Efficient Background Subtraction for Tracking in Embedded Camera Networks

Yiran Shen*#, Wen Hu*, Mingrui Yang*, Junbin Liu*,Chun Tung Chou# ,Bo Wei*#

* *CSIRO ICT centre, Australia*
{firstname.lastname}@csiro.au
School of Computer Science and Engineering, UNSW, Australia
{firstname.lastname}@unsw.edu.au

ABSTRACT

Background subtraction is often the first step in many computer vision applications such as object localisation and tracking. It aims to segment out moving parts of a scene that represents object of interests. In the field of computer vision, researchers have dedicated their efforts to improving the robustness and accuracy of such segmentations but most of their methods are computationally intensive, making them non-viable options for our targeted embedded camera platform whose energy and processing power is significantly more constrained. To address this problem as well as maintain an acceptable level of performance, we introduce Compressive Sensing (CS) to the widely used Mixture of Gaussian to create a new background subtraction method. The results show that our method not only can decrease the computation significantly (a factor of 7 in a DSP setting) but remains comparably accurate.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Systems

General Terms

Algorithm, Experimentation, Performance

Keywords

Compressive Sensing, Mixture of Gaussian, Background Subtraction, Object Tracking

1. INTRODUCTION

Many recent real-time object tracking methods require significant computation and energy consumption. These methods are difficult to implement on embedded camera networks because of the resource consumption. Robust background subtraction is typically the dominant factor, such as Mixture of Gaussian (MoG)[2]. The MoG models every pixel with a mixture of $3 \sim 5$ Gaussian distributions for the complex background and a pixel will be marked as foreground if it fits none of these background Gaussian models.

We propose a new background subtraction method that preserves the robustness of MoG meanwhile dramatically decreases the computation. We introduce Compressive Sensing (CS) in MoG to build a computation efficient background subtraction method by reducing data dimensions (termed CS-MoG).

The process of CS-MoG can be divided into three steps: In the first step, it segments the image into small blocks (e.g., 8×8 pixels) then it produces random projection vectors (e.g., 8 projections) for each image block, as

$$Y = \Phi X \tag{1}$$

where, X is from vectorising one of the 64-pixel blocks by row; Φ is ± 1 Bernoulli matrix (e.g., 8×64); Y is the projection vector whose dimension is significantly less than the block size (e.g., 1/8). In the second step, CS-MoG models every projection in Y as a MoG. For every projection CS-MoG will check if its value belongs to the recent background model. Finally, to make a final decision, we design a higher level fusion strategy to determine if this block contains foreground. We evaluate the decision strategy such as max, min and majority voting and we find that the best method is min voting.

2. RESULTS

In this section we firstly compare CS-MoG with an existing method based on CS for background subtraction [1] (termed CSBS). CSBS employs direct comparison between

(a) (b)

Figure 1: a) is a sample image of our dataset (b) is background subtracted image with 8-projections CS-MoG

consecutive images to subtract background. We evaluate the performance of proposed CS-MoG with CSBS in an outdoor pedestrian tracking dataset which consists of 400 images (resolution is 320×240), and Fig. 1(a) is one example image. Fig.1(b) is the result of CS-MoG for Fig.1(a). The number of projections is 1/8 of image size. Fig. 2 shows the performance results with different parameters settings. False

IPSN'12, April 16–20
ACM 978-1-4503-1227-1/12/04.

Alarm (FA) is the probability that we mistakenly decide the background as foreground and Probability of Detection (PD) is probability of correct detection for the foreground. Fig. 2 shows that CS-MoG outperforms CSBS significantly in term of subtraction accuracy. For example, CS-MoG achieves more than 98% PD when the FA is below 3%. Meanwhile CSBS only achieve about 75% of PD with the same FA.

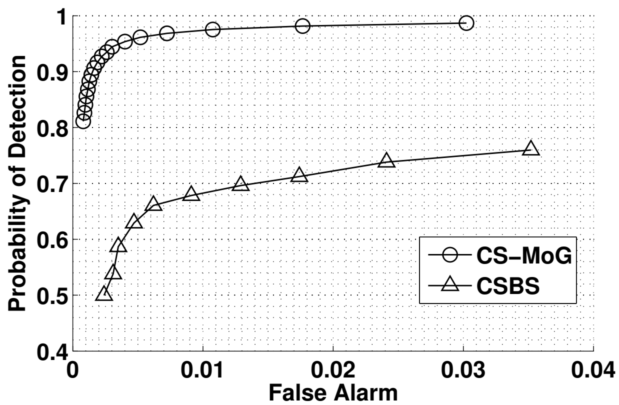


Figure 2: Performance comparison between CS-MoG with CSBS

We also compare CS-MoG with two other efficient MoG subtraction method using the same dataset. The results are shown in Fig. 3. Mean-MoG is to further divide the 8×8 blocks into smaller sub-blocks and calculate the mean of pixel values to represent a sub-block. And then mean values are modelled as MoG. RS-MoG is to randomly choose parts of the pixels to represent the whole block. They all have same dimension reduction ratio as CS-MoG. Fig.3(a) shows the performance of CS-MoG with different number of projections for one block. It demonstrates that when the number of projections increases above 8, CS-MoG will not obtain significant performance gain. Fig.3(b) shows that performance of CS-MoG is the best among efficient methods and very close to original MoG method. All efficient MoG produce 8 measurements for one block.

To evaluate its efficiency, we implement the CS-MoG with 1/8 dimension reduction and the original MoG in a Blackfin BF-537 DSP camera node (see Fig.4). The results show that with original MoG it takes approximately 250 $\sim$ 280ms for the camera node to process one 320×240 image meanwhile with CS-MoG it takes approximately 37 $\sim$ 42ms only. Therefore, CS-MoG is approximately 7 times faster than original MoG and makes it suitable for real-time tracking applications in embedded camera networks.

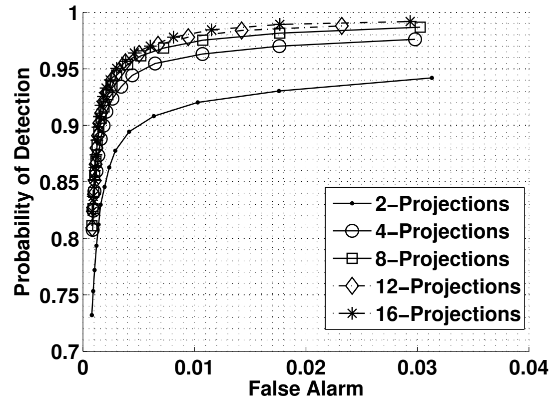
3. FUTURE WORK

we plan to further process the blocks containing both background and foreground in order to directly obtain the pixel scale results from compressed projections. Moreover, we plan to evaluate proposed CS-MoG with more datasets before implementing the whole tracking system in an embedded camera network.

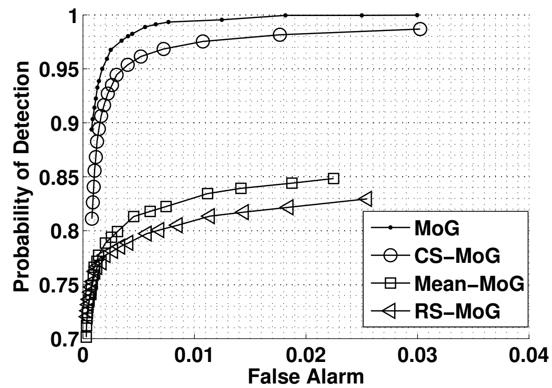
4. REFERENCES

[1] Cevher, Volkan, and etc. Compressive sensing for background subtraction. In *Computer Vision ECCV 2008.*, pages 155–168, 2008.

[2] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999.*, volume 15, pages 246–252, 1999.



(a)



(b)

Figure 3: a) is the results of CS-MoG with different number of projections and (b) is the comparison of different methods

Figure 4: Picture of a Blackfin DSP camera node