Implementation of Object Tracking Augmented Reality Markerless using FAST Corner Detection on User Defined-Extended Target Tracking in Multivarious Intensities

Nurhadi¹, Saparudin², Adam N. ³, D. Purnamasari³, Fachruddin¹, Ali Ibrahim⁴

¹ Department of Informatics Engineering, STIKOM of Jambi, Jambi, Jambi, Indonesia
² Department of Informatics Engineering, University of Sriwijaya, Palembang, South Sumatra, Indonesia
³ Department of Computer Engineering, University of Sriwijaya, Palembang, South Sumatra, Indonesia
⁴ Department of Information Systems, University of Sriwijaya, Palembang, South Sumatra, Indonesia

Abstract. This paper presents a FAST Corner Detection object scanning to improve SLAM technique, which is SLAM, has a weakness in extracting features in the real-world object. The use of User Defined Target and Extended Tracking are for making this work more convenient and reliable. We can trace the object even though the object does not exist, so this improves the function of markerless itself. The use of Raycast is for the make labeling the objects or features in the scanned object. In this research, we executed multivarious intensity to test the FAST Corner Detection to increase function in real world feature extraction and prove it better than SLAM. Then, we got the result where is a brighter condition will get faster recognition. The best environments for augmentation are in the range of 80-190, they took in less than 1 second. On the contrary, the intensity outside of the range such as ≤50 or ≥200, has a deficiency of augmentation. The range of ≤50, there was no augmentation cause of low intensity. For the range of ≥200, we haven't made measurements as we don't have the resources yet, but we hypothesize that the object would be corrupt or we may call it was overexposure cause of the intensity is too high. So, this could also lead to augmentation will not occur.

1. Introduction
In the Augmented Reality development, markerless has become a better form of application than the marker. Unlike marker, it doesn't require certain forms. Imagine if we bring the physical marker to perform augmentation in everytime and anywhere. That's not flexible and very inconvenient. However, it does not mean markerless has no obstacles. It has a low position accuracy than the marker. In the research of [1], markerless is lower in positioning accuracy because in it module that despite having the recognition range ranges ± 0.01º in longitudinal and latitude but not on the system, the system relies on the localization technology. We need more reliable technology to fix this obstacle. They may be an RFID, Ultra Wide Band, or SLAM.

Notwithstanding, there are some vendors or developers whose still actively developing Markerless technology that has penetrated the domain of Mixed Reality (MR) and even going to Extended Reality (XR). For example, Qualcomm, Microsoft, Google, Apple, Vuforia, and Kudan. This happens because markerless has the potential for more sophisticated, more broadly and more efficient to implement. So,
markerless is considered more suitable for development in future work. That is our consideration to make markerless the object of research this time.

SLAM's main role in AR is still lacks in extracting real-world scale information. It wasn't able to display the object content where it was a feature of the application of AR. Therefore, there needs to be tracking based on the object model (object tracking) [2]. In addition, SLAM also experiences limitations in compile time. With those factors, in this paper, we proposed the application of User Defined Targets supported by Extended Tracking within the Vuforia scope of work. Then, we will deploy to the Android platform as an entity to test the feasibility of the technique. The use of the FAST Corner Detection algorithm is to increase the feature extraction. So, the goal of increasing augmentation results that compared to SLAM achieved. The application of User Defined Target-Extended Tracking will further increase the convenience and flexibility of the proposed system. We will also present system testing at various intensities where the aim is to find out how influential the intensity is in the system. We tested the intensity at which augmentation would be successful.

2. Related Work
Tracking-based object model refers to the use of the line model predefined CAD or 3D texture models are reconstructed from the target tracking (eg, objects, space) and match them with a direct view object during the scan to reveal the 6DOF pose. This makes it possible to reduce the use of markers disturbing because the object itself is used as a target while maintaining full 3D information of the object being tracked so it automatically provides the correct coordinate frame for augmentation AR [3].

The technology has evolved and is very popular is applied to markerless SLAM technology. SLAM is a technology approach that seeks to address the challenges in estimating the camera pose of the features of the surrounding and simultaneously build a 3D construction [4]. Where the basis of this approach is the development of technologies in achieving the nomination of the autonomous control of the robot [5]. The most influential research on the development of AR is a monocular visual SLAM (MonoSLAM) of EKF (Extended Kalman Filter) [6]. In its development, some of the following approaches greatly affect the PTAM, where this approach minimizes the computational crimped by separating tracking and mapping as well as use the Bundle Adjustment (BA) to optimize the mapping [7]. Then, ORB-SLAM, this approach is claimed to have a very mapping management better than ever due to the use of techniques loop closure. With the effectiveness of this approach, Microsoft Inc. Microsoft finally released HoloLens, according to ORB-SLAM they have done an impressive performance in localization, especially in crowded environments reconstructed a map [8].

In the research by [3] , a localization accuracy camera with 6 Degrees of Freedom (6DOF) corresponding to the position and orientation in a coordinate system application, remain the primary technology that enables Augmented Reality. That is because to put a realistic virtual object in the real world, integration into it as well as for user interaction. Traditionally, the approach relies on the localization of the target tracking as fiduciary artificial or natural features of rich 2D image texture [9]. However, these approaches are considered obsessive to the scene while limiting an application to a range of visibility of the target tracking and become very sensitive to the target occlusion [10]. There are two different directions, trying to reduce the need for markers, which is based on tracking models and approaches simultaneously localization and mapping (SLAM).

3. Augmented Reality Systems
3.1 Markerless
Markerless AR is a term used to indicate an Augmented Reality application that is not bothered in the knowledge of the user's scope in build the 3D world into the scene and doing its accuracy in that virtual space [11]. In recently, the majority of AR included in the category of AR marker which requires the user to place the tracker. The tracker image is encoded with information that is translated by complex software to produce 3D objects maintain spatial orientation in a scene. An implementation of markerless is based on the performance of GPS technology.
On the other hand, the application of AR markerless recognize things that are not directly given to the previous application. This scenario is much more difficult to implement because of the introduction of running algorithms in AR application you must identify the pattern, color, or some other features that may be present in the camera frame. For example, if the algorithm we can identify the car, it means that we will be able to trigger the application of AR action whenever the car is detected in the camera frame, without having to provide a picture with all the cars in the world, or in other words, is called a training database.

### 3.2 SLAM
SLAM approach tries to tackle challenging problems in estimating the camera pose of the features in the environment while learning the 3D structure of this unknown environment. Seminal Works in visual SLAM-eyed is the Extended Kalman Filter (EKF) based MonoSLAM [6], followed by a large number of important publications in the field [2]. PTAM [7] SLAM reduce computing costs by separating the parts tracking and mapping to two different threads. Thus, the thread mapping computationally more expensive ones can run at lower frequencies without affecting the tracking to be real-time. Additionally, PTAM introduces methods Bundle Adjustment (BA) in visual SLAM for mapping optimization.

ORB-SLAM [8] follows the same principle but extends the capabilities of the application area coverage maps effective management and loop closure techniques. The basis of the probability of SLAM is as follows:

- Gaussian PDF

\[
p(x) = \det(2\pi \Sigma)^{\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \bar{x})^T \Sigma^{-1} (x - \bar{x})\right)
\]

SLAM algorithm or software is not specific but rather refers to the problem of trying to localize simultaneously (i.e. find the position/orientation) multiple sensors to the environment, while at the same time mapping the structure of the environment. Most modern visual SLAM system is based on a set point tracking through successive camera frames, and use these tracks to triangulate their 3D position; while using the approximate location of the point to compute the camera pose that can be studied.

### 3.3 Proposed Method
Proposed method in this research is the application of the algorithm of the method FAST Corner Detection on a system built with markerless. Our proposed system flow method is where we do the test as Figure 2, where the object of his real world in the form of input and then will be scanning with FAST Corner Detection method, this method will scan by taking the features of real-world objects before, the method of taking this method is to create a simulation of mapping the area to be in scanning (see Figure 6), and the result was such feature extraction will be stored in the database then. Stages of User Defined target application is done so that the system can be more robust to the environment, it is because the User Defined target is very reliable in scanning the augmentation.
application to be made. User Defined Target will make a wide range of angles of the object to be in augmentation (see Table 1). Then, by adding the Extended Tracking system in the proposed method will be more reliable and flexible due to the use of these techniques will make us do not have to bother to always direct object scan results before, in other cases can be implemented into a system in which the introduction of the parts on a tool, in other words, without his presence or real objects that have been scanned earlier, we are still able to access the system.

So we don’t have to worry if we want to move the point of view. Raycasting is applied to provide information on the features highlighted so with this we can know the feature or in other words, we can make a name label. This is very useful because, in addition, we get the information, we can also claim that the extraction characteristics that do work well. Then, we entered the stage of testing of the system implementation. We use the Android platform as a medium for testing with the operating system specification 5.0. Previously we had to make sure we have to deploy or build in the version for Android platform, please do the settings on the target SDK. Finally, we may execute the stage to perform the system in five intensities. By perform this stage, we may know how the FAST Corner is able to improve the SLAM. This is worthwhile as we get the information and also we may claim that the extraction characteristics of the algorithm do the work finely.

3.4 3D Object Tracking

In this section, we present a 3D object tracking path we are proposing. Having defined the mathematical notation that will be used throughout this work in Section 3.4.1, we describe the preparatory steps for an object to be used for tracking in Section 3.4.2. Next, we describe the tracking algorithm itself.

3.4.1. Math Notation. 3D Object Tracking is the stage where we can create a system that can recognize and track the desired object. Just as the concept of the image processing, in computer graphics also apply the name identification or pattern recognition[12]. In section 3, it has been described that the use of SLAM in augmented reality is making the feature extraction (feature extraction) so that the concept can be implemented markerless. In[3], the problem is handled from the camera tracking 6DOF pose consists of the rotation matrix estimate $R_{cw} \in [R]^{3\times3}$ representing the rotation of the world coordinate system W to the C camera coordinate system, and the translation vector $W_c \in [R]^{3}$ containing the home position in the world coordinate system of the camera coordinate system. In the case of object tracking, object model coordinate system can be regarded as the world coordinate system W. Using homogeneous representation of 3D points $p \in [R]^{3}$ the object coordinate system and homogeneous representation of 2D points $p \in [R]^{3}$ in the image coordinate system of the camera, the camera pose estimation problem requires a camera pose estimation $[R_{cw} | W_c]$ such that the mapping:

$$p = K[R_{cw} | W_c]P,$$

(2)

The best fit a set of known 3D to 2D correspondence $M = \{p_i \leftrightarrow p_i\}, K \in [R]^{3\times3}$ represents the intrinsic camera matrix.
3.4.2 Algorithm Object Tracking. The registration procedure at this stage of the 3D object tracking in place so that the object can be on the track with the algorithm. The algorithm used for 3D Object Scanning is a "FAST Corner Detection". At this point, we use the features of the feature extraction algorithm to scan the whole object through FAST Corner Detection. There is some good detection and even many of them are really nice but when viewed from the perspective of real-time applications, they are not fast enough. One of the best examples is the mobile robot SLAM (Simultaneous Localization and Mapping) which have limited computing resources. As a solution to this, FAST (Features of Segment Test) algorithm proposed in [13] (later revised in 2010) [14]. Here is a FAST Corner Detection algorithm:

1. Choose one pixel $P$ the image which must be identified as a point of interest or not. Suppose intensity $I_p$.
2. Select the appropriate threshold values $t$,
3. Consider a circle 16 pixels around the pixel being tested. For example, as shown below.

![Figure 3. Examples of application FAST Corner Detection.](image)

4. Now pixels $P$ is a corner (corner) if there are one set $\eta$ adjacent pixels in a circle (16 pixels), all of which are brighter than $I_p + t$ or all darker than $I_p - t$, (Published as red dashed line, see the Figure 3). $\eta$ was chosen to be 12.
5. Each pixel (say $x$) In 16-pixel can have one of three states:

   \[
   S_{p\rightarrow x} = \begin{cases} 
   d, & I_p - t \leq I_{p-x} \text{ (darker)} \\
   s, & I_p - t < I_{p-x} < I_p + t \text{ (similar)} \\
   b, & I_p + t \leq I_{p-x} \text{ (brighter)}
   \end{cases}
   \]  

   \[ (3) \]

6. Depending on the status, the feature vector $P$ is divided into three subsets, $P_d, P_s, P_b$.
7. Define a new boolean variable, $K_p$, Which is true if $P$ is the angle and one otherwise.
8. Use ID3 algorithm (decision tree classifier) to query each subset using a variable $K_p$ for knowledge about the correct classes. this pick $x$ which produces most of the information about whether the candidate pixel is the angle, measured by entropy $K_p$.
9. It is applied recursively for all the subsets to zero entropy.
10. Trees decisions made are used for rapid detection of the other images.

3.5 User Defined Target
As like as the Image Target, User Defined target allows the developer to select an image in front of the time recognized by the application. It allows end users to choose an image at runtime [15]. It should be taken and viewed under the lighting that is bright enough and spread out. For the surfaces, they must be evenly distributed.

In this study, we show you how to use the feature for the User Defined target objects to instance a class of TrackableSource that can be used to create new Trackables runtime. Two new classes, ImageTargetBuilder and ImageTargetBuilderState introduced:
If the latter two options are disabled, it is possible to create applications where scanning modes are never abandoned and tracked the target immediately after making it. With this configuration, it is possible to make multiple targets in succession.

Class ImageTargetBuilder expose APIs to control the progress of the development and take TrackableSource for instantiating new tracking after successful completion.

3.6 Extended Tracking
Extended Tracking is a concept that poses the target information to be available even when the target is no longer within the field of view of the camera or cannot be traced directly for other reasons. Extended use Driver Tracker tracking to improve tracking performance and maintain tracking even when the target is no longer visible.

We may use this feature to facilitate the creation of applications that are more powerful, because the addition of anything inherent in this target will still be there. In practice, this means that after you point your device away from the initial target, any augmentations to maintain their position in relation to the real world and are consistent with the initial terms of reference set by the target. More detailed environment and feature-rich, Extended Tracking better work.

4. Implementation and Results
4.1 Environment. In this study, we used a database of 3D Arduino Uno objects from Warehouse. For its development kit augmented reality, we are using Vuforia Augmented Reality SDK 6 in Unity IDE v5. The Android platform we choose as an emulator or media testing its system of at least Android 5.0.

4.2 Taking the object features. The use of FAST Corner Detection applied in making 3D features of this object. There are 405 points for Arduino Uno is a 3d object. For scanner environment, we suggest...
these specifications: light intensity about 50 cd – 200 cd, camera distance between object is about 20-25 (cm), camera lens about ≤ 8 MP, and capture in the top and left-right side camera angle.

Figure 6 is a system layout in positioning or object 3d vector matrix for later. 3d object retrieval features quite done with x and y-axis position taken because of the features is the upper part of the Arduino Uno. After making this feature on the 3d object where the results obtained for the Arduino Uno with specifications as shown in Table 1 and Table 2, we get the results of the detected features as many as 405 points.

![Figure 6. Layout for feature extraction.](image1)

![Figure 7. Results of retrieval FAST Corner Detection feature.](image2)

![Figure 8. Application of User Defined target system.](image3)

4.3 Storing the Vuforia database. For storage of results, Vuforia 4.1. We use an online database that goal was the result of 4.1 can be implemented to the Android platform. At this stage, it can be viewed on the basis of the creation of applications Augmented Vuforia Reality.

4.4 Implementation User Defined target. UserDefinedTargetBuildingBehaviour will be shooting at various events to registered UserDefinedTargetEventHandlers.

| Table 1. Presenting the results of the application of User-DefinedTarget. Retrieval features that previously performed very decisive result of User-DefinedTarget. |
|-----------------|-----------------|
| Number (N) | Real World Scene (A) | UserDefinedTarget Scene (B) |
| 1 | ![Image](image4) | ![Image](image5) |
| 2 | ![Image](image6) | ![Image](image7) |
| 3 | ![Image](image8) | ![Image](image9) |

In Table 3 (a) shows a device capturing an image target in its field of view (FOV). The result is Figure 3 (b), which shows the image target in view, along with augmentations. As 3 (b) shows, the augmentations of buildings on top of a wooden table are displayed on the screen of the device. This encourages the user to pan to Reviews These items by moving the angle of the camera to point upwards to see the top of the buildings, thereby making the Image Target disappear from view.

4.5 Application of Extended Tracking
When the Positional Device Tracker is disabled, all augmentations disappear, because the Target reference frame is lost. With the Positional Tracking Device Tracker and Extended, the buildings Werner visible even when the Image Target is not in the camera view. Figures 9 and 10 illustrate this:

![Figure 9](image9.png)  ![Figure 10](image10.png)

**Figure 9.** Augmentation in -90 linear angle side (baseline)  **Figure 10.** Application of Extended Tracking successfully implemented

### 4.6 Additional System

Additional feature in this system are as follows: Labeling using Raycast, Lean Touch [unity asset store] using basic concepts 6DOF but depending on the features extracted so there are only 3 DoF i.e., up-down, right-left and front-rear as well as zoom in and zoom out [3][6][7][16][17].

![Figure 11](image11.png)  ![Figure 12](image12.png)

**Figure 11.** Physical Raycast Labeling Features  **Figure 12.** Features Lean Touch.

This function returns a RaycastHit object with a reference to the collider that is hit by the ray (the collider property of the result will be NULL if nothing was hit). The layerMask can be used to detect objects selectively only on certain layers (this allows you to apply the detection only to Arduino 3d object parts, for example in this works).

### 4.7 Object Tracking in Different Intensity

Intensity refers to the amount of light or the numerical value of a pixel. For example, in grayscale images, it is depicted by the grey level value at each pixel (e.g. 80 is darker than 120). Meanwhile, brightness is a relative term. It depends on your visual perception. Brightness comes into picture when we try to compare with a reference.[17]

The intensity of our calculations carried out by taking the conditions represented in the image and then we do the calculation as follows:

\[
I(\lambda) = \rho(\lambda)E(\lambda)(NL)R(\sigma, \theta)
\]  \hspace{1cm} (4)

Where (NL) representation the orientation of the surface normal N and illuminance direction L relative to the viewer and \(R(\sigma, \theta)\)roommates is a reflectivity function characterizes the textural and specular that surface's properties. Even though an equation of image intensity (7) is a wavelength approach \(I(\lambda)\), this dependence will be ignored in subsequent derivations [18]. Besides that, you may have Utilize tools such as MATLAB from improfile or you just Easily help by using ImageJ.
Table 2. Here we show the results of the comparison are 5 kinds of circumstances intensity and one additional state of intensity.

| No. | Intensity (cd) | Tracking Results | No. | Intensity (cd) | Tracking Results |
|-----|---------------|------------------|-----|---------------|------------------|
| 1   | <50 cd        |                  | 4   | 120 ± 170 cd  |                  |
| 2   | 50 ± 70 cd    |                  | 5   | 180 ± 190 cd  |                  |
| 3   | 80 ± 110 cd   |                  | 6   | > 200 cd      | MEASURE NOT YET   |

We do object tracking in a state of normal intensity is in the range 65-200 cd [19]. In RGB channel, there are 256 * 256 * 256 possible (uint8) and 65536 * 65536 * 65536 (uint16).

In the digital image processing perception, the intensity of an image could refer to a global measure of that image, such as the mean pixel intensity. A relative measure of image intensity could be how bright (mean pixel intensity) the image appears compared to another image.

In Figure 13 in the span of less than 50 does not occur augmentation, is due to the intensity is too low (underexposure). For the condition of more than 50 to 200 seem that the higher the intensity, the faster they were augmented. However, in the case of more than 200, we have not been able to hold it due to limited tools for creating an intensity of more than 200, but we hold the principle of overexposure [20], so we predicted that the object will not be augmented.

5. Conclusion
In this paper, we represent the FAST Corner Detection algorithm in the form of the application into an android app. We use FAST Corner Detection algorithm due to deficiencies in SLAM in limited computing resources. FAST Corner Detection is used in the retrieval phase extraction features of the object to be augmented. The results obtained prove FAST Corner Detection capable of making a good trait in a way to cover the area of the object that will be made later on the system input.
With our method, users increasingly easy to perceive the advantages markerless by adding techniques, and Extended User Defined Target Tracking. Where User Defined Target will display the results of augmentation in accordance with the position of the object. This benefit because the accuracy in accordance reality shows something of the object. While Extended Tracking will make very useful reliability that users do not have to worry about these objects are missing from the camera or even manual focus switching positions with different background whatsoever. Raycasting additions made to add the functionality of the feature extraction which will display information raycasting predetermined area and selected by the user.

Even more interesting, we present the results of augmentation under varying conditions of the range of 0-200 cd. There are comparisons that can be inferred that the higher the intensity, the faster the augmentation. But in the case of > 200 cd, we have not been able to test because of the limitations of the tool in generating ambient light > 200 cd. However, we can hypothesize with by overexposure event where objects are illuminated will cause the loss of information so if it is too bright then the object is not detected properly and augmentation failed. Those conditions may have been improved by adding enhancements such as clustering or classification in the feature extraction stage.

6. References
[1] J. C. P. Cheng, K. Chen, and W. Chen, “Comparison of Marker-based AR and Markerless AR: A Case Study on Indoor Decoration System,” 2017, vol. 6, no. July, pp. 483–490.
[2] T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual SLAM algorithms: a survey from 2010 to 2016,” IPSJ Trans. Comput. Vis. Appl., pp. 1–11, 2017.
[3] J. Rambach, A. Pagani, M. Schneider, O. Artemenko, and D. Stricker, “6DoF Object Tracking based on 3D Scans for Augmented Reality Remote Live Support,” pp. 1–22, 2018.
[4] R. Egodagamage and M. Tuceryan, “A collaborative Augmented Reality framework based on Distributed Visual SLAM,” in International Conference on Cyberworlds, 2017.
[5] J. Halvarsson and J. Halvarsson, “Using SLAM-based technology to improve directional navigation in an Augmented Reality game,” Interaction Technology and Design, 2018.
[6] H. Kim, S. Prof, and A. Davison, “Qualcomm Augmented Reality Lecture Series: Visual SLAM with an Event-based Camera,” 2015.
[7] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces.”
[8] R. ul Mur-Artal and D. Tard’os, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” pp. 1–8, 2017.
[9] J. Park, S. You, and U. Neumann, “Natural Feature Tracking for Extendible Robust Augmented Realities.”
[10] E. Marchand, H. Uchiyama, and F. Spindler, “Pose Estimation for Augmented Reality: A Hands-On Survey,” IEEE Trans. Vis. Comput. Graph., vol. 22, no. 12, pp. 2633–2651, Dec. 2016.
[11] S. Kote and B. Borkar, “A Survey on Marker-less Augmented Reality,” vol. 10, no. 13, pp. 639–641, 2014.
[12] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, “Robust 3D Object Tracking from Monocular Images using Stable Parts,” PATTERN Anal. Mach. Intell., vol. 8828, no. c, pp. 1–14, 2017.
[13] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” pp. 1–14.
[14] E. Rosten, R. Porter, and T. Drummond, “Application of dynamic ray tracing in the 3-D inversion of seismic-reflection data,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 1, pp. 105–119, 2010.
[15] Y. Lee and J. Choi, “Tideland Animal AR : Superimposing 3D Animal Models to User Defined Targets for Augmented Reality Game,” vol. 9, no. 4, pp. 343–348, 2014.
[16] E. Pennestri, M. Cavacece, and L. Vita, “On the Computation of Degrees-of-Freedom: A Didactic Perspective DETC2005-84109,” in Proceedings of IDETC’05 2005 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2005, pp. 1–10.
[17] K. Chen et al., “MARVEL: Enabling Mobile Augmented Reality with Low Energy and Low Latency.”
[18] W. A. Richards, “Lightness scale from image intensity distributions,” vol. 21, no. 14, pp. 2569–2582, 1982.
[19] B. K. P. Horn, “Understanding Image Intensities,” Artif. Intell., vol. 8, pp. 201–231, 1977.
[20] C. Yang, H. Feng, Z. Xu, Q. Li, and Y. Chen, “Correction of overexposure utilizing haze removal model and image fusion technique,” Vis. Comput., pp. 1–11, 2018.