

Apr 20, 2023

Version 2

ARIAS: An AR-based interactive advertising system V.2

DOI

dx.doi.org/10.17504/protocols.io.e6nvwjyq2lmk/v2

wantchoosejoy¹

¹Southwest Jiaotong University Hope college



wantchoosejoy

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.e6nvwjyq2lmk/v2>

Document Citation: wantchoosejoy 2023. ARIAS: An AR-based interactive advertising system. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.e6nvwjyq2lmk/v2> Version created by [wantchoosejoy](#)

License: This is an open access document distributed under the terms of the **[Creative Commons Attribution License](#)**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Created: April 13, 2023

Last Modified: April 20, 2023

Document Integer ID: 80444

Keywords: gesture, AR, advertising, interactive advertising system aria, based interactive advertising system aria, advertising system, based advertising system, aria, ar, gesture

Abstract

ARIAS is an AR-based advertising system, it can be manipulated by gestures interactively.

Attachments



GestureImage.rar

24.4MB



Gesture recognition ...

42KB



UserTestData.xlsx

14KB



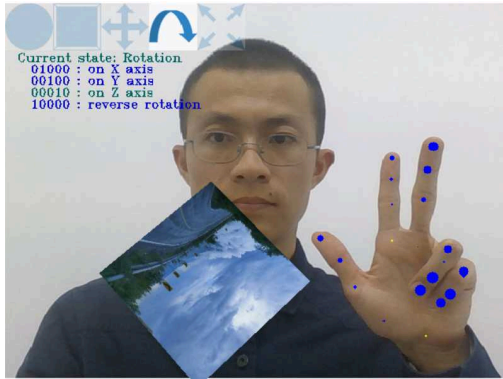
Source code.rar

20.7MB

Troubleshooting

Introduction

It is an interactive video playback system based on augmented reality and gesture recognition. The system recognizes the image captured by the camera in real time. When there is a predefined square marker, A cube is created at the marker position, and the pre configured video files are played on the six faces of the cube. The cube can be manipulated through gestures. It can be used for campus publicity, shopping mall guidance, scenic spot introduction, etc.



User guide

1. Ensure the computer has at least one webcam.
2. Install Python and other external library(MediaPipe, OpenCV, OpenGL...).
3. Show one marker in front of the webcam. The markers can be found in "Markers" folder. Now the feedback of the system will contain a virtual cube on the marker.
4. Show your hand and make gestures to manipulate the cube. There are six states when manipulating the cube. One gesture can indicate different instructions in different states.

Gestures statement

- | Gestures | statement | Gestures | states |
|----------------------|---|----------|--------|
| 1. Initialization: | the program has just been initialized and no marker has been detected | | |
| 2. Fronting state | | | |
| 3. Translation state | | | |
| 4. Movement state | | | |
| 5. Rotating state | | | |
| 6. Zoom state | | | |

State switching

make gesture 00000 then 11111 to switch state. The five binary digit mean the state of five fingers, 0 mean closed and 1 mean expanded. The order is thumb, index finger, middle finger, ring finger and little thumb.

Fronting operation

1. Front view 1: 01000
2. Front view 2: 01100
3. Front view 3: 01110
4. Front view 4: 01111
5. Front view 5: 11111
6. Front view 6: 10001

Translation operation

1. To left : 01000
2. To right : 01100
3. To up : 01110
4. To down : 01111

Rotation operation

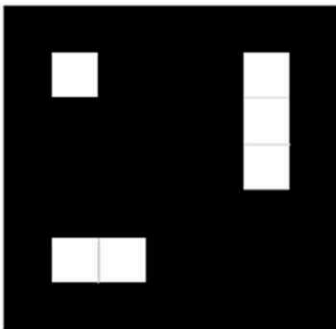
1. X +: 01000
2. Y +: 00100
3. Z +: 00010 The three fingers can be combined arbitrarily, and when the thumb is expanded, the cube has a reverse rotation.

Zoom operation

1. Zoom in: 01000
2. Zoom out: 11000

Marker definition

1. 5×5 white cells with black background. There are four corners of the marker and only one corner is an isolated cell, that is, the surrounding color is black. The last row and the last column have white cells, that is the sum of the values in the last row and the last column cannot be 0 (black 0, white 1)
2. The marker with isolated cell in the upper left corner is regarded as the standard marker. When the isolated cell is at the other three corners, the marker is considered to be the result of plane rotation on the basis of the standard marker.



$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Software architecture

- GeneralFunctions: Some common methods for the whole workspace.
- GestureRecognizer: Recognize the gesture signal.
- HandPointsProvider: Use MediaPipe as the frame for hand key point recognition to extract 2.5D coordinates of 21 key points of the hand.
- MarkerRecognizer: To recognize Markers.
- FaceBlanking: Optimization of box rendering, that is, not rendering the occluded faces.
- ARIASMain: The main entity.