

## **Foreground Object Detection Based on Chrominance and Texture Features with Enhancement by Canny Filter**

**Muna Ghazi Abdul Sahib**

[110096@uotechnology.edu.iq](mailto:110096@uotechnology.edu.iq)

Computer Science Department, University of Technology, Iraq,

### **Abstract**

The foreground object detection became very important in a computer vision system and has a many applications such as recognition, object tracking, counting, classifying, home surveillance, traffic monitoring, video monitoring, medical image and in other multimedia applications. So that each of these applications needs a method for object detection, therefore, requires improving new methods and algorithms for processing this information. This paper proposes foreground objects detection approach based on the chrominance and texture features with canny enhance filter. The input is background image and current image and the output are the detecting foreground objects. The proposed approach consists of three steps: first the features extracting which are chrominance and texture features (these features are robust against to illumination changes, noise, and shadows) from a current and background image. Then, the similarity matching is computed for each feature. Finally, canny

filter are used to enhance the results. Furthermore, we evaluate our approach using evaluation measures which are precision, recall, and F-measure, to give 0.922 as an average accuracy of the proposed method and with average consumption time about 0.5778923 seconds. This concludes that proposed method very efficient against the limitation of challenges and obstacles.

**Keywords:** Foreground Object Detection, Background Subtraction, Chrominance Feature, Texture Feature, Canny Filter.

#### المستخلص

أصبح الكشف عن الكائنات الامامية مهمًا جدًا في نظام الرؤية بالحاسوب ولديه مجموعة واسعة من التطبيقات مثل تتبع الكائنات ، والتمييز ، والعد ، والتصنيف ، والمراقبة المنزلية ، ومراقبة حركة المرور ، ومراقبة الفيديو ، والصورة الطبية ، وتطبيقات الوسائط المتعددة الأخرى. لذلك كل واحدة من هذه التطبيقات تحتاج إلى طريقة لاكتشاف الكائن ، وبالتالي ، يتطلب تحسين طرق وخوارزميات جديدة لمعالجة هذه المعلومات. في هذا البحث ، الطريقة المقترحة تستخدم خصائص اللون و القوام مع تحسينها من خلال canny filter لكشف الكائنات الأمامية. المدخلات هي صورة المقدمة وصورة الخلفية و المخرجات هي الكائنات الامامية المكتشفة. تتكون الطريقة المقترحة من ثلاث خطوات: أولاً استخلاص الخصائص و هي خاصية اللون و القوام (هذه الخصائص قوية ضد تغيرات الإضاءة والضوضاء والظلال) من صورة المقدمة والخلفية. ثم ، يتم حساب مطابقة التشابه لكل ميزة. وأخيرًا ، لتحسين النتائج ، يستخدم في هذا البحث فلتراً وهو canny. بالإضافة الى ذلك ، قمنا بتقييم طريقتنا باستخدام مقاييس التقييم التي هي الدقة ، والتذكير ، والمقياس F ، حيث اعطى 0.922 كمتوسط دقة للطريقة المقترحة وكان متوسط زمن المستغرق حوالي 0.5778923 ثانية. نستخلص من هذا أن الطريقة المقترحة فعالة للغاية ضد تقييد التحديات والعقبات.

## **1. INTRODUCTION**

Recently, detection foreground objects is one of the most fundamental and important tasks in many fields in computer vision and also play an important role in a wide range of applications such as recognition, object tracking, classifying, counting, visual surveillance, home surveillance, traffic surveillance, video monitoring, medical image and in another multimedia applications [1].

There have been many efforts to achieve high performance, accurate and robust method in order to overcome the obstacles such as shadows, dynamic backgrounds, illumination changes and noises. The most popular methods are used for objects detection likes: temporal difference method, optical flow method, segmentation method, background subtraction and frame difference method. In this paper proposes detecting foreground object method based on chrominance and texture features then enhancement by using canny filter [1, 2].

This paper is structured as follows: section 2 shows the related works; sections 3, 4 and 5 cover a background of foreground object detection, feature extraction and canny filter, respectively; section 6 describes the proposed method of foreground object detection; section 7 presents the experimental results, and finally section 8 clarifies the conclusions.

## **2. RELATED WORKS**

There are many types of research focusing on foreground object detection. Some of these types include:

S. Adhikari et al., 2014 [1] Proposed a foreground object extraction technique by using the Canny to detect the edges of the objects then fill the region that bounded by the edges by using the concept of boundary defined region.

P. Gujrathi et al., 2014 [2] Proposed detection moving the object based on the background subtraction in FPGA. The hardware architecture is proposed by implemented in FPGA. In this method to detect the object's edges used Sobel edge and then filtered the images to remove the noise. In order to classify the objects based on their shapes, the object edges are used.

M. Bhagyashri and A. Makwana , 2015 [3] present a method of background subtraction based on blocks. In which the image is divided into blocks and detect objects by using the variance between pixels of the blocks and the color histogram. Then, used the Gaussian Mixture to refine the detected foreground.

J. Son et al., 2016 [4] proposed for video surveillance an approach of fast illumination robust foreground detection. This method consists of foreground detection and candidate. Which, detect the location of blocks from the estimated distribution map and then detected from the extracted valid blocks.

A.M. Rahma and N. B. Abd., 2017 [5] Proposed a method to detect the moving object by using invariant moments. To

define the region of interest used background subtraction and then applied invariant moments for the region of interest and according to the database information, the moving car will be detected and recognized.

D.Yang et al., 2018 [6] Proposed an approach for foreground detection which is depends on the stability of adaptive multi attributes for background model. In this method, each pixel is described by multi features and then each feature measured its stability by its histogram statistics. Then this will be used to produce the final label.

### **3. FOREGROUND OBJECT DETECTION**

Foreground detection also called a background subtraction, the main idea of foreground objects detection is to compare the foreground image or current image with a background model and detect regions that are different. The performance of foreground detection based on background model. A background model must overcome the problems such as illumination changes, shadows, and noise. The foreground object detection is important fields in computer vision and used in many application such as recognition, object tracking, home surveillance, traffic surveillance and in a medical image like brain tumor [7, 8].

To get moving objects there are several methods are used such as: temporal difference method this method not used to detect static object, second method is optical flow method in this method take the region in which the brightness changes this

method suitable for the static and dynamic background, also can be used segmentation method, another method is background subtraction method which detects the motion object by finding the change between current image and background image [9]. The most popular used background subtraction methods are frame difference method, a mixture of Gaussian method, Eigen background method and approximated median filter method [5,9].

#### **4. FEATURE EXTRACTION**

The purpose of extracting feature is to extract important information. The chrominance and texture features will be extracted from the current and background image. These features are robust to noise, shadows and illumination changes [10]. Then perform matching between the features resulted to detect the foreground objects. The details of chrominance and texture features are defined as the following:

##### **4.1 CHROMINANCE FEATURE**

Two color spaces are used as features helps to detect foreground objects YUV and HSL color spaces. The YUV and HSL color model separation the chrominance component from luminance this have advantages to distinguish shading, illumination changes and noise from foreground objects, Because of chrominance component is independent of the luminance component and vice versa [11]. The details of YUV and HSL color spaces are defined as the following:

#### 4.1.1 YUV COLOR SPACE

The YUV color space separates the luminance from chrominance component this ability have advantages in applications such as foreground object detection. The YUV color space is as similar as human eye's retina. In YUV the Y channel represented the luminance which describes the intensity of light, while U and V channels represented the chrominance parts which carry the information of color. To obtain YUV from RGB the equations are as follows [12, 13]:

$$\begin{aligned}
 Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\
 U &= -0.147 \times R - 0.289 \times G + 0.436 \times B \quad \dots \dots \dots (1) \\
 V &= 0.615 \times R - 0.515 \times G - 0.100 \times B
 \end{aligned}$$

#### 4.1.2 HSL COLOR SPACE

The HSL color space is also known as HLS or HIS. HSL color space is abbreviated to Hue, Saturation, and Lightness. The HSL color space also separates the luminance from chrominance component so this have advantages when used in applications such as foreground object detection. In HSL the hue (H) channel represented the chrominance component, while the saturation (s) channel represented the measure of whiteness and the lightness (L) is the luminance which provides a measure of the brightness. The HSL include HSI is abbreviated to (Hue Saturation Intensity), HSV is abbreviated to (Hue Saturation Value) and HCI is abbreviated to (Hue Chroma Intensity) [12]. To obtain HSL from RGB will using HSI type color spaces and the equations are as follows [10]:

$$H = \cos^{-1}(0.5(R - G) + (R - B)) / (((R - G)^2 + (R - B)(G - B))^{0.5})$$

$$S = 1 - \left( \frac{3}{R + G + B} \right) \times \min(R, G, B) \dots \dots \dots (2)$$

$$I = (R + G + B) / 3$$

#### 4.2 TEXTURE FEATURE

The texture feature is robust to shadows and illumination changes. To extract texture feature there are several methods are used, one of the most used methods is the Local Binary Pattern (LBP) because it has the advantage of fast performance and simple implementation. The LBP operator worked for each pixel in an image with the 8-neighbors of a pixel in a block of  $3 \times 3$  pixels, which using the center pixel value as a threshold, so will be produced the binary code [14, 15]. LBP operator is defined as:

$$TLBP(x_c, y_c) \sum_{n=0}^{P-1} s(g_n - g_c) 2^n \dots \dots \dots (3)$$

For a given  $N \times M$  image size  $x_c \in \{0, \dots, N - 1\}$ ,  $y_c \in \{0, \dots, M - 1\}$ . Whereas  $g_c$  represents the center pixel grey value at the coordinates  $(x_c, y_c)$  and  $g_n$  represents the P neighborhood pixels grey values.

Where  $s(f)$  is the function of thresholding which is defined as the following:

$$s(f) \begin{cases} 1 & \text{if } f \geq 0 \\ 0 & \text{if } f < 0 \end{cases}$$

So when the value of neighbor's is greater than the value of center pixel's writes "1", otherwise, writes "0". This finally gives a binary code of 8-digit.



## 5. CANNY FILTER

Canny is one of the methods to detect the edges. The edge represents the boundary between the object and the image background. The edge detection is used to decrease the amount of data in the image so the edges are a robust feature for foreground object detection [13].

Canny Edge Detection is based on three criteria defined as the following [16, 17]:

- **Good Detection:** The probability of detecting false edge points must be low and the probability of detecting true edge points must be high.
- **Good Localization:** The points detected as edges must be as close to the true edges as possible.
- **A number of responses:** In case when there are two edges one of them must be considered false.

The Canny filter consists of five steps [7, 16, 17 ]: The first step is smoothing where the

The image is blurred for noise removal. The second step is finding gradients by the applied Gaussian mask to convolve with the image.

The equation for compute the magnitude of the gradient  $g(x, y)$  is:

$$M(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2} \quad \dots \dots \dots (4)$$

While the equation to find the direction of the edge of the maximum slope is the angle  $\alpha(x, y)$  is:

$$\alpha(x, y) = \tan^{-1} \left[ \frac{g_y(x, y)}{g_x(x, y)} \right] \dots \dots \dots (5)$$

Where  $g_x(x, y)$  represent the gradient estimation in the  $x$  direction at  $(x, y)$  pixel and  $g_y(x, y)$  represent the gradient estimation in the  $y$  direction at  $(x, y)$  pixel.

The third step is non-maximum suppression, is performed to the gradient magnitude to thin the edge by keeping only the local maxima values of the gradient magnitude. This is done by check each pixel in the image  $I(x, y)$ , If either of the two neighboring pixels in the direction of the gradient are smaller in magnitude than the current pixel then  $I(x, y) = g(x, y)$  else  $I(x, y) = 0$ .

The next step is applied double thresholding involves high and low threshold. Finally, hysteresis is applied to detect the edges. This step is performed to each pixel, If the value of the pixel's gradient is above the high threshold will be retained and marked as a strong edge pixel. If the value of the pixel's gradient is below the low threshold will be removed. If the value of the pixel's gradient is below the high threshold and above the low threshold will be retained only if the pixel is connected to the strong pixel where its gradient value is above the high threshold, then marked as a weak edge pixel. The output of the hysteresis provides the final edges.

So the canny edge detector detects the edges by removing all the other edges that are weak and not connected to strong edges.

## **6. THE PROPOSED FOREGROUND OBJECT DETECTION METHOD**

This paper proposed method for detection foreground object using background model based on chrominance and texture features then enhancement by the canny filter. In this proposes method ,the video is convert to sequence of frames, then get the first frame from the video and considered as the background image and consider any frame want to extract objects from it as a current image. This method consists of three steps, including: features extraction, similarity matching, and enhancement. The input is a current image and background image and the output are the detecting foreground objects. The proposed foreground object detection method is strong against to illumination changes, noise, and shadows. Figure (1) illustrates the framework of the proposed method. The details of each step are shown in the following:

**a) Features Extraction :**

In the proposed method the first step is feature extracting. This is a fundamental processing step in foreground object detection method. Where the features are generated from both current image and background image. The extracted features include chrominance and texture features. The chrominance feature will be extracting from YUV and HSL color space. The YUV and HSL color model separation the luminance part from

chrominance this have advantages to distinguish shading, illumination changes and noise from foreground objects.

In YUV the U and V channels will be obtained because it represents the chrominance components which carry the color information. The U and V will be computed as illustrated in equation (1) it is presented in subsection (4.1.1).

In HSL will using HSI type color spaces. In HSI the H channel will be obtained because it represents the chrominance components which carry the color information. The H will be computed as illustrated in equation (2) it is presented in subsection (4.1.2).

The texture feature will be extracted by applying the method of Local Binary Pattern (LBP). Because it has the advantage to distinguish shading, illumination changes and noise from foreground objects, so it used in the proposed method. The TLBP (Texture Local Binary Pattern) will be computed as illustrated in equation (3) it is presented in subsection (4. 2).

**b) Similarity Matching:**

The second step in the proposed method is the similarity matching. This step matches the similarity between current image features and background image features to determine the foreground objects by using chrominance and texture features. The matching measure is computed for each feature (U and V from YUV, H from HSI, LBP the texture) then will compute the matching measure that combines all of these features matching

measures to improve the results. The outputs of this step are foreground objects. The matching measure is computed for each feature as the following:

1–The matching measures  $Mu(i, j)$  and  $Mv(i, j)$  for chrominance feature of YUV is computed as follows:

$$Mu(i, j) = \begin{cases} U^F(i, j)/U^B(i, j) & \text{if } U^B(i, j) > U^F(i, j) \\ 1 & \text{if } U^B(i, j) = U^F(i, j) \\ U^B(i, j)/U^F(i, j) & \text{if } U^B(i, j) < U^F(i, j) \end{cases} \dots(6)$$

$$Mv(i, j) = \begin{cases} V^F(i, j)/V^B(i, j) & \text{if } V^B(i, j) > V^F(i, j) \\ 1 & \text{if } V^B(i, j) = V^F(i, j) \\ V^B(i, j)/V^F(i, j) & \text{if } V^B(i, j) < V^F(i, j) \end{cases}$$

Where  $U^F$  and  $V^F$  represents U and V in the current image, while  $U^B$  and  $V^B$  represents U and V in the background image. For  $N \times M$  image size  $i \in \{0, \dots, N - 1\}, j \in \{0, \dots, M - 1\}$ .

2– The matching measure  $Mh(i, j)$  for chrominance feature of HSL color space is computed as follows:

$$Mh(i, j) = \begin{cases} H^F(i, j)/H^B(i, j) & \text{if } H^B(i, j) > H^F(i, j) \\ 1 & \text{if } H^B(i, j) = H^F(i, j) \\ H^B(i, j)/H^F(i, j) & \text{if } H^B(i, j) < H^F(i, j) \end{cases} \dots (7)$$

Where F represents current image and B represents background image. For  $N \times M$  image size  $i \in \{0, \dots, N - 1\}, j \in \{0, \dots, M - 1\}$ .

3–The matching measure  $Mt(i, j)$  for texture feature is computed as follows:

$$Mt(i, j) = \begin{cases} T^F(i, j)/T^B(i, j) & \text{if } T^B(i, j) > T^F(i, j) \\ 1 & \text{if } T^B(i, j) = T^F(i, j) \\ T^B(i, j)/T^F(i, j) & \text{if } T^B(i, j) < T^F(i, j) \end{cases} \dots\dots\dots (8)$$

Where  $F$  represents current image and  $B$  represents background image. For  $N \times M$  image size  $i \in \{0, \dots, N - 1\}, j \in \{0, \dots, M - 1\}$ .

Then compute the matching measure that combines all of these features matching measures  $(Mu(i, j), Mv(i, j), Mh(i, j), Mt(i, j))$  as the following:

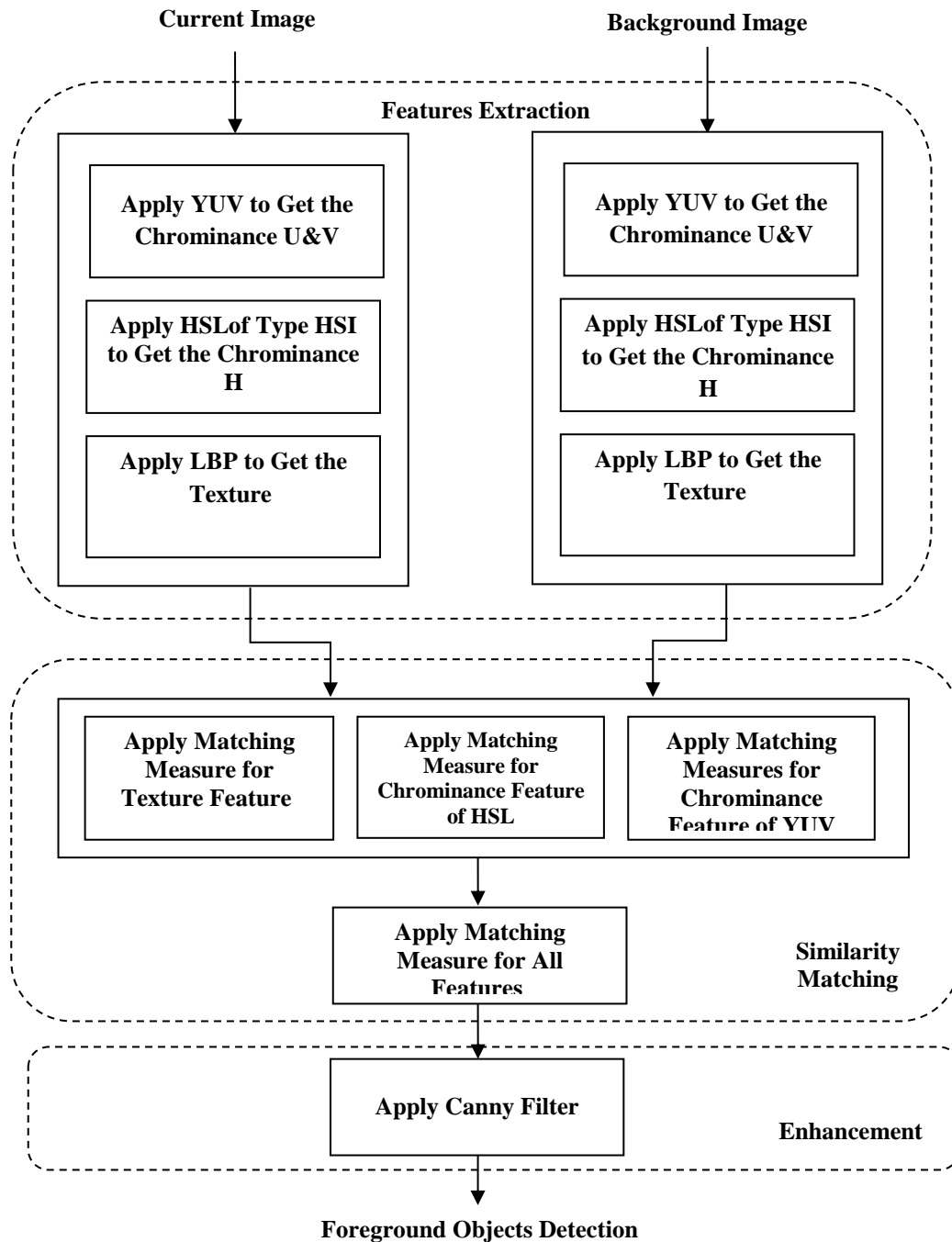
$$M(i, j) = Mu(i, j) \times 0.2 + Mv(i, j) \times 0.3 + Mh(i, j) \times 0.25 + Mt(i, j) \times 0.25 \dots (9)$$

Then if  $M(i, j) < Threshold$ . This pixel at position  $(i, j)$  detects as a foreground pixel.

c) Enhancement:

The third step in the proposed method is the enhancement. In the third step applying a canny filter to enhance the results of foreground object detection that produced from the second step of the proposed method. The canny edge detects the edges by retaining edge pixel that are strong (where the value of the pixel's gradient is above the high threshold) and the other edges that are weak and connected to strong edges (where the value of the pixel's gradient is below the high threshold and above the low threshold). The canny filter then removing all the other boundaries that are weak and not connected to strong edges. So that the low threshold value of canny edge filter is set to "0.05", where the high threshold value of canny edge filter is set to "0.2" and the sigma value of canny edge filter is set "1.5". Finally, the output of the canny filter provides the final foreground objects.

The apply proposed method with enhancement by canny gives better performance than without using it.



**Figure (1):** Framework of the Proposed Foreground Object Detection Method

**Algorithm (1)** describes the proposed detection approach of foreground object as follows:

**Algorithm (1): The Proposed Foreground Object Detection Method**

**Input: Current Image and Background Image**

**Output: Foreground Objects Detection**

**Process:**

**Step 1: Read Current Image and Background Image**

**Step 2: For  $i = 0$  to  $N - 1$  the width of the image //for both Current and Background Image**

**Step 3: For  $j = 0$  to  $M - 1$  the height of the image //for both Current and Background Image**

**Step 4: Apply YUV to Get the Chrominance Features U&V //Apply this Step in Current**

**And Background Image**

**Step 5: Apply HSL of type HSI to Get the Chrominance Feature H //Apply this Step in**

**Current and Background Image**

**Step 6: Apply LBP to Get the Texture Feature //Apply this Step in Current and Background Image**

**Step 7: Compute Matching Measures  $M_u(i, j)$  and  $M_v(i, j)$  for Chrominance Features of YUV**

**Step 8: Compute Matching Measure  $M_h(i, j)$  for Chrominance Feature of HSL**

**Step 9: Compute Matching Measure  $M_t(i, j)$  for Texture Feature**

**Step 10: Compute Matching Measure  $M(i, j)$  for All Features**

**Step 11: Then if  $M(i, j) < \text{Threshold}$ . This pixel at position  $(i, j)$  detects as a foreground pixel**

**Step 12: End for j**

**Step 13: End for i**

**Step 14: Apply Canny Filter for Enhancement the Results**

**Step 15: End**



## 7. EXPERIMENTAL RESULTS

This section illustrates the experimental results of the proposed foreground object detection method. A set of digital color images with different size and type have been used as test materials to evaluate the proposed approach performance. Figure (2) presents the set of used background images, figure (3) presents the set of used current images, figure (4) presents the result in binary (when apply threshold in matching measure  $M(i,j)$  if is true set 255 (white) otherwise set 0 (black)) for the detecting foreground objects by applying the proposed method and figure (5) presents the result in color (when apply threshold in matching measure  $M(i,j)$  if is true set its color value otherwise set 0 (black)) of foreground objects detection by applying the proposed method.



Figure (2): The Set of Background Images



Figure (3): The Set of Current Images

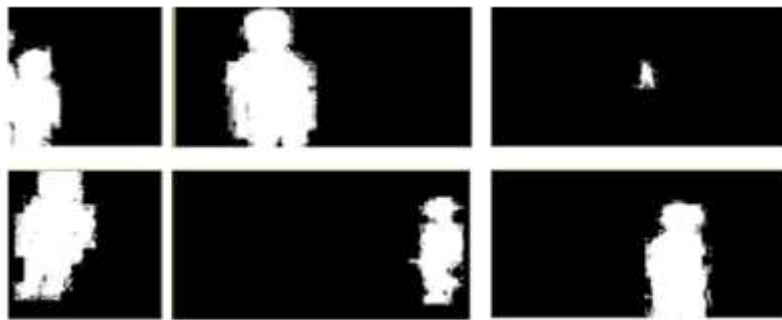


Figure (4): Binary Foreground Objects Detection Using Proposed Chrominance and Texture Features with Enhancement by Canny Filter



Figure (5): Color Foreground Objects Detection Using Proposed Chrominance and Texture Features with Enhancement by Canny Filter

According to figures (4, 5) illustrate that the results of detection object are robust against obstacles like shadows, illumination changes, and noises when applying the proposed object detection method.

To evaluate the proposed method was tested by using three performance measures which are precision, recall, and F-measure and then compared with other methods as presented in the table (1). The resulting value of these measures is a real number between 0 and 1 .whereas higher measure value represent more accurate foreground detection. The first measure is Precision is also known as (accuracy), which is defined as [18]:

$$Precision = \frac{TP}{TP+FP} \quad \dots\dots\dots (10)$$

The second measure is Recall also known as (sensitivity), which is defined as [18]:

$$Recall = \frac{TP}{TP+FN} \quad \dots\dots\dots (11)$$

Where *TP*, *FP* and *FN* represent the number of true positives, number of false positives and number of false negatives, respectively.

While F-measure is the measure that combined precision and recall together and is defined as [18]:

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad \dots\dots\dots (12)$$

Table (1): Comparison the performance of Gaussian Mixture Method, Background Subtraction Method and Proposed Method Based on Precision, Recall and F-measure

Method	Precision	Recall	F-measure
Gaussian Mixture Method(GMM)[19]	0.873	0.704	0.779
Background Subtraction Method(BSM)[20]	0.913	0.812	0.859
Proposed Method	0.924	0.922	0.922

According to the table (1) shows that proposed method gives better performance compared with other methods.

The time consumed to detect objects using the proposed approach has been tested. Table (2) presents the results of the time consumption in the proposed approach to detect objects for different testing images with different size and type.

Table (2): The test results of the time consumption in the proposed method for test images with different size and type

Tests Images	Size	Type	The Time Consumed in Second
Test Image 1	500*304 Pixels	.BMP	0.8468755 Sec.
Test Image 2	250*152 Pixels	.BMP	0.3723437 Sec.
Test Image 3	514*250 Pixels	.JPG	0.7625061 Sec.
Test Image 4	250*140 Pixels	.JPG	0.3510935 Sec.
Test Image 5	500*281 Pixels	.GIF	0.7839062 Sec.
Test Image 6	250*140 Pixels	.GIF	0.3506254 Sec.

According to the table (2) illustrates that the average consumption time for processing the proposed method is about 0.5778923 seconds.

## 8. CONCLUSION

This paper presented a new approach for foreground object detection based on chrominance and texture features then used a canny filter for enhancement. The chrominance and texture features are robust against the noise and shadows so this makes

our method robust against challenges and obstacles. Whereas the using of canny filter gives better results.

The experimental results verified the effectiveness of the proposed foreground object detection approach and the evaluation confirmed that the proposed method is more accurate than the other compared methods by using three evaluation measure which are precision, recall, and F-measure.

## REFERENCES

1. S. Adhikari, J. Kar, and J. G. Dastidar, "An Automatic And Efficient Foreground Object Extraction Scheme", International Journal of Science and Applied Information Technology
2. , Vol. 3, No.2, 2014.
3. P. Gujrathi, R. A. Priya, and P. Malathi , "Detecting Moving Object Using Background Subtraction Algorithm in FPGA", [Fourth International Conference on Advances in Computing and Communications](#), IEEE,2014.
4. M. Bhagyashri and A. Makwana, "Detection Background Subtraction Technique As Review", International Journal of Advance Engineering and Research Development (IJAERD), National Conference on Recent Research in Engineering and Technology (NCRRET), e-ISSN: 2348 – 4470, print-ISSN:2348-6406,2015.
5. J. Son, S. Kim, and K. Sohn, "Fast Illumination-Robust Foreground Detection Using Hierarchical Distribution Map For Real-Time Video Surveillance System", expert systems with applications, Elsevier, 2016.
6. A. M. Rahma and N.B. Abd, "Detection of a Moving Car Based on Invariant Moments", Journal of Computer Sciences, 2017.
7. D. Yang, C. Zhao, X. Zhang and S. Huang, "Background Modeling by Stability of Adaptive Features in Complex Scenes" [IEEE Transactions on Image Processing](#), Vol. 27, Issue 3, March 2018.

8. 7.Won–Y. S., M. Humayun K., M. Robiul H., and Sung–H. Y ., “Development of an Edge–Based Algorithm for Moving–Object Detection Using Background Modeling”, J. Inf. Commun. Converg. Eng. 12(3): 193–197, Sep. 2014.
9. L. Li, W. Huang and I.Y.H.Gu, ”Statistical modeling of complex backgrounds for foreground object detection”, [IEEE, Transactions on Image Processing](#), Vol.13, [No. 11](#),PP. 1459–1472, Nov. 2004.
- 10.A. Mohanty and S. Shantaiya, ”A Survey on Moving Object Detection using Background Subtraction Methods in Video”, International Journal of Computer Applications, National Conference on Knowledge, Innovation in Technology and Engineering, 2015.
- 11.R.C.Gonzalez and R.E.Woods,”Digital Image Processing”, Third Edition, Pearson Education International.
- 12.T. Horprasert, D. Harwood, and L. S. Davis, “A Statistical Approach for Real–Time Robust Background Subtraction and Shadow Detection”, IEEE, ICCV, Frame Rate Workshop, 1999.
- 13.Nishad PM and R.M. Chezian,” Various Colour Spaces And Colour Space Conversion Algorithms”, JGRCS, Vol. 4, No. 1, January 2013.
- 14.M. Podpora , G. P. Korba’s and A. K. Janik,” YUV vs RGB – Choosing a Color Space for Human–Machine Interaction”, ACSIS, Vol. 3, September 2014.
- 15.M. Pietikäinen , A. Hadid, G. Zhao and T. Ahonen, “Computer Vision Using Local Binary Patterns”, Computational Imaging and Vision 40, DOI 10.1007/978-0-85729-748-8\_2, Springer–Verlag London Limited, 2011.
- 16.M. Heikkila and M. Pietikinen, “A Texture–Based Method for Modeling the Background and Detecting Moving Objects”, IEEE, Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, [No. 4](#), PP. 657–662, April 2006.
16. J. Canny,” A computational approach to edge detection”, IEEE, Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–8, No. 6, PP. 679–698,Nov. 1986.

17. P. Rajendran and M.Madheswaran," Hybrid Medical Image Classification Using Association Rule Mining with Decision Tree Algorithm", Journal of Computing, VOL. 2, ISSUE 1, January 2010.
18. D. M. W. Powers, "Evaluation: From Precision, Recall and F Measure to ROC, Informedness, Markedness & Correlation", Journal of Machine Learning Technologies, Australia, December 2011.
19. C. Stauffer and W. E. L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking", *IEEE, [Transactions on Pattern Analysis and Machine Intelligence](#)*, Vol. 22, No.8, PP. 747–757, Aug. 2000.
20. W. Hossain and M. N. Das, "Moving Object Detection in Dynamic Backgrounds for Surveillance Systems", IEEE, [International Conference on Advanced Communication Control and Computing Technologies \(ICACCCT\), 2014](#).