

# A Design of Augmented Reality System based on Real-World Illumination Environment for Edutainment

Yang Shuo, Minho Kim, E-Jung Choi and Seoksoo Kim\*

Department of Multimedia, Hannam University, Korea;  
sskim0123@naver.com

## Abstract

Recently, edutainment system using augmented reality and transparent displays were researched. However, in the case of edutainment system using augmented reality due to the incongruity of environment light conditions, the virtual objects can't fit to real world environment with natural-looking. In this paper, we designed a tangible edutainment system for synthesizing a render layer which contains real world background illumination information in order to solve virtual object's low natural-looking problem.

**Keywords:** Augmented Reality, Edutainment System, Illumination Environment, Natural Feature

## 1. Introduction

Recently, the interest in smart grid, also known as an intelligent electrical grid, has been increasing due to the a result of planned blackout after the nuclear incident in Japan, increasing demand for electricity, spreading a sense of electricity crisis, environmental pollution, and aging electricity system. Major countries around the world are actively conducting research to activate smart grid projects<sup>1</sup>.

With the development of information communication technology, tradition paper educational media changes into digital mobile devices such as smart phone and tablet pc<sup>1</sup>. In particular, the traditional education method base on paper media is focusing on conveyance of knowledge, lack of interaction.

In current, educations tends to change into the edutainment system, which is developed in convergence of education and entertainment, it triggers learners' interest through their immersion and interaction. However, the existing edutainment systems provide stereotyped learning patterns in the virtual environment. Therefore, they

fail to motivate students' interest in learning, and cause poor user experience. To solve this problem, a realistic edutainment system using transparent display and augmented reality technology has been studied<sup>2</sup>. However, the systems using the augmented reality have the problem of low reality after image synthesis, because the system didn't considered influence of environment.

In this paper, we designs a real-world background based augmented reality edutainment system which include augmented reality system target recognition, operating background environment analysis and virtual objects rendering.

## 2. Related Works

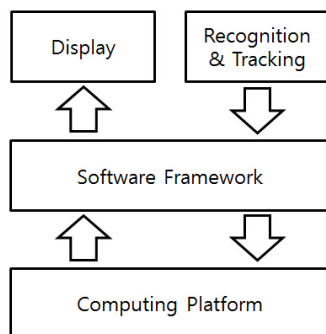
In related works part we analysis augmented reality system's structure, by studying the existing system we discover inadequacies of the existing system. By study of these inadequacies and advantage make our thinking more clearly to design the new system. In the first part we researched augmented reality system's structures to

\*Author for correspondence

find where we can improve to promote whole system performance. When the augmented reality system is running, we need keep virtual object projections stable, so we researched feature detection and feature extract of augmented reality system.

## 2.1 The Structure of Augmented Reality System

Augmented Reality (AR) is a variation of Virtual Environments (VE). VE technologies completely immerse a user inside a synthetic environment. AR allows the user to see the real world with virtual objects superimposed upon or composited with the real world<sup>3</sup>. Normally, an AR system working on static platform, but with the development of mobile computing, AR is becoming mobile<sup>4</sup>. Whether AR system mobile or not it still follow the basic structure:



**Figure 1.** Structure of augmented reality system.

Through Figure 1 the whole system include 3 layers and 4 parts, from bottom to top are computing platform, software framework, recognition & tracking method and display.

### 2.1.1 Computing Platform

Whole system's foundation, it can be normal personal computer or cloud server<sup>5</sup>.

### 2.1.2 Software Framework

Software framework used for control computing platform and manage data.

### 2.1.3 Recognition and Tracking

One of the most important part in system, affect system performance directly.

### 2.1.4 Display

Terminal interface, it has several different forms like Head-Mounted Display (HMD).

### 2.1.5 HMDs

It can provide users immersive, mobile and collaborative AR experiences<sup>6</sup>.

### 2.1.6 EyeTap

It is a glass-like HMD but without full head-mounted<sup>7</sup>.

### 2.1.7 Handheld

Displays employ a small display that fits in user's hand.



**Figure 2.** Head-mounted display (left) and eyetap (right).

## 2.2 Feature Detection and Feature Extraction

In order to output virtual objects at display device correctly, there need some method to confirm virtual objects coordinate in real-world. The feature detection is the most common way to determine target region, feature detection aim at computing abstractions of image information and making local decisions at every image point or region. Usually it is include edge, corner/interest points, blobs/regions or interest or interest points, ridges.

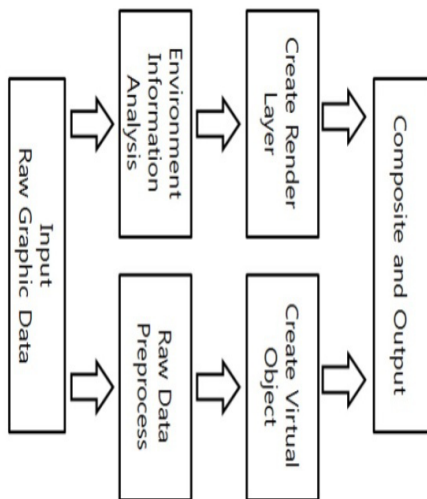
Once features have been detected, a local image patch around the feature can be extracted<sup>8</sup>. Compare with data set to decide detected feature set is the target or not. If detected feature set is the target feature set, then get the detected feature set position information for post-processing.

## 2.3 No Rendering

By observing the existed augmented reality software, we found there no software has rendered virtual objects, so the result is virtual objects which user through display devices watched, it just virtual objects itself material reflex. The contrast between virtual object and real-world environment make user has inadequate investment in the mood and make bad user experience.

### 3. Proposed Method

In this paper we proposed a real-world environment based natural-looking edutainment system. It included four parts, first is graphic data preprocessing, it about row data computing and feature detection. Second is about virtual objects projection. Third part is about environment analysis, and calculated the best fitted value of render mask. The last part is about merge the virtual with render layer make it natural-look and output it. This paper is presented following figure.



**Figure 3.** Structure of proposed system.

The system structure flow sheet is followed the next several steps. First, we need get basic input information from input devices (e.g. a camera) and decompose it to two parts, one is recognizable target area by feature detection method, and another is basic environment background area. Next step is use recognizing algorithm to calculate virtual object's coordinates and scale it, after virtual object scaled we should analysis the environment information, such as illumination and grey scale histogram to get background environment parameter's value. When environment information calculation is done virtual objects should be created but not output, through the virtual object's scale information to create a output device based object projection mask, the region of mask is the render layer's target zone, use environment information as render parameter to rendering virtual objects, and output final composited graphic to output devices.

#### 3.1 Graphic Data Preprocessing

After raw graph data are received, we need extract recognizable feature from graphic data. Normally, it's includes 2 steps, image data preprocessing and feature extraction.

In the part of de-noise we selected median filter as our system's de-noise method because it can reduce the impact noise effect and not fuzzy the edge of objects contour<sup>9</sup>.



**Figure 4.** Use of a median filter to improve an image corrupted by defective pixels.

It's perfect for our system, because our system are always operated in unstable light environments and parameter unknown devices, so we need increase graphic data's stability to our system for good robustness.

After de-noise finished next step is feature detection. In a digital image there lots of features existed, here we decided use Harris corner detector as our system's feature detector, because Harris corner detector provide an edge free interest points detect algorithm<sup>10</sup>, the algorithm's mathematical description is like this:

- Compute x and y derivatives of image

$$\begin{aligned} I_x &= G_x^x \times I \\ I_y &= G_y^y \times I \end{aligned} \quad (1)$$

- Compute products of derivatives at every pixel

$$I_{x2} = I_x \cdot I_x \quad I_{y2} = I_y \cdot I_y \quad I_{xy} = I_x \cdot I_y \quad (2)$$

- Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_x^1 \times I_{x2} \quad S_{y2} = G_y^1 \times I_{y2} \quad S_{xy} = G_{xy}^1 \times I_{xy} \quad (3)$$

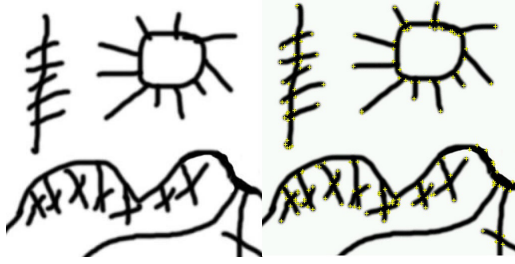
- Define at each pixel (x, y) the matrix

$$H(x, y) = \begin{bmatrix} S_{x2}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{y2}(x, y) \end{bmatrix} \quad (4)$$

- Compute the response of the detector at each pixel

$$R = \text{Det}(H) - k(\text{Trace}(H))^2 \quad (5)$$

- Threshold on Value of R. Compute non-max suppression.



**Figure 5.** Target (left) and target with features (right).

After feature detection is finished, compare detected features with target feature sequence, if the feature sequence's degree of similarity is in allowable range then deciding detected feature sequence is target feature sequence, which the region included is the target region.

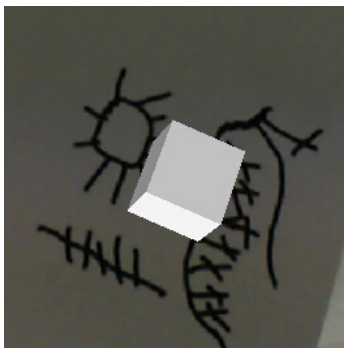
### 3.2 Create Virtual Objects

After target confirmed we should mapping virtual object to screen, first we need calculate objects transformation. For mapping a 2D or 3D object to a 2D screen there are several situations: translation, rigid, similarity affine, projective<sup>11</sup>.

In our system data input device will working in all positions and directions, from real world environment to virtual scene there must work through 3D to 2D projection. The most commonly used projection in computer graphics and computer vision is true 3D perspective<sup>11</sup>. Using inhomogeneous coordinates, it can be written as:

$$\bar{x} = p_z(p) = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} \quad (6)$$

By using 3D perspective to calculated feature sets position transform we can get the 2D screen based virtual object coordinate. Follow the above coordinate we can projection virtual object to 2D display screen correctly.

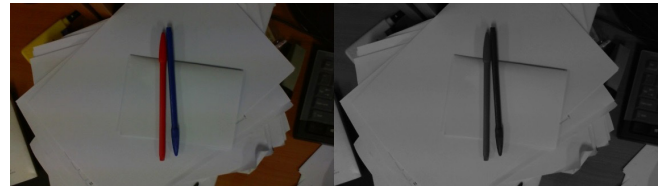


**Figure 6.** Virtual object projection.

### 3.3 Environment Analysis

After virtual object projection computing finished, the virtual objects are not output on screen directly, because we want the virtual object look-natural for users, so we need analysis environment first.

Normally target recognition process just need input graphic's gray scale data, but for Human Visual System (HVS) gray scale graphics are obvious insufficient. That's mean we also need collect color information of input environment.



**Figure 7.** 8bit RGB image and gray-scale image.

In order to build a natural looked virtual object projection, we need collect input environment data. In this case luminance and contrast ratio data are what we need. Before luminance and contrast ratio computing, we should remove the image target region's effects. So the luminance and contrast ratio computing region is this:

$$R_{t/c} = \sum_{i=0, j=0}^{i < width, j < height} I_{i,j} - \sum_{x=T_x, y=T_y}^{x < width, y < height} I_{x,y} \quad (7)$$

When the computing finished we extract input environment luminance and contrast data, using those data we can create a reference pattern that will provide necessary data to virtual object render layer. Next is calculated the illuminated environment colors. For collect those image color information, we need virtual object projection mask, fortunately, we already have it, we get it in virtual object create step, so just collect the color information near the mask we can got what we want:

$$I_{R,G,B} = RGB(x, y) \quad (x, y \in \text{mask dilate region}) \quad (8)$$

### 3.4 Virtual Object Rendering and Output

When we creating a virtual object, general the projection region also created, if we separate the region with image we got the rendering mask. When the rendering zone



mask is determined, we need fill the region with environment information value. There an image in-painting technique provides by Criminisi<sup>12</sup>, the original algorithm is used to image in-painting blob, we change the pattern to fill luminance and contrast ratio data.

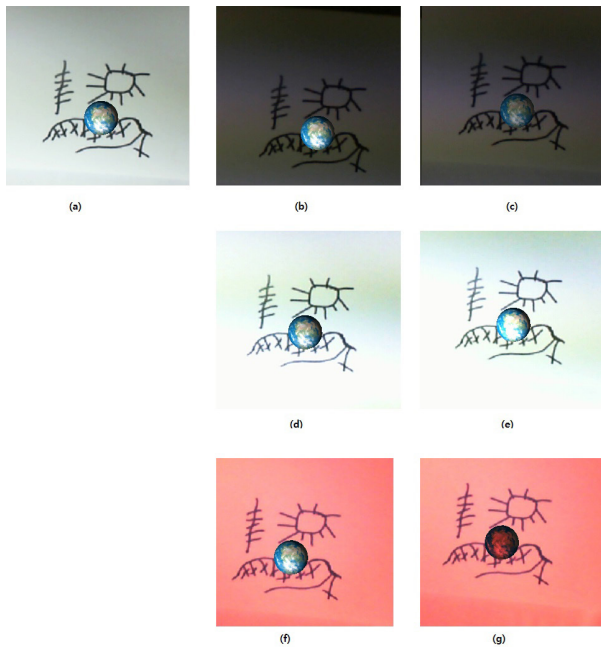
In Criminisi's paper  $P(p)$  is the product two terms' priority  $C(p)$  is the confidence term and  $D(p)$  is the data term.

$$P(p) = C(p)D(p) \quad (9)$$

In this time we set  $D(p)$  as the data of luminance and contrast ratio. The mask fill process is:

$$M_{x,y,w,h} = \sum_{i=x,j=y}^{i<width+x,j<height+y} R_{i,j} = P(p) \quad (10)$$

After rendering layer created we need composite rendering layer with virtual object, use rendering layer's data to override virtual object's basic data. When composite is finished, image will output as final result.



**Figure 7.** Virtual Object Rendering in Different luminance and Contrast ratio Situation. (a) Normal illuminated environment virtual object without rendering. (b) Dark illuminated environment virtual object without rendering. (c) Our method. (d) Strong illuminated environment virtual object without rendering. (e) Our method. (f) Colored illuminated environment virtual object without rendering. (g) Our method.

## 4. Conclusions

In this paper we proved the real-world information based augmented reality system for edutainment, we researched system structure of augment reality system, feature detection, and virtual objects projection, and improve old AR system weaknesses. But the system still has a lot of shortage in performance, so the next step is improve the target detection's accuracy of target texture regions and improve environment analysis module, because our system's environemnt analysis modules just considered luminance and contrast ratio's effect, make final result of render layer's performance are not good enough.

Our work proved a virtual object can be rendering by real-world environment, our method can improve augmented reality system's interactions and immersion. For edutainment system more interaction and immersion means more fun in knowledge learning process, it will drive learning initiative of students.

## 5. Acknowledgment

This research was supported by the INNOPOLIS Foundation grant funded by the Korea government (Ministry of Science, ICT and Future Planning) (No. 2013N180).

This paper has been supported by the 2015 Hannam University Research Fund.

## 6. References

1. Cloughm G, Jones AC, McAndrew P, Scanlon E. Informal learning with PDAs and smartphones. *Journal of Computer Assisted Learning*. 2010; 24(5):359–71.
2. Kim DH, Kim SS. Design of smart learning system based on AR-APM. *International Journal of Multimedia and Ubiquitous Engineering*. 2014; 9(6):119–26.
3. Azuma RT. A survey of augmented reality. *Teleoperators and Virtual Environments*. 1997; 6(4):355–85.
4. Zhanpeng H, Pan H. Mobile augmented reality survey: A bottom-up approach. *Cornell University Library*. 2013.
5. Zhanpeng H, Wei Kai L, Pan H, Christoph P. CloudRidAR: A cloud-based architecture for mobile augmented reality. *Proceeding of MARS'14 ACM*; 2014. p. 29–34.
6. Rolland J, Biocca F, Yanggang H, Martins R. Development of head-mounted projection displays for distributed. *Collaborative augmented reality applications. Teleoperators and Virtual Environments*. 2005; 14(5):528–49.

7. Mann S. Through the glass, lightly. IEEE Technology and Society. 2012; 31(3):10–4.
8. Wikipedia. Feature Detection. Available from: [http://en.wikipedia.org/wiki/Feature\\_detection\\_\(computer\\_vision\)](http://en.wikipedia.org/wiki/Feature_detection_(computer_vision))
9. Huang T, Yang G, Tang G. A fast two-dimensional median filtering algorithm. IEEE Trans Acoust, Speech, Signal Processing. 1979; 27(1):13–8.
10. Harris C, Stephens M. A combined corner and edge detector. Alvey Vision Conference; 1988. p. 147–51.
11. Richard S. Computer Vision: Algorithms and Applications. 2010.
12. Criminisi A. Region filling and object removal by exemplar-based image inpainting. IEEE Transactions on Image Processing. 2004; 13(9):1200–12.