Design of an Augmented Reality-Enabled Multimedia File Format with Embedded Object Tracking Metadata

Byoung-Dai Lee

Department of Computer Science, Kyonggi University, Suwon, Korea

blee@kgu.ac.kr

Abstract

Feature extraction and tracking is one of the core Augmented Reality (AR) technologies. A markerless AR provides natural synthesis of real-world and virtual objects as it is able to identify objects directly within a video and obtain relevant information. However, disadvantage of the markerless AR is that it may not be appropriate for resource-constrained devices, such as mobile phones, due to the considerable amount of computations necessary. In this paper, we propose a method to address this problem by putting into multimedia content the metadata necessary to provide AR services, such as virtual object information and supplementary information required for on-screen display. In order to show the feasibility of our approach, we extended the ISO base media file format that various well-known media formats such as MP4 are based on.

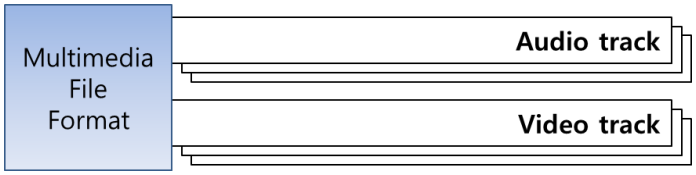
Keywords: *Augmented Reality, Feature Extraction and Tracking, ISO Base Media File, Metadata*

1. Introduction

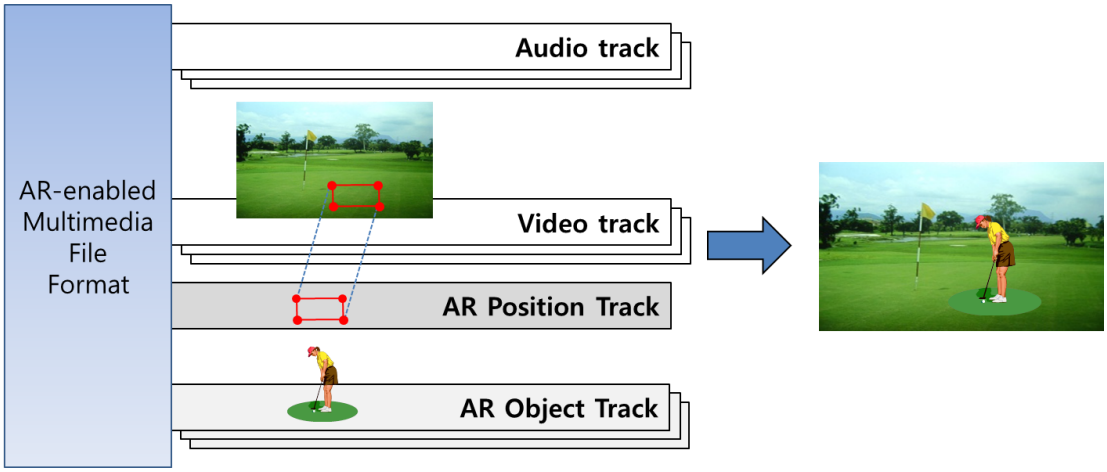
Augmented Reality (AR) is a technology that provides augmented information services by synthesizing real-time images/voices and virtual objects or supplementary information. Recently, mobile devices with various built-in sensors, such as cameras and global positioning system (GPS), have been widely distributed, presenting diverse convergence services that use high-speed mobile Internet and mobile AR services.

Most of the existing AR services use real-time image recognition results to provide virtual objects or supplementary information. For example, a played video is analyzed in real-time and the area where the virtual object will be rendered is identified. However, technology that extracts features by accurately recognizing the object within the image on a real-time basis requires the considerable amount of computations, which indicates that the quality of AR services depends on the complexity of feature extraction algorithms as well as the resource capability of the device. Along with the difficulty of extracting features, another weakness in existing AR services is that AR application programs and the augmented information shown to the users are tightly coupled. For instance, in the case of a service that shows a corporate logo image in the middle of a video to promote a product, when the logo will be presented to the users is determined by the logic of the application program. Thus, there is a possibility that a corporate logo unrelated to the actual video content that is being played appears on screen, resulting in decreased advertising effectiveness.

AR in multimedia services based on stored media is different from AR based on general real-time videos in that editing of multimedia content by the service provider can be preceded. In this paper, we aim to solve these problems by putting into the multimedia content the metadata necessary to provide AR services, such as virtual object information and



(a) Logical structure of typical media files



(b) Logical structure of AR-enabled media files

Figure 1. Logical Structure of the proposed Media File with Embedded Metadata for AR Services

supplementary information required for on-screen display. In particular, we propose a method to construct media files as the container that includes only the metadata to display AR content on screen so that the AR service is not tightly coupled to certain AR technologies. Constructing media files by putting in metadata, as we propose here, offers the following advantages: (1) Complicated processing is not required to extract features in a receiving device, thereby enabling easy use on a mobile device, which is typically resource constrained. (2) Deterioration of multimedia content can be prevented by determining in advance the most suitable location from each scene of the video in which the virtual object will be displayed. (3) The image and virtual object are not tightly coupled mutually and can provide the AR services most appropriate for the user context (*e.g.*, user location, performance of device).

The remainder of this paper is organized as follows: Section 2 summarizes related work, Section 3 explains the proposed method in detail, Section 4 presents an implementation based on ISO media file format, and Section 5 presents the conclusions.

2. Related Work

Significant research has been conducted to address efficient and effective tracking of objects so they can be used for resource-constrained devices, such as mobile phones. Klein *et al.*, [1] proposed Parallel Tracking And Mapping (PTAM) technology in which real-time tracking and mapping in a small working space is possible by using a single camera, and this technology was implemented in 2009 to a Simultaneous Location And Mapping (SLAM) system that is operated in real time in iPhones. Lee *et al.*, [2] proposed the hybrid tag method that converged the advantages of marker and markerless AR to efficiently recognize and track images on mobile devices. The markers contain information, such as the location and size of

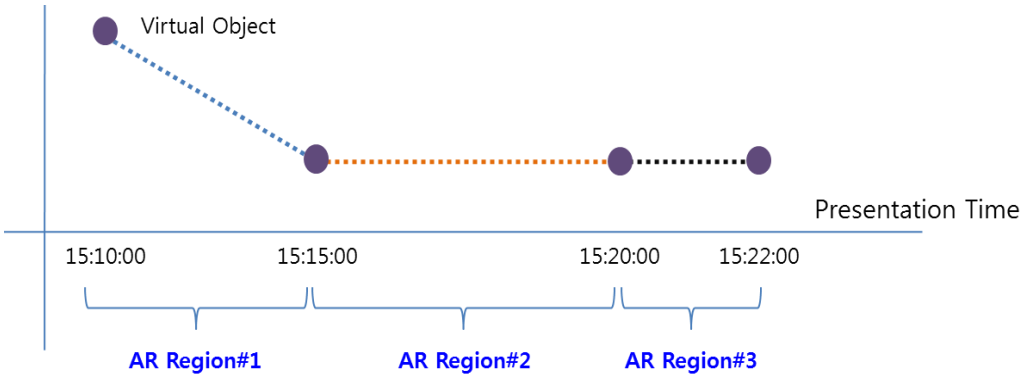


Figure 2. An Example of the AR Position Track

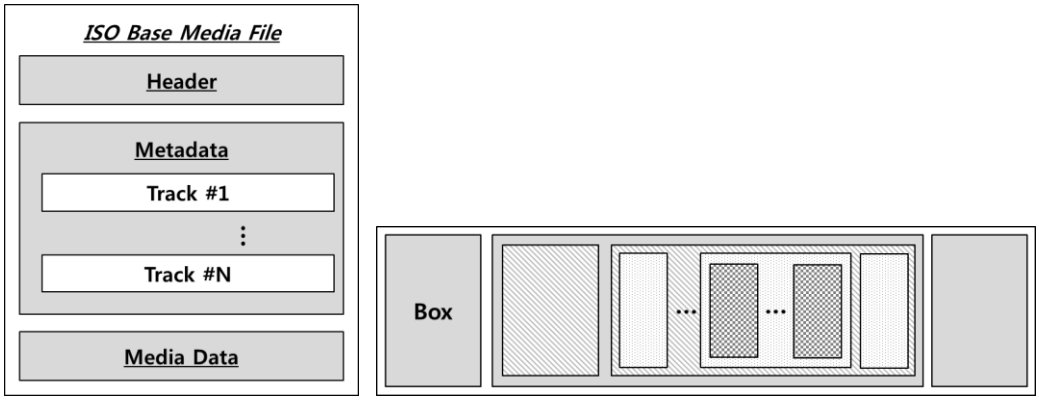
the object to be tracked. Wagner *et al.*, [3] proposed a markerless tracking technique that revised Scale-Invariant Feature Transform (SIFT) to reduce memory use and provide rapid processing for the mobile environment. ARhrrr! [7] is the first mobile AR game providing content at the level of commercial games. It is implemented so that all processing except tracking is operated in the Graphic Processing Unit (GPU), providing real-time AR game services on a mobile device while offering high-quality content.

3. Metadata for Augmented Reality

Figure 1 shows the simplified version of the logical structure of the media files. As shown in Figure 1a, typical media files include the Audio track and the Video track, each of which represents a time sequence of media components (*e.g.*, frames of video). The proposed AR-enabled media files include two additional tracks that are responsible for enabling AR services – the *AR Position track*, storing the location information of the virtual object provided in AR services, and the *AR Object track*, storing the actual virtual object information to be displayed at the relevant location (see Figure 1b).

The main role of the AR Position track is to save location information of the space that displays the virtual object within the image. In particular, for natural synthesis between the image and virtual object, the virtual object must sequentially move with the image based on the time scale when the image is played. For this study, we defined the AR region on a time scale based on the rate of movement of the virtual object (see Figure 2). As the virtual object can rotate or change its size, the AR region is further refined by the rate of the rotation and the scaling of the virtual object.

As shown in Figure 2, the virtual object is displayed on screen from 15:10 to 15:22 and moves from the upper-left to lower-right corner of the screen. In AR Region #1 (15:10:00–15:15:00), it moves from the upper-left to the lower-right corner at a constant rate (*e.g.*, α m/sec); in AR Region #2 (15:15:00–15:20:00), it moves from the lower-left to the lower-right corner at the same rate as in AR Region #1. However, in AR Region #3 (15:20:00–15:22:00), the traffic line of the object is the same as that of AR Region #2 (that is, from the lower left to the lower right) but the rate of movement is different (*e.g.*, β m/sec); thus, it is defined as a different AR region, and the separate location information of the object is to be saved.



(a) Logical structure (b) Physical elements

Figure 3. The Structure of ISO Base Media File Format

The AR Object track describes the actual virtual object to be displayed at the location within the image clarified in the AR Position track. Multiple AR Object tracks can exist in the media files to support various AR contents, and the AR Object track most suitable for user conditions is selected by using reference data on the virtual object included in the AR Position track. Another characteristic of the AR Object track is that it provides neutrality in certain representation techniques of virtual objects. Virtual objects used in AR can be represented using various techniques; however, the AR Object track is not tightly coupled with certain representation technique but plays the role of a container regardless of the internal representation, thereby providing neutrality.

4. An Implementation

4.1. ISO Base Media File Format

To verify the effectiveness of the aforementioned method for providing AR services using metadata, we extended the existing ISO base media format [5]. The ISO base media file format was specified as ISO/IEC 14496-12 (MPEG-4 Part 12) and defines the general structure for time-based multimedia files, such as audio and video, that facilitates interchange, management, editing, and presentation of the media [5]. As shown in Figure 3(a), the ISO base media file format consists of three logical components: header, metadata, and media data. The header contains general information for the media contained in the file, such as content identifier, content provider, and content creation date. The metadata contains information about individual media components (*e.g.*, audio and video). In particular, individual tracks in the metadata represent timed sequences of corresponding media components, and important information contained in the track includes the profile information required for media decoding, placement information of media samples in the media data component, and their timing information in presentation. Finally, the media data contain the actual coded media data. Note that the media data component may be in the same file or in other files.

Files conforming to the ISO base media file format are formed as a series of objects, called *boxes*, which are defined by a unique type identifier and length. All data are contained in the boxes, and there is no other data within the file (see Figure 3(b)). For example, individual tracks in Figure 3(a) are represented by the track boxes. As a container box, the track box only contains several sub-boxes for storing the track header information, the layout of the

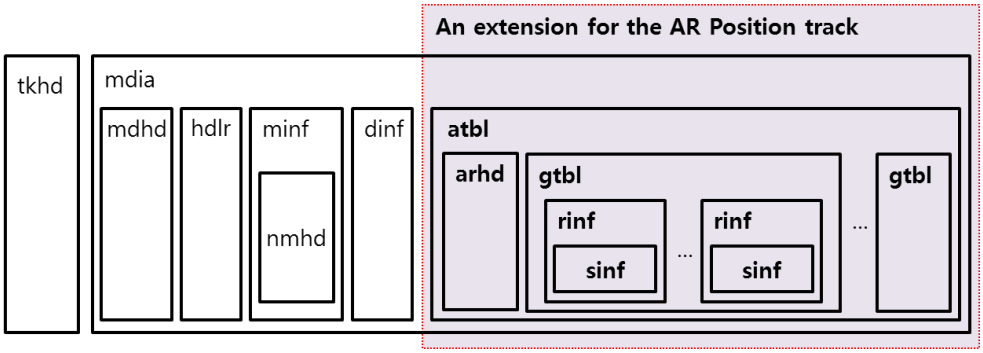


Figure 4. Box Structure of the AR Position Track

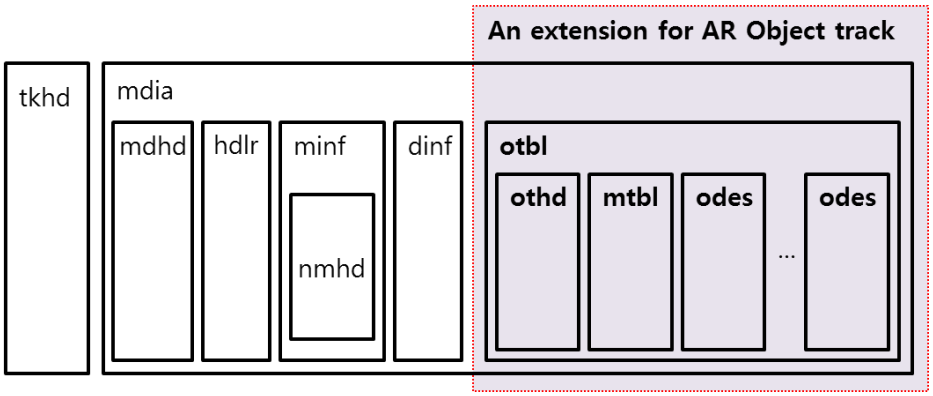


Figure 5. Box Structure of the AR Object Track

media data represented by the track, and the time ordering of the media. The sub-boxes, in turn, may contain their own sub-boxes, if necessary.

4.2. AR-Enabled ISO Base Media File

The AR Position track and the AR Object track exist as track boxes of ISO base media files, and Figure 4 and 5 shows the track box structure of an expanded ISO base media file. Refer to [5] for the detailed structure and syntax of track boxes of ISO base media files. Table 1 shows the detailed syntax and semantics of the main boxes included in the AR Position track. The AR Table (atbl) box is a container box made up of a single AR Header (arhd) box and multiple Group Table (gtbl) boxes. The arhd box saves the values necessary to generate an ID assigned in AR regions and the number of AR regions included in atbl. The gtbl box saves detailed data of individual AR regions in which gtbl exists in individual virtual objects. If the same virtual object appears in various parts of the image, there are multiple Region Information (rinf) boxes in the gtbl box, and the rinf boxes save the location information where the virtual object will appear. The Region (regi) box specifies the rectangular area within which the virtual object will actually appear. As the AR region represents the straight line on which the virtual object moves at an equal speed, it can be defined by the start and end points of the line, each of which is represented by the start_region and end_region fields in the rinf box, respectively. Note that the sizes of the rectangles specified by the start_region and the end_region fields are not necessarily the same. If the sizes differ, it indicates that the virtual object scales up or down while moving. Furthermore, as the virtual object can rotate

while moving, the regi box also contains the rotation information. Therefore, at each time point within the AR region, the degree of rotation and the scaling of the virtual object can be computed by linear interpolation using the information in the start_region and the end_region fields. The Sample Information (sinf) box associates virtual objects and the media samples on which the virtual objects will be superimposed. Therefore, the box plays the role of synchronizing between virtual objects and audiovisual content on a time basis.

Table 1. Detailed Syntax for the AR Position Track

Box	Syntax

atbl
(AR Table)
	aligned (8) class ARTable extends Box('atbl')
	{
	unsigned int (32) num_of_gtbl;
	ARHeader();
	for (int i = 0; i < num_of_gtbl; ++i)
	GroupTable();
	}

arhd
(AR Header)
	aligned (8) class ARHeader extends Box('arhd')
	{
	unsigned int (32) next_group_ID;
	unsigned int (32) group_entry_count;
	}

gtbl
(Group Table)
	aligned (8) class GroupTable extends
	Box('gtbl') {
	unsigned int (32) group_ID;
	unsigned int (32) next_region_ID;
	unsigned int (32) region_entry_count;
	for (int i = 0; i <_region_entry_count; ++i)
	RegionInformation();
	}

rinf
(Region Information)
	aligned (8) class RegionInfomation extends
	Box('rinf') {
	unsigned int (32) region_ID;
	Region start_region;
	Region end_region;
	unsigned int (32) velocity;
	SampleInformation();
	}

regi
(Region)
	aligned (8) class Region extends Box('regi') {
	unsigned int (32) x1, y1;
	unsigned int (32) x2, y2;
	unsigned int (32) x_rotation;
	unsigned int (32) y_rotation;
	unsigned int (32) z_rotation;
	}

sinf
(Sample Information)
	aligned (8) class SampleInformation extends
	Box('sinf') {
	unsigned int (32) frist_sample;
	unsigned int (32) sample_count;
	}

Table 2 shows the detailed syntax and semantics of the main boxes included in the AR Object track. The AR Object track is composed of the Object Table (otbl) box to display the

virtual object used in each AR region. The otbl box is a container box made up of a single Object Table (othd) box, a Mapping Table (mtbl) box, and multiple Object Description (odes) boxes. The mtbl box provides mapping for the virtual object used in individual gtbl boxes of the AR Position track, and the odes box contains the representation of the virtual object. In particular, the odes box is a container that supports various AR technologies to display virtual objects in which the virtual object description technology that is actually used is determined by the type field.

Table 2. Detailed Syntax for the AR Object Track

Box	Syntax

otbl
(Object Table)
	aligned (8) class 3DObjectTable extends
	Box('otbl') {
	  unsigned int (32) num_of_odes;
	  ObjectTableHeader();
	  MappingTable();
	  for (int i = 0; i < num_of_odes; ++i)
	  ObjectDescription();
	}

othd
(Object Table Header)
	aligned (8) class ObjectTableHeader extends
	Box('othd') {
	  unsigned int (32) next_object_ID;
	  unsigned int (32) object_entry_count;
	}

mtbl
(Mapping Table)
	aligned (8) class MappingTable extends
	Box('mtbl') {
	  unsigned int (32) mapping_entry_count;
	  for (int i = 0; i < mapping_entry_count; ++i)
	{
	  unsigned int (32) group_ID;
	  unsigned int (32) object_ID;
	  }
	}

odes
(Object Description)
	aligned (8) class ObjectDescription extends
	Box('odes') {
	  unsigned int (32) object_ID;
	  unsigned int (32) type;
	  string object_desc;
	}

Figure 6 and 7 shows an example of AR service and the contents for the AR Position track and the AR Object track to enable the service. In this example, three AR regions were identified, meaning that the virtual object moves, rotates, and scales up/down at constant rates within each AR region. AR Region #1 and AR Region #3 use the same virtual object (*e.g.*, the smiling face) and, therefore, there exist two rinf boxes in the gtbl box. Note that there are two odes boxes with object_ID equal to one. This indicates that the virtual objects described by the two boxes are the same but the description technologies differ. In the example, it is assumed that the smiling face is described either by Virtual Reality Markup Language or the 3D XML. The user device determines which object to use based on, for example, the availability to the software components to interpret the corresponding object description languages.

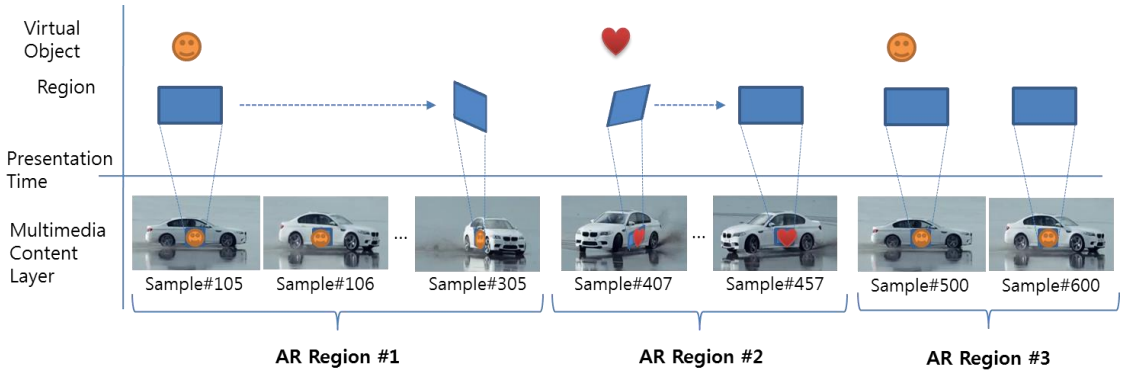
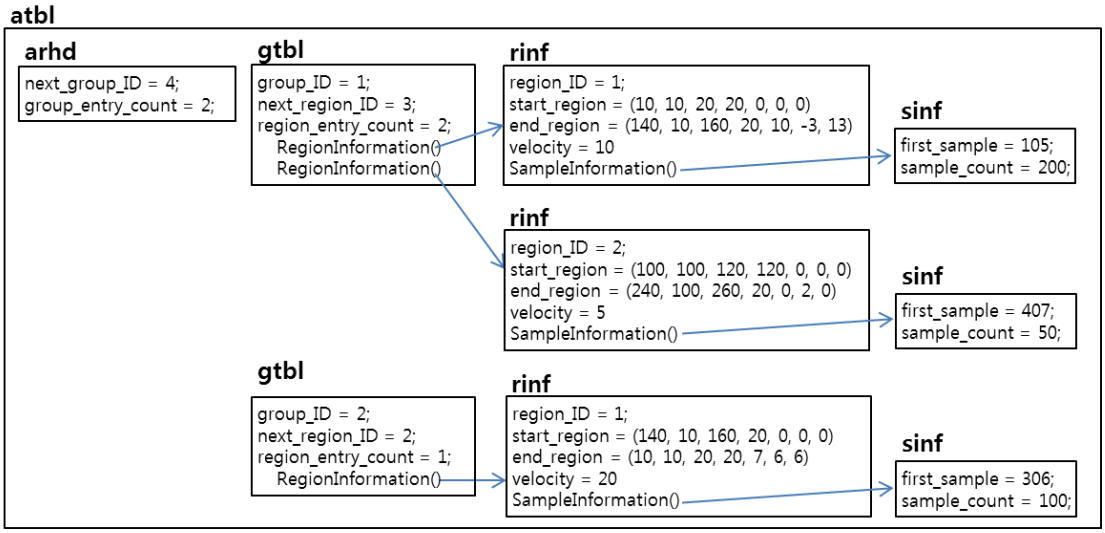
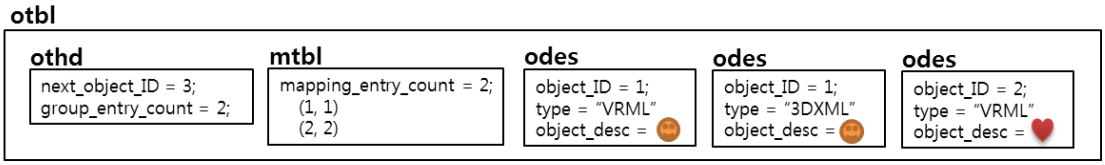


Figure 6. An Example an AR Scenario



(a) An example of the AR Position track



(a) An example of the AR Object track

Figure 7. Examples of the AR Position Track and the AR Object Track

5. Conclusions

Feature extraction and tracking, which is a core AR technology, is critical in determining the area on screen in which the virtual object is to be displayed; it is one of the most difficult fields of study. Currently, many related technologies exist but they have limitations for use on resource-constrained mobile devices because they generally require significant amount of computations. A large number of computations ultimately consumes many batteries, making such technologies more difficult to apply to mobile

devices. In this paper we proposed a method to construct AR-enabled media files by putting into multimedia content the metadata necessary to provide AR services, such as virtual object information and supplementary information required for on-screen display. We extended the ISO base media file format to apply our approach and we plan to implement the reference software in the future.

Acknowledgements

This work was supported by Kyonggi University Research Grant 2012

References

[1] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspace", Proceedings of the 6th IEEE/ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, (**2007**) November 13-16.
[2] W. Lee and W. Woo, "Real-Time Color Correction for Marker-based Augmented Reality Applications", Proceedings of the 3rd International Workshop on Ubiquitous Virtual Reality, Adelaide, Australia, **(2009)** January 15-18.
[3] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond and D. Schmalstieg, "Pose Tracking from Natural Features on Mobile Phones", Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, Washington D.C., U.S.A., **(2008)** September 15-18.
[4] ETSI 3GPP TS 26.44, Transparent End-to-End Packet Switched Streaming Service (PSS): 3GPP File Format (3GP), The 3rd Generation Partnership Project (**2009**).
[5] ISO/IEC 14496-12:2008(E), Information Technology-Coding of Audio-Visual Objects-Part 12: ISO Base Media File Format, ISO/IEC (**2008**).
[6] ISO/IEC 14496-14, Information Technology-Coding of Audio-Visual Objects-Part 14: MP4 File Format, ISO/IEC (**2003**).
[7] ARhrrr!, http://ael.gatech.edu/lab/research/hanheld-ar/arhrrr/.

Authors

Byoung-Dai Lee is an assistant professor at the department of Kyonggi University, Korea. He received his B.S. and M.S. degrees in Computer Science from Yonsei University, Korea in 1996 and 1998 respectively. He received his Ph.D. degree in Computer Science and Engineering from University of Minnesota, twin cities, U.S.A. in 2003. Before joining the Kyonggi University, he worked at Samsung Electronics, Co., Ltd as a senior engineer from 2003 to 2010. During the period, he has participated in many commercialization projects related to mobile broadcast systems. His research interests include mobile rich media, augmented reality, and mobile multimedia broadcast.