Interactive augmented reality games for knowledge acquisition

Svetoslav Dankov, Rafal Rzepka, Kenji Araki
Language Media Laboratory
Graduate School of Information Science and Technology, Hokkaido University
{dankov,kabura,araki}@media.eng.hokudai.ac.jp

Abstract

In this paper we present several games aimed at acquiring common sense knowledge from users. Common sense is a special type of knowledge that has proven significantly hard to extract automatically. 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. The games are aimed at acquiring common sense from users and use natural interaction with augmented reality objects. We have designed and implemented them through UIAR - a "User Interface through Augmented Reality" framework developed in our laboratory.

Keywords: augmented reality based games, knowledge acquisition, human-computer interaction, common sense

1. Introduction

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 our research we implement an Augmented Reality system (UIAR) which serves as an extension to existing computer interfaces, provides enhanced user-experience, and defines virtual objects and their actions in an ubiquitous way.

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.

We begin by presenting a summary of the related research, connecting our approach with previous work. Next we desribe the purpose and present specific details about the framework's functionality. We conclude by presenting the implementation scenarios for games collecting common sense knowledge from users using our framework.

2. Related Reasearch

2.1. Augmented Reality, Interfaces, 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 [12] are very robust but are often limited to detecting only "touch" behavior, 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 [6] 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 [10]. As is the purpose of [10], 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 [7].

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 [14]. In the course of few years the project had already collected over 1.6 mil-

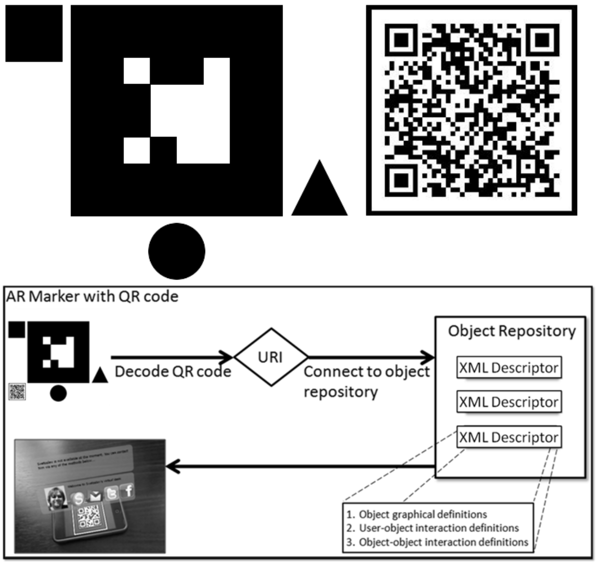

Figure 1. Marker design and system overview

lion statements.
Other systems, like Verbosity [1] and Common consensus [9] 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.
Yen-Ling Kuo et al. [5] have successfully utilized social games to collect common sense and their findings provide useful suggestions for designing community-based games.

3. System Design, software components and implementation

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 (UIAR) 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 behaviors to the markers associated with the AR Object via the AR Object's URI. The URI stores an address for the server the AR object is registered to. This allows us to provide ubiquity of object presence, object persistence, and ubiquity of object interactivity; this in turn makes the framework highly adaptable to developer's needs.

3.1. 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**

Figure 2. Example of the sentence-pattern game

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 [13], Nyatla [11] and Sparklib [8] named FLARToolKit. To design, draw and define our interfaces we use PaperVision3D library provided by [16]. 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 [8].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 [15].

3.2. System model

Figure 1 gives an overview of the system and an example of our modified markers.

- **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

Figure 3. Example of the arrange-by-feature game

Figure 4. Example of the arrange-by-sequence game

specific marker id. The physical file is an XML descriptor that contains definitions of the AR Object's graphical and interaction definitions.

3.3. Object Interaction

We implement two types of object interactivity - object-to-object and user-to-object. The actions are defined on a per-marker basis and are stored server-side.

- **User-Object Interaction**
  User-object interaction is implemented through control-blobs on the markers as seen in Figure 1. The system can detect and distinguish between the blobs. The user activates each blob by covering it with a finger for a predefined amount of time. The action that the blob triggers is stored in the database for each individual marker. Furthermore, the actions are positionally dependent on the marker's surroundings - those actions can be directed/performed to the AR object of a different marker if that marker is close by.
- **Object-Object Interaction**
  Object-object interaction is implemented through positional dependency of the markers in the scene. The system can track the position and distance between markers which allows us to activate specific actions on the AR objects whenever they are in proximity to other objects. By detecting which marker is moving and which is stationary we implement a sender-receiver scheme where the stationary marker "receives" the action, while the moving marker "initiates" it.

4. Common sense acquisition games

In this section we would like to present three 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 the 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 virtual object to an AR marker (3D models, textual and media content, etc.) which will be immersive (objects will blend in with the actual environment) and interactive (objects will be aware of the surrounding objects and react to users' input). Here we present 3 implementations of games to collect specific types of common sense inspired by such games as Verbosity [1] and Common Consensus [9]. Images for those games are acquired through Google Image Search. We use Wordnet to choose related concepts when needed.

4.1. Visual sentence pattern game

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 templates such as: "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", etc. A user is provided with AR markers, which after being registered via their QR codes will correspond to X, Y, and a description of the template 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 hold 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 until he gets 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 and the user can keep playing.

4.2. Arrange by feature game

The second type of game is for spatially oriented knowledge collection. In this case the number of markers/objects can be 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 (height, 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. The same game can be used to cluster individual objects in case there is more than one feature. For example, the user can be presented with representatives of fruits and animals, in which case he can group the markers together and choose their categorization.

4.3. Arrange in sequence game

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 figures **??**. 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.

5. 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, the quality of the collected data and how our system affects user retention.
We plan to implement an AR viewer for mobile devices using an HTC developer device running Android 2.3 OS. We plan to implement additional methods for user-object interaction, improve the overall usability of the system, and implement security schemes for the AR objects.

References

[1] L. V. Ahn, M. Kedia, and M. Blum. 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] M. Billinghurst. Artoolkit. `http://www.hitlabnz.org`, June 2009.

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

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

[5] Y. Kuo, K. Chiang, C. Chan, J. Lee, R. Wang, E. Shen, and J. Y. Hsu. Community-based game design: Experiments on social games for commonsense data collection. *HCOMP2009*, 2009. KDD-09 Workshop on Human Computation.

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

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

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

[9] H. Lieberman. Common consensus: a web-based game for collecting commonsense goals. paper presented at iuif07, 2007.

[10] P. Mistry, P. Maes, and L. Chang. 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.

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

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

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

[14] P. Singh. 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.

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

[16] C. Ulloa, J. Grden, R. Hauwert, T. Knip, and A. Zupko. Papervision3d: 3d engine environment for adobe flash. `http://www.papervision3d.org/`, June 2009.