

비디오 영상에 가상 물체의 그림자 삽입을 통한 향상된 AR 구현

Enhanced Augmented Reality with Realistic Shadows of Graphic Generated Objects

김태원, 홍기상
포항공과대학교 전자전기공학과 영상정보처리 연구실
Taeone Kim and Ki-Sang Hong*
Department of Electronic and Electrical Engineering
POSTECH

Abstract

In this paper, we propose a method for generating graphic objects having realistic shadows inserted into video sequence for the enhanced augmented reality. Our purpose is to extend the work of [1], which is applicable to the case of a static camera, to video sequence. However, in case of video, there are a few challenging problems, including the camera calibration problem over video sequence, false shadows occurring when the video camera moves and so on. We solve these problems using the convenient calibration technique of [2] and the information from video sequence. We present the experimental results on real video sequences.

1 서론

현재 비디오 증강 현실(Augmented Reality)에 대한 연구가 활발히 연구가 진행되고 있다 ([3, 4]). 증강 현실은 컴퓨터 비전과 컴퓨터 그래픽스의 융합으로 탄생하게 되었는데, 의료 진단, 원격 로봇 조정, 영화의 특수 효과 분야와 같은 다양한 분야에의 활용이 기대된다. 기본적으로 증강 현실 연구의 목적은 실제 비디오 영상이나 단일 영상에 컴퓨터로 만들어진 가상 물체를 넣어서 마치 예전부터 존재했던 것처럼 보이게 만드는 것이다.

본 논문에서는 보다 향상된 증강 현실을 위해 비디오 영상으로부터 실제 광원의 분포(Illumination distribution)를 분석한 후에 이를 이용해서 가상 물체(synthetic object)의 그림자를 만들어 내는 방법을 제안한다. 실제로 비디오 영상에 삽입된 가상 물체의 그림자는 가상과 현실 분간에 있어서 중요한 요소가 될 수 있다. 최근에 정지 영상안에 존재하는 실제 그림자를 분석하여 광원의 분포를 추정하는 방법이 개발되었다([1, 5]). 이 방법은 무한히 멀리 떨어져 있는 빛의 분포에 의해서 입사된 빛이 물체의 형태에 따라서 차단(occluded)되는 점을 이용하여 빛

의 분포를 측정된 후에 정지 영상에 가상 물체의 셰이딩(shading) 및 그림자를 삽입하여 그 현실감을 높일 수 있었다. 본 논문에서는 이 기법을 일반 웹코더에 의해 얻어진 비디오 영상에 효과적으로 확대 적용할 수 있는 방법을 제안한다. 이를 위해서 우선, 물체가 평면위에 놓여 있고 빛에 의해서 그 평면으로 그림자가 투사되고 있는 상황을 가정하고, 그림자가 투사되는 평면의 광학적 성질을 Lambertian으로 가정한다.

먼저 2장에서 그림자로부터 빛의 분포를 측정하는 기법을 간략히 기술한다. 3장에서는 빛의 분포 측정과 가상 물체를 비디오에 삽입하기 위해 선행되어야 할 비디오 카메라 캘리브레이션(calibration)에 대한 내용이 다루어진다. 5.1장에서는 기존의 정지 영상의 그림자를 이용한 빛의 분포 측정시의 한계점을 언급하고 그 한계점을 비디오 영상의 정보를 이용하여 극복할 수 있는 방법을 제안한다. 마지막으로 5장에 본 논문에서 제안된 방법을 실제 비디오 영상에 적용한 결과를 보이고 논문을 끝맺고자 한다.

2 빛의 분포 측정

[1]에 의하면, 우리가 최종적으로 카메라로부터 얻는 영상은 아래와 같이 표현될 수 있다.

$$P(x_k, y_k) = K_d(x_k, y_k) \sum_{i=1}^n L(\theta_i, \phi_i) S(\theta_i, \phi_i) \cos(\theta_i), \quad (1)$$

여기서 (x_k, y_k) 는 영상 좌표를, K_d 는 Lambertian 상수, L 는 빛의 광도(Illumination radiance), S 는 차단 상수(occlusion coefficient)를 나타낸다. 식 (1)는 geodesic dome의 node direction에 해당하는 모든 방향 $(\theta_i, \phi_i), i = 1, \dots, n$ 에 대해서 합을 취해서 얻어진다. 만약에 어떤 방향 (θ_i, ϕ_i) 의 빛이 물체에 의해서 차단된다면 $S(\theta_i, \phi_i) = 0$ 이고, 그렇지 않을 경우는 $S(\theta_i, \phi_i) = 1$ 이 된다. 식 (1)에서 $S(\theta_i, \phi_i)$ 를 제거해서 얻어진 영상을 $P'(x_k, y_k)$ 라하면 이 영상은 그림자를 투사하는 물체가 없는 상황의 영상에 해당한다. 그러면 두 영상의 비율을 구하면 아래와 같은

Address: San 31 Hyoja Dong, Pohang, Kyungbook, 790-784, Korea. E-mail: {kimm, hongks}@postech.ac.kr

식이 유도된다.

$$\frac{P(x_k, y_k)}{P'(x_k, y_k)} = \frac{K_d(x_k, y_k) \sum_{i=1}^n L(\theta_i, \phi_i) S(\theta_i, \phi_i) \cos(\theta_i)}{K_d(x_k, y_k) \sum_{j=1}^n L(\theta_j, \phi_j) \cos(\theta_j)}$$

$$= \sum_{i=1}^n \frac{L(\theta_i, \phi_i)}{\sum_{j=1}^n L(\theta_j, \phi_j) \cos(\theta_j)} S(\theta_i, \phi_i) \cos(\theta_i),$$

또는

$$A\mathbf{l} = \mathbf{b}, \quad (2)$$

여기서 \mathbf{A} 는 $S(\theta_i, \phi_i) \cos(\theta_i)$ 로 이루어진 $m \times n$ 행렬이고, \mathbf{l} 는 $n \times 1$ 의 벡터로 $\frac{L(\theta_i, \phi_i)}{\sum_{j=1}^n L(\theta_j, \phi_j) \cos(\theta_j)}$ (광도 비율(illumination radiance ratio))로 이루어져 있고, \mathbf{b} 는 두 영상의 화소값(pixel value)의 비율로 이루어져 있다. 이 식에서 광도 비율은 SVD(Singular Value Decomposition) 방법([6])에 의해서 선형적으로 계산할 수 있다. 이 광도 비율을 통해 4장에서 설명된 대로 가상 물체의 셰이딩 및 그림자를 생성할 수 있다. 주의할 점은 실제로 영상 $P'(x_k, y_k)$ 은 5장에서와 같이 그림자를 투사하는 물체를 제거한 후에 얻어진다.

3 비디오 영상으로의 확대

2장에서 알수 있듯이 카메라 칼리브레이션은 빛의 분포 측정을 위해서는 필수적이다. 그러나 일반적으로 비디오 영상에 가상 물체의 삽입, 셰이딩 및 그림자 생성을 위해서 필요한 비디오 카메라 칼리브레이션은 어려운 문제이다. 이를 위해서 본 논문에서는 virtual camera 개념을 도입한다([2]). 이 virtual camera는 이상적으로는 실제 비디오 카메라의 움직임에 따라서 똑같이 움직인다.

3.1 Projective Motion and Structure

비디오 영상에서 영상간에 특징점(x_{ki}) 대응점(correspondence)이 존재하면, projective motion(\mathbf{P}_k)과 structure(\mathbf{X}_i)를 구할 수 있고([7, 8]) 다음과 같은 식을 만족한다.

$$\lambda_{ki} x_{ki} = \mathbf{P}_k \mathbf{X}_i, \quad \text{for } i = 1, \dots, M, \quad k = 1, \dots, N, \quad (3)$$

여기서 λ_{ki} 는 scale factor를 나타낸다. 다음 장에서는 projective motion과 structure를 이용하여 graphic world coordinate system을 비디오 영상에 효과적으로 삽입(embedding)할 수 있는 방법을 설명한다.

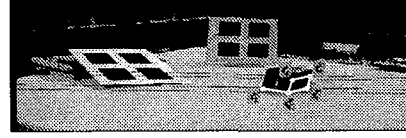
3.2 삽입(Embedding) 과정

Graphic world coordinate system을 비디오 영상에 삽입하기 위해서, 우선 graphic world coordinate system의 5개의 기저점(basis points)을 실험에 쓰인 비디오 영상으로부터 추출된 두 제어 영상(control image), \mathbf{I} 과 \mathbf{I}' ([2, 9])에 먼저 삽입을 한다. 삽입 단계는 아래와 같은 두 단계를 거친다:

1. Graphic world coordinate system의 5 개의 기저점($\{\mathbf{E}_0, \mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4\}$)을 첫번째 제어 영상(\mathbf{I})에 찍어준다($\{x_i^0\}_{i=0}^4$): 그림. 1(a) 참조



(a) 첫번째 제어 영상



b) 두번째 제어 영상

그림 1: 두 제어 영상에 5개의 기저점 삽입

2. Epipolar geometry를 이용하여 4개의 기저점($\{\mathbf{E}_0, \mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$)을 두번째 제어 영상(\mathbf{I}')에 찍어준다($\{x_i^1\}_{i=0}^3$): 그림. 1 (b) 참조

그림. 1에 삽입 과정을 나타냈다. 두 제어 영상은 실험에 사용된 비디오 영상에서 추출했다. 주의할 점은 두번째 제어 영상에서는 5 번째 기저점(x_4^1)을 따로 찍어 줄 필요가 없다. 왜냐하면 5 번째 기저점은 2D Homography에 의해서 자동적으로 결정되기 때문이다. 5 개의 기저점이 두 영상에 삽입되었으면, 이 대응점을 이용하여 3차원 projective coordinate를 구할 수 있다. 그러면 나머지 영상에서(두 제어 영상을 제외한)의 기저점의 위치는 3.1장에서 이미 구해진 projective motion(\mathbf{P}_k)에 의해서 자동적으로 결정된다.

지금까지 graphic world coordinate system을 전체 비디오 영상에 삽입하는 과정을 설명했다. 주의할 점은 graphic world coordinate system이 이상적으로 삽입되었다면 실제 world coordinate와 차이점이 없으나, 실제로는 [2]에서 언급되었듯이 projective distortion이 존재할 가능성이 있다.

3.3 Virtual Camera 계산 및 분해

전체 비디오 영상에 삽입된 5개의 기저점을 가지고(기저점의 3차원 좌표는 주어짐), 3-2차원 매핑 관계(\mathbf{E}_i, x_i^k)를 이용하여 zero-skew virtual camera(\mathbf{P}_k^V)를 계산한다([2]). 이렇게 얻어진 virtual camera를 셰이딩 및 그림자 생성에 필요한 그래픽 카메라로 사용하기 위해서는 아래와 같은 분해(decomposition) 과정이 필요하다.

$$\mathbf{P}_k^V \sim \mathbf{K}_k [\mathbf{R}_k | \mathbf{t}_k], \quad (4)$$

여기서 \mathbf{K}_k 는 3×3 zero-skew 칼리브레이션 행렬(calibration matrix), \mathbf{R}_k 는 3차원 회전, \mathbf{t}_k 는 전이(translation)에 각각 해당한다.

지금까지 전체 비디오 영상 각각에 대해서 virtual camera를 계산하였다. 한가지 보충할 점은 삽입된 graphic world coordinate system에 의해서 간단한 형태의 물체, 예를 들면 그림. 1의 박스,의 크기를 결정할 수 있다는 것이다. 이것을 통해 2장에서 필요한 그림자를 투사하는 물체의 형태 정보를 얻을 수 있다.

4 셰이딩(Shading) 및 그림자 생성

가상 물체의 셰이딩과 그림자 생성은 [1]에서와 같은 방식으로 수행한다. 차이점은 전체 비디오 영상에 대해 각각의 virtual camera를 그래픽 카메라로 대치하여 수행한다는 점이다. 본 논문에서는 OpenGL 환경에서 복수의 광원(multiple light sources)을 가지고 soft shadows generation technique을 적용하여 그림자를 생성한다([10]). 그림자 생성을 위한 식은 아래와 같다.

$$P(x_k, y_k) = \frac{E(x_k, y_k)}{E'(x_k, y_k)} P'(x_k, y_k), \quad (5)$$

여기서 $E(x_k, y_k)$ 는 그림자를 투사하는 물체가 존재할 때를 가정하고 만들어진 영상의 화소값을 $E'(x_k, y_k)$ 는 물체가 존재하지 않을 때의 영상의 화소값을 나타낸다. $P'(x_k, y_k)$ 는 가상 물체를 삽입하고자 하는 실제 영상이고 이 영상에 그림자가 만들어진다. 따라서 최종적으로 $P(x_k, y_k)$ 에 셰이드된 가상 물체를 삽입하면 그림자가 존재하는 가상 물체가 삽입된 영상을 얻을 수 있다.

5 실험 결과

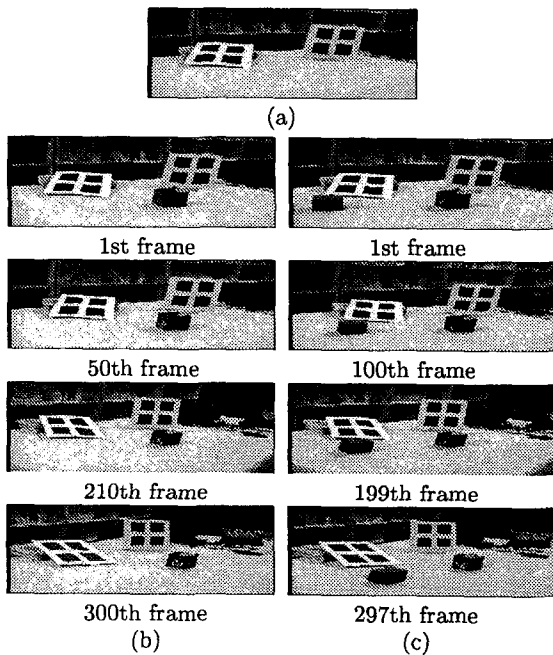


그림 2: (a) 배경 영상. (b) 예제 비디오 영상. (c) 하나의 가상 물체 삽입

그림. 2(b)에 실험 비디오 영상을, (a)에 그림자를 투사하는 물체를 제거한 영상, 즉 배경 영상을 나타냈다. 실험 비디오 영상은 전경에 거리가 떨어진 두 개의 평행한 형광등의 조명하에서 얻어졌다. 배경 영상은 비디오 카메라를 고정시킨 후에 물체를 제거한 후의 영상에 해당하고, 전경 영상(1st frame)은

카메라를 그대로 고정 시킨 상태에서 물체를 삽입하여 그림자가 보이게 한 후에 얻어졌다. 그 다음에는 카메라를 자유 자재로 움직이면서 영상을 획득했다. 또한, 특징점의 추적을 용이하게 하기 위해 평면 패턴을 삽입했다. 그림. 3에 배경 영상과 전경 영상을 표시했는데 전경 영상의 사각형 표시는 광도 비율(illumination radiance ratio)을 구하기 위해서 사용된 화소 영역을 표시한다. 식 (2)에 의해서 구해진 광도 비율을 그림. 4(a)에 표시했고, 하나의 가상 물체를 삽입한 결과를 그림. 2(c)에 표시했다. 그림. 4(a)를 살펴보면 북극 부분 근처에 두 개의 밝은 지역이 존재하는데 이것이 실제 두개의 형광등에 해당하는 것이고 또한 하얀 사각형에 싸여져 있는 다른 밝은 부분이 존재함을 확인할 수 있는데 결과적으로 이 밝은 부분은 광도 비율을 구할 시에 전경 영상만을 이용해서 나타나게 된다. 즉, 전경 영상에는 그림자를 투사하는 물체에 의해서 가려지는 평면 부분이 존재하는데 여기에 그림자가 존재하는지 없는지 알 수 없기 때문에 모호성(ambiguity)이 발생하게 된다. 이런 문제점은 [1]와 같이 정지 영상을 사용하는 방식에서 항상 나타나는 것으로 이것으로 인해서 카메라가 움직이게 될 시에 잘못된 그림자(false shadows)가 보이게 된다(그림. 5 참조).

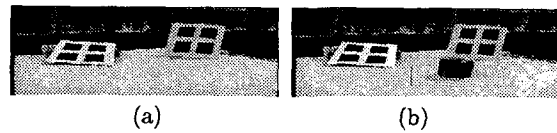


그림 3: (a) 배경 영상. (b) 전경 영상.

5.1 비디오 영상을 이용한 향상점

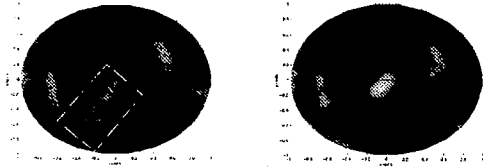
비디오 영상에서는 전경 영상에 존재하는 그림자가 투사되는 평면의 가려진 부분이 다시 보일 수가 있다. 따라서 2D Homography를 이용하면(실제로 virtual camera로부터 얻어짐) 가려진 부분에 해당하는 화소값을 가려진 부분이 다시 나타나는 다른 복수의 영상들로부터 얻을 수 있다. 본 논문에서는 잡음(noise)을 줄이기 위해서 복수 영상에서 얻어진 화소값의 평균을 하여 전경 영상의 평면의 가려진 부분의 화소값으로 취환했다. 이것을 복구된 전경 영상(recovered foreground image)이라 하고 이를 그림. 6에 표시했다. 이 복구된 전경 영상을 광도 비율을 계산하기 위한 식 (2)에 사용한다.

식 (2)을 계산하기 위해 사용한 SVD 방법을 이용하면 비디오 영상에 존재하는 잡음에 의해서 계산된 광도 비율 값이 음수(negative value)값을 가질 수 있다. 실제로 계산된 광도 비율에도 상당히 많은 음수의 광도 비율이 검출됨을 알 수 있다. 그러나 자연계에 존재하는 광원은 그 특성상 항상 양수의 광도(radiance)값을 가지게 된다. 따라서 이를 보정하기 위해서 본 논문에서는 비선형적인 constrained optimization 형태로 식 (2)을 아래와 같이 변환한다.

$$\min_l (Al - b)^2, \quad \text{constrained with } l \geq 0 \quad (6)$$

그림. 4(b)에 복구된 전경 영상과 식 (6)을 이용해서 구해진 광도 비율을 나타냈다. 예상 했던 대로 실제

형광등에 해당하는 부분은 그대로 존재하나 그림. 4(a)에서 보이던 사각형 안의 밝은 영역이 없어짐을 알 수 있다. 한가지 주의할 점은 북극(north pole) 부분에 존재하는 밝은 영역은 배경 영상과 전경 영상의 평균적 밝기 차이를 보정하기 위해서 존재하는 것으로 실제 두 영상의 밝기를 비교하면 밝기의 차이가 남을 알 수 있다. 마지막으로 그림. 7에 두 개의 가상 물체가 삽입된 결과를 보인다.



(a) Linear solution by SVD (b) Constrained optimization

그림 4: (a)광도 비율: 북쪽에서 내려다본 geodesic dome.

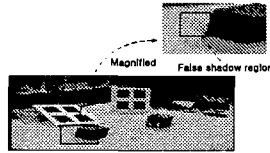


그림 5: 그림. 2 (c)의 297th frame 확대



그림 6: 복구된 전경 영상

6 결론

본 논문에서는 [1]의 방법을 비디오 영상에 확장하여 보다 향상된 증강 현실 구현을 할 수 있는 방법을 제안했다. 비디오 카메라 칼리브레이션을 위해 virtual camera를 사용했고, 또한 기존 방식의 제한점을 향상시켰다. 끝으로, 본 논문에 더해서 앞으로 연구되어야 할 부분을 언급하고 본 논문을 마친다. 1) Projective distortion of virtual camera. 2) 비디오 영상에서 삽입된 물체의 떨림 현상.

References

[1] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from shadows. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 306-312, 1999.

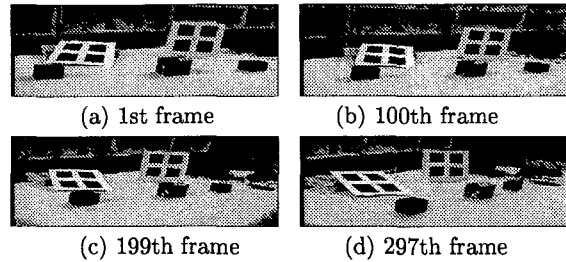


그림 7: 두개의 가상 물체가 삽입된 결과에.

[2] Yongduek Seo and Ki-Sang Hong. Calibration-free augmented reality in perspective. in print for *IEEE Trans. Visualization and Computer Graphics*.

[3] Ronald T. Azuma. A survey of augmented reality. *PRESENCE: Teleoperations and Virtual Environments*, 6(4):355-385, August 1997.

[4] K. Kutulakos and J. Vallino. Affine object representations for calibration-free augmented reality. In *IEEE Virtual Reality Annual International Symposium*, pages 25-36, 1996.

[5] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proc. Int. Conf. on Computer Vision*, pages 875-882, 1999.

[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[7] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Computer Vision - ECCV92*, volume 588 of *Lecture Notes in Computer Science*, pages 563-578. Springer Verlag, 1992.

[8] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, INRIA, Sophia Antipolis, France, 1996.

[9] Yongduek Seo. *Non-Metric Augmented Reality and Flexible Auto-Calibration*. PhD thesis, Department of Electronic and Electrical Engineering, POSTECH, Korea, 2000.

[10] OpenGL ARB. *OpenGL Programming Guide*. Addison Wesley, 1993.