Extending the Teknomo-Fernandez Background Image Generation Algorithm on the HSV Colour Space

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: Background subtraction, a procedure required in many video analysis applications such as object tracking, is dependent on the model background image. One efficient algorithm for background image generation is the Teknomo-Fernandez (TF) Algorithm, which uses modal values and a tournament-like strategy to produce a good background image very quickly. A previous study showed that the TF algorithm can be extended from the original 3 frames per tournament ($TF3$) to $TF5$ and $TF7$, resulting in increased accuracies at a cost of increased processing times. In this study, we explore extending the $TF3$, $TF5$ and $TF7$ from the original RGB colour space to the HSV colour space. A ground truth model background image for HSV was also developed for comparing the performances between the TF implementations on the RGB and HSV channels. The results show that the TF algorithm generates accurate background images when implemented on the HSV colour space. However, the RGB implementations still exhibit higher accuracies than the corresponding HSV implementations. Finally, background subtraction was applied on the HSV generated background images. A comparison with other promising baseline techniques validates the competitiveness of the TF algorithm implemented on HSV channels.

Key–Words: background subtraction, boolean operation, HSV, mode values

1 Introduction

Segmentation of the background and foreground objects is the primary step in many computer vision applications. From a given video sequence, an accurate segmentation of the foreground is desired to properly detect moving objects from the scene. Accurate segmentation and detection of moving objects result to accurate detection systems used for further image manipulation and analyses.

Background subtraction is a commonly used technique for object segmentation and detection. Tracking of moving objects have relied heavily on this technique, which is essentially a subtraction operation on each pixel between the current observed frame and the background model image. An extensive review of the background subtraction techniques proposed in literature was performed by Piccardi [17]. Some of the well known background subtraction algorithms are based on the principles of Gaussian distribution, kernel density, median filter and eigenbackgrounds.

The generated background image that is used as a reference model image determines the efficiency of the background subtraction technique. It is desirable that the background model image represents the ideal background image of the scene. Moreover, the model image should also be sensitive to detect and adapt to changes in the background scene such as lighting changes and waving tree branches [15]. Background image generation algorithms that fail to address these difficulties produce inaccurate background model images that lead to detection of non-moving object as moving objects and vice versa.

The Teknomo-Fernandez (TF) algorithm is an efficient algorithm for background image generation. This algorithm uses a tournament-like strategy to obtain the modal pixel values [24]. A previous study showed that the TF algorithm can be extended from the original 3 frames per tournament ($TF3$) to tournament of sizes 5 ($TF5$) and 7 ($TF7$), resulting in increased accuracies at a cost of increased processing times [2].

In this study, we explore the possibility of extending the $TF3$, $TF5$ and $TF7$ from the original RGB colour space to the HSV colour space.

2 Review of Related Literature

In video surveillance applications, background image modeling is subject to many challenges. Brutzer et. al. [4] and Maddalena et. al. [16] enumerated these challenges as gradual and sudden illumination changes, dynamic background, cast shadows, bootstrapping and camouflage. The background image generation algorithm should be able to adapt to these changes in the background scene in order to produce a background model that best represents the ideal background image.

Several proposed background modeling techniques in literature are Statistical And Knowledge-Based ObjecT detection (Sakbot) by Cucchiara et. al. [5] which addresses the issue of cast shadows and the Wallflower scheme by Toyama et. al. [25] that solves the difficulties on illumination changes and camouflage. PETS [28][30] has a collection of data sets with test sequences that incorporate these issues.

A classification of the proposed background subtraction and generation techniques have been proposed by Maddalena et. al. [16], considering each technique as either parametric or nonparametric, unimodal or multimodal, recursive or nonrecursive, and pixel-based or region-based method. However, the proposed methods may not be strictly classified and may be a combination of these classifications. Maddalena et. al. [16] proposed a background subtraction approach that is nonparametric, multimodal, recursive and pixel-based. The pixel-based and region-based techniques were also used in medical applications specifically in processing microscopic tissue images such as those proposed by Remenyi, et al. [18], Szenasi et al. [22] and Szenasi [23]

The parametric approach performs an estimate of the background model based on an assumed distribution of the pixel intensity values across all pixels in the image frame and across all frames in the video sequence. Some of the promising parametric approaches include W^4 [10], [19], [20], [21], Wallflower [25], Pfinder [26] and [27]. The work proposed by Appriah et. al. [3] developed GW4 that follows the principles proposed by W^4 [10] and [19] and was implemented on a FPGA hardware.

Background modeling techniques that do not rely on an assumed known distribution is the nonparametric approach. Some of the nonparametric methods proposed in literature are [7], [11], [12], [13] and [15]. The nonparametric approach is considered to be more robust over the parametric approach since it can easily adapt to pixel intensity data even with an unknown distribution. On the other hand, the parametric approach remains to be generally more efficient in terms of time and space complexities.

The previously proposed techniques have high computational complexity and process all or majority of the frames. One technique that considered a non-statistical approach in generating and updating the background model is [14]. The updating of the background model is based on logical relationships between the differences of the current frame to the background frame and previous frame. Thresholding and conversion to binary is applied to the resulting differences. The logical operations AND and XOR are performed on the binary represented values to update the background model.

One technique that uses logical operations and considers the speed of processing is the TF algorithm [24], a background modeling technique that is nonparametric, unimodal, nonrecursive and pixel-based. 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 novel algorithm also requires significantly less number of frames in order to generate a background image with considerably high accuracy with respect to the model background. Fig. 1(a) illustrates the generalized level-wise steps of the TF algorithm.

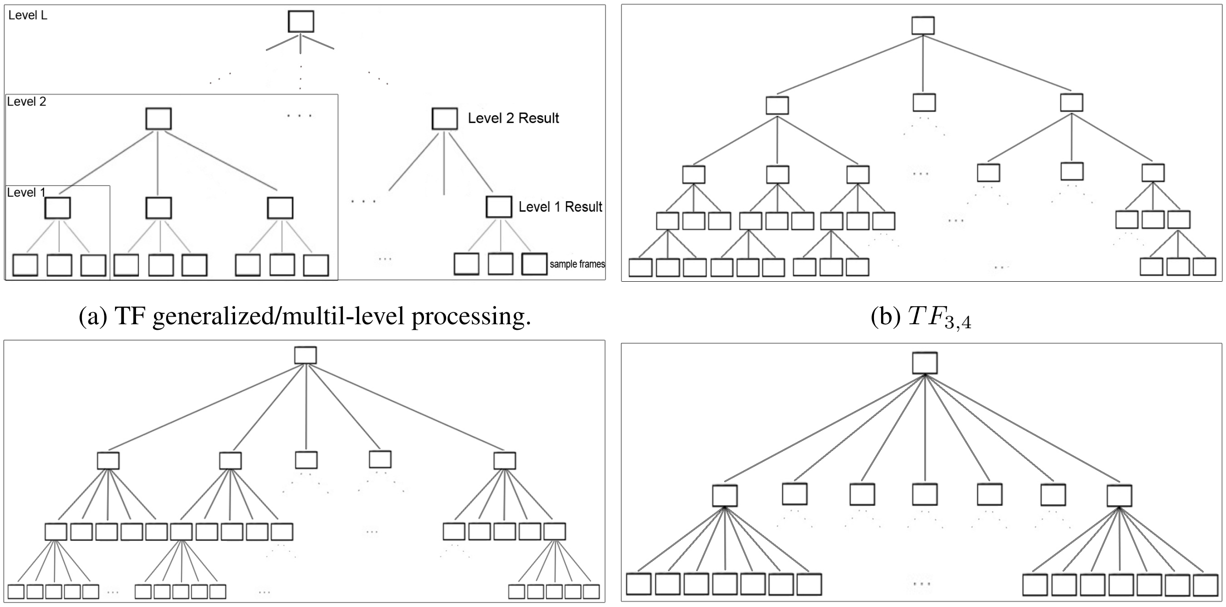
Abu et al. [1] explored replacing the boolean operations in the original TF algorithm (which they denote by $TF3$ to specifically indicate the tournament size) with a simple serial counting of the occurrences of bit values. Although this technique turned out to be slower than the $TF3$, it led to the discovery, through both theoretical derivations and empirical experimentations, 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. A succeeding study investigating the extendibility of the TF algorithm in order to have larger tournament sizes (so that even having fewer levels produces highly accurate results) led to the following findings [2]:

1. On extendibility: Theoretically, the original $TF3$ is extendible to $TF5$, $TF7$ and even higher configurations. However, it can only practically be extended up to $TF7$ because of the exponential growth in the required number of boolean operations.

2. On efficiency: The $TF3$ still remains to be most efficient, among the TF configurations, in terms of accuracy vis-a-vis the number of required sample video frames.

3. On accuracy: The $TF3$, $TF5$ and $TF7$ each generate a good background image that is competitive with other baseline methods even though only a significantly fewer sample frames are required by the TF configurations.

In this study, we attempt to extend the TF algorithm in a different direction, specifically by applying



(a) TF generalized/multil-level processing.

(b) $TF_{3,4}$

(c) $TF_{5,3}$

(d) $TF_{7,2}$

Fig. 1: TF processing diagram

the technique on the HSV colour space. The performance of TF in the HSV space is then compared with the results of the original RGB space.

3 Preliminaries

TF Algorithm. The TF algorithm is a background generation algorithm that assumes that the model background image consists of pixels which are individually the modal integer values at the corresponding pixel locations for the entire video sequence. It is able to estimate this assumed ideal background very quickly by combining the following techniques:

1. Random Sampling: The initial set of frames are randomly selected (with replacement) from the given video sequence, from start frame 0 to last frame $(N-1)$. The TF algorithm requires significantly less number of initial frames for processing, compared with many of the other methods, and yet results in a considerably high accuracy with respect to the ideal background. The number of initial random sample frames depends on the number of multi-level processing defined by the user.

2. Boolean Computation: The TF algorithm approximates the 24-bit modal integer value (at a given pixel position) by performing the processing on a bit level. It assumes that the background image pixel values represented in 24-bits can be approximated by taking the most frequently occurring bit value at each of the bit positions 0 through 23 using Eq. 1 [24].

$$B = x_3(x_1 \oplus x_2) + x_1 x_2 \tag{1}$$

In this equation, B is the resulting integer value from processing the corresponding pixel integer values x_1, x_2 and x_3 coming from 3 input frames. Note that this formula can be applied directly to the 3 frames, so that the processing is naturally done in parallel, thereby contributing to the high speed of execution of the TF algorithm.

3. Multi-level Processing: From a given video sequence, the TF algorithm uses a tournament-like strategy, with 3 frames per tournament, to approximate the background pixel value at every pixel position of the image frame. The tournament strategy incorporates a multi-level processing that is illustrated in Fig. 1(a). The details of the TF processing are discussed more comprehensively in [24].

Extended TF. Abu, et al. [2], examined the extendibility of the TF algorithm and implemented a multi-level processing having 5 and 7 frames per tournament. The difficulty in extending the TF algorithm lies in developing the boolean equations. The original TF algorithm was denoted as $TF3$ and the extended TF are labeled as $TF5$ and $TF7$, respectively. The derived boolean equations used in the tournament for $TF3$, $TF5$ and $TF7$ are given in Equations 2, 3 and 4 respectively. In these equations, the variables $A, \dots, F$ represent the image frames which are matrices of pixel integer values that are used for the tournament-like processing.

$$TF3_{image} \leftarrow AB + AC + BC \tag{2}$$

$$\begin{aligned} TF5_{image} \leftarrow & ABC + ABD + ABE \\ & + ACD + ACE + ADE + BCD \\ & + BCE + BDE + CDE \end{aligned} \tag{3}$$

$$\begin{aligned} TF7_{image} \leftarrow & ABCD + ABCE + ABCF \\ & + ABCG + ABDE + ABDF + ABDG \\ & + ABEF + ABEG + ABFG + ACDE \\ & + ACDF + ACDG + ACEF + ACEG \\ & + ACFG + ADEF + ADEG + ADFG \\ & + AEFG + BCDE + BCDF + BCDG \\ & + BCEF + BCEG + BCFG + BDEF \\ & + BDEG + BDFG + BEFG + CDEF \\ & + CDEG + CDFG + CEFG + DEFG \end{aligned} \tag{4}$$

The equations evidently show that the number of boolean operations increase exponentially as the number of frames per tournament increases. Thus, although the TF can be theoretically extended to $TF9$ and higher configurations by applying a similar pattern in the formula, it is not practical to go beyond $TF7$ (Although it is possible to reduce the number of boolean operations in the formula for higher TF configurations by applying some algebraic simplification, the problem of minimizing the number of boolean operations for a given boolean formula is, unfortunately an NP-Hard problem [6]). This finding revealed a limitation in extending the TF algorithm to greater tournament sizes.

RGB and HSV Colour Spaces. The TF algorithm was originally applied on the RGB colour model. In this study we explore the extendibility of the TF algorithm on the HSV colour space. A short discussion of the 2 colour models is first presented in this section.

The RGB colour model defines an image pixel value using 3 component images or channels, one for each primary colour Red, Blue and Green. When these images are to be displayed to a monitor, the three component images are combined. Each colour is represented by an 8-bit colour depth with values of 0-255 (per color channel). Combining the 3 colour channels provides a 24-bit full-colour image allowing for $(2^8)^3 = 16,777,216$ distinct colours.

The RGB colour model depicts how the human visual system works and best suits hardware implementations. However, it does not match the human intuitive way of describing colours. In describing colours, the HSV colour model is preferred.

Similar to the RGB colour model, the HSV also uses 3 values to describe a colour. These 3 measures are the Hue, Saturation and Value. The Hue value is a measure that describes the pure colour of the image, e.g., pure red, orange or yellow. These pure colours are diluted with white light and is described by the Saturation value. The measure of Value follows the same principles as grey images which have shades of grey that runs from white to black.

The HSV measures can be extracted from the RGB colour model. For the implementation of this study, the conversion from RGB to HSV colour model used the built-in function `cvtColor()` of the OpenCV library. Using the normalised RGB values, the HSV values are computed using Eq. 5 from [29].

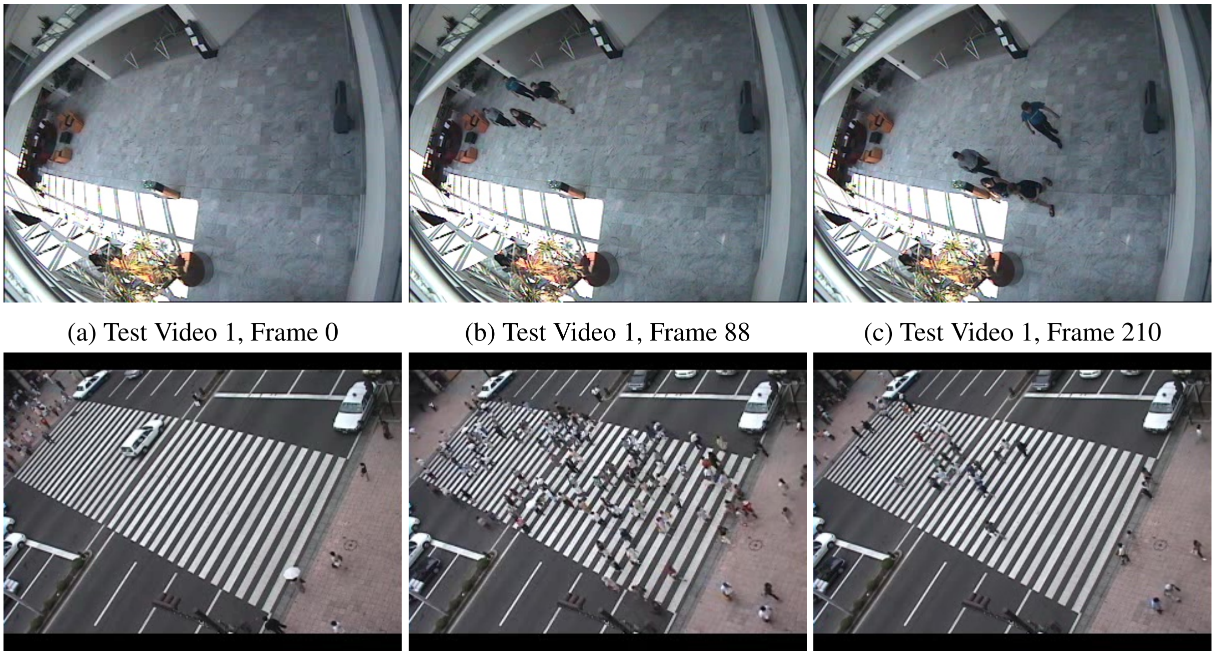
$$\begin{array}{l} V \leftarrow max(R,G,B) \\ S \leftarrow \begin{cases} \frac{V - min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \\ H \leftarrow \begin{cases} \frac{60(G-B)}{V - min(R,G,B)} & \text{if } V = R \\ \frac{120+60(B-R)}{V - min(R,G,B)} & \text{if } V = G \\ \frac{240+60(R-G)}{V - min(R,G,B)} & \text{if } V = B \end{cases} \end{array} \tag{5}$$

The range of the intermediate HSV values are listed in Eq. 6. These intermediate values are each converted to their equivalent 8-bit image representation using Eq. 7. A detailed discussion on the colour space conversion is available in [8].

$$\begin{array}{ccccc} 0 & \leq & V & \leq & 1 \\ 0 & \leq & S & \leq & 1 \\ 0 & \leq & H & \leq & 360 \end{array} \tag{6}$$

$$\begin{array}{l} V \leftarrow 255V \\ S \leftarrow 255S \\ H \leftarrow \frac{H}{2} \end{array} \tag{7}$$

After all manipulations have been made in the HSV values, the HSV values are converted back to their equivalent RGB values, ready for display. This is because the RGB colour model is best used for colour generation and hardware display, while the HSV colour model works better in extracting the colour descriptions.



(a) Test Video 1, Frame 0 (b) Test Video 1, Frame 88 (c) Test Video 1, Frame 210

(d) Test Video 2, Frame 0 (e) Test Video 2, Frame 1268 (f) Test Video 2, Frame 1586

Fig. 2: Test Videos Sample Frames.

4 Methodology

The random sampling, boolean computation and multi-level processing of the TF which were implemented in the previous study [2] were re-used in this study. An additional step involving the conversion of the image frame from the RGB to the HSV colour space was performed prior to actual TF processing of the sample frames. Fig. 1(a) illustrates the general sample and level processing of the TF algorithm. Specifically, 3 TF configurations were used for implementation, i.e., $TF_{3,4}$, $TF_{5,3}$ and $TF_{7,2}$ as illustrated in Fig. 1(b-d). The main difference in the implementation is performing the tournament-like strategy in the HSV color channels instead of the RGB color channels.

Two (2) test videos from [2] were used in the HSV simulation. The summary details of these videos are listed in Table 1. Video 1 was selected to represent videos where there is only sparse foreground, while Video 2 was selected to represent the more challenging case of having more foreground (both in terms of proportion of frames having foreground and average number of foreground objects per frame). Sample frames are shown in Fig. 2.

An HSV ground truth generator (GTGenHSV) was also implemented to compute the ideal model background in the HSV space. This GTGenHSV generated image is also used as a reference background to determine the accuracies of TF on HSV colour mode implementation. Two (2) additional GTGen images were generated – GTGen modal pixel integer mode and GTGen modal bit mode – in order to gather more insights about the performance of each of the different TF configurations. It should be reiterated that the assumed ideal model background is the modal pixel integer; the modal pixel bit is just an approximation. In many cases these two are the same. A sample difference between the pixel integer mode and pixel bit mode is illustrated in Table 2.

Table 1: Test video parameters.

Test Video	size (MB)	length (secs)	frame count	width (pixel)	height (pixel)
1 [28]	8.6	20	498	384	288
2 [31]	18.2	60	1740	320	240

Theoretical results were derived and empirical results were gathered for both ground truth and TF configurations. These help determine the performance of the TF tournament-like strategy on the HSV colour space. The expected accuracies were derived using the equations for p_0 (corresponding to the probability of getting a correct modal bit using an initial frame) and p_i (probability of getting a correct modal bit after i multi-level processing) presented in [2]. The actual accuracies were also obtained to verify the computed expected theoretical accuracies, using two test videos that were also used in [2]. To obtain the empirical accuracies, for both RGB and HSV colour spaces,

Table 2: Pixel Integer Mode vs Pixel Bit Mode.

Pixel	Pixel Value (R Channel)								
(r,c)	Integer	Equivalent 8-bit value							
Frame 1	240	1	1	1	1	0	0	0	0
Frame 2	240	1	1	1	1	0	0	0	0
Frame 3	240	1	1	1	1	0	0	0	0
Frame 4	7	0	0	0	0	0	1	1	1
Frame 5	7	0	0	0	0	0	1	1	1
Frame 6	255	1	1	1	1	1	1	1	1
Frame 7	255	1	1	1	1	1	1	1	1
Mode	**240**	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**
R Value	**240**	**247**							

the generated background images of the TF configurations were compared to the GTGen. Four (4) GTGen model background image were generated – RGB integer mode, RGB bit mode, HSV integer mode and HSV bit mode denoted here as $gtRGB_{int}$, $gtRGB_{bit}$, $gtHSV_{int}$, $gtHSV_{bit}$ respectively. All configurations for both RGB and HSV were compared to their respective GTGen images. Finally, to illustrate the competitiveness of the TF algorithm on HSV, a background subtraction was implemented together with a comparison with popular baseline methods.

5 Results

5.1 Ground Truth Generator

The TF ground truth generator (GTGen) [2] was reused as the model background image. To determine the empirical accuracies of the TF on HSV generated images, a GTGen for HSV was also developed for both integer and bit level processing. Fig. 3 illustrates the GTGen process for both RGB and HSV colour space as well as the main difference between the integer and bit based pixel value.

Four (4) GTGen images were used as model backgrounds to determine the accuracies of the TF configurations for both RGB and HSV based computation. These GTGen images are denoted here as: $gtRGB_{int}$, $gtRGB_{bit}$, $gtHSV_{int}$ and $gtHSV_{bit}$. The $gtRGB$ and $gtHSV$ are GTGen implementations using the RGB and HSV channels respectively. The difference in gt_{int} and gt_{bit} can be inferred from the example in Table 2. The per bit location processing is done for all bit positions 0 to 7 on the equivalent 8-bit value of the integer pixel value computations.

The generated GTGen for both RGB and HSV are illustrated in Table 3.

5.2 Performance

5.2.1 Theoretical Accuracy

The expected accuracy can be derived using a probabilistic estimation with an assumption that the modal pixel value at each pixel location is the background [1] [2]. The expected accuracy for each bit, given S frames per tournament and after i levels of processing, is given by

$$p_i = \sum_{k=\lceil \frac{S}{2} \rceil}^{S} \binom{S}{k} (p_{i-1})^k (1-p_{i-1})^{S-k} \qquad (8)$$

The base case p_0 is simply derived from the frequency of the modal bit with respect to the total number of frames:

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

Given Eq. 8, the expected accuracy can be computed for any TF configuration, for any tournament size S and level processing L. Theoretically, with a higher initial probability value p_0, a higher resulting accuracy is expected. Moreover, with a fixed tournament size S, the expected accuracy increases as the number of levels L increase.

5.2.2 Empirical Accuracy

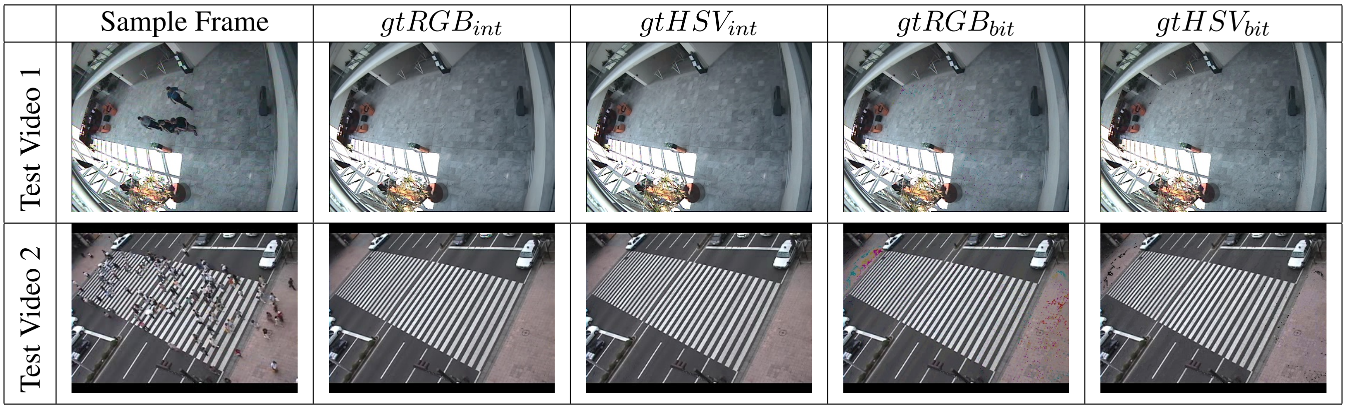
Using the 2 test videos, the actual p_0 values were examined in this study. This examination was performed exhaustively, that is for each bit position and each pixel location across all frames in the video. The summary values are given in Table 4, where the mean and standard deviation of the p_0 for every RGB channel for Test Video 1 are listed. A graph of the mean p_0 for each bit position is further illustrated in Fig. 4 to highlight an important finding: The most significant bits (bits 4 to 7) have high mean p_0 values. This implies that even if the least signifiant bits may have lower p_0 values, when a background pixel is constructed using the derived modal bit values, there is a good probability that its integer value per channel approximates the integer value of the ideal model background on the same colour channel. The 2 most significant bits also exhibit low standard deviation of 0.09-0.11 (while the 6 least significant bits show a slightly higher variability of 0.12-1.15) to further add stability to the TF method. Interestingly, these patterns are consistent across all the three colour channels of RGB.

The empirical accuracies of the TF configurations on HSV channels are illustrated in Fig. 5. These results verify the expected accuracies derived. Threshold values ranging from 0 to 10 were used to compare

Table 3: Ground Truth Generation Pixel Value Processing.

Image Pixel Location (r,c)	Integer Vaue						One(1) Bit Position Value					
	R	G	B	H	S	V	R	G	B	H	S	V
Frame 1	69	161	113	62	37	118	1	1	1	1	0	0
Frame 2	67	161	113	62	55	118	1	1	0	1	1	1
Frame 3	68	10	113	55	55	90	0	1	0	0	1	0
Frame 4	67	10	112	42	5	248	1	0	0	0	1	0
Frame 5	67	161	112	30	28	120	0	1	0	0	0	1
Pixel Mode	**67**	**161**	**133**	**62**	**55**	**118**	**1**	**1**	**0**	**0**	**1**	**0**
Notation	$\mathbf{RGB_{int}}$			$\mathbf{HSV_{int}}$			$\mathbf{RGB_{bit}}$			$\mathbf{HSV_{bit}}$		

Fig. 3: $GTGen$ images.



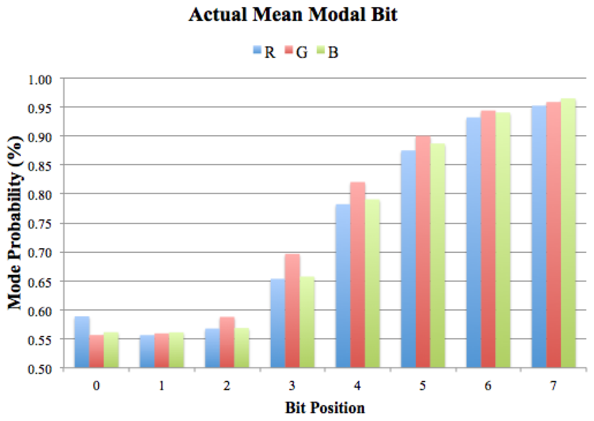


Fig. 4: Actual Mean Mode Probability for Test Video 1.

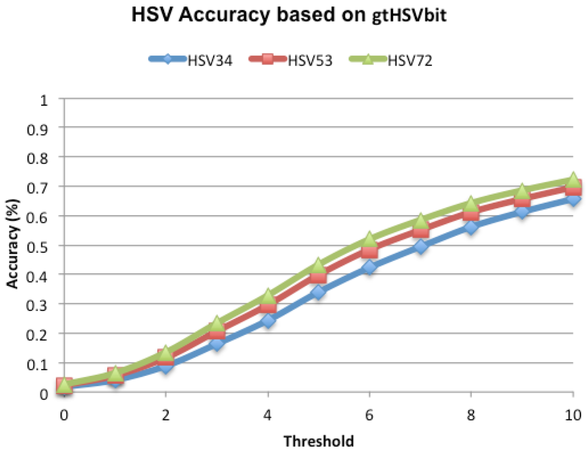


Fig. 5: Graph of HSV Accuracy based on $gtHSV_{int}$

the results with a ground truth. Threshold 0 indicates that the TF generated images are completely identical to that of the TF GTGen images. A threshold of 10 indicates that the pixel values differ by not more than 10 (in terms of resulting integer values) in each of the colour channels.

A comparison of the RGB and HSV accuracies are shown in Fig. 6. From the graph, the TF implementation on the HSV colour channels has a lower accuracy level compared to RGB channels.

5.3 Background Image Generation

The resulting background images of the TF configurations shows that after Level 1 processing, the generated background images still have visible moving

Table 4: Actual $GTGen$ Modal Value Probabilities for Test Video 1.

Bit	Mean (%).			SD (%).		
Pos	R	G	B	R	G	B
0	0.5888	0.5566	0.5615	0.1205	0.1271	0.1248
1	0.5565	0.5595	0.5613	0.1218	0.1272	0.1251
2	0.5678	0.5879	0.5687	0.1228	0.1289	0.1258
3	0.6536	0.6966	0.6575	0.1351	0.1448	0.1349
4	0.7823	0.8206	0.7907	0.1574	0.1529	0.1543
5	0.8750	0.9002	0.8870	0.1471	0.1342	0.1428
6	0.9320	0.9439	0.9404	0.1180	0.1084	0.1138
7	0.9523	0.9586	0.9649	0.0983	0.0944	0.0931

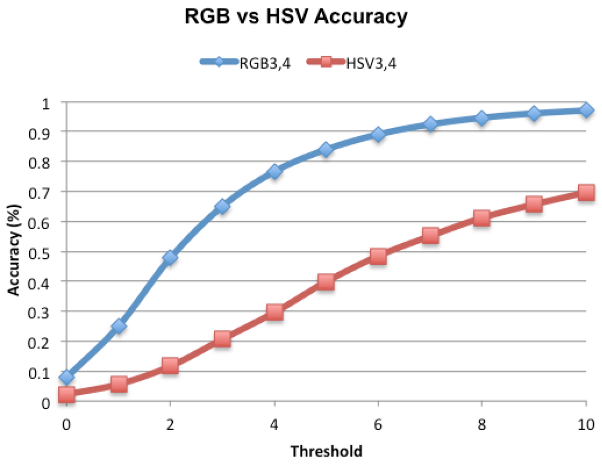


Fig. 6: Graph of RGB vs HSV accuracy on $TF_{3,4}$

object. As the number of processing level increases, the “cleaner” is the resulting top level images. These are comprehensively explained in [2], with TF implemented on the RGB color channels.

On the other hand, the same background image generation and TF configurations were implemented on the HSV colour space. The resulting HSV background images are shown in Fig. 7, depicted side by side with the corresponding TF generated images based on RGB colour channels. As illustrated, the TF implementation on HSV has good performance, similar to that of TF on RGB.

5.4 Background Subtraction

To further test the efficiency of the TF on HSV generated images, a background subtraction was performed. The difference of the background images and an observed sample frame results to a binary image. The binary images for both integer and bit based computation are shown in Table 5. In a particular pixel location (r,c), a pixel difference of zero is indicated by a binary value 1 and a pixel difference greater than zero is indicated by the binary value 0. The final binary image is the majority bit of the color channels in

Fig. 7: Generated background images of TF in RGB and HSV.

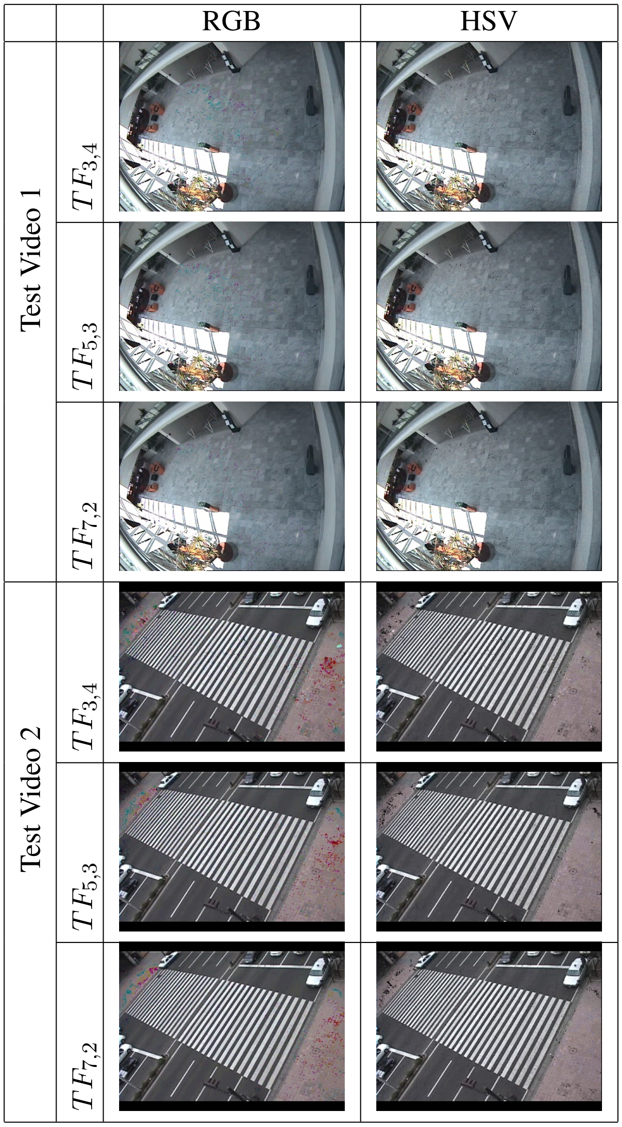


Table 5: Background Subtraction Pixel Value Processing.

Image Pixel Location (r,c)	Integer Vaue						One(1) Bit Position Value					
	R	G	B	H	S	V	R	G	B	H	S	V
Background	69	161	113	62	37	118	1	1	1	1	0	0
Observed	67	161	113	62	55	118	1	1	0	1	1	1
Subtracted(1 if diff=0; 0 if diff>0)	0	1	1	0	0	1	1	1	0	1	0	0
Binary (subtracted majority)	1			0			1			0		

the subtracted images. Thus, a foreground pixel 0 is shown as a black coloured pixel in the binary image and a background pixel 1 is shown as white. For a fair comparison, no de-noising or any post-processing techniques were implemented on the binary images.

The resulting background subtraction images are illustrated in Fig. 8. These difference images were compared with commonly used baseline methods – MOG, Mean, Median and Mode, and the TF on RGB subtracted images in [2]. These techniques were used in literature [5], [9], [13], [16], [19] and [27] as a basis for some of the current promising background subtraction algorithms. The MOG2 algorithm used for comparison was proposed by Zivkovic in [27] and is the built-in BackgroundSubtractorMOG2( ) in OpenCV library [29]. The mean, median and mode algorithms were developed for comparison by getting the mean, median and mode pixel value across all frames in the given video sequence. The comparison of the TF on HSV colour space to these baseline methods are illustrated in Fig. 8.

6 Conclusion

In this paper, we have shown that the TF algorithm can be extended to the HSV colour space. The TF bit based processing over the usual pixel integer mode was clearly differentiated for both RGB and HSV implementations. A ground truth based on HSV was developed to determine the empirical accuracies of the generated TF based on HSV images. Comparing the RGB and HSV results, the RGB implementation remains to be the more accurate between the two. Based on the actual p_0 obtained, the HSV values shows high variability. Moreover, the HSV implementations require a lot more processing time because of the RGB-HSV conversions. Furthermore, the conversion from RGB to HSV, and then back to RGB colour values have introduced discrepancies in the values that have contributed to the decrease in the accuracy.

Further improvements on this study may include examining the actual p_0 on the HSV colour space as well as the the discrepancies due to the RGB to HSV conversion.

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, *Philippine Information Technology Journal*, vol. 6, no. 1, 2013, pp.4-10.

[2] P. Abu and P. Fernandez, Extendibility of the Teknomo-Fernandez Algorithm for Background Image Generation, *WSEAS Int. Conf. on Applied Computer and Applied Computational Science*, 2014, pp. 28–37, ISBN: 978-960-474-368-1.

[3] K. Appiah, A. Hunter, and T. Kluge, GW4: A Real-Time Background Subtraction and Maintenance Algorithm for FPGA implementation, *WSEAS Transactions on Systems*, vol. 4 no. 10, 2005, pp.1741–1751.

[4] 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.

[5] R. Cucchiara, C. Grana, and M. Piccardi, Detecting Moving Objects, Ghost and Shadows in Video Streams, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25 no.10, 2003, pp.1337–1342.

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

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

[8] R. Gonzalez and R. Woods, *Digital Image Processing, 3rd ed.* Prentice-Hall, Inc., 2006.

[9] 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.

[10] I. Haritaoglu, D. Harwood and, L.S. Davis, W4: real-time surveillance of people and their activities, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22 no. 8, 2000, pp.809-830.

[11] C. Ianasi, V. Gui, F. Alexa and C. Toma, Fast and Accurate Background Subtraction for Video Surveillance, Using an Adaptive Mode-Tracking Algorithm, *WSEAS Int. Conf. on Dynamical Systems and Control*, 2005, pp.391-397.

[12] C. Ianasi, V. Gui, F. Alexa, and C. I. Toma, Noncausal adaptive mode tracking estimation for background subtraction in video surveillance, *WSEAS Transactions on Signal Processing*, vol. 2 no. 1, 2006, pp.5259.

[13] K. Kim, T. Chalidabhongse, D. Harwood, L. Davis, Real-time foreground-background segmentation using codebook model, *Real Time Imaging*, vol. 11 no. 3, 2005, pp.172-185.

[14] N. Li, D. Xu, and B. Li, A novel background updating algorithm based on the logical relationship, *WSEAS Int. Conf.on Signal, Speech and Image Processing*, 2007, pp.154–158.

[15] Y. Liu, H. Yao, W. Gao, X. Chen, and D. Zhao, Nonparametric Background Generation, *J. Vis. Comm. Image Rep.*, vol. 18 no. 3, 2007, pp. 253–263.

[16] L. Maddalena and A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, *IEEE Trans. on Image Processing*, vol. 17 no. 7, 2008, pp.1168–1177.

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

[18] A. Remenyi, S. Szenasi, I. Bandi, Z. Vamossy, G. Valcz, P. Bogdanov, S. Sergyn and M. Kozlovszky, Parallel biomedical image processing with GPGPU-s in cancer research", *3rd IEEE International Symposium on Logistics and Industrial Informatics (LINDI 2011)*, Budapest, 2011, pp. 245-248.

[19] 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.

[20] K. Sundaraj, Real-Time Face Detection using Dynamic Background Subtraction,*WSEAS Transactions on Information Science and Applications*, vol. 5 no. 11, 2008, pp.1531-1540.

[21] K. Sundaraj, Real-Time Background Subtraction using Adaptive Thresholding and Dynamic Updating for Biometric Face Detection, *WSEAS Transactions on Computers*, vol. 7 no. 10, 2008, pp.1762-1771.

[22] S. Szenasi, Z. Vamossy and M. Kozlovszky, Preparing Initial Population of Genetic Algorithm for Region Growing Parameter Optimization, *4th IEEE International Symposium on Logistics and Industrial Informatics: LINDI 2012*, Smolenice, 2012, pp. 47-54.

[23] S. Szenasi, Distributed Region Growing Algorithm for Medical Image Segmentation, *International Journal of Circuits, Systems and Signal Processing*, vol. 8, no. 1, 2014, pp.173-181, ISSN 1998-4464.

[24] K. Teknomo and P. Fernandez, Background Image Generation Using Boolean Operations, *Philippine Computing Journal*, vol. 4 no. 2, 2009, pp.43–49.

[25] 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.

[26] 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*, vol. 19 no. 7, 1997, pp.780–785.

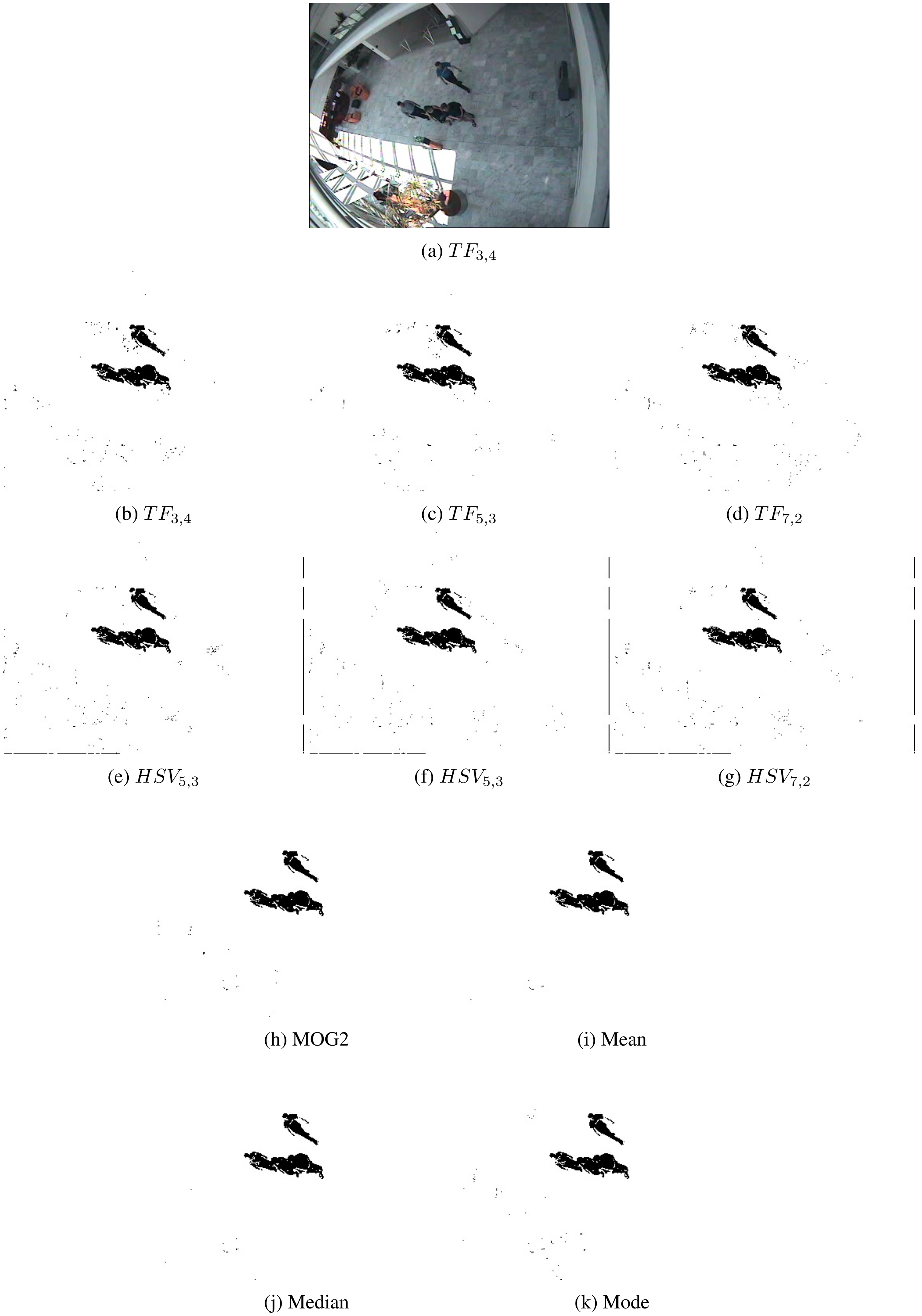
[27] Z. Zivkovic, Improved adaptive Gausian mixture model for background subtraction, *Int. Conf. on Pattern Recognition*, 2004.

[28] 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/. February 24, 2014

[29] The OpenCV Reference Manual Release 2.4.9.0, 2014. Available from URL: http://docs.opencv.org/opencv2refman.pdf. February 24, 2014.

[30] 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. February 24, 2014

[31] K. Teknomo. Available from URL: http://people.revoledu.com/kardi/download/-download.php?file=PedestrianMovie1. February 24, 2014



(a) $TF_{3,4}$

(b) $TF_{3,4}$

(c) $TF_{5,3}$

(d) $TF_{7,2}$

(e) $HSV_{5,3}$

(f) $HSV_{5,3}$

(g) $HSV_{7,2}$

(h) MOG2

(i) Mean

(j) Median

(k) Mode

Fig. 8: Background subtraction binary images.