{y=y_0}^{H+y_0} f(x,y)} \tag{27}$$

Let $(X[i],Y[i])\Big|_{i=0}^{N-1}$ represent the coordinates of target contour which have been obtained through aforementioned algorithm, where N is the number of contour pixels. The normalized contour mean value vector can be computed as

$$\vec{V}_m=\frac{\begin{bmatrix}\sum_{i=0}^{N-1}X[i]-x_c & \sum_{i=0}^{N-1}Y[i]-y_c\end{bmatrix}^T}{\left(\sum_{i=0}^{N-1}X[i]-x_c\right)^2+\left(\sum_{i=0}^{N-1}Y[i]-y_c\right)^2} \tag{28}$$

Let $C_m(k-1)$ and $\vec{V}_m(k-1)$ be the centroid moment and normalized contour mean value vector at time k-1 respectively, then the target contour matching degree

$$M_{contour}=\beta_1\left(1-\frac{C_m(k-1)-C_m(k)}{C_m(k-1)}\right)+(1-\beta_1)\frac{\vec{V}_m{}^T\cdot\vec{V}_m}{\left\|\vec{V}_m\right\|\cdot\left\|\vec{V}_m\right\|} \tag{29}$$

2.3. Edge Gradient Feature Extraction

Mathematically, the derivative can be used to measure the change of function value with respect to the change of variable. Likewise, the image edge gradient can be used to represent the gray change of pixels. The gradient $G(x, y)$ for each pixel (x, y) in image $f(x, y)$ can be computed as

$$G(x,y)=\begin{bmatrix} f_x(x,y) \\ f_y(x,y) \end{bmatrix}=\begin{bmatrix} \partial f(x,y)/\partial \text{x} \\ \partial f(x,y)/\partial y \end{bmatrix} \tag{30}$$

Actually, the differential operation in image processing can be replaced by difference operation [14]. The aforementioned first-order derivative can be transformed by

$$\begin{aligned} f_x(x,y) &= f(x,y)-f(x-1,y) \\ f_y(x,y) &= f(x,y)-f(x,y-1) \end{aligned} \tag{31}$$

The norm of gradient at pixel(x, y)

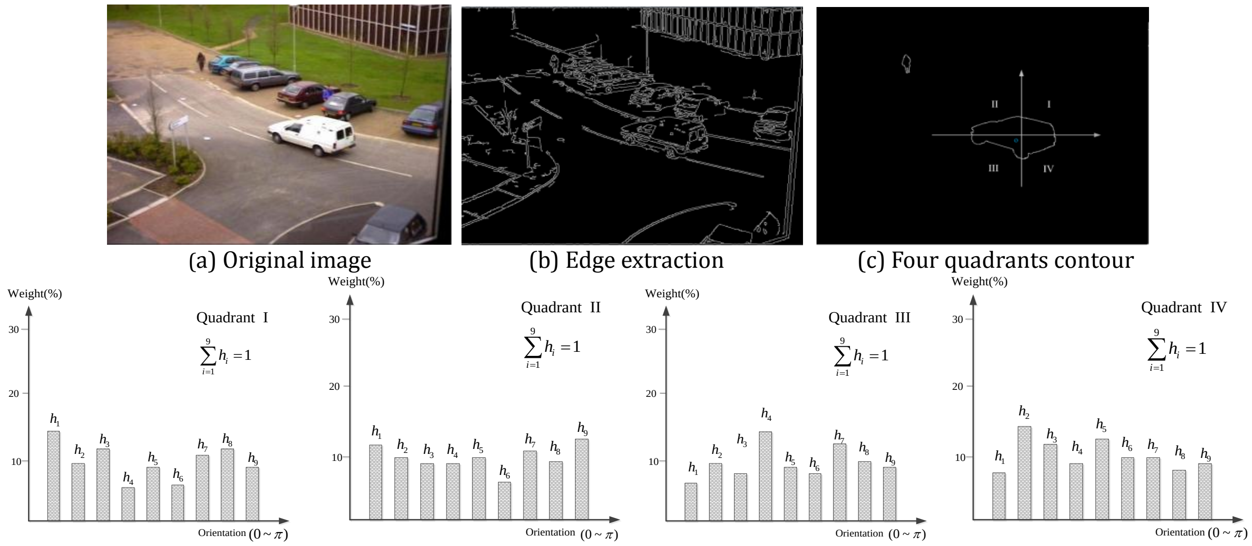
$$N(x,y)=\sqrt{(f(x,y)-f(x-1,y))^2+f(x,y)-f(x,y-1)^2} \tag{32}$$

Apparently, the edge pixels have larger gradient than neighboring pixels. In this paper, we employ divisional edge gradient histogram (EGH) to compare the candidate contour with target contour. The EGH can be defined as:

$$H_E=[h_1,h_2,h_3,\ldots,h_9] \tag{33}$$

The whole orientation is preferred to be divided into nine equal parts, so the angle for each part is 22.5 degrees, $h_i\,|_{i=1}^{9}$ represent the weight for each direction.

We have obtained the contour of candidate target via wavelet transform combined with multi-frame difference and background difference in part *B*. To boost the matching degree, the full contour is divided into four parts which separately locate in four quadrants as shown in Fig. 10(c); the EGH for each contour segment are calculated and shown in Fig. 10(d)-Fig. 10(g). The origin of coordinate is the centroid of video target.



(a) Original image (b) Edge extraction (c) Four quadrants contour

(d) (e) (f) (g)

Fig. 10. Divisional edge gradient histogram.

Let $H_{E\_\mathrm{I}}, H_{E\_\mathrm{II}}, H_{E\_\mathrm{III}}$ and H_{E_IV} represent the EGHs of the candidate contour in four quadrants. Suppose the corresponding reference EGHs are $R_{E\_\mathrm{I}}, R_{E\_\mathrm{II}}, R_{E\_\mathrm{III}}$ and R_{E_IV}, the Bhattacharyya distance between H_{E_I} and R_{E_I} can be calculated by

$$B_{ER\_\mathrm{I}} = \sqrt{1-\sum_{i=1}^{9}\sqrt{H_{E\_\mathrm{I}}(i)H_{R\_\mathrm{I}}(i)} \Big/ 9\sqrt{\overline{H}_{E\_\mathrm{I}}\cdot\overline{R}_{E\_\mathrm{I}}}} \qquad (34)$$

where $\overline{H} = \frac{1}{9}\sum_{i=1}^{9}H(i), \quad H(i) = h_i$

Similarly, we can calculate $B_{ER\_\mathrm{II}}, B_{ER\_\mathrm{III}}$ and B_{ER_IV}. The target contour matching degree based on EGH can be given by

$$M_{EGH} = (4-(B_{ER\_\mathrm{I}} + B_{ER\_\mathrm{II}} + B_{ER\_\mathrm{III}} + B_{ER\_\mathrm{IV}}))/4 \qquad (35)$$

3. Multi-feature Fusion and Algorithm Optimization

The color spatial feature, contour feature and edge gradient histogram should be extracted simultaneously in a certain local region which is predicted by motion state estimator. As well as, the matching degree of the three features are computed. We adopt linear weighted fusion method [15]-[17] to calculate the final matching degree which is given by the following equation:

$$M_{mf} = \alpha_1 M_{SpatialColor} + \alpha_2 M_{Contour} + \alpha_3 M_{EGH} \qquad (36)$$

where $\alpha_1, \alpha_2, \alpha_3$ are the corresponding weighted coefficients which in general are equal each other. However, when the illumination changes frequently, α_1 should be assigned smaller value than other two coefficients. Likewise, if the moving target is non-rigid object, α_2 and α_3 should be smaller than α_1.

The video tracking systems have high requirement in real-time performance. However, the extraction and fusion for multiple features would definitely cost a large amount of computational effort. Therefore, parallel computation would be particularly important. In this paper, an efficient thread-pool model is introduced to simultaneously extract multiple features. On a single processor platform, multi-thread is performed by time-division multiplexing, while for a truly multi-core system, multi-thread can be concurrently implemented. In this thread-pool model, we create twelve threads which are responsible for respective functions.

'*Thread_TFD'* is to perform three consecutive frames difference and detect video targets; '*Thread_BD'* is in charge of background difference; '*Thread_WT'* takes charge of discrete wavelet transform; '*Thread_Contour*' is the thread in charge of integrating the result from aforementioned three threads and calculating the final target contour. '*Thread_PDCV'* is responsible for calculating the probability density centroid vector of image; '*Thread_EGH'* takes charge of the extraction of EGH feature. '*Thread_PMV'* is for the calculation of pixels mean value vectors; '*Thread_NCH'* is used to calculate the normalized color histogram and its matching degree. '*Thread_Fusion'* is used to fuse different video features and calculate target matching degree and then send the result to the main thread. The schematic of thread-pool model for multi-feature fusion is shown in Fig. 11.

Multi-thread synchronization technology plays an important role in guaranteeing the efficient operation of thread-pool. On the windows platform, we can employ event notification and synchronization functions, as shown in Fig. 12, to realize multi-thread synchronization.

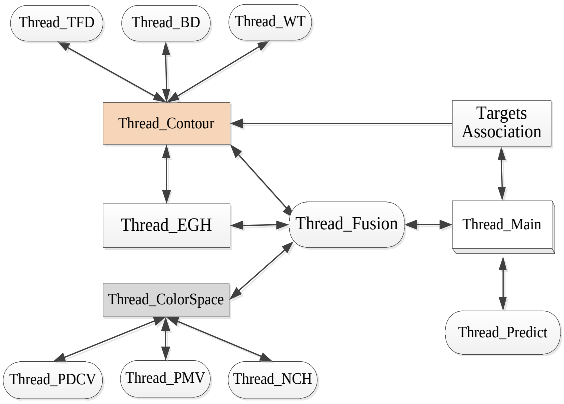
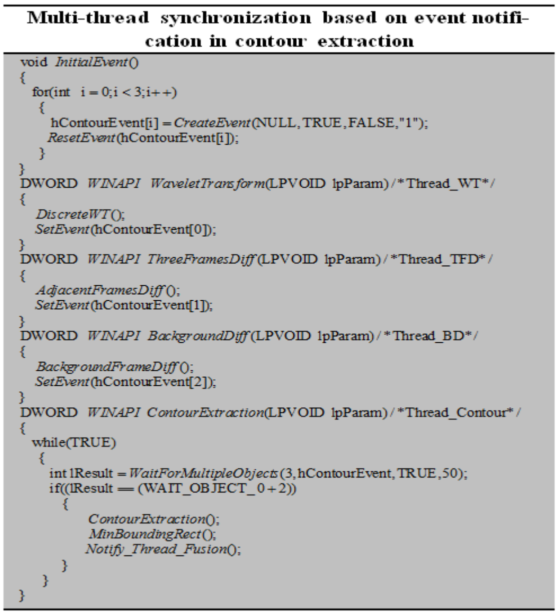


Fig. 11. Multi-feature fusion and algorithm.



```
Multi-thread synchronization based on event notifi-
cation in contour extraction
void InitialEvent()
{
  for(int  i = 0;i < 3;i++)
   {
     hContourEvent[i] = CreateEvent(NULL,TRUE,FALSE,"1");
    ResetEvent(hContourEvent[i]);
   }
}
DWORD  WINAPI  WaveletTransform(LPVOID lpParam) /*Thread_WT*/
{
   DiscreteWT();
   SetEvent(hContourEvent[0]);
}
DWORD  WINAPI ThreeFramesDiff(LPVOID lpParam) /*Thread_TFD*/
{
   AdjacentFramesDiff();
   SetEvent(hContourEvent[1]);
}
DWORD  WINAPI BackgroundDiff(LPVOID lpParam) /*Thread_BD*/
{
   BackgroundFrameDiff();
   SetEvent(hContourEvent[2]);
}
DWORD  WINAPI ContourExtraction(LPVOID lpParam) /*Thread_Contour*/
{
  while(TRUE)
   {
     int lResult = WaitForMultipleObjects(3,hContourEvent,TRUE,50);
     if((lResult == (WAIT_OBJECT_0 + 2))
       {
           ContourExtraction();
           MinBoundingRect();
          Notify_Thread_Fusion();
       }
    }
}
```

Fig. 12. Multi-thread synchronization.

4. Motion State Estimation Based on *H*∞ Filtering

Multi-feature extraction methods based on global image searching tend to be time-consuming and impractical. Actually, there is no need to search the entire image to extract target features. Video targets such as vehicles and pedestrians move according to a certain motion law. We can utilize an efficient filter to estimate and predict the motion state of video targets [17]. The velocity and location are the two motion states of video targets. The top left corner of the image is the origin of coordinate in Fig. 13.

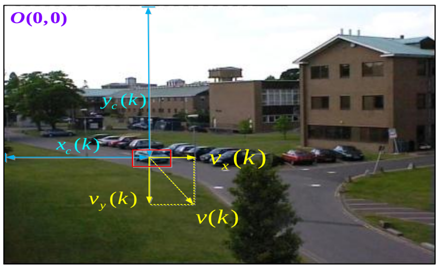


Fig. 13. Motion state representation of moving targets.

Suppose $v_x(k)$ and $v_y(k)$ are respectively the horizontal speed and vertical speed of the moving target at time step k; $x_c(k)$ and $y_c(k)$ represent the target centroid coordinate. The dynamic characteristics of video target can be given by

$$\begin{cases} v_x(k) = v_x(k-1) + v_x(k-1)T \\ v_y(k) = v_y(k-1) + v_y(k-1)T \\ x_c(k) = x_c(k-1) + v_x(k-1)T + T^2 v_x(k-1)/2 \\ y_c(k) = y_c(k-1) + v_y(k-1)T + T^2 v_y(k-1)/2 \end{cases} \tag{37}$$

where T is the sampling time interval. The system state can be represented as:

$$\begin{gathered} X_k = \begin{bmatrix} x_c(k) & y_c(k) & v_x(k) & v_y(k) \end{bmatrix}^T \\ v_x(k) = \frac{x_c(k) - x_c(k-1)}{T}, v_y(k) = \frac{y_c(k) - y_c(k-1)}{T} \end{gathered} \tag{38}$$

The motion state model and observation model of video target can be built as follows:

$$\begin{aligned} X_k &= \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 1 & 0 & 0 & T \\ 0 & 1 & 0 & 0 \end{bmatrix} X_{k-1} + \begin{bmatrix} \frac{T^2}{2} \\ \frac{T^2}{2} \\ T \\ T \end{bmatrix} u_{k-1} \\ Y_k &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} X_k \end{aligned} \tag{39}$$

In practical applications, the motion state of moving target is easy to be affected by some uncertain factors such as movement resistance and ambient variation [18]. As well as, the measurements of motion state are impacted by unpredictable noises. Therefore, we need to modify the system model and observation model as follows:

$$\begin{aligned} X_k &= \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 1 & 0 & 0 & T \\ 0 & 1 & 0 & 0 \end{bmatrix} X_{k-1} + \begin{bmatrix} \frac{T^2}{2} \\ \frac{T^2}{2} \\ T \\ T \end{bmatrix} u_{k-1} + w_{k-1} \\ Y_k &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} X_k + v_k \end{aligned} \tag{40}$$

The Kalman filter, known as the optimal linear quadratic estimator, is widely used in tracking systems. It utilizes present measurement and previous updated estimate, combining with recursive equations to predict and update the state at the current time step. Therefore, the Kalman filter is one kind of highly efficient recursive filter which can estimate the state of linear dynamic system from a series of incomplete measurements and noise. Usually, the probability density function (PDF) of noise in Kalman filtering is assumed to be known. More specifically, the Kalman filter requires prior knowledge of the process and measurement noises, so the optimal state estimator based on the given PDF of noise can be designed. However in the real world, it is difficult to obtain the noise distribution model. Moreover, the Kalman filter only minimizes the expected value of the variance of estimation error [19].

In the process of H_∞ filtering, the noises are assumed to be the worst case and the filter desires to minimize the worst-case estimation error [20]. So compared with the Kalman filter, the H_∞ filter has better performance in robustness and accuracy when having no knowledge about system model and disturbance model. In addition, the H_∞ filter has the similar recursive equations with the Kalman filter, which makes it highly efficient in practical applications. Hence the H_∞ filter can also be seen as the robust version of the Kalman filter [21], [22].

To solve the H_∞ filtering problem, G. Zames proposed frequency domain method [23], U. Shaked presented transfer function method [24], Z. D. Wang introduced an algebraic matrix inequality [25] method to solve the H_∞ problem and designed a variance constrained state estimator for linear discrete time-invariant systems with parametric uncertainties in both the system model and measurement model. Y. S. Hung proposed a unilaterally coupled Riccati difference equations approach to analyze the performance of robust H_∞ filtering over a finite horizon discrete time variant system with unmeasured but admissible perturbations, and derived the sufficient conditions satisfying state estimation error variance constraints for the state estimator [26], R. Banavar discussed a game theory approach to H_∞ filtering [27]. Consider the discrete-time dynamic system shown as follows:

$$\begin{aligned} X_{k+1} &= A_k X_k + w_k \\ Y_k &= C_k X_k + v_k \qquad , \ k \in (0, N-1] \\ Z_k &= L_k X_k \end{aligned} \tag{41}$$

where w_k and v_k are the noises which could be random or deterministic, but not must be zero mean. $X_k \in R^n$ is the state vector; $Y_k \in R^p$ is the measured output vector; $w_k \in R^q$. Matrices *A_k*, *C_k* have corresponding dimensions. The system matrix A_k is non-singular, which means the dynamic system should be controllable. Likewise, the eigenvalues of measurement matrix C_k are none-zero, that is, the system is observable. Z_k is the output to be estimated, it represents a linear combination of system state X_k via user-defined full rank matrix L_k. Suppose the estimate of Z_k is $\hat{Z}_k$, estimation error $\tilde{Z}_k = Z_k - \hat{Z}_k$, the initial value of state estimation is $\hat{X}_0$, the initial estimation error $\tilde{X}_0 = X_0 - \hat{X}_0$. The H_∞ filter aims to compute $\hat{Z}_k$ based on the measurements up to and including previous time step, and minimize the estimation error $\tilde{Z}_k$. While the process and measurement noises tend to maximize the estimation error $\tilde{Z}_k$, that is ,both sides desire to realize profit maximization. The cost function in game theory [28] is suitable for equilibrate the estimation error and natural noises, it is designed as follows:

$$J = \frac{\sum_{k=0}^{N-1} \left\| \tilde{Z}_k \right\|_{M_k}^2}{\left\| \tilde{X}_0 \right\|_{P_0^{-1}}^2 + \sum_{k=0}^{N-1} (\left\| w_k \right\|_{Q_k^{-1}}^2 + \left\| v_k \right\|_{R_k^{-1}}^2)} \tag{42}$$

where M_k, P_0, Q_k, R_k are symmetric positive definite weighting matrices and generally user-specified based on real world applications. P_0 denotes the role of the initial estimation error $\tilde{X}_0$, Q_k and R_k respectively represent the influence of process noise and measurement noise to the filter. M_k represents the different treatment on the elements in Z_k; N is the current time step. The weighted 2-norm of an $n \times 1$ vector is computed as

$$\left\| X \right\|_P^2 = X^T P X, \left\| X \right\|_P = \sqrt{X^T P X} \tag{43}$$

The designer of the H_∞ filter and the disturbances are the two game players. The H_∞ filter desires to obtain the minimum estimation error $\tilde{Z}_k$, however, the disturbances such as w_k and v_k would definitely maximize the estimation error. Actually, it is difficult to directly calculate the minimum value of the cost function *J*. In the H_∞ filter, the cost function has a user-defined performance bound which makes the H_∞ filter more robust than the Kalman filter. In the extreme case, when the performance bound goes to infinite, the H_∞ filter turns into the Kalman filter. Suppose the performance bound is $1/\lambda$, then

$$J_1 = \lambda \sum_{k=0}^{N-1} \left\| \tilde{Z}_k \right\|_{M_k}^2 - \left\| \tilde{x}_0 \right\|_{P_0^{-1}}^2 - \sum_{k=0}^{N-1} (\left\| w_k \right\|_{Q_k^{-1}}^2 + \left\| v_k \right\|_{R_k^{-1}}^2) < 0 \tag{44}$$

According to the dynamic system equation (41), we have

$$\tilde{Z}_k = L_k (X_k - \hat{X}_k) \tag{45}$$

For a linear discrete-time system, the measurement noise can be denoted by measurements and state value:

$$v_k = Y_k - C_k X_k \tag{46}$$

The H_∞ filter is to minimize the cost function J_1 via seeking the optimal state $\hat{X}_k$ when the noises w_k and v_k are in the worst case. So the H_∞ filtering problem can be converted to solving minimax problem [29], that is

$$J_2 = \min_{\hat{X}_k} \max_{X_0, w_k, Y_k} (J_1) \tag{47}$$

The minimax problem can be interpreted as two steps: calculate the maximum value of *J* with respect to

X_0, w_k, Y_k; calculate the minimum value of J with respect to $\hat{X}_k$. The inequality (44) can be transformed into

$$J_1 = -\left\|\tilde{X}_0\right\|^2_{P_0^{-1}} + \sum_{k=0}^{N-1}(\lambda\left\|\tilde{X}_k\right\|^2_{L_k^T M_k L_k} - \left\|w_k\right\|^2_{Q_k^{-1}} + \left\|v_k\right\|^2_{R_k^{-1}}) < 0 \tag{48}$$

The updated covariance of estimate error can be computed by the following discrete-time Riccati equation [30]:

$$P_{k+1} = A_k P_k [I - \alpha L_k^T M_k L_k P_k + C_k^T R_k^{-1} C_k P_k]^{-1} A_K^T + Q_k \tag{49}$$

The updated state estimate

$$\hat{X}_{k+1} = (A_k - G_k C_k)\hat{X}_k + G_k Y_k \tag{50}$$

The H_∞ filter gain is given by

$$G_k = P_k [I - \alpha L_k^T M_k L_k P_k + C_k^T R_k^{-1} C_k P_k]^{-1} C_k^T R_k^{-1} \tag{51}$$

In this paper, color spatial distribution feature, contour feature and edge gradient feature are extracted, and the corresponding target matching degrees are calculated. We utilize weighted linear fusion method to obtain the final matching degree between the candidate region and target region, if the matching degree is more than user-specified threshold, that means the video target is found; the features will be updated. Otherwise, the new searching and matching in the same frame will be performed. The location measurements are chosen as the input of the H_∞ filter which has been initialized the parameters such as initial state X_0, covariance P_0 and the weighting matrices M_k,Q_k,R_k .The updated motion state calculated by the H_∞ filter is fed back to the features extraction to avoid global searching, and speed up the searching and matching. If the moving target is partially occluded, the matching degree based on multi-feature fusion would definitely reduce. We can calculate the threshold range when partial occlusion occurs. If the moving target is totally occluded, the matching degree will be a small value goes to zero. The schematic of video tracking based on multi-feature fusion and H_∞ filtering is shown in Fig. 14.

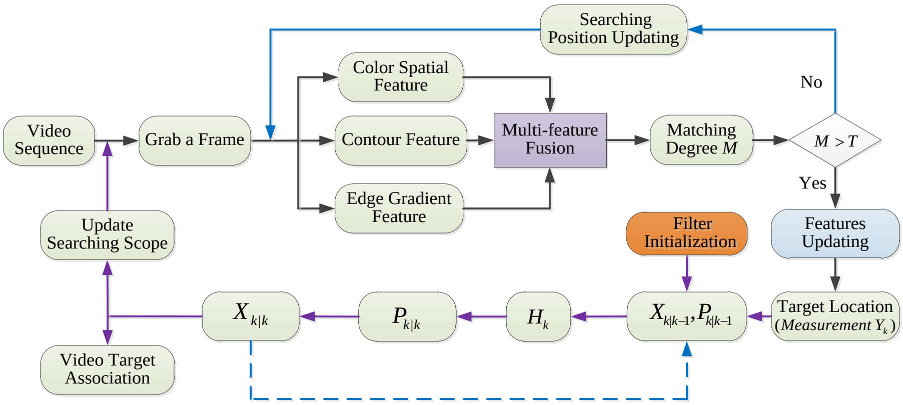


Fig. 14. Schematic of video tracking based on multi-feature fusions and H∞ filter.

5. Experiment

To evaluate the video tracking method based on multi-feature fusion and H_∞ filtering, we mainly focus on the tracking accuracy and robustness. A standard test video sequence 'PEST2001' with resolution of 768×576. We can calculate the matching degrees as shown in Fig. 15.

The blue car is totally blocked by the tree from the 380th frame to 400th frame; in the meantime, the measured data are missing. Since we are interested in the position and velocity of video target, the linear combination L_k is assigned as identity matrix. The weighting matrices of the H_∞ filter can be initialized as follows:

$$P_0 = \begin{bmatrix} 12 & 0 & 0 & 2 \\ 0 & 5 & 1 & 0 \\ 0 & 1 & 5 & 0 \\ 2 & 0 & 0 & 1 \end{bmatrix}, M_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_k = \begin{bmatrix} 0.3 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}$$

The initial motion state of the blue car

$$X_0 = \begin{bmatrix} 0 & 288 & 10 & 10 \end{bmatrix}^T$$

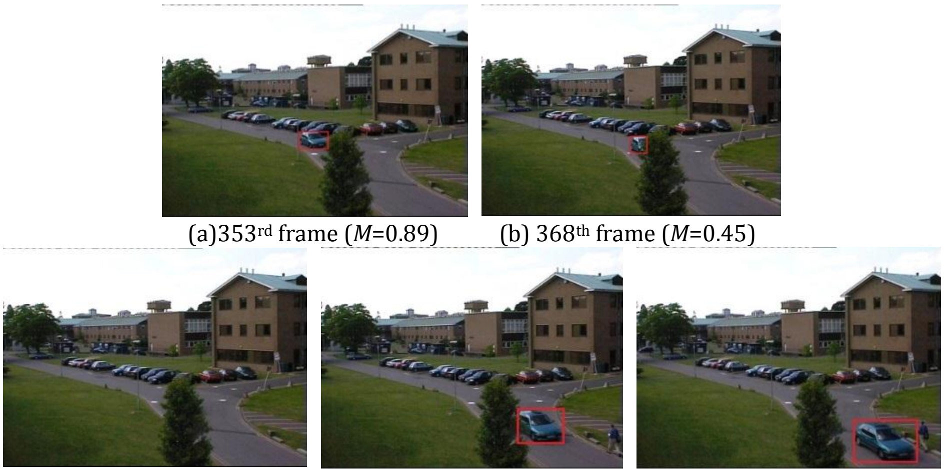
The performance bound of the H_∞ filter

$$1/\lambda = \frac{1}{2} \Rightarrow \lambda = 2$$

By contrast, the Kalman filter is employed to track the same video target. The process noise w_k and measurement noise v_k are assumed to be uncorrelated at each time step. Let Q_k and R_k respectively denote the covariances of w_k and v_k, then

$$w_k \sim N(0, Q_k),\ v_k \sim N(0, R_k)$$

$$Q_k = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}, R_k = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$



(a)353rd frame (M=0.89) (b) 368th frame (M=0.45)

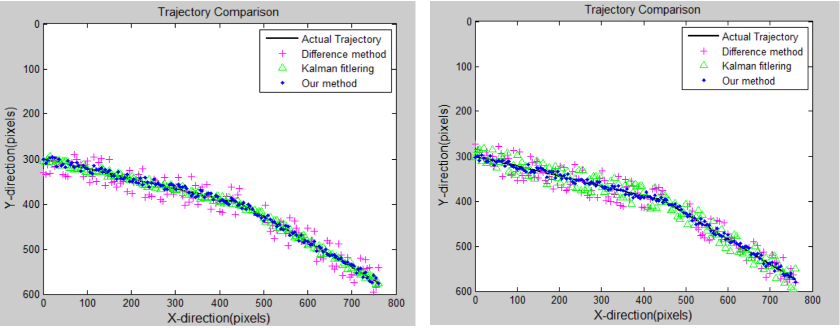
(C) 381st frame (M=0) (d) 420th frame (M=0.75) (e) 420th frame (M=0.92)

Fig. 15. Matching degree calculated by multi-feature fusion.

If there is no other disturbance when the video target moves, the actual target trajectory can be calculated by Kalman filter based method; difference value method and our method are shown in Fig.16(a). When the moving target is occluded from the 380th frame to 420th frame, the difference method cannot track the video target, however the H_∞ filter and Kalman filter can still estimate the motion states, and the H_∞ filter based method proposed in this paper has a similar performance in accuracy with the Kalman filter, and both have good fitting degree with the actual trajectory. However, the process noise and measurement noise sometimes cannot always be Gaussian. When the impulse noise was introduced into the video sequence from the 200th frame to 250th frame, the blue scatterplot in Fig.16(b) has better fitting degree than the Kalman filter.

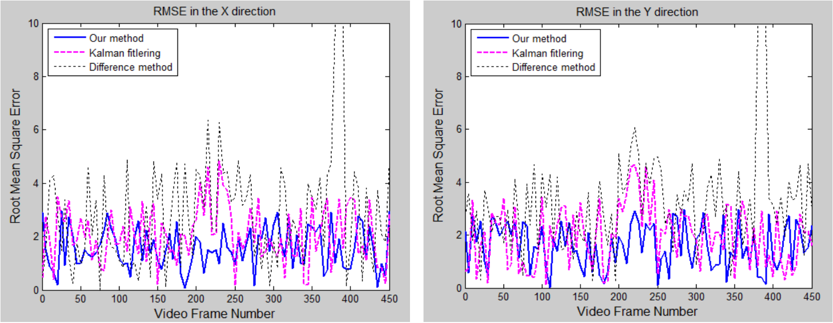
It can be seen from Fig. 17, the blue curve has the smaller RMSE in both the X and Y directions, especially the frame range that the non-Gaussian noises may affect. The experiment shows that when the measurements are missing in short time, the H_∞ filter can still track the video target. When the video target is occluded, difference method cannot detect and track the target. When the non-Gaussian noises are imposed to the tracking, the Kalman filter cannot realize optimal motion state estimation, however, the H_∞

filter can still work well.



(a) No impulse noise (b) Impulse noise impact

Fig. 16. Trajectories comparison between different methods.



(a)RMSE in the *x* direction (b) RMSE in the *y* direction

Fig. 17. Root mean square error (RMSE) comparison.

6. Conclusion

This paper includes two main contributions. Firstly, we proposed a multi-feature fusion method to describe video target. Color spatial distribution feature includes not only the color statistics but also the color spatial position information, which can provide better accuracy in color feature matching. Wavelet transform combined with three consecutive frame difference and dynamic background updating is adapted to accurately and efficiently extract target contour and centroid information. Moreover, the normalized contour mean value vector and edge gradient histogram can further boost the accuracy of matching degree. Single feature cannot be viewed as the discriminative feature for video tracking, multi-feature fusion and matching have a better immunity to the variation of ambient. We fuse these three features and optimize the fusion algorithm via thread pool and multi-thread synchronization. Secondly, we introduce H_∞ filtering to estimate and predict the motion state for video target. The H_∞ filter does not have to know the prior knowledge of the noise. When the process and measurement noises are non-Gaussian distributed, the Kalman filter cannot be an optimal linear filter; however the H_∞ filter can still estimate the motion state in the case of worst noise. In addition, the H_∞ filter updates the state by using efficient recursive equations which are similar to those of the Kalman filter, therefore, the H_∞ filter based video tracking can guarantee both robustness and efficiency.

Acknowledgment

The authors would like to appreciate Liang Huang and Kevin Wang.

References

[1] Jalal, A. S. (2012). The state-of-the-art in visual object tracking. *Informatics*, *36*, 227-248.

[2] Rong, H. & Li, J. H. (2009). *The Research of Moving Objects Tracking Algorithm Based on Video sequence*. Master Thesis, Xi' an Industry University.

[3] Asgarizadeh, M., & Pourghassem, H. (2015). A robust object tracking synthetic structure using regional mutual information and edge correlation based tracking algorithm in aerial surveillance application. *Signal, Image and Video Processing*, *36(8)*, 175-189.

[4] Optical_flow. Retrieved September 8, 2014, from http://en.wikipedia.org/wiki/Optical_flow

[5] Barron, D. J. L., Fleet, J. & Beauchemin, S. (1994). Performance of optical flow. *International Journal of Computer Vision Techniques*, *12(1)*, 43-77.

[6] Vanam, R. (2012). *Motion Estimation and Intra Frame Prediction in H.264/AVC Encoder*. Master Thesis. University of Washington.

[7] HSL_and_HSV. Retrieved July 20, 2014, from http://en.wikipedia.org/wiki/HSL_and_HSV

[8] Birchfield, S. T., & Rangarajan, S. (2005). Spatial histograms for region-based tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 1158-1163).

[9] OpenCV Organization. (2014). Open source computer vision library reference manual.

[10] Kass, M., *et al*. (1988). Snake: Active contour models. *International Journal of Computer Vision*, *1(4)*, 321-331.

[11] Wang, H., & Nguang, S. K. (2014). Video target tracking based on fusion state estimation. *Proceedings of the First IEEE International Symposium on Technology Management and Emerging Technologies: Vol. 1* (pp. 337-343).

[12] Mallat, S., & Zhong, S. (1992). Characterization of Signals from multi-scale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14(7)*, 710-732.

[13] Zhang, L., & Bao, P. (2002). Edge detection by scale multiplication in wavelet domain. *Pattern Recognition Letters*, *23*, 1771-1784.

[14] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*: *Vol. 1* (pp. 886-893).

[15] Zhou, Z. Y., *et al*. (2011). Object tracking based on multi-feature fusion and motion prediction. *Journal of Computational Information Systems*, *7(16)*, 5940-5947.

[16] Peng, Z. M., *et al.* (2004). Image matching based on multi-feature fusion. *High Power Laser and Particle* Beams, *16(3)*, 235-240.

[17] Wang, H., & Nguang, S. K. (2013). Intelligent and comprehensive monitoring system for swimming pool. *International Journal of Sensors Wireless Communications and Control*, *3(2)*, 85-94.

[18] Li, X., & Zell, A. (2007). H_∞ filtering for a mobile robot tracking a free rolling ball. *Robocop 2006*: *Robot Soccer World Cup X, Lecture Notes in Computer Science*, *44(3)*, 296-303.

[19] Simon, D. (2006). *Optimal State Estimation Kalman, H_∞ and Nonlinear Approaches* (pp. 333-371). A John Wiley & Sons, Inc., Publication.

[20] Ling, J. G. (2005). Approach of infrared small target motion prediction and tracking based on H_∞ filter. *Journal of Infrared Millimeter Waves*, *24(5)*, 366-369.

[21] Wang, X. & Liu, M. Q. (2011). Comparison of Kalman filter, H_∞ filter and robust mixed Kalman/H_∞ filter. *Proceedings of the 30th Chinese Control Conference*, *25(10)*, 3277-3281.

[22] Hao, Y. L., & Chen, M. H., *et al.* (2008). Comparison of robust H_∞ filter and Kalman filter for initial alignment of inertial navigation system. *Journal of Marine Science and Application*, *7(2)*, 116-121.

[23] Zames, G., & Francis, B. A. (1983). Feedback, minimax sensitivity, and optimal robustness. *IEEE Trans. on Automatic Control*, *28(5)*, 585-601.

[24] Yaesh, I., & Shaked, U. (1991). A transfer function approach to the problems of discrete-time systems H_∞ optimal linear control and filtering. *IEEE Trans. Automatic Control*, *36(11)*, 1264-1271.

[25] Wang, Z. D. (1997). Robust state estimation for discrete-time systems with error variance constraints. *IEEE Trans. on Automatic Control*, *42(10)*.

[26] Hung, Y. S., & Yang, F. W. (2003). Robust H_∞ filtering with error variance constraints for discrete time-varying systems with uncertainty. *Automatica*, *39(7)*, 1185-1194.

[27] Banavar, R. (1992). *A Game Theoretic Approach to Linear Dynamic Estimation*. PhD dissertation, University of Texas at Austin.

[28] Shen, X. M., & Deng L. (1997). Game theory approach to discrete filter design. *IEEE Transaction on Signal Processing*, *45(4)*, 1092-1095.

[29] Wang, X. B., *et al*. (2011). Target tracking based on the extended H_∞ filter in wireless sensor networks. *Journal of Control Theory and Application*, *9(4)*, 479–486.

[30] Rawicz, P. L. (2000). *H_∞/H_2/Kalman Filtering of Linear Dynamic Systems via Variational Techniques with Applications to Target Tracking*. PhD dissertation, Drexel University.

Howard Wang received his bachelor degree and master degree (with first class honors) from the Department of Automation of Zhejiang University, China, in 2001 and 2004 respectively. He is the PhD candidate in University of Auckland, New Zealand. His interests include but not limited to image intelligent analysis and processing, large scale video surveillance systems, fusion state estimate.

Sing Kiong Nguang received his BE (with first class honors) and PhD degrees from the Department of Electrical and Computer Engineering of University of Newcastle, Australia, in 1992 and 1995, respectively. He is the professor of the Department of Electrical and Computer Engineering, University of Auckland, New Zealand. He has published over 250 refereed journal and conference papers on nonlinear control design, nonlinear control systems, nonlinear time-delay systems, nonlinear sampled-data systems, fuzzy modeling and control. He is the senior member of IEEE.