모바일 그래픽스 라이브러리들: 크로노스 그룹의 표준을 중심으로

Mobile Graphics Libraries: focusing on the standards from Khronos Group

백낙훈 (경북대학교), 이환용 (㈜휴원 CTO, 경북대학교 박사과정)

차 례

1. 서론
2. OpenGL ES
3. OpenGL SC
4. OpenVG
5. 결론

1. 서론

컴퓨터는 그 출현 이후, 계속해서 그 크기를 줄여 왔고, 현재는 휴대폰, 네비게이터, 휴대용 게임기 등의 형태로 모바일 컴퓨팅 환경을 제공하고 있다. 모바일 컴퓨팅 환경의 대두와 함께, 이를 이용한 그래픽스 출력이 요구되었고, 모바일 환경에서의 그래픽스 기능을 의미하는, 모바일 그래픽스 환경이 구축되었다. 이 글에서는 현재의 모바일 그래픽스 환경 중에서도 특히 표준 라이브러리들에 대해 이야기하고자 한다. 지난 10여년 사이에 모바일 그래픽스 분야는 눈부신 발전을 이루었다. 가장 대표적인 기기인 휴대폰을 기준으로 본다면, 최초에 나온, 저해상도에 흑백의 투박한 액정을 가진 모델이 불과 몇 년 사이에 상당한 수준의 컬러 액정을 가진 제품으로 변했고, 3차원 그래픽스를 지원하는 특별한 하드웨어 칩들을 사용하는 것도 이상하지 않은 상황이 되었다. 이에 따라, 소프트웨어 기술들도 눈부시게 발전하여, 그림 1에서 보는 바와 같이, 2차원 저해상도에서 3차원 화면을 거쳐, 거의 PC 화면에 필적하는 수준의 고해상도 3차원 화면이 급속히 사용되고 있다. 모바일 그래픽스 환경은 태생상, 기존의 데스크 탑 또는 그 이상의 고급 기종들에 비해, 상대적으로 열등한 하드웨어 환경을 제공할 수 밖에 없다. 가장 문제가 되는 것은 제한된 용량의 전원, 즉 배터리의 수명을 고려해야 한다는 점이다. PC급의 고해상도 액정이 기술적으로 문제가 있는 것이 아니라, 채택하게 되면 겪게 될 짧은 배터리 수명 때문에, 저해상도의 전력 소모가 낮은 기기를 채택할 수 밖에 없는 것이다. 액정뿐만 아니라, 모든 컴퓨팅 자원들이 궁극적으로 전력 문제에 민감하여야 하고, 이 때문에, 제한된 기능만을 제공할 수 밖에 없는 상황이다.

(a) 저해상도 2D (b) 3D

(c) 하드웨어로 가속되는 3D

▶▶ 그림 1. 모바일 그래픽스 분야의 발전 과정

이러한 제약은 모바일 그래픽스 시장이 기존의 데스크탑 그래픽스 시장과는 조금 다른 양상을 띠게 만들었다. 데스크 탑 시장에서 우위를 점하고 있던 마이크로소프트의 다이렉트 X가 플랫폼의 문제와 다른 요인들로 별다른 우위를 점하지 못하고 있고, 하드웨어 측면에서는 nVIDIA나 ATI의 그래픽스 칩들이 그다지 매력적이지 않은 상황이 벌어지고 있다. 애플의 아이폰(iPhone)이 출시되기 이전에는 심지어 그래픽스 전용 프로세서(GPU)의 채택이 필요한 지에 대해서도 논란이 벌어질 여지가 있는 상황이었으나, 아이폰이 보여준 GUI의 품질을 통해, 역설적으로 모바일 그래픽스 기술에 대한 요구가 커졌고, 하드웨어나 소프트웨어 기술들도 한층 업그레이드되는 시점에 와 있다.

모바일 그래픽스에 사용되는 GPU 시장은 아직까지 절대 강자가 없으며, 관련 기업들 간의 인수합병이 경쟁적으로 이루어지고 있는 상황이다. 2010년 상반기의 현 시점에는 표 1에서와 같이, 대략 4~5개의 주요 GPU 업체들이 주축을 이루고 있고, Vivante, DMP, Nexus 등의 기타 군소 업체들이 가세해 있는 상황이다.

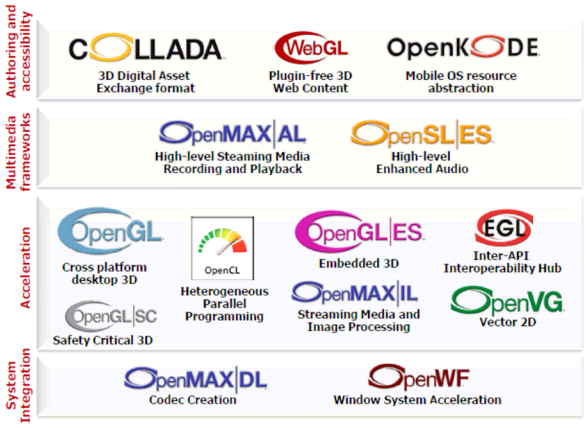
표 1. 주요 GPU 생산 업체

업체명	GPU 라인	특징
QualComm	Adreno	AMD 모바일 사업부를 인수
ARM	Mali	Falanx사를 인수
nVIDIA	Tegra	우수한 성능
Imagination	SGX	라이센싱에 성공적
Samsung	AP	자체 core 개발

▶▶ 그림 2. 크로노스 그룹 회원 현황

현재 모바일 그래픽스 관련 하드웨어나 소프트웨어를 제공하는 거의 모든 업체들은 예외 없이 크로노스 그룹(Khronos Group)[1]의 API 표준을 채택하고 있다. 사실상 이 분야에서는 크로노스 그룹이 표준을 제정하고, 시장의 방향에 따라 미리 표준을 준비하는 등, 관련 분야 전반을 선도하는 역할을 충실히 수행하고 있다. 이는 그림 2에 나온 것과 같이, 모바일 그래픽스는 물론, 모바일 미디어에 관련된 거의 모든 업체가 크로노스 그룹의 회원이고, 많은 업체가 적극적인 활동을 통하여 적절한 시기에 표준을 제정하는, 순순환 구조를 잘 이루고 있기 때문이다. 현시점에도 삼성, LG, SKT, 휴원, 엠텍비젼 등의 많은 국내 업체들도 여기에 참여하고 있으며, 경북대, 서울대, 한양대 등의 국내 대학들도 아카데믹 멤버로 활동하고 있다.

크로노스 그룹(Khronos Group)은 2000년 1월에 임베디드 시스템 분야에서 로열티가 필요 없는 공개 API 표준들을 제정하려는 목표로 설립된 산업체 컨소시엄이다. "산업체에 의한, 산업체를 위한 표준"을 지향하며, 회원사들의 상업적 성공을 뒷받침하기 위해, 로열티를 받지 않는, 완전 공개된 표준을 제공하는 것이 주요 임무이다. 제공하는 표준을 통해서 어떤 상업적 가치를 창출하기보다는, 이를 공개해서, 회원사를 비롯한 산업계 전체가 공동 이익을 창출한다는 이념으로 운영되고 있고, 현재까지 성공적인 비즈니스 모델로 평가 받고 있다.



▶▶ 그림 3. 크로노스 그룹에서 제공하는 표준들의 계층 구조

크로노스 그룹은 이제까지 그림 3에서와 같이, 레이어별로 다양한 표준들을 제공하고 있다. 각 표준에 대한 간단한 설명은 표 2에 제시되어 있다. 크로노스 그룹은 이러한 표준을 제정할 뿐만 아니라, 표준을 구현한 결과에 대한 검증도 담당하고 있다. 즉, 각 표준 별로 적합성 테스트 (conformance test suites, CTS)를 제공하고 있으며, 이를 통과한 구현 결과에 대해서만 크로노스 표준의 명칭이나 로고의 사용을 허용하고 있다. 각 적합성 테스

트는 다양한 종류의 예제 프로그램들로 구성되어 있으며, 새로 구현한 라이브러리는 해당 예제 프로그램들을 수행한 결과를 적합성 테스트에서 규정한 결과와 비교하여, 표준의 준수 여부를 검증하게 된다. 현재 크로노스 그룹에서 제공하는 표준들 중에서 모바일 그래픽스에 직접 관련된 것들로는 OpenGL ES, OpenGL SC, OpenVG 등을 들 수 있다. 다음 절부터는 각각에 대해서 설명하고, 결론 부분에서는 모바일 그래픽스 분야의 표준들에 대한 전망을 보이겠다.

표 2. 크로노스 표준들의 분류

표준명	다루는 내용
OpenGL	데스크 탑에서의 3D 그래픽스 가속
OpenCL	병렬 처리 가능한 CPU, GPU의 통합 운용
OpenGL ES	임베디드 시스템에서의 3D 그래픽스 가속
OpenGL SC	고신뢰도를 요구하는 시스템에서의 3D 그래픽스 가속
EGL	장치와 무관한 window system 기능을 제공
COLLADA	3D asset의 이동을 담당
OpenKODE	크로노스 표준들 간의 통신을 담당
OpenVG	2D 벡터 그래픽스 가속
OpenMAX	미디어 코덱의 가속
OpenSL ES	오디오 API의 가속
OpenWF	로우-레벨 윈도우 시스템 가속
OpenML	디지털 미디어 교환 및 동기화 지원

2. OpenGL ES

OpenGL ES [2] 는 휴대폰, 게임용 콘솔, 차량용 장치 등을 포함하는 다양한 임베디드 시스템에서 2차원 및 3차원 그래픽스 기능들을 완벽하게 제공하기 위한 목적으로 제정된 API 표준이다. OpenGL ES는 근본적으로 임베디드 시스템에서의 3차원 그래픽스 성능을 가속하기 위한 저수준(low-level) 인터페이스를 담당한다.

OpenGL ES 1.0 표준[3]은 OpenGL 1.3 표준을 기반으로, 2003년 7월에 발표되었다. 이 표준은 임베디드 시스템에 적용된 최초의 3차원 그래픽스 표준이라는 점에서 큰 의의를 가지지만, 보완이 필요하여, 2004년 8월에 OpenGL 1.5 표준을 기반으로 하는 OpenGL ES 1.1 표준[4]이 발표되었다. 또한, 기존의 그래픽스 하드웨어들에 프로그래머블 파이프라인(programmable pipeline)을 도입한OpenGL 2.0의 발표에 맞추어, 2005년 8월에는 OpenGL ES 2.0 [5]이 제정되었다.

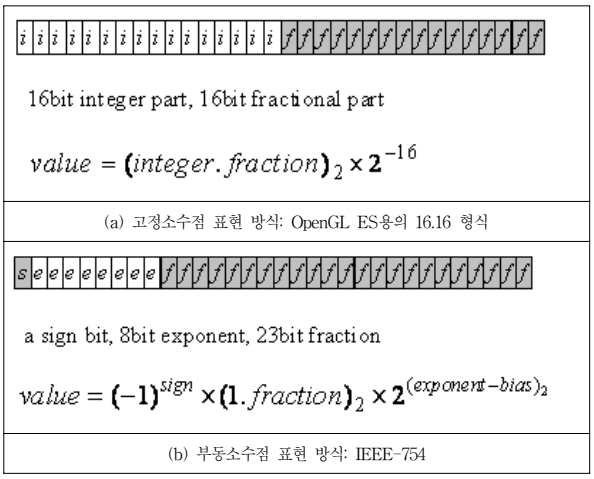
주의할 점은 OpenGL과는 달리, OpenGL ES 2.0에서는 OpenGL ES 1.1에서 제공하는 기능들을 모두 제공하지는 않는다는 것이다. 데스크탑 기반의 OpenGL에서는 하위 호환성을 유지하려는 노력이 절대적이지만, OpenGL ES의 입장에서는 임베디드 기기들에서 가능한 한 적은 자원을 차지하려는 노력이 더 강해서, 하위 호환을 보장하는 대신, 기기의 종류나 목적에 따라, 1.1 또는 2.0 표준을 선택적으로 사용하도록 하였다.

OpenGL ES 1.0의 제정 시에는 이미 산업계 표준으로 쓰이고 있는 OpenGL의 영향을 강하게 받아, OpenGL 1.3의 일부 함수들만 모은 부분 집합 형태에서 출발했지만, 실제 제정된 표준에는 임베디드 시스템의 하드웨어 및 소프트웨어 특성을 반영하기 위해 OpenGL과는 다른 특성들도 상당수 포함하고 있다. 예를 들어, 기존의 OpenGL이 고성능의 부동소수점 연산(floating-point operation)을 기반으로 개발되었음에 비해, OpenGL ES는 정수 연산만 가능한 하드웨어를 채택하는 중저가 임베디드 시스템을 위해 고정소수점 연산(fixed-point operation)만을 사용하여 3차원 그래픽스 기능을 제공할 수 있다. OpenGL ES 1.1에서는 이에 따라, 부동소수점과 고정소수점 연산 모두를 사용할 수 있는 Common Profile과 고정소수점 연산만을 사용하는 Common-Lite Profile의 2가지 세부 표준을 제공한다.

OpenGL ES 1.1에서 사용하는 고정소수점 표현은 32비트 정수 자료형을 변형한 방식으로, 그림 3에서 보는 바와 같이, 전체 32비트 중에서, 상위 16비트를 정수 부분으로 할당하고, 하위 16비트를 실수 부분으로 사용하는 방식이다[6]. 이 방식은IEEE 754[7]와 같은 부동소수점 표현에 비해서 표현할 수 있는 숫자의 범위가 좁다는 단점을 가지지만, 소수점의 위치가 고정됨으로써, 정수 연산만 가능한 하드웨어에서도 빠른 처리가 가능하다는 장점을 가진다.

OpenGL 2.0의 발표 이후에 제정된 OpenGL ES 2.0의 경우는shader language의 채택이 가장 중요한 변화가 될 것이다. vertex shader와 fragment shader가 완전히 지원되고, 대신 1.1에서 제공하던 고정된 파이프라인 함수들을 대폭 삭제하였다. OpenGL ES 2.0 표준의 제정은 시간적으로 OpenGL 2.1의 발표와 거의 비슷한 시기가 되었고, 이 무렵에는 OpenGL 표준에도 크로노스 그룹이 깊이 관여하기 시작하여, OpenGL에서 최신 기능으로 채택된 framebuffer object extension[8]이 OpenGL ES에서는 extension이 아니라, core 기능으로 들어가는 등의 영향을 받게 된다. shader language 측

면에서도 GL shader language의 시행 착오를 반영하여, ES shader language는 프로그래밍 언어 자체의 완성도를 한층 높였고, shader language를 지원하는 하드웨어에 대한 세부 요구 조건도 더욱 강화되었다. 이러한 변화는 나중에 OpenGL 3.0 표준의 제정에 긍정적인 영향을 미쳐, 서로간의 선순환 구조를 만들게 된다.



▶▶ 그림 4. 부동소수점 및 고정소수점 표현 방식

현재 OpenGL ES의 사용은 크게 OpenGL ES 1.1과 OpenGL 2.0이 공존하고 있지만, 임베디드 시스템에서는 향후 수년간은 OpenGL ES 1.1의 사용이 월등히 많을 것으로 예상하고 있다. 이는 2.0 표준에서 프로그래머블 파이프라인을 채택하면서, 고정소수점을 위한 Common-Lite Profile을 포기해야 했고, 이에 따라, 2.0 표준은 부동소수점 계산을 위한 별도의 하드웨어를 가진, 비교적 고가의 임베디드 시스템에서만 사용 가능하다는 제약을 가지기 때문이다. 2010년 상반기의 현 시점에서는 아직까지 OpenGL 2.0 하드웨어를 본격적으로 채택한 휴대폰 모델이 매우 드물다는 점도 고려해야 한다. 현재 시장에서 인지도가 높은 아이폰과 안드로이드 모두 아직까지 1.1 표준을 기준으로 하고 있다.

그림 5와 6은 각각 OpenGL ES 1.1과 2.0의 벤치마크 테스트[9]에 사용되는 프로그램들의 화면인데, 둘의 시각적 완성도 차이를 느낄 수 있을 것이다. 즉, 그림 5에서는 전형적인 3차원 그래픽스의 출력이 주였다면, 그림 6에서는 매트릭스 팔레트 기능과 shader language를 이용한 부드러운 텍스쳐 매핑의 지원으로, 더 부드러우면서도 사실적인 영상이 가능해 졌다. OpenGL ES 표준은 현재 다음 버전을 준비 중인데, 코드명 "Halti"인 이 새로운 표준이 OpenGL ES 2.1이 될지, 3.0이 될지는 아직 미정이지만, geometry shader를 비롯한 최신 기능의 추가 등이 논의되고 있다.

▶▶ 그림 5. OpenGL ES 1.1 하드웨어의 벤치마크 프로그램

▶▶ 그림 6. OpenGL ES 2.0 하드웨어의 벤치마크 프로그램

3. OpenGLSC

OpenGL SC(safety critical)는 3차원 그래픽스를 사용해야 하는 환경들 중에서, 안정성에 대한 고려가 특별히 필요한 경우를 위해 제정된 표준[10]이다. 이러한 요구가 가장 크게 대두된 곳은 항공기용 기기들이다. 아날로그 계기들이 지배하던 시대에서, 디지털 기기들이 출현하면서, 최첨단 기기들이 장착되는 항공기 시장에서도 디지털 기기들로의 교체가 활발히 진행되었고, 이의 정점에는 디지털 계기판의 등장이 있다. 특히 매우 좁은 공간에서 많은 정보를 처리해야 하는 전투기 계기판에서는 2-3개의 통합된 대형 LCD 화면을 사용하되, 기존 아날로그 계기판은 물론, 더 많은 정보를 표시하는 방식이 선호되고 있다. LCD 화면과 같은 디지털 디스플레이 장치

의 도입 후에는 3차원 정보를 표시하려는 요구가 자연스럽게 등장하였고, 이에 따라, 특히 Seaweed, Barco, ALT, Diehl, Esmertec 등의 군사용 디스플레이 장치 제작사들을 중심으로OpenGL에 초한 특별한 라이브러리 표준이 제정되었다.

▶▶ 그림 7. 아날로그 계기판과 대체되는 디지털 계기판

이렇게 해서, OpenGL SC (safety critical profile)가 군사 목적으로도 사용되어야 한다는 요구 조건에 맞추어 제정되었고, 현재는 군사용은 물론이고, 비슷하게 안정성이 요구되는 자동차용 전장 시스템, 의료 기기용 시스템 등에도 적용되고 있다. 이러한 안정성에 대한 요구 사항 때문에, OpenGL SC는 표준을 준수하였다는 적합성 테스트(CTS)의 통과는 물론이고, 설계 및 개발 단계에서 표준화된, 고신뢰성을 가지는 코드로 구현되었다는 것을 별도의 기준으로 인증받는 것이 일반적이다. 현재는 소프트웨어 공학 분야에서 제정된 DO-178B 스펙[11]에 따른 구현을 요구하고 있다.

OpenGL SC는 원래 OpenGL ES의 profile 중의 하나로 계획되었지만, 고성능을 추구하려는 휴대폰 시장의 요구와는 상반되는 요구 사항들이 많아서, 별도의 표준으로 spin-out 하여, 2005년에 1.0 표준이 제정된다. 2009년에 제정되어, 현재 사용되고 있는 OpenGL SC 1.0.1 표준[10]을 살펴 보면, 기능 선택에 있어 매우 보수적인 측면을 볼 수 있다. 즉, 최신의 우수한 기능이 도입되더라도, 충분한 신뢰성이 확보되지 않으면, 여전히 이전 기술을 고수하려고 하고, 개발 비용의 측면에 따라, 상대적으로 이전 기술들에 대한 지원에 충실하고, Shading language와 같은 최신 기술들의 도입은 유보된 상황이다. 특히, OpenGL ES에서는 효율성의 문제 때문에 표준에서 제외된glBegin / glEnd 패러다임이 OpenGL SC에서는 여전히 중요한 역할을 담당하고 있다. 이는 호환성을 보장하고, 가능한한 기존 코드를 그대로 유지하려는 측면에서의 배려로 볼 수 있다.

OpenGL SC는 최종적으로 101개의 함수로 구성되었고, 모든 함수는 OpenGL 1.3의 서브셋이 된다. OpenGL ES와의 비교에서는 새로 32개의 함수가 도입되어, 서로 간의 호환성은 보장되지 않는다. 또, OpenGL ES의 common-lite profile에서와 같은, fixed-point number representation에 대해서는 지원하지 않는다. 반면에, single precision 에 대한 지원은 OpenGL ES에 준하는 수준에서 제공되고 있다.

OpenGL ES와 비교해서 상당히 특이한 점으로, paletted texture에 대한 지원을 들 수 있다. 이 기능은 color index를 쓰는 시스템과 거의 동일한데, PC 시장에서는 메모리 가격의 하락으로 이미 90년대 후반부터 도태되기 시작한 기법이고, OpenGL ES에서는 전혀 고려되지 않았다. 반면에, OpenGL SC에서는 이에 대한 지원이 필수 요구 사항이고, 이 때문에, 기존의 OpenGL 또는 OpenGL ES 하드웨어와의 완전한 호환은 매우 어려운 일이다.

OpenGL SC는 그 특성상, 일반 사용자들이 쉽게 접할 수 있는 표준은 아니지만, 사용처가 점차 확대되어 가려는 추세이다. 즉, 절대적인 안정성과 신뢰성에 기반한 소프트웨어라는 점은 의료용 기기나 자동차용 전자 장비 등의 민간 시장에서도 충분한 매력을 가지고 있다. 현재 이들 분야를 중심으로, OpenGL SC를 그대로 사용하거나, 새로운 표준으로 업그레이드하려는 노력이 나타나고 있다.

4. OpenVG

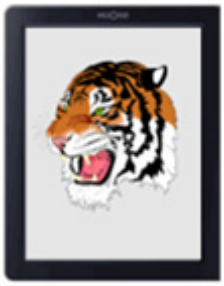
3차원 그래픽스 출력들이 사실적이고 매혹적인 화면들을 만들어 낼 수 있음은 분명하지만, 최종 출력의 비중으로 본다면, 아직까지는 2차원 그래픽스 기술들이 상당한

비중을 차지하고 있다. 최근의 2차원 그래픽스 시스템들은 기존의 래스터 그래픽스 기능이나 이미지 처리 기능들에 추가하여, 스케일러블 벡터 그래픽스(scalable vector graphics) 기능에 초점을 맞추고 있다. 벡터 그래픽스는 확대나 축소 시에 계단 현상(jaggy effect) 등의 이미지 손상이 발생하지 않으며, 원본의 품질을 그대로 유지하기 때문에 디스플레이 장치의 크기나 해상도에 무관하게 동일한 품질을 제공할 수 있다는 장점을 가진다. 이는 특히 그래픽스 출력을 다운사이징해서 사용하는 경우가 많은 소형 임베디드 시스템에 적합하다.

컴퓨터 그래픽스의 상업적 응용들에 있어서는 임베디드 시스템에 대한 고려가 강조되고 있다. 최근 모바일 시스템의 성능이 향상되면서 GUI, MMS (multimedia message service), 내비게이션, SVG 플레이어와 같은 다양하고 고수준의 그래픽스 환경을 필요로 하는 응용 프로그램들이 많이 탑재되고 있다. 이러한 응용 프로그램들을 텍스트나 비트맵 형태로 처리하기에는 한계가 있고, 다양한 그래픽스 기능을 지원하기 위해서 벡터 그래픽스가 대두되고 있다.

OpenVG[12]는 2차원 벡터 그래픽스 시스템의 가속을 위한, 플랫폼 독립적인 공개 표준이다. 데스크 탑에서의 일반적인 3차원 그래픽스 표준들과는 달리, 픽셀 단위가 아니라, 벡터 단위로 처리가 이루어진다는 것이 큰 특징이다. 이 때문에, OpenVG의 공식 규격에 명시된 대로, 이 표준은 PostScript, PDF, Flash, Java2D, SVG 등의 2차원 API와 유사한 역할을 담당한다. 즉, 벡터 폰트의 처리나, 해상도와 무관하게 플래쉬 영상을 처리하거나 등의 작업에서, 매우 적합한 기능들을 제공한다. 그림 8은 OpenVG를 이용한 전형적인 출력의 예제들이다. OpenVG는 사용자 UI의 비중이 커지면서, 중요성이 강조되고 있고, 3차원 표준인 OpenGL ES와의 협력 관계에 대해서도 적절한 방법들이 모색되고 있다.

(a) anti-aliased map and navigation



(b) tiger: path stroke and fill

(c) outline fonts

(d) stroke parameters

(e) image effects

(f) sample animation

(g) drawings

▶▶ 그림 8. OpenVG 프로그램의 실행 화면들

5. 결론

크로노스 그룹은 원래 OpenGL ES만을 담당하였지만, 점차 더 많은 기업들이 참여하고, OpenGL 2.0 이후에는 OpenGL과 OpenGL ES 사이의 협력 관계가 중시되면서, 최근에는 OpenGL 표준의 관리도 기존의 OpenGL ARB에서 크로노스 그룹으로 이관되었다. 이외에도, 크로노스 그룹에서 제공하는 표준들을 종합적으로 고려해 보면, 그래픽스 시장에서는 PC 시장의 OpenGL과 모바일 시장의 OpenGL ES가 서로 보완 관계를 보이면서 발전해 가고 있으며, 이들 사이의 협력 관계는 당분간 계속될 전망이다.

OpenVG 는 임베디드 시스템에 벡터 그래픽스를 도입했다는 점에서 매우 긍정적이며, 결과적으로 옳은 선택을 하였고, 현재로서는 임베디드 시스템에서의 사용자 인터페이스에서 중요한 역할을 할 수 밖에 없을 것이다.

또한, 아이폰과 Adobe flash의 갈등 관계로 인해, 모바일 시장에서는 플래쉬를 대신하는 역할을 일정 부분 수행할 수밖에 없을 것으로 보인다. 종합적으로 보아, 임베디드 시스템에서의 모바일 그래픽스 환경에서는 당분간 크로노스 그룹의 표준들이 절대적인 위치를 유지할 것으로 전망된다.

감사의 글

본 연구는 지식경제부 사업화연계기술개발사업의 지원으로 수행되었음

참고문헌

[1] The Khronos Group, http://www.khronos.org/
[2] OpenGL ES, http://www.khronos.org/opengles/
[3] David Blythe, OpenGL ES - Common/Common-Lite Profile Specification, version 1.0.02, Khronos Group, 2002.
[4] Aaftab Munshi and Job Leech, OpenGL ES - Common/Common-Lite Profile Specificatoin, version 1.1.12(Full Specification), Khronos Group, 2008.
[5] Aaftab Munshi and Job Leech, OpenGL ES - Common Profile Specification, version 2.0.24 (Full Specification), Khronos Group, 2009.
[6] ARM, *Fixed Point Arithmetic on the ARM, Application Note 33*, ARM, 1996.
[7] D. Hough, "Applications of the proposed IEEE-754 standard for floating point arithmetic", *IEEE Computer,* vol.14, no.3, pp.70-74, 1981.
[8] Aaftab Munshi, GL_OES_framebuffer_object, Khronos Group, 2008.
[9] Rightware, 3DMark Mobile, http://www.rightware.com/en/Benchmarking+Software
[10] Bruce Stockwell, OpenGL SC - Safety-Critical Profile Specification, version 1.0.1 (Difference Specification), 2009.
[11] Radio Technical Commission for Aeronautics, DO-178B - Software Considerations in Airborne Systems and Equipment Certification, 1999.
[12] Daniel Rice and Robert J. Simpson, OpenVG Specification, version 1.1, 2008.

저자소개

● 백 낙 훈(Nakhoon Baek) 정회원

- 1992년 2월 : 1990년 한국과학기술원 전산학과 학사
- 1992년 한국과학기술원 전산학과 석사
- 1997년 한국과학기술원 전산학과 박사
- 1997~1998년, 2008~2009년 조오지 워싱턴대 연구원
- 2004년 ~ 현재 경북대학교 IT대학 컴퓨터학부 교수

● 이 환 용(Hwanyong Lee)

- 1990 한국과학기술원 전산학과 학사
- 1992 포항공과대학 컴퓨터공학과 석사
- 1995 포항공과대학 컴퓨터공학과 박사 과정 수료
- 2009~ 현재 경북대학교 전기전자컴퓨터 박사과정
- 1995~1999 포항공대정보통신연구소 연구원
- 1999~2004 액트시스템㈜ 대표이사
- 2004~현재 ㈜휴원 연구소장 및 기술마케팅이사

<관심분야> 컴퓨터그래픽스, 가상현실, 임베디드 시스템 소프트웨어