Enhancing User Interactions by Combining AR Markers with Various Kinds of Markers

SHENGYU FANG[†1] JIRO TANAKA [†1]

Abstract: This research uses various kinds of markers combining with AR markers to enhance user interactions. Our new AR system consists of three new functions, i.e., multiple markers recognition, hidden markers recognition, and auxiliary markers. As multiple markers recognition, we have integrated various kinds of markers in our new system. Our system can detect and locate their position concurrently. As hidden markers recognition, we have designed button areas on the hidden marker based on natural feature points. When the users cover the button area by their finger, the input events will be triggered. As auxiliary marker, user can get specific or decrypted information by putting auxiliary markers.

Keywords: Augmented reality, User Interfaces and Interactive Systems, Multiple markers

1. Introduction

Augmented Reality (AR) brings digital information and virtual objects into the physical space [1]. Nowadays, numerous augmented reality applications have been used in people's daily life of different kinds and disseminated all over the world. There are three types of augmented reality: marker-based, marker-less and location-based. The marker-based AR is the best-known type which uses cameras and visual markers.

Recently, with the development of marker-based AR system [3], a large number of visual markers appear. And just only using single AR marker sometimes cannot meet needs of users.

In this study, we have proposed the framework of new marker-based AR system which can combine multiple markers to make the traditional marker-based AR system more attractive and effective.

2. Background and related work

2.1 Augmented Realty

Augmented reality (AR) is an interactive experience, in which objects that of the real world are enhanced by computer-generated perceptual information [1].

AR can be defined as a system that fulfills three basic features.

1. A combination of real and virtual worlds.
2. Real-time interaction.
3. Accurate 3D registration of virtual and real objects [2].

AR aims to combine virtual world with real-world, so that virtual world belongs to the real-world. As a feature of the integration of virtual and real scenes, many applications of Augmented Reality are emerging, such as in the field of education, medical treatment, and entertainment [3].

In this research, we focus on the marker-based AR system and try to enhance it.

Marker-based Augmented Realty system uses a camera and a visual marker to estimate the orientation and its coordinate by computing the pose matrix [3]. Image recognition is an imperative component of augmented reality systems. By using visual markers there are already registered within the system, virtual contents can be generated on top of the real-world.

2.2 Visual Markers

The 3D virtual contents need to be registered in AR system for rendering. Visual markers are widely used in existing AR systems. For example, ARToolKit [4], CyliCon [5], ArLoc [6], and CyberCode [7] all use visual marker-based systems for motion tracking and pose estimation. Currently, there are more than a few existing marker tracking systems. Depending on the application, there are different tracking and pose estimation requirements. It is desirable to enable the AR application developer to determine which marker system is the most suitable for a given AR application [8]. In this paper, we focus on using different markers based on their features in one system and try to build a "bridge" to connect them for better performance and new user interactions. We will also introduce several kinds of markers in this chapter at first.

Image Target marker

Vuforia [9] is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to view the position and orientation of virtual objects (such as 3D models and other media) relative to real objects through the camera of a mobile device [10].

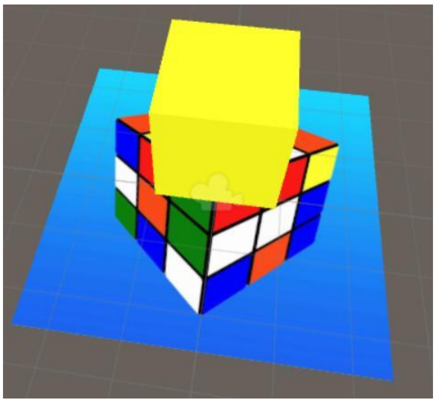


Figure 1 Image target marker.

In Li's work [11], he also used image target marker to archive his two method: encryption and permission of the secure AR (Augmented Reality). After the system detects the encrypted 3D

[1] Waseda University

virtual content, an interface will pop-up to let the user input the password. Each input will be recorded by the system database. System will match password entered by user with the correct password recorded in the database.

AR toolkit marker [4]

AR ToolKit is a software library for building Augmented Reality (AR) applications. These applications involve the overlay of virtual imagery on the real world. The AR ToolKit video tracking libraries calculate the real camera position and orientation relative to physical markers in real time. This makes it easy to develop a variety of augmented reality applications.

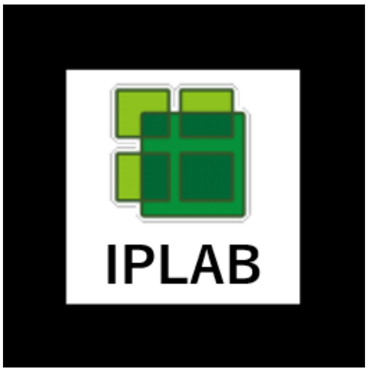


Figure 2 The AR toolkit marker.

QR Code

QR code (abbreviated as Quick Response code) is a type of trademark for a matrix barcode (or two-dimensional barcode), which was first designed for the Japanese automotive industry in 1994 [13]. QR codes are used in some augmented reality systems to determine the positions of objects in 3-dimensional space. QR Codes are also used to deliver Augmented Reality experiences. By scanning a QR Code, smartphone users can capture everyday objects in 3D and place them in their surroundings via their camera. IKEA's Place App, Puma's NYC flagship store and Lego toys have used QR Codes to deliver AR leading to a better consumer experience.

Figure 3 QR code for the URL of the IPLAB home page.

2.3 Problem

Marker-based AR system just shows separate and independent AR virtual contents above the marker. Each marker needs the specific scanner to detect the information. It means there are no connection and relationship among multiple markers.

For multiple markers, current AR system have a lot of limitations. Users cannot interact or control the AR contents by combining multiple markers as they desire. The user experience is not enough in this field.

3. A New AR System

Our goal is building a new AR system that can use various kinds of markers combining with AR markers to enhance user interactions. We have implemented a new system which can recognize various kinds of markers, i.e., multiple markers, hidden markers, and auxiliary markers. We combined these markers with AR markers to achieve our goal.

3.1 Multiple Markers Recognition

AR markers are widely used in existing AR systems. However, in current implementation. Each marker needs the specific scanner to detect their information. In our research, we try to combine multiple kinds of markers and recognize them in one system at the same time instead of just using one kind of markers.

Multiple Markers

We try to use multiple markers in our research: QR code, AR toolkit marker and Image target marker.

QR code: A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to store data efficiently. The Quick Response system became popular due to its fast readability and greater storage capacity compared to standard UPC barcodes. A QR code consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as a camera and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data is then extracted from patterns that are present in both horizontal and vertical components of the image.

AR Toolkit Marker: It is an open-source computer tracking library for creation of strong augmented reality applications that overlay virtual imagery on the real world. In order to create strong augmented reality, it uses video tracking capabilities that calculate the real camera positions and orientations relative to square physical markers in real time. Once the real camera position is known a virtual camera can be positioned at the same point and 3D computer graphics models can be drawn accurately on the real marker. The marker used is simpler than image target marker but fast enough for real time AR applications.

Image target markers: Image Target markers represent images that Vuforia Engine can detect and track. The engine detects and tracks the features that are naturally found in the image itself by comparing these natural features against a known target resource database.

Multiple Detection

In our research, the camera in the system can recognize the image target marker, ATK marker and QR code at the same time. We integrated these three kinds of markers recognition in one scanner. And the information from different markers will interact with each other.

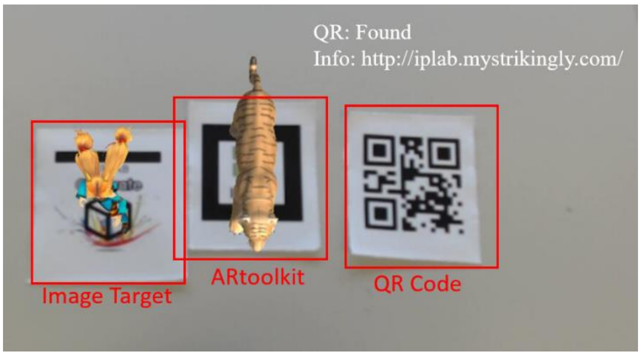


Figure 4 Multiple recognition.

Position Measure & Unify

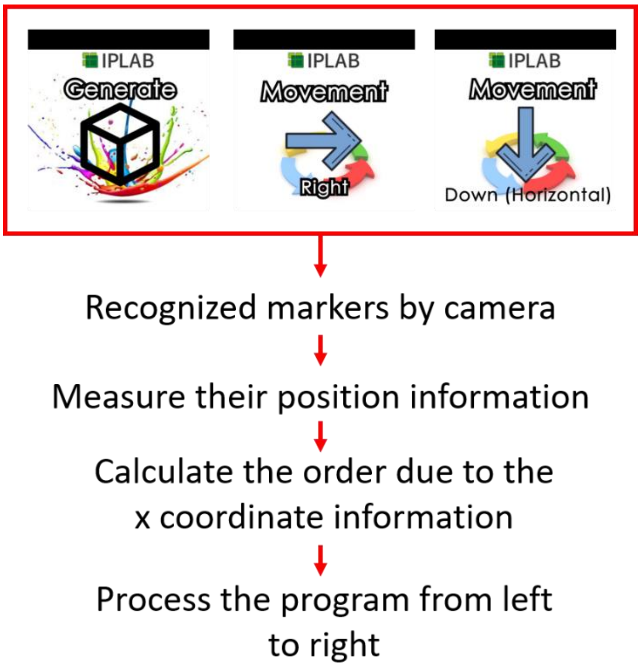
After detecting multiple markers, the system can also measure different kinds of marker positions. Each kind of SDK has different default coordinate system (origin point, left-hand or right-hand rules). Because Unity3D uses left-hand coordinate systems to define transformations. We need to convert all kinds of coordinate system into left-hand rules.

We try to calculate the pose matrix which returns the 3D position of the different kinds of markers in different environment and convert them into one camera space. All the markers position will be unified in one coordinate axis system.

Then we can use one script to storage and compare different kinds of markers x axis and y axis information. Due to the compared result, the processing order will be confirmed.

Example of Application: AR Action Sequence Manager

This application will detect all the markers in the camera and generate the command classes. And then try to measure them in one world coordinates. First, processing order is confirmed based on the x-axis information. If there are some markers placed below the main order lines, system will add them after the same x axis information marker. After all the classed lining up, system will process them at last.



Recognized markers by camera

Measure their position information

Calculate the order due to the x coordinate information

Process the program from left to right

Figure 5 Processing program by lining up command cards.

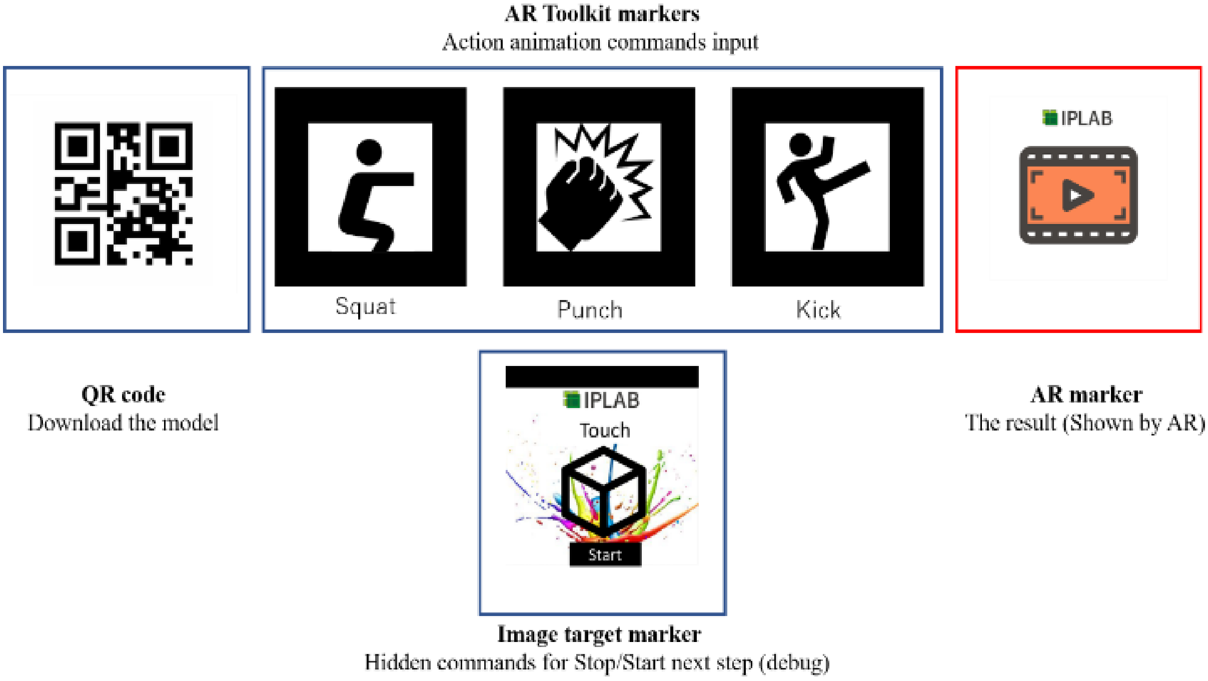


Figure 6 Different usage of markers.

We use QR code to storage the URL and simple texture for model. User can use them to download the 3D virtual contents. And the ATK marker contains some commands, like movement and action to control the 3D virtual contents. The image target marker because we can use natural feature points to make users have some intuitive interactions (put finger on button area to input commands), we can use them to debug the program that user build. Also, the result will be shown in the form of AR and generated by image target marker. In this example, the system downloads the model by recognize the QR code at first. And due to the ATK markers place order, battle action is triggered. Because the hidden marker based on image target marker is placed below the "punch" marker. After the model do the punch action, system will wait for user's touch. If user press the button area which is defined in the hidden marker. The model will start to do the next action: kick. All the result will be shown by AR.

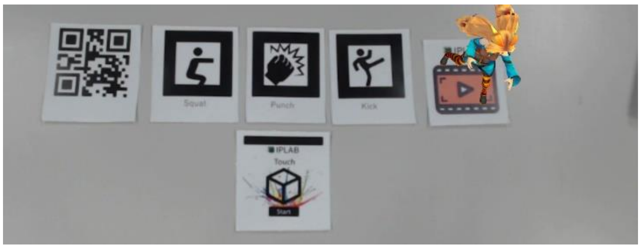


Figure 7 Output result by AR.

3.2 Hidden markers recognition

Based on the augmented reality technology, we can use the natural feature points which are recognized by system to define the button area.

When user's finger hides the feature points in one button area we defined, the system will recognize which button has been pressed. So, we also can call this kind of auxiliary marker as Hidden Marker.

Target star rating

Image target markers are detected based on natural features that are extracted from the target image and then compared at run time with features in the live camera image. Targets have a star rating between 1 and 5 stars; although targets with lower ratings (1 or 2 stars) are hard to detected and tracked. For best results, we should aim for 4 or 5 stars.

Button definition

Hidden button provides a useful mechanism to make marker interactive. The size and placement of hidden buttons must be considered when designing them. There are several factors that will affect the responsiveness and usability of hidden buttons.

1. The length and width of the button.
2. The area of the marker that it covers.
3. The placement of the button in relation to the both the border of the image, and other buttons on the marker.

Button events are triggered when a significant proportion of the features underlying the area of the button are concealed from the camera. This can occur when the user covers the button or otherwise blocks it in the camera view. For this reason, the button should be sized appropriately for the source of the action it is intended to respond to.

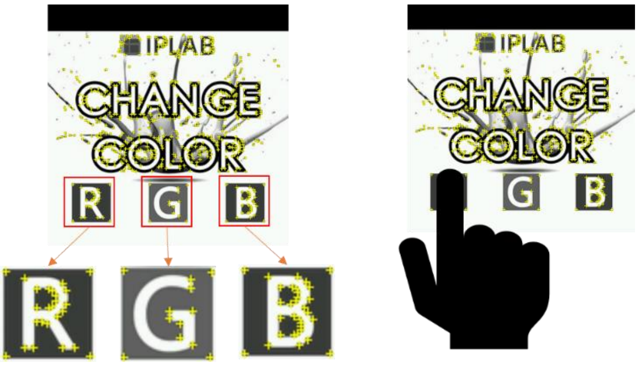


Figure 8 Hidden marker's button area.

Sensitivity Evaluation

We also need to evaluate the sensitivity of buttons. Buttons with a HIGH sensitivity will triggered more easily than those with a LOW sensitivity. The button's sensitivity reflects the proportion of the button area that must be covered, and the coverage time. We need to test the responsiveness of each buttons in a real-world setting to verify that they perform as expected.

Example of Application: AR Calculator

The AR calculator consists of two kinds of visual markers. One is AR marker for show the result by AR. The another one is auxiliary markers for user can input the numbers and operation.

The AR marker part can show the 3D virtual contents which behaves similarly to a traditional AR system. In this application, a 3D virtual calculator will be displayed on top of maker. This 3D model will output information to the user: displaying the number that user inputs or the calculated result, performing as the monitor part in a traditional physical calculator. So, we also can call this kind of marker as monitor marker.

For the user input, we also design the input markers. This kind of auxiliary markers cannot generate the 3D virtual AR contents but can help user to make some interaction with the AR contents which is generated by AR marker in the same camera view. Because the user needs to use their finger to hide the specific button area to input the information, we also can call this kind of markers as hidden marker.

Users can combine these three markers and scan them by one camera. Then they will get a 3D virtual calculator by AR and they can put the finger on the button area defined in hidden marker to input some numbers and operations to calculate. Result will be shown by AR. The interaction between the user and AR calculator is very intuitive and understandable. Every user can realize how to use this AR calculator in very short time even without any guidance.

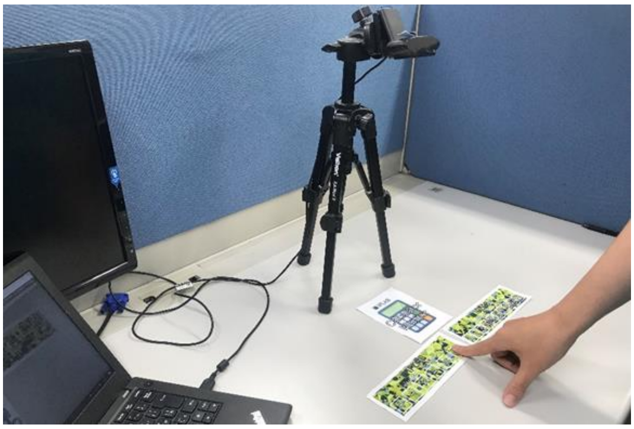


Figure 9 Combine three markers.

As for the hidden marker's algorithm, we try to enhance the hidden logic. We use 1-dimension buttons layout for inputting information. But if the user's hand hides multiple area, system will recognize the multiple button has been pressed and have a lot of wrong inputs. In our AR calculator's case, 1-dimension buttons layout for numbers input is not attractive. Commonly, calculator's keyboard buttons are 3X3 2D-Array of buttons. So, we improved our number input marker.

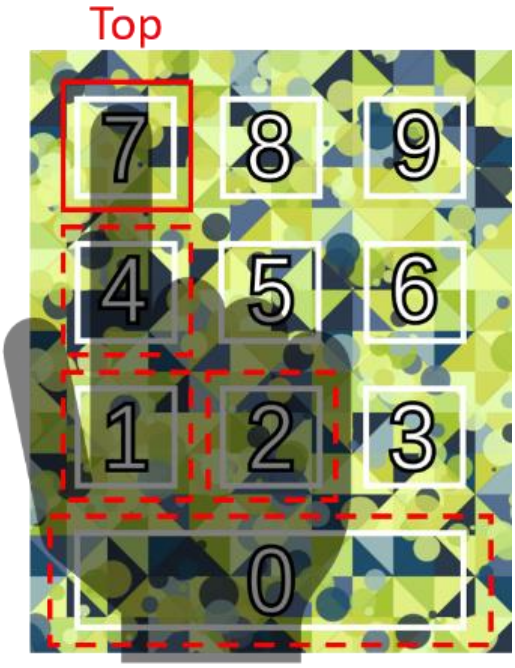


Figure 10 2D-Array of buttons.

The new number input marker uses this kind of 2D-Array of buttons. For the user input we add a new logic for it. When the system detects the button area has been hidden, the system will not generate the input commands at once. System will start to wait for 1 second give the time to user's hand slide over the marker and fix his gesture. When the user's gesture is fixed, the system will try to recognize the multiple button areas hidden by user's hand and calculate which is the top button area and generate the top button area's command.

Our AR calculator can use this kind of algorithm to provide more natural number input layout for the user.

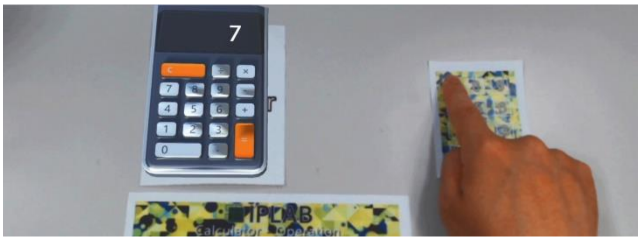


Figure 11 AR Calculator

Example of Application: AR Piano

In this part we use the method introduced in AR calculator to make the new application named AR piano. The application allows users to have multi-touch capabilities to enter multiple commands simultaneously

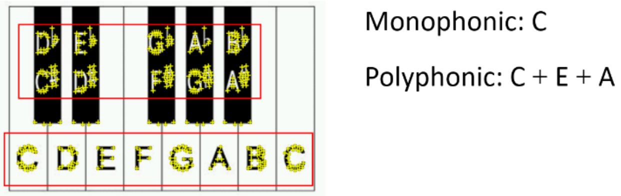


Figure 12 AR piano marker.

Similar to the formal piano in the real-world, if user touches one key, the piano will generate single tune at once, which we call monophonic. And if user touch several keys at the same time, the piano will generate several tunes at same time and combine them into a new voice, we call it polyphonic. In this research we make the AR piano by our AR system, and can generate polyphonic by multiple touches.

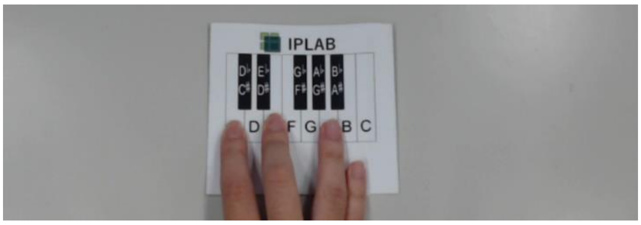


Figure 13 Multiple touch.

Users touch the C, E and A keys at the same time, the system will generate the C＋E＋A piano tune voice together and make a polyphonic voice.

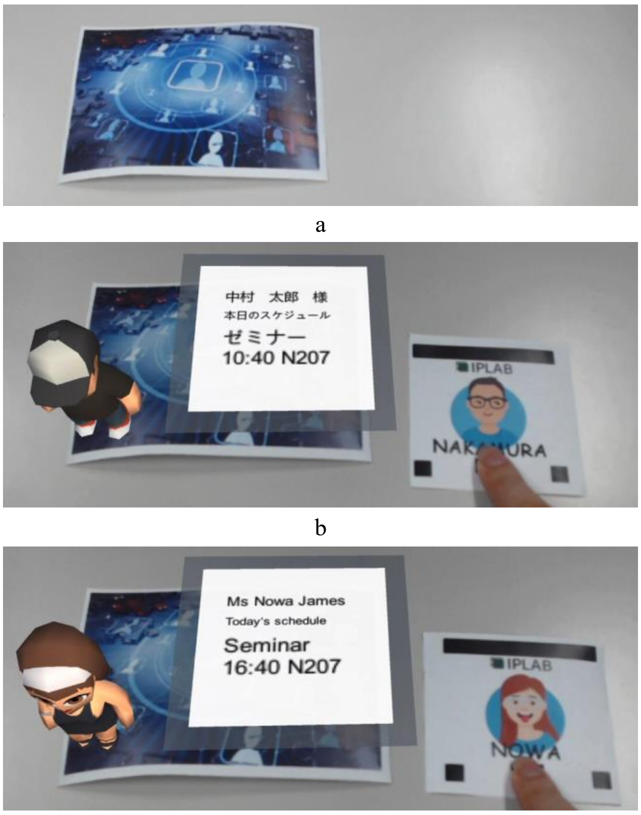
Because we use the 2D-Array of buttons's algorithm for recognition in this application, touching the black key to play some tunes and touching multiple black key to play polyphonic are also available.

3.3 Auxiliary Marker

Personalization

The identity of the user is an important condition for the customized AR display. One of the cores of customized Augmented Reality is to display different 3D virtual content for users with different identities. In our system, we use a kind of auxiliary marker to connect to the database which stores the user information. And when the user puts the auxiliary marker with the AR marker, the user information will be checked by system and show the customized contents to the user.

In this research we have set an AR marker in the lab. The fictitious student A and student B will receive different information on the 3D-virtual Callboard activated by this AR marker. Because when they want to get information from this AR marker, they need to put their own auxiliary markers which contains their personal information with the AR marker together.



a

b

c

Figure 14 Get customized information from the AR marker.

When there is no auxiliary marker, the AR marker will not show anything, like figure (a) shows. And after the student put his own auxiliary marker with it, the AR content will be shown. We can see:

> **Taro Nakamura**: He will see his today's schedule by Japanese according to his class and prefer language. He also can see his 3D virtual avatar by AR, like figure 28 (b) shows.
>
> **Nowa James**: She will see her today's schedule by English according to her class and prefer language. She also can see her 3D virtual avatar by AR, like figure 28 (c) shows.

As the result, our system supports two different nationalities and class students in one lab can get their different schedules from one AR marker just by putting their auxiliary markers with the AR marker without any ID or password input.

Encryption/Decryption

Sometime the providers of 3D virtual content do not want all the users to see the hidden information. On the other hand, giving other people your personal account and password is very dangerous. In our research, the user can use the auxiliary card which can perform as a key to unlock some encrypted AR contents. The key is only for decrypting and does not contain any personal information. We designed two simple usage scenarios to explain how encryption/decryption part is working in this system.

In the usage scenario of this research, there are two users: user A and user B. User A prepares a mail which contains AR contents invite user B and his family to user A's house. Because the mail contains address information, user A does not want other people to see the contents, so he encrypted the AR mail then post it and pass the auxiliary key marker to user B by another way.

User B receives mail and key by two different ways. If he just uses the camera to scan the mail marker without auxiliary key marker, he can only see the text "No permission". And if he puts the auxiliary key marker with the mail marker, the 3D virtual contents containing 3D avatars, address information, map and photo will be shown. And if he wants to let his family know this, he just needs to pass the mail and auxiliary key marker. He does not need to share his account and password.

4. Implementation

4.1 Unifying the Coordinates

Unity3D uses coordinate systems to define object's position. The x, y, z axis directions are visualized with different colors, and are following left-hand rules. We need to unify all the visual marker's coordinates in left-hand rules.

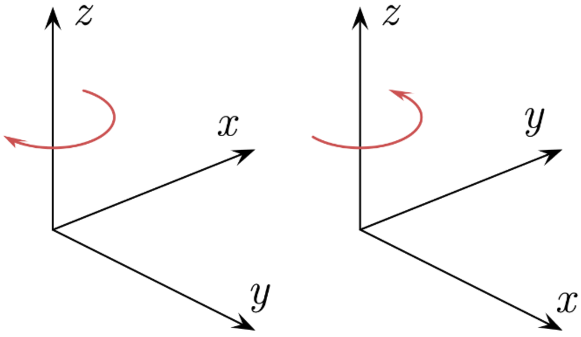


Figure 15 Left-hand and right-hand coordinate system.

Coordinate of QR code: The QR code captured by the camera is a perspective transformation. From the surface where the QR code is to the image, a perspective transformation is applied. We need to solve a projection matrix for that transformation. The camera detects the QR code and returns the 4 corners and the correct direction. Then we use this matrix to calculate the left-hand coordinate that this QR code is currently at.

Coordinate of AR Toolkit: Different from the Unity coordinate system (left-hand matrix). AR Toolkit uses right-hand coordinate systems. We need to convert the information by using a utility function existing in the Unity package of AR Toolkit:

public static Matrix4x4 LHMatrixFromRHMatrix (Matrix4x4 RightHandM)

It can convert the tracking result of AR Toolkit to Unity environment. Then we can get left-hand pose matrix of AR Toolkit markers.

Coordinate of Vuforia: In Vuforia, the World Center Mode allows specifying how the World reference frame is set during tracking. And all the modes are using left-hand coordinate systems.

- FIRST_TARGET: the world is fixed relative to the first detected.
- SPECIFIC_TARGET: same as FIRST_TARGET, except here (in the editor) it is explicitly specified which target is the "anchor".
- CAMERA: in this case the AR Camera is fixed (fixed position and orientation) in the scene, while the Target objects move around it.

We assume the camera is at (0,0,0) in the different coordinate systems then get the pose matrix which returns the 3D position of the different kinds of markers in the real-world. Then we convert this information into 2D camera space, and transfer this into our screen space. All the markers position will be unified in one coordinate axis system.

4.2 Hidden Marker

For designing the hidden marker, we need to define the button area at first. Because we achieve this function by Vuforia virtual button service, the image target marker's natural feature points can help us to distinguish the different area we predefined if the image is recognizable enough.

We set the events that are triggered when a significant proportion of the features underlying the button area we defined are concealed from the camera. This can occur when the user covers the button by their finger in the camera view. Users also can have multiple touches with the hidden marker and trigger multiple events at the same time.

After designing and uploading them to the Vuforia database as markers, we can import them into unity and start to define the button area. In our case, we use green color to define the button area.

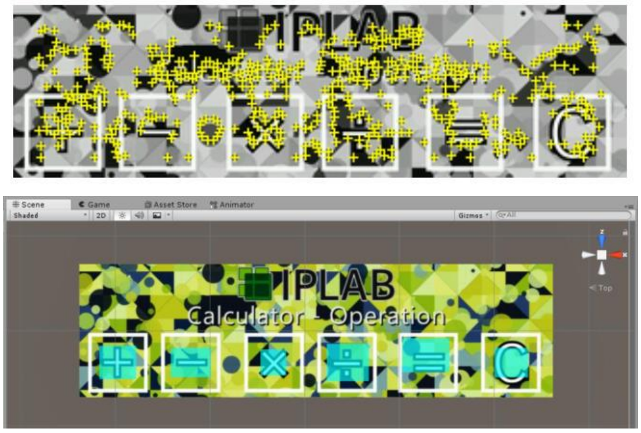


Figure 16 Feature points and button area.

In the example of AR Calculator, we use hidden areas to trigger the input event and the system can calculate the result after user input the number and operation commands.

We use two functions to achieve our goals and integrate them in one script which we call it calculator manager.

About the 2D-Array of buttons we designed an additional logic based on hidden algorithm:

1. When system detects the button area has been blocked, it will wait for 0.5s so that user has time to slide his finger over the marker and stop to fix the gesture.
2. After 0.5s, system will detect all the blocked button and find which is the top one.
3. The top blocked button will be realized by system when it is pressed by user's finger. Then system will then process this button trigger event.

To design this kind of hidden marker, we need to design the button area at first and make them clear for user to understand and can be detected easily.

5. Conclusion

In summary, we have raised the problem of the tradition marker-based AR system and design a new system to create new interactions between users and markers by combining multiple markers, hidden markers and auxiliary markers.

For combining multiple markers, we integrated three kind of markers in our system: Image target marker, AR Toolkit marker and QR code. Different kinds of markers have their own attributions and are suitable for different application areas. Due to their different features, we give them different function and combine them in one system. The scanner in our system can detect these three kinds of markers at the same time and process them in a certain order with 2-dimension layout in one camera view.

The hidden marker is kind of additional controller for the AR marker in our marker-based AR system. These markers cannot generate any 3D virtual AR content in user's view. However, based on the natural features extracted from the target image markers and comparing them with the features in the live camera image at runtime, we can define several button areas on the marker. When a significant proportion of the features underlying the area of the button are concealed from the camera, it will trigger some events. This can occur when the user covers the button or otherwise blocks it in the camera view. As the result, the hidden marker allows user to have very intuitive interactions with our AR system. We also enhance the recognition logic to make the system more effective so that it can support 2-dimension layout hidden marker working. For the auxiliary markers, they perform as a token or key to get specific or decrypted information. This kind of marker can help user to protect some unpublic AR contents in some cases. In the personalization part, the system can display different 3D virtual content for users with different identities contained in the auxiliary marker. In the encryption/decryption part, the system can unlock some encrypted AR contents through auxiliary markers which are only used for decrypting and do not contain any personal information.

In this research, we also present some applications based on our new system. We have implemented Multiple markers combination, AR Calculator and AR piano. We also discussed some usage scenarios for personalization and encryption/decryption parts.

Reference

[1] Wikipedia contributors. “Augmented reality”. Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2020, from https://en.wikipedia.org/w/index.php?title=Augmented_reality&oldid=939905238.

[2] David Chen, William Garrett, Mark Livingston, Andrei State, Gentaro Hirota. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 429–438, 1996.

[3] Anuroop Katiyar, Karan Kalra, Chetan Garg. Marker Based Augmented Reality. Proceedings of Computer Science and Information Technology (ACSIT), pp. 441-445, 2015.

[4] Eric Woods, Paul Mason, Mark Billinghurst. MagicMouse: An Inexpensive 6-Degree-of-Freedom Mouse. Proceedings of Graphite 2003, pp. 257-260, 2003.

[5] Nassir Navab, Echeyde Cubillo, Benedicte Bascle, Jurgen Lockau, Klaus-D Kamsties, Michael Neuberger. CyliCon: a software platform for the creation and update of virtual factories. Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98), pp. 459-463 vol.1, 1998.

[6] Xiang Zhang, Echeyde Cubillo. Tracking and pose estimation for computer assisted localization in industrial environments. Proceedings of the 5th IEEE Workshop on Applications of Computer Vision, pp. 214-221, 2000.

[7] Jun Rekimoto, Yuji Ayatsuka. CyberCode: designing augmented reality environments with visual tags. Proceedings of the 2000 ACM Conference on Designing Augmented Reality Environments (DARE 2000), pp. 1-10, 2000.

[8] Karan Kalra, Anuroop Katiyar, Chetan Garg. Augmented reality and improved ransac. IJETSR, pp. 55-62, 1998.

[9] Andrey Borisov, Jürgen Sieck, Leonard Ashikoto, Gabriel Kamenye, Jason Mwenyo, Nawa Likando. Development of an efficient, cost-reducing content management system for augmented reality applications. Proceedings of the Second African Conference for Human Computer Interaction: Thriving Communities, pp. 1-4, 2018.

[10] Wikipedia contributors. “Vuforia Augmented Reality SDK”. Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2020, from https://en.wikipedia.org/w/index.php?title=Vuforia_Augmented_Reality_SDK&oldid=915325078.

[11] Bowen Li. “Research on Customized and Secure Augmented Reality”, Master thesis, Graduate School of Information, Production and Systems WASEDA University Interactive Programing Laboratory (IPLAB), 2019-07.

[12] Wikipedia contributors. “Barcode”. Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2020, from https://en.wikipedia.org/w/index.php?title=Barcode&oldid=940244179.

[13] Wikipedia contributors. “QR code”. Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2020, from https://en.wikipedia.org/w/index.php?title=QR_code&oldid=939675549.

[14] Jun Rekimoto, Yuji Ayatsuka. Active cybercode: A directly controllable 2d code. Proceedings of the CHI 2006, April 22–27, pp. 490-495, 2006.

[15] Ronald Azuma. A survey of augmented reality. Presence: Teleoperators & Virtual Environments, pp.355-385, 1997.