Algorithms and Tools for Big Data Application

Suhail Ahamed
Communication Research Laboratory
Ilmenau University of Technology
Ilmenau, Germany
Email: suhail.ahamed@tu-ilmenau.de

***Abstract* - Identifying moving objects in a video sequence is a fundamental and critical task in many computer-vision applications. This paper evaluates and compares the classical statistical approach and the new compressed sensing based approach of separating a video sequence into a background sequence and a foreground sequence that consists of one or more moving objects/regions. The statistical approach includes the mean filter, the median filter and the frame differencing method. Simulation results show that the compressed sensing approach outperforms the statistical approach in all cases. But it has the highest computational complexity and its performance strongly depends on the length of the training sequence.**

I. Introduction

Identifying and tracking moving objects in a video sequence is a fundamental and critical task in many video applications. In surveillance, using background subtraction it is possible to identify and track foreign objects. This is important for security in train stations and airports, where unattended luggage can be a major hazard. In Sports, background subtraction is used when important decisions need to be made quickly. In tennis it is used for 'Hawk-Eye' which has become a key part of the game. It is used in traffic monitoring on traffic footage to count the number of vehicles, to detect and to track vehicles on the road. In optical motion capture, background subtraction is used in animation to give a character life and personality. An actor is filmed on up to 200 cameras which monitor their movements precisely, using background subtraction these movements can then be extracted and translated onto the character. Since background subtraction is often the first step in many applications, it is important that the extracted foreground pixels accurately describe the moving objects of interest. A good background subtraction algorithm should include fast adaptation to changes in environment, robustness in detecting objects moving at different speeds, and low implementation complexity.

A classical approach to identify the moving objects is to compare video frames to their background image. This background image can be obtained using several approaches.Foe eg., the mean and median filters can be used to extract the background image. Pixels in the current frame that deviate significantly from the background are considered to be moving objects. The basic idea of these statistical approaches have been discussed in [5]. These statistical approaches are adopted in this paper and their performance is evaluated via numerical simulations.

Another prominent method of the background and foreground separation algorithm makes use of the compressed sensing. Here it is assumed that the video sequence is separated into a sequence of sparse vectors (s_t-foreground image) and a sequence of dense vectors (l_t-background image). In this case, s_t is the quantity of interest and l_t is the potentially large but structured low-dimensional noise. This approach has three main assumptions , which will be discussed later in this paper. The state of the art method in compressed sensing based approach has been designed in [1]. In this paper implement the algorithm and compare the results to the statistical approaches.

Paper Organization: We give the system model of all the approaches Sec II. The simulation results are discussed in in Sec III. It also evaluates the performance of the different approaches. Conclusions and future work are discussed in Sec IV.

II. System Model

All the studied algorithms include a training procedure and an estimation procedure. In the training produce, a training sequence M_{train} is used. In the training sequence only the background exists. The background image sequence is assumed to be slowly changing and lies in a low-dimensional space, and the foreground

image sequence is assumed to be sparse. Also, the foreground and the changes in the foreground, are assumed to be small. In the compressed sensing based approach the frame matrices are reshaped into a vector for ease of calculation. These methods have been discussed in [5].

A. Statistical Approach: Frame differencing

This method is one of the basic methods to extract the foreground from the video sequence. In this method, there is no need for a training sequence since it updates the background at every frame. Here the background for the current frame is assumed to be the previous frame. The difference between the current frame and the previous frame is calculated and compared to a pre-assigned threshold. The pixels which are greater than the threshold are the background pixels and hence we get the background image for every frame.

$$B(x, y, t) = I(x, y, t - 1) \tag{1}$$

where,

$$|I(x, y, t) - I(x, y, t - 1)| > Th \tag{2}$$

$B(x, y, t)$ and $I(x, y, t - 1)$ are the background for pixel (x, y) at time t and the image frame for pixel (x, y) at time $(t-1)$. Th is the threshold which decides if a pixel has changed enough to be classified as the background.

This approach is naive and primitive, and is unreliable in most cases. Depending on the object structure, speed, frame rate and global threshold, this approach may or may not be useful. An improvement on this is discussed in the next section using the mean and median filters to obtain a background.

B. Statistical Approach: Mean and Median Filters

These approaches are similar to the frame differencing method. Instead of using the previous frame as the background for the current frame, the mean/median of training sequence is used as the background (hence a training sequence is required here). Each frame is compared to this image and the pixels whose values are above the threshold are treated as the foreground.

During the training period, the mean or median of each pixel is calculated and this forms the background. Once the background is formed, it is updated if and only if a drastic change is expected in the background (Lighting, etc.). Each pixel of every frame of the actual video is compared to the threshold to decide on the foreground pixels. This threshold is calculated using the Otsu's method, which chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels. This method can be used for slowly changing background by updating the background as the video progresses.

$$B(x, y, t) = \frac{1}{n} \sum_{i=0}^{n} I(x, y, t - i) \tag{3}$$

where,

$$|I(x, y, t) - \frac{1}{n} \sum_{i=0}^{n} I(x, y, t - i)| > Th \tag{4}$$

The only difference in the mean and median filters is that when estimating the background the training sequences mean/median is taken.

$$B(x, y, t) = median\{I(x, t, t - i)\} \tag{5}$$

where,

$$|I(x, y, t) - median\{I(x, t, t - i)\} > Th \tag{6}$$

C. Compressed sensing based approach: ReProCS

In this method the video is treated to be composed of a slowly changing low-dimensional subspace and a sparse component. Another way to see this is that the frame at time t (m_t) is an n dimensional vector which can be decomposed as

$$m_t := s_t + l_t \tag{7}$$

where l_t is the low-dimensional subspace and s_t is the sparse component. Also, it is assumed that l_t and s_t satisfy the three assumptions given in the next subsections. Suppose that an initial training sequence which does not contain the sparse components is available, i.e. we are given $M_{train} = [m_t; 1 \leq t \leq t_{train}]$ with $m_t = l_t$. This is used to get an initial estimate of the subspace in which the l_t's lie 1. At each t ¿ t_{train}, the goal is to recursively estimate l_t and s_t and the subspace in which l_t lies. Recursively meaning: use S_{t-1}, l_{t-1} and the previous subspace estimate to estimate l_t and s_t. Here we use p_t to denote the support of l_t, t_t the support of s_t.

1) Slowly changing low-dimensional subspace (background): By slow subspace change, its meant that: for $t \in [t_j, t_{j+1})$, $\|(I - p_{(j1)}p'_{(j1)})l_t\|_2$ is initially small and increases gradually.

$$\|(I - p_{(t-1)}p'_{(t-1)})l_t\|_2 \ll min(\|l_t\|_2, \|s_t\|_2) \tag{8}$$

The changes in the background are not as significant and the background itself or the sparse foreground. The above piecewise constant subspace change model is a

simplified model for what typically happens in practice. In most cases, p_t changes a little at each t in such a way that the low-dimensional assumption approximately holds.

Since background images typically change only a little over time (except in case of a camera viewpoint change or a scene change), it is valid to model the mean-subtracted background image sequence as lying in a slowly changing low-dimensional subspace.

2) Dense background: Very often, the background images primarily change due to lighting changes (in case of indoor sequences) or due to moving waters or moving leaves (in case of many outdoor sequences). All of these result in global changes and hence we can assume that the subspace spanned by the background image sequences is dense. Hence sparse vectors are recoverable from :

$$(I - p_{(t-1)}p_{(t-1)})m_t = (I - p_{(t-1)}p'_{(t-1)})s_t \quad (9)$$

3) Small support size, some support change, small support change assumption on s_t: In the video application, foreground images typically consist of one or more moving objects/people/regions and hence are sparse. Also, typically the objects are not static, i.e. there is some support change at least every few frames. On the other hand, since the objects usually do not move very fast, slow support change is also valid most of the time. The time when the support change is almost comparable to the support size is usually when the object is entering or leaving the image, but these are the exactly the times when the objects support size is itself small

Algorithm: $\hat{s}_t$, $\hat{l}_t$ is used to denote estimates of s_t and l_t respectively; and $\hat{p}_t$ is used to denote the basis matrix for the estimated subspace of l_t at time t. Also, let

$$\Phi_t := (I - \hat{p}_{t-1}\hat{p}'_{t-1}) \quad (10)$$

Given the initial training sequence which does not contain the sparse components, $M_{train} = [l_1, l_2, ...l_{ttrain}]$ we compute $\hat{P}_0$ as an approximate basis for M_{train}, i.e. $\hat{P}_0 = approx - basis(M_{train}, b\%)$. Let $\hat{r} = rank(\hat{P}_0)$. We need to compute an approximate basis because for real data, the l_t's are only approximately low-dimensional. We use b% = 95% or b% = 99.99% depending on whether the low-rank part is approximately low-rank or almost exactly low-rank. After this, at each time t, ReProCS involves 4 steps:

(a) **Perpendicular Projection:** In the first step, at time t, project the measurement vector, m_t, into the space orthogonal to range(p_{t1}) to get the projected measurement vector,

$$y_t := \Phi_t m_t. \quad (11)$$

(b) **Sparse Recovery (recover t_t and s_t) :** Multiplying the obtained orthogonal complement to the current frame (mt) will eliminate the background since $(I - p_{(t-1)}p'_{(t-1)})l_t$ is small. s_t can be recovered by solving :

$$min_x \|x\|_1 s.t. \|y_t - \Phi_t x\|_2 \leq \xi \quad (12)$$

After this, compressed sensing algorithm, e.g., basis pursuit, is used to recover the s_t. ξ here is the constraint equal to $\|\hat{\beta}_t\|_2$. $\hat{\beta}_t = \Phi_t \hat{l}_{t-1}$.

(c) **Recover** l_t: The estimate $\hat{S}_t$ is used to estimate l_t as $\hat{L}_t = M_t \hat{S}_t$. Thus, if s_t is recovered accurately, so will l_t.

(d) **Subspace Update (update $\hat{P}_t$) :** Within a short delay after every subspace change time, one needs to update the subspace estimate, $\hat{P}_t$. To do this in a provably reliable fashion, we introduced the projection PCA (p-PCA) algorithm in [2].

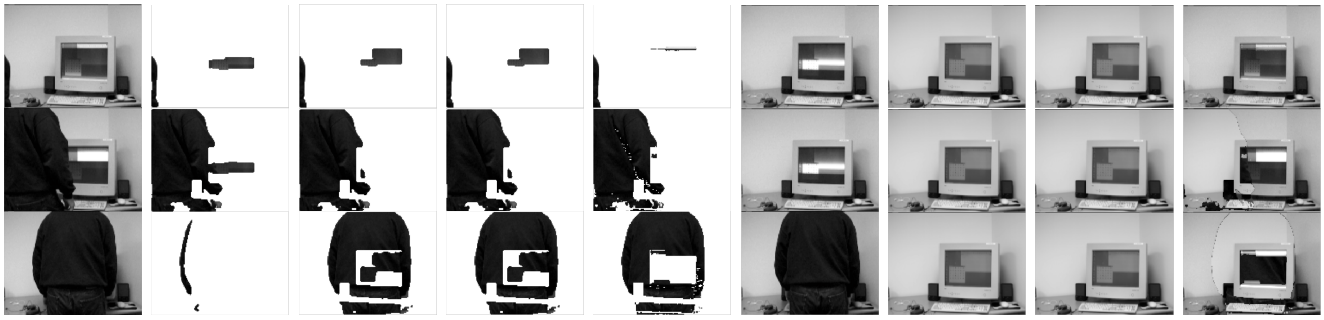
III. Simulation and experimental results

The comparison has been performed in two ways. Firstly a partly *simulated video sequence* is used where a foreground is manually introduced in the video sequence. Lake sequence serves as a real background sequence and moving rectangular block of size 45x25 is used as a foreground. The use of a real background sequence allows the evaluation of the performance for data that only approximately satisfies the low-dimensional and slow subspace change assumptions. The use of the simulated foreground allows the control of its intensity so that the resulting s_t is small or of the same order as L_t (making it a difficult sequence).

Figures 1 and 3 show the performance comparison between different approaches for solving video background/foreground separation problem using the simulated video sequence. 80 realizations of the video sequence were generated. The comparisons in terms of the normalized mean error (NME) is shown in Fig. 3. Visual comparisons of both foreground and background recovery for one realization are shown in Fig. 1. All the statistical based approaches provide a bad performance since the pixel colour variation does not vary much between the background and the introduced foreground. The compressed sensing based approach performs very well.

Original frame | Mean Filter (fg) | Median Filter (fg) | ReProCS(fg) | Mean Filter (bg) | Median Filter (bg) | ReProCS (bg)

Fig. 1: Simulated lake video sequence at $t = t_{train} + 20, 60, 80$ and its foreground(fg) and background(bg) result using mean filter, median filter and ReProCS algorithms



Original frame | Frame Diff.(fg) | Mean Filter (fg) | Median Filter(fg) | ReProCS (fg) | Frame Diff.(bg) | Mean Filter (bg) | Median Filter(bg) | ReProCS (bg)

Fig. 2: Person video sequence at $t = t_{train} + 42, 45, 52$ and its foreground(fg) and background(bg) result using frame differencing, mean filter, median filter and ReProCS algorithms

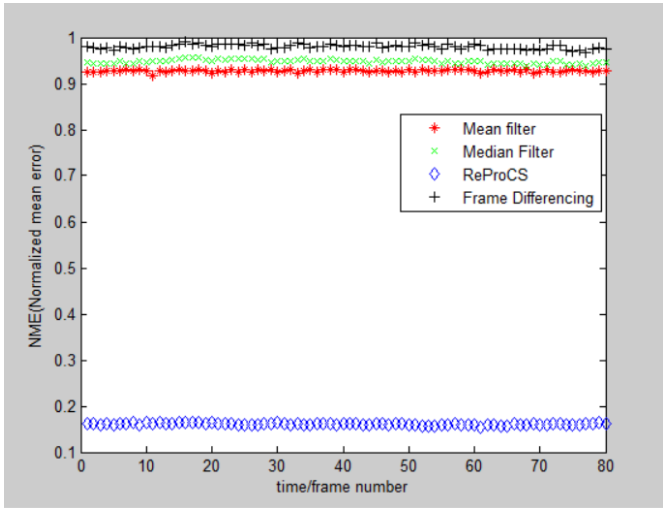


Fig. 3: Experiment on the party simulated video (lake). Normalized mean error of the different methods.

Next we compare the different algorithms using *real video sequences*. Firstly we have the video sequence with the person walking into a PC shown in Fig. [2]. The number of training frames here is 200. The mean filter, median filter and the ReProCS methods perform very well here, having a clear distinction between the back ground and the foreground. The frame differencing method performs poorly since the foreground(the person) moves in too fast. All the methods fail to recognize the moving PC screen and update the background accordingly.

Next we have the curtain video sequence shown in Fig. 4. Here the number of training frames is large ($t_{train} = 1755$). Four separate instances of the video sequence is shown. In the first instance we have the person in black shirt. All methods perform well in this frame, except for the frame differencing method. In the next two instances we have a person in white shirt. The statistical approaches perform poorly and cannot

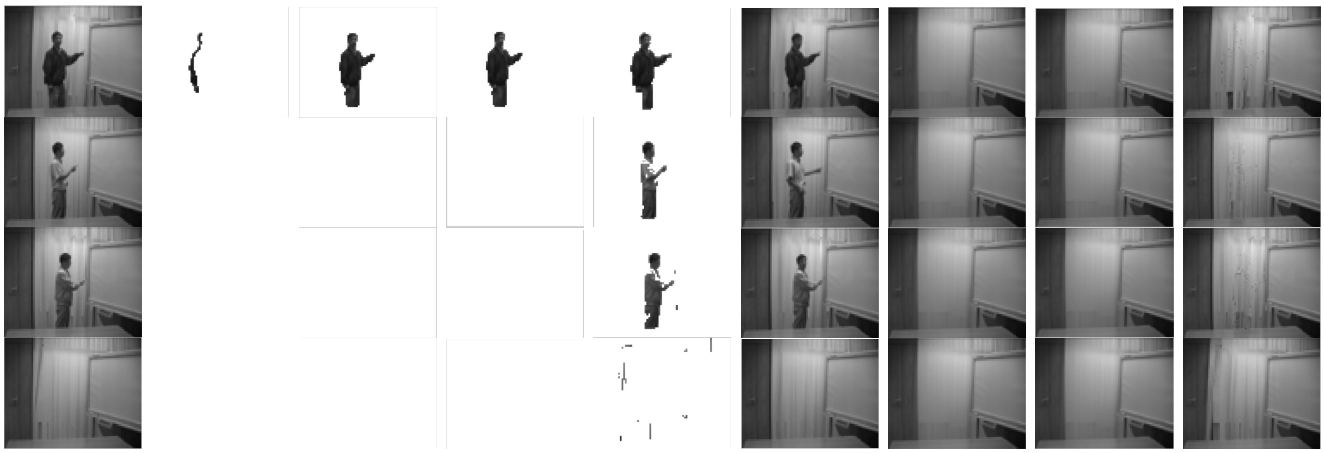


Fig. 4: Curtain video sequence at $t = t_{train} + 74, 477, 1084, 1029$ and its foreground(fg) and background(bg) result using frame differencing, mean filter, median filter and ReProCS algorithms

distinguish the foreground and the background. The compressed sensing based approach clearly outperforms the others and has a clear distinction between the foreground and the background. We also notice that when the number of training frames is increased it improves the performance of the compressed sensing based approach.

IV. Conclusion

In this work we have compared the statistical approaches and the compressed sensing approach in the applications of foreground and background separation. It is shown that the compressed sensing based approach outperforms the classical statistical approach for the simulated data. Also, the median filter based background estimation required relatively a lot of memory. The Frame differencing method yields poor results when either the object speed or the frame rate is low. Since the global threshold is not a function of time in any of the statistical approaches, it is highly susceptible to global changes in the background. The statical approaches require that the foreground and background pixel colours should have a large variation. But their performance does not depend on the length of training sequence in contrast to the compressed sensing based approaches.

None of these methods use the temporal component of the video sequence to its full potential. We can further improve upon these methods methods by using the three dimensional nature of video sequences. This can be done by using tensor based signal processing.

References

[1] Han Guo, Chenlu Qiu, Namrata Vaswani, "An Online Algorithm for Separating Sparse and Low-dimensional Signal Sequences from their Sum," *IEEE Transactions Signal Processing*, vol. 62, pp. 4284-4297, Jun. 2014

[2] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust PCA or recursive sparse recovery in large but structured noise," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.5954-5958, May 2013

[3] J. Grosek and J. Nathan Kutz, "Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, arXiv:1404.7592, Apr. 2014

[4] Andrew E. Waters, Aswin C. Sankaranarayanan, Richard G. Baraniuk, "SpaRCS: Recovering Low-Rank and Sparse Matrices from Compressive Measurements," *Neural Information Processing Systems (NIPS)*, Dec. 2011

[5] Birgi Tamersoy, "Background Subtraction", Sep. 2009