OpenVG™ Conformance Test Suite의 개발

성현찬*0, 이준영*, 이환용*, , 백낙훈**, 박기현***
*㈜휴원 솔루션사업부, **경북대학교 전자전기컴퓨터공학부, ***계명대학교 정보통신대학
{jylee, hylee, hcsung}@hu1.com, nbaek@knu.ac.kr, khp@kmu.ac.kr

Development of OpenVG™ Conformance Test Suite

Hyunchan Sung*0, Junyoung Lee*, Hwanyong Lee*, Nakhoon Baek** and Kihyun Park***
*Solution Division, HUONE Inc.
**School of Electrical Engineering & Computer Science, Kyungpook National Univ.
***College of Information and Communication Engineering, Keimyung Univ.

요 약

OpenVG 는 Khronos Group 에 의해 제정된 2D 벡터 그래픽스를 위한 하드웨어 가속 표준이다. OpenVG 적합성 테스트 도구(conformance test suite)는 다양한 기관에서 구현된 OpenVG 엔진이 OpenVG 표준에서 정의한 요구사항들을 만족하는가를 측정하는 도구이다. 적합성 테스트의 통과 여부는 시장에서 해당 제품의 기술적 우위성 및 안정성을 보장하는 지표가 되기에 적합성 테스트 개발 과정에서 형평성 및 객관성을 확보하는 것은 대단히 중요한 일이다.

본 논문에서는 OpenVG Conformance Test Suite 의 설계 및 구현 과정을 소개함으로써 표준의 제정과 더불어 꼭 필요한 과정인 적합성 테스트 도구 개발 시에 형평성과 객관성을 유지하기 위한 바람직한 개발 모델을 제시한다. 이를 통해 향후 타 표준의 적합성 테스트 개발 시에 참고가 되고자 한다.

[본 연구는 산업자원부 지방기술혁신사업(RT1040-03-02) 지원으로 수행되었음]

1. 서론

데스크탑 및 무선단말기를 위한 컨텐츠를 구성하는데 있어 가장 중요한 정보 표현 수단은 이미지이다. 기존에는 비트맵 방식의 이미지 표현이 주를 이루었으나, 최근에는 Flash 와 SVG[7] 파일 포맷을 주축으로 한 벡터 방식을 많이 사용하고 있다. 벡터 방식을 사용하면 수학적 값의 변경만으로 확대/축소가 가능해 단말기의 화면 사이즈에 독립적이며, 애니메이션이 자유롭고, 데이터 사이즈가 상대적으로 작아 네트워크 전송에 유리한 장점을 가진다[5]. 이러한 장점으로 인해, 상대적으로 자원이 한정적인 임베디드 시스템에 벡터 방식의 그래픽 엔진을 도입하려는 노력이 이루어 지고 있으며 몇몇 솔루션들은 시장에서 좋은 반응을 얻고 있다.

임베디드 시스템을 위한 3D 그래픽스 표준 엔진인 OpenGL/ES 를 제정한 Khronos Group[1]에서는 2005 년 8 월에 임베디드 시스템에서 벡터 그래픽스를 하드웨어적으로 가속하기 위한 저-레벨(Low Level) 표준으로 OpenVG[2]를 발표하였다.

OpenVG 의 표준 발표와 더불어 다양한 제품이 개발 또는 개발 예정이다. 대표적인 제품으로, 휴원사의 AlexVG 는 PC, ARM 기반의 임베디드 시스템에서 동작하는 소프트웨어 OpenVG 엔진이다[6]. ATI 와 nVidia 등 그래픽 카드 업체에서도 2007 년 상반기에 OpenVG 가 탑재된 제품을 출시할 계획이다.

이렇게 다양한 회사들에 의해 OpenVG 제품이 속속 구현됨에 따라 Khronos Group 의 OpenVG 워킹그룹에서는 OpenVG 적합성 테스트 도구(Conformance Test Suite)를 개발하고 있다. 적합성 테스트 도구는 해당 제품이 OpenVG 표준에서 정의한 요구사항들을 만족하는가를 측정하는 도구이다. 적합성 테스트를 통과한 제품은 Khronos Group 의 공인된 로고를 제품에 부착할 수 있다. 이는 곧 시장에서 해당 제품의 기술적 우위성 및 안정성을 보장하는 지표가 되기에 많은 회사들은 적합성 테스트 도구의 개발 과정부터 참여하고 있다.

하지만, 이 과정은 실제 회사의 이익과 직결된 부분이다. 이로 인해 논의 과정에서 경쟁사간 충돌이 발행 할 수 있으며, 적합성 테스트 통과 기준 값을 너무 낮게 설정하면 변별력이 없는 테스트로 전락할 수 있으며, 너무 높게 설정하면 특정 회사 만을 위한 테스트 도구로 전락할 수도 있다. 이렇듯 적합성 테스트 도구 개발 과정에서 형평성과 객관성을 유지하는 일은 대단히 중요하다.

본 논문에서는 OpenVG 적합성 테스트 도구 개발 과정을 소개함으로써 적합성 테스트 도구를 개발할 때 고려하여야 할 점이 무엇인지 살펴 보고, 형평성 및 객관성을 유지하기 위한 바람직한 적합성 테스트 도구 개발 모델을 제시하고자 한다. 2 장에서는 Khronos Group 의 OpenVG 및 Conformance Test 절차에 대해 간략히 살펴보고 3 장에서는 OpenVG Conformance Test 의 요구사항에 대해 살펴본다. 4 장에서 적합성 테스트 도구 개발 모델을 제시하고 5 장에서 이러한 모델이 가지는 특징에 대해 논의하고자 한다. 마지막으로 6 장에서 결론을 맺는다.

2. 관련 사항

2.1 Khronos Group

Khronos Group은 2000년 1월에 설립된, 임베디드 시스템을 위한 표준을 제정하는 비영리 단체로 다른 표준 기구와는 달리 대부분의 멤버가 기업체로 구성되어 있다.

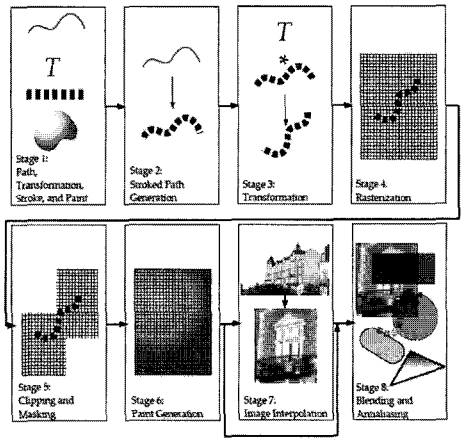
Khronos Group에서 현재 발표 또는 제정 중인 표준으로는 하드웨어 가속 표준으로 OpenVG, OpenGL/ES, OpenMAX, OpenSL/ES가 있으며 저작 포맷 표준으로는 Collada와 OpenML이 있다. 최근에는 임베디드 시스템을 위한 통합 플랫폼 성격의 OpenKode 또한 논의하고 있다.

Khronos Group의 회원 레벨은 3단계로 구성된다. Khronos Group의 운영 및 각종 안건에 대해 의결권을 가지는 Promoter 레벨, 각종 표준의 제정에 직접 참여할 수 있는 Contributor 레벨, 제정된 표준으로 제품을 개발할 수 있는 Adopter 레벨이 있다.

2.2 OpenVG

OpenVG 는 2D Vector Graphics 의 하드웨어 가속을 지원하기 위한 플랫폼 독립적인 표준이다. 하드웨어 가속을 통해 고성능, 고품질의 벡터 그래픽 이미지를 얻을 수 있고 전력 소모도 줄일 수 있다. 또한 소프트웨어로도 구현이 가능해 플랫폼 독립적인 응용프로그램을 개발할 수 있다.

OpenVG 는 [그림 1]에서 보듯 8 개의 State 를 가지는 파이프라인으로 구성되며 각 State 의 역할은 아래와 같다.



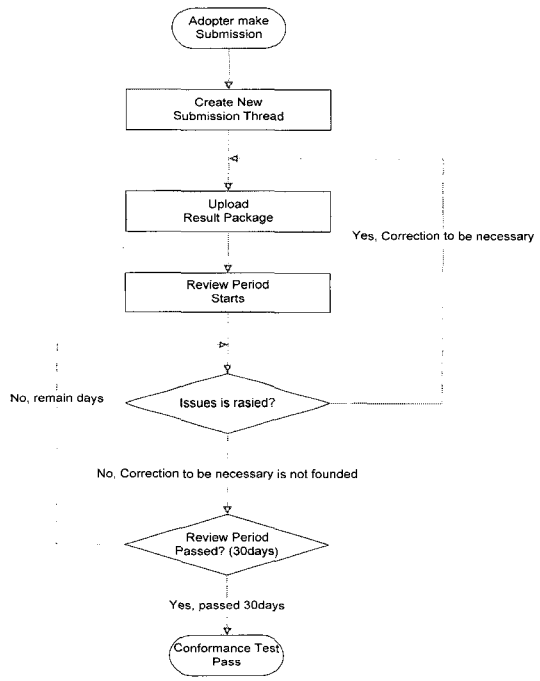
[그림 1] OpenVG 의 파이프라인[2]

State1 은 응용프로그램 개발자가 OpenVG 워킹 그룹에서 제공한 API 를 이용하여 각종 Path 의 형태, 색상, Transform, Line Width, Dash 등 다양한 Drawing Parameter 을 정의하는 단계이다. State2 는 개발자가 정의한 Path 정보와 Stroke Parameter 를 결합하여 실제 출력이 가능한 Path 로 재정의하는 과정이다. State3 은 해당 Path 를 개발자가 정의한 Transform 정보와 결합하여 새로운 좌표를 계산해 내는 단계이며 Stage4 에서 이러한 Vector Data 를 실제 화면에 1:1 매핑하게 된다. State5 에서는 실제 출력해야 하는 Pixel 들을 결정하는 단계이며, State6 에서 해당 Pixel 의 색상을 결정한다. Image 를 출력하는 경우에는 State7 에서 Image 에 의한 Pixel 값을 추가로 결정한다. State8 에서는 앞 Stage 를 통해 완성된 이미지를 이전의 결과물과 합성하여 최종 결과를 얻는다.

2.3 OpenVG Conformance Test

OpenVG 표준을 이용하여 제품을 개발하기 위해서는 Adopter 멤버로 가입하여야 하며 적합성 테스트를 받기 위해서는 추가적인 비용이 필요하다.

OpenVG의 적합성 테스트 승인 절차는 [그림 2]의 과정으로 이루어진다. Adopter가 제품을 개발하여 적합성 테스트를 수행하고 그 결과물을 OpenVG 워킹 그룹(WG)에 업로드 한다. 이후, 다른 Adopter, OpenVG WG 검토 이사회, Khronos Group 검토 이사회에서 해당 결과물을 검토하여 문제점을 제기한다. 문제가 제기된 제품은 수정 후 다시 업로드하거나, 해당 사항이 문제가 아님을 증명하여야 한다. 30일 동안 어떠한 이의 제기도 없으면 해당 제품은 적합성 테스트를 통과하게 된다.



[그림 2] OpenVG 적합성 테스트 절차

모든 OpenVG를 구현한 제품은 적합성 테스트를 통과하여야 하며 적합성 테스트와 관련한 절차 및 비용 등을 기재한 문서는 Adopter 계약시 별도의 문서로 제공된다.[2][3]

3. OpenVG 적합성 테스트의 요구사항

기본적으로 OpenVG을 최종적으로 사용하는 개발자는 가볍고, 빠른 성능을 요구하며 시스템이 안정적이기를 희망한다. 표준의 엄격한 강제 조항들은 이러한 안정성을 보장하기 위한 수단이다. 반면에, Adopter들은 불필요한 부분을 제거하고 최적화하여 속도를 빠르게 하고자 한다. 이들에게는 표준의 엄격한 강제 조항은 속도를 저하시키고 제품 단가를 높이는 이유가 된다.

OpenVG 적합성 테스트는 이를 어느 범위까지 허용할 것인가의 기준을 제공한다. 적합성 테스트를 만족하는 범위안에서의 구현 최적화는 바람직한 개발자의 자세일 것이다.OpenVG 적합성 테스트 도구가 요구 받는 몇 가지 중요한사항은 아래와 같다.

3.1 Window System Independence

OpenVG는 플랫폼 독립적인 API이다. 이는 OpenVG가 어떠한 디스플레이 API도 포함하고 있지 않기 때문이다. Conformance Test Suite 또한 display에 종속적인 기능들은 독립된 모듈로 제공하여야 한다.

이는 적합성 테스트 개발 시 디스플레이 모듈이 EGL[5] 기반으로 작성되어야 함을 의미한다. OpenVG는 EGL를 통해 메모리 프레임버퍼 기반의 Surface를 확보하고 여기에 결과물을 출력한다. 이러한 프레임버퍼의 내용은 플랫폼 또는 운영체제의 디스플레이 모듈을 이용하여 비로소 화면에 출력된다. Adopter들은 OpenVG 구현 시 자신의 플랫폼에 맞는 EGL 루틴을 제공함으로써 플랫폼 독립적인 Conformance Test Suite를 포팅할 수 있다.

3.2 Anti-aliasing Algorithm Independence

현재 2D Vector Graphics 시장에는 수 많은 Anti-aliasing 알고리즘이 개발되어 사용되고 있고, 몇몇 알고리즘은 지적 재산권으로부터 보호를 받고 있다. OpenVG 적합성 테스트에서는 특정 Anti-aliasing 알고리즘을 사용할 것을 강요하지 않는다. 이는 공개된 표준을 지향한다는 Khronos Group의 정책과도 일맥 상통하는 부분이다. 알고리즘과 무관하게 일정 수준의 Anti-Aliasing 품질만 보장하면 적합성 테스트를 통과할 수 있을 것이다.

3.3 On-Device and Off-Device Testing

OpenVG 적합성 테스트 도구는 OpenVG가 구현된 플랫폼의 메모리 및 리소스 자원을 최소한으로 사용하는 것을

목표로 한다. 이는 OpenVG가 자원이 한정적이고 프로세스 속도가 상대적으로 낮은 임베디드 시스템을 목표로 한다는 것을 비추어 볼 때 바람직한 일이다.
결과물의 생성은 플랫폼(On-Device)에서 직접 수행하고, 이 결과 데이터는 테스크탑 환경(Off-Device)에서 미리 저장되어 있는 참조 이미지와 비교를 수행하고 그 결과를 통보한다.

4. OpenVG Conformance Test의 설계 및 구현

4.1 설계 원칙

OpenVG 적합성 테스트 도구를 개발함에 있어 Adopter들 간에 논쟁이 되는 이슈와 이에 대한 대안을 차례로 살펴보면 다음과 같다.

4.1.1 무엇을 테스트 할 것인가?

테스트 항목을 무엇으로 할 것인지는 Adopter에게 민감한 사안이다. 예를 들어 Vector Rasterization 기술에 우위가 있는 회사는 해당 부분을 집중 강조하고자 할 것이며, 이는 결국 다른 회사에게는 불이익으로 다가온다.
하지만, 결국 최종 개발자가 참조하는 것은 표준 문서이기에 표준 문서에 의거하여 테스트 항목을 결정하고 코드를 부여하였다. 이들 코드의 범위 안에서 OpenVG 워킹 그룹 멤버들이 테스트 항목을 제안하고 워킹 그룹의 동의를 받는다.

4.1.2 어떤 결과물이 바른 결과물인가?

OpenVG는 그래픽 요소를 정의하고 있는 표준이다. 일례로 (x0, y0), (x1, y1), (x2, y2), (x3, y3)를 제어점으로 하는 베지어(Bezier) 곡선을 그렸을 때 이의 결과를 픽셀 단위로 살펴보면, 구현 방식에 따라 상당히 다양한 결과가 가능하다. 각 OpenVG Adopter들은 자기들의 결과 이미지가 정답이라고 주장할 것이며 자신의 이미지가 참조 이미지로 채택되길 주장할 것이다.
이러한 논쟁은 뚜렷한 해답이 없는 경우가 많으며 적합성 테스트 도구의 개발을 지연시키는 요인이 된다. OpenVG 적합성 테스트에서는 기존의 다른 적합성 테스트 도구들이 특정 회사의 결과물을 참조 이미지로 채택하는 방식이 아닌 수학적으로 시뮬레이션한 결과물을 참조 이미지로 채택한다. 이러한 수식은 대부분은 OpenVG 표준에 서술되거나 참고문헌으로 소개되어 있다.

4.1.3 어떻게 테스트 할 것인가?

각 업체에서 구현된 OpenVG의 결과 이미지가 수학적인 참조 이미지와 완전히 동일할 수는 없으며, 이를 평가하는 기준이 필요하다.
OpenVG 적합성 테스트에서는 다양한 비교 알고리즘이 구현되어 있다. 크게 텍스트 값의 비교와 이미지 픽셀 값의 비교로 구분되며 이미지 픽셀의 비교는 다시 최대/최소값 비교, 입실론(Epsilon)값 비교 등 다양하다.
아울러, 해당 알고리즘으로 결과물을 비교했을 때 참조 이미지와 어느 정도의 오차까지 허용할 것인가 역시 중요한 이슈이다. OpenVG 적합성 테스트에서는 각 테스트마다 최소값과 최대값을 정하고 있다.

4.2 Adopter의 참여 극대화 방안

OpenVG Conformance Test 구현에 있어서 Adopter들의 참여를 극대화 시키는 것은 대단히 중요하다. 결국 OpenVG를 구현하고 Conformance Test를 수행할 당사자들이기 때문이다.
OpenVG Conformance Test의 Test Case는 개발자 관점에서 정의된 목차임을 앞에서 이야기 하였다. 하지만, 내부적인 동작인 파이프라인의 동작을 테스트 하는 과정 역시 중요하다. 이를 보완하기 위해 Adopter들이 자유롭게 테스트 항목을 제안할 수 있도록 하였다. 테스트 항목을 제안하기 위해서는 아래의 4가지를 제공하여야 한다.

- 테스트 내역 설명서 : 분류, 목표
- On-Device 에서 결과 값을 생성하는 함수의 소스
- 수학적으로 계산된 값 또는 이미지
- Off-Device 에서 값을 비교하기 위한 함수의 소스

소스를 직접 제출함으로써 해당 테스트의 성격을 좀더 명확히 할 수 있고, 적합성 테스트 도구에 따른 업무 부하를 분산시킬 수 있다.
이렇게 제안된 테스트 항목은 논의 및 다른 Adopter들의 검토하고 해당 Test를 통과하기 위한 조건이 결정된다.

4.3 시스템의 구성

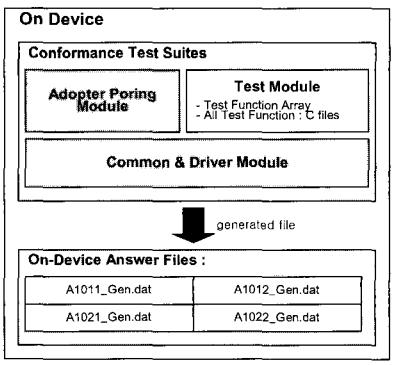
위의 과정에 의해 정의된 테스트 항목들은 각각 Adoter에게 소스 형태로 제공되는 On-Device 모듈과 최종 실행파일 형태로 제공되어 데스크탑에서 수행되는 Off-Device 모듈로 구성된다.

4.3.1 On-Device Module

On-Device 모듈은 아래[그림 3]에서 보듯 크게 3개의 모듈로 구성된다.

Adopter Porting 모듈은 3장에서 서술한 EGL 모듈과 결과물을 저장하는 내용을 포함하고 있다. Adopter는 이 부분을 자신의 플랫폼에 맞게 수정하여야 한다. Test Module은 각 테스트 항목별로 결과값을 생성하는 함수의 묶음으로 구성되어 있다. Common 및 Driver 모듈은 전체 Test Case를 관리한다.

최종 산출물은 각 테스트 항목을 수행한 값 또는 이미지로 각 테스트 항목별로 독립된 파일로 저장되어 Off-Device 테스트의 입력 데이터로 사용된다.

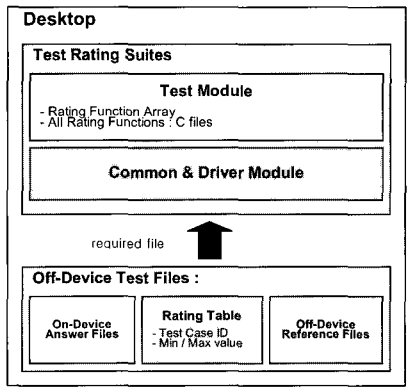


[그림 3] On-Device 모듈

4.3.2 Off-Device Test Module

Off-Device 모듈은 [그림 4]에서 보듯 크게 2개의 모듈로 구성되어 있다.

Test Module은 각 테스트 항목별 Test Case별 참조 이미지와 On-Device 테스트의 결과 이미지를 서로 비교하기 위한 함수들의 묶음으로 구성되어 있다. 이들 함수를 수행하여 그 이미지간의 점수를 계산한다. Common & Drive 모듈은 전체 테스트 항목을 관리하면서 각 항목별로 계산된 점수 값이 미리 정의된 범위를 만족하는지 체크하여 해당 테스트 항목의 통과 여부를 결정한다.



[그림 4] Off-Device 모듈

5. Features of Implementation

이렇게 구현된 OpenVG Conformance Test는 아래와 같은 특징 및 장점을 가진다.

첫째, 특정 업체의 결과물을 표준으로 사용하던 형태에서 벗어나 수학적인 결과물을 표준으로 채택함으로써 적합성 테스트 도구 개발에서 가장 중요한 형평성 문제를 해결하였다. 또한 Adopter 입장에서는 로열티 등의 문제에서 자유롭게 되었다.

둘째, OpenVG WG에서 기본적으로 정의된 뼈대 위에 다양한 Adopter들이 직접 Test Case를 제안함으로써 Adopter들의 다양한 의견을 반영하고 Conformance Test 개발이 특정 업체에 의해 주도되는 것을 차단하고 각 업체들 사이의 선의의 경쟁을 이끌어 낼 수 있도록 하였다.

셋째, OpenVG Conformance Test는 On-Device Test와 Off-Device Test를 분리하여 Device에서 수행되는 부분을 최소화하였고, Display 모듈을 분리하고 이를 Adopter가 직접 수정이 가능하도록 함으로써 플랫폼 종속성 문제를 해결하였다.

6. 결론

오늘날은 모든 산업이 서로 연관되어 있다. 2D 벡터 그래픽이 지금처럼 발전하게 된 뒤에는 매크로미디어 사의 Flash의 역할을 높이 평가하지 않을 수 없다. PC 환경에서 시작한 단순한 하나의 File Format이였지만, 사용자들의 요구에 의해 이동통신 사업자들이 해당 컨텐츠를 서비스하기 시작했고, 이는 무선단말기 제조업체에 영향을 주었으며, 좀더 빠른 처리를 위해 하드웨어 업체들도 해당 시장에 뛰어 들었다. 이 와중에 SVG라는 2D 벡터 그래픽스 표준 파일 포맷도 만들어 졌고, 하드웨어 가속을 위한 2D Vector 표준인 OpenVG도 만들어졌다.

이러한 산업 관계의 연관성에 의해 더욱 표준의 중요성이 강조되고 있다. 즉, 오늘날은 표준의 시대이며, 표준을 지키지 않은 제품은 그 활용이 지극히 제한적일 수 밖에 없다. 표준의 제정 이후에는 Conformance Test Suite가 꼭 필요하며 이는 표준을 제대로 이행하여 구현하였는지 테스트 하는 도구이다.

본 논문에서는 대부분 표준 제정 단체에서 고민하는 적합성 테스트 도구 개발에 있어 형평성 및 객관성을 보장하기 위한 방법에 대해 살펴보았다.

객관성 유지를 위해 수학적 모델을 사용하고, Adopter의 참여를 통해 테스트 항목을 결정함과 더불어, Adopter들이 단순 의사개진이 아닌 구체적 소스를 제시함으로써 적합성 테스트 개발에 따른 업무를 분산하기 위한 노력 등은 향후 다른 표준에서도 널리 참조할 만하다.

요즘은 표준의 제정과 더불어 제품이 발표되는 경우가 많다. 이는 곧 Conformance Test Suite의 제공이 그만큼 빨라져야 함을 의미한다. 이를 위해서는 표준의 제정과 동시에 적합성 테스트의 개발되어야 함을 의미한다. WIPI 등의 국제 표준화를 위해 노력하고 있는 국내의 표준제정기관에서도 우수한 적합성 테스트 방법 개발을 위한 노력을 계속하여야 할 것이다.

Acknowledgements

본 논문의 많은 부분은 Khronos Group의 OpenVG 워킹그룹 내에서 논의와 투표를 통해 결정된 내용입니다. OpenVG 적합성 테스트개발에 도움을 주신 대구 전략산업기획단과 동 기관의 모바일 그래픽스 연구그룹 회원들께 감사드립니다.

References

[1] Khronos Group, " http://www.khronos.org/"

[2] Khronos Group, " OpenVG Specification 1.0" , OpenVG Workgroup, August 2005.

[3] Khronos Group, " OpenVG Conformance Test Process 1.0" , OpenVG Workgroup , June 2005.

[4] Khronos Group, " EGL Native Platform Graphics Interface 1.2" , Khronos Technical Workgroup, November 2005.

[5] 이준영 외, " 모바일 통신 환경을 위한 벡터 그래픽스 커널의 설계" , 모바일학회논문지 Vol 2, June 2005.

[6] 휴원, " http://www.hu1.com"

[7] W3C, " Scalable Vector Graphics (SVG) Tiny 1.2" W3C SVG Workgroup, April 2005.