Extendibility of the Teknomo-Fernandez Algorithm for Background Image Generation

PATRICIA ANGELA ABU[1] and PROCESO FERNANDEZ[2]
Ateneo de Manila University
Department of Information Systems and Computer Science
Quezon City
PHILIPPINES
[1]patricia.reyes-abu@obf.ateneo.edu [2]pfernandez@ateneo.edu

Abstract: The Teknomo-Fernandez (TF) Algorithm is an efficient algorithm for generating a background image from a given video sequence. It uses a tournament-like strategy, with 3 frames per tournament ($TF3$), to approximate the background pixel value at every pixel position of an image frame. In this study we perform both theoretical and empirical analyses of the extendibility of this TF version by considering tournament sizes of 5 ($TF5$), 7 ($TF7$) and even higher. The expected accuracies of the TF configurations are derived which are verified by the gathered experimental performances. A model background image and a framework for comparing the performance of the algorithms in terms of accuracy, space and time complexities are developed. From the theoretical and empirical results, the $TF3$ configuration is extendible to $TF5$ and $TF7$. However, it becomes impractical to extend it to beyond $TF7$ because of the exponential growth in the required number of Boolean operations. This study also shows $TF3$ to be the clear winner among the TF configurations in terms of marginal accuracy and processing time. Applying a background subtraction on the generated background images validates the competitiveness of this technique against other background modeling techniques in literature.

Key–Words: background generation, boolean operation, modal values

1 Introduction

Segmentation of the foreground in a video sequence is a basic step for many computer vision applications and analyses. An accurate segmentation of the foreground, or moving objects, results to an accurate object tracking for further processing. Background subtraction is a commonly used technique for moving object detection by getting the difference of the foreground and background images.

The efficiency of the background subtraction lies on the accuracy of the generated background image that is used as a reference background. It is desirable that the generated background image matches the ideal background image. The difficulties in generating the background image involve the development of a model that should represent the ideal background scene and a model that is sensitive enough to detect changes in the background scene such as lighting effects and moving tree branches [9]. Failure to address these difficulties results to an inaccurate background image and leads to detecting false objects.

Maintaining a model background of a scene is a challenging task in background subtraction. The main objective of a background subtraction approach is to determine the foreground scene. Segmentation and tracking of moving objects has relied heavily on the background subtraction approach. Piccardi [11] performed an extensive review on several background subtraction techniques such as Gaussian distribution, kernel density, median filter and eigenbackgrounds.

2 Review of Related Literature

Background image modeling in video surveillance are subject to many challenges. These difficulties were enumated by Brutzer et. al. [2] and Maddalena et. al. [10] as gradual and sudden illumination changes, dynamic background, cast shadows, bootstrapping and camouflage. The ideal background modeling technique should be able to avoid these challenges. Cucchiara et. al. [3] proposed the Statistical And Knowledge-Based ObjecT detection (Sakbot) that addresses the challenge of cast shadows. Toyama et. al. [14] proposed the Wallflower scheme that solves the issues on illumination changes and camouflage. Many data sets were developed such as PETS [17][18] that offer test sequences that account to these issues.

Maddalena et. al. [10] lists the common classifications of the proposed techniques as parametric or nonparametric, unimodal or multimodal, recursive or nonrecursive, and pixel-based or region-based method. Previously proposed methods are not strictly classified and can be a combination of these classifications. Maddalena et. al. proposed a nonparametric, multimodal, recursive and pixel-based approach.

A background modeling method that is based on a parametric approach estimates the background model based on an assumed distribution of the pixel intensity values. Parametric approaches that were proposed in literature include W^4 [7], [12], Wallflower [14], Pfinder [15] and [16]. A nonparametric approach does not assume a known distribution of the pixel intensity values. Proposed nonparametric approaches include [5], [9] and [8]. The nonparametric approach is more robust than the parametric approach because it can easily adapt to pixel intensity data with unknown distribution. However, in terms of time and space complexity the parametric approach remains to be generally more efficient.

The previously proposed techniques have high computational complexity and process all or majority of the frames. One technique that considers the speed of processing is the TF algorithm [13] . This algorithm assumes that the background image is composed of pixels whose values can be approximated by taking the most frequently occurring bit value at each pixel bit position (e.g., each bit of a 24-bit pixel value). This algorithm also requires significantly less number of frames in order to generate a background image with considerably high accuracy to that of the model background.

The work presented in this paper presents a theoretical and empirical analyses of the extendibility of the TF algorithm. The original TF algorithm uses 3 sample frames in its tournament-like main processing step and is denoted here by $TF3$. In this algorithm, the *pixel bit mode* value at a specific bit position and at a specific pixel location is the bit that occurs in at least 2 out of the 3 sample frames. Aggregating these modal bits generates a level-1 background image. Taking 2 other level-1 background images produces a level-2 background image and so on. Figure 1 illustrates the generalized level-wise steps of the TF algorithm.

Abu, et al. [1], explored replacing the Boolean operations in the $TF3$ algorithm with a serial counting of occurrences of bit values. Although this technique is slower than the $TF3$, they further showed both theoretically and empirically, that for a fixed total number of sample frames, having fewer levels but larger tournament size produces a more accurate background image than having more levels but smaller tournament size. This finding has generated an interesting research question: Can the TF algorithm be extended in order to have larger tournament sizes so that even having fewer levels produces highly accurate results? This study attempts to answer this question using a nonparametric, unimodal, nonrecursive and pixel-based background modeling technique.

3 Methodology

The TF algorithm combines the strategies of (1) Random Sampling, (2) Boolean Computation and (3) Multi-level Processing. The difficulty in extending the TF algorithm lies in the Boolean computation. Hence an appropriate Boolean formula was first developed for $TF5$ and $TF7$. The ease of extending this to $TF9$, $TF11$, etc., is then investigated. A ground truth generator (GTGen) was implemented to produce a *pixel bit mode* model of the ideal background image. The TF configurations $TF3$, $TF5$ and $TF7$ and the GTGen were implemented in C++, incorporating the OpenCV library.

Theoretical results were derived and empirical results were gathered to determine the performance of the TF configurations based on their accuracies, and space and time complexities. The ground truth image is produced by the GTGen wherein all frames of the video sequence were processed, not just a random set of sample frames as opposed to TF. Consequently, the computation of the *pixel bit mode* had to be done per bit in a serial manner (instead of the parallel frame-to-frame manner in TF).

The expected accuracies were derived using an initial probability p_0 that describes the probability of obtaining the ideal background pixel from a given video stream. For the empirical results, data were collected from running the TF configurations on 5 test videos. Table 1 lists summative details of the test video files and a sample frame for each of the test videos are shown in Table 6.

Table 1: Test video parameters.

Test Video	size (MB)	length (secs)	frame count	width (pixel)	height (pixel)
1 [17]	3.74	19	594	640	480
2 [2]	2.45	5	166	480	360
3 [19]	12.10	60	1824	640	480
4 [18]	186.0	107	2695	768	576
5 [18]	150.0	107	2695	768	576

To obtain the empirical accuracies, the generated background images of the TF configurations were compared to the GTGen *pixel bit mode* model background image. Since the number of frames and the number of Boolean operations vary significantly

ISBN: 978-960-474-368-1

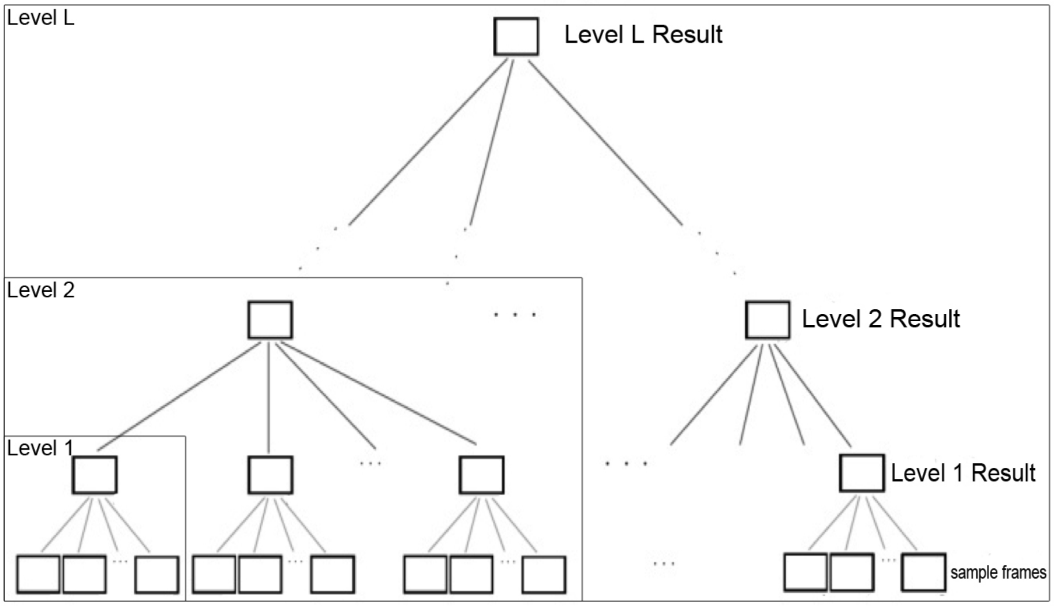


Fig. 1: TF frame processing diagram.

among the TF configurations, a framework for comparison was also developed. The application of this framework was crucial in arriving at one of the main conclusions for this study. Finally, a background subtraction was implemented on the generated background images.

4 Results

4.1 Boolean Formulas

Consider first a mechanism to develop Boolean formulas for $TF3$, $TF5$, and $TF7$. Let the frames selected for 1 tournament be labeled by A, B, C and so on. The "2 out of 3" rule for the $TF3$ can be achieved using the following Boolean formula:

$$TF3_{image} \leftarrow AB + AC + BC \tag{1}$$

We use the standard representation of addition for the Boolean OR and multiplication for the Boolean AND in Equation (1). This formula contains 5 Boolean operations (3 ANDs and 2 ORs) and is said to be in *disjunctive normal form* (DNF).

A correct DNF for $TF5$ (or $TF7$) can be produced using a similar technique of exhaustively applying the "3 out of 5" (or "4 out of 7") rule as follows:

$$\begin{aligned} TF5_{image} \leftarrow ABC + ABD + ABE \\ +ACD + ACE + ADE + BCD \\ +BCE + BDE + CDE \end{aligned} \tag{2}$$

$$\begin{aligned} TF7_{image} \leftarrow ABCD + ABCE + ABCF \\ +ABCG + ABDE + ABDF + ABDG \\ +ABEF + ABEG + ABFG + ACDE \\ +ACDF + ACDG + ACEF + ACEG \\ +ACFG + ADEF + ADEG + ADFG \\ +AEFG + BCDE + BCDF + BCDG \\ +BCEF + BCEG + BCFG + BDEF \\ +BDEG + BDFG + BEFG + CDEF \\ +CDEG + CDFG + CEFG + DEFG \end{aligned} \tag{3}$$

This demonstrates that, hypothetically, the TF algorithm can be extended to any TF*n* (where *n* is an odd number) because it is always possible to produce a correct DNF for TF*n* by generating this disjunctions of all unique

$$\binom{n}{\lceil n/2 \rceil} \tag{4}$$

conjunctions among the *n* frames of a tournament. The total number of Boolean operations N_{ops} can also be computed deterministically from the number of *n* of tournament frames as follows:

$$N_{ops} = \left(\binom{n}{\lceil n/2 \rceil} - 1\right) + \left(\left\lceil \frac{n}{2} \right\rceil - 1\right)\binom{n}{\lceil n/2 \rceil} \tag{5}$$

Table 2: Expected number of Boolean operations.

Frames per Tournament	Total # of Boolean operations
3	5
5	29
7	139
9	629
11	2,771
15	51,479
21	3,879,875

Table 2 applies this formula to demonstrate the exponential growth in the required Boolean operations per TF tournament, with respect to the number of frames.

It is important to mention that the problem of minimizing the number of Boolean operations for a given formula is an NP-Hard problem [4]. Thus, although a transformation technique may be applied to reduce the number of operations, such technique does not guarantee optimality unless an exhaustive search is done. In order to reduce the formulas for $TF3$, $TF5$ and $TF7$, we apply a Karnaugh mapping technique and perform manual simplification. We have derived the following simplifications:

$$TF3_{image} \leftarrow (A+B)C + AB \tag{6}$$

$$\begin{aligned} TF5_{image} \leftarrow\ & (D+E)(AB+AC+BC) \\ & + DE(B+C) + A(BC+DE) \end{aligned} \tag{7}$$

$$\begin{aligned} TF7_{image} \leftarrow\ & A(B(C(D+E+F+G) \\ & +D(E+F+G)+E(F+G)+FG) \\ & +C(D(E+F+G)+E(F+G)+FG) \\ & +D(E(F+G)+FG)+EFG) \\ & +B(C(D(E+F+G)+E(F+G)+FG) \\ & +D(E(F+G)+FG)+EFG) \\ & +C(D(E(F+G)+FG)+EFG)+DEFG \end{aligned} \tag{8}$$

Equations (1), (2) and (3) show the DNF, and their equivalent simplified expression with fewer Boolean operations are given in Equations (6), (7) and (8) respectively. The values, therefore, listed in Table 2 may be interpreted as upper bounds for the number of Boolean operations vis-à-vis the number of tournament frames.

4.2 Ground Truth Generator

The Ground Truth Generator (GTGen) is developed to generate the model background image. This is used as a reference image to determine the accuracies of the empirical results. Given a video sequence with n frames, each with r rows and c columns of pixel per frame, the GTGen runs through all the RGB pixels components (r, c) and selects the *pixel bit mode* from start frame to frame n.

Figure 2 illustrates the *pixel bit mode* processing and differentiates this method with the conventional modal pixel value. As an example, with $N = 6$ at pixel location (r, c), the 8-bit R component values are listed in Fig. 2. The GTGen selects the modal pixel value per bit position 0 to 7 and the resulting 8 bit value represents the *pixel bit mode*.

If the modal pixel value (for a fixed position) occurs in the majority of the frames, then GTGen *pixel bit mode* method guarantees to produce the modal pixel value also, as illustrated in Figure 2(b). Moreover, assuming that the modal pixel corresponds to the background, and that the bit values at a specific bit position is uniformly distributed for the non-background images, then the *pixel bit mode* would also correspond to the modal pixel.

	(a)Modal pixel is not majority	(b) Modal pixel is also majority
	1111 0000	1111 0000
	1111 0000	1111 0000
	1111 0000	1111 0000
	0000 0111	1111 0000
	0000 0111	1010 1010
	0001 1100	1100 1100
	0001 1100	1110 0111
Modal Pixel:	**1111 0000**	**1111 0000**
***Pixel Bit Mode*:**	**0001 0100**	**1111 0000**

Fig. 2: Modal Pixel vs *Pixel Bit Mode*.

4.3 Performances

4.3.1 Theoretical Accuracy

To analyze the expected accuracy of each of the TF configurations, a probabilistic estimation is applied. The usual assumption that the background image consists of the modal pixel value at each of the (r,c) positions is considered.

To derive the expected accuracy of the different TF configurations, an analysis is first made based on the *pixel bit mode* value. First, we fix the bit position and pixel location. Let p_0 be the probability that the bit from the randomly selected frame is the same as the actual *pixel bit mode*. Numerically, this is just given by

ISBN: 978-960-474-368-1

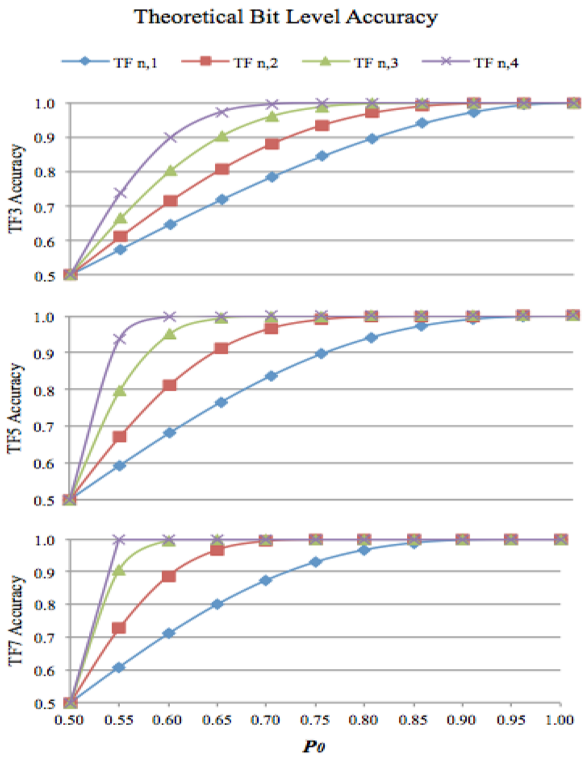


Fig. 3: Graph of theoretical bit level accuracies of $TF3$, $TF5$ and $TF7$.

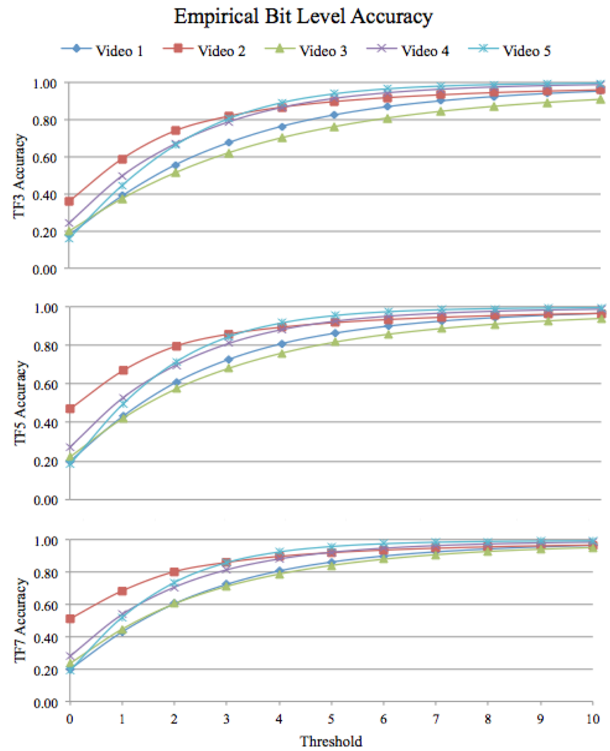


Fig. 4: Graph of empirical bit level accuracies run on 5 sample test videoes.

$$p_0 = \frac{\text{\# of occurrences of the modal bit}}{\text{\# of frames}} \quad (9)$$

The number of different possible cases where the *pixel bit mode* is actually generated (from the random samples) is a function of the number S of samples taken. For example for $S = 5$, as long as any 3, 4 or 5 of the bits are the same, then the correct *pixel bit mode* "wins", i.e., is generated. Thus the probability of the correctness of the generated *pixel bit mode* should consider each of these winning cases. This probability formula is

$$p_1 = \sum_{k=\lceil \frac{S}{2} \rceil}^{S} \binom{S}{k} (p_0)^k (1 - p_0)^{S-k} \quad (10)$$

For multi-level computations, the probability that the output bit on the ith level ($i > 0$) is dependent on the probability of success in the previous level, and is given by

$$p_i = \sum_{k=\lceil \frac{S}{2} \rceil}^{S} \binom{S}{k} (p_{i-1})^k (1 - p_{i-1})^{S-k} \quad (11)$$

Applying the formula allows us to compute for the expected accuracy, on a bit level, for each of the TF configurations. The graphs of the theoretical bit level accuracies of the different configurations are given in Fig. 3. As expected, the higher the initial probability p_0, the higher is the theoretical accuracy of the final result.

Furthermore, for a fixed tournament size, the higher the number of levels, the better is the expected accuracy. Thus, in terms of accuracy, $TF_{5,4} > TF_{5,3} > TF_{5,2} > TF_{5,1}$. This is to be expected since the number of sample frames in a configuration $TF_{n,L}$ increases by a factor of *n* (the tournament size) for every increment in *L* (the number of processing levels).

These graphs also show that, generally, the accuracy of the TF algorithm increases as the tournament size increases. That is, $TF_{3,L} < TF_{5,L} < TF_{7,L}$ when only accuracy is considered. Again, this can be explained by infering the total number of samples used to generate the background image.

4.3.2 Empirical Accuracy

To support the theoretical results, we run the TF configurations on 5 sample test videos. We report the interesting results for TF configurations where the tournament size is increased while the number of levels is

ISBN: 978-960-474-368-1

decreased: $TF_{3,4}$, $TF_{5,3}$ and $TF_{7,2}$. At a fixed initial probability p_0 = 0.60, the expected accuracies of the final results of $TF_{3,4}$, $TF_{5,3}$ and $TF_{7,2}$ are 0.8991, 0.9516 and 0.8868 respectively. Thus, in terms of expected accuracy, $TF_{5,3} > TF_{3,4} > TF_{7,2}$. This is again consistent with correlating the accuracy with the sampling size (since $5^3 > 3^4 > 7^2$).

A threshold value is used so that if the pixel value returned by a TF configuration is not significantly different from the corresponding *pixel bit mode* value of the GTGen, then the pixels are still considered similar. To illustrate, if the threshold value is 10, then corresponding pixels from different configuration results are considered the same if they do not differ by more than 10, in any of the color bands (RGB). Note that in this scheme, if the threshold is zero then the pixel values between corresponding frame locations have to be completely identical for them to be considered similar.

Figure 4 illustrates the competitive accuracy of the TF configurations that can attain an accuracy of 0.9 and above at threshold of ± 5. Empirical results show the efficiency of the TF configurations in terms of accuracy considering that it processed significantly fewer frames than the GTGen.

4.3.3 Space and Time Complexities

Since the GTGen based on *pixel bit mode* values processes all frames in a given video sequence, this already signifies that it requires a very high space complexity. The TF algorithm, on the other hand, requires only S^L frames to generate a background image of considerably high accuracy. With sufficiently small S and L values, the TF algorithm requires a much smaller computing space than GTGen.

Between the TF configurations, a lower number of sample S and level L of processing results to fewer initial frames to be saved into an array. Taking for example a 4-level processing, $TF3$, $TF5$ and $TF7$ would require a space allocation that can handle the initial random frame samples of 81, 625 and 2,401 respectively.

Running over Test Video 1, the GTGen needs a space allocation for all 594 frames of the video sequence to generate the background image. $TF_{3,4}$ on the other hand, would only require 3^4 = 81 random frames (including possible repetitions). As for the $TF_{7,4}$ configuration, it requires 7^4 = 2,401 sample frames, which is more than the total number of frames of the video.

The theoretical runtime of the TF algorithm is $O(nrc)$ where n is the number of frames, with r rows and c columns of pixels and each pixel having a constant number of bits.

To further test the performance of the algorithms, the time elapsed of each configuration was monitored. This time starts from reading an input video file, saving the required number of random frames in an array, processing the frames and saving the generated background image.

Since the frames are randomly selected, 30 sets of simulations were conducted and the mean results were recorded. Table 3 shows the mean actual processing time for each algorithm and TF configuration.

Table 3: Actual elapsed time in seconds.

Sample	GTGen	$TF_{3,4}$	$TF_{5,3}$	$TF_{7,2}$
Video 1	466.610	1.0952	1.0608	0.7546
Video 2	64.270	0.2506	0.2376	0.2730
Video 3	1518.740	1.1185	1.8101	0.7883
Video 4	3544.074	7.1101	11.2184	4.4499
Video 5	3797.248	6.8176	10.5740	4.1298

Actual simulation results therefore show that the TF configurations remain more efficient in terms of space allocation and processing time compared to the GTGen. However, for the configurations of the TF algorithm, as the number of frames S and level of processing L increases, the space allocation and processing time increase also. In addition, the number of Boolean operations significantly increases and thus an additional time is needed for processing.

4.4 Framework of Comparison

The general observation that the number of sample frames correlates positively with accuracy introduces some bias for configurations that require more sample frames. Thus, marginal accuracy and marginal time are introduced for a better comparison of the different TF configurations. This framework is important especially when both the tournament size S and the number of levels L are different (e.g., $TF_{3,4}$, $TF_{5,3}$ and $TF_{7,2}$).

The marginal accuracy is computed by dividing the empirical accuracy by the number of sample frames used. The accuracy of the TF configuration, on the other hand, is based on the similarity of the generated background to the ground truth GTGen. This similarity compares *pixel bit mode* values.

Table 4 lists the marginal accuracy of each configuration at threshold equal to 10. From Table 4, the TF configurations still show significantly higher marginal accuracy than the GTGen. With this, the TF algorithm exhibits very high efficiency taking into consideration that it generated the background image using just a small number of frames (i.e., 81, 125 and 49) as compared to 594 frames processed by GTGen for Test

ISBN: 978-960-474-368-1

Table 4: Marginal accuracy and marginal time run on the test videos at threshold = 10.

Sample	GTGen	$TF_{3,4}$	$TF_{5,3}$	$TF_{7,2}$
Marginal Accuracy (% per processed frame)				
Video 1	0.0016	0.0117	0.0077	0.0197
Video 2	0.0057	0.0118	0.0077	0.0197
Video 3	0.0005	0.0112	0.0075	0.0194
Video 4	0.0004	0.0122	0.0079	0.0201
Video 5	0.0004	0.0123	0.0080	0.0203
Marginal Time (seconds per processed frame)				
Video 1	0.7855	0.0135	0.0085	0.0154
Video 2	0.3872	0.0031	0.0019	0.0056
Video 3	0.8326	0.0138	0.0145	0.0161
Video 4	1.3151	0.0878	0.0897	0.0908
Video 5	1.4090	0.0842	0.0842	0.0843

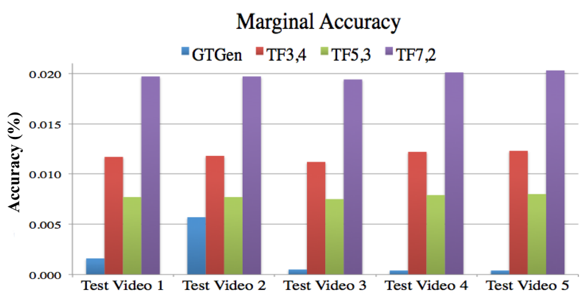


Fig. 5: Marginal accuracy vs threshold graph of each configuration run on test videos.

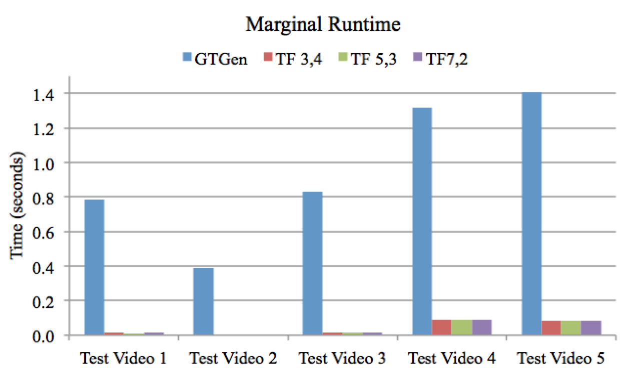


Fig. 6: Marginal processing time of each configuration.

Video 1. Similar results were gathered for the rest of the test videos.

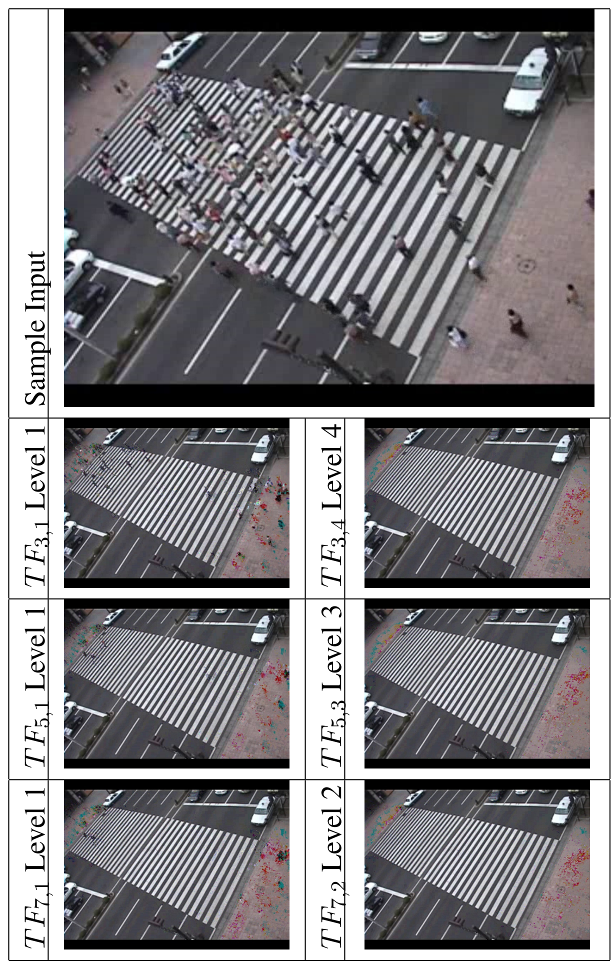
Referring back to the theoretical bit level accuracies on Figure 3, actual simulation results show that as the number of frame sample S increases and as the level of processing L increases, the accuracy of the TF configuration increases as well. Figure 5 illustrates the graph of the marginal accuracies of each configuration across 5 test videos.

Table 4 shows also the marginal processing time, each computed by dividing the actual runtime by the total number of sample frames processed. It is clear from these results that the TF algorithm is more efficient than the GTGen in terms of processing time and is clearly illustrated in Figure 6.

4.5 Background Image Generation

Table 5 shows the generated background image at the first and final level of frame processing for $TF_{3,4}$, $TF_{5,3}$ and $TF_{7,2}$. From the level 1 output images, moving objects are still visible. However, as the level increases the generated background images become "cleaner", with lesser foreground pixels.

Table 5: Level 1 and final background images.



The TF configurations were evaluated in several test videos. Extensive experimental results illustrate a good performance of the TF configurations. Table 6 shows the generated background of the extended TF configurations and the GTGen.

Table 6: Generated background images of GTGen and TF configurations on all test videos.

	Test Video 1	Test Video 2	Test Video 3	Test Video 4	Test Video 5
Sample Frame					
GTGen					
TF_3					
TF_5					
TF_7					

4.6 Background Subtraction

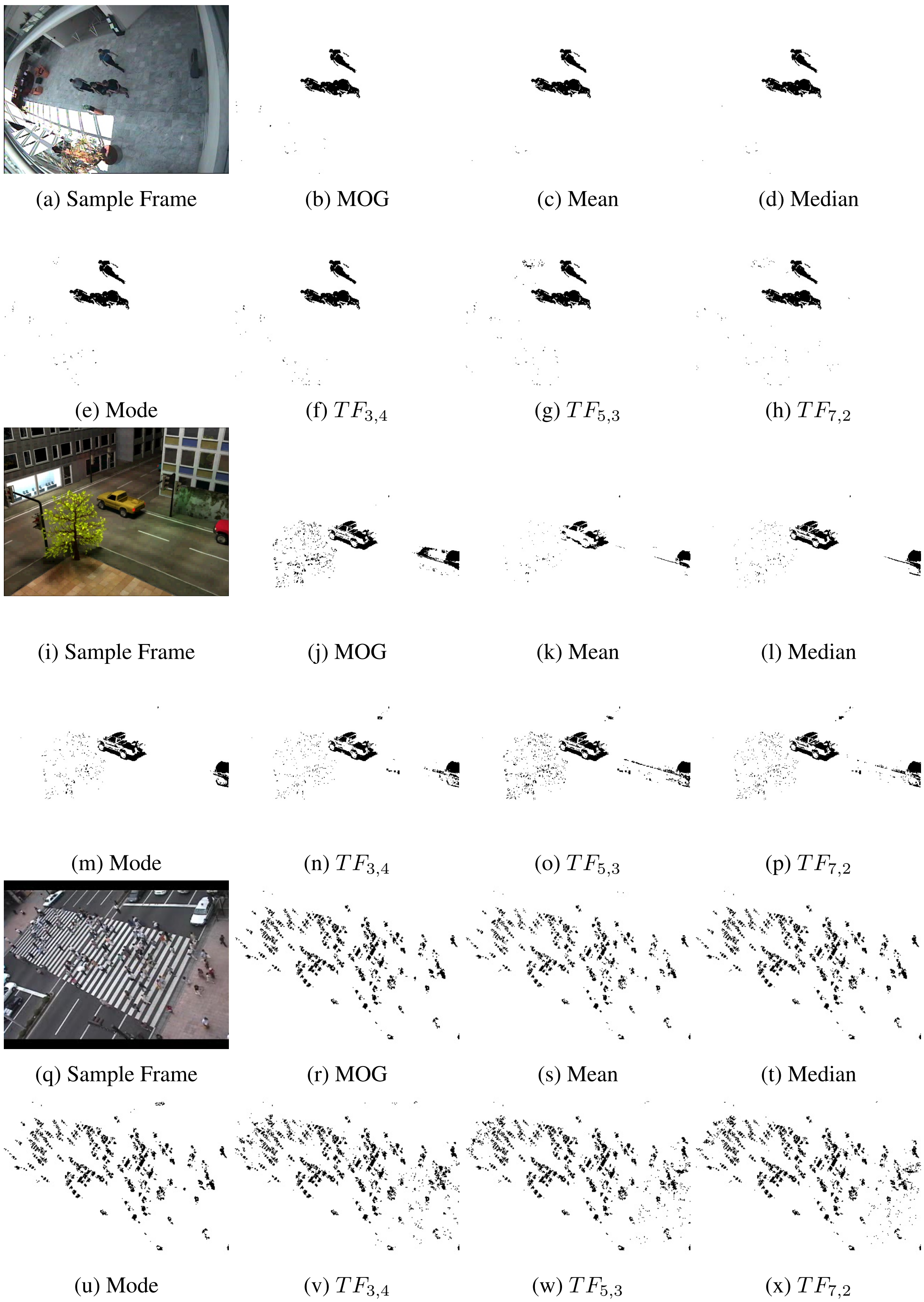
By computing the difference between an observed frame and the generated background image, the foreground becomes prominent. The difference is implemented on all RGB pixel components. A large and small absolute difference determines if the particular pixel component is to be considered as a moving object or part of the background. The differences were saved in a binary image frame with values 0 and 1 that pertains to a moving object or a background pixel. To have a fair comparison, the difference was implemented without any denoising or post processing.

The difference images of the TF configurations were compared with commonly used baseline methods – MOG, Mean, Median and Mode, which were also used as a basis to develop the promising techniques proposed in [3], [6], [8], [10], [12] and [16]. These baseline methods were also implemented and developed in C++ using OpenCV library. The MOG used is the built-in BackgroundSubtractorMOG2( ) with default values provided by OpenCV that implements the technique proposed by Zivkovic in [16]. The mean, median and mode methods were developed by computing the corresponding mean, median and mode pixel value of each pixel location across all frames in the video file.

The test sequences in Table 1 were selected so as to cover different senarios that are considered to be the difficult conditions for background subtraction. These scenerios range from slow to fast moving persons and vehicles, low to high density crowd, indoor and outdoor scenes with bright sunshine and shadows, camouflage, and a multimodal background with waving tree branches.

The difference images are shown in Fig. 7. For slow moving-object and dynamic background conditions, $TF_{3,4}$ in Fig. 7(n) performs better than MOG in Fig. 7(j). $TF_{3,4}$ exhibits mode pixel intensity based processing that is tolerant to outliers as opposed to the means of Gaussians that is sensitive to outliers. The resulting images validate that the TF algorithm can have competitive high accuracy as that of the baseline methods by processing as few as 49 frames as opposed to the baseline methods that process all the frames in the test sequence.

ISBN: 978-960-474-368-1



(a) Sample Frame (b) MOG (c) Mean (d) Median

(e) Mode (f) $TF_{3,4}$ (g) $TF_{5,3}$ (h) $TF_{7,2}$

(i) Sample Frame (j) MOG (k) Mean (l) Median

(m) Mode (n) $TF_{3,4}$ (o) $TF_{5,3}$ (p) $TF_{7,2}$

(q) Sample Frame (r) MOG (s) Mean (t) Median

(u) Mode (v) $TF_{3,4}$ (w) $TF_{5,3}$ (x) $TF_{7,2}$

Fig. 7: Background Subtraction application and comparison to commonly used baseline methods.

ISBN: 978-960-474-368-1

5 Conclusion

In this paper, the TF algorithm was shown to be extendible to $TF5, TF7$ and even higher configurations. However, both theoretical and simulation results reveal that the extendibility of the TF algorithm involves practical issues such as increasing number of Boolean operations and space and time requirements.

Based on the model background image and framework that was developed for comparison of simulation results, $TF3$ still remains to be most efficient and is the best TF configuration for actual implementation. An application of a background subtraction on the TF generated background images validates the efficiency and competitive high accuracy with fewer processed frames of the TF configurations compared to commonly used baseline methods.

Further improvements on this study may include the development of a performance index that incorporates properly the factors of accuracy, space and time complexities of the TF configurations.

Acknowledgements: The authors would like to acknowledge the Engineering Research and Development Technology (ERDT) for supporting this research.

References:

[1] P. Abu, V. Chu and P. Fernandez, A Monte-Carlo-based Algorithm for Background Generation, *National Conf. on Information Technology Education*, 2012.

[2] S. Brutzer, B. Hoferlin and G. Heidemann, Evaluation of Background Subtraction Techniques for Video Surveillance, *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 1937–1944.

[3] R. Cucchiara, C. Grana, and M. Piccardi, Detecting Moving Objects, Ghost and Shadows in Video Streams, *IEEE Trans. on Pattern Analysis and Machine Itelligence*, 25(10), 2003, pp. 1337-1342.

[4] G. De Micheli, *Synthesis and Optimization of Digital Circuits.* McGraw-Hill, 1994.

[5] A. Elgammal, D. Harwood, and L. Davis, Non-parametric Model for Background Subtraction, *European Conf. on Computer Vision*, 2000, pp. 751-767.

[6] B. Han, D. Comaniciu and L. Davis. Sequential Kernel Density Approximation through Mode Propagation: Applications to Background Modeling, *Asian Conf. on Computer Vision*, Vol. 4, 2004.

[7] I. Haritaoglu, D. Harwoodand, L.S. Davis, W4: real-time surveillance of people and their activities, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22 (8), 2000, pp. 809-830.

[8] K. Kim, T. Chalidabhongse, D. Harwood, L. Davis, Real-time foreground-background segmentation using codebook model, *Real Time Imaging 11*, (3), 2005, pp. 172-185.

[9] Y. Liu, H. Yao, W. Gao, X. Chen, and D. Zhao, Nonparametric Background Generation, *J. Vis. Comm. Image Rep.*, 18(3), 2007, pp. 253-263.

[10] L. Maddalena and A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, *IEEE Trans. on Image Processing*, 17(7), 2008, pp.1168-1177.

[11] M. Piccardi, Background Subtraction Techniques: A Review, *IEEE Int. Conf. on Systems, Man and Cybernetics*, Vol. 4, 2004.

[12] C. Stauffer and W. Grimson, Adaptive Background Mixture Models for Real-Time Tracking, *IEEE Conf. on Computer Vision and Pattern Recognition*, 1999, pp. 246-252.

[13] K. Teknomo and P. Fernandez, Background Image Generation Using Boolean Operations, *Philippine Computing Journal*, 4(2), 2009, pp. 43–49.

[14] K. Toyama, J. Krumm, B. Brumitt and B. Mayers, Wallflower: Principles and Practice of Background Maintainance, *IEEE Int. Conf. on Computer Vision*, 1999, pp. 255-261.

[15] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, Pfinder: Real-time Tracking of the Human Body, *IEEE Trans. on Pattern Analysis and Machine Itelligence*, 19(7), pp. 780785, 1997.

[16] Z. Zivkovic, Improved adaptive Gausian mixture model for background subtraction, *Int. Conf. on Pattern Recognition*, 2004.

[17] IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, 2004. EC Funded CAVIAR project/IST 2001 37540. Available from URL: http://www.dai.ed.ac.uk/homes/rbf/CAVIAR/

[18] IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, 2001. Available from URL: http://www.cvg.cs.rdg.ac.uk/PETS2001/pets2001-dataset.html.

[19] K. Teknomo. Available from URL: http://people.revoledu.com/kardi/download/-download.php?file=PedestrianMovie1.