E-Training Content Delivery Networking System for Augmented Reality Car Maintenance Training Application

Yu-Doo Kim and Il-Young Moon

Korea University of Technology and Education
kydman@koreatech.ac.kr

Abstract

The widespread use of smart devices such as smartphones and tablet PCs has contributed to the development of a variety of content for such devices. In particular, smart devices can be exploited for educational purposes. Therefore, we implement an e-training mobile application using augmented reality (AR). However, existing AR applications do not support real-time content. Therefore, it is necessary to manually update an application when the content is modified. In this study, we implement a content delivery network system for AR e-training mobile applications, which supports real-time content update when the content is modified.

Keywords: *CDN, Network, AR, e-Training, Mobile*

1. Introduction

The proliferation of mobile devices such as smartphones and tablet PCs has contributed to the development of various services based on mobile networks. However, existing content for smart devices is not sufficiently versatile; it primarily constitutes games and messaging services. Therefore, in a previous study [1], we described the development of mobile devices and networks using augmented reality (AR) technologies for a virtual education system. Existing AR e-training systems are based on only non-network services. Hence, it is difficult to modify such systems with new content. Therefore, in this study, we investigate an AR e-training system that supports real-time updating using the content delivery network (CDN) technique.

A CDN is a large distributed system of servers deployed at multiple data centers in the Internet. The objective of a CDN is to serve content to end users with high availability and high performance [2]. CDNs currently serve a large fraction of Internet content, including web objects (text, graphics, URLs, scripts), downloadable objects (media files, software, documents), applications (e-commerce, portals), live streaming media, on-demand streaming media, and social networks. Therefore, AR systems based on CDNs have been investigated in previous studies [3, 4].

In this paper, we implement a CDN system for an AR e-training mobile application, and we design protocols for fast content delivery.

The remainder of this paper is organized as follows. In Section 2, we describe existing AR e-training educational systems. In Section 3, we present the system architecture of a real-time updating system, and in Section 4, we describe the implementation of this system. Finally, we conclude this paper in Section 5.

2. AR E-Training System

E-training focuses on experience and practice to facilitate active learning and impart knowledge on the basis of existing theory-based education. The most typical hands-on training system is an augmented reality system [5, 6].

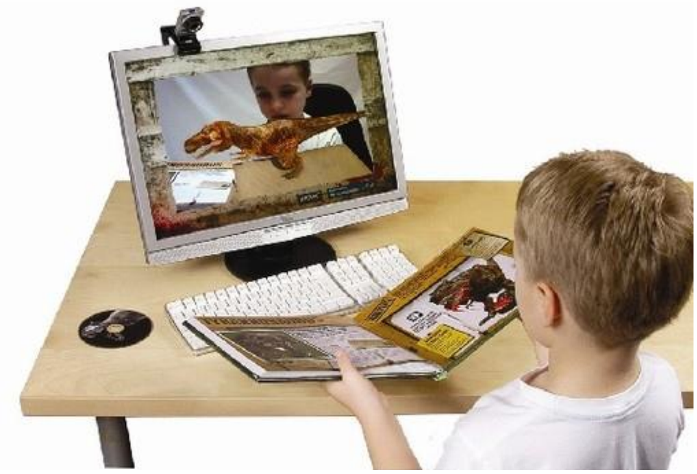
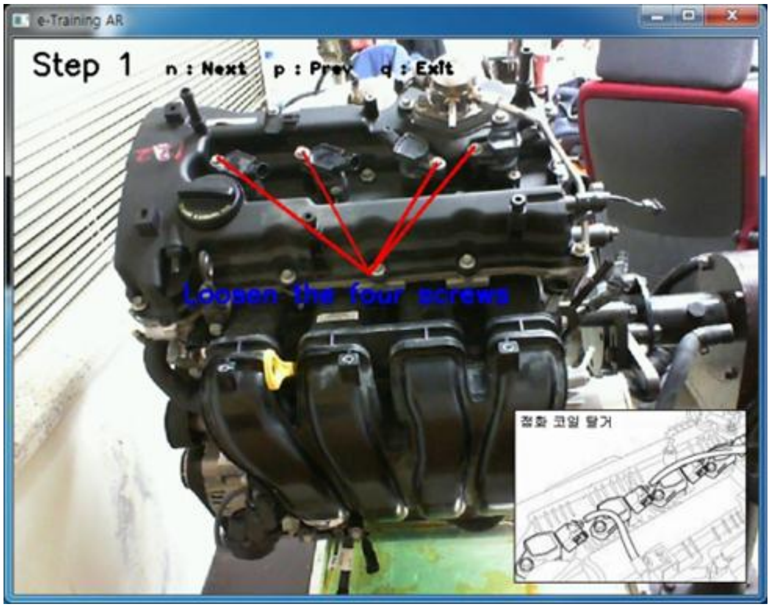


Figure 1. Example of Virtual Training using AR

Figure 1 shows an AR educational application for children. If the camera is focused on the book, multimedia objects can be displayed on the monitor. Hence, the child can enjoy learning with multimedia content rather than static objects in the book.



Step 1

Loosen the four screws

Figure 2. E-training AR System for Car Maintenance

Figure 2 shows a car maintenance system using AR. A camera and mobile computer are used to project multimedia objects on an image of a real engine. This setup can guide mechanics in car repair tasks, thereby facilitating car maintenance.

Current AR educational systems are not supported by network services. Hence, such systems require software reconfiguration for content modification. We believe that a CDN-based AR educational system can facilitate content modification without software reconfiguration.

3. System Design

As shown in Figure 3, the system architecture consists of the content server, content modification system, and content player. We consider the case of an AR system for car maintenance.

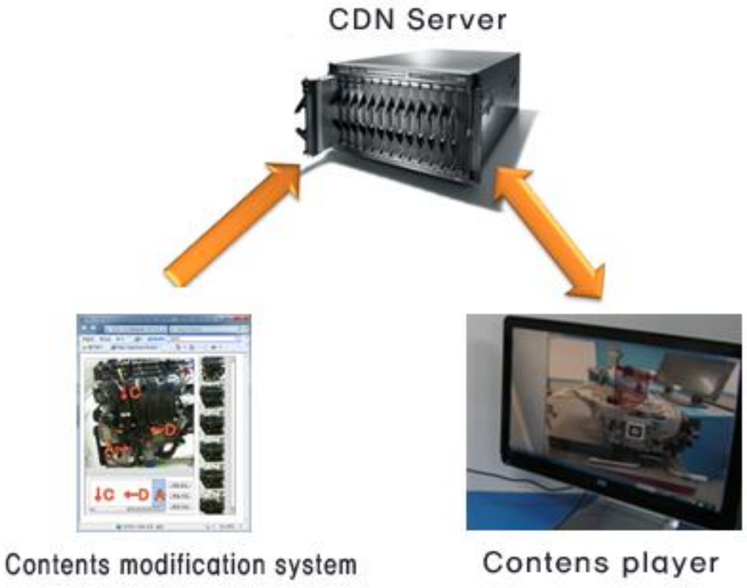


Figure 3. System Architecture

Table 1. System Configuration

System	Language	Role
CDN Server	Apache Node.js	Contents Delivery Store meta-data
Contents Modification	HTML5 Java Script	Contents modify Sending updating message
Contents Player	Visual C++ HTML5	Show AR contents Contents updating

3.1. Content Server

The AR content of mobile devices does not include local systems. Such content is stored in the CDN server and transmitted in real time.

The CDN server stores meta-content that is including engine information and guide objects. If the content player requests engine guide information, the CDN server sends the guide information to the content player in real time.

3.2. Content Modification System

Content developers use the content modification system to create guide objects. This system enables the creation and modification of engine guide objects. The latest guide information will always be available to users, even if it is modified by content developers. Moreover, content developers need not notify users of changes in the content.

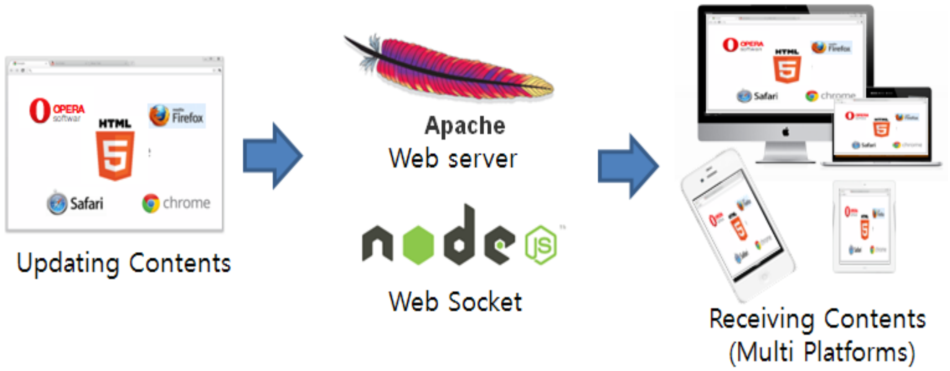
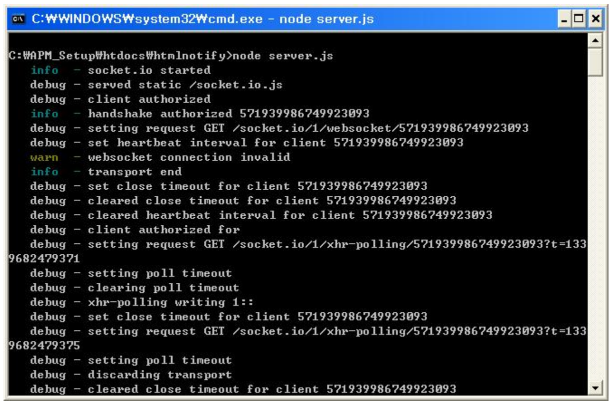


Figure 4. Contents Update Procedure

It is separated by three parts, first is update site. Contents providers are connecting web site that is update contents and send notify to web socket server. After that, web socket server send contents updating message with contents address to connected clients. Finally clients who connecting contents update client web site receive contents updating message and contents URL (Uniform Resource Locator). So clients receive latest e-Training contents via web server.

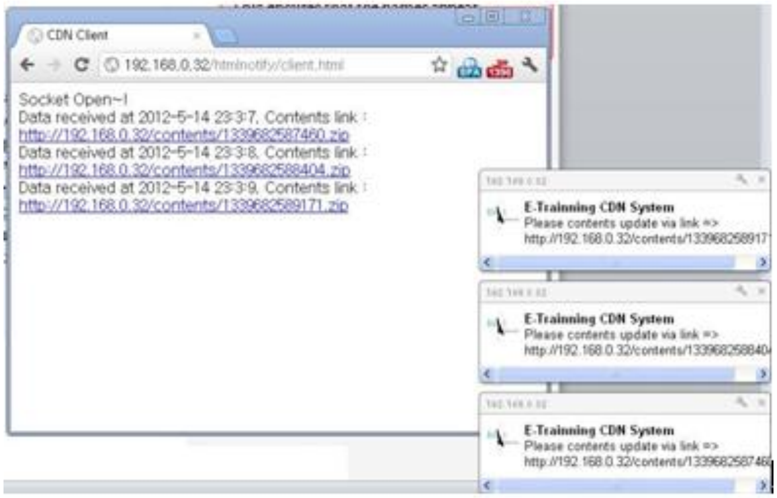
3.3. Notification of Contents Updating

When the contents are updating, contents modification system send a updating message to notification server. After that, server is sending a updating message to client systems. Through this procedure, contents players maintain latest contents in real-time. Web socket server developed by node.js language. Web socket receive contents updating message from contents provider page of web server and send a contents updating message to client web page. The server web page provides contents upload capability. So contents developer updates AR contents on the server web page. Client web page shows updating message and contents URL from web socket server.



```
C:₩APM_Setup₩htdocs₩htmlnotify>node server.js
   info  - socket.io started
   debug - served static /socket.io.js
   debug - client authorized
   info  - handshake authorized 571939986749923093
   debug - setting request GET /socket.io/1/websocket/571939986749923093
   debug - set heartbeat interval for client 571939986749923093
   warn  - websocket connection invalid
   info  - transport end
   debug - set close timeout for client 571939986749923093
   debug - cleared close timeout for client 571939986749923093
   debug - cleared heartbeat interval for client 571939986749923093
   debug - client authorized for
   debug - setting request GET /socket.io/1/xhr-polling/571939986749923093?t=133
9682479371
   debug - setting poll timeout
   debug - clearing poll timeout
   debug - xhr-polling writing 1::
   debug - set close timeout for client 571939986749923093
   debug - setting request GET /socket.io/1/xhr-polling/571939986749923093?t=133
9682479375
   debug - setting poll timeout
   debug - discarding transport
   debug - cleared close timeout for client 571939986749923093
```

Figure 5. Socket and Contents Update Server



Socket Open~!
Data received at 2012-5-14 23:3:7, Contents link :
http://192.168.0.32/contents/1339682587460.zip
Data received at 2012-5-14 23:3:8, Contents link :
http://192.168.0.32/contents/1339682588404.zip
Data received at 2012-5-14 23:3:9, Contents link :
http://192.168.0.32/contents/1339682589171.zip

E-Trainning CDN System
Please contents update via link =>
http://192.168.0.32/contents/133968258917

E-Trainning CDN System
Please contents update via link =>
http://192.168.0.32/contents/133968258840

E-Trainning CDN System
Please contents update via link =>
http://192.168.0.32/contents/133968258746

Figure 6. Running Contents Update in Local System

3.4. Content Player

Table 2. Content Player Operation

Order	Process
1	Runs on mobile device.
2	Recognizes engine using camera of mobile device.
3	Requests guide object from CDN server.
4	Receives guide object from CDN server.
5	Displays guide object over engine on mobile device.

If a user is running the application and focuses the camera on the engine, the content player requests a guide object from the CDN server. Next, it receives the guide object that matches with the part of the engine captured by the camera, and displays this object on the mobile device. The operation of the content player is outlined in Table 2.

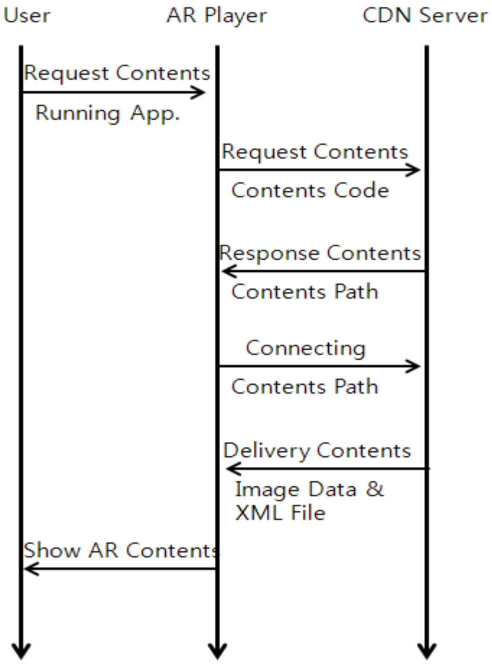


Figure 7. Content Delivery Process

Figure 7 shows the process for transferring content. First, the AR player requests content from the CDN server with content code when the user runs the application. The content code is obtained using the AR engine in the AR player application. If content that matches with the code is found, the CDN server sends a content path to the AR player. Next, the AR player connects to the CDN server through the content path and receives the content, including image data and XML (eXtensible Markup Language) files [7]. XML is widely used for efficiently storing string data [8, 9].

Finally, the AR player displays a configuration with the real view (camera capture) and virtual objects (AR objects). Thus, users can view the AR content.

4. Implementation

4.1. System Summary

The system implementation is summarized in Table 3. First, the CDN server saves AR content in an XML file. Next, the content developer uses a web-based content creation system, which facilitates the creation of guide objects for explaining engine maintenance tasks. Finally, the content player runs on the mobile device. If the user focuses the camera of his/her mobile device on the engine, AR content is displayed over the engine on the screen of the mobile device.

Table 3. System Implementation

System	Role
CDN server	Database. Save engine and guide object information in an XML file.
Content creation & modification	Web-based system. Content developer creates or modifies guide objects.
Content player	Mobile device. Display AR content to user.

4.2. Contents Creation and Modification

Content developers must connect to a web-based content creation system (shown in Figure 8) to create guide objects that can be displayed over the engine. Guide objects can be easily created using drag and drop operations. Similarly, content developers can modify existing content. Then, the content is saved in the CDN server as an XML file. Table 4 shows an example of an XML file.

This XML file consists of one “step” tag and six “aspect” tags, each of which contains many “object” tags. The attribute of “step” is “index.” Similarly, each aspect tag has an “index” attribute. The aspect tags represent images that match with the actual engine; these images are taken from six different angles. The object tags represent the AR guide images, with information regarding the location and image path.

Table 4. Example of XML File

```
<step index=”1”>
  <aspect index="1" img=”back1.jpg”>
    <object x="389" y="265" img="guide1.jpg"/>
    <object x="167" y="320" img="guide2.jpg"/>
    <object x="321" y="344" img="guide3.jpg"/>
    <object x="329" y="146" img="guide4.jpg"/>
  </aspect>
  <aspect index="2" img=”back2.jpg”>
    <object x="389" y="265" img="guide5.jpg"/>
    <object x="329" y="146" img="guide7.jpg"/>
  </aspect>
  <aspect index="3" img=”back3.jpg”>
    <object x="150" y="200" img="guide8.jpg"/>
    <object x="300" y="146" img="guide10.jpg"/>
  </aspect>
  <aspect index="4" img=”back4.jpg”>
    <object x="380" y="165" img="guide11.jpg"/>
    <object x="121" y="344" img="guide12.jpg"/>
    <object x="229" y="156" img="guide13.jpg"/>
  </aspect>
  <aspect index="5" img=”back5.jpg”>
    <object x="329" y="146" img="guide16.jpg"/>
  </aspect>
  <aspect index="6" img=”back6.jpg”>
    <object x="389" y="265" img="guide17.jpg"/>
    <object x="321" y="344" img="guide18.jpg"/>
  </aspect>
</step>
```

Table 5. Folder Structure

```
- Step1
    - aspect1 : back1.jpg, guide1.jpg, guide2.jpg,
                            guide3.jpg,guide4.jpg
    - aspect2 : back2.jpg, guide5.jpg, guide7.jpg
    - aspect3 : back3.jpg, guide8.jpg, guide10.jpg
    - aspect4 : back4.jpg,guide11.jpg, guide12.jpg
                           guide13.jpg
    - aspect5 : back5.jpg, guide16.jpg
    - aspect6 : back6.jpg, guide17.jpg, guide18.jpg
+ Step2
      …
+ Step3
      …
+ Step4
```

```
        …
    + Step5
        …
    + Step6
        …
```

Each tutorial comprises several steps. Hence, several xml files are required, including content such as images, 3D objects, and video. Therefore, it is necessary to manage the files using a folder. Table 5 shows the folder structure. A folder has three levels. The first is step number, the second is aspect number, and the third is file name. There are two file types, background image and guide image. Background image represents aspect images, whereas guide image represents object images. Each object image is included in the aspect folder.

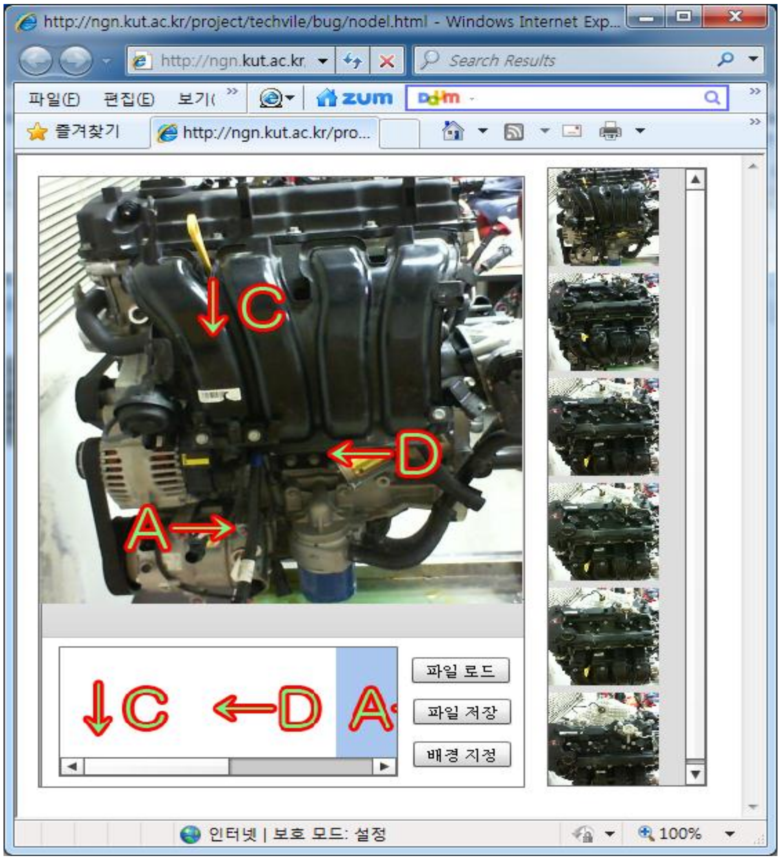
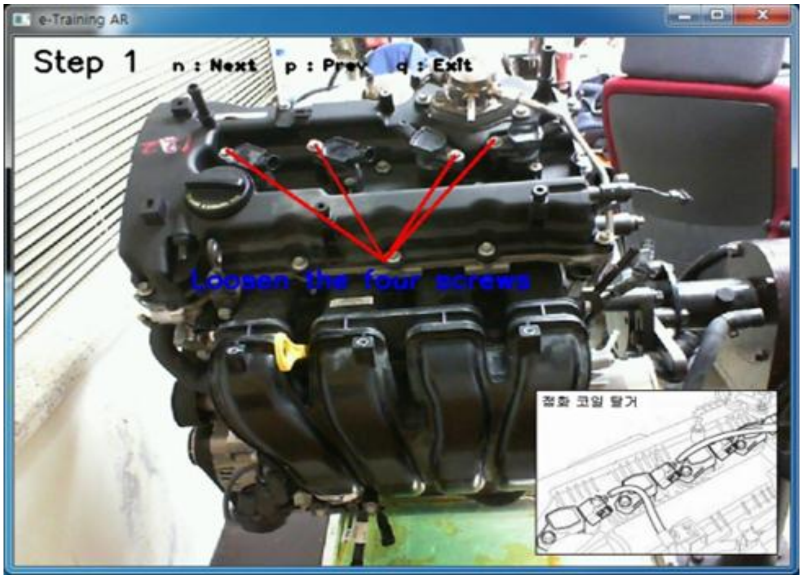


Figure 8. Web-based Content Creation System

4.3. Content Player

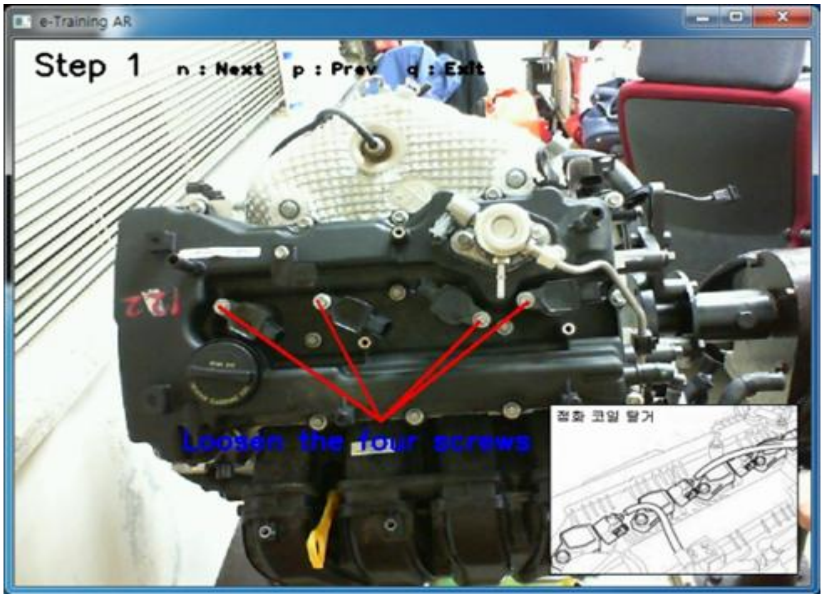


Loosen the four screws

점화 코일 탈거

Figure 9. Content Player (Step 1-1)

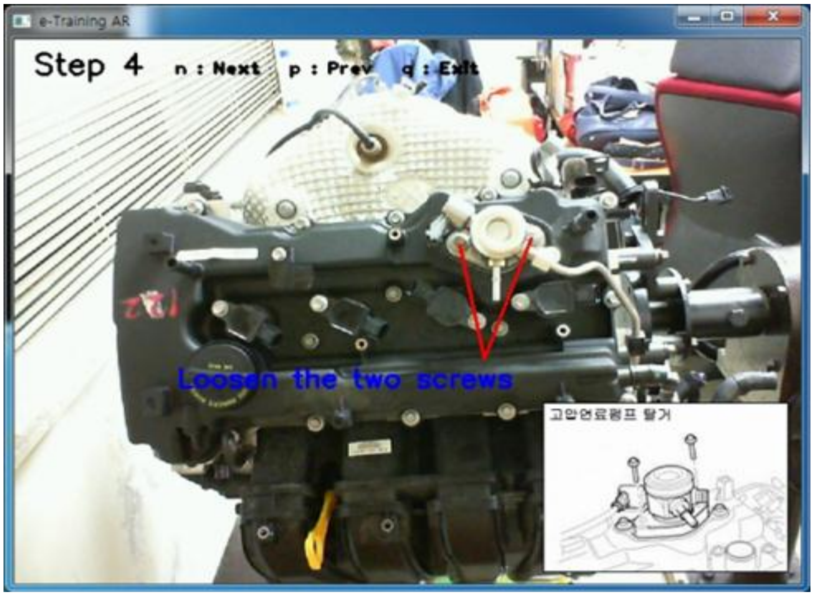
Figures 9 to 11 show the execution of the AR e-training application. When a user runs the program, real-time images are displayed. When the user focuses the camera on the engine, the AR player requests guide information from the CDN server and receives this information from the server. Next, guide information is displayed over engine. The user can then proceed to the next step. Similarly, guide information is displayed for this step.



Loosen the four screws

점화 코일 탈거

Figure 10. Content Player (Step 1-2)



Loosen the two screws

Figure 11. Content Player (Step 4)

5. Conclusion

Educational content has vast potential for future use in mobile devices. In particular, the AR technique is extremely useful for learning because it can display guide information over real objects. For example, users can repair their own cars by using AR training applications installed on their mobile devices. However, it is difficult to update existing content. Therefore, we proposed a real-time CDN-based content update system that always provides users with the latest information. Though this system, contents developers can provide latest contents to users easily. Therefore automotive companies will save cost because it is not need off-line learning when launched new car models.

Acknowledgements

This research was supported by a grant from the R&D Program(Industrial Strategic Technology Development) funded by the Ministry of Knowledge Economy(MKE), Republic of Korea. Also, The authors are deeply thankful to all interested persons of MKE and KEIT(Korea Evaluation Institute of Industrial Technology).

References

[1] Y. D. Kim , S. U. Lee and I. Y. Moon, "A Study on Mobile Virtual Training System using Augmented Reality", The Journal of Korea Navigation Institute, vol. 15, **(2011)**, pp. 1047-1052.

[2] H. Markus and L. R. Beaumont, "Content Networking: Architecture, Protocols, and Practice", Morgan Kaufmann Publishers, **(2005)**.

[3] Z. H. Lu, X. H. Gao, S. J. Huang and Y. Huang, "Scalable and Reliable Live Streaming Service through Coordinating CDN and P2P", Parallel and Distributed Systems, **(2011)**.

[4] J. H. Jung, G. Y. Oh, N. K. Lee, C. W. Yoon, H. W. Lee, W. Ryu and S. C. Lee, "Architecture and Server Selection for DHT-based Distributed CDN", The Journal of The Institute of Webcasting, Internet and Telecommunication, vol. 11, **(2011)**, pp. 217-228.

[5] P. Vate-U-Lan, "Augmented Reality 3D pop-up children book: Instructional design for hybrid learning", e-Learning in Industrial Electronics, **(2011)**.

[6] P. O. Boyinbode, A. Bagula and D. Ngambi, "An Opencast Mobile learning Framework for Enhancing Learning in Higher Education", International Journal of u - and e - Service, Science and Technology, vol. 4, no. 3, SERSC, **(2011)**, pp. 11-18.

[7] C. G Lim and K. T. Rho, "Design of Integrated Medical Information System based on XML", The Journal of The Institute of Webcasting, Internet and Telecommunication, vol. 10, **(2010)**, pp. 167-172.

[8] Md. S. Shahriar and J. Liu, "Towards a Definition of Referential Integrity Constraints for XML", International Journal of Software Engineering and Its Applications, vol. 3, **(2009)**, pp. 69-82.

[9] Md. S. Shahriar and J. Liu, "Towards the Preservation of Referential Constraints in XML Data Transformation for Integration", International Journal of Database Theory and Application, vol. 3, **(2010)**, pp. 1-10.

Authors

Yu-Doo Kim

He received B.S. degree in Internet Software Engineering from Korea University of Technology and Education, Korea in 2007, and M.S. degree from the same university in 2009. Now, He is a Ph.D. course student in the same university. His research interests include P2P network and web platform based on HTML5.

Il-Young Moon

He received his B.S., M.S. and Ph.D. degree from the department of information telecommunication, Korea Aerospace University in 2000, 2002, 2005 respectively. He was senior researcher in Korea Agency for Digital Opportunity and Promotion at 2005. Now, he is working as an associate professor in Korea University of Technology and Education. His research interests include wireless network and mobile internet.