International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.8 (2015), pp.83-96
http://dx.doi.org/10.14257/ijmue.2015.10.8.09

Video Background Subtraction Algorithm for a Moving Camera

Jinjiang Li, Jie Guo and Hui Fan

Shandong Institute of Business and Technology, Yantai, China
Shandong Normal University, Jinan, China
Shandong Institute of Business and Technology, Yantai, China
lijinjiang@gmail.com

Abstract

At present, the video background extraction algorithm of static scene has been nearly mature. However, video background extraction in dynamic scenes remains a challenge. In order to solve this problem, this paper proposes a dynamic scenes video background extraction algorithm. Here, our dynamic scene is based on camera movement. Firstly, we detect saliency target of video frame according to context information and do a processing of fuzzy enhancement. Meanwhile, we analyze flow filed by SIFT Flow method to do a nonlinear fusion with fuzzy enhancement result. Up to now, we can obtain moving target .Because of other gray information in the foreground affect target extraction, so we have to do a process of binarization and find out bounding box. After these preparations, we will track moving object with real-time algorithm. Finally, we use KNN algorithm to get accurate moving targets. Experiment results show that the proposed method for dynamic scenes video background extraction could get better results.

Keywords: *Dynamic scenes, Video background subtraction, Foreground objects*

1. Introduction

Real-time detection of moving target has a broad application prospects in many areas which include virtual reality, intelligent monitoring, video compression, automatic navigation and human-computer interaction. It is an important research component of computer vision and a basic portion of intelligent video systems (*e.g*. Video surveillance, automatic traffic monitoring) that extract moving targets correctly from video stream. At present, moving target detection technology under static scene has almost matured; researchers have proposed a variety of target detection technology. The common used methods of moving target detection include optical flow method, fame difference method and background subtraction method. The extraction effects of these methods in moving target extraction under static scenes are very better.

Moving target detection under dynamic scenes (*e.g*. Camera motion) is that using background detection template to detect and extract moving background in complex scenes to complete extraction of moving target. It is complex under dynamic scenes to extract moving target. Not only because of background exposure and masking, but also the global change of background due to the camera motion, there exist parts of entering into view filed and leaving view filed. Using differential cannot detect moving target accurately. Lipton *etc*. [1] combine time-domain differential with template matching, when the target move, we use time-domain differential method, else use template matching to detect target. Time-domain differential has good flexibility for dynamic scenes, and can detect the moving objects easily. However, when the camera is shifted, time domain (FDTD) method will fail, and cannot well extract all the pixels of moving objects, also form cavities inside the moving object. In addition, these methods have poor adaptability for interference factors in scenes. Brown [2] also summarizes some application of image registration; most important application is perspective registration

ISSN: 1975-0080 IJMUE

and time series registration. After image registration, we should start motion detection. This paper summarizes variety of researchers' achievement and proposed a dynamic scenes video background Subtraction algorithm.

2. Related Work

Moving target detection in video image is the separation of foreground and background, and then extract moving target. There are many methods of moving target detection, background subtraction is the most commonly method of motion target detection. Background subtraction is image subtraction between current frame and background image which was stored previously, if a pixel difference is bigger than threshold, then we can take it as moving target pixel. It is not difficult to realize background subtraction method, complexity of procedure is relatively low, but its requirements to background modeling are high. When illumination changed suddenly, background pixel would be mistaken for foreground target. So we need to look for some better method to complete background extraction. There are many domestic and international background extraction and updating algorithm, such as statistical histogram method, statistical median, Kalman filtering, multi-frame image averaging method, Gaussian model, stochastic update method and so on. These background extraction algorithms have good effect, but computational cost is larger, instantaneity is not good.

The core of video background extraction algorithm is background modeling and update. At home and abroad, many researchers are working for background modeling under static scenes. Tian, Y, *etc.* [3] proposed a selective Eigen background method to solve the problem of background modeling under crowd scenes. Mohamed Hammami *etc.* [4] presented a background modeling method which is moving target segmentation based on dynamic matrix and temporal analysis. Bohyung Han *etc.* [5] proposed background modeling and extraction technology based on intelligent pixel with many features. These features are identified through classification. Colors, gradients and Haar features are integrated in the algorithm to handle the temporal variation of each pixel .Youdong Zhao *etc.* [6] proposed a new background model which based on video surveillance short space segment of background modeling. This model performs good effect for variety scenes with illumination changes. Barnich, O, *etc.* [7] proposed a motion detection technology which integrate variety of innovative mechanism. It stores each pixel, the same position or nearby values. Then comparing the current pixel value to determine whether the pixel is background and selecting value at random from the background to replace the model. This method differs from the classical approach; the oldest value is first replaced. Finally, when the pixel is found to be part of the background, its value will be passed to the neighboring pixel of background model.

Background modeling and extraction under static scene is approaching maturity, researchers are focused on background extraction problem under dynamic scenes. Taegyu Lim, *etc.* [8] proposed an online video segmentation of foreground and background with moving camera. The algorithm combined time-domain model with airspace model to generate foreground and background model, and did a model-based likelihood diagram calculations. Combined with energy minimization and graph cut algorithm iteratively update background. Li Cheng *et al* [9] studied video segmentation problem for the foreground when the texture of the background scene changes over time. Background extraction algorithm should minimize the risk of unexpected adverse factors and adapt to changes of time and space quickly, for this question, they proposed a global algorithm. The algorithm used the maximum posterior to distinguish the object and the background clearly on Markov random field, and make use of the nature of parallel algorithms to develop a highly parallel GPU, and effective operation .Xinyi Cui, *etc.* [10] proposed a unified and robust framework to deal with different types of video effectively, such as stationary or moving cameras shoot video. We combine low rank factorization and group

sparse constraint technology, disintegrating trajectory into the foreground and background. Experimental effect is very competitive.

These methods listed above are almost all new video background extraction methods based on static and dynamic scenes. Video background extraction based on dynamic scenes is mainly aimed at background motion situation. The proposed algorithm is also the video background extraction based on dynamic scenes situation which caused by camera motion. The following section 3 will describe this algorithm in detail.

3. Dynamic Scene Background Model Algorithm

Firstly, we use the context to detect saliency target of video frame, detect out the foreground and background of significant portion, and make saliency target be fuzzy enhancement. We Combine the fuzzy set theory and image processing, and use fuzzy set theory for image enhancement. We select threshold using Otsu method, and select the gray value which satisfy maximum variance between classes as the optimum threshold value. After detect significant objectives and their relevant background ,using SIFT flow to detect moving targets and background flow field, and make nonlinear fusion with extracted significant area to get approximate foreground target. We use real-time algorithm to track target. Finally, we use KNN algorithm to get accurate foreground moving targets. The processing flow of the algorithm as shown:

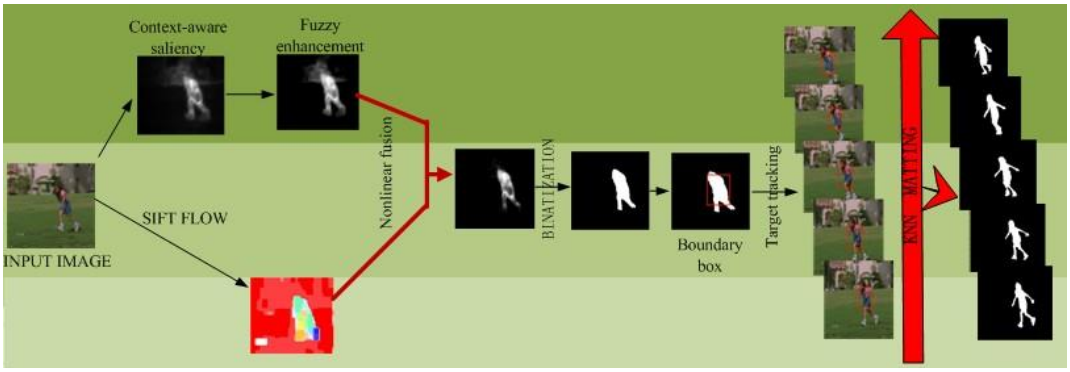


Figure 1. Algorithm Pipeline

3.1. Context-Aware Saliency Detection

Traditional saliency detection algorithms, often only detected significant objects in the foreground, while ignored the background, so that detection result cannot contain certain semantic information (such as location, environment, scenes), and maybe lead to image understanding inaccuracies. Thus, Stas Goferman *et al* proposed a context-related significant detection algorithm, which cannot only detect the significant object in foreground, but also give priority to the significant point in the background; the distance between the point and focus point is minimum. We need to calculate Euclidean distance between the points and nearest foreground significant point and its significance to detect the correlate significant background [11].

Context saliency detection follows four basic principles:

1. Consider the local low-level characteristics like contrast and color.
2. Considering the global features: the frequency of these features is not high.
3. Vision organization rules.
4. Consider advanced Features.

In this section, firstly, defining single-scale local-global significance based on first 3 rules. Then we make use of multi-scale to enhance significant. Next, the algorithm is improved to meet the third rule. The fourth rule is resorted lastly.

Step1: single-scale saliency detection

In the case of single-scale r, first consider the pixel i and the pixel blocks p_i whose center is i. Defining $d_{color}(p_i, p_j)$ as Euclidean distance between two color pixel blocks in Lab color space, normalized to [0, 1].The bigger of color distance between p_i and any p_j, the stronger of saliency. We don't need to consider all j, only consider the k pixel blocks which are most similar to p_i.Because if the color gap between p_i and its most similar pixel blocks is very large, certainly, the other is larger. We call the gap as $d_{color}(p_i, q_k)$.The algorithm also express Euclidean distance of two pixel blocks position with $d_{position}(p_i, q_k)$.According to rule 3, for a pixel block whose center is i, if the color gap between it and its nearest blocks is bigger, then the saliency of pixel is higher. So definition of distance as follow:

$$d(p_i, q_k) = \frac{d_{color}(p_i, q_k)}{1 + c g d_{position}(p_i, q_k)} \tag{1}$$

Where c is a constant, c=3.

Significant calculation formula is:

$$S_i^r = 1 - \exp\{-\frac{1}{k}\sum_{k=1}^{K} d(p_i^r, q_k^r)\} \tag{2}$$

Where, r is scale variable.

Step2: multi-scale saliency detection

In order to make saliency more obvious, we use the multi-scale significance test. The significance of each pixel is normalized by the following formula:

$$S_i^r = [1 - \exp\{-\frac{1}{k}\sum_{k=1}^{k} d(p_i^r, q_k^{r_k})\}] \tag{3}$$

Where, r_k same as r, represent scales of k most similar pixel blocks of i.

M elements are assumed in scale space, final significant is average value of M scale:

$$\overline{S_i} = \frac{1}{M}\sum_{r \in R} s_i^r \tag{4}$$

Step3: Increase the direct context related content

We define d_{foci}^r(i) as position distance between i and its nearest salience pixel. So the salience of i as follow:

$$S_i = \frac{1}{M}\sum_{r \in R} s_i^r (1 - d_{foci}^r \text{ (i} \tag{5}$$

Through the calculation of formula (5), we can increase saliency of background around attention points.

Step4: Preset center

Through the above three steps we have found a significant zone and associated significant scene. But because of people's photograph habits is to put significant objects on middle of the screen. So this algorithm presents that we should find out a center $G(\sigma_x, \sigma_y)$ primarily, and then computes the distance between center and other pixel. The final significant calculation formula as follow:

$$S_i = S_i G_i \tag{6}$$

The effect of this method on the significant figure is not better than formula (5), but it is better than (5) in the actual application. Figure 2 (b) is significance test effect through the contextual detection

(a) (b)

Figure 2. (a) Original Image (b) The Significant Results of Formula (6)

3.2. Moving Targets Tracking

3.2.1 Moving Target Location:

(1) Image Binarization

There are many traditional binarization ways, including the global threshold, local threshold method, dynamic threshold method, *etc*. Although these methods simple to implement, has a wide adaptability and some can also change the threshold value adaptively, but there are a certain degree of defects. Such as limited application, prone to artifacts or exaggerate the neighborhood gray-scale changes of pixel, leading to uneven distribution of the background be divided into goal. For the insufficient of these algorithms, this paper does a new image binarization processing with canny operator. Before processing, we need the image pre-processing steps.

Step1: Compute SIFT flow

SIFT flow is the improvements based on optical flow method. Traditional optical flow method is intensive sampling for extracted the video image data set in the time domain, and adjacent frames within a short time are registered. The SIFT flow register adjacent frames [12] in an image set consisting of variety of scenarios set. Each pixel of every frame is extracted SIFT features, and using the angle of flow to process flow between adjacent frames. We obtain moving target according to the different variation characteristic of the flow field between foreground object and the background. Flow field calculation can be summarized as an optimization problem:

$$E(\mathrm{M}) = \sum_{k} \| N_1(p) - N_2(\mathrm{p}+M) \|_1 + \frac{1}{\sigma^2} \sum_{k} (u^2(p) + v^2(p)) + \sum_{(p,q)\in\varepsilon} \min(\alpha \,|\, \mathrm{u}(p) - \mathrm{u}(q), \mathrm{d}) + \min(\alpha \,|\, \mathrm{v}(p) - v(q), \mathrm{d}) \tag{7}$$

Where, $M(\mathrm{p}) = (\mathrm{u}(\mathrm{p}), \mathrm{v}(\mathrm{p}))$ is displacement vector of point P(x, y). $N_i(\mathrm{p})$ is SIFT descriptor of pixel p of the i-th image. ε is neighborhood size of pixel p, equals 4. α is a constant, equals 0.5. σ is 300.

Target detection process of SIFT flow technology:

1. Feature extraction of pixel point

For the image of each pixel, SIFT features are extracted, and the extracted partial SIFT descriptor forms a 128-dimensional feature vector.

2. Frame matching

Select a frame as the frame to be matched (*e.g.*: Figure 2a) from our video database. Then we use the histogram intersection method, our method is the SIFT space histogram

matching method [10, 11]. We find 30 video frames which are closest with the frame to be matched as an alternative offer of flow field calculation. Select frame as the target from alternative selection 30 frames, the frame has minimum energy compared with frame which to be matched.

3. Flow field calculation

We compute displacement of the image according to difference between the frames to be matched and the target frame, obtained displacement field. We describe the displacement field with STFT descriptor, calculate the minimum energy, and represent with color-code as the final flow field. Flow filed computed by Figure 2 (a) and the optimal matching frame as Figure 3.

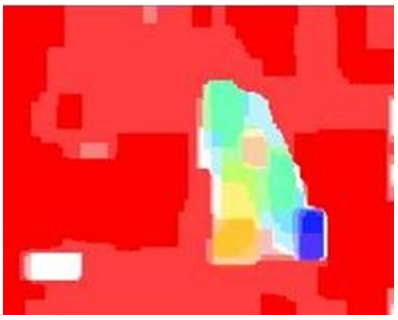


Figure 3. Flow Field

Step2: Fuzzy enhancement

Image would lose some information since three-dimensional objects mapped to the two-dimensional space, so for the image itself, it is uncertainty, namely: ambiguity. So we can use fuzzy set theory to process the image. The traditional and classical fuzzy enhancement algorithm is Pal-King. The algorithm combined fuzzy set theory and image processing for the first time, not only created a new era of fuzzy set theory application areas, but also injected new vitality [13] for the further development of digital image. Every algorithm is not perfect, this algorithm also exist some drawbacks. It is random for fuzzy enhancement threshold selection and depending on experience and multiple try to compare; we need to improve the method to overcome this weakness. Therefore, we improve the fuzzy enhancement method using Otsu to select fuzzy enhancement threshold adaptively [14].

Otsu method based on the principle that use the between class variance as the criterion for selecting the maximum variance value between class as the optimal threshold.

Divided gray value of image into 0~(L-1) levels, set the number of pixel whose gray value is i as n_i,so total pixel number is $N=\sum_{i=0}^{L-1} n_i$.The probability of each pixel values is $p_i=\frac{n_i}{N}$. We divide pixels of image into C_0 and C_1 in an integer t according to gray level, that is $C_0=\{0,1,....,t\}$, $C_1=\{t+1,t+2,...,L-1\}$. The probability of C_0 is $w_0=\sum_{i=0}^{t} p_i=w_t$, and the mean is $\mu_0=\frac{1}{w_0}\sum_{i=0}^{t} i\,p_i=\frac{\mu(t)}{w(t)}$; The probability of C_1 is $w_1=\sum_{i=0}^{t} p_i=w_t$, and the mean is $\mu_1=\frac{1}{w}\sum_{i=t+1}^{L-1} i\,p_i=\frac{\mu-\mu(t)}{1-w(t)}$; $\mu=\sum_{i=0}^{L-1} i\,p_i$ is statistical average of whole image gray, so $\mu=w_0\mu_0+w_1\mu_1$. Optimal threshold value is $t^*=Arg^{\max}_{t=\{0,1,...,L-1\}}\sigma^2(t)$, where $\sigma^2(t)=w_0(\mu_0-\mu)^2+w_1(\mu_1-\mu)^2=w_0w_1(\mu_1-\mu_0)^2$. Therefore, we can select threshold adaptively.

Set matrix of two-dimensional image X whose gray is L and scale is $M\times N$ as follow:

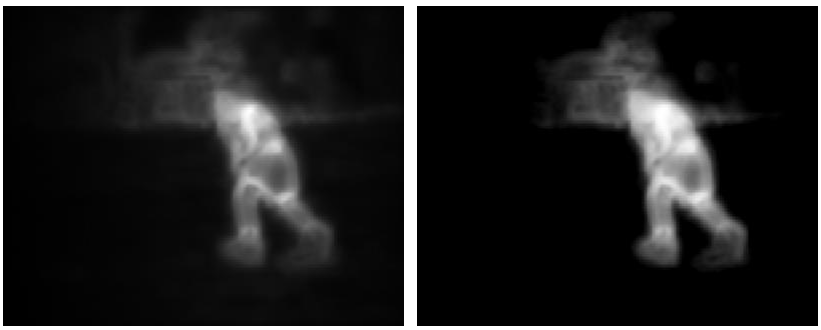
$$X = \begin{pmatrix} \frac{u_{11}}{X_{11}} & \frac{u_{12}}{X_{12}} & \cdots & \frac{u_{1N}}{X_{1N}} \\ \frac{u_{21}}{X_{21}} & \frac{u_{22}}{X_{22}} & \cdots & \frac{u_{2N}}{X_{2N}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{u_{M1}}{X_{M1}} & \frac{u_{M2}}{X_{M2}} & \cdots & \frac{u_{MN}}{X_{MN}} \end{pmatrix}$$

Where, X_{ij} is the special gray level of pixel (i, j), $u_{ij}\left(0 \le u_{ij} \le 1\right)$ is membership corresponding to X_{ij} . Choices of membership function have a great influence on the image detection effect. Membership function of Pal is $u_{ij} = F\left(x_{ij}\right) = \left[1 + \frac{(L-1) - x_{ij}}{F_d}\right]^{-F_e}$. Where, F_d and F_e are Called inverse fuzzy factors and index fuzzy factors respectively.

Generally F_e equals 2. To the image fuzzy enhancement processing, we should use transformation firstly:

$$\begin{cases} \mu_{ij} = I_r(\mu_{ij}) = I_1(I_{r-1}(\mu_{ij})), \mathrm{r} = 1, 2, \ldots & \text{(a)} \\ I_1(\mu_{ij}) = \begin{cases} 2\mu_{ij}^{\ 2} & 0 \le \mu_{ij} \le 0.5 \\ 1 - 2(1-\mu_{ij})^2 & 0.5 \le \mu_{ij} \le 1 \end{cases} & \text{(b)} \end{cases}$$

Where, the result of (b) formula is to increase ($\mu_{ij} > t^*$) or decrease ($\mu_{ij} \le t^*$) value of μ_{ij} .We do inverse transformation to μ_{ij} , and then get image X' through fuzzy enhancement, the gray value of (i, j) is $x_{ij} = F^{-1}(\mu_{ij})$. The effect though fuzzy enhancement as follow:



(a) (b)

Figure 4. (a) Moving Object (b) Effect after Fuzzy Enhancement

Step3: Nonlinear fusion

There are many image fusion techniques, based on pixel level, feature-level, as well as decision-making level. Xuerong Chen, *etc.* [15] proposed a non-linear fusion based on fuzzy integral used in face recognition, this is the decision level fusion. Here, we use pixel level fusion technique to solve this problem [18] based on the synchronous orthogonal matching pursuit.

The key technology of this process is synchronous orthogonal matching pursuit (SOMP).Firstly, we define a Directory object $D = [d_1, d_2, \ldots, d_r]$ to present each signal

$(x_k)_{k=1}^{K}$. Where, K=2. Set the initial value of error term is $r_k^{(0)} = x_k$, number of iteration is $l = 1$. Define subscript variables $\hat{t}_l$, present subscript of pixel which can fuse all of signals optimally. The compute formula is as follow:

$$\hat{t}_l = \arg \max_{t=1,2T} \sum_{k=1}^{K} \left\langle r_k^{l-1}, d_t \right\rangle \tag{8}$$

Next update data set: $\Phi_l = [\Phi_{l-1}, d_{\hat{t}_l}]$, where the initial value of Φ is null.

Calculate the new coefficient, similarity and error:

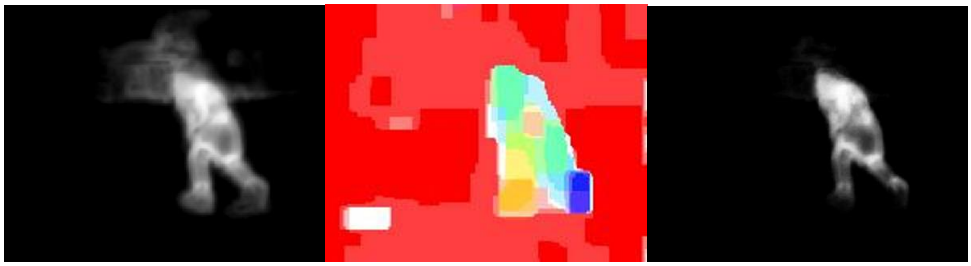
$$\alpha_k^{(1)} = \arg \min_{\alpha} \| x_k - \Phi \alpha \|_2^2 = (\Phi_t^T \Phi_l)^{-1} \Phi_t^T x_k \quad for\ k = \tag{9}$$

$$\hat{x}_k^{(1)} = \Phi_t \alpha_k^{(1)}, \quad for \quad k = 1, \tag{10}$$

$$r_k^{(1)} = x_k - \hat{x}_k^{(1)}, \quad for \quad k = 1, \tag{11}$$

Where, l=l+1. If $\sum_{k=1}^{K} \| r_k^{(l)} \|_2^2 > \varepsilon^2$, we need to return to continue to find $\hat{t}_l$. Otherwise, iterated progress is over. When we get optimal coefficient, we can fuse it as image (5).

Figure 5(c) is fusion between moving target in 5(a) and flow filed in 5(b):



(a) (b) (c)

Figure 5. Nonlinear Fusion

Step4: Binarization

Firstly, we should detect edge of the fused image using canny descriptor, and then get high and low thresholds adaptively according to the edge. Point which has high gray value and on the edge of image as high threshold, gray value of the point which is not isolated and near the edge of image as the seed points. We can do a seed filling for the high threshold point in the binary image, gray average value of low gray value point as low threshold. If the ratio of seed points is bigger in the edge of area and the proportion is bigger than certain threshold, we will set the point of filling area as target point. That area without seed point or with a little ratio of seed point will be set as background point, so we can get binary image.

Binarization algorithm with canny descriptor as follow:

1) Canny detection on the fused graph, getting the edge;

2) To remove isolated edge, the four neighborhood of the non edge point is divided into two parts, one part has high gray value, and the other is low;

3) The average gray value of two part point will be as high or low threshold respectively;

4) To get binary image of high threshold, then we will fill the seed point;

5) If the ratio of seed points is bigger in the edge of area and the proportion is bigger than certain threshold, we will set the point of filling area as target point. That area without seed point or with a little ratio of seed point will be set as background point, so we can get binary image.

Figure 6. Binarization

(2) Identify the target boundary box

To determine the target bounding box, we require the optimal bounding box. There are many ways to find bounding box, here we use Otsu: Let the value of black background is 0 and white foreground is 1.The ratio of pixel inner bounding box is w_0, average gray is u_0. The ratio of pixel outer bounding box is w_1, average gray is u_1. Thus the average gray of image is $u = w_0 * u_0 + w_1 * u_1$.As the same as Otsu in fuzzy enhancement, the Otsu is also the biggest variance that distinguish foreground and background. The formula as follow:

$$g = w_0 * (u_0 - u) + w_1 * (u_1 - u) = w_0 * w_1 * (u_0 - u_1) * (u_0 - u_1) \quad (12)$$

When the variance g is maximum, difference between foreground and background can be considered the biggest at this time, that is, gray at this time is the best threshold. After threshold is determined, according to the AABB bounding box algorithm to calculate the minimum bounding box that contains the moving target. Shown in Figure 7, that red line is the bounding box of the target.

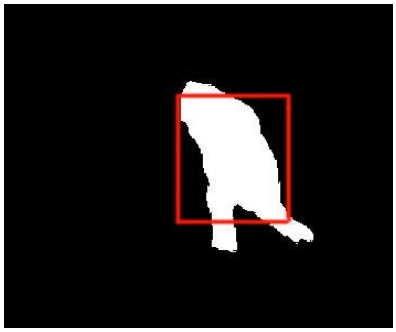


Figure 7. Bounding Box

3.2.2 Target Tracking: For the extraction of a goal in video, the key task is target tracking. There are many ways for target tracking, but due to the target object be affect at light, posture, and many other factors, and the application of some real-time algorithms are to face difficulties because of training samples appear as poorly cut, or other problems. In order to make up for the deficiency of these algorithms, we need to extract target feature in compressed domain.

By sparse sensing theory, we can know that do projection of the original image feature space by very sparse measurement matrix which meet compressed sensing distance constraints (RIP), a low dimension subspace can be got. Low dimension subspace can preserve the high dimensional image feature space information[16]. So through our sparse measurement matrices to reduce image dimension, extract feature of foreground and

background. We take these features as positive samples and negative samples of online learning classifier and then use the naïve Bias classifier to classify the target image slice of the next frame image.

The main processing steps:

Step1: In t frame, we sample a number of target (positive samples) and background (negative sample), and then they do multi-scale transform, through a sparse measurement matrix to reduce the dimension of multi-scale image features, and then apply dimensionality reduction feature (including the target and background , belong to two classification problems) to train Naive Bayes classifier.

(1) A random matrix which scale is $n \times m$,it can convert X(m dimension) in high dimension image space to V(n dimension) in low dimension image space. Mathematical expression is V=RX. We hope that V can reserve information of X, or keep distance between each sample in original space, so classification in low dimension is meaningful.

A typical measurement matrix which satisfied RIP condition is random Gaussian matrix; the matrix element satisfies the N (0, 1) distribution. However, if m is large dimension matrix, then the matrix is relatively dense, and its operation and the storage consumption are relatively large. This paper uses a very sparse random measurement matrix whose matrix elements are defined as:

$$r_{i,j} = \sqrt{s} \times \begin{cases} 1 & with\ probability\ \frac{1}{2s} \\ 0 & with\ probability\ 1-\frac{1}{s} \\ -1 & with\ probability\ \frac{1}{2s} \end{cases} \tag{13}$$

Where, if s=2 or s=3, matrix can meet Johnson-Lindenstrauss in 13 ference. This matrix is very easy to calculate, since it requires only a uniform random number generator on the line, and when s = 3, this matrix is sparse; calculation will be reduced by 2/3. If s = 3, then the matrix element has a 1/6 probability of 1.732, 1/6 probability of -1.732, 2/3 probability of 0;

Here s = m / 4, each row of the matrix R is only necessary to calculate c (less than 4) element. So its computational complexity is O (cn). In addition, we only need to store non-zero elements of R, so very little storage space is required.

(2)For each sample z (m-dimensional vector), its low-dimensional representation is v (n-dimensional vector, n is much smaller than m). Assuming each element of v in the distribution is independent. We can use Naive Bayesian classifier to model.

$$H(v) = \log \frac{\prod_{i=1}^{n} p(v_i \mid y=1)\, p(y=1)}{\prod_{i=1}^{n} p(v_i \mid y=0)\, p(y=0)} = \sum_{i=1}^{n} \frac{p(v_i \mid y=1)}{p(v_i \mid y=0)} \tag{14}$$

Where, $y \in \{0,1\}$ represent sample label, y = 0 indicates a negative sample, y = 1 indicates a positive sample, assuming priori probability of the two classes is equal. P (y = 1) = p (y = 0) = 0.5. Diaconis and Freedman proved the random projection of high dimensional random vector is almost Gaussian distribution.

Therefore, we assume conditional probability p (v_i | y = 1) and p (v_i | y = 0) in the classifier H (V) is a Gaussian distribution, and can be described by four parameters:

$$(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0),\ p(v_i \mid y=1) \sim N(\mu_i^1, \sigma_i^1),\ p(v_i \mid y=0) \sim N(\mu_i^0, \sigma_i^0)$$

Four parameters will update incremental:

$$\mu_i^1 \leftarrow \lambda \mu_i^1 + (1-\lambda)\mu \tag{15}$$

$$\sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2} \tag{16}$$

Where, learning factor $\lambda>0$, $\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n=1}(v_i(k)-\mu^1)^2}$, $\mu^1 = \frac{1}{n}\sum_{k=0|y=1}^{n=1} v_i(k)$.

Step2: In t +1 frame, we sample n scan window (to avoid going to scan the whole image) around the target position which was tracked last time, through the same sparse matrix to reduce its dimension, and extract feature, and then classify them with Naive Bayes classifier which were trained using the t-th frame. We take the window which has maximum score as target window. This realize target tracking from t frame to t +1 frame.

3.3. Foreground Extraction

After track to the target, we need to extract the target. We use the K-nearest neighbor matting algorithm to extract tracked target [17].

First use K-nearest neighbor algorithm to calculate K-nearest neighbor j of each pixel i in the feature space, here assumed that K = 3. Feature vector is presented by spatial coordinates X. Feature vectors are usually presented by following formula:

$$X(i) = (\cos(h), \sin(h), s, v) \tag{17}$$

Where: h, s, v represents the three components in HSV space, x, y are the horizontal and vertical coordinates of pixels.

We use kernel function to complete Soft partition. Choice of kernel function is generally 1-x:

$$K(i, j) = 1 - \frac{||X(i) - X(j)||}{C} \tag{18}$$

Here, C is the upper bound of $||X(i) - X(j)||$, so you can ensure that $K(i, j)$ is in the [0, 1] interval.

We define a sparse matrix A whose scale is N * N, A (i, j) = K (i, j), the other values of matrix would be set to 0. The objective function to be extracted is defined as:

$$E(x) = \sum_i \sum_j A_{i,j}(x_i - x_j)^2 + \lambda \sum_{i \in m-v} x_i^2 + \lambda \sum_{i \in v} (1 - x_i)^2 \tag{19}$$

Where, m represents bounding box of all layers of the target track; v is the bounding box of current level of tracking target.

This formula can be optimized using follow formula:

$$x^* = (L + \lambda D)^{-1}(\lambda v) \Leftrightarrow (L + \lambda D)x^* = \lambda v \tag{20}$$

Figure 8 (a) is the original image; Figure 9 (b) is the extracted target through KNN matting.

(a) (b)

Figure 8. Foreground Extraction

4. Experiment and Result Analysis

As shown in Figure 9, this paper extracted moving target for two videos. The first video is on the sports field, camera followed the little girl in the course of the movement, process tracking and extracted from the scene of the little girl. For the selection of each frame in the experiment, we all select the closest frame through the histogram cross matching selection method. The video in fifth line is the progress that camera shoot following parachute moving. Frame selection process is the same as first row.

To demonstrate the effectiveness of the proposed algorithm, we compared detection and extraction effect of our paper with the other two algorithms. The following are results of this experiment, as well as the other two methods, and comparison of the effects. The first line in Figure 10 is a video frame, the second and sixth lines are VIBE extraction effect, the third and seventh lines are real-time background subtraction algorithm, the fourth line and the last line is the effect of this algorithm. It can be seen from the figure that second (six) and third line (seven) for the effect of the method are not satisfactory, extracted a lot of background, the edge is not clear, the foreground and background are very fuzzy, the boundary is not clear. The two methods can detect changes in the background, from second (six) and third (seven) rows can be seen in figure. However, the background changes effected target detection and extraction of an impact, so that the target and the background cannot be separated completely. The extraction effect of our method is very good, extraction foreground object is complete and clear, the foreground edge is significantly, outline is clear. Background changes, have no impact on detection and extraction of background, instead, we can ignore the background and only extract the foreground objects accurately. Subjective evaluate the experimental results, we can get good effect compared to the other two methods.

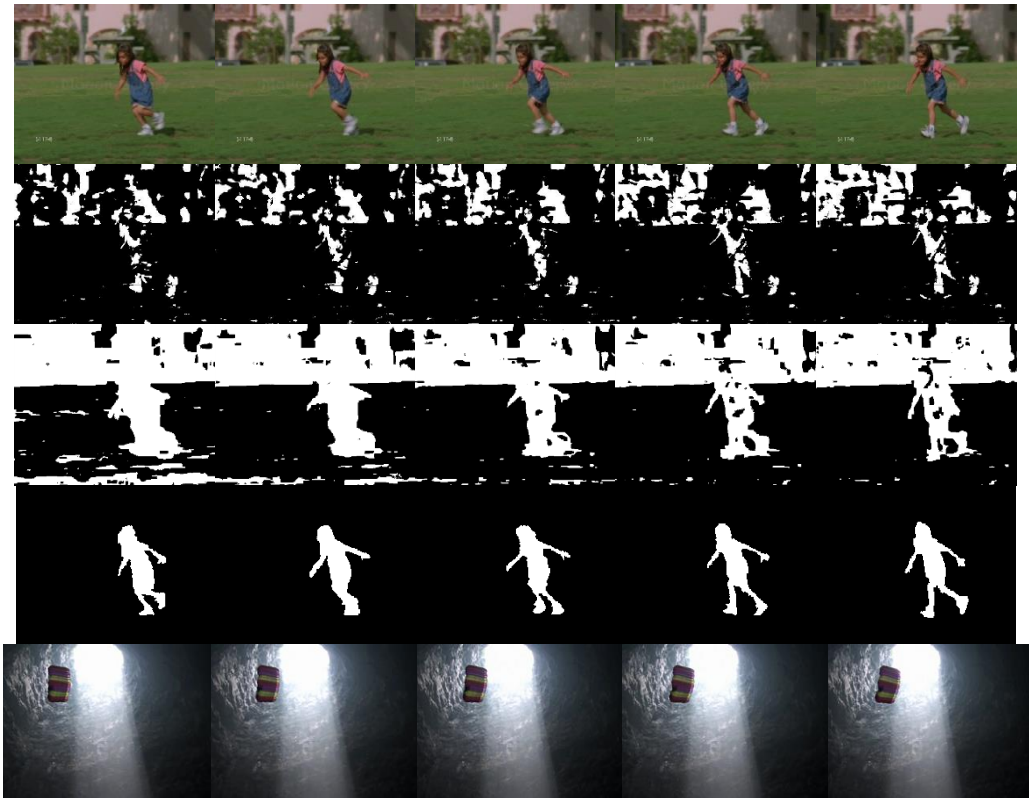


Figure 9. Row 1and Row 4: Video Frame; Row 2 and Row 6: VIBE; Row 3 and Row 7: Real-Time Discriminative Background Subtraction; Row 4 and Row8: our Method

5. Conclusion

There have been some related algorithms about extraction technology of dynamic video background, for the advantages and disadvantages of the existing algorithms; this paper put forward the extraction algorithm of dynamic scenes video background. The algorithm proposed a method for dynamic scene background model: first detect the target and the neighboring background pixels using context correlation detection algorithm and using the SIFT flow method to detect flow filed of moving object and background. Second, the significant test results were fuzzy enhanced and using Canny operator to do binarization with flow field; once again using Otsu method to get optimal bounding box to determine the boundaries, using real-time compression algorithm for target tracking; Final, applying KNN Matting extract foreground objects. Experiments show that: the effect of our method is very good, very complete and clear, the foreground edge is significantly, line is clear. Next, we will study dynamic scene for zoom.

Acknowledgements

Project supported by the National Nature Science Foundation of China (No. 61272430, 61173173, 61272245, 61472227); the Provincial Natural Science Foundation of Shandong under Grant No. ZR2013FM015.

References

[1] A. J. Lipton, H. Fujiyoshi and R. S. Patil, "Moving Target Classification and Tracking from Real-time Video," Proceeding of the 4th IEEE Workshop on Application of Computer Vision; Princeton, New Jersey, October 19-21, **(1998)**.
[2] L. G. Brown, "A Survey of Image Registration Techniques," ACM Computing Surveys, vol. 4 no. 24, **(1992)**.
[3] Y. Tian, Y. Wang, Z. Hu and T. Huang, "Selective Eigen background for Background Modeling and Subtraction in Crowded Scenes," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23 no. 11, **(2013)**.
[4] M. Hammami, S. K. Jarraya and H. B. Abdallah, "On line background modeling for moving object segmentation in dynamic scenes," Multimedia Tools and Applications, vol. 3 no. 63, **(2013)**.
[5] B. Han and L. S. Davis, "Density-Based Multifeature Background Subtraction with Support Vector Machine," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5 no. 34, **(2012)**.
[6] Y. Zhao, H. Gong, Y. Jia and S. Zhu, "Background modeling by subspace learning on spatio-temporal patches," Pattern Recognition Letters, vol. 9 no. 33, **(2012)**.

[7] O. Barnich and M. V. Droogenbroeck, "ViBe: A powerful random technique to estimate the background in video sequences," Proceeding of the 34th IEEE International Conference on Acoustics, Speech, and Signal Processing; Taipei, April 19-24, **(2009)**.
[8] T. Lim, B. Han and J. H. Han, "Modeling and segmentation of floating foreground and background in videos," Pattern Recognition, vol. 4 no. 45, **(2012)**.
[9] L. Cheng, M. Gong, D. Schuurmans and T. Caelli, "Real-Time Discriminative Background Subtraction," IEEE Transactions on Image Processing, vol. 5 no.20, **(2011)**.
[10] X. Cui, J. Huang, S. Zhang and D. N. Metaxas, "Background Subtraction Using Low Rank and Group Sparsity Constraints," Proceeding of the 12th European Conference on Computer Vision; Firenze, Italy, October 7-13, **(2012)**.
[11] S. Goferman and L. Z. Manor, "Context-Aware Saliency Detection," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 10 no. 34, **(2012)**.
[12] C. Liu, J. Yuen, A. Torralba, J. Sivic and W. T. Freeman, "SIFT Flow: Dense Correspondence across Different Scenes," Proceeding of the 10th European Conference on Computer Vision; Marseille, France, October 12-18, **(2008)**.
[13] S. K. Pal and R. A. King, "Image Enhancement Using Fuzzy Sets," Electron Device Letters, vol. 9 no. 16, **(1980)**.
[14] N. Otsu, "Threshold Selection Method From Gray Level Histograms," IEEE Transaction on System and Man Cybernetics, vol. 1 no. 9, **(1979)**.
[15] X. Chen, Z. Jing and G. Xiao, "Nonlinear fusion for face recognition using fuzzy integral," Communications in Nonlinear Science and Numerical Simulation, vol. 5 no. 12, **(2007)**.
[16] K. Zhang, L. Zhang and M. H. Yang, "Real-Time Compressive Tracking," Proceeding of the 12th European Conference on Computer Vision; Firenze, Italy, October 7-13, **(2012)**.
[17] Q. Chen, D. Li and C. K. Tang, "KNN Matting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9 no. 35, **(2013)**.
[18] B. Yang and S. Li, "Pixel-level image fusion with simultaneous orthogonal matching pursuit," Information Fusion, vol. 1 no. 13, **(2012)**.

