**Augmented Reality**

**Using Python**

***Project Submitted in Partial fulfillment of the***

***Requirement for the Award of the Degree of***

Master of Computer Application

***Semester VI***

## Session Jan-May, 2023

**Under the guidance of Submitted By**

**Name of Internal Guide: Name & Roll No.:**

### Dr. Rupesh Sendre Ayush Jajoo

IC-2K20-16

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**2023**

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**DECLARATION**

I hereby declare that the project entitled “***Augmented Reality using Python***” submitted by me for the partial fulfillment of the requirement for the award of Master Computer Application Technology (5 Years) Semester VI to International Institute of Professional Studies, Devi Ahilya Vishwavidyalaya, Indore, comprises my own work and due acknowledgement has been made in text to all other material used.

Signature of Student:

Date:

Place: Indore

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**CERTIFICATE FROM GUIDE**

It is to certify that dissertation on “Augmented Reality using Python”, submitted by Mr. Ayush Jajoo to the International Institute of Professional Studies, DAVV, Indore has been completed under my supervision and the work is carried out and presented in a manner required for its acceptance in partial fulfillment for the award of the degree of “Master of Computer Application (5 Years) Semester VI”.

**Project Guide** Signature:

**Name:**

**Date:**

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**CERTIFICATE**

It is to certify that we have examined the dissertation on “***Augmented Reality using Python***”, submitted by Mr. Ayush Jajoo to the International Institute of Professional Studies, DAVV, Indore and hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance in partial fulfillment for the award of the degree of “er of Computer Application (5 Years) Semester VI”.

**Internal Examiner External Examiner**

Signature**:** Signature**:**

Name**:** Name**:**

Date**:** Date**:**

**ACKNOWLEDGEMENT**

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I acknowledge my sincere thanks to those who have contributed significantly to this project.  It is a pleasure to extend my deepest gratitude to **Dr. Rupesh Sendre**, for her valuable guidance and constant supervision as well as providing necessary information regarding the project and also for her support in completing the project.

A special mention needs to be made for Dr. A. K. Sapre, Director IIPS, DAVV, former

director and Mr. Jugedre Dongre, Program In-charge MCA course who helped very

willingly as often as we wanted and cooperated with interest.

Our classmates also deserve a mention for their constant review, praise and critical

appraisal.

Before ending we would like to thank our parents, and friends who have always stood

by us whenever we were in need of them. We sincerely regret any inadvertent

omissions.With our heartiest thanks to all.

A special mention needs to be made for Dr. A. K. Sapre, Director IIPS, DAVV, former

director and Mr. Jugedre Dongre, Program In-charge MCA course who helped very

willingly as often as we wanted and cooperated with interest.

Our classmates also deserve a mention for their constant review, praise and critical

appraisal.

Before ending we would like to thank our parents, and friends who have always stood

by us whenever we were in need of them. We sincerely regret any inadvertent

omissions.With our heartiest thanks to all.

A special mention needs to be made for **Dr. B. K. Tripathi, Director IIPS, DAVV, Mr. Jogendra Dongre, Program In-charge MCA course, Dr. Shraddha Soni, Batch Mentor, MCA VI Semester** and all the faculty members of IIPS, DAVV who helped very willingly as often as I wanted and cooperated with interest.

I thank and acknowledge each and everyone's efforts that helped me in some or the other way for small and significant things.

Before ending I would like to thank my parents, and friends who have always stood by us whenever I was in need of them. I sincerely regret any inadvertent omissions.

With my heartiest thanks to all.

**Name and Roll No.**

**Ayush Jajoo**

**IC-2K20-16**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Serial Number | Topic | Page Number |
| 1 | Abstract | V |
| 2 | Introduction | 7 |
| 3 | Analysis   1. Aim 2. Objective 3. Benefits 4. Applications | 8 |
| 4 | Project Planning   1. Gantt Chat | 10 |
| 5 | System Design   1. Logical Design 2. Implementation | 11 |
| 6 | Software development methodology | 23 |
| 7 | System testing | 24 |
| 8 | Limitations | 25 |
| 9 | Conclusion | 26 |
| 10 | Bibliography | 27 |

**Abstract**

Augmented reality (AR) is a technology that allows virtual objects to be overlaid onto the real world in real-time. OpenCV is a popular computer vision library that can be used to develop AR applications using Python programming language.

In this project, we will focus on using OpenCV with its ORB (Oriented FAST and Rotated BRIEF) feature detector to create an AR application. ORB is a feature detection and description algorithm that can be used to find key points in an image and match them with key points in another image.

To start, we capture a live video stream from the camera and apply ORB to detect the key points in the live video stream. We then match these key points with the key points from a target image, which has been previously defined. By matching the key points, we can determine the orientation, scale, and position of the target image in the live video stream.

Once we have determined the position of the target image, we can overlay virtual objects on top of the target image. These virtual objects can be anything from 3D models to simple 2D images. We can use the position and orientation of the target image to position the virtual object in the correct location relative to the target image.

Overall, this project demonstrates the power of computer vision and OpenCV in creating AR applications. By using ORB to detect and match key points, we can create an AR application that accurately overlays virtual objects onto the real world.

**Introduction**

Augmented reality (AR) is a technology that has gained a lot of attention in recent years for its potential to enhance user experiences in various fields, including entertainment, education, advertising, and more. AR allows digital information to be overlaid onto the real world, creating an immersive and interactive experience for users.

Python, a popular programming language, has several libraries that can be used for developing AR applications, including OpenCV (Open-Source Computer Vision Library). OpenCV is a widely used open-source computer vision and machine learning software library that can be used for various applications, including AR.

OpenCV provides a range of tools and functions that allow developers to analyze, manipulate, and understand visual data from the real world. By combining OpenCV with other libraries and technologies, such as machine learning and image processing, developers can create AR applications that accurately detect and track real-world objects, and overlay virtual content on them in real-time.

AR applications created using OpenCV and Python can have a wide range of use cases, including gaming, education, product visualization, and more. These applications can provide users with an interactive and engaging experience that blurs the lines between the digital and physical worlds. Overall, the combination of AR, Python, and OpenCV provides an exciting platform for developing innovative and immersive applications.

**Analysis**

**Aim:**

The problem that AR applications developed using Python, OpenCV, and ORB feature detector aim to solve is the ability to overlay digital content onto real-world objects accurately and efficiently while providing a natural and engaging user experience.

**Objectives:**

In AR development using OpenCV, the first step is to capture the real-world environment using a camera. The camera feed is then processed to detect and track specific objects or markers. Once the objects or markers are detected, the digital information can be overlayed onto the real-world environment to create the AR experience.

**Basic Analysis:**

OpenCV provides a wide range of computer vision techniques and algorithms, such as object detection, image processing, and feature extraction, which can be used to create AR applications. These techniques can be used to track and augment objects, recognize hand gestures, and create AR interfaces, among others.

Step-1: Capturing the real-world environment using a web cam or camera.

Step-2: Detecting object for AR to display on.

Step-3: Tracking specific object or marker for overlaying.

Step-4: Information is overlayed(displayed) onto the real-world environment.

**Benefits:**

1. Enhanced user experience: AR applications provide a more immersive and engaging experience for users by overlaying digital content onto real-world objects.
2. Improved visualization: AR applications can be used to visualize complex data or information in a more intuitive and interactive way. For example, an AR application can be used to visualize medical data, allowing doctors to view a patient's anatomy in 3D and make more informed decisions.
3. Increased efficiency: AR applications can be used to improve efficiency and productivity by providing real-time information and feedback. For example, an AR application can be used in manufacturing to provide workers with real-time instructions and feedback, reducing errors and improving productivity.
4. Scalability and Flexibility: AR applications developed using Python and OpenCV are highly scalable and can be easily adapted to different devices and platforms.

**Applications:**

1. Education: AR applications can be used in education to enhance the learning experience by providing interactive and immersive content
2. Entertainment: AR applications can be used in entertainment to create unique and memorable experiences for users.
3. Marketing and Advertising: AR applications can be used in marketing and advertising to create interactive and engaging campaigns that capture the user's attention.
4. Healthcare: AR applications can be used in healthcare to visualize medical data and assist in surgical planning.
5. Manufacturing and Maintenance: AR applications can be used in manufacturing and maintenance to provide workers with real-time information and feedback.
6. Tourism and Travel: AR applications can be used in tourism and travel to provide users with interactive and immersive experiences that enhance their travel experience.

**Project Planning**

**Gantt Chart:**

|  |  |  |  |
| --- | --- | --- | --- |
| PROJECT NAME | PROJECT By | PROJECT START DATE | PROJECT END DATE |
| Augmented Reality using Python | Ayush Jajoo | 20-02-23 | 01-05-23 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **PHASE ONE** | | | **PHASE TWO** | | | **PHASE THREE** | | |
| **TASK ID** | **TASK** | **START DATE** | **END DATE** | **WEEK**  **1** | **WEEK**  **2** | **WEEK**  **3** | **WEEK**  **4** | **WEEK**  **5** | **WEEK**  **6** | **WEEK**  **7** | **WEEK**  **8** | **WEEK**  **9** |
| 1 | Project Initiation | 20-02-23 | 05-03-23 |  |  |  |  |  |  |  |  |  |
| 1.1 | Research |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 | Project Concept |  |  |  |  |  |  |  |  |  |  |  |
| 1.3 | Guidelines |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Project Definition and Planning | 06-03-23 | 12-03-23 |  |  |  |  |  |  |  |  |  |
| 2.1 | Scope and Goal Setting |  |  |  |  |  |  |  |  |  |  |  |
| 2.2 | Risk Management |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Project Execution | 20-03-23 | 16-04-23 |  |  |  |  |  |  |  |  |  |
| 3.1 | Programming |  |  |  |  |  |  |  |  |  |  |  |
| 3.2 | Monitoring |  |  |  |  |  |  |  |  |  |  |  |
| 3.3 | Status and Tracking |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Project Performance / Monitoring | 17-04-23 | 01-05-23 |  |  |  |  |  |  |  |  |  |
| 4.1 | Project Objectives |  |  |  |  |  |  |  |  |  |  |  |
| 4.2 | Debugging |  |  |  |  |  |  |  |  |  |  |  |
| 4.3 | Project Performance |  |  |  |  |  |  |  |  |  |  |  |

**Red: Just Initiated Yellow: InProgress**

**Green: Completion**

**System Design**

**Logical Design:**

The logical design for an augmented reality (AR) system using Python and OpenCV with ORB (Oriented FAST and Rotated BRIEF) feature detection and matching can be broken down into the following components:

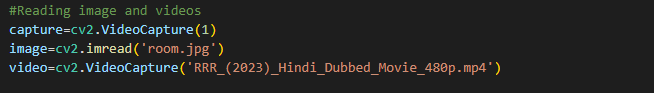
1. Input: The input component is responsible for capturing video input from a **camera** or other input device. This involve setting up a video capture device and configuring it to capture frames of video.
2. Feature detection and matching: The feature detection and matching component is responsible for identifying key features in the video input and matching them to reference images or models. This involve using the **ORB algorithm and BF Matching** to detect and match features, as well as other image processing techniques such as filtering and thresholding.
3. Tracking: The tracking component is responsible for tracking the position and orientation of the camera in real-time. This involve using **Matrix points** as well as computer vision techniques such as optical flow or feature tracking.
4. 2D model generation: The 2D model generation component is responsible for generating a 2D model of the object or scene being tracked. This involve using a **OpenCV Homograph** tool to create a 3D model, or using photogrammetry techniques to generate a 3D model from a series of images.
5. Augmentation: The augmentation component is responsible for overlaying augmented content on top of the video input. This involve rendering 2D models, by **OpenCV Polygraph and bitwise operator** displaying video content.
6. Output: The output component is responsible for displaying the final augmented reality experience to the user. This involve rendering the augmented content through **New Masking overlay** to a display device such as a screen or a headset, or transmitting the augmented content to a remote location for collaboration or remote viewing.

**Implementation:**

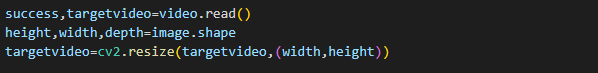
1.Importing libraries which are going to be used.



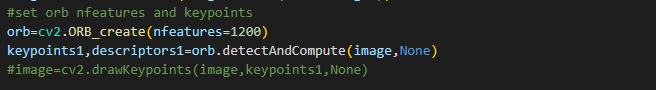
2.Reading image and web cam capture.



3.Reading video and its height, width, depth and resizing it



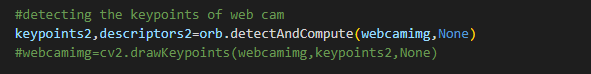
4.Create and set the ORB nfeatures and key points and descriptors for image



5.Initialising the while loop and reading the webcam image in loop and while will be running until escape key is not press, all code will run in while loop.



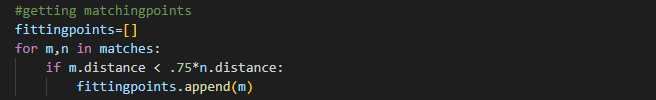
6. Detecting the key points for Webcam image.



7. Creating and Initialing the Brute Force Matcher and matching both image’s descriptors.



8. For Brute Force matcher getting the matching points in a array using for loop with the condition i.e. distance for the points should be between 0.75\*n unit.



9. Drawing the matches collected for the matcher.



10. Starting a if condition for if the matching points are greater than 18 then only the video will augment.



11.Creating the source and destination points in 3D model of 3D Matrix and reshaping it using the library named NumPy.



12. Creating a new mask of homograph using the matrix points with the width of 5px



13. Reshaping the points of homograph using the size of the image with NumPy library.



14.Creating perspectiveTransform for destination points using points and matrix then using polyline tracking the webcam image in real time with the width of 2px.



15. Warping the image to the web cam image using OpenCV’s warpPerstective function and determining the length and width to wrap.



16. Creating a new raw mask to overlay the augmented image



17. Using fillpoly to fill the remain area with black colour using umPy library.



18.Using the bitwise operator to invert the new mask black colour.



19. Again Using the bitwise operator to augment the inverted image.



20.Again bitwise operator to wrap the image and augmented image together.



21. OpenCV imshow function using to show or display the augmentation on the screen.



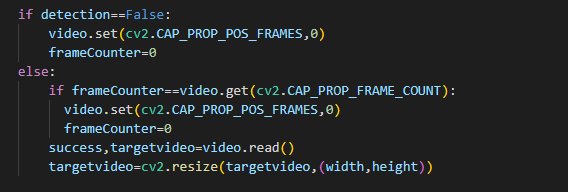
22. Waitkey is used for Ture condition of while loop and if condition for escape key to exit the application.



23. the framecounter and detection are decleared.

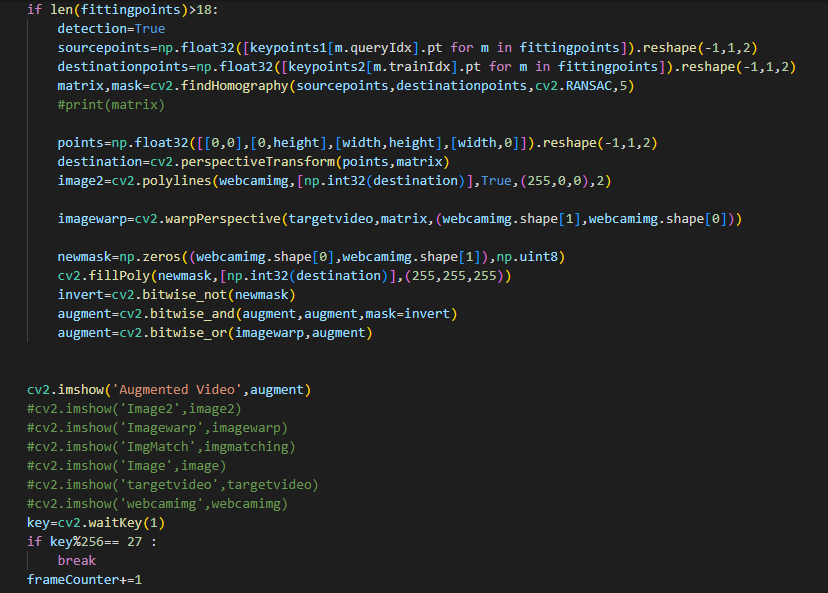


24.This if condition to display the video in video format if video id finished to restart the video.



**Full Code:**

****

****

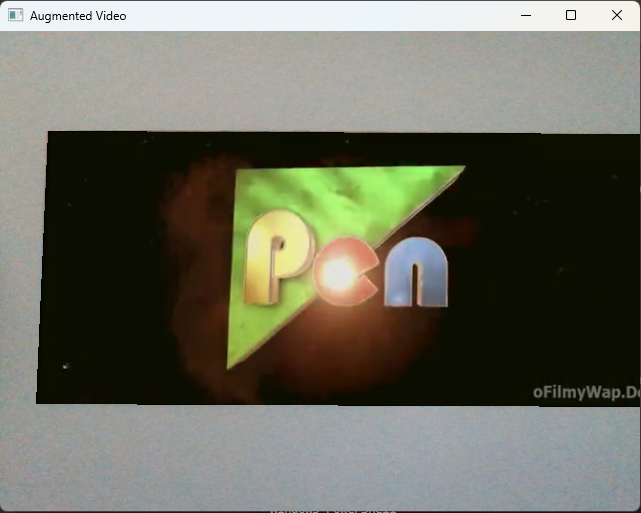
**Execution**

**Image:**



**Output:**

****

****

**Software Development Methodology**

When developing an augmented reality application using Python, OpenCV, NumPy, and the ORB feature detector, a typical methodology that can be followed for developing such an application is the following:

1. Requirements Gathering: In this phase, the requirements of the application are identified by interacting with the problems. This involves understanding the target audience, defining the scope of the application, and identifying the required features and functionality.
2. Design: In this phase, the overall design of the application is created. This involves identifying the different components of the application, the algorithms for detecting and tracking real-world objects, and the virtual content to be overlaid. The design should also take into consideration the performance requirements of the application, such as the frame rate, accuracy.
3. Development: In this phase, the actual code for the application is written. The code should be modular and well-structured, with proper documentation and comments. The development phase involves integrating the different components and libraries, such as OpenCV and NumPy, and testing the application for correctness and performance.
4. Testing: In this phase, the application is tested to ensure that it meets the requirements and is free of bugs and errors. Testing should be done at different stages of development, including unit testing, integration testing, and system testing.
5. Deployment: In this phase, the application is deployed to the target platform, such as a desktop computer or a mobile device. This involves packaging the application, creating installers, and providing documentation and support for users.

**System Testing**

Testing an augmented reality (AR) application developed using Python, OpenCV, NumPy, and ORB feature detector is crucial to ensure that the application functions correctly and meets the desired performance and accuracy requirements. The following is a system testing approach that can be used for such an application:

1. Functional Testing: This type of testing focuses on testing the functionality of the AR application. It involves testing the different components of the application, such as the image processing algorithms, object detection and tracking, and virtual content overlay.

(By performing functional testing, I have debugged an error on accurate image colour detection)

1. Performance Testing: This type of testing focuses on testing the performance of the AR application. It involves testing the application's speed, accuracy, and responsiveness under different conditions, such as varying lighting conditions and camera angles. Performance testing should be performed to ensure that the application can handle the load and perform optimally without lag or crashes.
2. Compatibility Testing: This type of testing focuses on testing the compatibility of the AR application with different platforms and devices. It involves testing the application on different operating systems, browsers, and hardware configurations to ensure that it functions correctly and displays properly.

(By performing compatibility testing, I was encountered with the incompatibility of my internal webcam quality)

1. Regression Testing: This type of testing focuses on testing the application after changes have been made to ensure that no new bugs or errors have been introduced. Regression testing should be performed to ensure that the application remains stable and functional even after modifications.

**Limitations**

There are some limitations of using augmented reality (AR) with Python and OpenCV, specifically when using the ORB (Oriented FAST and Rotated BRIEF) algorithm for feature detection and matching. Some of the limitations are:

1. Limited detection and matching accuracy: The ORB algorithm have limited accuracy in detecting and matching features, which can result in inaccurate AR experiences. This can be particularly challenging in environments with complex or changing lighting conditions, or when objects are partially obscured.
2. Limited performance on large datasets: The ORB algorithm may not perform well on large datasets, which can result in slower performance and longer processing times. This can be particularly problematic when developing AR applications that require real-time performance, such as gaming or industrial applications.
3. Limited robustness to occlusion: The ORB algorithm may not be robust to occlusion, which occurs when one object is partially or completely hidden by another object. This can result in inaccuracies when detecting and matching features, which can impact the overall AR experience.
4. Limited support for 3D tracking: The ORB algorithm is primarily designed for 2D feature detection and matching, which can limit its ability to track objects in 3D space. This can be particularly challenging when developing AR applications that require precise 3D tracking, such as in medical or industrial applications.
5. Limited cross-platform compatibility: While Python and OpenCV are cross-platform compatible, some AR libraries or hardware may not be compatible with all platforms. This can limit the portability of AR applications and impact their accessibility to users.

**Conclusion**

In conclusion, augmented reality using Python and OpenCV with ORB (Oriented FAST and Rotated BRIEF) feature detection and matching provides a powerful platform for developing augmented reality applications. This technology has many potential applications in fields such as education, gaming, marketing, and industrial training.

Through the use of the ORB algorithm, the system is able to detect and match features in real-time, providing a seamless augmented reality experience for users. However, the system also has its limitations, such as limited accuracy and robustness to occlusion, which need to be taken into account when developing AR applications.

To develop an effective AR system, a rigorous system development methodology and testing plan should be followed, with careful consideration given to factors such as hardware and software requirements, input/output devices, and user interface design.

Despite these challenges, the potential benefits of augmented reality technology are immense, with the ability to provide immersive and interactive experiences that can enhance learning, engagement, and productivity. As such, augmented reality using Python and OpenCV with ORB feature detection and matching is a promising technology that has the potential to revolutionize many industries and applications.

**Bibliography**

* <https://pyimagesearch.com/>
* <https://bitesofcode.wordpress.com/>
* <https://medium.com/>
* <https://github.com/>