EXTENDED SCALE INVARIANT LOCAL BINARY PATTERN FOR BACKGROUND SUBTRACTION

Parth S. Khadse   Aniket A. Aitawade   Sandeep Kumar

Indian Institute Of Technology, Kharagpur, India

ABSTRACT

Background Subtraction is a important part in many Video surveillance systems and applications. A good Background subtraction algorithm should be able to classify the Background and the Foreground correctly and accurately and should be able to adapt background scenes robustly. In this paper, it is proposed by the authors that along with colour intensities local texture features represented by extended scale invariant local binary patterns could be used to characterize the pixel representations. These Local Texture features show good tolerance against the illumination in the regions where there is illumination variations in rich texture area. But in areas with illumination varations and uniform texture regions, this texture features are not of much help so a photometric invariant measurement of colour is proposed to overcome this limitation.

1. INTRODUCTION

In today's age of Artificial Intelligence and Machine Learning we use computer vision for activity recognition,object tracking,video surveillance which are rely on pixel level segmentation of scene into foreground and background. Background subtraction has four main steps:

- Initialisation of model
- Representation of model
- Model update
- Foreground detection

There are many methods for background subtraction but they all need to deal with some challenges such as illumination variations,dynamic backgrounds, camera jitter, bad weather, noise and shadows.Most of the background subtraction algorithms are mainly based on two aspects, the first one is to represent of background in advanced probabilistic models, the second one is to employ a more powerful feature descriptor or combine different features together. Generally the colour features are used for background subtraction process but they generally face the issue of illumination varations, camoflage and Shadows. So to overcome this Texture features are developed, to deal with these situations.

In this paper, the authors propose a single model, single update scheme, spatio-temporal-based background subtraction algorithm. The key aspects of our method are as follows:

1. It has been proposed a new powerful texture feature called Extended Scale Invariant Local Binary Patterns (ESILBP) over Traditional Local Binary Patterns (LBP) methods.
2. To deal with illumination variations a Photometric Colour Invariant method is proposed.
3. ESILBP and colour features have their own pros and cons but they can compensate each other, so the authors are combining them in the background subtraction framework for better performance.
4. In this a model update method is derived from the ViBe algorithm for the Background model update over time.

2. TEXTURE AND COLOUR FEATURES

In this section, the features proposed previously called the ESILBP and Photometric invariant Colour measurements will be introduced.

2.1. Texture description and ESILBP

The Local Binary Pattern is a powerful texture detector. The pixels in the small image block are labelled by thresholding the neighbourhood pixels of the central pixel and then each of the neighbouring pixel is assigned a binary number. Let the coordinate of the central pixel be (x_c,y_c), and there are N neighbouring pixels spaced at a radius of R. The LBP operator applied on pixel $c(x_c,y_c)$ can be represented as,

$$LBP_{N,R}(C) = \sum_{i=0}^{N-1} S(g_i - g_c)2^i \tag{1}$$

Where the g_c is the grey intensity value of the central pixel c, g_i is the grey intensity value of the N neighbouring pixels spaced around the central pixel at a radius of R and S is the Thresholding function expressed as,

$$S(x) = \begin{cases} 1, & \text{if x} \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Still the LBP is not powerful enough to resist the image noise; even a small variation in the central pixel can lead to variations in the output code. To overcome this problem another method called Local Ternary Pattern (LTP) operator which solves the problem mentioned above by adding a small offset value for comparison.

$$S(x) = \begin{cases} 1, & \text{if x} \geq T; \\ \text{-1}, & \text{x} < T. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Where T is the Threshold value used to add robustness. However, the LTP would also sometimes fail as it is not invariant to the scaling of the intesnity values by multiplying a constant. For example, say all the intensity values are multiplied by 3 still the LTP descriptor cannot keep its invariance against the scaling. To solve this problem agaist the scale transform proposed SILTP operator. Given a central pixel $c(x_c,y_c)$ SILTP operator can be expressed as,

$$SILTP^{\tau}_{N,R}(c) = \bigoplus_{k=0}^{N-1} s_{\tau}(I_c, I_k) \tag{4}$$

Where I_c is the gray value of the central pixel c, I_k is the grey value of the N neighhbouring pixels which are spaced uniformly around the central pixel at a radius of R. Here the $\oplus$ operator represents the concatenation operator of the binary strings, τ represents the parameter affecting the tolerance range. s_τ is defined as,

$$s_{\tau}(I_c, I_k) = \begin{cases} 01, & if I_k > (1+\tau)I_c; \\ 10, & if I_k < (1-\tau)I_c; \\ 00, & \text{otherwise.} \end{cases} \tag{5}$$

The main properties of the SILTP operator are its resistance to the illumination variations and image noise within the range. However, the SILTP operator only uses the four neighbouring pixels for getting the SILTP encoding. Here in this paper authors have introduced the ESILBP operator in which more amount of information from the neighbourhood is taken into consideration. That is along with the four neighbouring pixels the four corner pixels of k x k neighbourhood are also considered. The main advantages of using the ESILBP operator are: firstly, it can represent more amount of texture data as compared to the previous two methods, secondly, it is more tolerant to the image noise and the variations in the illuminations due to the scale factor introduced here. And Finally, it is computationally efiicient as it requires only one more comparison as compared to the LBP for each neighbouring pixel.

2.2. Photometric invariant colour measurement

Only relying on the texture features is not enough as they may fail in some cases where there is a uniform and flat regions. In such cases the colour features play an importatnt role. RGB representation of colours is sensitive to the illumination variations, so many algorithms use normalized RGB representation of colours to tackle this problem. Still they do not work very well in the dark regions. Based on certain observations, here they compared the colour difference using the relative angle with respect to the origin colour point (0,0,0) in the RGB plane and the change of range of the brightness of pixels of image. The Colour Difference between the input pixel $I^t(p)$ and the background model pixel $I_k^{t-1}(p)$ can be defined as,

$$\begin{aligned} Dist(I_k^{t-1}(p), I^t(p)) = max(D_A(I_k^{t-1}(p), I^t(p)), \\ D_R(I_k^{t-1}(p), I^t(p))) \end{aligned} \tag{6}$$

where $D_A(I_k^{t-1}(p), I^t(p))$ denotes the relative angle of $I^t(p)$ and $I_k^{t-1}(p)$, and $D_R(I_k^{t-1}(p), I^t(p))$ is the range within which we allow the colour changes to vary. The D_A is defined as

$$D_A(I_k^{t-1}(p), I^t(p)) = 1 - e^{-max(0, \theta - \theta_n)} \tag{7}$$

where θ is the angle between two RGB vectors I_k^{t-1} and $I_t.\theta_n$ is the largest angle between the RGB vector of I_t and the noise RGB vector of $\tilde{I}^t$, and the authors have empirically set it to 3°. The D_R is defined as

$$D_R(I_k^{t-1}, I^t) = \begin{cases} 0, & if \tilde{I}_{s,k} < I^t < \hat{I}_{h,k}; \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

where $\tilde{I}_{s,k} = min(\lambda I_k^{t-1}, \tilde{I}_k^{t-1})(\lambda \in [0.4, 0.7])$, and $\hat{I}_{h,k}$= max$(\eta I_k^{t-1}, \hat{I}_k^{t-1})(\eta \in [1, 1.2])$.

3. BACKGROUND MODELLING

In this section we take on the ESILBP features and the colour features to the statistical model of Background subtraction and also describe in detail the steps included in the structure of Background subtraction model which include the initialization of background model, representation of background model, Foreground Detection and Updatation of the Background model. Fig.1 shows the flowchart of the structure of the Background Subtraction model.

3.1. Background model representation

Most of the algorithms used for Background subtraction depend heavily on the Probablity density functons and statistical parameters for the background generation process. ViBe has presented a creative mechanism for background model updation. It presented that there is a high probablity that the obeserved pixel sample could appear again, so it relies on the collection and maintainence of the background model using a

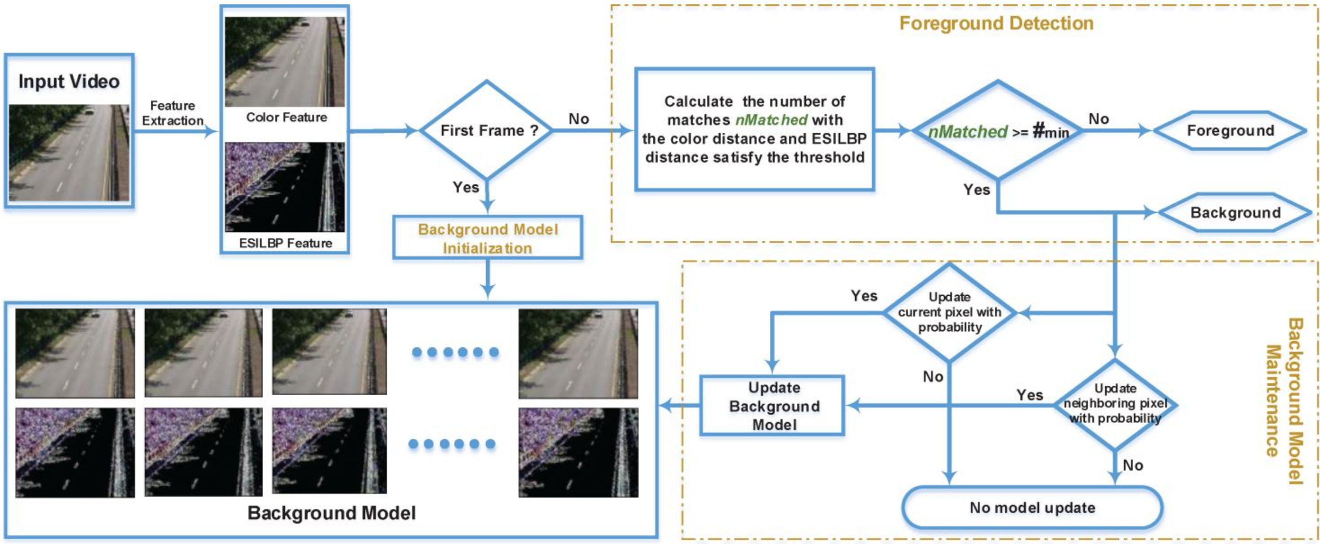


Fig. 1. Flow chart of the proposed background subtraction algorithm

random approach. The authors try to use the similar approach here as that of ViBe for the background modelling. ViBe only takes the colour features in consideration while here authors tried to amalgamate the colour features with the ESILBP features to denote the pixel representation. So here for each pixel $p(x, y)$, the Background model $B(p)$ would contain N background sample values.

$$B(p) = \phi_1(p), \phi_2(p), \phi_3(p), \ldots, \phi_N(p) \tag{9}$$

where $\phi_i(p) = I_i(p), ESILBP_i(p)$ is the previously observedbackground samples and each containing a colour feature $I_i(p)$ and an ESILBP feature $ESILBP_i(p)$.

3.2. Background model initialisation

Generally many Background Subtraction models need a sequence of frames for the background initialization. But here try to find a convinient solution to update the background model using only one frame. As not much temporal information is not contatined in a single frame, it seems to be no other choice than to take values from the spatial neighbourhood. It is assumed that the neighbouring pixels share a same temporal distribution. For a pixel $p(x, y)$ in the initialized image, the background model B(p) is initialized by randomly selecting the feature values from the neighbouring pixels for N times.

$$B(p) = \{\phi(\bar{p}) | \bar{p} \in N(p)\} \tag{10}$$

Where N(p) is the neighbouring pixel of p and the probability of choosing $\bar{p}$ follows a Gaussian distribution. In our experiments, a 5×5 neighbourhood region has proved to be a good choice. It is recommended to take the number of background sample values per pixel $N = 20$. The value of N controls the Balance of precision and sensitivity of the model. Also choosing the value of N to be large could increase the memory requirement and computational complexity.

3.3. Foreground detection

In the Foreground detection process, the the input frame at time t is represented as I^t, to classify a pixel $p^t(x, y)$ as foreground or background, for that authors try to calculate number of matches between the input pixel $p^t(x, y)$ and backgorund model pixel and the $B(p)$. This process is represented by the following equation,

$$M(p^t(x, y)) = \#\{i | dist(p^t(x, y), \phi_i(p)) < T, i \in [1, N]\} \tag{11}$$

Where $M(p^t(x, y))$ is the number of matches, $dist(p^t(x, y), \phi_i(p))$calculated the distance between $p^t(x, y)$ with its background model samples. It should be noticed that the input pixel $p^t(x, y)$ contains colour feature $I^t(p)$ and ESILBP feature $ESILBP^t(p)$, these two distances are calculated in two different ways. To calculate the colour similarity generally L1 or L2 norms are used but they have a demerit that they perform poorly under illumination variations and shadows. So here authors have used a Photometric invariant colour measurement which was described earlier in this paper in section 2.2 to measure the colour similarity. If the colour distance measured using equation (6) is less than the threshold value T_A then it can be said that a colour match is found. Then, we try to calculate the similarity between two ESILBP vectors, the similarity is found by taking the hamming distance between them, If this distance is less than T_desc then it is said that a ESILBP match is found. If both the features match then it can be said that the pixel is matched to the

Input : pixel $p(x, y)$
Output : the FG/BG label of $p(x, y)$

$colorDist = 0$, $textureDist = 0$, $nMatches = 0$, $i = 0$
while $nMathes < \#_{min}$ && $i < N$
  $colorDist = Dist(I_i(p), I^t(p))$
  if $colorDist > T_A$
    goto failedMatch;
  for $c = 1 : nChannels$
    $textureDist$ += $ESILBP^t_c(p) \oplus ESILBP_{i,c}(p)$
  if $textureDist > nChannels \cdot T_{desc}$
    goto failedMatch;
  $nMatches$++;
  failedMatch:
    i++;
if $nMatches < \#_{min}$
  $p(x, y)$ is foreground;
else
  $p(x, y)$ is background;

Fig. 2. Algorithm 1: foreground detection

background model. If either of them do not match then that pixel is classified as a Foreground pixel.

$$M(p^t(x,y)) = \#\{i|dist(I^t(p), I_i(p)) < T_A\&\& \\ dist(ESILBP^t(p), ESILBP_i(p)) < T_desc, i \in [1, N]\} \tag{12}$$

After the number of matches is found, the label of pixel $p^t(x, y)$ is classified as follows:

$$S^t(p) = \begin{cases} 1, & if M(p^t(x,y)) < \#_{min} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Where $S^t(p)$ is the output segmentation map and 1 means foreground and 0 means background. $\#_{min}$ is the minimum number of matches required for a pixel. In this paper, authors have set $\#_{min}$ = 2 to get a reasonable trade-off between computational complexity and noise resistance. The pseudocode of this process is shown in 2.

3.4. Background model maintenance

Most of the Algorithms generally use the First In First out strategy to update the background model, but it is not proven to be optimal. Here in this paper, we update the model with kind of similar strategy.When the pixel $p(x, y)$ is classified as a Background pixel out of N samples, we randomly choose any one sample and update its colour and ESILBP features with the features of pixel $p(x, y)$. The reason for this random updation of the Background model is that it retains the long-short term representations of the Background Model, which would not be the case if the model was updated in the FIFO sense. Also it makes sure that the model not only depend on the near time Background representation but also depend on the long time before background model data also.

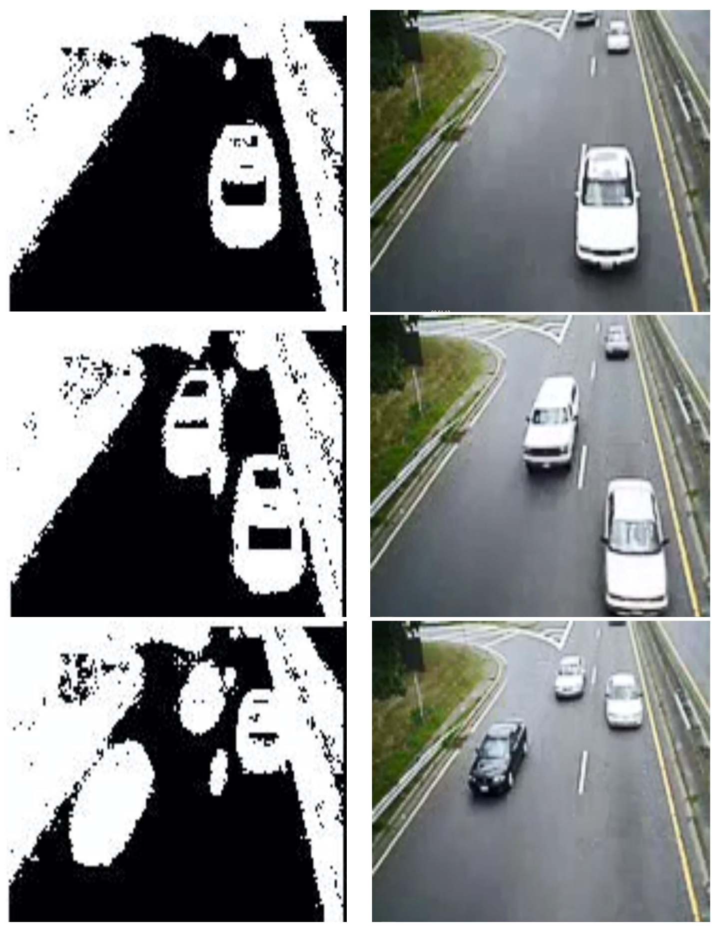
4. EXPERIMENTAL RESULTS

We have taken a input video of cars moving on a road as input to the Algorithm. It can be seen from the results that road is classified as background correctly and the cars are classified as Foreground.The details of the video are as follows:

- Length of Video : 8 seconds
- Frame Rate : 15 Frames per second
- Dimensions of each frame : 120px x 160px
- Total Computation Time : approx. 4 hours

In our experiment we have taken the following parameter values:

1. $T_A = 0.2$: Colour distance Threshold.
2. $T_{desc} = 2$: Texture Descriptor Threshold.
3. $N = 20$: Number of samples stored in the background model.
4. $_{min} = 2$: Number of minimum matches to classify pixel as background.
5. $\lambda = 0.5, \eta = 1.2$: parameters used in equation6.
6. $\tau = 0.3$ scale factor used to calculate ESILBP features in 5



5. REFERENCES

Zeng, D., Zhu, M., Xu, F. and Zhou, T. (2018), Extended scale invariant local binary pattern for background subtraction. IET Image Processing, 12: 1292-1302. https://doi.org/10.1049/iet-ipr.2016.1026