

증강현실에서 빛 정보를 이용한 색상 보정기법

박자람[○] 윤희용

성균관 대학교

qkfa315@gmail.com, youn@ece.skku.ac.kr

A Color Correction Method using Lighting Information In Augmented Reality

Jaram Park[○] Hee Yong Youn

SungKyunKwan University

요 약

증강현실은 이미 이를 이용한 다양한 서비스들이 존재하고 있고, 최근 스마트폰이 주목받으며 덩달아 부각되고 있는 개념 중의 하나이다. 기존 증강현실을 이용한 시스템에서는 객체를 증강할 때 미리 만들어진 객체를 증강하므로 객체가 증강할 당시의 주변환경의 변화가 반영되지 않는다. 따라서 증강현실을 할 때 카메라로 들어오는 현재 배경이미지와 객체가 동떨어져 보이는 현상이 발생한다. 이는 사용자가 현실세계에 가상의 물체가 올라간 것이 어색해 보여 증강현실기반의 프로그램에 몰입하는데 방해가 된다. 본 논문에서는 증강현실기반의 시스템이 사용될 때 실시간으로 주변환경의 빛 정보를 분석하여 그 정보를 객체에 반영하는 방법을 제안한다. 다양한 주변환경 중에서도 시각적으로 많은 정보를 제공하는 빛 정보를 이용하여 보다 자연스럽게 객체의 색을 보정을 하며, 이는 실시간으로 객체에 적용된다. 이는 주변환경 정보가 다양하게 바뀌는 모바일 증강현실에서 유용하게 쓰일 것이다.

1. 서 론

증강현실은 증강현실을 이용한 게임에서부터 의료분야, 제조분야에 이르기까지 이미 이를 이용한 실내·외 다양한 서비스들이 존재하고 있고, 최근 스마트폰이 주목을 받음에 따라 다양한 증강현실기반의 모바일 어플리케이션이 만들어지고 있다. 이에 증강현실은 더욱 부각되고 있는 기술중의 하나이다. 증강현실은 사용자가 실제 환경을 볼 수 있으면서 실제 환경과 가상의 객체가 혼합된 형태를 띤다. 다시 말해서 증강현실은 현실세계에 가상의 물체를 중첩함으로써 현실세계를 보충하여 사용자에게 보여준다는 차별성을 가지며, 가상현실에 비하여 사용자에게 보다 나은

현실감을 제공한다는 특징이 있다.

다양한 분야로 응용되고 있는 증강현실에 필요한 공통적인 특징을 알아보면, 먼저 증강현실 시스템은 사람이 보는 기준 프레임에 가상 프레임을 잘 정합하여야 한다. 그렇지만 일반적으로 사람들은 가상환경에서 발생하는 시각-운동 오류보다 시각적 부정합에 더 민감하기 때문에, 증강현실 시스템에서 정합이 매우 어렵다. 즉, 가상물체는 실세계와 같이 움직이지 않는 것으로 생기는 어색함보다 환경과 일치하지 않아 발생하는 어색함이 사람에게 더 크다.

증강현실의 핵심요소인 장면은 영상 장치로 나타나며 카메라는 3D세계에서 2D영상으로 투영된다.

가상영상은 컴퓨터 그래픽 시스템에서 만들어진다. 가상 물체는 물체 기준 프레임에서 모델링이 되어 만들어진다. 그래픽 시스템은 이 가상 물체를 정확히 렌더링 하기 위해서 실제 영상에 관한 정보가 반드시 필요하다. 이 데이터는 가상 물체 영상을 가지고 합성 카메라로 제어된다. 마지막으로 이 영상은 실제 영상에 합성되어 증강현실 영상을 형성한다. [1]

이와 같이 증강현실에서 현실세계와 가상세계가 잘 정합된다는 것은 아주 중요한 조건이다. 그러나 기존 증강현실을 이용한 어플리케이션에서는 객체를 증강할 때 미리 만들어진 객체를 증강하므로 객체가 증강할 당시의 주변환경의 변화가 반영되지 않는다. 따라서 증강현실을 할 때 카메라로 들어오는 현재 배경이미지와 객체가 동떨어져 보이는 현상이 발생한다. 이는 사용자가 현실세계에 가상의 물체가 올라간 것이 어색해 보여 증강현실기반의 프로그램에 몰입하는데 방해가 될 수 있다.

본 논문에서는 사용자가 사람이 시각-운동 오류보다 시각적 부정합에 민감하다는 점에 초점을 맞추어 주변 환경, 그 중에서도 빛 색깔 정보를 이용하여 자연스러운 객체 증강을 위한 색 보정 방법을 제안하고자 한다. 객체가 증강될 때 프레임으로 들어오는 영상의 색깔 값 데이터를 통하여 객체가 증강될 당시의 빛의 색을 분석하였다. 이때 다른 하드웨어 장치 등을 필요로 하지 않고 카메라 입력 값으로 들어온 R, G, B 색깔 값을 이용하여 분석하게 하였다. 그 분석된 결과를 객체에 자연스럽게 적용시켜 주변 환경의 빛 변화가 객체에도 나타나 자연스럽게 정합되게 하였다.

본 논문은 2장에서 관련 연구에 대하여 알아보고, 3장에서 제안된 색 보정 방법을 기술하였다. 4장에서는 제안된 방법을 통하여 구현된 결과를 보여주며 5장은 연구의 결과와 향후 연구에 대한 설명이다.

2. 관련 연구

2.1 증강현실

증강현실(Augmented Reality)는 일반적으로 VE(Virtual Environment) 혹은 VR(Virtual Reality)에서 파생한 것으로 실제 환경(real environment)에 컴퓨터 그래픽스 영상을 삽입하여 실제와 가상영상 혼합을 뜻한다[2]. 실세계는 컴퓨터로 만들기 힘든 많은 정보들이 존재한다. 그렇지만 그 정보들 중에 우리가 필요하지 않은 많은 정보도 있고, 때로는 우리가 필요로 하는 정보가 부족할 수도 있다. 컴퓨터로 재생한 가상환경을 이용하면, 필요하지 않은 정보들을 단순하게 만들거나 보이지 않는 정보들을 만들어 낼 수 있다.

사람들은 실제와 가상의 혼합체를 통하여 주위 환경을 쉽게 인식하고, 이를 보다 더 잘 이용할 수 있다. 효율적인 증강현실 시스템은 실세계와 가상현실을 합성하며, 실시간으로 사람과 상호 작용이 이루어져야 한다. 따라서 증강현실 시스템은 삼차원 세계에 잘

정합되어야 좋은 시스템이다. 실세계와 가상세계 간에 공존하는 세계를 이른바 혼합현실(mixed reality)이라 한다. 증강현실 시스템은 사용자가 실세계 환경 하에서 가상물체를 부가되어 증강한 실세계를 말한다. 증강현실에서 사용된 중요한 기술들은 가상물체를 만드는 3D 모델링, 실제와 가상환경을 합성하여 장면을 보여주는 디스플레이 장치, 투영에 따른 왜곡을 분석, 카메라 변수를 계산하는 카메라 교정, 사용자의 위치와 방향을 찾는 움직임 추적, 가상물체를 실제환경에 합성되는 정합 등이 있다.[1]

2.2 주변 환경 정보를 이용한 증강현실 연구

증강현실을 기반으로 한 시스템이 다양한 분야에서 활용 됨에 따라 증강현실에 관련된 연구도 활발히 진행되고 있다. 하지만 그 동안은 주로 증강현실을 이용하여 다른 시스템을 응용하는 것에 초점이 맞춰져 있었다. 그 중에서도 특히 다양한 환경에서 사용되는 모바일 증강현실이 최근 큰 주목을 받고 있으며, 주로 이동성에 초점을 맞춘 연구들이 진행되고 있었다. 최근에는 증강현실 자체의 보다 높은 완성도를 위한 연구도 활발히 진행되고 있다[2][3][4].

기존 연구 중에는 증강현실에서 가장 흔히 쓰이는 마커에 색깔을 입혀 이용한 색 보정기법이 있다[5]. 이 연구에서는 주변의 조명 환경에 의하여 마커의 색깔이 변하여 제대로 트래킹이 되지 않는 문제를 해결하는 연구를 진행하였다. 마커가 일정한 색인 흰색과 검은색으로 주로 나뉘어 있다는 점에 착안 해서 마커에 검은색과 흰색 외에 R, G, B 세 개의 색깔을 포함한 컬러 패턴의 마커 이미지를 이용하였다. 이를 통해 본 논문과는 반대의 방법이지만 같은 원리를 이용한 것을 알 수 있다. 컬러 패턴을 통하여 현재 영상으로 들어온 마커의 색을 인식한 후 R, G, B 값을 다시 맞춰주어 마커의 흰색부분과 검정색부분을 보정해주는 것이다.[6]

주변 환경 정보를 이용한 또 다른 증강현실 시스템은 조도 센서를 달아 스톱 홀드 값을 변경시키는 연구이다[7][8]. 정적인 카메라 입력영상의 경우에는 스톱 홀드 값을 일정하게 정해주면 마커의 이진화가 잘 되지만 실제 증강현실이 사용되는 곳에서는 빛의 세기가 달라지기 때문에 같은 스톱 홀드 값으로는 제대로 인식이 안되어 이진화가 안될 수도 있다[9]. 그래서 이 연구에서는 마커 위에 조도 센서를 달아 실시간으로 빛의 세기를 알아낸 다음 이에 따라 가장 최적화된 스톱 홀드 값을 적용한다. 이를 통하여 마커의 인식률을 높인다[10].

이미지에서 들어온 데이터를 이용해 자연스러운 객체를 증강하기 위한 연구도 진행되었다. 이 연구의 목적 역시 객체정합이 이루어졌을 때 카메라의 속도 등이 반영되어 있지 않아 영상에서 불러된 실세계를 반영하지 못한 점을 보완하기 위하여 이루어졌고,

카메라의 입력영상에서 R, G, B, A 데이터 값과 카메라 속도 값을 이용하여 객체의 색 보정 및 블러 효과를 주었다[11].

3. 제안 시스템

본 논문에서는 데스크탑 환경에서 마커를 사용하지 않는 Markerless 기반의 증강현실 시스템을 기반으로 한다. 2D 평면상에서 오브젝트를 카메라 영상으로 인식한 후 트래킹과 동시에 2D 객체를 증강한다.

본 시스템의 핵심은 기존 주변환경정보를 이용한 증강현실 연구에서 조도 센서 등의 하드웨어를 필요한 점을 보완하여 기타 하드웨어는 사용하지 않으면서도 실시간으로 효과적인 정보 반영을 목표로 하였다. 따라서 오브젝트 인식 등의 기본적인 증강현실 기능은 구현되어 있는 상태라고 가정하고 주변환경의 조명정보를 분석하고, 그것을 적용하는 데에 초점을 맞춘다.

본 시스템은 크게 두 구조로 나눌 수 있다. 실시간으로 실행되는 Online 과정과 프로그램이 실제 실행되기 전에 준비과정인 Offline 과정이다.

3.1 Offline

Offline 부분은 미리 영상데이터가 들어왔을 때 비교할 원본 이미지를 저장해놓은 것이다. 이는 주변 환경정보가 들어왔을 때 객체에 적용할 데이터를 도출하기 위한 비교 원본 데이터이다. 원본 자체도 저장할 때 다른 조명등의 환경에 의하여 왜곡될 가능성이 있기 때문에 이를 최소화하기 위하여 스캔하는 방법을 선택하였다. 이 외에도 Offline 상에 증강할 2D객체도 저장해 둔다. 원본데이터를 항상 가지고 있으면서 실시간으로 색을 보정할 때 이용하게 된다.

3.2 Online

Online 부분은 실시간으로 데이터가 들어오면서 시작되는 부분이다. 이때 본 논문의 핵심인 카메라로부터 들어온 영상 이미지 데이터를 원본 이미지와 비교하여 색을 보정하는 기능이 수행된다.

3.2.1 Pixel by pixel color comparing

원본 이미지 데이터를 카메라 영상에 찍혀 들어오면 프레임별로 그 영상에서 오브젝트의 Homography값을 구하여 왜곡되어있는 오브젝트 영상부분을 다시 원본이미지처럼 만드는 작업이 수행된다. 이 작업을 통하여 원본이미지가 어느 방향으로 틀어져있는지 알 수 있으며 왜곡된 정보를 객체에도 적용하여 실시간으로 객체가 알맞게 정합될 수 있는 것이다. 본 연구에서는 이렇게 다시 만들어진 Homography 처리된 이미지의 픽셀 하나하나 각각을 미리 Offline 과정에서 저장해 두었던 원본이미지와 R,

G, B 세 개의 색깔 값을 비교한다. 이때 비교한 색깔 차이 값을 역시 Offline과정에서 미리 저장해 두었던 2D 객체의 대응되는 각 픽셀 값에 더해준다. 이때 R, G, B 색깔 값은 0~255범위의 값을 갖기 때문에 객체에 더해지고 난 후의 값이 0미만일 경우는 0으로, 255이상일 경우는 255로 처리해 주는 과정이 필요하다.

R, G, B 색깔 값을 c , Offline 과정에서 저장된 원본 이미지의 픽셀 색깔 값을 I_c , Homography 처리 후의 영상 데이터 픽셀 색깔 값을 H_c , 2D 객체를 O_c 라 하자. Pixel by Pixel color comparing 방식을 의사코드로 나타내면 다음과 같이 나타낼 수 있다.

```

if(  $(O_c + (I_c - H_c)) > 255$  )
     $O_c = 255$ 
else if(  $(O_c + (I_c - H_c)) < 0$  )
     $O_c = 0$ 
else
     $O_c = O_c + (I_c - H_c)$ 
    
```

이 방식은 실시간으로 조명에 의한 빛 변화뿐만 아니라 객체 부분에 직접적으로 가해지는 외부 자극들도 반영할 수 있다는 장점이 있지만 색깔 별로 조명 색을 반사하는 값이 달라 발생하는 원본이미지가 객체에 투영되는 단점이 있다. 이에 대한 해결 방법은 아래에서 다시 설명할 것이다.

3.2.2 Average color comparing

위에서 설명한 Pixel by pixel color comparing 방식은 픽셀 하나하나 값을 각각 비교하였던 방식이다. 지금부터 설명할 Average color comparing 방식은 객체가 증강될 부분의 카메라 입력데이터, 즉 Homography처리된 이미지와 원본이미지의 전체 픽셀 색깔 값 차이의 평균을 구해 그 평균값을 2D 객체에 더해주는 것이다.

위에서와 마찬가지로 R, G, B 색깔 값을 c , Offline 과정에서 저장된 원본 이미지의 픽셀 색깔 값을 I_c , Homography 처리 후의 영상 데이터 픽셀 색깔 값을 H_c , 평균 차이 값을 A_c , 2D 객체를 O_c 라하고, 2D객체의 높이와 너비 값을 곱한 값, 즉 모든 픽셀 개수의 값을 N 이라고 할 때 Average color comparing 방식을 의사코드로 나타내면 다음과 같다.

```

 $A_c = \sum (I_c - H_c) / N$ 
if(  $(O_c + (I_c - H_c)) > 255$  )
     $O_c = 255$ 
else if(  $(O_c + (I_c - H_c)) < 0$  )
     $O_c = 0$ 
else
     $O_c = O_c + A_c$ 
    
```

Average Comparing 방식의 장점은 객체에 색이 반영될 때 원래 초기에 달성하고자 했던 자연스럽게 색반영이 잘 된다는 점이 있다. 단점은 객체 내에 부분별로 가해지는 변화를 나타낼 수 없다는 것이다.

3.2.3 Hybrid Approach

지금까지 Pixel by pixel로 색깔 값을 비교하는 방식과 픽셀 값의 전체적인 차이를 평균 내어 객체에 적용하는 방식을 알았다. 두 방법 모두 장단점이 존재하며 색깔 별로 색 반사율이 달라 색 차이 역시 다르게 나타나는 것을 확인할 수 있었다.

다음은 본 연구에서 최종 목표로 하였던 주변환경 정보를 반영하여 자연스러운 객체 증강을 하기 위한 기본 방법이다. 위에서 살펴보았던 Pixel by pixel 비교 방식과 Average 비교 방식을 하이브리드하여 최종 객체에 적용할 값을 구한다.

전체 Average 평균 차이 값을 A_D , 픽셀 별로 비교하여 구한 차이 값을 P_D , 원본 2D 객체는 Original 2D Object라 할 때 최종 적용되는 객체를 의사코드로 나타내면

$$\text{modified 2D object} = \text{original 2D object} + \alpha A_D + \beta P_D$$

로 나타낼 수 있다. α 와 β 값은 임의의 값이며 시스템에 따라 사용자와 인터랙션이 많아 부분 정합이 많이 들어갈 경우에는 β 를 더 크게 하면 되고, 사용자와의 인터랙션보다는 전체적인 조명 값의 변화가 클 경우에는 α 를 크게 하여 Average 값이 더 많이 반영되게 하면 된다.

3.2.4 Color compensation

위의 두 방식에 색 범위에 따른 색깔 값의 차이만큼 다시 색을 보정해주지 않으면 어두운 부분과 밝은 부분의 색 차이만큼 객체에 적용 되 결국 최종 객체 모습엔 원본 이미지 모습이 그려져 뒷 배경(그림 1)이 반영되는 것과 같은 효과가 나타나는 것을 볼 수 있다.



그림 1 원본 이미지의 색깔에 따라 객체에 반영되는 모습

그림 2는 R, G, B 각 색깔 별로 색 차이가 다르게 나타나는 모습을 정량적으로 보여주는 그래프이다. 다시 말하여 똑 같은 조명, 빛의 색을 이미지에 적용이 되어도 대체적으로 R, G, B 세 개의 채널 모두 255값에 가까이

갈수록 색 차이 값이 크고, 0에 가까이 갈수록 색 차이 값이 작다는 것을 확인할 수 있다.

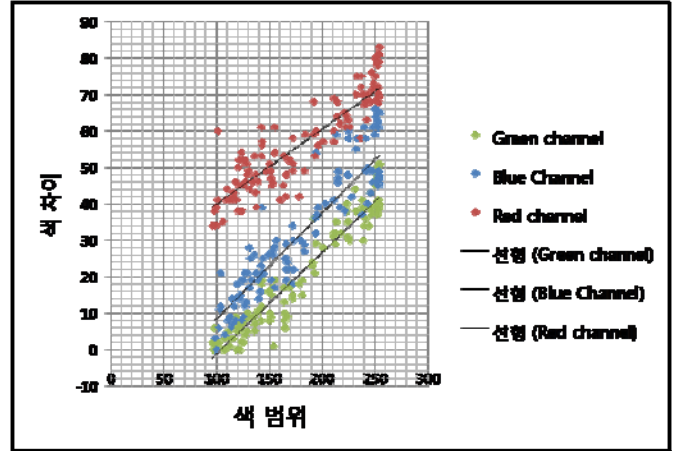


그림 2 색깔 범위 별 반사율에 따른 색 차이

3.2.5 최종 색깔 보정법

위에서 Hybrid 방식에 대해서 살펴 보았다. 여기에 색 보상을 해주어야 뒤에 배경이 그려지는 현상을 없앨 수 있다. 그래서 본 논문에서는 스캔한 원본 이미지와 카메라 영상으로 들어오는 이미지의 color 차이 값을 각각의 배열 값에 저장한 후 각 색깔에서 255에 가장 가까운 색과의 차이를 구하여 그 차이만큼 색을 보상하는 방법을 제안한다.

$R[255], G[255], B[255]$ 세 개의 배열을 두고 이를 크게 $C[255]$, 미리 저장해 둔 이미지의 색깔 값을 $Scan_C$ 라 하고, 영상으로 들어오는 이미지의 색깔 값을 $Input_C$ 라 하였을 때, 의사코드로 나타내면 다음과 같다.

$$\begin{aligned} C[Scan_C] &= Input_C - Scan_C \\ \text{modified 2D object} &= \text{original 2D object} \\ &+ \alpha A_D + \beta P_D \\ &+ C[255\text{에 가까운 수}] - C[Scan_C] \end{aligned}$$

가령 원본이미지 데이터 중 한 픽셀의 R값이 100이라면 이미 만들어 놓은 배열 $R[100]$ 에 현재 그 픽셀 값과 그에 대응하는 새로 들어온 영상의 픽셀 값의 차이를 저장하는 것이다. 그 다음에 원본 이미지 내에 255값이 없을 수도 있음을 가정하여 255에 가장 가까운 수의 배열에 저장되어 있는 차이 값에서 아까 저장해 두었던 값을 뺀다. 즉 255에 가장 가까운 값을 기준으로 하여 부족한 차이 값만큼 다시 더해 주는 것이다. 모든 수의 차이 값을 정확하게 미리 저장해 놓을 수 없기 때문에 완벽한 보상은 힘들지만 훨씬 자연스럽게 색이 보상되는 것을 확인할 수 있다.

4. 구현 결과

다음은 본 시스템의 최종 구현 결과이다. 기존 증강현실 어플리케이션의 경우 미리 만들어진 객체이기

때문에 주변의 조명조건이 변하여도 객체는 처음에 만들어진 색 그대로 이다. 본 색 보정 방법을 이용한 결과는 다음과 같다.

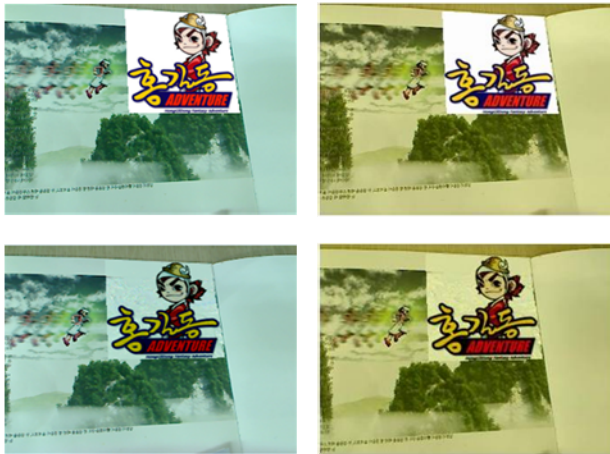


그림 3 객체의 전체적인 조명 변화에 따른 변화. 위 : 객체에 색정보를 적용하기 전, 아래 : 제안된 방법을 적용한 후



그림 4 객체 증강 부분별 변화 적용. 위 : 객체에 색 정보를 적용하기 전, 아래 : 제안된 방법을 적용한 후

위의 그림 3,4에서 볼 수 있듯이 기존 증강현실에 비하여 주변 조명 색이 반영되어 현실세계의 영상에 가상의 2D객체가 잘 정합되어 마치 현실세계에 있는 것과 같은 효과를 줄 수 있음을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서 제안하는 증강현실 시스템은 미리 만들어진 객체를 바로 올리는 것이 아니라 실시간으로 주변의 조명정보나 정합 정보를 분석한다. 그 다음에 그 결과를 이용하여 다시 객체 이미지를 다시 텍스처링하여 사용자가 보는 디스플레이 환경에 렌더링해 주는 것이다. 이는 사용자가 기존증강현실 어플리케이션에서 객체가 뒷 배경과 동떨어져 보여 몰입에 방해가 되었던 단점을 해결해준다. 또한 기존 연구에서는 자연스러운 객체를 증강하기 위하여 다른

센서나 Ball 등의 하드웨어를 부착하였던 것과 달리 카메라에서 들어오는 이미지만을 이용하여 주변환경의 조명정보, 정합정보를 분석하는 것이기 때문에 미리 만들어진 증강현실 기반 어플리케이션에도 쉽게 적용 가능하며 특히 모바일 증강현실에서 유용하게 사용될 것이다.

본 논문은 데스크탑 환경에서 구현되어 효과적인 성능 및 결과를 보여주었으나 아직 모바일 환경에서 테스트해보지 않았다. 따라서 향후 모바일 플랫폼에서 적용해보는 연구가 필요하다. 또한 현재는 많은 주변 환경 정보 중에서도 조명정보와 정합정보만을 이용하였지만 객체 주변이 카메라 초점에 의한 블러 현상을 보정하는 연구가 진행되어야 할 것이다.

참고 문헌

- [1] 윤용인, 김진태, 김동욱, 최종수, 증강현실 기술 현황, 한국해양정보통신학회지 제 5권 1호
- [2] HOUGEN D.,ABUJA N, Estimation of the lightsource distribution and its use in integrated shape recovery from stereo and shading. In IEEE 4th Int. Conf. On Computer Vision, pp.148-155, (1993)
- [3] Yuchi Otha, Hideyuki Tamura, "Mixed Reality-Merging Real and Virtual Wolds", Augmented Reality 2003.
- [4] R.T. Azuma, A Survey of Augmented Reality, Peleoperators and Virtual Environments, vol.6, no. 4, pp.355-385, 1997.
- [5] Wonwoo Lee, Woontack Woo, Real-time Color Correction for Marker-based Augmented Reality Applications. IWUVR2009, pp.32-35
- [6] Bunyo OKUMURA, Masayuki Kanbara, Naokazu Yokoya. Augmented Reality Based on Estimation of Defocusing and Motion Blurring from Captured Images. 2006 IEEE
- [7] Stephen Diverdi, Tobias Hollerer, Combining Dynamic Pysical and Virtual Illumination in Augmented Reality.
- [8] Julien Pilet, Vincent Lepetit, Pascal Fua, Retexturing in the Presence of Complex Illumination and occlusions, 2007 IEEE
- [9] Dongpyo Hong, Julian Looser, Hartmut Seichter, Mark Billinghurst, Woontack Woo, A Sensor-based Interaction for Ubiquitous Virtual Reality Sytems., ISUVR 2008
- [10] Wonwoo Lee, Woontack Woo, Exploiting Context-awareness in Augmented Reality Applications. ISUVR 2008
- [11] Georg Klein, David Murray, Compositing for Small Cameras, 2008 ISMAR