



A REVIEW OF EDGE IMAGE DETECTION FOR MARKER-BASED AUGMENTED REALITY

Samuel P. A. H. Pandiangan, Tito Waluyo Purboyo and Randy Erfa Saputra

Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

E-Mail: samuelpandiangan5@gmail.com

ABSTRACT

The earlier way to do an image detecting is to use edge image detection. Edge image detection is using some kind of mathematical expression to detect the boundaries between objects between images in real world, with media like camera phone. Many aspects have used edge image detection, one of them is Augmented Reality, especially marker-based Augmented Reality. Augmented reality used edge image detection for reading and detecting the marker in real world and transfer the data to the database in application. This paper will review how the edge detection works and how it works in Augmented Reality.

Keywords: augmented reality, edge detection, edge image detection, mobile, marker-based.

INTRODUCTION

Augmented Reality has become one of the most famous technology right now [11]. After Virtual Reality, AR become the most advanced visual technology that has been anticipated to have much impact in our daily life.

Augmented reality allows us to augment a virtual computer-generated element into real world, whether its direct or using some kind of media called "Marker". Because of this reason, Augmented Reality divided into Augmented Reality that use marker, we call it marker-based and AR that didn't use marker, or we call it marker-less [23] [1]. In Marker-less Augmented Reality, we don't need some kind of marker to help us detect the object's position, but it uses the global positioning [23], while in Marker-Based we need some kind of marker to help the program detect the object and pinpoint the exact position that the computer-generated elements will appear [23].

Both of the Augmented Reality need some kind of method that allow detection of some image in real world. But Marker-Based is the one that needed some kind of detection the most. There are many methods of detecting this marker, including ArToolkit [2], ARTag [20], Line Detection [22], and edge detection [22]. In this paper, we only explain the edge detection and how it works.

Theoretical concepts of edge detection

In Theory Edge detection is divided into 2 main categories by using the derivative of the images as the main foundation [14]. There are Gradients and Laplacian [12] [7]. In the gradient method, the edge will be detected by finding the highest and lowest value of the first derivative of the vector in images [7]. While in Laplacian will first search the second derivative of the images, then find the zero crossing of it [7] [12].

Gradient edge detection

In Gradient Edge Detection, the first order or the first derivative of the image plays a big role of finding the edges [5] [7] [19]. Also in Gradient Edge Detection, it will automatically assume that there are a high gradients at the edge of the each pixel in an image [5]. If we have an

image with a function $f(z, y)$ the pixel gradient can be defined with [21] [27] :

$$|G| = |Gz| + |Gy| \quad (1)$$

G = the overall gradient of pixel
 Gz = gradient of z (height)
 Gy = gradient of y (width)

While The point of vector is :

$$\nabla f = \text{mag}(\nabla f) = [(Gz)^2 + (Gy)^2]^{1/2} \quad (2)$$

Gz = gradient of z (height)
 Gy = gradient of y (width)
 ∇f = vector of gradient
 ∇ = vector
 Mag ∇f = Magnitude of gradient vector

The direction of vector will be found by using:

$$\phi(z, y) = \tan^{-1}(Gy / Gz) \quad (3)$$

Gz = gradient of z (height)
 Gy = gradient of y (width)
 ϕ = direction of vector z,y

Laplacian edge detection

While in Gradient the one that plays the big role is the first derivative, in Laplacian, the one that decided the edge is the second derivative of the image [7] [19] [12]. First of all, we need to find the second derivative, and then we must find the zero crossings of it, then we can pinpoint where the edge exactly [19] [21].

Figure-1 show us the gradient of the first derivative that has been detected as "edge" in the image [7]. This gradient will be used with the conjunction of the second Derivative result to find its zero crossing [7].

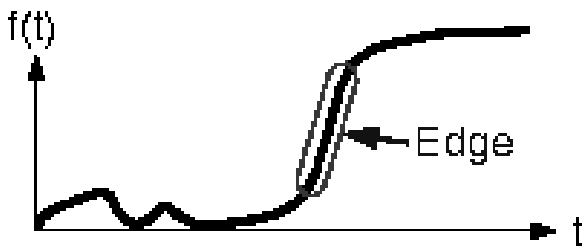


Figure-1. First Derrivative [7].

Figure-2 show us the gradient which has a large peek centered on the edge [7]. Whenever you compare the gradient into the treshhold, whenever the threshold is overexceed, tha is the signt that the edge has been found but has been concecrated below that peek [7].

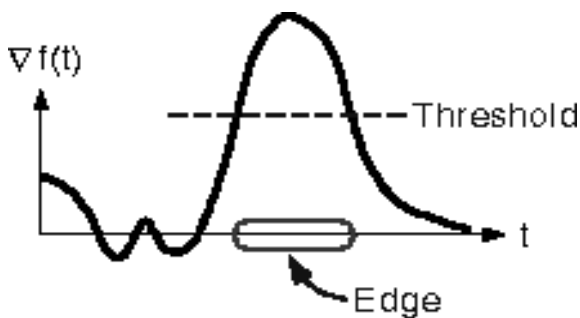


Figure-2. Second Derrivative [7]

After you've found the first and second derrivative in Figures 1 and 2, In Figure-3, We pictures the laplacian operation and the zero crossing between the first derrivative and the second derrivative of the images [7].

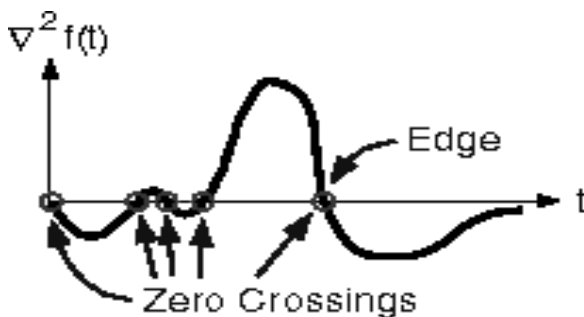


Figure-3. Zero Crossing [7].

Edge detection operators

We already saw the main categorize of edge detection by using the first and the second derrivative [3]. Normally, the easiest way to do the Edge Detection with the Gradient and Laplacient is to blur the image first, find the zero crossing and the edge like in Figure-4 and Figure-5.

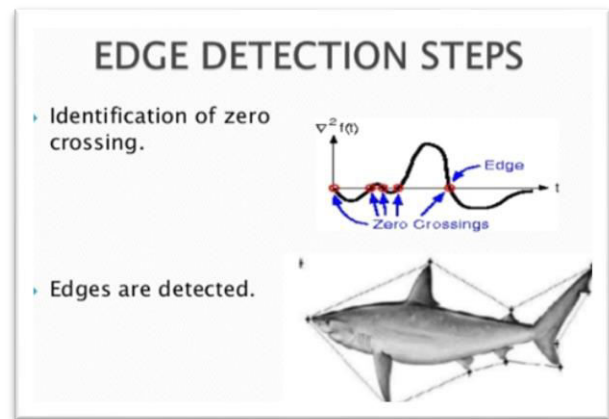


Figure-4. How to do Edge Detecting a Picture [7].

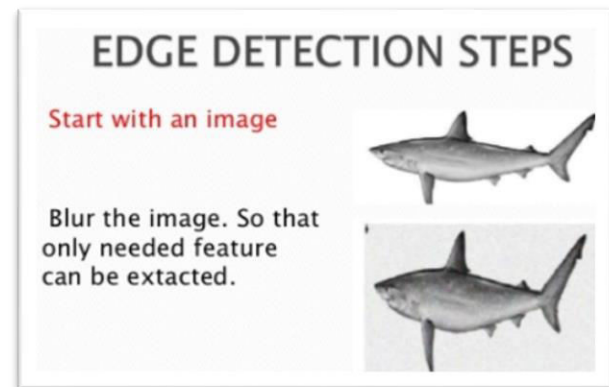


Figure-5. How to do Edge Detecting a Picture [7].

The first 2 ways in detecting an edge in a picture is to pick an image then, blur the image like hown in Figure-4 [7]. Blurring the image is the first thing to do to any edge detecting method [21] [7].

The next step is to apply a Laplacian First Derrivative and second derrivative into the image [7]. Then we must determine the zero crossing of the two derrivative and that's how the edge will detect the shark picture [7].

But in many cases, edge detection have another aspect that affect its performance, The image brightness [11], The blurring effect [27], noise of the image [28], image features [12] and many others. While its hard to remove all the aspect, the new operators of edge detecting has been found to counter this problem [12]. The three most famous edge detection operators are : Robert, Sobel and Prewitt [27] [9] [12] [18]. We will only discuss the ones that can be used In Augmented Reality.

Roberts operator

Roberts operator is the most simple structure and the fastest of all three that mentioned above [9] [27]. Robert can quickly measure the 2D point in input image, just by calculating the dissimilarities between pixel of the image edges [9]. Robert using a 2x2 with a number inside that use as a kernel to represent the Z and Y (height and width) of a pixel in image [7] [27]. You can see the example of Roberts Operator in Figure-6.



G_z	
1	0
0	-1

G_y	
0	-1
1	0

Figure-6. G_z and G_y Robert Operator [7].

Sobel operators

While Roberts operator use a 2x2 matrix to mask the position of the objects, Sobel use a 3x3 matrix to avoid having a gradient calculated by interpolated point between pixels [7] [4]. You can see the example of Sobel Operators in Figure-7.

S_z		
-1	0	1
-2	0	2
1	0	1

S_y		
1	2	1
0	0	0
-1	-2	-1

Figure-7. S_z and S_y Sobel Operator [7].

Prewitt operators

Prewitt is almost the same as Sobel Operators. It uses a 3x3 Matrix and the number inside as a kernel of Z and Y (height and width) gradient of the pixel in images [4] [21]. The difference is while Sobel use a "2" as a kernel, Prewitt only uses "1" and "0" as a kernel. And its only suitable for well-contrasted images [19]. You can see the example of Prewitt Operators in Figure-8.

P_z		
-1	0	1
-1	0	1
-1	0	1

P_y		
1	1	1
0	0	0
-1	-1	-1

Figure-8. P_z and P_y Prewitt Operator [7].

Other edge detection method

Beside some of the method that we already talk above there're still many other edge detecting method that we cant discuss it all in here. Like Canny operator [8].

Edge image detection in augmented reality

We already discuss some of the edge image detection method that can be used in Augmented Reality.

The full advantages of Edge Image Detection can be feel when we used a Marker-Based Augmented Reality. Because the use of Marker in that method really need some kind of detecting method to let the application knows where is the position of the marker [17] [16].

The form of Marker in Marker-Based Augmented Reality can be different from one another. But the basic of it is using ATK Makers to looks like a barcode with only black and white and using a square form [2] just like in Figure 9.



Figure-9. Example of Simple Marker [2].

The marker of AR must be made special so that when the program scan the AR inside the program can be called out by the program [15]. There are many ways to detect a simple marker, and the edge detection had already been proven can detect this kind of simple marker [17] [24] [22]. While the first study by Kim Jaeyoung shows us that edge detecting can be used [17], in another study by Martin Hizer [22] and Kumar [24] shows that there are still flaws that left by using edge detecting method in detecting simple marker, like when the border is thin, the marker is harder to detected [24].

So, how exactly an Edge detection can detect a picture as a marker for augmented reality?

Well, the first step like said above, the camera will blur the images to help for finding the edge, but if its using an operator the images than divided into some kind of small block that will be categorized into appropriate classes, and relationship [16] [17] just like in Figure-10. After that we will fill this block with numbers that ranged from 0 to 4 like Figure-11.

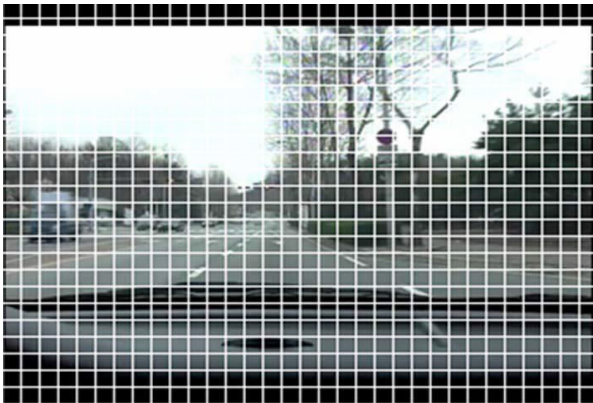


Figure-10. Dividing the picture into small blocks [16].

Dividing the block like we used in Figure-10 can be done in blurred images or not. The step of blurring the image first or not depends on the user.

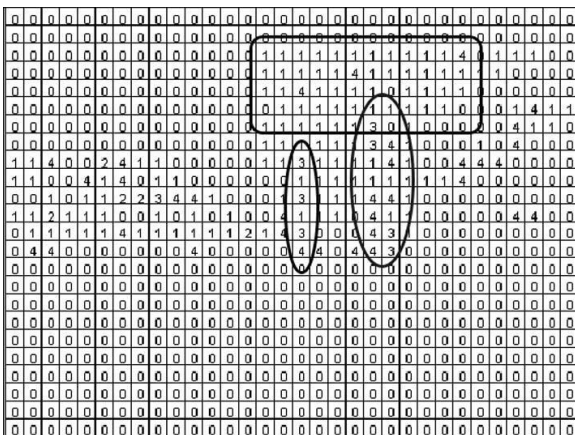


Figure-11. Dividing Categorizing the Blocks [16].

Number 1 inside each block represent the branch of the background trees, Number 3 and 4 represent most of the stems and lastly area between marker and the background represent by number 0 [16]. After the image of marker has been divided and categorized like in Figure-11, Every Edge in the picture can be founded through using a noise removal process [16]. You can see the result of noise removal process in Figure-12.

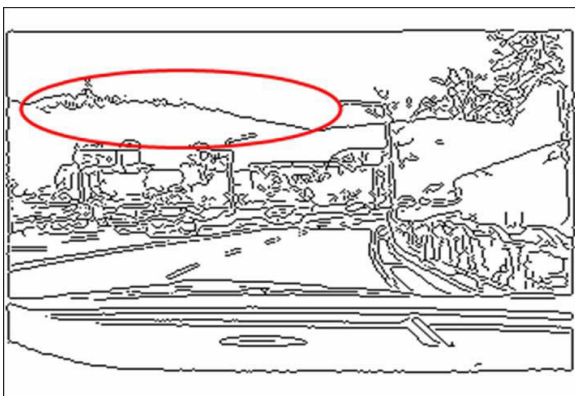


Figure-12. Background Edges [16].

Area determination

If we want to find a specific “edge” in the input pictures, we need to remove the area that we didn’t need it. For example if we want to find the marker, we only need the outside edge of the marker and remove the background. Like in Figure-13, The edge of marker is just as simple dot that really hard to figure if the background still exist [2].

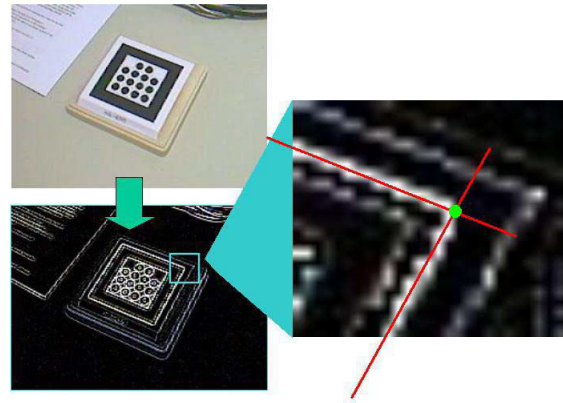


Figure-13. Finding Edge of Marker [2].

Area determination can be approach by edge segments in horizontal and vertical of area that we need [16]. Then the data will be evaluated by an equation [16]:

$$\begin{aligned} &\text{If } P_r(i) > A\mu, \text{ then } S(i) = 1, \\ &\text{Otherwise, } S(i) = 0 \end{aligned} \tag{4}$$

- $P_r(i)$ = it’s a presentasion value of i in search area r [16]
- i = represent a value to determine the search area
- r = distance of area
- $r \mu$ = the median or the middle value of r_p [16]
- A = coefficient for threshold value [16]
- S = search area

The final search region is chose from the value of i that makes $S(i) = 1$, and we must search it throughly from minimum to maximum value [16].

After the area has been decided, and the edge can be found, the application will now have an input of marker. And if the angle and the position of light is perfect, than the Augmented Reality inside it can be shown from database [15].

Edge detecting method can also be used to detect a fingerprints. Just like shown in Figure-14.

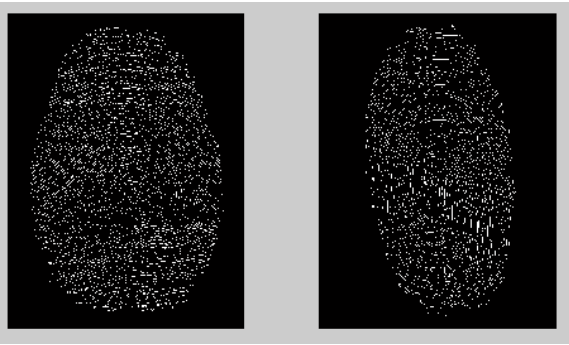


Figure-14. The Input of detecting fingerprints with edge detection [29].

First we take the input of the fingerprints, and show it to the camera where the edge detection can be used. And you can see the results in Figure-15. Even if not perfect but as you can see it can match the base shape of the objects that you want to detect [29].



Figure-15. The output of detecting fingerprints using edge detection [29].

Tracking

The Edge detection can be used in tracking the marker too but its not effective. For an AR detects a marker it will need at least the four corner of the marker

detected [24] [25]. By using this concept, we can input some algorithm into the program and make the area determination only determines the four corner of the marker [24].

For Example in detecting a red blood cell which is keep moving [5]. The detecting method will need some kind of tracking method to do that. The consistency will keep dropping if you want to detect a moving objects because it keep updating the position of the objects. [5]

Other things and mobile apps

Other things that we can do to help detect some step edges and edge that cannot be detected in normal ways is to give a different scale combination [10]. Or by using noise robustness and parameter sensitivity [10]. Or we can do various image enchanting with any method [31] [30].

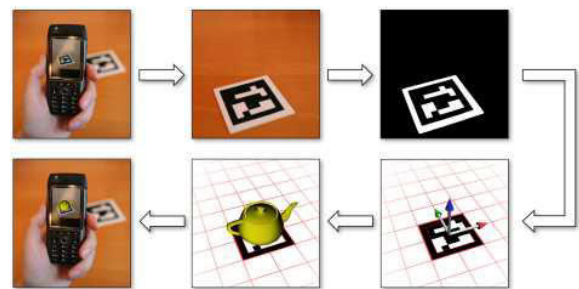


Figure-16. Augmented Reality in Mobile [17].

Nowadays, the AR technology can easily be applied in mobile devices using camera handphone as the media to detect the marker [13]. There are many apps that can help us make a simple AR such as Vuforia [6]. And the detection in mobile is not much different from other media, and so, edge detection method still can be used even in mobile.

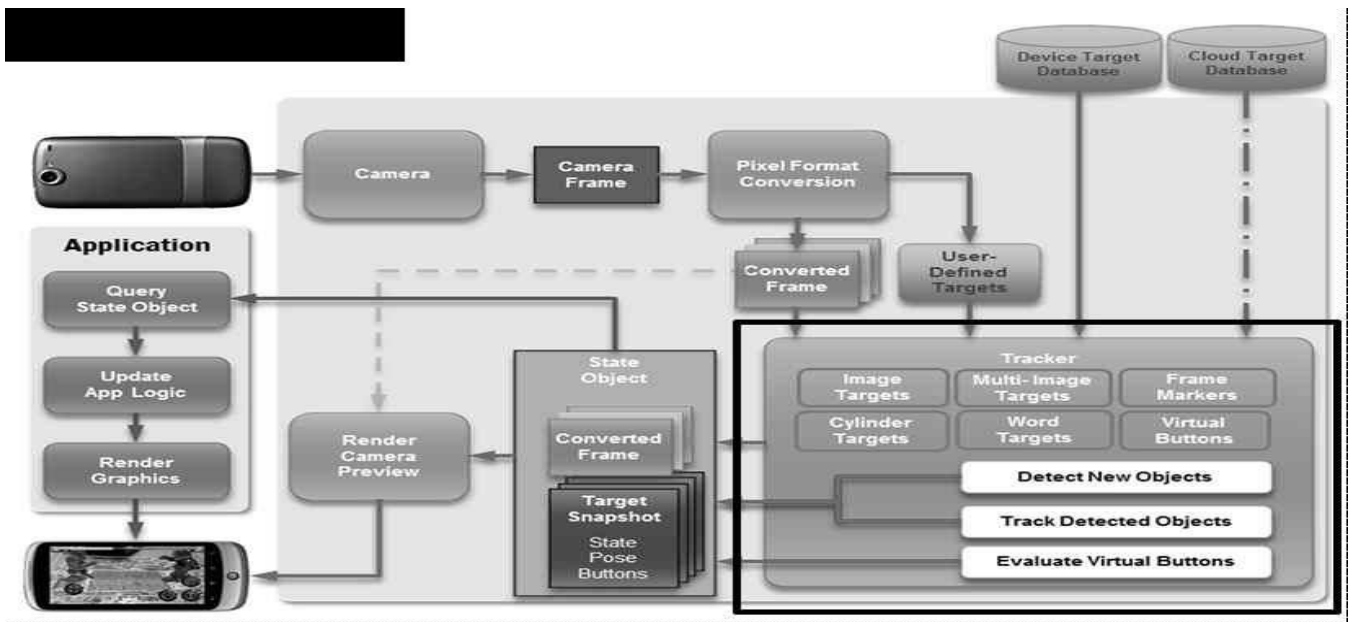


Figure-17. How Vuforia works [6].

In Figure-17, we can see how Augmented Reality works in mobile by using Vuforia Software. First the Camera will take the picture, and looks for the edge of the marker [17], after that, the picture will be converted into some pixel block like we already discussed. Then the tracker will determine the object if it's a word target, an image, a cylinder or if it's a moving object or a marker [17].

After the marker has been found, the application will now can render and augmented the virtual graphics in

database into the marker [17]. We can see how Vuforia helps augmented Reality in mobile by making it easy to develop the program and the database used in augmenting Reality [6].

REVIEW AND DISCUSSIONS

In Table-1, we will see the review of how edge detection can be used in some cases.

Table-1. Image Detection and Result.

Author's Name	Image Detection	Marker/Image	Result
(Siwa Suwanmane & Pedro, 2013) [5]	Edge Detection (Sobel, Robert and Prewitt)	Red Blood Cell	Stable but, the efficiency consistently dropped
(Kim Yongkwon et al.,2006) [16]	Edge Detection with Block based approach	Building and road	88,9% detection rate and can be improved up to 95,9%
(Jaeyoung Kim & Heesung Jun,2011) [17]	Edge Detection (Sobel Laplacian and Canny)	Simple Marker	Can be detected
(Ghassan Mahmoud & Ahmed Mohamed, 2015) [29]	Edge Detection (Sobel, Prewitt, Laplacian, Canny, and Gradient)	Fingerprints and some random images	Canny operator performed better
(Martin Hirzer, 2008) [22]	Line and Edge Detection	Simple Marker	Can only detect 4 corner
(Mayank Pandey et al., 2014) [11]	Edge and Corner Detection	Image	Possibility tha can be used in AR
(Vipul Chauhan & Manish Kayasth, 2015) [25]	Edge and Line Detection	Marker and Images	Line Detection is much more reliable than edge detection
(Ajaya Kumar et al.,2018) [24]	ARToolkit, Edge Detection	Simple Marker	Edge Detection have a problem detecting marker with a thin border



CONCLUSIONS

While the uses of Edge detection in Augmented Reality can be done the fact that people rarely uses it still become the main problem in this study. Many more method that much more easier method has already been found and for better or worse, make a better result than edge detection, for example line detection method. Also the main disadvantages of this detection is that there are certain angle and distance that can make detection cannot worked or will have a false detection. We hope that the edge detection can be developed in the future so it can still exist even in the newest technology like Augmented Reality.

REFERENCES

- [1] F. Zhou, H. B.-L. Duh and M. Billinghurst. 2008. Trends in Augmented Reality Tracking, Interaction and Display: A Review of ten years ISMAR. in 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, Cambridge, UK.
- [2] X. Zhang, S. Fronz and N. Navab. 2003. Visual Marker detection and decoding in AR systems: a comparative study. in International Symposium on Mixed and Augmented Reality, Darmstadt, Germany.
- [3] V. Torre and T. A. Poggio. 1986. On Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-8(2).
- [4] H. W. Thomas and S. P. Kumar. 2015. A review of segmentation and edge detection methods for real time image processing used to detect brain tumour. in IEEE International Conference on Computational Intelligence and Computing Research (ICIC), Madurai, India.
- [5] S. Suwanmanee, S. Chatpun and P. Cabrales. 2013. Comparison of Video Image Edge Detection Operators on Red Blood Cells in Microvasculature. in 6th Biomedical Engineering International Conference, Ampur Muang, Thailand.
- [6] D.-H. Suh. 2015. A Study on the Effectiveness of the Image Recognition Technique of Augmented Reality Contents. *Cartoon and Animation Studie.* 41: 337-356.
- [7] G. Shrivakshan and C. Chandrasekar. 2012. A Comparison of various Edge Detection Techniques used in Image Processing. *IJCSI International Journal of Computer Science.* 9(5).
- [8] K. Shah and K. Patel. 2013. Various Edge Detection Techniques: Survey, Implementation and Comparison. *International Journal of Advanced Research in Computer Sciences.* 4(4).
- [9] P. Selvakumar and S. Hariganesh. 2016. The Performance Analysis of Edge Detection Algorithms for Image Processing. in International Conference on Computing Technologies and Intelligent Data Engineering, Kovilpatti, India.
- [10] F. A. Pellegrino, W. Vanzella and V. Torre. 2004. Edge Detection Revisited. *IEEE Transactions on Systems, Man, and Cybernetics.* 34(3).
- [11] M. Pandey, P. Nair, M. Wadhwa and A. Chaudhary. 2014. Augmented Reality based on Image Processing. *International Journal of Engineering and Computer Science.* 3(5): 5924-5929.
- [12] S. Öztürk and B. Akdemir. 2015. Comparison of Edge Detection Algorithms for Texture Analysis on. *Procedia - Social and Behavioral Sciences.* 195: 2675-2682.
- [13] M. Lowney and A. S. Raj. 2016. Model Based Tracking for Augmented Reality on Mobile Devices. California.
- [14] T.-H. H. Lee. 2012. Edge Detection Analysis. *IJCSI International Journal of Computer Science Issues.* 5(6).
- [15] D. Kurpytė and D. Navakauskas. 2014. An Efficiency Analysis of Augmented Reality Marker Recognition Algorithm. *Electrical Control and Communication Engineering.* 5(1).
- [16] Y. Kim, K. Lee, K. Choi and S. I. Cho. 2006. Building Recognition for Augmented Reality Based Navigation System. in 6th IEEE International Conference on Computer and Information Technology, Seoul, Korea.
- [17] J. Kim and H. Jun. 2011. Implementation of Image Processing and Augmented reality programs for smart mobile device. in 6th International Forum on Strategic Technology, Harbin.
- [18] K. Kaur and S. Malhotra. 2013. A Survey on Edge Detection Using Different Techniques. *International Journal of Application or Innovation in Engineering & Management.* 2(4).
- [19] M. Juneja and P. S. Sandhu. 2009. Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain. *International Journal of Computer Theory and Engineering.* 1(5).
- [20] J. Jun, Q. Yue and Z. Qing. 2010. An Extended Marker-based Tracking System for Augmented Reality. in Second International Conference on Modeling, Simulation and Visualization Methods, Sanya, China.



- [21] S. R. Joshi and R. Koju. 2013. Study and Comparison of Edge Detection Algorithms. in Third Asian Himalayas International Conference, Kathmandu, Nepal.
- [22] M. Hirzer. 2008. Marker Detection for Augmented Reality Applications. Inst. for Computer Graphics and Vision. p. 27.
- [23] D. Gayathri, S. O. Kumar and S. R. C. 2016. Marker Based Augmented Reality Application in Education: Teaching and Learning. International Journal for Research in Applied Science & Engineering Technology. 4(VIII).
- [24] A. Dash, S. Behera, D. Dogra and P. Roy. 2018. Designing of Marker-based Augmented Reality Learning Environment for Kids Using Convolutional Neural Network Architecture. Displays.
- [25] V. P. Chauhan and M. M. Kayasth. 2015. Object Detection and Pose Tracking in Augmented Reality. International Journal of Science and Research (IJSR). pp. 2319-7064.
- [26] J. Canny. 1986. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-8(6).
- [27] S. Bhardwaj and A. Mittal. 2012. A Survey on Various Edge Detector Techniques. Procedia Technology. 4: 220-226.
- [28] M. Bennamoun. 1997. Edge Detection: Problems and Solutions. in IEEE International Conference on Systems, Man and Cybernetics, Computational Cybernetics and Simulation, Orlando, USA.
- [29] G. M. H. Amer and A. M. Abushaala. 2015. Edge Detection Methods. in 2nd World Symposium on Web Applications and Networking (WSWAN), Sousse, Tunisia.
- [30] Aziz, M. N., Purboyo, T. W., Prasasti, A. L. 2017. A survey on the implementation of image enhancement. International Journal of Applied Engineering Research.
- [31] Putra R. D., Purboyo T. W., Prasasti A. L. 2017. A Review of Image Enhancement Methods. International Journal of Applied Engineering Research.