Turkish Journal of Computer and Mathematics Education Vol.12 No.10 (2021), 2511-2517

Research Article

Augmented Reality Appling with Consistency of Behavior using Oriented Bounding Box Algorithm

Abdul-Wahab Sami Ibrahim[1], Duraid Saafan Mosa[2]

Computer science, College of Education, University of AL-Mustansiriyah, Baghdad, Ir[1]
Computer science, College of Education, University of AL-Mustansiriyah, Baghdad, Ir[2]
[1]dr.wahab.sa@uomustansiriyah.edu.iq,[2] duried96@gmail.com
Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Augmented reality can be defined as the process of merging digital information in real videos, it plays big role in many application such as education, medicine, and media industries. The merging of virtual moving object into real videos consists of two main stages: the detection of moving objects into real videos and the collision detection between real and virtual objects. In this study, an algorithm for merging virtual moving object into real video has been proposed, it is based on the three frame differencing technique followed by two pre-processing steps to detect real moving object. While the collision detection stage is based on the intersection between moving object path and the bounding box of real object. The experiment results show the good accuracy of the proposed algorithm in extracting moving object and merging virtual object into video frames.

Keywords: moving object detection, collision detection, frame differencing, and bounding box

1. Introduction

The Interest in Augmented AR has significantly increased in the past two decades, its aims to merge virtual objects that generate by computer into real videos. AR consists of two main stages: the detection of moving object and the collision detection. The detection of moving objects aims to tracking mobile objects in surveillance video with respect to background region. It has wide spectrum of computer vision application such as augmented reality AR [1], human tracking [2], road condition monitoring [3], and airport safety [4]. Various techniques have been recognized in literature to tackle the problem of object detection such as background subtraction, frame differencing, and optical flow [5, 6, 7, 8]. The basic idea behind the background subtraction techniques is to construct a model for the color intensity of the background and every pixel that not comply to this model can be considered a moving object pixels. These techniques are not suitable to scenes that have dynamic background [9, 10]. In the frame differencing techniques, a pixel can be categorized as moving object (foreground) when an observable change in its intensity between the previous frame and current frame[10, 11]. In the other hand, the optical flow technique uses vectors to represent image pixels and the vectors which have direction and magnitude can be classified as moving object pixels since background pixels are static and its magnitude approximately close to zero[12,13,14]. Table 1 illustrates the strength points and the weak points of the three techniques.

Table 1. The strength points and the weak points of the techniques: background subtraction, frame differencing, and optical flow

Technique	Strength points	Weak points
Frame Difference	Easiest way. Perform well for static background.	It requires a background without moving object
Background subtraction	1. Low memory requirement. 2. Not need for frame sub-sampling to create model for background.	It computation requires a buffer with the recent pixel values.
Optical Flow	It can extract all the moving object information.	A large amount of calculation is required.

Finally, Deori et al. [15] specified the main steps to detect moving object in surveillance video which are: video frames, preprocessing, proposed algorithm, post processing, and moving object detection and tracking as shown in Figure 1.

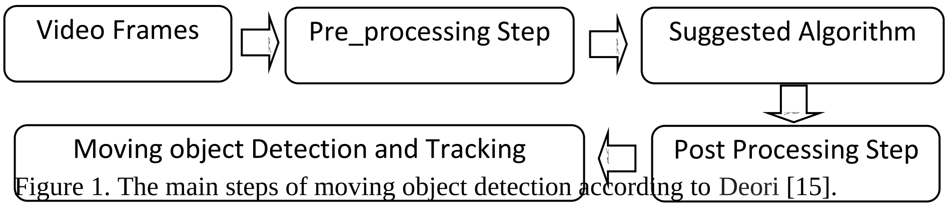


Figure 1. The main steps of moving object detection according to Deori [15].

On the other hand, most the collision detection method tends to represent virtual object with small circles, rectangles, and spheres. The computations are achieved to locate where and when virtual object strikes real object, one of the most complex techniques in collision detection are the physically-based since it takes into consideration the gravity and forces. Finally, this work aims to merge circular virtual object that follows a specific path into AVI video. The object detection of this study is based on three difference technique and the collision detection is based on the equations of the bounding box and the path.

2. Related Works

The detection of moving object is the first and the most crucial step in the analysis of video. The tracking algorithm needs an object detection technique applied in every frame or when an object appears on the video[16]. Various works has been conducted to detect moving object in surveillance video Kartika et al. [17] enhanced the accuracy of moving object detection for frame difference technique by using two preprocessing steps which are adaptive threshold method and shadow detection in HSV color space. Srivastav et al. [18] proposed a hybrid algorithm based on the frame differencing and background subtraction techniques. Three frames have been used in the frame differencing to tackle the problem of holes gains from two frames differencing while background subtraction has been used to tackle the problem of dynamic background. However, this algorithm is time consuming. Sengar et el.[19] Applied histogram-based frame differencing technique and W4 algorithm separately on video frame sequence. Then combined the outcomes of the two techniques using logical OR and operations to detect the moving objects. Zhang et al. [20] extract moving object by using three frame subtraction, and then manipulate the extracting image by mathematical morphology method to eliminate noise in image. Sengar et al. [21] enhanced three-frame difference technique and combining it with background subtraction to improve the extraction of multiple moving objects from outdoor and indoor in real video dataset. On the other hand, collision detection must be simply estimated base on approximate models such as circle, rectangle, and sphere data. various studies have been suggested to estimate the collision detection, Lee at el. in [22] proposed an algorithm to estimate the collision between virtual object (ball) and an arbitrarily-shaped objects (human hand). The arbitrary shape has been divided to set of spheres in order to simplify the collision detection. The suggested method is suitable for augmented reality application. Another method for collision detection was presented by Daeho et al. [23] based on the ratio of the overlapping area between real and virtual objects. This method estimated the collision detection by analyzing the relationship between the motion vector of virtual object and a normal vector of the collided plane, the method can be used efficiently in augmented reality system. Chang at el in [24] suggested an algorithm for collision detection between rigid objects by using oriented bounding box enhanced with bounding sphere. The results show that the algorithm considerable enhanced the computational cost as compared with OBB algorithms.

3. Proposed Algorithm

In this study, a method for merging virtual object V_{obj} into AVI video has been proposed, the method consists of two stages: the moving object detection and the collision detection as described in the next sections.

3.1 Moving Object detection

The moving object detection in this work is based the on three frame differencing technique followed by two pre-processing step as listed in steps below:

1. in the first step, the backward difference BD and the forward difference FD are calculated for every three consecutive frames F_{i-1}, F_i, and F_{i+1} where BD = (Frm_i - Frm_{i-1}) and FD = (Frm_{i+1} – Frm_i) then a logical AND is executed between BD and FD.

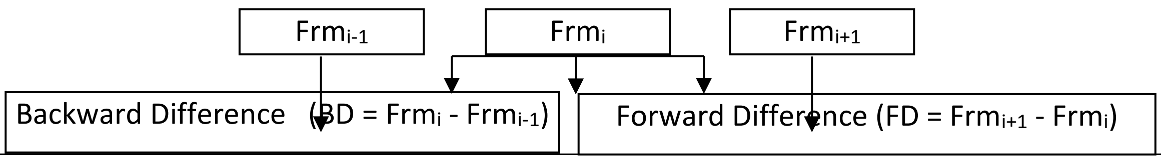
$$I_{RGB}=AND(BD, FD)$$

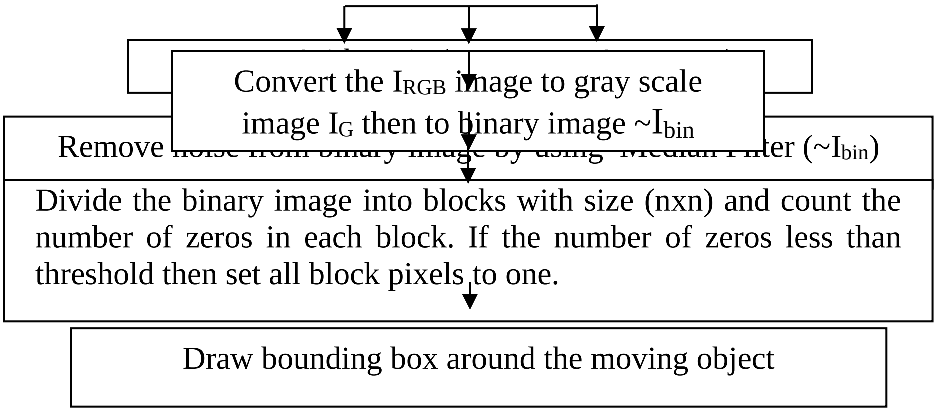
2. Convert the I_{RGB} image to gray scale image I_G then to binary image I_{bin} then find the negative (~) of the binary image.

$$I_{bin}= \sim binary(Gray(I_{RGB}))$$

3. Remove noisy pixels from I_{bin} by using Median filter with large size (9x9).
4. Remove image parts which belong to small moving objects such as tree branches by partition the image into small block of size (nxn) and calculate the number of zero pixel in each block, if the number of zeros less than threshold then set all the block to ones otherwise keep the same block.
5. Scan the binary image to find coordinate of the leftmost pixel (x_{min}, y), the rightmost pixel (x_{max}, y), the higher pixel (x, y_{max}), and the lower pixel (x, y_{min}) then draw the moving object bounding box ((x_{min}, y_{min})- (x_{max}, y_{max})) .

Figure (2) illustrates The block diagram of the proposed algorithm to detect the moving object in surveillance video.





3.2 Collision detection

In this stage, a circular virtual object V_{obj} that follows a specific path is merged into AVI video then the collision between the V_{obj} path and the real object R_{obj} is estimated in each frame as shown in Fig3.a, the path of V_{obj} can be defined in Eq. 1.

$$Y=A+B*|Sin(x)| \tag{1}$$

Where A represents the height of the path from the base of the frame and B is the amplitude of the path as illustrated in Fig3.b. The scale of x is between (0-2PI).

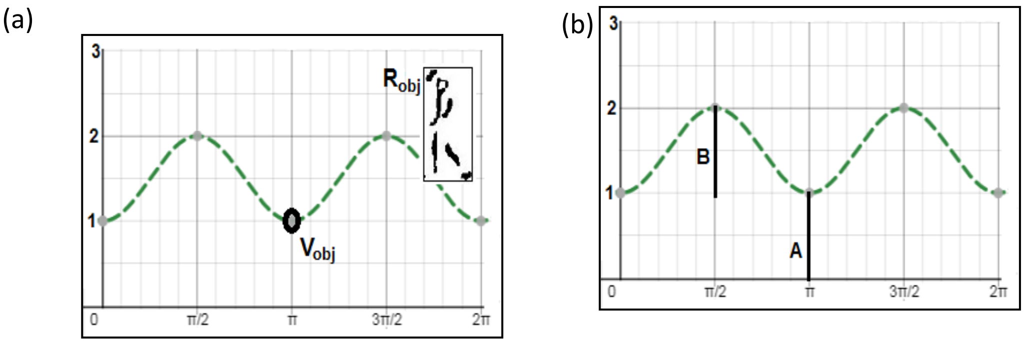


Figure (3) collision detection stage **a.** collision between bounding box and path. **b.** V_{obj} path.

Collision detection algorithm:

Input: Radius (R) and path equation (Y=A+B*|Sin(x)|) of circular V_{obj}

The sides coordinate of the bounding Box of R_{obj} [x_{min}, x_{max}, y_{min}, y_{max}];

Output: collision=(true/ false)

Step1: for i=1 to Number of frames in AVI video

Step2: Collision=false; G_{int}=[empty];

Step3: Locate the center of V_{obj} (x_c, y_c) in ***F(i)*** frame according to its path equation x_c= i (frame number) and yc= A+B*|Sin(x_c)| if the center of V_{obj} (x_c, y_c) inside bounding box of real object then collision= true, and exit.

Step4: Determine the circle equations of the V_{obj}:

1. $y_{int1} = y_c - \sqrt{R^2 - (x - x_c)^2}$
2. $y_{int2} = y_c + \sqrt{R^2 - (x - x_c)^2}$
3. $x_{int1} = x_c - \sqrt{R^2 - (y - y_c)^2}$
4. $x_{int2} = x_c + \sqrt{R^2 - (y - y_c)^2}$

Step5: Find the coordinates of the intersection points $I_1=(x_{min}, y_{int1})$ and $I_2=(x_{min}, y_{int2})$ between the circular V_{obj} and the left sides of the bounding box for the R_{obj} by replacing the value of x in equation 1 and 2 (step 4) by x_{min} :

If (y_{int1} is not complex number) and ($y_{min}< y_{int1} < y_{max}$) then insert I_1 into G_{int}

If (y_{int2} is not complex number) and ($y_{min}< y_{int2} < y_{max}$) then insert I_2 into G_{int}

Step6: Find the coordinates of the intersection points $I_3=(x_{max}, y_{int1})$ and $I_4=(x_{max}, y_{int2})$ between the circular V_{obj} and the right sides of the bounding box for the R_{obj} by replacing the value of x in equation 1 and 2 (step 4) by x_{max} :

If (y_{int1} is not complex number) and ($y_{min}< y_{int1} < y_{max}$) then insert I_3 into G_{int}

If (y_{int2} is not complex number) and ($y_{min}< y_{int2} < y_{max}$) then insert I_4 into G_{int}

Step7: Find the coordinates of the intersection points $I_5=(x_{int1},y_{min})$ and $I_6=(x_{int2}, y_{min})$ between the circular V_{obj} and the lower side of the bounding box for the R_{obj} by replacing the value of y in equation 3 and 4 (step 4) by y_{min} :

If (x_{int1} is not complex number) and ($x_{min}< x_{int1} < x_{max}$) then insert I_5 into G_{int}

If (x_{int2} is not complex number) and ($x_{min}< x_{int2} < x_{max}$) then insert I_6 into G_{int}

Step8: Find the coordinates of the intersection points $I_7=(x_{int1},y_{max})$ and $I_8=(x_{int1}, y_{max})$ between the circular V_{obj} and the lower side of the bounding box for the R_{obj} by replacing the value of y in equation 3 and 4 (step 4) by y_{max} :

If (x_{int1} is not complex number) ($x_{min}< x_{int1} < x_{max}$) then insert I_7 into G_{int}

If (x_{int2} is not complex number) ($x_{min}< x_{int2} < x_{max}$) then insert I_8 into G_{int}

Step9: if (G_{int} is empty) then get the next frame (Goto step 1) else collision = true; exit;

The proposed algorithm has been proposed to detect the collision between the path of virtual object and the bounding box of real object. Step 3 locates the center of V_{obj} based on its equation, if the center of V_{obj} inside the bounding box then collision detection signal is triggered and the algorithm is terminated. Otherwise, the algorithm calculate the intersection between the V_{obj} and each side (left, right, top, and bottom) of the bounding box (steps 5,6,7, and 8) as shown in Figure (4). If intersection points is found then there is a collision otherwise test the next frame (step 9).

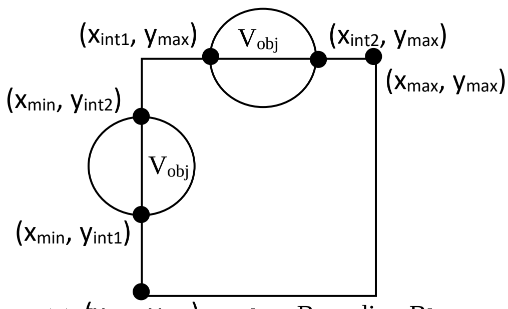


Figure (4) the intersection between V_{obj} and R_{obj}

3. Results

Two videos have been used to evaluate the detection accuracy of the proposed system. The first one consists of one moving object (football player) with static background and the camera is perpendicular to the scene. The second video (cable car) also consists of one moving object with static background. The main properties of the two videos are shown in Table 1.

Table 2. The main properties of the used videos

Property	Football player video	Cable Car video
Duration (sec)	10.171	3.333
Bits Per Pixel	24	24
Frame Rate (FPR)	30.477	15
Height (pixels)	720	240
Video Format	RGB24	RGB24
Width (pixels)	1280	320

The football player video was used in the first experiment as shown in Figure (5.a). The forward difference FD and the backward difference BD for any three consecutive frames such as 33, 34, and 35 are found then a logical And between them is computed I_{RGB} =(FD AND BD). The I_{RGB} image was converted to gray scale image I_G then to binary image $\sim I_{bin}$ as shown in figure (5.b). a median filter of size (9x9) has been supported to remove noise from $\sim I_{bin}$ as illustrated in the Figure 5.c. After that, the $\sim I_{bin}$ has been divided to small blocks of size (8x8) then count the number of zeros pixel, if the count of zeros is less than threshold value (16) then set all cells value in the block to one as shown in Figure 5.d. Finally, scan the binary image to find the leftmost, rightmost, most higher, and most lower pixels to draw the bounding box as shown in Figure (5.e).

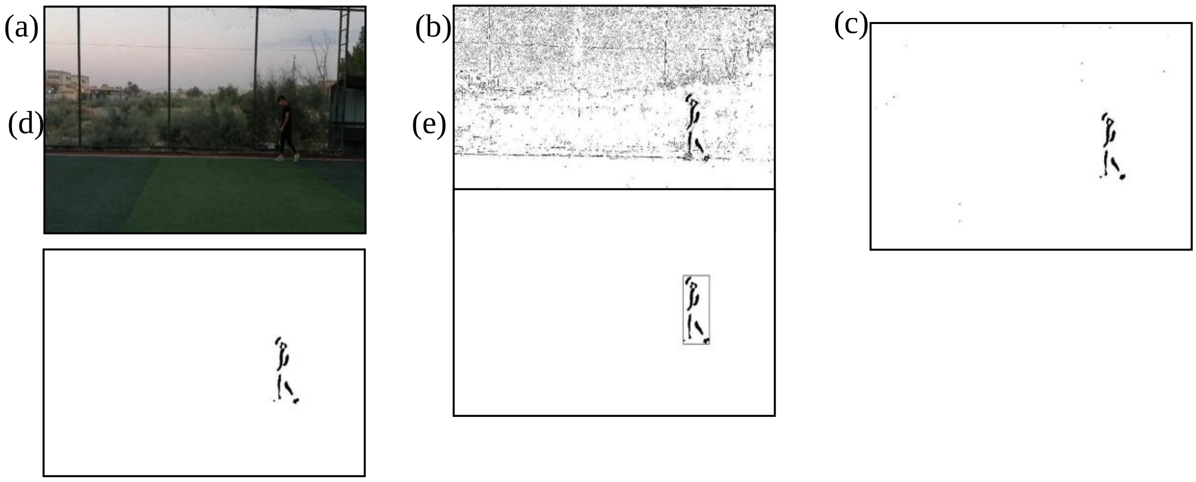


Figure 5. the result of each step for football player video: **a**. original frame. **b**. binary image after three frame difference and convert to binary image. **c**. image after noise removal. **d**. image after small moving object removal **e**. bounding box around the moving object.

The car cable video was used in the second experiment as shown in Figure (6.a). The forward difference FD and the backward difference BD for any three consecutive frames such as18, 19, and 20 are found then a logical And between them is computed I_{RGB} =(FD AND BD). The I_{RGB} image was converted to gray scale image I_G then to binary image $\sim I_{bin}$ as shown in figure (6.b). A median filter with size (9x9) has been supported to remove noise from $\sim I_{bin}$ as illustrated in the Figure 6.c. After that, the $\sim I_{bin}$ has been divided to small blocks of size (8x8) then count the number of zeros pixel, if the count of zeros is less than threshold value (32) then set all cells value in the block to one as shown in Figure 6.d. Finally, scan the binary image to find the leftmost, rightmost, most higher, and most lower pixels to draw the bounding box as shown in Figure (6.e). Various works has been conducted to detect moving object in surveillance video the experiments show that the proposed algorithm is more accurate to detect moving object in football player video since the camera location is perpendicular to scene.

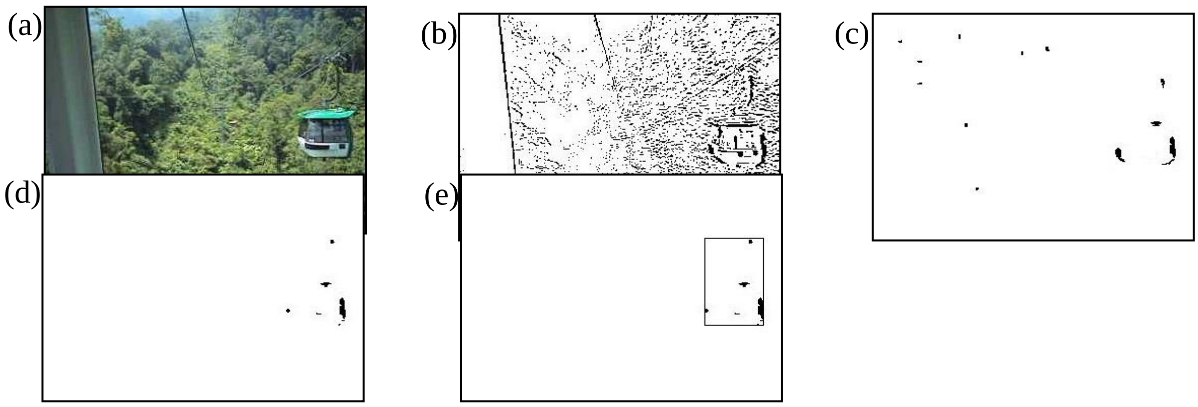


Figure 6. the result of each step for cable car video: **a**. original frame. **b**. binary image after three frame difference and convert to binary image. **c**. image after noise removal. **d**. image after small moving object removal **e**. bounding box around the moving object.

On the other hand, two experiments have been done to estimate the collision detection between V_{obj} and R_{obj}. In the first experiment, a circular virtual object has been merged into video that follow the equation Y=250+10*|Sin(x)| where 250 is the distance in pixels between the base of the frame and the V_{obj} while 10 is the amplitude of the V_{obj}. The moving direction and the speed of the V_{obj} and $R_{obj}$was the same, the algorithm did not detect any collision between V_{obj} and R_{obj} for all the frames into the video. In the second experiments, the parameter has been used except that the moving direction of V_{obj} was in reverse with direction of the moving real object. The algorithm detected collision between V_{obj} bath and the bounding box of R_{obj} in frame 45.

4. Conclusions

In this work, an algorithm for merging virtual moving object into AVI video is proposed, the algorithm consist of two main stages: the moving object detection which is based on three differencing technique and two preprocessing steps and the collision detection stage which is based on the intersection between the bounding box of the real object and the path of the virtual object. The path of virtual object can be refused if the collision algorithm detects any intersection between the virtual path and real object in any frame into the AVI video.

References

1. Apicharttrisorn, Kittipat, Xukan Ran, Jiasi Chen, Srikanth V. Krishnamurthy, and Amit K. Roy-Chowdhury. "Frugal following: Power thrifty object detection and tracking for mobile augmented reality." In Proceedings of the 17th Conference on Embedded Networked Sensor Systems, pp. 96-109. 2019.
2. Visakha, K., and Sidharth S. Prakash. "Detection and Tracking of Human Beings in a Video Using Haar Classifier." In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1-4. IEEE, 2018.
3. Dkhil, Mejdi Ben, Ali Wali, and Adel M. Alimi. "Towards a Real Time Road Moving Object Detection and Tracking System." Journal of Information Assurance and Security 11 (2016): 39-47.
4. Thupakula, Kiran, Adishesha Sivaramasastry, and Srikanth Gampa. "A methodology for collision prediction and alert generation in airport environment." SAE International Journal of Aerospace 9, no. 2016-01-1976 (2016): 1-7.
5. Thapa, Gopal, Kalpana Sharma, and M. K. Ghose. "Moving object detection and segmentation using frame differencing and summing technique." International Journal of Computer Applications 102, no. 7 (2014): 20-25.
6. Balaji, S. R., and S. Karthikeyan. "A survey on moving object tracking using image processing." In *2017 11th* international *conference on intelligent systems and control (ISCO)*, pp. 469-474. IEEE, 2017.
7. Savitha, C., and D. Ramesh. "Motion detection in video surviellance: A systematic survey." In 2018 2nd International Conference on Inventive Systems and Control (ICISC), pp. 51-54. IEEE, 2018.
8. Sengar, Sandeep Singh. "Motion segmentation based on structure-texture decomposition and improved three frame differencing." In IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 609-622. Springer, Cham, 2019.
9. Shaikh, Soharab Hossain, Khalid Saeed, and Nabendu Chaki. "Moving object detection using background subtraction." In *Moving object* detection *using background subtraction*, pp. 15-23. Springer, Cham, 2014.
10. Paul, Nihal, Ashish Singh, Abhishek Midya, Partha Pratim Roy, and Debi Prosad Dogra. "Moving object detection using modified temporal differencing and local fuzzy thresholding." *The Journal of Supercomputing* 73, no. 3 (2017): 1120-1139.
11. Qu, Huiyan, Wenhui Li, and Wei Zhao. "Human-Vehicle Collision Detection Algorithm Based on Image Processing." *International Journal of Pattern Recognition and Artificial Intelligence* 34, no. 08 (2020): 2055015.
12. Kurnianggoro, Laksono, Ajmal Shahbaz, and Kang-Hyun Jo. "Dense optical flow in stabilized scenes for moving object detection from a moving camera." In 2016 16th International Conference on Control, Automation and Systems (ICCAS), pp. 704-708. IEEE, 2016.
13. Agarwal, Anshuman, Shivam Gupta, and Dushyant Kumar Singh. "Review of optical flow technique for moving object detection." In *2016 2nd International Conference on Contemporary*
14. Agarwal, Anshuman, Shivam Gupta, and Dushyant Kumar Singh. "Review of optical flow technique for moving object detection." In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 409-413. IEEE, 2016.

15. Deori, Barga, and Dalton Meitei Thounaojam. "A survey on moving object tracking in video." International Journal on Information Theory (IJIT) 3, no. 3 (2014): 31-46.
16. Algethami, Nahlah, and Sam Redfern. "Combining Accumulated Frame Differencing and Corner Detection for Motion Detection." In CGVC, pp. 7-14. 2018.
17. Kartika, Intan, and Shahrizat Shaik Mohamed. "Frame differencing with post-processing techniques for moving object detection in outdoor environment." In 2011 IEEE 7th International Colloquium on Signal Processing and its Applications, pp. 172-176. IEEE, 2011.
18. Srivastav, Neha, Shubh L. Agrwal, Sandeep K. Gupta, Saurabh R. Srivastava, Blessy Chacko, and Hemant Sharma. "Hybrid object detection using improved three frame differencing and background subtraction." In 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence, pp. 613-617. IEEE, 2017.
19. Sengar, Sandeep Singh, and Susanta Mukhopadhyay. "Moving object detection based on frame difference and W4." *Signal, Image and Video Processing* 11, no. 7 (2017): 1357-1364.
20. Zhang, Yanzhu, Xiaoyan Wang, and Biao Qu. "Three-frame difference algorithm research based on mathematical morphology." Procedia Engineering 29 (2012): 2705-2709.
21. Sengar, Sandeep Singh, and Susanta Mukhopadhyay. "Foreground detection via background subtraction and improved three-frame differencing." *Arabian Journal for Science and Engineering* 42, no. 8 (2017): 3621-3633.
22. Lee, D., S. G. Lee, W. M. Kim, and Y. J. Lee. "Sphere-to-sphere collision estimation of virtual objects to arbitrarily-shaped real objects for augmented reality." Electronics letters 46, no. 13 (2010): 915-916.
23. Lee, Daeho, and Youngjae Lee. "Estimation of collision response of virtual objects to arbitrary-shaped real objects." IEICE Electronics Express 5, no. 17 (2008): 678-682.
24. Chang, Jung-Woo, Wenping Wang, and Myung-Soo Kim. "Efficient collision detection using a dual OBB-sphere bounding volume hierarchy." Computer-Aided Design 42, no. 1 (2010): 50-57.