SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 27 (2011) 274 – 280

Procedia
Social and Behavioral Sciences

Pacific Association for Computational Linguistics (PACLING 2011)

UIAR Common Sense: an augmented reality framework for creating games to collect common sense from users

Svetoslav Dankov[a*], Rafal Rzepka[a] and Kenji Araki[a]

[a]*Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9, Sapporo 060-0814, Japan*

Abstract

Augmented Reality (AR) applications have become widespread with the continued miniaturization of technology. With the increasing use of smart phones, which often provide increased processing power, enhanced and open software platforms, Augmented Reality has become instrumental in the way we perceive our surroundings and the information that it carries. Augmented Reality has also become a welcome visualization tool for many fields, not restricted to Human-Computer Interaction. In this paper we present a novel approach for building interactive interfaces using Augmented Reality and we give an example how one can use our framework for creating games to collect common sense knowledge from users. We present a software framework for ubiquitous Augmented Reality enhancement for human-computer interaction called UIAR (User Interface through AR). Our framework improves on four areas in Augmented Reality development that we currently see lacking.

Keywords: Augmented Reality Based Games; Human-Computer Interaction; Common Sense Acquisition

* Corresponding author. Tel.: +81-11-706-7389; fax: +81-11-709-6277.
E-mail address: dankov@media.eng.hokudai.ac.jp

doi:10.1016/j.sbspro.2011.10.608

1. Introduction

In the field of collecting common sense knowledge data from a large number of voluntary users, one of the biggest challenges is to keep the users engaged, entertained, and focused so as to collect a sufficiently large amount of high-quality data. Most such projects start with a somewhat big user-base, which unfortunately dwindles in numbers and activity as the project grows older. We believe using UIAR to implement games can prove useful in this area.

In our research we implement an Augmented Reality system which will serve as an extension to existing computer interfaces, provide enhanced user-experience, and define virtual objects and their actions in an ubiquitous way. Augmented Reality is a fairly young area of research which is currently expanding in many of the already existing fields of Human-Computer Interaction, Computer Interfaces, etc. During the design stage of our system, however, we had to first address two major areas which were problematic for implementing user interfaces through Augmented Reality - ubiquity and interactivity.

We begin by presenting a summary of the related research, connecting our approach with previous work. Next we present use cases which describe specific functionalities that our framework enables and describe the purpose of the framework. This is followed by a description of the software tools and libraries we take advantage of while implementing the framework. We conclude by presenting some implementation scenarios for games collecting common sense knowledge from users using our framework.

2. Related Works

2.1. Interfaces, Augmented Reality applications and frameworks

The technologies for hand and finger-based interfaces can be roughly split in two categories - sensing-based and computer-vision based. Sensing-based systems like [11] are very robust but are often limited to detecting only ”touch” behaviour, not able to recognize hands or other physical object that come into view. Computer-vision based systems like [3] are often limited by the lighting conditions and may not respond well to sudden changes in the field of view. However, systems like [5] have proven to be robust and accurate enough. We are using a computer-vision based system since wearing special hardware to enable ”touch” capability reduces the mobility of the system.

There have been many Augmented Reality applications, using either multiple-camera hand and object tracking or a single camera (like a webcam). Those applications vary in both their mobility and complexity. Our project was inspired for the most part by the Sixth Sense project developed by Pranav Mistry in the MIT Media Lab [9]. As is the purpose of [9], we strive to provide mobility, affordability and ubiquity to Augmented Reality applications.

2.2. Approaches to collecting common sense

With the realization of the importance of common sense to the field of artificial intelligence, considerable research has been done towards collecting and structuring this type of knowledge. The biggest research effort by far has been the Cyc project, which has already collected over a million common sense assertions in little over two decades. As the project became more of a commercial venture, a much smaller set of data is available free of charge. The work required, however, has been considerable. Common sense knowledge is manually input by experts in particular areas, who first give a complete ontological structure to the data, using a specially developed knowledge representation language called CycL, and then insert domain specific data based on their expertise [6].

Another attempt to collect common sense data is the Open Mind Common Sense project. OMCS collects common sense statements from untrained volunteers over the Web in the form of natural language statements [13]. In the course of few years the project had already collected over 1.6 million statements.

Other systems, like Verbosity [1] and Common consensus[8] identified and addressed one of the major problems with such systems - user interest. In order to consistently gather quality knowledge from a large set of volunteers, they must be given enough motivation to continue to participate, especially in light of the fact that the number of volunteers drops over the life of the project.

3. Framework purpose

In the following section we formally introduce the problems and issues current AR systems suffer from and briefly outline how our framework tackles them:

- Ubiquity of Object presence:
  Objects associated same independent of system used to view the AR environment. Objects are first registered via a QR code which encodes the URI of the repository from which the object graphics model and other data is to be drawn. This gives each AR marker (independent of its graphical representation) an unique identifier. This unique identifier allows us to implement the next three objectives.
- Object Persistence:
  Objects must carry associated data and object states across different AR environment viewers. Object data and object states are stored in the database defined via the AR marker's URI.
- Ubiquity of Object Interactivity and Definition of Object Interaction Models:
  Objects must behave the same way independent of AR environment viewer. Object's interaction definition are stored in the database defined via the AR marker's URI. Interaction models with AR systems have so far been system/application dependent. Each system defines for itself how users interact with the AR objects and the interaction model cannot be extended or redefined.

Our framework allows developers to define how a specific AR Object will interact with the users and with other AR Objects introduced to the scene. They do so by assigning behaviours to extra control markers associated with the AR Object via the AR Object's URI.

4. System design, software components and implementation

All of the below mentioned libraries are distributed under licenses allowing developers to use them free of charge for non-commercial purposes. Our framework is built using Flash Builder 4 and ActionScript 3.5 SDK. Developers can produce their modules using any ActionScript compiler as long as they run the same SDK and use the same versions for PaperVision3D, FLARToolKit, and FLARManager.

4.1. Software components

Our system is based on several existing technologies that allow us to perform AR overlay, QR decoding, marker recognition, tracking and handling and draw our interfaces programmatically. In this section we will look at each one in more detail.

- AR Overlay and Interface Design:
  The original AR toolkit was first developed by Dr. Hirokazu Kato from the University of Washington [4] and is currently supported by the Human Interface Technology Lab at the University of

Canterbury in New Zealand [2]. As we are building our framework in Adobe ActionScript programming language, we are using a language port of the ARToolKit to AS3 provided by Saqoosha [12], Nyatla [10] and Sparklib [7] named FLARToolKit. To design, draw and define our interfaces we use PaperVision3D library provided by [15]. PaperVision3D is a set of libraries that give ActionScript developers a 3D engine for Flash.

- QR Decoding and Marker Handling:
  For decoding QR codes in ActionScript we use the QR library provided by Sparklib [7]. To manage marker registration efficiently for multiple markers and predict marker motion we use the FLARManager 0.7 toolkit which is provided by Eric Socolofsky [14].

4.2. System model

- AR Marker with QR:
  We designed our AR markers to include QR codes encoding the Unique Resource Identifiers for the object that the AR marker identifies. This allows the developer to define his own AR marker patterns and objects independent of the viewer. It also allows the AR environment viewer to recognize AR markers without the need to include the patterns in the program. The QR code can be placed either inside of the AR marker as part of the pattern or on the back of the AR marker. Note that if the QR code becomes a part of the AR marker's pattern it must do so in an asymmetrical fashion, since AR marker patterns must be asymmetrical to enable correct marker detection.
- Database:
  The database component of the system implements a simple MySQL scheme with database entries containing developer information and pointing to a local directory for specific marker id. The physical file is a precompiled Adobe SWF file that contains the AR Object's graphical and interaction definitions.

4.3. System interaction

In order to continue to the next section we must define the control scheme for AR Objects. In the database, each parent AR Marker has associated with it a set of control markers that define a single action. Those control markers are our equivalents of a ”button”. Each control marker is defined by a ”timing” parameter and an ”action” parameter. To initiate the control one must simply introduce the control marker into the scene. The system detects that the control is activated if it is not registered longer than the ”timing” parameter specifies, and performs the action based on the ”action” parameter. To unregister the control marker from the scene one must introduce an ”unregister” pattern as defined per marker (that pattern can simply be printed on the back of the control pattern). We can now describe how our system's user-object and object-object interaction paradigms.

- User-Object Interaction:
  For each marker ecology (defined by the database the system is connecting to) there will be a single marker pattern for a global control marker. The purpose of the global control marker is to select between the active AR Objects on the scene. The detection technique the system uses is as described above. Once an AR Object is active the user can move it around and perform actions as defined by its control markers.
- Object-Object Interaction:
  The last type of interaction we define in our system is object-object interaction. In this paradigm the developers of the AR Objects are allowed to define a ”collision area” defined by an offset to the area of the AR marker. Each marker can either be on the receiving end (marker stationary) or the sending

end (marker moving). Each AR object is associated with both a receiving action and a sending action. One simple example of such object-object interaction is when both AR objects are business cards. When business card A detects the proximity of business card B, the information on B will be attached to A's contact list.

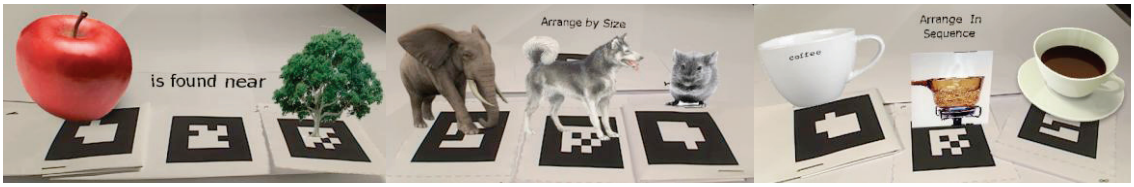

Fig. 1. Examples of common sense collection games (a) visual sentence pattern; (b) arrange by feature (c) arrange in sequence

5. Collecting common sense with UIAR

In this section we would like to present a few possible implementations of games to collect common sense from users. We believe that using our framework can prove useful when it comes to both acquiring new volunteers and keeping the existing volunteers interested and engaged in the process. Using UIAR framework we can implement games that are rewarding, engaging, and interactive. Moreover, those games can be targeted towards younger audiences who naturally spend more time playing games. With UIAR, developers can assign any object to an AR marker (a 3D model of an apple, a word, a sentence, an animation, etc.) which will be immersive (objects will blend in with the actual environment) and interactive (objects will be aware of the surrounding objects). Here we present 3 implementations of games to collect specific types of common sense inspired by such games as Verbosity [1] and Common Consensus [8].

5.1. Visual sentence patterns

One type of exercise commonly used to collect common sense is to simply fill in the gaps in a sentence pattern. While natural language sentences can often prove difficult to process, the use of different sentence patterns allows for collecting data that can be disambiguated, categorized and easily parsed. Our example game uses the following templates: ”X is a kind of Y”, ”X is used for Y”, ”X is typically found near/in/on Y”, ” X is the opposite of Y”, ”X is related to Y”. A user is provided with 3 AR markers , which after being registered via their QR codes will correspond to X, Y, and the sentence pattern respectively. As the visual content of AR markers is dependent on the QR code only, the game can be setup so that every time the markers representing X or Y is re-reregistered, the content is changed. For example, on first registry marker X can a 3D model of an apple and Y a 3D model of a tree. If the sentence pattern does not make sense, the user can re-register it to get the correct one (in this case, ”X is typically found near/in/on Y”). To submit the entry, the user needs to arrange all three markers so that they touch each other. After submission, the game refreshes the markers with new objects and sentence patterns.

5.2. Object ranking: arrange by feature

The second type of game is for spatially oriented knowledge collection. In this case the number of markers/objects can as many as the screen can allow. A sample exercise of this game would be to ask the user to arrange the markers in a certain order based on a certain criteria (high, length, size, etc.). As the markers are spatially aware of each other, the user will complete the exercise by putting the markers close to each other in the order needed. For example, the user can have 4 markers. After registering each marker, he is presented with 4 different 3D (or 2D) objects which he/she will arrange by a predefined feature and submit to the system.

5.3. Object ranking: arrange in sequence

The third type of game is goal oriented. Just like the ”arrange by feature” game, in this game the user will be presented with two markers representing the beginning and the end of an activity, with the rest of the markers representing actions that must fit in a sequence. For example, the user can start with 2 markers, one showing - the rising sun (or a person coming out of bed) and another a steaming cup of coffee. The rest of the markers could represent ”boiling water”, ”mixing water in cup”, ”opening coffee”, ”pouring sugar”. The user will complete the exercise by arranging the markers/activities in the right order. Examples of all three game types can be seen in Figure 1. In these sample implementations each object/activity can be represented either by a visual (a 2D or 3D model) or just text (the text being overlaid over the AR marker). In order to represent both simple physical objects and abstract concepts with more visual appeal it is better to use images or models. There is a vast collection of images on the Internet available under a Creative Commons License and which can be mined to populate a large database of objects. The example implementations given are specific to the realm of common sense acquisition. However, the system can be generalized to serve any number of language acquisition tasks.

6. Conclusions and future work

In our research we are trying to address the need for enriching textual knowledge with interaction driven knowledge acquisition. We are currently in the process of deploying a prototype version of the UIAR framework as an open source project. We are performing evaluations on the common sense knowledge collection games as part of an evaluation process of the framework as a whole, after which we plan to implement an AR viewer for mobile devices using an HTC developer device running Android 2.3 OS. We are also planning a collaboration with the Japanese version of the Open Mind Common Sense database, as it is still very small compared to its English version. We plan to implement additional methods for user-object interaction. To improve the overall usability of the system, we would like to implement marker-less fingertip detection in the future.

Acknowledgements

The authors gratefully acknowledge support from Hokkaido University Global Centre of Excellence program which has made this research possible.

References

[1] Ahn L. V, Kedia M, Blum M. Verbosity: a game for collecting common-sense facts. In *In Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems, volume 1 of Games*, pages 75–78. ACM Press, 2006.

[2] Billinghurst M. Artoolkit. http://www.hitlabnz.org, June 2009.

[3] Do-Lenh S, Kaplan F, Sharma A, Dillenbourg P. Multi-finger interactions with papers on augmented tabletops. *TEI2009*, p. 16–18, 2009. The 3rd InternationalConference on Tangible and Embedded Interaction, Cambridge,UK.

[4] A. Inc. Artoolkit. http://www.hitl.washington.edu/artoolkit/, June 2009.

[5] Lee B, Chun J. Manipulation of virtual objects in markerless ar system by fingertip tracking and hand gesture recognition. *ICCIT'09*, 2009. Seoul.

[6] Lenat D. Cyc: Towards programs with common sense. *Communications of the ACM*, pages 30–49, 1990.

[7] S. P. A. C. Library. http://www.libspark.org/, June 2009.

[8] Lieberman H. Common consensus: a webbased game for collecting commonsense goals. paper presented at iuif07, 2007.

[9] Mistry P, Maes P, Chang L. Wuw - wear ur world - a wearable gestural interface. *CHI'09*, 2009. In the CHI '09 extended abstracts on Human factors in computing systems, Boston, MA.

[10] Nyatla. Nyartoolkit for as3. http://d.hatena.ne.jp/nyatla/, June 2009.

[11] Rekimoto J. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. *SIGCHI*, pages 113–120, 2002. Conference on Human Factors in Computing Systems.

[12] Saqoosha. Flartoolkit. http://saqoosha.net/, June 2009.

[13] Singh P. The public acquisition of common sense knowledge. In *Proceedings of the AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, 2002.

[14] Socolofsky E. Flarmanager. http://words.transmote.com/wp/flarmanager/, December 2009.

[15] Ulloa C, Grden J, Hauwert R, Knip T, Zupko A. Papervision3D: 3D engine environment for Adobe Flash. http://www.papervision3d.org/, June 2009.