Improving Segmentation Accuracy for Motion Detection using Bidirectional Video Processing

Abhishek Thakur, Ankesh Kumar
CSIS department,
BITS Pilani Hyderabad Campus
Hyderabad, India
thakur@ieee.org, ankeshkumar@outlook.in

Abstract— **Most of content in video capture from static cameras consist of fixed background. Invariably the important information in such video is associated with some motion. Background Subtraction models help extract such information. Adaptive approaches of background subtraction (like Mixture of Gaussian) do not mandate prior input for object detection. But adaptive approaches suffer from misclassification during learning phase. This work extends Mixture of Gaussian to improve upon the short comings of adaptation phase. Video chunks are processed in both forward and reverse direction. Different approaches of combining the output values in both directions are analyzed. Best results are observed when different kernel sizes are used in morphological operations for forward and reverse direction, depending on the temporal offset. Experimental results for standard video data sets and farm like scenarios demonstrate improvement in detection accuracy across numerous frames.**

Keywords- ***Multimedia, Video Processing, Motion Detection, Background Subtraction, Bidirectional Processing***

I. INTRODUCTION

This work forms a significant part of object detection, highlight generation and video compression of in farm like scenarios. Large areas are monitored using cameras mounted at strategic locations. These cameras are affordable Android devices communicating over 802.11 series of protocols. In rural scenarios, actual network connectivity should be delay and disruption tolerant [1]. Continuous video is not streamed on such networks. Video must be heavily compressed to be shared over such networks within reasonable timeframe.

For such heavy compression significant portion of the video must be discarded. The challenge is to accurately segment the regions are to be discarded, from the regions to be retained. It is obvious that for such scenarios, motion is associated with portion of video to be retained. This work improves segmentation accuracy by improving motion identification.

Typical captures, for farm monitoring, involve large open areas with trees and other vegetation. The intent is to detect human / animal or inanimate objects intruding into the farm without requiring explicit human interaction. Broader work involves highlight creation and messaging using Delay Tolerant Networks [1]. Videos are stored and processed in chunks varying from tens of seconds to a few minutes, to extract key information. It is only the key extracted information that is sent over the delay tolerant network.

Accuracy can be improved by identifying regions of significant motion while discarding small movements generated by the fixed objects. For example swaying branches, tree-tops etc. show lots of small movements and do not have significance. It is assumed that the natures of objects are not known. A priori knowledge based approach could cause unknown objects to be missed. Though the wider work involves object tracking across frames, path-identification and communication mechanisms, they are not within the scope of this work. This paper focusses only on improving segmentation accuracy.

While this paper discusses farm like scenarios as primary use case, it can be applied to any scenario where video is captured using static cameras. Such applications will have impact on information extraction from archived videos, e.g. conserving storage space for archived videos.

Layout of the document: Next section provides an overview of key object detection techniques and includes an overview of Background subtraction using Gaussian Mixture Model. Section III provides the implementation details of bi-directional video processing for object detection. Section IV covers the experiment details, and metrics. Section V presents the results and related analysis. Section VI concludes the paper including a discussion of future work.

II. BACKGROUND WORK

Multiple approaches have been developed to detect motion of objects in the video. Three of the common approaches are a) Optical Flow; b) Background subtraction; and c) Color histogram based methods [2]. For brevity of this paper only background subtraction method is discussed in details. The other two approaches need prior information (histogram based) or do not scale well for large areas (optical flow).

Motion detection using background subtraction involves segmenting each video frame into background and foreground regions. The background region stays relatively static throughout the length of the video. The foreground is the region of pixels that moves across the frames while mostly maintaining its shape and color attributes. Multiple approaches have been proposed for motion detection. This work builds

upon background subtraction based object detection [3] [4]. The background subtraction approaches performs well with modern systems that have more graphical processing cores dedicated to the algorithm [5].

A. Background subtraction - Mixture of Gaussian: A brief Overview

Background subtraction methods rely on creating a model for the background. Each frame is compared with the background model to find the difference and identify regions of motion. Background subtraction Mixture of Gaussian (referred to as BGS-MOG in subsequent parts) differs from the other background subtraction methods in the way the background model is created. The background model is continuously updated for every frame that is processed. Every pixel of the background is represented as a mixture of Gaussian distributions (mean, variance and weight) [6]. This approach allows efficient handling of gradual illumination changes (e.g. shadows, clouds etc.) as well as adapts to swaying trees and branches. BGS-MOG does not require bootstrapping (externally identifying the object) or prior learning (training). It does suffer from camouflage issues (especially for slow moving objects) and does not work if camera view is not static.

For each pixel in a new frame, the pixel value is compared to the corresponding Gaussian distributions in the background model. If a match is found then the corresponding distribution is updated for mean, variance and the weight. If no match is found a new distribution is added to the mixture (if number of distributions is exceeding the maximum allowed entries then the least probable distribution is replaced by a new distribution). This new distribution is allocated the default weights and variance. All weights are normalized following updates/additions and the mixture model is sorted on descending order for weights. A learning rate (α) drives the weights applied to updates and additions.

A Gaussian mixture model is a more accurate description of the background as compared to any single distribution. The adaptive nature of the algorithm accounts for lighting changes and works even if no pristine background model was available.

If a pixel does not match, or matches against a very low weight in the mixture, it is marked as foreground in resulting mask. The mask provided by BGS MOG for a frame is generated on individual pixels without a sense of adjacency. This results in a perforated mask with varying pixel values for a given object. Morphological operations are used to smoothen out the mask.

Morphological operations are matrix based mathematical operation on images. These operations alter a pixel's value based on the values of the pixels in its neighborhood. The area and shape of the neighborhood affects the results of the operation. The operations used in this paper are erode and dilate. The erode operation reduces the thickness (shrinks) around the edges whereas dilate increases the thickness. If random small objects are present, erode tends to remove them while dilate will end up joining them. Size and shape of kernel decides how effective the removal or joining is.

B. Shortcoming of BGS-MOG

1) Spurious motion in initial frames:

The initial frames show a lot of motion since the background model is still under construction. The learning rate (α) drives the weight for pixel value in current frame. If moving objects or swinging branches are present in the first frame, the algorithm incorrectly classifies the initial region where the moving object was present, as part of the background. In subsequent frames, a lot of incorrect motion is reported where the moving body was originally present. The inaccurate motion is removed after a few frames, as the background model adjusts to the real values for background pixels. The adjustment period depends upon the learning rate α.

2) Missed trailing edges:

Trailing edges of some objects are missed, if they are narrow along their motion path. This is because mixture model has not adapted to the object.

It's difficult to ensure that initial frame does not have any motion; hence the initial frames will have false positives. To reduce this learning period, a very high learning rate (α) can be used. However a high α causes slower moving objects to be mislabeled as background. This makes tracking slower moving objects difficult. Generally a compromise is made on the learning rate and some inaccuracy in reporting motion in the first few frames is accepted. This work targets to reduce such inaccuracy.

III. BIDIRECTIONAL PROCESSING

Bi-directional processing deals with chunks of video of 30 seconds to a few minutes. The stream is individually analysed from starting to end and from end to start. The results obtained from the two analyses are then aggregated to generate a mask that that had improvements in identifying the region of motion.

The first step in the process is to create a separate reversed video of the given video chunk (last frame to first frame).

Then BGS MOG is applied on both video chunks. The result is two separate videos of masks. One is for the first-to-last frame video (henceforth called the forward stream) and the other for the last-to-first frame. The BGS MOG stream for the reversed video frame has the masks in last to first order. The two streams need to be aligned so that for a given frame number both streams provide the mask for the same frame. This is done by reversing the mask stream for the last to first frames (subsequently referred to as the reversed stream).

The two streams can be used individually to detect motion in a given frame. However both suffer from the shortcomings mentioned in II-B.

In order to improve detection accuracy, the two streams were then aggregated. Three aggregation methods were experimented upon. In all the aggregation methods dilation and erosion have been used. Three different sized square kernels were used. SMALL(3x3), MEDIUM(5x5) and LARGE(7x7).

The initial aggregation methods used were Bitwise OR and Bitwise AND. Prior to the Bitwise operation the forward and reverse stream the frames are dilated using the MEDIUM sized kernel. After the Bitwise operation the resulting mask is eroded

using LARGE kernel. This is done to minimize the addition of noise from the individual streams. This may result in loss of information. Mathematical addition and multiplication were not used since the masks are binary in nature and bitwise operations are faster.

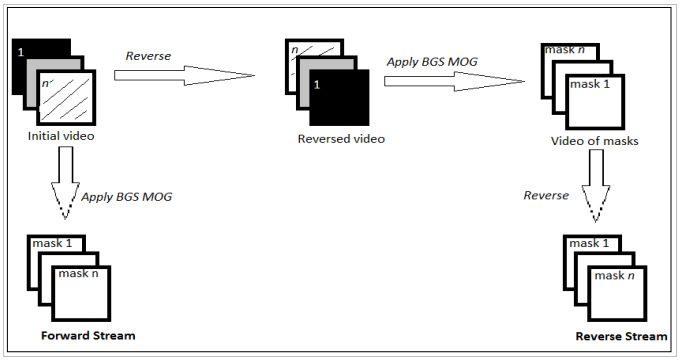


Figure 1 Creation of reverse and forward mask streams

The accuracy of the various aggregate streams was then compared to the accuracy of individual forward and reverse streams. For comparison, the individual forward and reverse streams were dilated and eroded with the same sized kernel (MEDIUM), in order to smoothen out the edges and fill in small gaps within the shapes. Figure 1 captures the processing done for individual streams (forward, reverse) and aggregated streams (AND, OR).

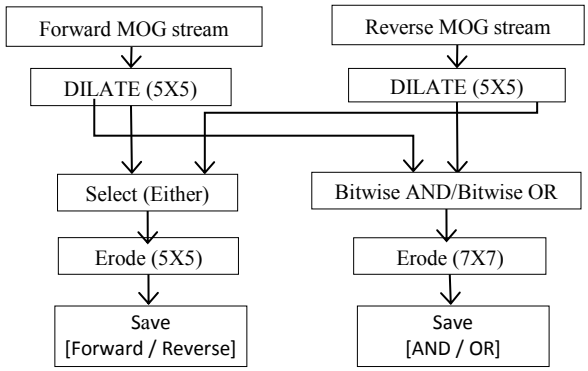


Figure 2 – Morphological operations for individual and simple aggregated streams

It was expected that the Bitwise OR aggregation method would lead to additional noise in the mask. This effect would be pronounced in the initial and terminal part of the video. Similarly it was expected that the Bitwise AND aggregation method would lead to loss of information.

To overcome these issues, a third aggregation approach based on Weighted Bitwise OR was implemented. For this method the masks from forward stream were preferred during the terminal part of the video and the masks from the reverse stream were preferred during the initial part of the video. The approach used morphological operations as weight functions for individual streams. Different sized kernels were used as function weights. The algorithm is captured below.

Algorithm 1: Weighted Bitwise OR

Initialization
    SMALL kernel = matrix of size (3x3) with all 1
    MEDIUM kernel = matrix of size (5x5) with all 1
    LARGE kernel = matrix of size (7x7) with all 1
    Acquire *forward_MOG_stream*
    Acquire *reverse_MOG_stream*
    Initialize *output_stream*
for *all frames* **in** *video_stream* **do**
    acquire *forward_mask*
    acquire *reverse_mask*
    if *frame* **in (***0 to 5% of video_stream***) then**
        mask1 =Dilate(*forward_mask*, *SMALL kernel*)
        mask2 =Dilate(*reverse_mask*, *LARGE kernel*)
    else if *frame* **in (***95% to 100% of video_stream***) then**
        mask1 =Dilate(*forward_mask*, *LARGE kernel*)
        mask2 =Dilate(*reverse_mask*, *SMALL kernel*)
    else
        mask1 =Dilate(*forward_mask*, *MEDIUM kernel*)
        mask2 =Dilate(*reverse_mask*, *MEDIUM kernel*)
    end
    combined_mask = BitwiseOr(*mask1, mask2*)
    final_mask = Erode(*combined_mask, LARGE kernel*)
    output_stream.add(*final_mask*)
end

For first 5% of the video forward stream was in learning phase. Similarly for last 5% of the video reverse stream was in learning phase. The stream that is not in learning phase was preferred in such scenarios. The preferred stream was dilated with the LARGE kernel and the other mask (corresponding less preferred) was dilated with the SMALL kernel. After the Bitwise OR operation, the resulting mask was eroded using a LAGE kernel. During the middle portion of the video (5%-95%) both the forward and the reverse stream are given equal preference and dilated by the MEDIUM kernel.

In the first 5%, since the reverse MOG stream the model is mostly accurate, hence it has better representation of actual motion. A Bitwise OR of the two dilated masks creates an aggregate that has dilated objects from both the streams, i.e. highly expanded regions from the reverse MOG and minimally expanded regions from the forward MOG stream. Subsequent erosion of the aggregate mask using the LARGE kernel diminishes the regions of motion that are only contributed by the forward MOG (as it was dilated by a smaller kernel). Most of the small random regions, contributed by forward MOG, are removed and larger regions, contributed by forward MOG, are diminished in area. For regions contributed by the reverse MOG stream, the factor of dilation is the same as the factor of erosion; hence no detail is lost for them.

Similar logic used in for frames in last 5% of the video ensures that motions identified in forward stream are prominent. For the remaining part (i.e. in the middle), both the streams are dilated by the same factor. A kernel of MEDIUM dimensions is used on both the forward and reverse MOG streams.

IV. EXPERIMENTAL SETUP

In experimental runs, Background Subtraction Model with Mixture of Gaussian is used. There were 20 mixture channels and learning rate (α) was set as 0.005. For the morphological operations square kernels were used with dimensions as 3x3, 5x5 and 7x7 for SMALL, MEDIUM and LARGE respectively.

The algorithm was implemented using OpenCV (2.4.9). "Improved adaptive Gaussian mixture model for background subtraction" implementation was used for all the experiments [7].

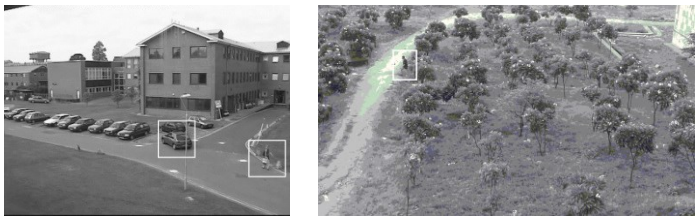
A. Test Videos

Two video sources were used.

- Video from within the authors' campus was chosen such that it presented challenges that were similar to the farm monitoring scenario. (From here on referred to as internal video). The video was shot in HD Ready resolution at 30 fps. Due to the large distance, the common moving objects were relatively small. The paths are such that objects are occluded by tree and shrubs. The trees and shrubs have moderate swaying motion.
- Clip from PETS 2001 video -Camera 1; from here on referred to as the PETS video [Reference: PETS2001 datasets (University of Reading, UK)]. It has motion of people and vehicles at 25 fps.

Across the video sources, series of representative frames were chosen. Each frame series consists of four individual frames (e.g. frame numbers 5, 10, 15, 20). Frame series were chosen from start, middle and end of video. The actual region of motion (Ground Truth) for the representative frames was identified manually.

For each video source, chunks of 30 seconds were used in experimental run. Sample reference frames are captured in Figure 3. The ground truth for these frames is shown in and Figure 4. Similar work was repeated for all the twenty four frames (3 series of four frames each, for both videos).



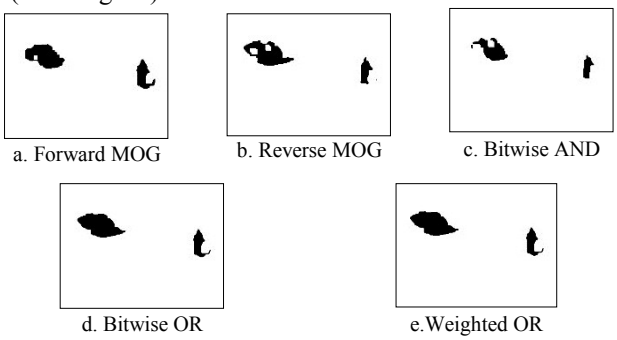
a. PETS - frame 400 b. Internal - frame 400

Figure 3 Sample Video frames

a. PETS video b. Internal Video

Figure 4 Ground truth video frame 400

For each video the metrics were calculated for forward BGS MOG, reversed BGS MOG, Bitwise OR , Bitwise AND , and Weighted OR. The representative masks for frame 400 in PETS video is shown in each frame series are shown in Figure 5 (a. through e.).



a. Forward MOG b. Reverse MOG c. Bitwise AND

d. Bitwise OR e.Weighted OR

Figure 5 Resulting motion masks for frame 400 zoomed 4x to bottom right

B. Metrics For evaluation

Accuracy of motion detection was measured by three metrics. These were 1) Missing Foreground, 2) Added Foreground and 3) Rate of Misclassification [8].

Missing foreground refers to the percentage of foreground pixels mislabeled as background pixel. It's calculated for each representative frame as

$$MF_{rel} = (\text{Missing foreground pixels in the mask}) / (\text{Number of foreground pixels in the Ground truth}) \quad (1)$$

Added foreground is a measure to determine the degree to which background pixels are mislabeled as foreground pixels.

$$AF_{rel} = (\text{Added foreground pixels in the mask}) / (\text{Number of Background pixels in the Ground Truth}) \quad (2)$$

MF_{rel} and AF_{rel} do not account for the degree of misclassification for a missed foreground pixel or added background pixel. A falsely detected pixel that is closer to the actual foreground is better as compared to a farther pixel. This measure is provided by the metric Rate of Misclassification. It's given as:

$$Fsc(n) = \frac{1}{F(n)} \sum_{j=1}^{F(n)} \left(\frac{d_{Err}^{j}}{D} \right) \quad (3)$$

Where F(n) contains set of pixels in MF ∪ AF. d^{j}_{err} is the distance of the j^{th} misclassified pixel from the closest contour of the ground truth. D is the length of the frame diagonal.

V. RESULTS AND OBSERVATIONS

Table 1 captures the values for AF_{rel}, MF_{rel} and F_{SCrel}, across the three frame series for both video sets.

MF_{rel}: Bitwise AND is the worst performer as it missed the foreground objects consistently. This is visible in Figure 5 (c) where large part of the car is missed.

AF_{rel}: Bitwise AND, shows only those regions that are

common to both the forward and reverse MOG streams. Since most noise generated by the two streams do not overlap therefore a very low AF_{rel} value is obtained. In the initial frame series reverse streams has the lowest AF_{rel} followed closely by Weighted Or approach. Correspondingly for the end frame series (of the PETS video) the reverse stream has the lowest values followed very closely by Weighted OR.

F_{SCrel}: It measures the distance of the misclassified pixel (missing foreground and added background) from the nearest contour. Bitwise AND has the best result in all cases followed closely by the remaining methods. The reason for high F_{SCrel} for Bitwise AND is due to the missing foreground pixels. Since the contours are small this effect is not prominent.

TABLE 1 RESULTS FOR INTERNAL VIDEO (MULTIPLIED BY 1000)

			Forward	Reverse	Bitwise OR	Bitwise AND	Weighted OR
Internal Video	Series 10	MF_{rel}	0	0	186	757	36
		AF_{rel}	17.56	0.25	0.14	0	0.07
		F_{SCrel}	298.02	1.98	11.47	1.11	1.65
	Series 400	MF_{rel}	3	15	265	480	265
		AF_{rel}	1.58	1.13	0.23	0.04	0.23
		F_{SCrel}	7.9	8.62	8.4	9.33	8.4
	Series 800	MF_{rel}	152	152	411	508	197
		AF_{rel}	1.02	3.66	0.41	0.07	0.43
		F_{SCrel}	9.63	86.66	21.75	3.02	17.42
PETS Video	Series 10	MF_{rel}	140	3	45	207	6
		AF_{rel}	1.62	1.16	2.64	0.36	1.60
		F_{SCrel}	29.77	3.25	51.60	2.81	13.52
	Series 400	MF_{rel}	36	138	36	271	36
		AF_{rel}	5.44	5.95	5.85	2.83	5.85
		F_{SCrel}	165.25	179.13	179.37	166.45	179.37
	Series 700	MF_{rel}	197	156	24	449	24
		AF_{rel}	10.78	13.08	13.76	4.48	13.76
		F_{SCrel}	269.28	289.20	294.46	260.27	294.46

We conclude from the above table that

- Forward and reverse MOG streams do not perform well in the initial and the final regions of the video respectively. Otherwise they have good performance (low AF_{rel} and F_{SCrel}) and moderate MF_{rel}.
- Combination approaches (both AND and OR) use larger dilate kernel hence they have lower MF_{rel} during the middle part of video. OR is better than AND.
- Near start / end of video chunks, either of forward or reverse are best, while Weighted-OR is close second (sometimes even tying for best performance).

When comparing the three aggregation approaches, Weighted OR algorithm effectively removed the noise while keeping good values for missed foreground across all parts of video.

VI. CONCLUSION AND FUTURE WORK

Efficient motion detection techniques enable creation of accurate tracking solutions. The proposed method provides a novel way to improve motion detection. The improvement comes at the cost of additional processing required to revere the video stream and analyzed the reversed video as well as the original video. For internal video dataset, the compressed output is shared at www.swifiic.in/sambv/index.html. Fourteen megabytes was compressed to below 600 kilobytes.

Experiments with bidirectional video processing showed excellent results for Weighted OR. It improved segmentation efficiency while keeping the false positives within the bounds. Authors' intend to extend this work to study the effects of different learning rates, kernel shapes and kernel sizes across different video datasets. Further the team intends to apply bidirectional approaches to other object detection and tracking approaches that do not expect static camera view (E.g. [9][10]).

Such advances can be put to use in the fields of surveillance, video compression, event generation etc. The method is not real time but is implemented in close to real time when using chunks of video. The authors are targeting to get the system to perform in real time when working in parallel on samples of 30 seconds to a minute.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers whose feedback has helped improve the quality of this work.

VII. REFERENCES

[1] Fall, K. (2003, August). A delay-tolerant network architecture for challenged internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (pp. 27-34). ACM.

[2] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey." *Acm computing surveys (CSUR)* 38, no. 4 (2006): 13.

[3] Bouwmans, Thierry. "Traditional and recent approaches in background modeling for foreground detection: An overview." Computer Science Review 11 (2014): 31-66.

[4] Sobral, Andrews, and Antoine Vacavant. "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos." Computer Vision and Image Understanding 122 (2014): 4-21.

[5] Bauer, Sebastian, Sebastian Kohler, Konrad Doll, and Ulrich Brunsmann. "FPGA-GPU architecture for kernel SVM pedestrian detection." In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 61-68. IEEE, 2010.

[6] Friedman, Nir, and Stuart Russell. "Image segmentation in video sequences: A probabilistic approach." In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pp. 175-181. Morgan Kaufmann Publishers Inc., 1997.

[7] Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, pp. 28-31. IEEE, 2004.

[8] Schlögl, Thomas, Csaba Beleznai, Martin Winter, and Horst Bischof. "Performance Evaluation Metrics for Motion Detection and Tracking." In *ICPR (4)*, pp. 519-522. 2004.

[9] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-learning-detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.7 (2012): 1409-1422.

[10] Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie. "Robust object tracking with online multiple instance learning." Pattern Analysis and Machine Intelligence, IEEE Transactions on 33.8 (2011): 1619-1632.