International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 11, Issue 6, November-December 2024

Impact Factor: 7.394

DOI:10.15680/IJARETY.2024.1106019

Augmented Reality Using Camera to Object Detection

Ms. B. Ranjitha[1], Mr. S. M. A. Meharoof[2], Mr. V. Sanjay[3], Mr. S. Krushidhar Reddy[4]

Assistant Professor, Department of Computer Science & Engineering, Guru Nanak Institute of Technology, India[1]

Student, Department of Computer Science & Engineering, Guru Nanak Institute of Technology, India[2]

Student, Department of Computer Science & Engineering, Guru Nanak Institute of Technology, India[3]

Student, Department of Computer Science & Engineering, Guru Nanak Institute of Technology, India[4]

ABSTRACT: Augmented Reality (AR) has revolutionized various fields, including industry, healthcare, gaming, and education, by seamlessly integrating virtual content with the real world. This project aims to develop a versatile AR framework using Python and OpenCV, enabling the incorporation of graphical elements compatible with diverse markers. At the core of the framework is a user-provided marker configuration file that defines marker types, dimensions, placement, and orientation of graphical elements, offering flexibility and adaptability for different applications.

The framework leverages OpenCV's ArUco model to detect and process a wide range of markers, including custom markers and QR codes, ensuring compatibility with various real-world scenarios. By providing interactive AR experiences, the system allows users to dynamically configure visual elements and retrieve detailed marker data, including marker distance and metadata. This approach creates a robust, scalable, and user-friendly AR system that bridges the gap between virtual and physical environments.

I. INTRODUCTION

Augmented Reality (AR) merges the physical and digital worlds, creating the illusion of digital content seamlessly integrated into the real environment. Using computer technology, AR overlays digital elements onto live camera feeds, allowing users to interact with data in real time. This technology relies on markers such as logos or QR codes to position and trigger digital content within the user’s view. Marker-Based AR specifically utilizes physical markers as activation points for superimposing digital content, with the camera detecting and interpreting the markers to display augmented elements.

One powerful tool for developing AR experiences is OpenCV, an open-source library for computer vision and image processing. OpenCV was designed to accelerate the application of machine perception in commercial products and provide a standard infrastructure for computer vision tasks. It supports processing images and videos for various applications, including face and object recognition, and can integrate with Python and libraries like NumPy for efficient analysis. By leveraging vector spaces and mathematical operations, OpenCV identifies visual patterns and features, forming the foundation for AR implementations.

OpenCV for Augmented Reality

To create a marker-based AR experience, OpenCV employs **ArUco markers**. These fiducial square markers are used to estimate camera position and serve as a cornerstone of AR systems.

ArUco Marker in OpenCV

ArUco markers are detected in video frames using the computer's camera. OpenCV identifies square shapes that might function as markers by analyzing each frame. Once identified, the system verifies the inner codification of these squares to confirm them as ArUco markers.
OpenCV provides the cv2.aruco.detectMarkers() method to handle this process. Key parameters required by this method include:

- **The grayscale image**: For detecting marker edges and shapes.
- **The marker dictionary**: Specifies the type of markers being used.
- **Detector parameters**: Configurable settings, though default values often suffice for many applications.

II. REVIEW OF LITERATURE

1. **D.G. Lowe (1999)** In *Object recognition from local scale invariant features*, Lowe introduces an innovative object recognition system leveraging local image features that are invariant to scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. The system is inspired by neural processes in the primate inferior temporal cortex. Using a staged filtering approach to detect stable points in scale space, it creates image keys representing blurred gradients across multiple scales and orientations. The system uses these keys for nearest neighbor indexing to identify object matches, achieving robust recognition even in cluttered, occluded images with computation times under two seconds. This seminal work laid the foundation for modern feature-based object recognition techniques.
2. **Takumi (Year Unknown)** In the study presented at the *IEEE Conference on Computer Vision and Pattern Recognition*, Takumi proposes a novel approach for handling occlusions in augmented reality without requiring 3D reconstruction. This contour-based method labels contour points as "in front of" or "behind" virtual objects, tracked across frames. A proximity graph groups contours of the same occluding object, and active contours refine the occlusion mask. The methodology is efficient, relying only on contour tracking rather than full scene reconstruction, making it valuable for real-time augmented reality applications.
3. **A.P. Dempster, N.M. Laird, and D.B. Rubin (1997)** The paper *Maximum likelihood from incomplete data via the EM algorithm* outlines the Expectation-Maximization (EM) algorithm, a general-purpose method for computing maximum likelihood estimates with incomplete data. The authors demonstrate the algorithm's monotonic convergence properties and illustrate its utility across a range of applications, including missing value handling, censored data, and mixture model estimation. This work is a cornerstone in statistical modeling and machine learning, offering a robust framework for problems involving incomplete datasets.
4. **Kadiri, P et. al (2024)** In their work Morphed Picture Recognition using Machine Learning Algorithms, the authors propose a system combining OpenCV and scikit-learn to address the growing challenge of identifying altered images. The system focuses on detecting morphed content by analyzing subtle distortions and changes in images. OpenCV is employed for feature extraction and pre-processing, while scikit-learn is utilized to develop a machine learning model trained on a dataset of both real and altered images. The study highlights the importance of robust training datasets in improving the system's accuracy and resilience. By advancing the field of image forensics, this system offers a dependable method for detecting image alterations and can be integrated into existing verification pipelines to enhance the security of digital content.
5. **W. Förstner and E. Gülch (1987)** In their study, *A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features*, Förstner and Gülch address the shortcomings of gray-feature-based image registration algorithms like SIFT, particularly in multisource images with significant grayscale differences. They propose a geometry-based corner detection algorithm that bypasses the limitations of gray features, improving matching speed and registration accuracy. This method is especially suited for registering multisource images with varied wavebands or significant differences in brightness.
6. **Prakash C et. al (2024)** In their article published in the ISIR Journal of Arts, Humanities and Social Sciences (ISIRJAHSS), the authors delve into the intricate relationship between artificial intelligence (AI) and human creativity. They examine the transformative effects of advanced AI systems on innovation and human ingenuity, offering insights into both the opportunities and risks posed by this integration. Through the analysis of case studies and emerging trends, the study provides a comprehensive understanding of how the creative process evolves in the AI-driven era. This research also explores philosophical implications, emphasizing the potential for both collaboration and conflict between AI and human creativity.
7. **S.M. Smith and J.M. Brady (1995)** the paper *SUSAN – A new approach to low-level image processing* presents a corner-based image alignment algorithm. The method involves a training phase using corner detection to build pyramid images and a matching phase comparing gradient vectors of template and inspection images. Employing a parabolic function for evaluating geometric relationships, the approach achieves higher efficiency and robustness against nonlinear lighting variations than edge-based template matching. The SUSAN algorithm offers precision and accuracy in challenging image alignment tasks, reinforcing its utility in low-level image processing.

III. EXISTING SYSTEM

Together, these methods enable CNNs to handle a broad range of computer vision problems, including object identification, segmentation, image classification, and more. The CNN's architecture and design are shaped for best performance and comprehension of visual data by the combination and selection of various strategies, which are particular to the situation at hand. To solve particular issues and enhance training stability, CNNs frequently use variations of normalizing techniques, such as Layer normalizing and Group Normalization. Another crucial choice that affects training speed and model performance is the optimizer, such as Adam, RMSprop, or stochastic gradient descent (SGD) with momentum.

Disadvantages

➢CNNs need a lot of labeled training data, but they need less data.

➢ Computational Intensity: CNNs require a large amount of processing power to train and operate.

➢ Overfitting: When working with limited datasets, CNNs are prone to overfitting, which reduces their ability to generalize to new data.

➢ Interpretability: Because CNNs are sometimes seen as "black-box" models, it can be difficult to comprehend and analyze how they make decisions.

IV. PROPOSED SYSTEM

A fiducial square marker for estimating camera postures is the ArUco marker. If an ArUco marker is found in the video, you can overlay the identified markers with an image to enhance the digital content. An ArUco marker is a synthetic square marker with a unique identifier and an internal binary matrix surrounded by a broad black border.

In the ArUco marking, zero is represented by white, and one by black.The internal binary matrix's size is determined by the marker size. In the ArUco marker, data bits are represented by even squares, and parity bits are represented by odd blocks. The binary matrix enables the image's identification, while the black border speeds up detection within the image. The ArUco marking aids with the camera's angle comprehension.

Advantages

- ArUco markers are used to estimate the pose (position and orientation) of objects in the camera view.
- They are highly accurate and provide high-precision pose estimation.
- Versatile Uses: ArUco markers are used in computer vision, augmented reality, virtual reality, and robotics.
- Broad Support: It is simpler to include ArUco markers into software because they are supported by a number of libraries and frameworks.
- Customization: By altering the ArUco marks' size, type, and content, users can personalize them.

V. SYSTEM ARCHITECTURE

The architecture that is being shown is an example of a system that integrates computer vision and machine learning procedures for marker-based augmented reality (AR) applications. The procedure starts with a camera-captured live video stream, which is preprocessed to get the input data ready for additional analysis. The next step is marker detection, which finds possible AR markers in the video frame. Following detection, the system moves on to marker identification, where particular characteristics of the found marker are examined in order to identify it. This is followed by identifying the marker's pose, including its position and orientation in the 3D space.

In orderto improve accuracy, background removal is carried out concurrently, which separates the marker from its surroundings. A Convolutional Neural Network (CNN) model, trained with a classifier and a synthetic marker dataset, receives the processed data. By using the artificial dataset produced during the training stage, the CNN classifier guarantees strong marker recognition. This process's end result makes it possible to render virtual items precisely superimposed on the marker in the live video feed, creating smooth augmented reality experiences. In order to preserve real-time engagement and immersion, marker tracking makes sure the system continuously tracks the marker as it moves.

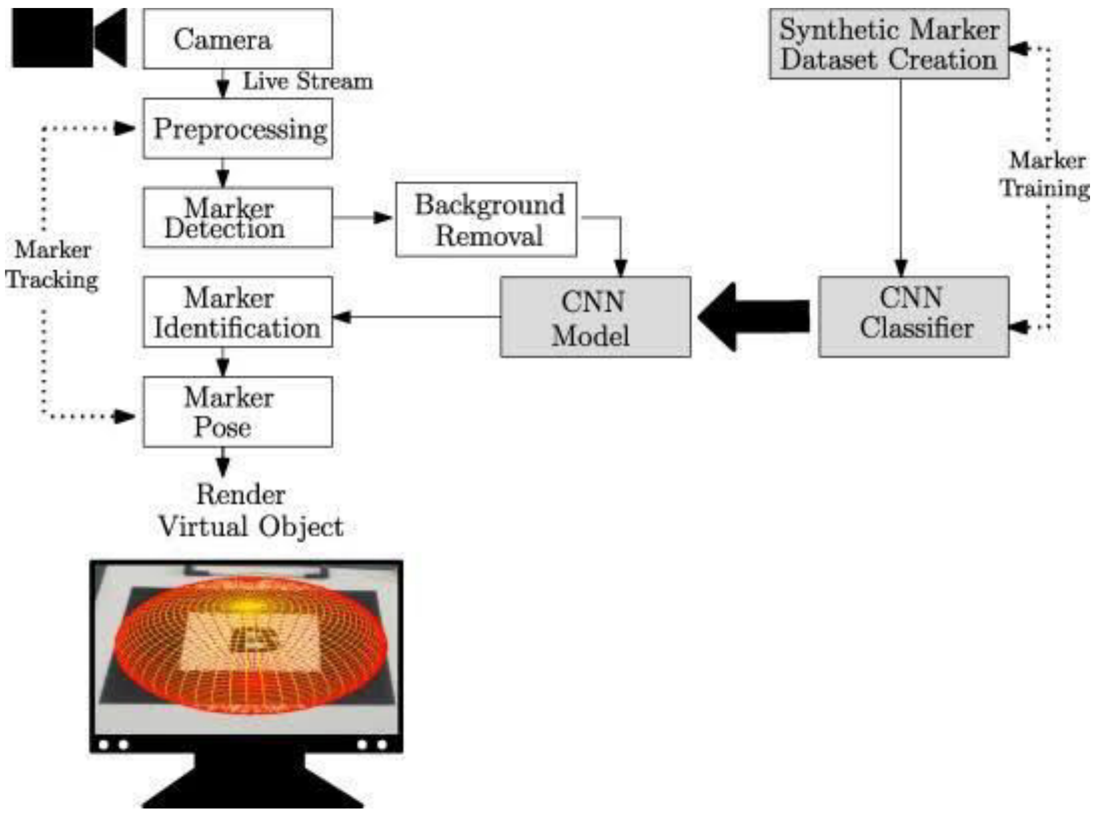

Figure A:- SYSTEM ARCHITECTURE

VI. METHODOLOGY

A thorough procedure for an augmented reality system based on markers is shown in the diagram. The first step is to use a camera as the input source to record a live video stream. To improve the accuracy of marker detection, the collected stream is preprocessed, which refines the visual data. The system uses image processing algorithms to find possible markers in the video frames during the marker detection phase.

The system then carries out marker identification, identifying the particular marker by utilizing its distinct characteristics. The position and orientation of the marker in the 3D space are simultaneously determined using marker pose estimation. Together, these procedures enable the system to precisely align and superimpose virtual items on the identified marker.

A CNN (Convolutional Neural Network) model is essential for improving the process of identification and detection. The CNN classifier validates the model once it has been trained on synthetic marker datasets to guarantee precise detection even under different circumstances. To increase the system's resilience, a variety of training data is produced during the synthetic dataset production process. To create the augmented reality experience, the system renders virtual items after determining the marker's posture. These things are then superimposed on the live video feed. When moving, the virtual object stays in line with the marker thanks to the marker tracking module. The user's screen displays a smooth augmented reality experience as a result of this real-time interaction.

VII. MODULES

Camera Module:
The camera first records a live feed or an image to start the procedure. The raw visual data that will be further processed comes from the camera, which acts as the input device.
Photo Taking Module:
This module is in charge of picking particular frames for analysis or collecting screenshots from the live camera feed. It guarantees that the photos are prepared for further processing.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 6, November-December 2024 ||

DOI:10.15680/IJARETY.2024.1106019

Image Processing Module:
In order to improve their quality, reduce noise, and get them ready for marker detection, the acquired images are processed. This covers activities such as segmentation, filtering, and edge detection.
Markers Tracking:
This module tracks the markers' movements and changes their positions in real time after they have been detected. It guarantees the augmented reality (AR) experience's consistency and continuity.
Service Provide Module:
The control center is this module. Through alignment with the position and orientation of the tracked marker, it incorporates the virtual object (AR content) into the live camera stream. Based on the identified indicators, the module chooses which virtual goods or services to show.
Displaying on Screen:
The screen shows the processed output along with the integrated virtual objects. This stage guarantees that the AR material will show up as planned.
AR Display:
In this last stage, the system smoothly blends virtual objects with the screen's depiction of the real world to provide the user an augmented reality experience.

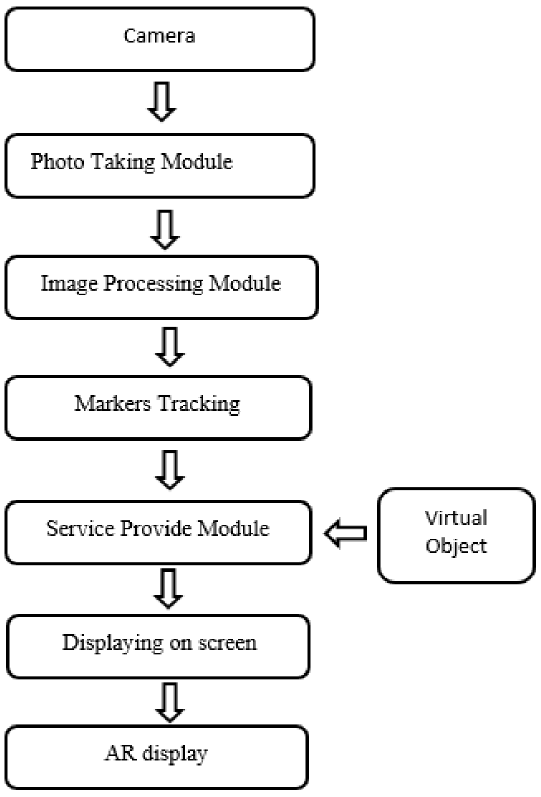

Figure B:- MODULE DIAGRAM

VIII. IMPLEMENTATION

Algorithm:-
1. Set up the parameters and ArUco marker dictionary:
Make use of a predetermined dictionary, such as DICT_4X4_250. Establish the marker's size (e.g., 8 cm or 400 pixels).
2. Create markers:
Create and save ArUco marker pictures for marker IDs 0 through 9.
3. Configure the camera and calibration information:
Enter the distortion coefficients, camera matrix, and additional calibration information. Start the video recording.
4. The primary marker detection loop:

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 6, November-December 2024 ||

DOI:10.15680/IJARETY.2024.1106019

Take pictures with the webcam. Make the frame grayscale.
5. Look for ArUco marks in every picture:
To find marker corners and IDs, use aruco.detectMarkers().
6. If markers are found, estimate their pose:
The rotation and translation vectors can be computed using aruco.estimatePoseSingleMarkers().
7. Mark the frame with markers:
When markers are found, show their IDs and draw marker bounding boxes.
Determine the marker's distance from the camera and show it.
8. Present and handle enhanced images:
Use homography to overlay augmented images from a specified list for markers with particular IDs.
9. Loop of exit:
Pressing the 'q' key will end the loop.
10. Make resources available:
Close the OpenCV windows and release the camera.

IX. EXPERIMENTAL RESULTS

Generate Markers

This uses the aruco module in OpenCV to create and save distinct ArUco markers. It starts by setting the marker size to 400 pixels and defining a predetermined vocabulary of 4x4 markers. After that, the code generates each marker by iterating through the marker IDs and stores them as PNG files in the markers directory.

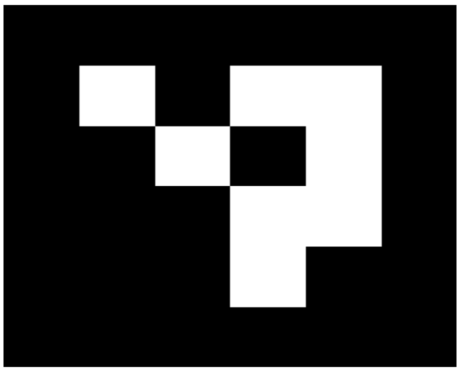

Figure:- GERATED MARKER

Marker Detection

Real-time ArUco marker detection from a webcam feed is accomplished using OpenCV. It first establishes detection parameters and imports apredetermined dictionary of 4x4 markers (DICT_4X4_250). Aruco.detectMarkers() is used to detect ArUco markers after the video frame has been collected and converted to grayscale in a loop. The marker ID appears in the upper-right corner, and the corners of any markers that are located are drawn on the frame.

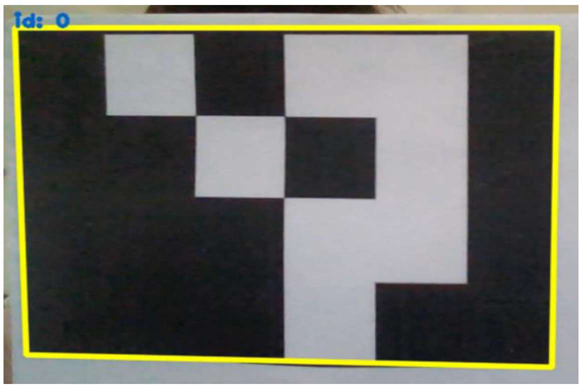

Figure:- MARKER DETECTION

Image Augmentation

This code performs image augmentation on a webcam feed by overlaying different images onto detected ArUco markers. It first loads a list of images and detects markers in real-time from the video feed. When a marker is detected, the image augmentation function is used to warp and overlay the corresponding image onto the marker's position on the frame using homograph.

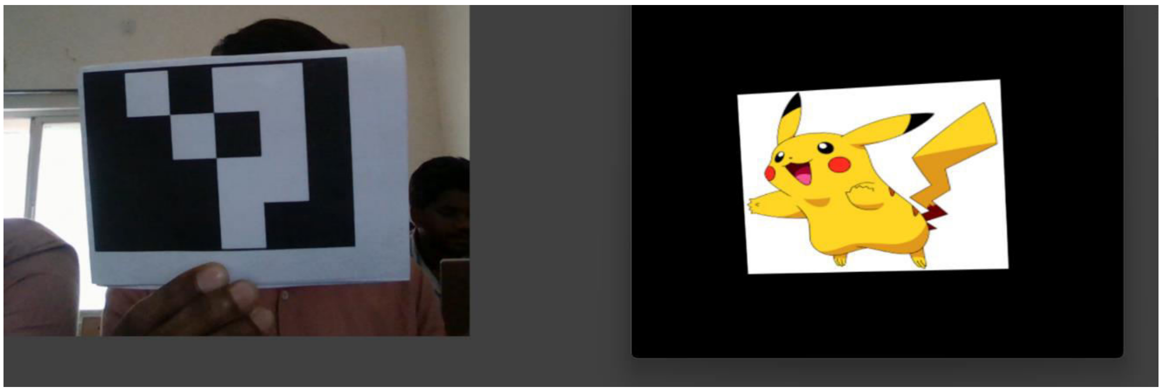

Figure:- IMAGE AUGMENTATION

Distance Estimation

It uses a webcam feed to identify ArUco markers in real time, determines their pose, and shows extra data like the marker's ID and distance from the camera. First, it loads camera calibration information from a file, including the distortion coefficients and intrinsic matrix. The code determines the distance from the camera, computes the rotation and translation vectors for each detected marker using aruco estimate Pose Single Markers(), and then displays the marker's pose (axes) on the video frame.

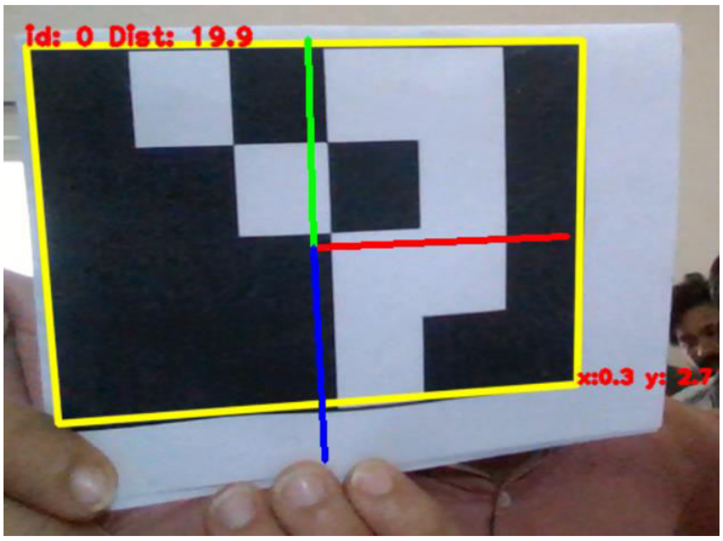

Figure:- DISTANCE ESTIMATION

X. CONCLUSION

An inventive and exciting advancement in the field of augmented reality is denoted by the designation "MARK-GP (Marker-Based Augmented Reality Framework with Graphical Placement)". With its comprehensive capabilities for graphical placement within the AR world and marker-based technologies for accurate item anchoring, this framework probably marks a significant leap in the industry. A project like this indicates a strong emphasis on providing a smooth and engaging user experience. From marketing and industrial applications to gaming and education, the framework's possible applications cover a wide range of fields and have the potential to have a big impact on the AR market.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 6, November-December 2024 ||

DOI:10.15680/IJARETY.2024.1106019

XI. FUTURE ENHANCEMENT

By integrating state-of-the-art technology and creating new opportunities for AR research and development, MARK-GP suggests interdisciplinary collaboration. With the goal of making AR technology more approachable and broadly available, this work may solve usability and accessibility issues. All things considered, MARK-GP seems ready to make a significant impact on the augmented reality scene and provide an intriguing look at what lies ahead for AR innovation.

REFERENCES

1. R. T. Azuma,A survey of augmented reality, Presence 6 (1997) 355–385.
2. H. Kato, M. Billinghurst, Marker tracking and HMDcalibration for a Video-Based augmented reality conferencing system, Augmented Reality, InternationalWorkshop on 0 (1999) 85–94.
3. V. Lepetit, P. Fua, Monocular model-based 3d tracking of rigid objects: A survey, in: Foundations and Trends in Computer Graphics and Vision, 2005.
4. B. Williams, M. Cummins, J. Neira,P.Newman,I. Reid, J. Tard´os, A comparison of loop closing techniques in monocular slam, Robotics and Autonomous Systems.
5. W. Daniel, R. Gerhard,M.Alessandro,T. Drummond, S. Dieter, Real-time detection and tracking foraugmented reality on mobile phones, IEEE Transactions on Visualization and Computer Graphics (2010) 355–368.
6. G. Klein,D. Murray, Parallel tracking and mapping for small are workspaces, in:Proceedings of the2007 6th IEEE and ACM International Symposium12on Mixed and Augmented Reality, ISMAR '07, IEEE Computer Society, Washington, DC, USA, 2007, pp.1–10.
7. K. Mikolajczyk, C. Schmid, Indexing based on scale invariant interest points., in:ICCV, 2001, pp. 525–531.
8. D. G. Lowe, Object recognition from local scale invariant features, in:Proceedings of the International Conference on Computer Vision-Volume 2Volume 2, ICCV '99, IEEE Computer Society, Washington, DC, USA, 1999, pp. 1150.
9. P. Bhattacharya, M. Gavrilova, A survey of landmark recognition using the bag-of-words framework, in: Intelligent Computer Graphics, Vol. 441 of Studies Computational Intelligence, Springer Berlin Heidelberg, 2013, pp. 243–263.
10. H. Kato, M. Billinghurst, Marker tracking and hmdcalibration for a video-based augmented reality conferencing system, in: Proceedings of the 2nd IEEE and ACM International Workshopon Augmented Reality, IWAR '99,IEEE Computer Society, Washington, DC, USA, 1999, pp.85.
11. M. Fiala, Designing highly reliable fiducial markers,IEEE Trans. Pattern Anal. Mach. Intel. 32 (7) (2010)1317–1324.
12. D. Schmalstieg, A. Fuhrmann, G. Hesina,Z.Szalav´ari,L. M. Encarna¸c¨ao,M. Gervautz,W.
13. Purgathofer, The studierstube augmented reality project, Presence:Teleoper. Virtual Environ. 11 (1)(2002) 33–54.
14. Dorfmller,H.Wirth,Real-timehandandheadtrackingforvirtualenvironmentsusing infrared beacons :in Proceedings CAPTECH98. 1998, Springer, 1998, pp. 113–127.
15. Prakash, C., Reddy, K. S., Anusha, P., & Bethapudi, A. (2024). ISIR Journal of Arts, Humanities and Social Sciences (ISIRJAHSS).
16. M. Ribo, A. Pinz, A. L. Fuhrmann, A new optical tracking system for virtual and augmented reality applications, in: In Proceedings of the IEEE Instrumentation and Measurement Technical Conference, 2001,pp. 1932–1936.
17. V. A. Knyaz, R. V. Sibiryakov, The development of new coded targets for automated point identification and non-contact surface measurements, in:3D Sur-face Measurements, International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, part5, 1998, pp. 80–85.
18. L. Naimark, E. Foxlin, Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker, in: Proceedings of the 1st
19. International Symposium on Mixed and Augmented Reality, ISMAR '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 27.
20. Kadiri, P., Anusha, P., Prabhu, M., Asuncion, R., Pavan, V. S., & Suman, J. V. (2024, July). Morphed Picture Recognition using Machine Learning Algorithms. In 2024 Second International Conference on Advances in Information Technology (ICAIT) (Vol. 1, pp. 1-6). IEEE.
21. J.Rekimoto,Y.Ayatsuka,Cybercode:designingaugmented reality environments with visual tags, in Proceedings of DARE 2000 on Designing augmented reality environments, DARE '00, ACM, New York, NY, USA, 2000, pp. 1–10.
22. M. Rohs, B. Gfeller, Using camera-equipped mobile phones for interacting with real-world objects, in Advances in Pervasive Computing, 2004, pp. 265–271.

IJARETY

INNO SPACE
SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 Impact Factor: 7.394

www.ijarety.in editor.ijarety@gmail.com