

# Component Level Interaction of a 3D Model in an Interactive Augmented Reality Environment

Peng Peng Leim and Guat Yew Tan

**Abstract**—In this paper we define interactive Augmented Reality (AR) as the ability to display, interact and control of the virtual 3D model in the real environment. We also define 3D components as the pieces which form the whole 3D model. Interactive AR involves the interaction between two entities, i.e. input device and the displayed virtual model. Our emphasis is placed on the displayed virtual model, in which its components can be moved and controlled individually. Our AR application contains 3 main processing units, i.e. OBJ file parser, rendering machine, and display & interactive unit. The file parser is the most important unit in our application, used to segregate each component of the 3D model from the input OBJ graphics file into different containers. The segregation is the key process to make the interaction with the 3D model at component level feasible. Rendering machine performs the transformations of the individual 3D components based on the information stored in the component containers and the input value received from the display & interactive unit. The display & interactive unit displays the virtual 3D model in the real environment and capture the input instructions given by the users.

**Index Terms**—User interaction, 3D component level interaction, augmented reality.

## I. INTRODUCTION

The key attraction in Augmented Reality (AR) is its fundamental capability to enable the virtual and real objects co-exist in the real environment. However, human is adventurous and thus unsatisfied with only the co-existence status. The urge to interact with the virtual objects in the real environment has caused further research work and resulted in interactive AR. Typically, interactive AR involves the interaction between two entities, input device and the displayed virtual object. Our main discussion is on the displayed virtual object, which is commonly called a three dimensional (3D) model.

A 3D model can be created in two ways, by drawing the model within the AR program itself or by importing the model which was created in a third-party modeling application such as Ogre [1], Autodesk 3D Studio Max [2], Autodesk AutoCAD [3], AC3D[4], etc. A 3D model is formed by parts called components. Sometimes, it is called *entities* in DXF [5], *groups* in OBJ file [6] and AC3D [7], *objects* in 3D Studio Max [8]. Our definition of interactive AR includes display, interact and manipulate the virtual objects at component level.

Manuscript received Decemberr 19, 2012; revised February 23, 2013. This work was supported by the Malaysian Ministry of Higher Education under Exploratory Research Grant Scheme (ERGS) with reference number 203/PMATHS/6730018.

The authors are with the School of Mathematical Sciences, Universiti Sains Malaysia, and Penang (e-mail: pengpeng\_leim89@hotmail.com, gytan@cs.usm.my).

For a simple 3D model which involves only a few components, it can be created within the AR application itself by defining the properties of the components when the program is developed. In this case, each component in the 3D model created can be controlled individually and thus animated easily as the programmer knows the attributes of the model especially coordinates of each component. ARTTT [9], ARWorm [10] and ARPuzzle [11] are three examples of AR applications which draw the virtual objects within its program. In ARTTT, the virtual pens control where it should place the virtual cross and circle marks; in ARWorm, the body of the virtual worm grows sphere-by-sphere and the virtual book can be placed randomly; in ARPuzzle, each virtual puzzle piece is moveable by a virtual rod, and scattered all over the board until the complete puzzle is formed. [9]–[11] enables user-interactions at component level.

However, it is common that a 3D model tends to be more complex and involves hundreds or even thousands of components. Thus, it is much more efficient to create the complex 3D model in a third-party application and import the graphics file of the 3D model to the AR application. The format of the graphics files produced is important in interactive AR. Raster (bitmap) format contains only the attributes of each pixel, massive programming work has to be carried out in order to recognize the model. Furthermore, the depth values of the model are not preserved in the raster file and thus unsuitable to be used in interactive AR. Vector format contains attributes of the components (instead of pixel as compared to raster format), interaction with the object is achievable by extracting the coordinates of each 3D component.

Many AR applications which claimed to be interactive provide the user interaction only at global level, i.e. the whole model is responsive to the user-interaction by repeating the animation which was pre-created in the third-party application. Though there are user interactions at component level in some AR applications, the interactions are limited to only a few components out of hundreds of components of the 3D model.

A few automobile companies have incorporated AR technology into their 2D advertisements to provide 3D visual effect in the real environment. MINI displays the virtual 3D model of its Cabrio convertible on an advertisement of a magazine with no user interaction feature [12]. Audi displays its Audi A1 model on an A1 marker. Users seem to be able to interact with four of its components, i.e. sunroof, doors, car boots and the four wheels, individually. However, the animations of the components are actually separated video files with pre-created animations and responding to the mouse selections on the virtual buttons [13]. Interactions with

other car components, e.g. switching on the head light, opening of the back door, adjusting the side mirror, etc. are not part of the AR interactions in the advertisement.

In our project, we endeavor to achieve the interactive AR which is able to display, interact and control the 3D model at component level.

## II. PROJECT OVERVIEW

Our project imports an OBJ graphics file which contains a 3D car model in vector format. The virtual car will be displayed on an AR marker, allowing the users to interact with 14 car components as shown in Table I. Global interactions such as disassembling the car completely are included as well. The existing version of our project focuses on the 3D model and thus the keyboard is used as the input device to simplify the application.

## III. COMPONENTS AND PLATFORM

Our project contains 3 main processing units, i.e. OBJ file parser, rendering machine and display & interactive unit (Fig. 1).

### A. OBJ File Parser

File parser is the main engine in our project to identify each 3D component of the car model from a graphics file. The graphics file type used in our project is Wavefront's object (OBJ) file version 3.0 in ASCII format. After reviewing a set of common graphics files available [6], it is selected for its readability, ability to store unlimited number of triangle patches (only bound by the hard disk space) and available freely [14], [15].

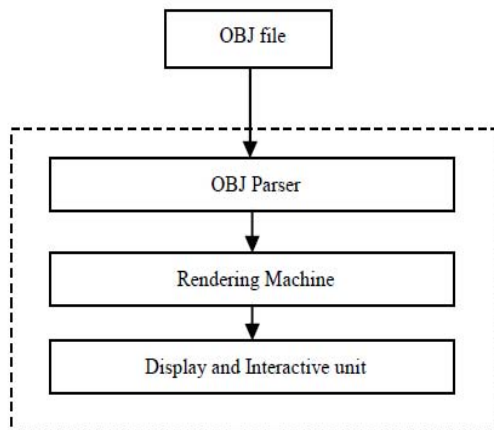


Fig. 1. Three main processing units

In the OBJ file, the first keyword appears in each line is parsed to decide the data type for the line. To identify the coordinates and components, we are interested in a few keywords, i.e. g, v, vn and f. The keyword g denotes a group line and also determines the start of a group. The definition of *group* in OBJ is similar to *component* in our definition. The blocks of information following the group line belong to the same component, until the next group line is encountered. The keyword v denotes the 3D coordinates of the vertex, keyword vn denotes vertex normal and f contains the

relationships between the vertices and vertex normals [16]. A template container is created to store the component (Fig 2). As the file is being parsed, a new instance of the component container is created whenever a group is encountered. The information of the relevant vertices, vertex normals and faces are being assigned to sub-containers of the group container, as shown in Fig 2. The sequence of every vertex lines and vertex normal lines appear in the OBJ file are counted and recorded, as the keyword f relates to respective vertex and vertex normal by referring to the sequence number.

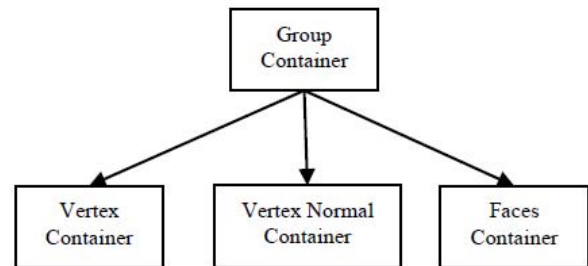


Fig. 2. Component container

### B. Rendering Machine

This unit performs rendering of the 3D car by taking the information stored in the component instances. Triangles are generated as 3D patches to form the components, which in turn form the 3D car.

Transformations (motions) of each car component are calculated based on the input instructions given by the users in the display & interactive unit.

### C. Display & Interactive Unit

TABLE I: KEYBOARD FUNCTION MENU

Key	Part of the car
a	Car disassembling
b	Bumper
c	Tyre
d	Hood and Fender
f	Door and Window
g	Mirror
q	Head Light
r	Fog Light
s	Tail Light
t	Roof
u	Boots
v	Interior
w	Exhaust
x	Trunk
1	Restore defaults
2	Zoom In
3	Zoom Out
ESC	exit

Display & interactive unit is in charge of the AR part by merging the 3D car and the real environment. In this unit, A Toolkit is used after reviewing the set of open-source AR tools available [17]. Keyboard is used as the input device at this stage. There are 14 components, as shown in Table I, which have been identified by the file parser. The components are moveable individually with respect the input key selected.

#### IV. PLATFORM

A few application programming interface (API) library tools are used. OpenCV is used on the functions related to camera, OpenGL is used on the graphical functions for animation, ARToolkit is used for displaying the virtual model in the real environment.

In terms of hardware components, our project requires a laptop and a normal web-camera for displaying the AR objects and the interactions.



Fig. 3. The car is disassembled, showing each component, when key 'a' is pressed. The car interior can be viewed clearer when zooming in.

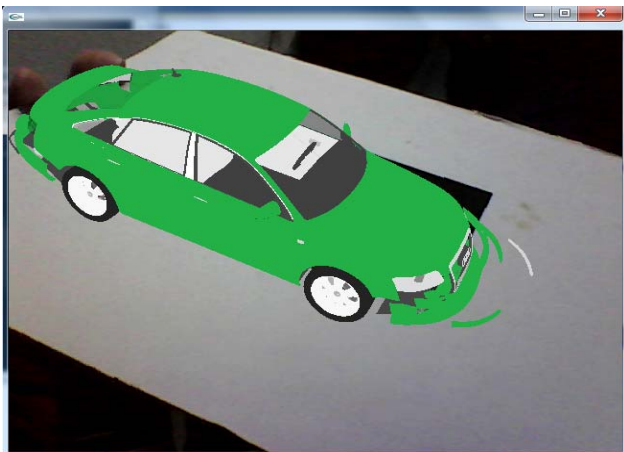


Fig. 4. The car bumper is moved away from the car with key 'b', displaying the shape and thickness of the bumper.

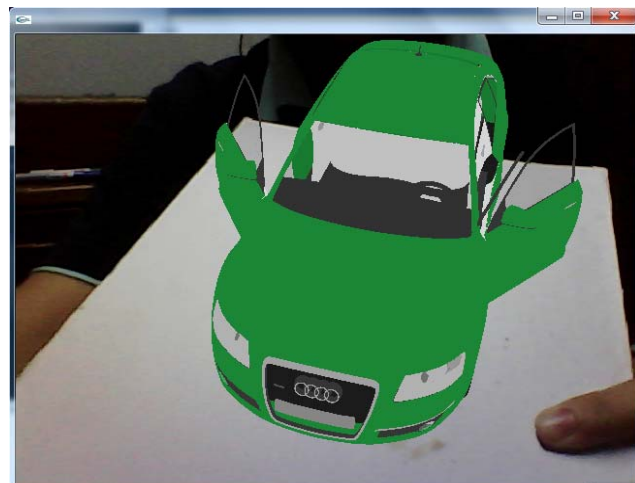


Fig. 5. Moving front door and window with key 'f'.



Fig. 6. Moving the car roof with key 't'. Car interior can be viewed from the top elevation.

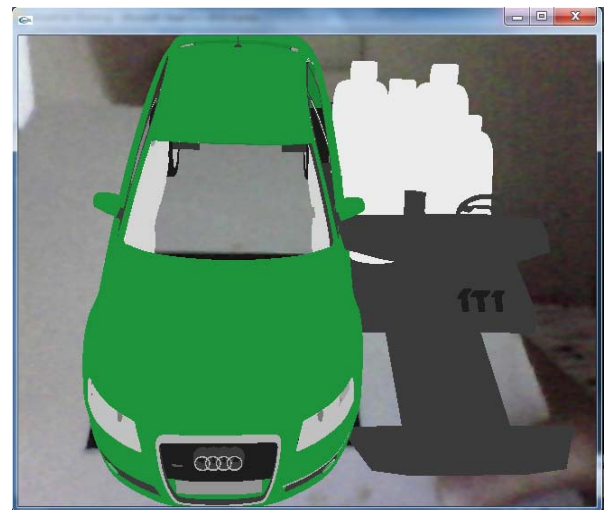


Fig. 7. Moving the car shell away from its interior for better view with key 'v'

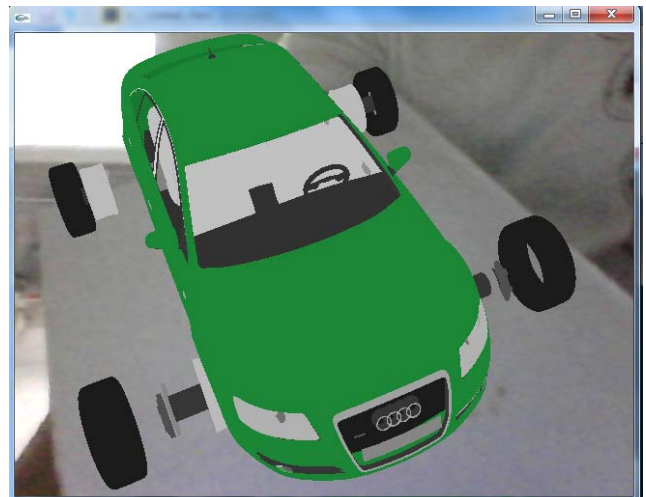


Fig. 8. Press key 'c' to view the tyres.

#### V. RESULT

This section shows images of various car components moveable from its main body by respective key-press. The car and its separated components can be viewed from different angles. Fig. 3-8.

## VI. CONCLUSION AND FUTURE PLAN

An interactive AR application which imports an OBJ file and enables the users to interact with the virtual 3D model is developed. The strength of our application is its ability to perform user-interaction with the virtual model at component level. As our application is still at its infancy and we shall see enhancement such as replacing keyboard with bare-hand interaction. Further, the file parser can be extended to work with more file types.

### REFERENCES

[1] OGRE - Open Source 3D Graphics Engine. [Online]. Available: <http://www.ogre3d.org/>

[2] 3ds Max - 3D Modeling, Animation, and Rendering Software – Autodesk. [online]. Available: <http://usa.autodesk.com/3ds-max/>

[3] AutoCAD 2013 - 3D CAD Design Software - AutoCAD 2012 Upgrade. [Online]. Available: <http://usa.autodesk.com/autocad/>

[4] AC3D - 3D Software. [Online]. Available: <http://www.inivis.com/>

[5] AutoCAD 2012, DXF Reference. [Online]. Available: [http://images.autodesk.com/adsk/files/autocad\\_2012\\_pdf\\_dxf-reference\\_enu.pdf](http://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf)

[6] J. D. Murray and W. Vanryper, Encyclopedia of Graphics File Formats, 2nd Edition, O' Reilly & Associates, 1996.

[7] AC3D v6.8, User Manual. [Online]. Available: <http://www.inivis.com/ac3d/download/ac3dman68.pdf>.

[8] Autodesk Composite User Guide. [Online]. Available: <http://www.scribd.com/doc/58085720/>

[9] P. L. Kok, M. Y. Ow, B. S. Woun, K. Y. Lee, and G. Y. Tan, "ARTTT: The Facelift of Tic-Tac-Toe," presented at the Artificial Intelligent Workshop (AIW 2011) in conjunction with Knowledge Technology Week (KTW 2011), Universiti Tenaga Nasional, Putrajaya, Malaysia, Jul 18-22, 2011.

[10] S. Ng, T. K. Fong, K. Y. Lee, B. S. Woun, and G. Y. Tan, "ARWorm: An Augmented Reality Game for Hand-Eye Coordination Training," presented at the Artificial Intelligent Workshop (AIW 2011) in conjunction with Knowledge Technology Week (KTW 2011), Universiti Tenaga Nasional, Putrajaya, Malaysia, Jul, 18-22 2011.

[11] B.S. Woun and G. Y. Tan, "ARPUZZLE: Deployment of Interactive Augmented Reality in Jigsaw Puzzle," presented at the International

Conference in Artificial Intelligent, WorldComp 2012, Nevada, Las Vegas, Jul 16-19, 2012.

[12] Mini Augmented Reality Ads Hit Newstands. [Online]. Available: <http://technabob.com/blog/2008/12/17/mini-augmented-reality-ads-hit-newstands/>.

[13] Audi A1 – Augmented Reality – Audi.co.uk. [Online]. Available: <http://www.audi.co.uk/new-cars/a1/a1/augmented-reality.html>

[14] A Tiny Wavefront Object Loader – Part I | Benjamin Sommer Weblog. [Online]. Available: <http://weblog.benjaminsommer.com/blog/2012/02/12/a-tiny-wavefront-object-loader-part1/>

[15] Shatters.net • View topic - cmodview: 3D mesh file utility for Celestia. [Online]. Available : <http://www.shatters.net/forum/viewtopic.php?f=25&t=15775>

[16] Wavefront .obj file - Wikipedia, the free encyclopedia. [Online]. Available: [http://en.wikipedia.org/wiki/Wavefront\\_obj\\_file](http://en.wikipedia.org/wiki/Wavefront_obj_file)

[17] K. P. Ng, G. Y. Tan, and L. Y. Iman, "Overview of Augmented Reality Tools," presented at the National Seminar in Science and Mathematical Applications (SKASM) 2010 in conjunction with 18th National Symposium in Mathematical Sciences (SKSM), Johor Bahru, Dec. 8-10, 2010.



**Peng Peng Leim** was graduated with BSc degree in Applied Science major in computer modelling in 2012, from Universiti Sains Malaysia (USM), Penang, Malaysia.

She is a full-time research student at USM at present.



**Guat Yew Tan** was graduated with Master of Applied Science (1996) from Nanyang Technological University, Singapore.

She is a senior lecturer at School of Mathematical Sciences, Universiti Sains Malaysia (USM), Malaysia. Prior to her assignment in USM, she worked at the industrial sectors including consulting work in the area of Information technology.

Ms Tan is a life member in PERSAMA, Malaysia.

She has written the book *C++ Programming: An Introduction*.