ISSN 1975-4736 ⓒMITA2022

Computer Vision-Based AR Shooting Game

*HONGNYEOM SUNG, **JONGHYEON PARK, ***HEEJAE HWANG, †CHILWOO LEE
* Department of Software Engineering, Chonnam National University, korea
, *, † Department of Computer Engineering, Chonnam National University, korea
*e-mail : 6590f1@naver.com, **belline0124@gmail.com, ***certificate@kakao.com, †leecw@jnu.ac.kr

***Abstract*—Real-world shooting games that can be enjoyed outdoors require dedicated equipment such as helmets and goggles to avoid danger. Additional tools such as laser sensors and paint balls are also needed to determine the shooting. This means that there is room for improvement in terms of accessibility and safety to enjoy games easily and safely anytime, anywhere. To solve this problem, this paper proposes a computer vision-based AR shooting game that can be experienced in the real world while improving the accessibility and safety of shooting games. To implement this game, we propose a basic algorithm that recognizes the movement of the camera from the input image to track the player's trajectory and recognize the opponent (target) by artificial intelligence techniques. A prototype game was produced using the proposed algorithm, and it was confirmed through experiments that the user's trajectory estimation and target (opponent) recognition were possible in real time.**

Keywords—AR Shooting game; First perspective shooting game; odometry; target detection;

I. INTRODUCTION

Shooting games enjoyed outdoors can be used to attack in response to the movement of enemies while using real-world topography, so we can feel the best realism that the game requires. Since this sense of reality or tension is the most important factor in enjoying the game, reality-based First Perspective Shooting (FPS) games have attracted a lot of attention in the game industry. The table below shows the characteristics of the currently commercialized FPS-related games.

TABLE I. COMPARISON OF FEATURES OF COMMERCIALIZED REAL-WORLD SHOOTING GAMES

Name	Charicteristics
Suvival Game	- It's a traditional survival game that uses paint guns -Wear of designated protective gear in authorized locations is essential because actual (paint) bullets are used
RazorTag[1]	-Play the game using headgear with laser detection sensors and a gun model that can shoot a laser
father.io[2]	-Play the game with a smartphone equipped with a laser sensor and high-performance GPS capability

Games of the genres mentioned in Table 1 have the following problems in common. The first problem is that accessibility (the degree to which games can be easily enjoyed anytime, anywhere) is low. Most survival games use paint guns, and no matter how paint guns are, there is a high risk of injury when a paint bullet hits a person's vital point when shooting. Because of this risk, existing survival games will be played under the control of the supervisor and wearing protective gear in a licensed place. To improve this risk, AR games such as RazorTag and father.io have been developed. However, these games also have a problem that is difficult to access because they need to use a mock gun equipped with a laser sensor and detection equipment to determine the shooting.

The second problem is the accuracy of the strike. The conventional method is to judge a shot using a laser sensor. Due to the nature of the laser, it is difficult to detect the signal if the part that detects the laser signal is slightly covered or the direction is reversed. In addition, even if other parts of the body other than the hit determination part (laser signal detection sensor part) are hit, the hit determination is not performed.

These problems are not limited to augmented/virtual reality-based content (hereafter referred to as metaverse content) belonging to the FPS genre. Since most metaverse content today requires players to purchase expensive HMD such as Oculus and HoloLens, and can be used mainly indoors, the above-mentioned problems are always a major obstacle to activating metaverse content.

To solve this problem, this paper describes the implementation of AR-based FPS games that can be simply enjoyed in the real world by combining computer vision-artificial intelligence technology and game engine technology based on smartphones. This game uses computer vision and artificial intelligence technology to recognize people in the real world and to track players' three-dimensional positions, combining physical play environments and virtual objects in real life. The composition of this paper is as follows.

Chapter 2 describes computer vision-based Odometry tracking. Chapter 3 describes object recognition algorithms. Chapter 4 then describes the mobile environment, experiments, and results of the system implemented so far. Chapter 5 concludes the paper by describing the significance of this study.

II. ODOMETRY ESTIMATION

The key technologies to implement the game proposed in this paper are odometry estimation with computer vision and object recognition technology using artificial intelligence technology. The system configuration based on the two elements is shown in Figure 1 below.

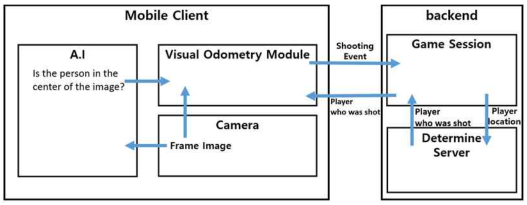

Figure 1. The overall configuration of the computer vision-based AR shooting game proposed in this paper.

A. Computer Vision-Based Odometry Estimation

In order to implement the interaction, namely attack and defense, in the FPS game, it is first necessary to recognize the movement of the players in the field and combine it with the situation of the virtual space. To estimate the movement we usually use the Methods using GPS, sensors of Inertial force, and Visual Odometry. Since FPS requires a higher level of accuracy in path estimation, algorithms that combine those methods are generally used.

Before conducting this study, Madgwick's work[3] based odometry algorithm is tested by an acceleration sensor of a smartphone. The result is shown in Figure 2. When tracking a position using an IMU sensor in a mobile device, error occurs continuously and that is accumulated every frame. In addition, when a sudden change in movement and rotation occur simultaneously, the error becomes larger, so it is difficult to accurately estimate the path by using only the IMU sensor. Therefore, in this paper, an algorithm for estimating a three-dimensional path using computer vision technology was developed and used. In this study, Odometry was calculated as follows by referring to Avi Singh's github[4] and report[5].

B. Extraction of feature points using FAST(Features from Accelerated Segment Test)[6][7]

In steps 1 and 2 of the algorithm, it is necessary to extract a feature point every frame. Algorithms for finding key points include SIFT, SURF, and Harris Corner detection. Since real-time processing is required within the limited resources of mobile devices, in this paper, we use the FAST algorithm to extract key points.

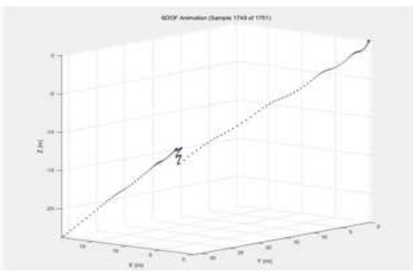

Figure 2. The result of path estimation using the acceleration sensor of a smartphone. Errors occur where the movement changes rapidly as the errors accumulate.

Step 1. Obtain the current frame and the previous frame image

Step 2. Identify feature points through FAST algorithms in previous frames

Step 3. Through the KLT Tracker, find the feature point identified in the corresponding feature point in the current frame

Step 4. Estimate the Essential Matrix using the RANSAC algorithm

Step 5. Calculate the trajectory between frames by obtaining the rotational matrix R and the translation vector t from the Essential Matrix

In this method, as shown in Figure 3, when determining whether the central pixel p is a feature point or not, draw a circle with a radius of 3 around the pixel and see how much the difference in brightness between 16 pixels on the circumference and the central pixel occurs to determine whether the feature point or not. Whether p is a feature point or not is determined as a feature point when n points brighter than p or more than n points darker than p out of 16 points centered on p exist continuously

C. KLT (Kanade-Lucas-Tomasi) algorithm-based feature point correspondence extraction

In order to estimate the odometry, motion information must be calculated from the continuous image. That is, it is necessary to find a relationship to where the feature point found in the previous frame has been moved in the next frame. In this paper, the KLT algorithm was used to find corresponding feature points between frames.

The KLT algorithm uses the constraint $I(x, y,t) = I(x + \Delta x, y + \Delta y,t + \Delta t)$ assuming that the brightness of a feature point does not change much in an adjacent frame, and that the adjacent feature point also has similar motion when it moves. By developing this equation into a Taylor series and removing the higher-order term to solve the system of equations, it is possible to calculate the movement between the two images. The point at which the root of the quadratic equation is minimized, that is, the point at which the differential value is zero, becomes the motion vector (u, v) in the horizontal and vertical directions. It can be estimated that the feature point of the previous image corresponds to the feature point of the motion (u, v) by obtaining the feature point from the first image and extracting the motion from the next image.

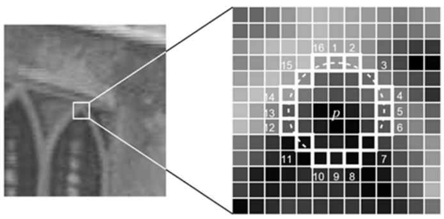

Figure 3. Overview of the FAST algorithm[7].

ISSN 1975-4736 ⓒMITA2022

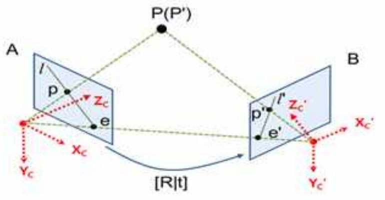

Figure 4. Correspondence point relationship in the epipolar geometry plane

D. Essential Matrix Estimation Using RANSAC

Epipolar geometric constraints are used to estimate 3D depth information using multiple cameras. Figure 4 shows the configuration. When the feature point p of the left image (previous image) has a corresponding relationship with p' of the right image (current image), the Essential Matrix constraint is established in the epipolar geometry space. Therefore, the Essential Matrix may be obtained using coordinate information of corresponding feature points, and the Essential Matrix may be decomposed to determine the motion of rotation and translation([R|t]).

In this paper, the Essential matrix is calculated using a 5-point algorithm using five feature points. To obtain an accurate estimate, a matrix was obtained by applying five feature point pairs from the corresponding point list to the RANSAC algorithm.

The RANSAC (RANdom SAmple Consensus) algorithm randomly selects five corresponding points and then tentatively obtains one Essential matrix. After applying this matrix to each corresponding point, calculate the error, record the corresponding points within a certain range as inliers, and estimate this matrix as the optimal matrix if the number of inliers falls within a certain range of the total number of corresponding points. In other words, since the matrix is obtained from the five points with the highest consensus among the combinations obtained in the correspondence, it is a method of selecting the model supported by the largest number of data among various models. As shown in Figure 5, even if the outlier is mixed with the data, the optimal model with the most inliers is obtained, so the matrix with the least errors can be obtained.

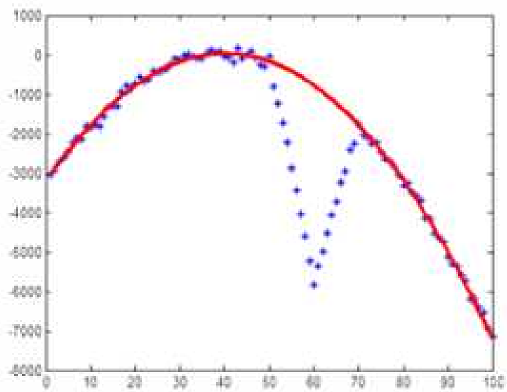

Figure 5. Example of curve model extraction using RANSAC.

E. Pose Estimation with Essential Matrix

Since Essential Matrix is a transformation matrix between frames, it is a matrix that satisfies the corresponding feature points between two frames, the relationship between them. At this time, when the relationship between the camera coordinate axes of two frames is expressed by Equation (1) and the Essential Matrix is expressed by Equation (2), E becomes the external enemy of T and R as shown in Figure 4, Equation (3) is established.

$$P' = RP + t \tag{1}$$

$$E = [t]_X R \tag{2}$$

$$P'^T E P = 0 \tag{3}$$

In Equation (2), R represents the rotation matrix and the external enemy of the translation matrix as a matrix, so R, t can be obtained through the following equation with the constraints in the SVD (Single Value Decomposition) and the rotation matrix in the Essential Matrix.

$$\begin{cases} E = U\sum V^T \\ [t]_X = VW\sum V^T \\ R = UW^{-1}V^T \end{cases} \tag{4}$$

III. Recognition of Objects Through Artificial Intelligence

In FPS games used outdoors, the accuracy of shooting determine is very important. However, it is difficult to accurately determine the direction of each player's shot in the event of a shooting event because only information on which player shot the ball is available in the path tracking through a computer vision. In order to solve this problem and accurately determine the shooting, this paper uses artificial intelligence to check whether there is actually a person in the center of the camera at the time of shooting.

F. Pose Detection

The method of recognizing people and extracting skeleton from the Pose Detection provided by Google ML Kit[8] checks the presence of people in the image through the Pose Detector along the pipeline shown in Figure 6. And it can obtain the location, size, and directional information of the person called Person Box as shown in Figure 8. Then, through the Pose Tracker starting with the human head through the artificial intelligence model, and extracting the keypoints, or landmark information (coordinates on the image) in Figure 7.

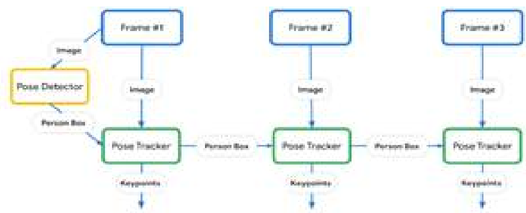

Figure 6. The pipe line process used in Google ML kit.

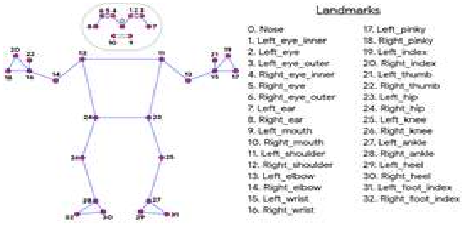

Figure 7. The keypoints used for human target recognition.

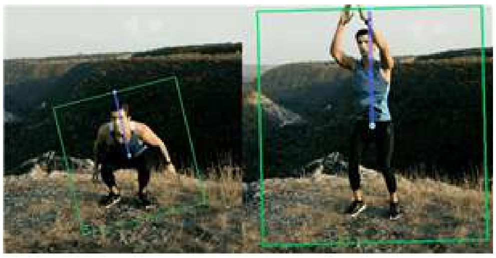

Figure 8. The person box area extracted from input image.

Figure 9. An example of recognizing a human target and combining the recognition results.

G. *Landmark information for determine shooting event*

Finally, the location information on the extracted LandMark image is used to check whether the player actually shot a person. At this time, there are two shooting areas, and if the range of shooting determination composed of the coordinates on the image of LandMark on the head and upper body overlaps the coordinates of the center of the image, the shooting determination is made.

Figure 10. Path estimation result with proposed method.

IV. EXPERIMENT AND RESULTS

To implement the proposed game, a Samsung Galaxy S22+ smart phone was used. To check the performance of the algorithm before implementing it directly on a smartphone, the accuracy was checked using video files taken with the smartphone camera. After checking the accuracy, the game was implemented on the smartphone and the route was calculated in real time.

Figure 10 shows the results of taking images with a mobile camera, extracting and testing frame images, and mapping the estimated Odometry to a map with actual movements.

In the actual data used in the experiment, the overall translation vector t was found fairly accurately between frames, but on the contrary, the rotation matrix value could not be found accurately, so the accuracy was high in linear motion, but relatively low in reverse rotation. It is presumed that an error occurred in the calculated rotation matrix when turning a curve rapidly as the cause.

V. CONCLUSION

This paper describes the path tracking algorithm and human recognition algorithm for hit determination developed to realize real-world metaverse content in FPS games. This method estimates the player's path based on computer vision and uses the skeleton information to estimate the human area to determine a hit. As a result of the actual experiment, it was confirmed that the accuracy of the trajectory calculation result through Vision Odometry was improved compared to the result using the INS sensor built into the current low-end mobile device. However, it is inaccurate in certain parts, especially in the part that obtains the rotation matrix. As a result, we plan to develop it into an element technology that can create metaverse content that is easy to use anywhere while providing realism and safety even without special equipment or a set environment. Authors and Affiliations

Acknowledgment

This research is supported by Ministry of Culture, Sports, and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program(R2020060002) 2020.

References

[1] http://www.lasertagsports.com/

[2] https://father.io/

[3] Sebastian O.H. Madgwick, "Estimation of IMU and MARG orientation using a gradient descent algorithm", 2011

[4] https://github.com/avisingh599/mono-vo

[5] http://avisingh599.github.io/assets/ugp2-report.pdf

[6] E. Rosten and T. Drummond, "Machine learning for high speed corner detection," in 9th Euproean Conference on Computer Vision, vol. 1, 430–443p, 2006.

[7] Viswanathan, Deepak Geetha. "Features from accelerated segment test (fast)." Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK. 2009.

[8] https://developers.google.com/ml-kit/vision/pose-detection