MULTI-FEATURE FUSION BASED BACKGROUND SUBTRACTION FOR VIDEO SEQUENCES WITH STRONG BACKGROUND CHANGES

Zhenkun Huang[⋆†] *Ruimin Hu*[⋆†] *Bouwmans Thierry*[‡] *Shihong Chen*[⋆†]

[⋆] National Engineering Research Center for Multimedia Software,
School of computer, Wuhan University, China
[†]Research Institute of Wuhan University in Shenzhen, China
[‡] Lab. MIA, Univ. La Rochelle, France

ABSTRACT

Current background subtraction algorithms are sensitive to sudden changes. In this paper, we propose a multi-feature fusion scheme to background subtraction for video sequences with strong background changes. We reconstruct the whole videos frame by frame by fusing several video features. In this fusing step, we design an energy function based on enforcing every features with an equal weight. By comparing reconstruction videos with the original videos, pixels with small differences are classified as background pixels. Thus, we can identify background areas in advance and then we construct a contour-based mask combining mechanism. Experimental results conducted on the OTCBVS, BMC 2012 and PETS 2001 datasets show that our method improves the performance of the Zivkovic's GMM and SubSENSE for video sequences with strong background changes.

Index Terms— Background subtraction, Background modelling, Feature Fusion

I. INTRODUCTION

Background subtraction is a kind of effective moving area detection algorithm for videos captured by static cameras, and is a fundamental pre-processing step in computer vision. For example, it supports the coding of surveillance videos [1][2], object extraction and tracking in the surveillance videos [3][4], and other further applications. Background subtraction consists of initializing the background image, updating the background model and comparing the model to the input frames. Pixels with an obvious difference are assumed to be foreground pixels, which means these pixels belong to moving objects. Recently many background subtraction algorithms (such as Gaussian Mixture Model (GMM) [5], [27] and Kernel based Density Estimation (KDE) [6]) have been proposed to solve difficult scenarios, such as dynamic backgrounds, camera jitter, intermittent object motion and so on. Several approaches use a matching mechanism with collected sample and random replacement such as ViBE [13] ans SubSENSE [14]. Other approaches use low-level features like LBSP [14] and LBP histograms [15]. Among these algorithms, there is no algorithm for sudden change background scenarios while sudden changes in background are very common in surveillance video. Sudden changes in background can be easily assumed to be moving objects in all background subtraction algorithms, because distinguishing between sudden changes in background pixels and moving object just from a pixel difference is very difficult. Because, illumination changes are usually the reason why background areas suddenly change, many authors pay more attention on illumination-invariant features. In this paper, we propose a background subtraction method for video sequences with strong background changes (illumination changes, etc...). Our contributions are as follows: First, we propose a background pixel detecting method. Different from previous methods which are based on illumination models [25][26], we do not build an illumination model. Practically, we firstly reconstruct every frame in video sequences by combining several features through an energy function, and then we compute differencing frame between the original frame and the reconstructed frame. In the difference frame, pixels which are above to a predefined threshold are classified as background pixels. In previous approaches, pixels which are above to predefined thresholds are considered as foreground pixels. The rationale is that foreground objects don't change obviously in our differencing frame scheme, meantime background areas change obviously. Through the above step, we can get background masks that show background areas. By using a background mask, we can predefined impossible foreground area. Because, proposed foreground areas predefined step does not use illumination information, videos with background change which aren't caused by illumination will be also effective. The second contribution consists of a contour-based comparing method to incorporate background masks into foreground masks

The research was supported by National Nature Science Foundation of China(61671336,61671332), science and technology program of Shenzhen (JCYJ20150422150029092), Hubei Province Technological Innovation Major Project(2016AAA015), Natural Science Fund of Hubei Province (2015CFB406)

which are obtained with a background subtraction method. The rest of this paper is organized as follows. The proposed method is shown in Section II. In Section III, we analyze the experimental results conducted on the OTCBVS, BMC 2012 and PETS 2001 datasets. Conclusions are discussed in Section IV.

II. THE PROPOSED ALGORITHM

Our enhancing algorithm differentiates from previous background subtraction enhancing works such as [16], [17] in the sense that we enhancing algorithms is based on changing video sequences themselves, while other algorithms enhance detection results through detecting illumination changes. The feature fusion step is shown in Fig.1. Thus, we use different features together to obtain one feature. This fusion of features is robust to figure out foreground pixels. Blocks in the dashed line is a feature fusing process. Note that we convert the color space from RGB to YCbCr, and then we only do fuse in Y channel. In Fig.1, one pixel represents a pixel in one frame. Nearby pixels represent pixels around this pixel. The fusion scheme can be described as follows: First, the initial value is set to the average of three channels. For instance, if the R, G, B channel of one pixel is 100, 125 and 111, respectively. The initial value is 112. We details this fusion process in the following sub-sections. Noted that there is no temporal changes is considered.

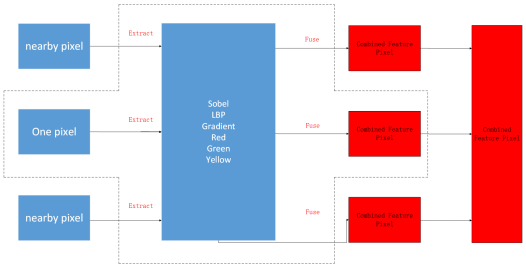

Fig. 1. Muti-Feature Fusion

II-A. Feature Fusion

In this subsection, we describe the proposed fusion scheme based on energy minimization to combine different features together. As features, we choose gradient obtained by the Sobel edge detector, LBP and the three color features RGB. Although, there are many other descriptors like HOG [18], SIFT [19] and Deformable Parts Model [20], background subtraction is a low-level video pre-processing that sophisticated descriptors should be avoided due to high computing time. Furthermore, HOG, SIFT and Deformable Parts Model as high level descriptors which have less connections with the pixel, so using these high level descriptors cannot be directly reacted to the pixel. To fuse these chosen features, they are several prospective approaches such as classical operators (mean, median, etc..), statistical operators or fuzzy operators [29]. For the sake of simplicity and efficiency, we choose an energy minimization approach. Note that the fusion is pixel by pixel. First, we define an energy function to minimize the difference between the original video frame and the reconstructed frame in every feature space:

$$E(S) = \sum_{f \in F} U + \sum_{p \in A} \sum_{f \in F} U \tag{1}$$

f stands for a single feature. F stands for a set of all features. p stands for fused features. A stands for a set of fused features around pixel S. We use 5×5 neighbourhood. U is defined as follows:

$$U = i(x, y) - f(x, y) \tag{2}$$

In this equation, $i(x, y)$ is the fusion pixel value at position at (x, y) in the fusion image. As previously explained, the initial pixel of $i(x, y)$ is the average value of three RGB channels.

Various methods can be considered to solve this energy minimization. For example, hill climbing, simulated annealing, grabcut and so on. Because, hill climbing leads to local optimum and grabcut is very slow, we use a simulated annealing algorithm. The shift value that we used is a Gaussian noise. Its mean value is 0 and its variance is 10 (experimentally determined). We set the start temperature at a very high value. The terminal condition is the lowest energy which do not change more than 10 consecutive iterations. The thresholds are experimentally determined and used in the three datasets. There are three channels in color channels. Before fusing, we convert RGB domain to YCbCr domain. We only fuse these features in Y channel, while we do nothing on other channels(CbCr). After we fuse these features in Y channel, we convert YCbCr domain to RGB domain. The fusion method is shown in pseudocode in Algorithm 1. Note that this pseudocode only shows the process of fusing one pixel. $RAND\_MAX$ is a maximum number of bytes in multi-byte character in the current local. Practically, we repeat this process pixel by pixel all over the image. Fig. 3 shows the original image and the feature combined image in three datasets. Comparing the original image to the feature combined image, we can see that pixel values in background areas are very different and the foreground areas nearly the same. For example, in the first line,buildings are changed heavily. The pixel value in foreground cars don't change much, while background areas changed heavily. In the second line, the intensity of the pixels which correspond to the car doesn't change very much while the intensity of the pixels which correspond to the buildings changes heavily. The third line also shows the same appearance. The objective pixels in reconstructed image are shown in Fig. 2. The left image of Fig. 2 and the right image of Fig. 2 represent the pixel change in the temporal domain for the original video and for the fusion videos, respectively.

Algorithm 1 - Pixel Feature Fusion Algorithm

Require: Initialize the starting $i(x, y)$ is equate to the average of three RGB channels and temperature $currentT = 1000$. $\alpha = 0.2, T\_min = 1, energy\_best = E(i(x, y)), energy\_current = E(i(x, y)), shift = Gaussian(0, 10)$

while $currentT \geq Tmin$ **do**
  while $True$ **do**
    $energy\_next = E(i(x, y) + Gaussian(0, 10))$
    if $next\_energy \leq energy\_current$ **then**
      $energy\_best = energy\_next$
      $i(x, y) = i(x, y) + Gaussian(0, 10)$
    else
      $val = -1 \times (\frac{(energy\_next - energy\_current)}{currentT})$
      $p = exp(val)$
      $randP = random()/RAND\_MAX$
      if $p > randP$ **then**
        $energy_best = energy_next$
        $i(x, y) = i(x, y) + Gaussian(0, 10)$
      end if
    end if
    if $energy\_best! = energy\_next$ **then**
      $count++$
      if $count > 10$ **then**
        Break
      end if
    end if
  end while
  $currentT = currentT \times \alpha$
  Break
end while

For the pixel at the position $x = 132$ and $y = 92$ in the 500^{th} frames, we can observe that this pixel become a foreground pixel. Furthemore, we can see that the change at 500^{th} frame become lower than other frames. By comparing the difference, we extract the foreground pixels.

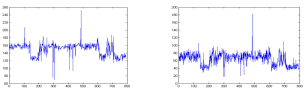

Fig. 2. Pixel change in original videos and fusion videos.

II-B. Fusion Strategy

Fusing many foreground masks together is an efficient method to enhance the final performance of background subtraction as developed in Bianco et al. [17][28]. For this, we propose a contour-based mask combining method to combine the background mask which indicates background pixels and foreground mask which is generated by a background subtraction algorithm. The pseudo-code of this combining step is shown in pseudo-code in Algorithm 2. $cmask$ is a mask with all pixels that are set to 255. $foremask$ is the output of a background subtraction algorithm. $outmask$ is our final output mask. $contours(i)$ is the i^{th} contour in one frame. By detecting contours in $foremask$ and filling holes in every contour, we divide the whole foreground into several parts. The obvious different from previous comparing approaches is that we change comparing method from pixel by pixel to contour by contour. The value of threshold is experimentally set to 20 and this threshold was used for the evaluation on three datasets.

Fig. 3. Original frame and feature combined frame. The first line shows the results on one frame of the OTCBVS dataset. The second line and the third line present the results on one frame of BMC and PETS 2001 datasets, respectively.

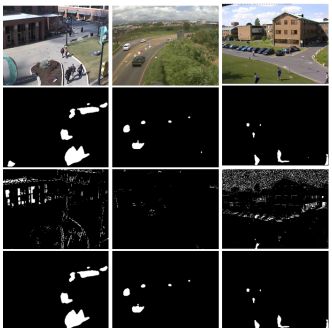

Fig. 4. The first row shows the original image in the three datatests. The second row presents the results of SubSENSE [14]. The third row shows the differencing mask and the last row is our final results.

III. EXPERIMENTAL RESULTS

We use three datasets to evaluate the proposed method: OTCBVS dataset[1] [22], PETS 2001 dataset[2] [23], and the BMC 2012 dataset [3] [24] (Video_008.avi). All the videos are taken under outdoor environments with dynamic illumination changes. More specifically, the PETS 2001 dataset contains gradual illumination changes while the lighting condition quickly alters by passing clouds across the buildings in OTCBVS dataset. The F-measure is adopted for the performance evaluation. The F-measure is shown below. $F =$

[1]http://vcipl-okstate.org/pbvs/bench/
[2]http://www.cvg.reading.ac.uk/slides/pets.html
[3]http://bmc.iut-auvergne.com/

Table I. Performance evaluation for moving objects extraction over three datasets

Datasets	OTCBVS			PETS 2001			BMC 2012		
Methods	Recall	Precision	F-score	Recall	Precision	F-score	Recall	Precision	F-score
GMM [8]	0.62	0.60	0.58	0.79	0.38	0.49	0.56	0.52	0.54
GMMour	0.77	0.51	**0.61**	0.79	0.39	**0.50**	0.57	0.52	**0.55**
SuBSENSE [14]	0.63	0.76	0.68	0.63	0.60	0.67	0.5682	0.5578	0.5630
SuBSENSEour	0.58	0.89	**0.69**	0.60	0.79	**0.68**	0.5683	0.5582	**0.5632**

Table II. Compare to other background subtraction methods

Methods	Recur.MOG[8]	Ensem.MOG[7]	Phase-based[9]	Kernel score[10]	Kernel density[11]	Low rank [12]	IISC[21]	Our method
F(PETS2001)	0.49	0.27	0.18	0.36	0.47	0.69	0.50	0.68
F(OTCBVS)	0.58	0.28	0.38	0.09	0.18	0.17	0.52	0.69

Algorithm 2 - Foreground Mask Fusion

Require: $diff,\ foremask,\ i = 0,\ cmask$
  set threshold $T = 20$
  binarize $diff$ to $foremask$ using T
  detect contour in $foremask$
  fill all contours in $foremask$
  count the number of contours and store the number into N
  while $i < N$ **do**
    $ddmask(i) = contours(i) \cdot foremask$
    count non-zero in $ddmask(i)$ and store the number into z
    if $z = 0$ **then**
      $cmask(contours(i)) = 0$
    end if
    i=i+1
  end while
  $outmask = foremask \cdot cmask$

$2 \times Recall \times Precision/(Recall + Precision)$. To evaluate the efficiency of the proposed method, the whole comparing process is divided into two parts. First, we select GMM [8] and SuBSENSE [14] as original background subtraction algorithms in Algorithm 1. Then, we compare our proposed method to the original GMM [8] and SuBSENSE [14], respectively. In GMM [8], we adjust the parameters of GMM to assume the best performance. Specifically speaking, the history of previous frames is 100 and shadows detecting is turned off. Second, we compare our method with six state of-the art background subtraction algorithms. Examples of the foreground detection results are presented in Fig.4. The three lines of these two figures show a frame of a video from the OTCBVS dataset, a frame of a video from the BMC 2012 dataset, a frame of a video from the PETS 2001 dataset, respectively. Compare to original detection results, the proposed method improves through taking out impossible foreground areas. Due to the limitation of pages and because the difference between two masks is small, we do not show the difference mask between the two foreground masks. Moreover, it is clear that the proposed algorithm considerably outperforms original GMM and original SuBSENSE, particularly for SuBSENSE. From the Table I, we can see that our method improves the performance of the original methods in term of F-score. We also compared our approach with six state-of-art methods: recursive MoG [8], ensemble MoG [7], phase-based[9], variable kernel score-based [10], kernel density-based [11], DECOLOR [12] and IISC [21]. Table II shows the quantitative comparison on PETS 2001 and OTCBVS datasets by using F-measure. Thus, we can see that our method is the top of rank over the two datasets. Thus, the proposed method is effective for background subtraction under background change scenarios. The processing time of our method which concerns the fusion time of fusing one frame with frame size of 320×240 is nearly 0.5 second with a computer Xeon(R) CPU 4 E31220 3.10GHz and 16G.

IV. CONCLUSION

In this paper, we propose a background subtraction improving method that utilizes multi-feature fusion. The key idea of the proposed method is that we use the difference between original frames and multi-feature frames to predefine foreground objects. Experimental results show the interest of the proposed approach. However, the actual method computes relatively slowly due to that we fuse multi-feature pixel by pixel, further work will concern feature fusion areas by areas to reach real-time requirement.

V. REFERENCES

[1] J. Xiao, R. Hu, L. Liao, Y. Chen, Z. Wang and Z. Xiong, " Knowledge-based Coding of Objects for Multi-source Surveillance Video Data." in IEEE Transactions on Multimedia, vol. 18, no. 9, pp. 1691-1706, 2016.

[2] J. Xiao, Z. Wang, Y. Chen, Y. Chen, L. Liao, J. Xiao, G. Zhan and R. Hu, " A Sensitive Object-Oriented Approach to Big Surveillance Data Compression for Social

Security Applications in Smart City." Software: Practice and Experience, 2016.

[3] A. Li and S. Yan, "Object Tracking With Only Background Cues", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 24, No. 11, pp. 1911-1919, 2014.

[4] T. Huynh-The, O. Banos, S. Lee, B. Kang, E. Kim and T. Le-Tien, "NIC: A Robust Background Extraction Algorithm for Foreground Detection in Dynamic Scenes", IEEE Transactions on Circuits and Systems for Video Technology, 2016.

[5] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.8, pp.747-757, Aug 2000.

[6] A. Elgammal, D. Harwood, and L. Davis, Non-parametric model for background subtraction, in European Conf. Comput. Vision (ECCV), Vol. 1843 of Lecture Notes in Comp. Science, pp. 751C767, Springer, June 2000.

[7] B. Klare and S. Sarkar, Background subtraction in varying illumina- tions using an ensemble based on an enlarged feature set, in Proc. IEEE Comput. Vis. Pattern Recog. Workshops (CVPRW), Jun. 2009, pp. 66-73.

[8] Z. Zivkovic and F. V. D. Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern Recognit. Lett., Vol. 27, No. 7, pp. 773-780, Jul. 2006.

[9] G. Xue, J. Sun, and L. Song, Background subtraction based on phase and distance transform under sudden illumination change, in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2010, pp. 3465-3468.

[10] M. Narayana, A. Hanson, and E. L. Miller, Background modeling using adaptive pixelwise kernel variances in a hybrid feature space, in Proc. IEEE Comput. Vis. Pattern Recog. (CVPR), Jun. 2012, pp. 2104-2111.

[11] M. Narayana, A. Hanson, and E. L. Miller, Improvements in joint domain-rangemodelingforbackgroundsubtraction,inProc.Brit.Mach. Vis. Conf. (BMVC), Sep. 2012, pp. 1-11.

[12] X. Zhou, C. Yang, and W. Yu, Moving object detection by detecting contiguous outliers in the low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 35, No. 3, pp. 597-610, Mar. 2013.

[13] O. Barnich and M. Van Droogenbroeck, ViBe: A universal background subtraction algorithm for video sequences, IEEE Trans. Image Process., Vol. 20, pp. 1709-1724, June 2011.

[14] P. St-Charles, G. Bilodeau, and R. Bergevin, SuBSENSE: A universal change detection method with local adaptive sensitivity, IEEE Trans. Image Process., vol. 24, pp. 359-373, Jan. 2015.

[15] M. Heikkila and M. Pietikainen, A texture-based method for modeling the background and detecting moving objects, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 28, pp. 657-662, April 2006.

[16] A. Schick, M. Buml, R. Stiefelhagen "Improving Foreground Segmentations with Probabilistic Superpixel Markov Random Fields", IEEE Workshop on Change Detection, 2012

[17] S. Bianco, G. Ciocca, and R. Schettini, How far can you get by combining change detection algorithms?, CoRR, vol. abs/1505.02921, 2015

[18] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in:Proc. IEEE Conf. Comput. Vis. Pattern Recogn., 2005, pp. 886-893.

[19] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. J.Comput. Vis. 60 (2004) 91-110.

[20] P. Felzenszwalb et al., Object detection with discriminatively trained part based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1627-1645.

[21] W. Kim and Y. Kim, "Background Subtraction Using Illumination-Invariant Structural Complexity," in IEEE Signal Processing Letters, vol. 23, no. 5, pp. 634-638, May 2016.

[22] J. Davis and V. Sharma, Background-subtraction using contour-based fusion of thermal and visible imagery, Computer Vision and Image Understanding, Volume 106, Issues 2C3, MayCJune 2007, Pages 162-182, http://www.cse.ohio-state.edu/otcbvs-bench

[23] Performance evaluation of tracking and surveillance dataset 2001, http://ftp.pets.rdg.ac.uk/pub/PETS2001

[24] A. Vacavant, T. Chateau, A. Wilhelm and L. Lequievre. A Benchmark Dataset for Foreground/Background Extraction. In ACCV 2012, Workshop: Background Models Challenge. November 2012, Daejeon, Korea. http://bmc.univ-bpclermont.fr/

[25] J. Pilet, C. Strecha and P. Fua. Making Background Subtraction Robust to Sudden Illumination Changes, European Conference on Computer Vision, ECCV 2008, October 2008.

[26] J. Wang and Y. Yagi. Efficient Background Subtraction under Abrupt Illumination Variations, Asian Conference on Computer Vision, ACCV 2012, November 2012.

[27] T. Bouwmans, F. El Baf and B. Vachon, Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey, Recent Patents on Computer Science, RPCS 2008, Vol. 1. No. 3, pp. 219-237, November 2008.

[28] T. Bouwmans, Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview. Computer Science Review, Vol. 11, pp. 31-66, May 2014.

[29] V. Balopoulos, A. Hatzimichailidis and B. Papadopoulos. Distance and similarity measures for fuzzy operators, Information Sciences, Vol. 177, No. 11, pp. 2336-2348, June 2007.