

증강현실 개발 툴킷 프로세스 및 성능 분석

심진욱^{○*} 한탁돈^{*}

*연세대학교 컴퓨터과학과

Jin99foryou@msl.yonsei.ac.kr, hantack@msl.yonsei.ac.kr

Augmented Reality Development ToolKit Process & Performance Analysis

Jinwook Shim^{○*} Tackdon Han^{*}

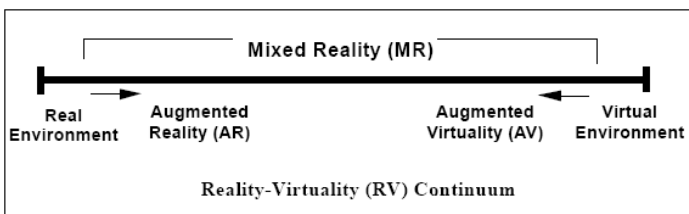
*Dept. of Computer Science, Yonsei University

요 약

본 논문에서는 증강현실 기술에 대하여 살펴보고, 워싱턴 대학교에서 개발한 프로그래밍 라이브러리의 형태인 ARToolKit과 오스트리아, 그라츠 대학교에서 개발한 ARToolKit PLUS에 대하여 각 툴킷의 프로세스 및 성능에 대하여 분석하고자 한다. 가상현실 기술은 실 세계와 가상 세계를 실시간으로 혼합하여 사용자에게 제공함으로써 정보 사용의 효율성과 효과성을 극대화하는 기술이며, 이는 향후 IT기술 전문가의 발전과 변화에 많은 영향을 줄 '주목해야 할 기술'이다. 가상현실 기술이 사용자들에게 쉽게 받아들여지고, 보다 적극적으로 널리 활용되기 위해서는 기술적 한계의 극복과 가상현실에 특화된 사용자 상호작용 기술 개발 및 응용 서비스 창출 등이 요구된다.

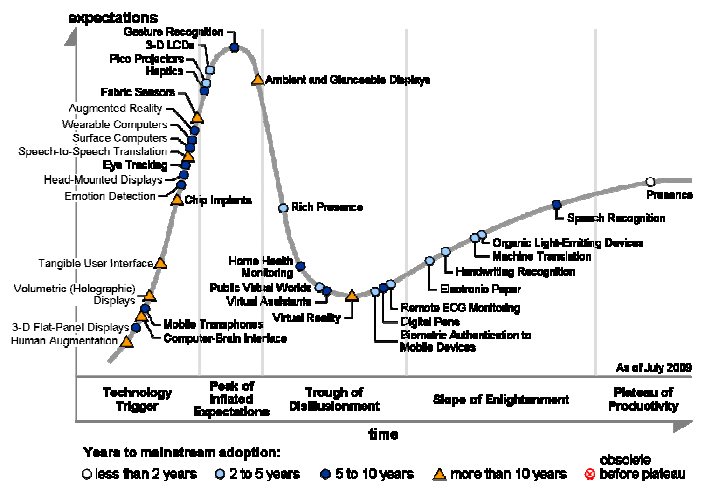
1. 서 론

증강현실(Augmented Reality: AR)은 실제 환경에 가상으로 생성한 정보를 이음새 없이(seamless) 실시간으로 혼합하여 사용자와 상호작용 하도록 함으로써, 사용자에게 보다 향상된 몰입감과 현실감을 제공하여 정보의 사용성과 효용성을 극대화하는 차세대 정보처리 기술이다.[1] 증강현실은 가상현실과 TV영상과 같은 현실의 중간에 위치하는 기술로, 가상현실(Virtual Reality: VR)과 같이 가상성에 바탕을 두고 있으나, 가상현실이 컴퓨터가 구축한 가상공간 속에 사용자를 몰입하게 하는 기술이다. 반면, 증강현실은 사용자의 실제 환경에 가상의 정보를 더해줌으로써 실제감을 향상시키는 기술이다. 가상현실 기술이 실제환경을 컴퓨터가 생성한 환경과 완전히 대체하는 것이라면 증강현실 기술은 사용자가 가지고 있는 기존의 실제환경 정보를 유지한다는 점에서 차이점을 지닌다. 그림 1은 실세계와 가상세계의 연속성 상에서의 증강현실의 위치를 보여준다. [2]



[그림 1]. 실세계 환경과 가상세계 환경의 연속성

최근 Gartner사는 그림 2에서 보이는 바와 같이 연례 기술 분석 보고서에서 유망 기술의 하이프 곡선상의 '유발 기술(Technology Trigger)' 부문에 혼합현실 기술을 위치시키고 있으며, 향후 IT 기술 전문가의 발전과 변화에 미칠 파급효과가 매우 큰 '주목해야 할 기술'로 표현하고 있다. [3]



[그림 2]. 유망 기술의 하이프 곡선(Hype cycle for emerging technologies 2009)

증강현실 기술은 사용자의 위치와 주변 환경의 정보를 추적하고, 추적 결과에 따라 적합한 영상 정보를 자연스럽게 배치하여, 사용자에게 디스플레이하는 3가지의 요소 기술로 구성된다. 먼저 정보의 입력

단계인 ‘추적’에서는 사용자의 현재 위치, 응시하는 방향, 접촉하여 조작하는 물체 등에 대한 모든 정보를 실시간으로 추적한다. 입력된 정보의 처리 단계인 ‘배열’에서는 추적을 통해 입력된 실세계의 정보들에 가상의 정보를 결합하여 자연스럽게 정확하게 정렬한다. 마지막으로 입력, 처리된 정보의 출력 단계인 ‘디스플레이’에서는, 실세계의 정보와 가상의 정보를 사용자의 눈앞에 실감나게 결합하여 보여주게 된다.[4]

2. 관련 연구

이러한 증강현실을 표현하기 위해 사용되는 대부분의 개발 툴킷은 컴퓨터 비전 기술을 이용하고, 특히 마커를 기반으로 동작한다. 현재 컴퓨터 비전 기술을 이용한 개발 툴킷을 이용한 연구는 데스크탑 환경뿐 아니라, 모바일 환경에서 보다 정확하고 현실감 있는 객체 증강을 위하여 현재 카메라 포즈에 대한 정확한 분석과, 다양한 방법을 통하여 정확도와 속도를 향상시키기 위한 방법에 대한 연구가 증가하고 있다.

마커를 기반으로 동작하는 증강현실 개발 툴킷의 특징은 대표적인 사각형의 마커가 실세계와 가상 세계간의 기준 좌표계 역할을 한다. 이 마커를 이용하여 상대적 좌표를 추출하고 이를 활용하여 정합 및 추적을 가능케 하기 때문에 여러 방향과 거리에서 인식을 용이하게 한다. 하지만, 조명의 종류 및 밝기가 마커 검출에 영향을 미치기도 하고, 손이나 그림자로 마커의 일부를 가리면 마커 검출에 영향을 미치기도 하는 단점이 있다.

이러한 대표적인 마커기반 증강현실 개발 툴킷으로는 미국의 워싱턴 대학교의 HIT lab.에서 개발한 ARToolKit이 있다. ARToolKit은 마커 기반의 증강현실 툴킷을 제일 먼저 공개하여 가장 광범위하게 이용되고 있고, 마커 검출은 템플릿 매칭을 이용한다. 실제로 프로그래밍 라이브러리의 형태인 ARToolKit이 개발된 이래 수많은 증강현실 콘텐츠가 이 툴킷을 기반으로 제작되었다.[5]

두번째로 캐나다, NRC에서 개발한 ARTag는 ARToolKit의 취약점인 가려짐으로 인한 불안정성 문제를 극복했다. 코드 인식 기술을 기반으로 안정된 인식 성능 보장과 제한된 수의 마커를 제공한다. 하지만, 마커 기반 툴킷이기 때문에 역시 가려짐 정도가 심해질 경우 불안정성 문제는 존재한다.[6]

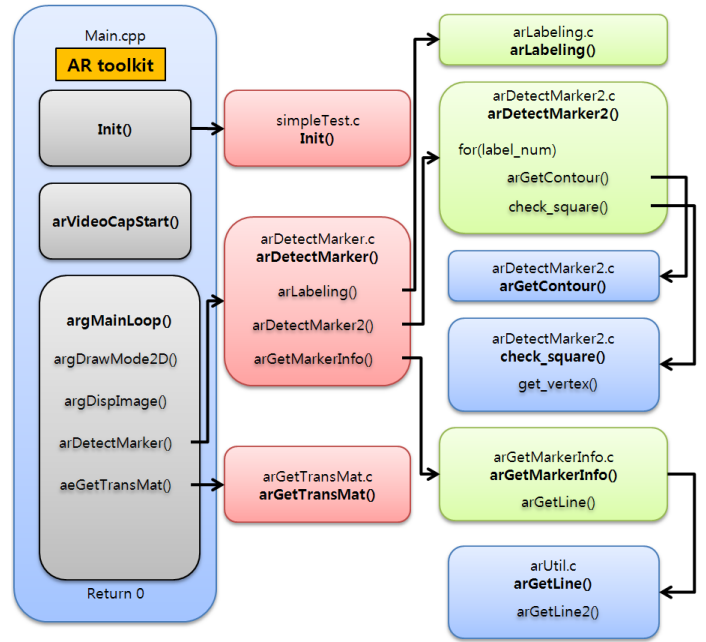
다음으로 오스트리아, 그라츠 대학교(Graz Univ.)에서 개발한 ARToolKit PLUS는 4096개까지의 ID기반 마커를 제공하지만, 다양한 형태의 마커 기반 인식을 제공한다. 또한, 마커의 가로 길이를 가변적으로 할 수 있다.[7]

3. ARToolKit 과 ARToolKit PLUS 프로세스 다이어그램

위에 설명한 마커기반 증강현실 개발 툴킷 중 대표

중 워싱턴 대학교 HIT Lab.에서 개발한 ARToolKit과 그라츠 대학교에서 개발한 ARToolKit PLUS의 샘플 예제인 simpleTest를 중심으로 프로세스를 간략히 설명하고 각 툴킷의 성능을 간단히 평가해보고자 한다.

3.1 ARToolKit 프로세스 다이어그램



[그림 3] ARToolKit 프로세스 다이어그램

그림 3은 ARToolKit의 SimpleTest.c 프로세스의 간략한 다이어그램이다. 초기 init()함수는 카메라의 파라미터를 셋팅하고 비디오 경로, 윈도우 창, 그래픽 윈도우 설정 등 초기 증강현실 개발 툴킷을 동작시키기 위한 초기 전처리 과정을 수행한다.

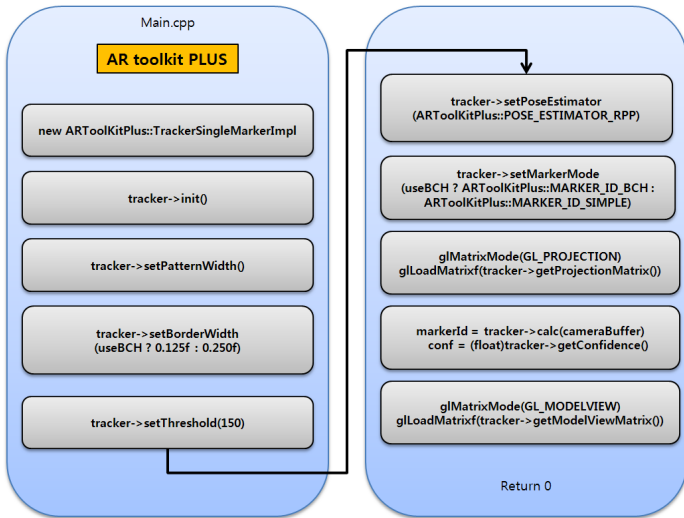
argMainLoop()함수는 가장 핵심인 함수로 컴퓨터에 연결된 웹캠으로부터 비디오 프레임을 가지고 와 arDetectMarker()함수로 비디오 프레임을 보내 웹캠으로 받아들이는 영상에서 마커를 찾는 역할을 수행한다. arLabeling()함수에선 입력된 영상에서 마커 영역에 대한 Labeling함수를 수행해 마커로 판단되는 영역에 대해 arDetectMarker2()함수를 수행한다.

arDetectMarker2()함수에서는 영상에서 마커라고 판단되는 사각형이 있는 라벨에 대하여 외곽선을 찾고, 사각형임을 판별하게 된다. 판별된 사각형 위치 및 면적 정보를 이용하여 arGetMarkerInfo()함수에 의해 영상에서 찾은 마커의 정보 예를 들어, 라벨링된 영역 내의 픽셀 개수, 마커의 중앙좌표값, 외곽선내의 픽셀 개수, 외곽선 픽셀의 x좌표, y좌표, 마커 점들의 위치가 저장되고, arGetCode()함수를 통해 마커 내부의 패턴 일치 여부를 판단 후 ID값 및 회전 정보, 마커일 확률을 얻게 된다.

그 후, arGetTransMat()함수를 통해 실제 카메라와

마커 사이의 위치를 판별하여, 입력받은 카메라 영상에 가상의 물체를 표현하여 준다.

3.2 ARToolKit PLUS 프로세스 다이어그램



[그림 4] ARToolKit PLUS 프로세스 다이어그램

그림 4은 ARToolKit PLUS의 Simple 프로세스의 간략한 다이어그램이다. ARToolKit PLUS는 ARToolKit과는 달리 객체지향 구조로 개발되었으며, ARToolKit을 발전시킨 개발 툴킷이지만, ARTag 방식의 셀을 기반으로 한 마커를 사용하고 있다. 또한, ARToolKit PLUS는 ID값 error corrector로 BCH 알고리즘을 사용한다.[7]

ARToolKit PLUS 초기에 기본적인 설정을 결정하고, TrackerSingleMarkerImpl()클래스를 이용하여 6x6 마커 이미지와 최대 6x6의 샘플, 그레이 이미지, 1가지 패턴, 한 이미지에서 찾을 수 있는 최대 패턴의 개수를 설정한다. init()함수는 ARToolKit과 마찬가지로 기본적인 카메라의 파라미터를 셋팅하고 비디오 경로, 윈도우 창, 그래픽 윈도우 설정 등 초기 증강현실 개발 툴킷을 동작시키기 위한 초기 전처리 과정을 수행한다.

setPatternWidth()함수는 마커의 기본 사이즈를 결정하고, setBorderWidth()함수는 초기 설정한 BCH 함수 사용 여부에 따라 마커의 테두리를 판별하는 기본 값을 결정한다. setThreshold()함수는 컴퓨터에 연결된 웹캠으로부터 비디오 프레임을 가지고 와 마커를 찾을 때 이진화를 위한 임계값을 설정한다.

setPoseEstimator()함수에서는 카메라와 마커 사이의 위치를 판별하여 입력받은 카메라 영상에 가상 물체의 위치를 계산해주는 함수인데, ARToolKit PLUS에서 사용되는 알고리즘은 RPP를 사용한다. 이는 기존 ARToolKit에서 사용하는 pose estimator보다 강건하게 물체의 위치를 계산한다.[7]

setMarkerMode()함수는 BCH 알고리즘의 사용여부에

따라 심플마커인지, 멀티마커인지 마커의 ID값을 찾을 방법을 설정한다.

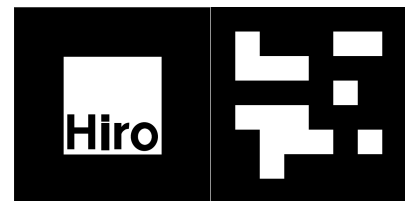
glMatrixMode(), glLoadMatrixf()함수는 마커를 찾았을 경우, 마커 ID값에 해당하는 가상의 물체를 표현하기 위한 OpenGL함수이다.

calc()와 getConfidence()함수는 실제 카메라로부터 받은 영상에서 마커를 찾아 마커의 ID값을 알아내고, 정확한 마커의 ID값을 찾았는지 신뢰도에 대한 함수이다.

4. ARToolKit 과 ARToolKit PLUS 프로세스 성능분석

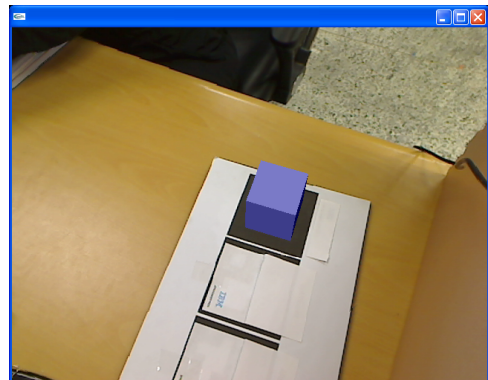
위에 설명한 워싱턴 대학교 HIT Lab.의 ARToolKit과 오스트리아, 그라츠 대학교의 ARToolKit PLUS에 대한 프로세스 성능 분석은 기본적으로 마커의 ID값을 판단하는 방법에서 템플릿 매칭과 셀 기반 코드인식 방법으로 차이가 있다. 아래에서 하고자 하는 프로세스 성능 분석은 카메라로부터 영상을 입력받아 마커의 ID값을 찾고 가상의 물체를 표현하는데 걸리는 프로세스의 동작 시간은 입력받는 영상 1프레임을 기준으로 측정하였다.

실험에 사용한 마커는 ARToolKit과 ARToolKit PLUS에서 기본으로 제공하는 마커(그림 5)를 사용했으며, 크기는 8x8 Cm이다. 카메라와 마커와의 거리는 직선 거리로 50Cm정도에서 실험한 결과이다.



[그림 5] ARToolKit과 ARToolKit PLUS 마커

4.1 ARToolKit 성능 분석



[그림 6] ARToolKit 동작 화면

그림 6은 ARToolKit의 기본적인 예제인 simpletest를 실행한 화면이다. simpletest를 정확한 동작이 수행되는 상황에서 1000프레임동안 동작시키며 전체 프로세스 및

각 함수 별 수행시간을 측정하였다. 위 그림 3에서 초기화 예를 들면, 카메라 파라미터를 셋팅하는 등의 단계는 프로세스 수행시간을 측정하는 부분에서 제외시켜, 순수 마커를 찾고, 가상의 물체를 표현하기 위한 매트릭스 계산 부분을 중심으로 측정하였다.

ARToolKit에서는 크게 3부분으로 나누어 측정하였다. 우선적으로 argMainLoop()함수에 대하여 1000프레임동안 측정하여 마커를 찾고, 가상의 물체를 표현하는 전체 프로세스에 대한 시간을 측정하였고, 표1에 간략하게 작성하였다. 평균적으로 메인 프로세스가 수행되는 시간은 12.7637ms 정도의 시간이 걸렸고, 그 중에 마커를 찾는 arDetectMarker()함수에서 대략 6.94495ms 정도의 시간이 걸렸다.

가상의 물체를 표현하기 위해 필요한 매트릭스를 계산하기 위한 arGetTransMat()는 2.42567ms 정도의 시간이 걸렸다.

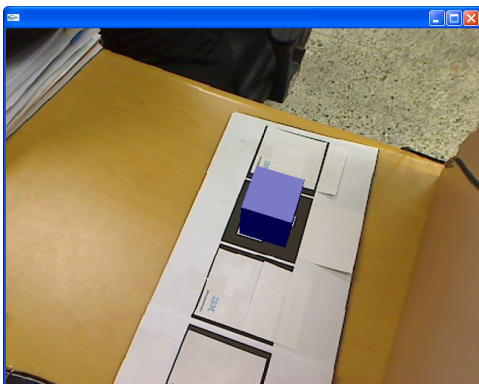
argMainLoop()함수의 argDrawMode2D(), argDisplmage() 함수에서 평균적으로 3.393067ms 정도의 시간을 할애한 것을 알 수 있다. 위 두 함수들은 MainLoop가 수행되면서 마커를 찾게 되면 해당 마커 ID에 맞는 가상의 물체를 표현하는데 쓰이는 함수이다.

프레임	Main(ms)	Detect(ms)	Trans(ms)
1frame	13.47441	7.08321	2.53844
2 frame	13.27376	6.68804	2.24692
~	~	~	~
1000 frame	12.42655	6.98796	2.37741
평균	12.7637	6.94495	2.42567

Main = argMainLoop(), Detect = arDetectMarker(), Trans = arGetTransMat()

[표 1] ARToolKit 프로세스 측정 테이블

4.2 ARToolKit PLUS 성능 분석



[그림 7] ARToolKit PLUS 동작 화면

그림 7은 ARToolKit PLUS의 기본적인 예제인 simple을 실행한 화면이다. simple을 정확한 동작이 수행되는 상황에서 1000프레임동안 동작시키며 전체 프로세스 및 각 함수 별 수행시간을 측정하였다.

프레임	Main(ms)	Calc(ms)	Mode(ms)
1frame	21.37508	1.08385	0.001623
2 frame	21.59778	1.10314	0.001227
~	~	~	~
1000 frame	21.22032	1.07386	0.000995
평균	21.28772	1.08630	0.001092

Main = init()~return, calc = tracker->calc(), Mode = setPatternWidth()~setMarkerMode()

[표 2] ARToolKit Plus 프로세스 측정 테이블

위 표2에서와 같이 ARToolKit PLUS를 3부분으로 나누어 측정하였고, Main은 ARToolKit PLUS의 tracker클래스의 생성부터 가상의 물체를 표현하는 calc() 함수까지 전체 프로세스를 측정한 항목, calc는 ARToolKit과 마찬가지로 마커를 찾아 가상의 물체를 표현하는 제일 중요한 함수를 측정한 항목이다. 그리고 Mode는 초기화 단계로 마커의 사이즈, BCH 알고리즘 사용여부, pose estimator 함수등 기본적인 사항을 결정하는 함수들을 포함한다.

우선적으로 Main 전체 프로세스에 대하여 1000프레임동안 측정하여 평균적으로 전체 프로세스가 수행되는 시간은 21.28772ms 정도의 시간이 걸렸고, 그 중에 calc()함수에서 대략 1.08630ms 정도의 시간이 걸렸다. 초기 마커의 사이즈나 기타 정보를 설정을 하는 함수들 부분에선 0.001092ms 정도의 시간이 걸렸다.

위 결과를 보면 ARToolKit PLUS는 초기 카메라 파라미터나 클래스 생성자 등에서 대부분의 프로세스 시간을 할애 했으며, 초기화 단계가 마친 후에는 ARToolKit의 arDetectMarker() 보다 ARToolKit PLUS의 calc()가 수행 속도가 더 빠른 것을 알 수 있다.

5. 결론

두 개발 툴킷 모두 30fps 정도의 성능을 유지했지만, 현재 증강현실 기술은 아직 이음새 없이(seamless) 현실감있게 표현하기에는 아직 더 많은 연구가 필요하고, 또한 현재 모바일 디바이스에서 증강현실을 동작시키기 위한 많은 데이터들이 있지만, 데스크탑에서 동작할 수준의 성능을 모바일에서 유지하기 위해서는, 초기 영상에서 마커를 찾는 단계나, 마커와 카메라의 위치를 계산하여 가상의 물체를 표현하는 pose estimator, 가상의 물체가 흔들리는 jittering문제 등 기타 해결해야 할 문제가 아직 산재해 있다. 위 증강현실 기술이 사용자들에게 쉽게 받아들여지고, 보다 적극적으로 널리 활용되기 위해서는 현재 증강현실 개발 툴킷뿐만 아니라 보다 기술적 한계의 극복과 가상현실에 특화된 사용자 상호작용 개발 툴킷 및 응용 서비스 창출 등이 요구된다.

6. 참고 문헌

- [1]Azuma, R.T. A survey of augmented reality. In presence: Teleoperators and virtual Environment 6(4), pp355-385, 1997.
- [2]Milgra,p., &Keshino,F. A taxonomy of mixed reality visual display. IEICE transactions on Information and Systems, E77-D,12, 1321-1329., 1994.
- [3] Gartner, <http://gartner.com/>
- [4]김정현, 계보경, 서진석, 고범석, 증강현실 기반의 체험형 학습 콘텐츠 개발 및 현장적용 연구. 한국교육학술정보원, 2005
- [5]ARToolKit, <http://hitl.washington.edu/artoolit>.
- [6]ARTag, <http://www.artag.net/>
- [7]ARToolKitPLUS,http://studierstube.ieg.tu-graz.ac.at/handheld_ar/artoolkitplus.php