모바일 통신 단말기를 위한 벡터 그래픽스 커널 개발

이환용* · 박기현** · 우종정***

Development of a Vector Graphics Kernel for Mobile Communication Terminals

Hwanyong Lee · KeeHyun Park · Jongjung Woo

이 논문은 산업자원부 지방기술혁신사업(RT104-03-02) 연구비를 지원받았음

요 약

모바일 통신 단말기의 급속한 발전과 다양한 사용자들의 요구로 인하여, 이미지 정보를 포함한 멀티미디어 정보가 모바일 통신에서 컨텐츠의 기반을 이루고 있다. 전송 지연시간과 경비를 고려할 경우에 비트맵 방식 보다 유리한 벡터 그래픽스 방식의 이미지 정보를 효율적으로 이용하기 위해서는 효율적인 벡터 그래픽스 지원 시스템이 필요하다. 따라서, 많은 벡터 그래픽스 커널 시스템들이 제안되고 있으며, 호환성을 높이기 위하여 벡터 그래픽스 커널에 대한 표준화 작업이 진행되고 있다.

본 논문에서는 자원 제한적인 모바일 단말기에 적합한 벡터 그래픽스 커널의 요구 사항을 살펴보고, 표준으로 제안된 Khronos Group의 OpenVG 기반 벡터 그래픽스 커널을 설계 구현한다. 또한, 구현된 그래픽스 커널을 검증하기 위하여 PC 에뮬레이터 환경과 ARM 탑재 개발보드 환경에서 각각 포팅한 후, 성능을 측정한다.

ABSTRACT

Due to rapid development of mobile communication terminals and various requests of their users, multimedia information including image information has been the basis of mobile communication contents. In order to use vectored image information efficiently, which is more favorable than bit-mapped image information when transmission delay time and costs are considered, efficient vector graphics supporting systems are needed. Therefore, vector graphics kernel systems have been proposed and standardization attempts have been made in order to increase interoperability.

In this paper, a vector graphics kernel based on OpenVG is designed and implemented. OpenVG was proposed as a standard vector graphics kernel by Khronos Group recently. The implemented vector graphics kernel, named by alexVG, is developed on a PC emulator as well as on a development board equipped with an ARM processor. In addition, performance tests are made in order to verify its functions.

키워드

vector graphics kernel, mobile communication, ARM, OpenVG

* (주)휴원 연구소장

** 계명대학교 정보통신학부 교수

*** 성신여자대학교 컴퓨터정보학부 교수

접수일자 : 2006. 3. 28

Ⅰ. 서 론

모바일 정보통신 단말기의 급속한 발전과 다양한 사용자들의 요구로 인하여, 이미지를 포함한 멀티미디어 정보가 모바일 통신에서 컨텐츠의 기반을 이루고 있다. 이미지 정보의 표현 방식으로는, 비트맵 방식과 벡터 그래픽스 방식[2,5,7,8]이 있는데, 비트맵 방식은 주로 픽셀 복잡도가 크게 요구되는 사진 및 동영상 등의 정보에 사용되고, 벡터그래픽스 방식은 전송 지연 시간과 비용을 고려하여 픽셀 복잡도가 작아도 되는 만화 및 주가 그래프 등의 정보에 사용되고 있다.

벡터그래픽스 방식의 이미지 정보를 효율적으로 이용하기 위해서는 이를 지원하는 시스템에 대한 설계와 구현이 보다 기술적이고 효율적으로 이루어 져야 하기 때문에, 많은 그래픽스 커널 시스템들이 제안 개발되고 있다. 그런데, 서로 다른 모바일 단말기에서도 같은 그래픽스 응용 프로그램이 실행될 수 있도록 호환성을 높이기 위하여 그래픽스 커널에 대한 표준화 작업이 진행되고 있다. WWW 컨소시엄의 벡터 그래픽스(SVG, Scalable Vector Graphics)[9], Khronos Group의 OpenVG[5,7], OpenGL[6] 등이 그 예이다.

그래픽스 커널은 그래픽스 응용 프로그램과 하위 계층의 그래픽스 시스템과의 인터페이스 역할을 한다. 즉, 응용 프로그램 개발자들은 그래픽스 커널에서 제공하는 원시 명령어(primitive operation)를 이용하고, 원시 명령어는 하위 계층의 그래픽스 시스템을 이용하여 개발자들의 요구를 수행한다.

본 논문에서는 이러한 벡터 그래픽스를 지원하기 위해 자원 제한적인 모바일 단말기에서 필요한 그래픽스 커널 시스템의 설계 및 구현을 목표로 한다. 이를 위해, Ⅱ절에서는 벡터 그래픽스 관련 기술에 대해 살펴보고, Ⅲ절에서는 모바일 환경에 적합한 벡터 그래픽스 커널 시스템의 요구사항을 제시하고 이에 맞는 시스템을 설계한다. Ⅳ절에서는 설계된 시스템을 구현 및 검증하며, Ⅴ절에서 결론을 맺는다.

Ⅱ. 관련 기술

2.1. SVG- Tiny

1999년 WWW 컨소시엄은 신축적인 벡터 그래픽스(SVG, Scalable Vector Graphics)라고 부르는 개방된 포맷을 어도비 회사가 주축이 되어 개발하기 시작하였다. WWW 컨소시엄이 2001년 9월에 발표한 SVG 1.0 스펙은 매크로미디어 회사의 플래시와 동일한 그래픽과 애니메이션 기능을 제공할 뿐만 아니라 아래와 같은 장점을 추가로 가지고 있다[9].

- SVG는 WWW 컨소시엄에 의해 표준으로 승인된 개방된 포맷이다. 모바일 표준 기구인 3GPP는 미국과 유럽에서 사용할 차세대 이동전화를 위한 플랫폼을 정의하는 자리에서 SVG -Tiny를 멀티미디어 메시징 서비스(MMS, Multimedia Messaging Service)에 필수적인 포맷으로 채택하였다.
- SVG가 채택한 구문 형식은 XML 문서가 가지는 장점을 모두 가지고 있으며, XML 문서와 통합될 수 있다. 또한 XML DOM 구조에 익숙한 개발자라면 SVG의 접근이 용이하며, 웹 서비스와 호환된다.
- SVG는 SMIL 애니메이션 구문을 사용한다. 이 구문은 독립적 SVG 파일에 유용할 뿐만 아니라, SVG 문서가 멀티미디어 SMIL 표현에 내장될 수 있기 때문에 오디오 및 비디오 컴포넌트가 벡터 그래픽 애니메이션과 겹치도록 해준다. 또한 자바 기반의 스크립트를 지원한다.

현재 SVG는 전체 기능을 지원하는 SVG와 모바일용에 최적화된 SVG-Tiny로 구분되어 스펙 작업이 이루어지고 있으며, 현재 1.2 드래프트(Draft)가 배포되어 있다.

2.2. Flash-Lite

인터넷 환경에서의 벡터 기반 동영상의 사실상의 표준인 플래시는 매크로미디어 회사의 제품이다. 2003년 2월 매크로미디어사는 Flash 4 스크립팅 엔진 기반의 새로운 Flash 프로필인 Flash-Lite를 발표하였다. 이 프로필은 데스크 탑에서 사용할 수 있는 Flash Player 7의 전체 기능을 지원하기에는 처리 능력과 메모리가 부족한 대중 시장용 휴대폰을 겨냥한 제품이다[8].

Flash-Lite는 일본의 NTT 도코모(DoCoMo) 회사가 i-mode를 통해 무선 단말기에 서비스하였으며, 국내에서는 디지털아리아가 2002년 최초로 Flash 플레이어를 모바일 단말기에 구현하였다. Flash-Lite 1.1에서는 플래쉬 만의 장점인 액션스크립트(ActionScript) 4를 지원함과 동시에 아래와 같은 장점을 가진다.

- 네트워크 연결성이 향상되었다. Flash-Lite 컨텐츠

는 HTTP 연결을 통해 서버와 데이터를 교환할 수 있으며 휴대폰으로 데이터를 스트리밍 할 수 있어 동적 컨텐츠 업데이트가 가능하다.

- 모바일 SVG를 지원하여 단일 플레이어로 모든 컨텐츠 재생이 가능토록 하고 있다. 이는 SVG-Tiny를 3GPP에서 의무화 하도록 규정한 이유이다.
- 네트워크 연결성, 날짜 및 시간, 진동 기능, 언어 지원, 오디오 지원 및 기타 기능을 비롯해 특정 휴대폰 기능에 접근할 수 있도록 개발자에게 제공한다. 이로 인해 사용자 맞춤형 컨텐츠의 개발이 가능하다.
- Flash-Lite 1.1 CDK(Content Development Kit)를 이용할 수 있다. 개발자들은 이 키트를 사용하여 향후 출시될 플랫폼에 알맞은 컨텐츠를 제작 및 검증할 수 있다.

Ⅲ. 벡터 그래픽스 커널 시스템 설계

3.1. 커널 시스템의 요구사항

모바일 단말기에서 벡터 그래픽스 지원을 위한 커널은 독립적인 계층 구조로써 디바이스와 사용자와의 연동을 지원하여야 한다. 이를 위하여 설계하고자 하는 벡터 그래픽스 커널은 아래와 같은 요구 사항을 만족하여야 한다.

- 모바일 단말기의 낮은 사양의 하드웨어 환경과 자원을 최적으로 사용하여야 한다.
- 기술 변화의 상이함에 의한 모바일 단말기의 특성상 다양한 하드웨어 사양에 간편한 적용이 가능해야 한다.
- 모듈화를 통하여 표준 그래픽 기능을 단위 API 함수로 구현하여야 하며 이를 통해 SVG 또는 플래시 플레이어가 구현 가능하여야 한다.
- 기존 서비스, 솔루션 및 커널과 기능상 충돌이 없어야 한다.

3.2. 커널 시스템의 설계

본 연구에서 제안된 벡터 그래픽스 커널 시스템(alexVG)은 그림1과 같이 응용 프로그램과 하드웨어 중간 계층에 위치하게 되며, 응용 프로그램은 벡터 그래픽스 커널이 제공하는 API를 통해 원하는 요구사항을 그래픽스 커널에 전달하게 되고, 그래픽스 커널 엔진은 사용자의 요구사항에 대해 내부 계산 또는 모바일 단말기의 디바이스를 이용하여 원하는 결과를 사용자에게 전달하여 준다.

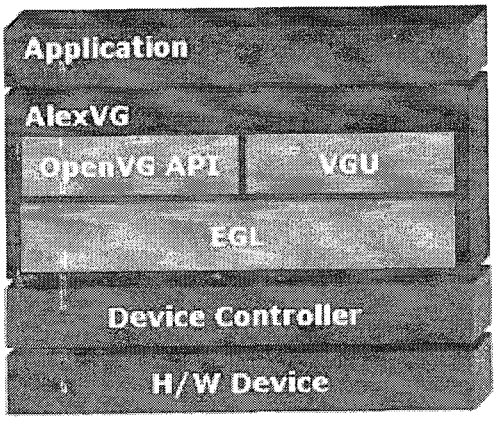


그림 1. 제안된 벡터 그래픽스 커널 시스템 구성
Fig 1. Components of Proposed Vector Graphics Kernel System

OpenVG[5,7]의 중요한 설계 원칙들 중 하나는, 플랫폼에 독립적으로 동작할 수 있어야 한다는 것이다. 이는 OpenVG 자체가 플랫폼 종속적인 최종 화면 출력, 키보드 입력 등의 입출력 루틴을 가질 수 없다는 것을 의미한다. 이를 위해 Khronos Group에서는 EGL이란 표준 API를 통해 프레임 버퍼를 할당 및 관리토록 하고 있다. OpenVG는 EGL이 생성해준 프레임 버퍼에 랜더링 결과 이미지를 저장하고 플랫폼은 EGL을 통해 프레임 버퍼에 저장되어 있는 이미지를 화면에 출력함으로써 최종적으로 사용자의 눈에 보이게 된다.

그림2는 본 논문에서 설계 구현한 alexVG 벡터 그래픽스 커널의 좀 더 자세한 구성을 보여준다. 개발자는 응용 프로그램 개발에 있을 위해 EGL API를 이용해 드로잉 파라미터들의 묶음인 Context와 최종적으로 Rendering될 이미지가 저장될 Surface를 생성한다. 이후 OpenVG API를 이용하여 아래의 요소들을 정의하게 된다.

- Path 및 Image의 정의
- Stroke 및 Fill Paint의 정의
- 위치 및 크기 정보를 담은 Transformation의 정의
- 기타 화면 출력 관련 정의

이러한 정의를 좀 더 간단하게 하기 위해 일종의 유틸리티인 VGU API를 사용할 수 있다. 각종 정의가 완료되어 OpenVG를 이용한 2D Rendering이 시작되면 OpenVG의 8단계의Pipeline을 순차적으로 실행하여 최종 이미지를 화면에 출력한다.

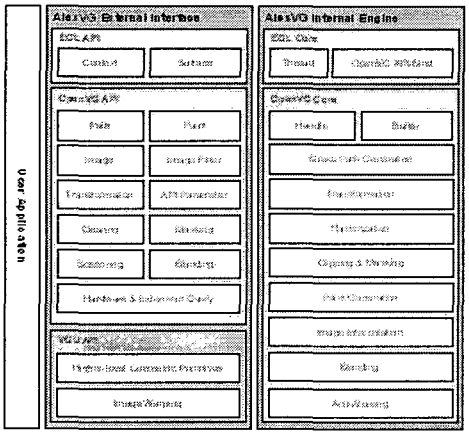


그림 2. alexVG 벡터 그래픽스 커널
Fig 2. alexVG Vector Graphics Kernel

OpenVG의 그래픽스 엔진은 기본적으로 그림 3과 같은 8 단계의 파이프라인(pipeline)으로 구성되며, 디스플레이의 특성에 따라 최종적인 화질 보정을 위한 디더링(dithering) 및 여과(filtering) 과정을 최종 단계에서 선택적으로 사용할 수 있다.

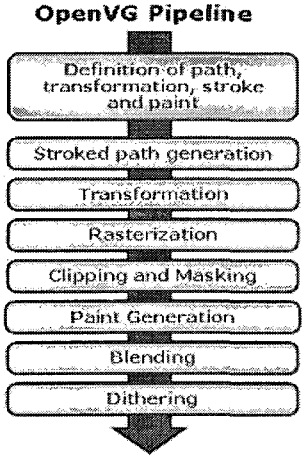


그림 3. OpenVG의 파이프라인
Fig 3. Popeline of OpenVG

기본 요소들이 렌더링되는 OpenVG 파이프라인 메커니즘은 이상적 파이프라인의 각 단계를 맞추기 위하여 구현될 필요는 없다. 적합성 검증 처리에 의하여 정의된 허용 범위 내에서 최종 결과가 일치한다면 어떤 방식의 렌더링을 적용해도 무방하다[7]. 다음은 OpenVG 파이프라인의 각 단계별 역할이다.

- 1 단계(드로잉 파라메터 및 좌표 설정) : 사용자가 원하는 좌표, 도형 등을 입력하고, 원하는 드로잉 파라미터를 결정한다. 예를 들면, 스트로크(stroke)하는 선의 두께 설정, 스트로크, 채움(fill)을 위한 색상의 결정, 방법의 결정, 변형 행렬의 적재 등이다.
- 2 단계(스트로크된 경로 생성) : 사용자가 입력한 경로를 스트로크해야 한다면 1 단계의 스트로크한 선의 두께만큼 offset을 수행하여 새로운 경로를 생성한다. 만일 dash가 설정되어 있다면 이를 반영한 새로운 경로를 생성한다.
- 3 단계(변형) : 사용자 좌표로부터 화면 좌표로 변환하는 과정으로 3x3의 행렬과 벡터간의 곱셈으로 표현된다. 그래픽 파이프라인 중 실제로 가장 많은 연산을 요구하는 부분이다.
- 4 단계(래스터라이제이션: rasterization) : 사용자의 경로를 채움 혹은 스트로크하여 실제로 화면의 각 픽셀이 On/Off인지 결정한다. 화면에 대한 프레임 메모리의 alpha 값을 바꾸어 줌으로써 이를 수행한다. 벡터 데이터를 비트맵 데이터로 변환하는 첫 번째 단계이다.
- 5 단계(클리핑과 마스킹) : 화면의 원하는 영역만 디스플레이하기 위하여 클리핑(clipping)과 마스킹(masking)을 수행한다.
- 6 단계(페인트 생성) : 5 단계에서 만들어진 결과물을 이용하여 실제 사용할 색을 결정한다. 단색 채움, 단색 스트로크, 그래데이션(gradation)을 이용한 채움 및 스트로크, 패튼을 이용한 채움 및 스트로크 등을 수행한다.
- 7 단계(혼합: blending) : 새롭게 생성된 페인트 생성과 이전의 화면을 합성하는 단계로, 위/내부/외부/아래/투명 등을 이용한 다양한 혼합이 가능하다.
- 8 단계(디더링: dithering) : 화면의 화소 특성에 따라 필요한 작업으로 반복되는 패턴에 의해서 발생하는 문제점을 해결하기 위해 잡음 분산 필터(noise diffusion filter)를 이용하여 디더링을 수행한다. 화

면 픽셀의 특성상 이 과정이 필요하지 않은 경우도 있을 수 있다.

이와 같은 8 단계의 파이프라인을 어플리케이션 개발 측면에서 분석해 보면 개발자는 1 단계에서 OpenVG API를 이용하여 출력하길 원하는 경로 또는 이미지 등의 객체에 대해 드로잉 파라메터를 설정한다. 2~6 단계에서 드로잉 함수(vgDrawPath, vgDrawImage)를 호출하면 엔진에서 사용자가 정의한 드로잉 파라메터를 분석하여 객체를 구체적인 비트맵 정보로 변환한다. 7 단계에서 이미 출력된 비트맵과의 합성이 이루어진다. 각 객체에 대해서 이러한 과정을 반복적으로 수행하게 된다. 마지막 단계인 8 단계에서 최종 비트맵을 LCD에 출력 시 필요에 따라 디더링 단계를 거친다.

Ⅳ. 시스템 구현 및 검증

4.1. OpenVG 엔진의 PC 환경에서의 구현

실제 OpenVG 표준을 따르는 그래픽 파이프라인 엔진의 단말기 구현에 앞서 PC환경에서 이를 구현하였다. 이 엔진은 다양한 그래픽기능의 개발 및 테스트 환경으로 사용 가능할 뿐만 아니라 고수준의 벡터 그래픽 응용프로그램을 개발하는 데에도 유용하게 사용할 수 있도록 개발되었다.

그림4는 alexVG의 Function List를 보여준다.

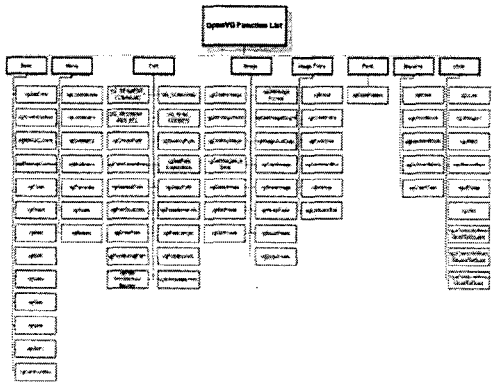
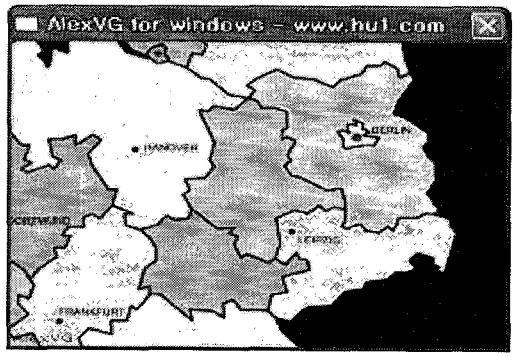


그림 4. alexVG의 Function List
Fig 4. Function List of alexVG

- Base Mudule은 커널 의 기본 기능을 수행 하는데 필요한 함수들로 에러 처리 및 복구 이벤트 발생, 각종 그래픽스 Parameter의 Set/Get, Drawing을 수행하는 기본 Context의 Create/Destroy를 담당한다.
- Matrix Module은 기본 수학 모듈들을 포함하고 있으며, 최적화된 행렬의 연산 및 Transform 함수(Scale, Rotate, Translate) 들이 있다.
- Path Module은 엔진내부에 Path Data Structure를 만들고 사용자의 Path를 Engine 내부의 Path로 변환하는 기능과 각종 Path Manipulation 함수들이 들어 있다.
- Image Module은 Image Memory의 관리, Image의 복사, 이동, Child Image의 관리 등의 기능들을 수행하며 Image Resampling, Transformation에 관련된 함수로 구성되어 있다.
- Image Filter는 Image Convolution을 기반으로 하는 각종 Image Filter를 갖고 있다. Kernel의 크기는 자유롭게 정의할 수 있다. 또한 Color Space 변환을 위한 함수, Color Look-up Table 의 기능을 포함한다.
- Paint module은 그래픽 파이프라인에 대한 처리를 위한 함수들로 구성되어 있다.
  Blending module은 Blending 및 Masking을 위한 함수를 포함한다.
- Utility module은 사용자가 편리하게 API를 사용할 수 있도록 하는 함수들로 자주 사용되는 도형, 필터 등을 포함한다. 이러한 함수를 이용하여 사용자는 간편하게 필요한 그래픽을 생성할 수 있다.

그림 5는 PC에서 구현된 OpenVG 엔진으로 표현 가능한 그래픽스 기능의 예(Path Rendering)를 보여 준다.



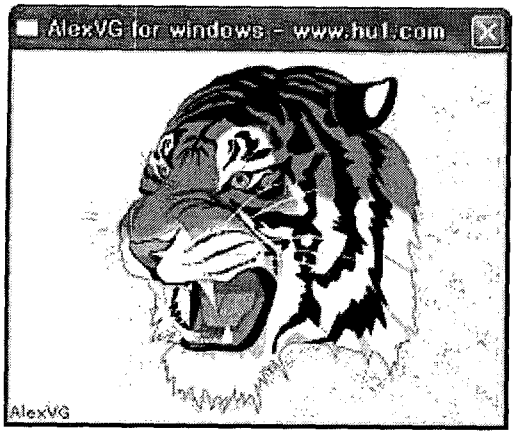


그림 5. 구현된 alexVG의 동작 상태(PC 환경)
Fig 5. Running State of Implemented alexVG (PC Environment)

4.2. OpenVG 엔진의 ARM9 환경에서의 개발

PC 환경에서의 개발 결과를 바탕으로 하여 엔진 내부의 수치 처리를 부동소수점 연산에서 고정소수점 연산으로 변경하고, 자원의 사용을 최소화 하여 무선 단말기 환경에서 수행 가능하도록 개발 되었다. OpenVG 표준이 갖는 영상의 후처리 과정의 복잡성으로 인하여 모바일 단말기는 영상의 처리과정에서 낮은 성능을 나타내었으나 벡터 그래픽 처리의 경우 충분히 활용 가능한 성능을 나타내었다.

본 논문에서 구현한 벡터 그래픽 커널인 alexVG를 ARM Core[3]가 탑재된 개발보드에 적용하였다. 본 개발보드는 기본적으로 ARM926EJ-S 프로세서를 탑재하고 있으며 CPU 속도는 210Mhz이다. 64MB의 NOR Flash와 128MB의 SDRAM을 가지고 있다. 아울러 본 개발보드는 자체적으로 GSM 모듈을 가지고 있으며, 추가적인 인터페이스 보드를 통해 휴대단말기와 유사한 형태의 개발 환경을 제공한다. 이를 통해 본 그래픽 커널이 임베디드 시스템 및 휴대 단말기에 효율적인지 테스트하기 위함이다. 그림6은 ARM Core 탑재 개발보드에 적용된 벡터 그래픽 커널인 alexVG의 동작 상태에 대한 예(Path Rendering 데모 화면)를 보여준다.

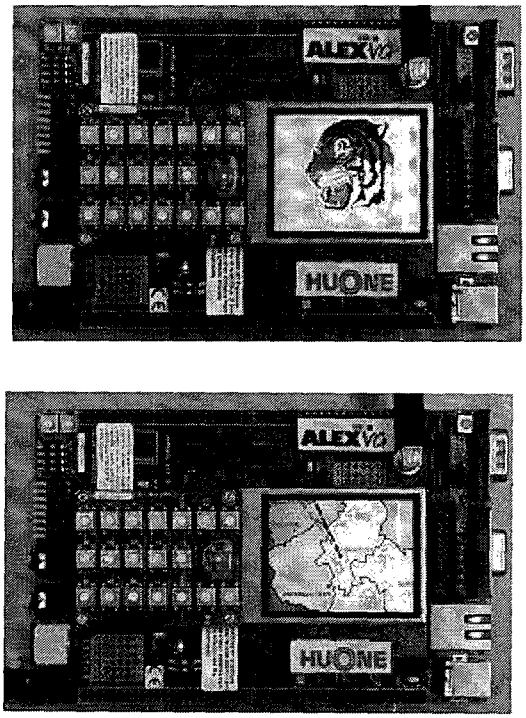


그림 6. 구현된 alexVG의 동작 상태 예(ARM Core 개발보드 환경)
Fig 6. Running State example of Implemented alexVG(ARM Core Development Board Environment)

4.3. alexVG의 성능 측정

구현된 alexVG의 성능을 측정하기 위하여, 위에서 언급한 2개의 개발환경에서 305개의 Path로 구성된 PostScript의 Tiger 이미지를 하나의 Frame으로 구성하고, 이를 100회 반복 출력하면서 그 시간을 측정하고, 이를 [표 1]에 나타내었다. OpneVG 표준이 최근에 제안되었으며, OpenVG 기반의 벡터 그래픽스 커널 구현에 관한 보고가 국내외적으로 아직 보고되지 않고 있기 때문에, 본 논문에서 구현한 커널의 성능과 비교할 수 는 없다. 그러나, 표 1에서 보듯이, 주어진 데이터의 Path의 양과 해상도의 크기를 볼 때, ARM Core 환경에서도 충분한 성능을 보여준다고 생각되며, 이후 코드 최적화 및 Floating Point를 Fixed Point로 변경할 때[1], 추가적인 성능 향상이 가능할 것으로 기대된다.

표 1. 구현된 alexVG의 성능 측정
Table 1. Performance Measures of Implemented alexVG

	PC 환경	ARM Core 환경
CPU	1.7Ghz	210Mhz
RAM	512MB	128MB
해상도	320 * 240	320 * 240
평균 Rendering Time	163ms	1,547ms
Frames/sec	6.1 fps	0.6 fps

V. 결 론

모바일 정보통신 단말기의 급속한 발전과 사용자들의 다양한 요구로 인하여, 이미지를 포함한 멀티미디어 정보가 모바일 통신에서 컨텐츠의 기반을 이루고 있다. 벡터 그래픽스 방식의 이미지 정보를 효율적으로 이용하기 위해서는 이를 지원하는 시스템에 대한 설계와 구현이 보다 기술적이고 효율적으로 이루어 져야 하기 때문에, 많은 벡터 그래픽스 커널 시스템들이 제안 개발되고 있으며, 그래픽스 응용 프로그램들을 위한 호환성을 높이기 위하여 벡터 그래픽스 커널에 대한 표준화 작업이 진행되고 있다.

본 논문에서는 자원 제한적인 모바일 단말기에 적합한 벡터 그래픽스 커널의 요구 사항을 살펴보고, 표준으로 제안된 Khronos Group의 OpenVG 기반 벡터 그래픽스 커널을 설계 구현하였다. 구현된 OpenVG 커널인 alexVG는 PC 환경과 ARM926EJ-S 프로세서 탑재 개발보드 환경에서 각각 구현되었다. 구현된 alexVG의 성능을 측정하기 위하여, 위에서 언급한 2개의 개발 환경에서 하나의 Frame으로 구성된 이미지를 100회 반복 출력하면서 그 시간을 측정하였다. OpneVG 표준이 최근에 제안되었고, OpenVG 기반의 벡터 그래픽스 커널 구현에 관한 보고가 국내외적으로 아직 보고되지 않고 있기 때문에, 본 논문에서 구현한 커널의 성능과 비교할 수 는 없다. 그러나, 주어진 데이터의 Path의 양과 해상도의 크기를 볼 때, ARM Core 환경에서도 충분한 성능을 보여준다고 생각되며, 이후 코드 최적화 및 Floating Point를 Fixed Point로 변경할 때 추가적인 성능 향상이 가능할 것으로 기대된다.

참고문헌

[1] 백낙훈 외, "고정소수점 수치 자료를 사용하는 벡터 그래픽스 연산에서의 오차 전달 모델", 한국통신소프트웨어학회 학술대회, pp. 3-7, 2005년 7월.

[2] 이준영, 이환용, 박기현, 장명숙, 우종정, "모바일 환경 통신을 위한 벡터 그래픽스 커널의 설계," 한국모바일학회 논문지, 2권 1호, pp. 73-80, 2005년 6월.

[3] ARM, "Fixed Point Arithmetic on the ARM", Application Note 33, ARM, September 1996.

[4] David Hough, "Applications of the Proposed IEEE-754 Standard for Floating Point Arithmetic", IEEE Computer, Vol.14, no.3, pp.70-74, 1981년 3월.

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

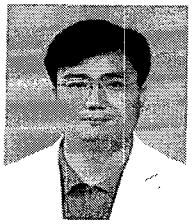
[6] Mark Segal and Kurt Akeley, " The OpenGL graphics system: A specification," Tech. Report, Silicon Graphics Computer Systems, 1992.

[7] OpenVG - The Standard for Vector GraphicsAcceleration, http://www.khronos.org/openvg

[8] Troy Evans, "Introducing Macromedia Flash Lite 1.1," Macromedia.

[9] W3C, "Scalable Vector Graphics (SVG) Tiny 1.2 Specification Draft 13," W3C SVG Workgroup, April 2005.

저자소개



이 환 용(HwanYong Lee)

한국과학기술원 전자계산학과 학사
포항공대 전자계산학과 석사, 박사
서울여대/계명대학교 정보통신대학 겸임교수

현재: ㈜엠티스 부설연구소 연구 소장
※ 관심분야 : 모바일통신, 컴퓨터 그래픽스, 유비쿼터스 컴퓨팅 등

박 기 현(KeeHyun Park)

경북대학교 전자공학과 학사
한국과학기술원 전자계산학과 석사
미국 밴드빌트 대학교 컴퓨터공학과 박사

현재: 계명대학교 컴퓨터공학부 교수
※ 관심분야 : 병렬운영체제, 모바일통신 소프트웨어, 성능분석 등

우 종 정(JongJung Woo)

경북대학교, 전자공학과 학사
텍사스주립대학교(Austin) 전기 컴퓨터공학과 석사
텍사스주립대학교(Austin),전기 컴퓨터공학과 박사

현재 성신여자대학교 컴퓨터정보 학부 교수
※ 관심분야 : 컴퓨터구조, 병렬처리, 임베디드시스템, e-Learning