한국산학기술학회논문지
Vol. 10, No. 9, pp. 2334-2339, 2009

임베디드 시스템을 위한 터치스크린 패널의 터치 영역 인식 기법의 성능 비교

오삼권[1], 박근덕[1*], 김병국[1]
[1]호서대학교 컴퓨터공학부

Performance Comparison of the Recognition Methods of a Touched Area on a Touch-Screen Panel for Embedded Systems

Sam Kweon Oh[1], Geun Duk Park[1*] and Byoung Kuk Kim[1]
[1]Division of Computer Engineering, Hoseo University

요 약 터치스크린 기능이 있는 LCD 패널이 부착된 임베디드 시스템의 경우, 사용자 입력 명령의 전달을 위해 사각형, 오각형, 원, 화살표 같은 다양한 도형들이 자주 사용된다. 이 경우, 특정 명령 입력 여부의 판단을 위해 터치된 지점이 그 명령을 의미하는 도형 내에 있는지의 여부를 판단하는 알고리즘이 필요하다. 그러나 이런 알고리즘들은 제한된 컴퓨팅 자원을 갖는 임베디드 시스템의 경우 상당한 오버헤드를 유발 할 수 있다. 본 논문의 목적은 현재 널리 쓰이고 있는 터치 영역 인식 알고리즘들을 구현하고, 그 성능을 평가하여 가장 효율적인 인식 방법을 제시하는데 있다. 따라서 본 논문은 우선 터치스크린 LCD 모듈이 부착된 LN2440SBC 임베디드 보드를 위한 터치스크린의 초기화 및 구동 방법과 터치스크린의 좌표를 LCD 패널의 좌표에 맞춰 조절하는 좌표보정 방법을 설명하고, 다음에 도형 영역의 터치 여부의 판단을 위해 사용 되고 있는 사각형의 범위 검사법, 오각형과 같은 다각형의 교차 수(crossing number) 검사법, 원의 거리 측정법, 그리고 모든 도형들에 적용 가능한 색상 비교법을 구현한다. 이 방법들의 성능 평가를 위해, 사각형, 오각형, 원, 그림 등을 그리기 위한 이차원 그래픽스 함수들을 구현하고 도형을 생성한 후, 각 방법에 따라 해당 도형들의 영역 터치 여부를 판단하는데 걸리는 시간을 측정한다. 이 측정 결과, 사각형은 범위 검사법이, 원은 거리 측정법이, 다각형 및 그림을 이용한 도형일 경우에는 색상 비교법이 가장 적합한 것으로 나타났다.

Abstract In case of an embedded system having an LCD panel with touch-screen capability, various figures such as rectangles, pentagons, circles, and arrows are frequently used for the delivery of user-input commands. In such a case, it is necessary to have an algorithm that can recognize whether a touched location is within a figure on which a specific user-input command is assigned. Such algorithms, however, impose a considerable amount of overhead for embedded systems with restricted amount of computing resources. This paper first describes a method for initializing and driving a touch-screen LCD and a coordinate-calibration method that converts touch-screen coordinates into LCD panel coordinates. Then it introduces methods that can be used for recognizing touched areas of rectangles, many-sided figures like pentagons, and circles; they are a range checking method for rectangles, a crossing number checking method for many-sided figures, a distance measurement method for circles, and a color comparison method that can be applied to all figures. In order to evaluate the performance of these methods, we implement two-dimensional graphics functions for drawing figures like triangles, rectangles, circles, and images. Then, we draw such figures and measures times spent for the touched-area recognition of these figures. Measurements show that the range checking is the most suitable method for rectangles, the distance measurement for circles, and the color comparison for many-sided figures and images.

Key Words : Embedded System, Touch Screen, Calibration, Bresenham Algorism, Range Checking, Crossing Number Checking, Color Comparison

*교신저자 : 박근덕(gdpark@hoseo.edu)
접수일 09년 07월 23일 수정일 (1차 09년 09월 01일, 2차 09년 09월 11일) 게재확정일 09년 09월 16일

1. 서론

하드웨어의 발전에 따라 많은 임베디드 시스템에서 터치스크린 LCD가 사용 되고 있으며, 이런 임베디드 시스템의 경우, 사용자 입력 명령의 전달을 위해 사각형, 오각형, 원, 화살표 같은 다양한 도형들이 자주 사용된다. 이 경우, 특정 명령에 대한 입력 여부의 판단을 위해 터치된 지점이 그 명령을 의미하는 도형 내에 존재하는지의 여부를 판단하는 알고리즘이 필요하다. 그러나 이런 알고리즘들은 제한된 컴퓨팅 자원을 갖는 임베디드 시스템의 경우 상당한 오버헤드를 유발 할 수 있다. 따라서 현재 널리 쓰이고 있는 터치 영역 인식 기법들을 구현하고 성능 비교 후, 그 결과를 근거로 각 상황에 적합한 판단 방법을 제시한다.

본 논문은 터치스크린 LCD 모듈인 LP35가 부착된 LN2440SBC 임베디드 보드[1]를 위한 터치스크린 초기화 및 구동 방법과 터치스크린 좌표를 LCD 패널 좌표에 맞춰 조절하는 좌표 보정 방법(calibration)[2]을 설명하고, 도형 영역의 터치 여부를 판단하기 위한 방법으로, 임의의 점이 사각형 영역의 X좌표와 Y좌표의 범위 내에 존재하는지 검사하는 사각형의 범위 검사법과 임의의 점의 오른쪽으로 무한한 길이의 선분을 긋고, 이 선분이 도형의 변과 교차하는 횟수가 홀수번인지, 짝수번인지에 따라 판단하는 다각형의 교차 수 검사법[3,4], 임의의 점과 원점의 거리를 구한 후, 반지름의 길이와 비교함으로써 판단하는 원의 거리 측정법, 그리고 배경화면 색과 도형의 색을 달리 하여, 임의의 점의 색과 도형의 색을 비교함으로써 모든 도형에 적용 가능한 색상 비교 방법을 구현한다.

이 방법들의 성능 평가를 위해 사각형, 오각형, 원, 그림 등을 그리기 위한 이차원 그래픽스 함수들을 구현하고, 이를 이용해서 도형을 생성한 후, 각 방법에 따라 해당 도형들의 영역 터치 여부를 판단하는데 소요되는 시간을 측정한다. 이 측정된 시간을 근거로 각 방법의 사용에 적합한 상황을 판단한 결과, 사각형은 범위 검사법이, 원은 거리 측정법이, 다각형과 그림은 색상 비교법이 적합한 것으로 나타났다.

본 논문의 구성은 다음과 같다. 2장은 기존의 터치 영역 인식 기법과 LN2440SBC의 터치스크린 인터페이스에 대해 설명한다. 3장은 터치스크린의 구동을 위한 초기화 방법과 좌표보정 방법 그리고 터치 영역 인식기법 및 이차원 그래픽스 라이브러리 함수의 구현을 설명 한다. 4장에서는 성능평가를 하며, 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 터치 영역 인식 기법

터치스크린을 사용한 임베디드 시스템의 경우 도형으로 표현된 특정 영역의 터치 인식을 판단하기 위해 다음과 같은 방법들이 사용된다.

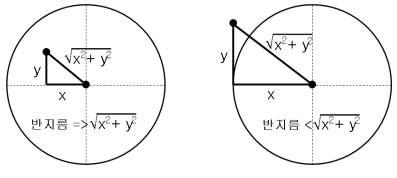
일반적으로 사용되는 사각형 영역의 경우, 임의의 점이 사각형 영역의 최소 및 최대 X, Y 좌표의 범위 내에 존재하는지를 검사하는 범위 검사법이 사용된다. 범위 검사법은 구현이 단순한 반면, 사각형 이외의 도형에는 적합하지 않다는 단점이 있다.

오각형 이상의 다각형 영역의 경우에는 교차 수 검사법이 사용 된다. 교차 수 검사법은 그림 1과 같이 임의의 점의 오른쪽으로 무한한 길이의 가상 선분을 긋고, 이 가상 선분이 도형의 변과 홀수 번 교차하면 점이 도형의 내부에, 짝수 번 교차하면 외부에 존재한다. 이 방법의 경우, 오른쪽으로 가상의 선을 그어, 다각형의 모든 변과 교차하는지를 계산해야 해야 하는 오버헤드가 존재한다.

a. 도형 내부에 존재 b. 도형 외부에 존재

[그림 1] 교차 수 검사법

원형의 영역인 경우, 도형의 둘레가 직선이 아니라 호로 이루어져 있으므로 교차 수 방법을 적용 할 수 없다. 따라서 원의 거리 측정법을 사용한다. 거리 측정법은 그림 2와 같이 피타고라스 정의를 이용하여 임의의 점과 원 사이의 거리를 구하고 이 거리가 원의 반지름 보다 작거나 같다면 원의 내부에, 크다면 외부에 존재하는 것으로 판단한다. 이 방법 또한 원 이외의 도형에는 적합하지 않다는 단점이 있다.



a. 도형 내부에 존재 b. 도형 외부에 존재

[그림 2] 거리 측정법

범위 검사법과 거리 측정법, 특히 교차 수 검사법은 도형 영역에 임의의 점의 존재여부를 판단하기 위해 많은 연산이 필요하며, 다각형, 원 등 도형마다 다른 방법들을

적용해야 한다. 이와 달리, 색상 비교 법은 모든 도형에 적용 가능한 효율적이고 방법으로서, 하트, 화살표 등의 다양 그림에도 적용할 수 있다. 색상 비교 방법은 그림 3과 같이 도형 내부의 색과 외부의 색을 다르게 한 후, 터치된 픽셀의 색이 도형 내부의 색인지 외부의 색인지를 비교함으로써 판단 할 수 있다. 또한 색상 비교법은 컬러 LCD뿐 아니라 기본 흑백 LCD와 그레이 코드(gray code)를 적용할 수 있는 흑백 LCD에서도 적용 가능하다. 하지만 도형의 색이 한 가지 이상의 색인 경우에는 적용하기 힘들다는 단점이 있다.

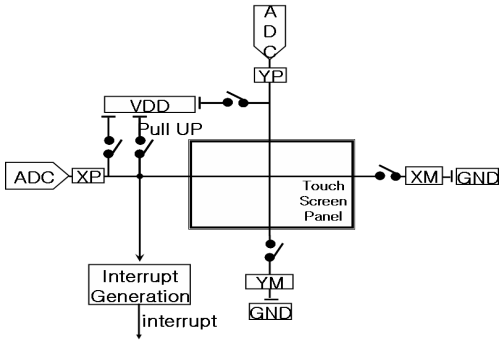
[그림 3] 색상 비교를 이용한 영역 판단 방법

2.2 LN2440SBC 임베디드 시스템

LN2440SBC 임베디드 시스템의 S3C2440A 마이크로프로세서는 ARM920T ARM 코어, 아날로그-디지털 변환기(ADC)[5], 터치스크린 인터페이스, TFT LCD 컨트롤러 등이 내장되어 있으며, 터치스크린 기능의 LP35 TFT LCD 모듈이[6] 부착되어 있다.[1,2]

S3C2440A에는 8 채널의 ADC가 내장되어 있으며, 이 중 4-7번 채널(YM, YP, XM, XP)은 터치스크린 인터페이스로 사용할 수 있다. 그림 4는 터치스크린의 인터페이스를 보여준다. 풀업(PULL UP), YP, XP는 스위치(FET)를 활성화할 경우, 전원(VDD)에 연결되며, YM, XM는 스위치를 활성화할 경우, 접지(GND)에 연결된다. 터치스크린이 눌리지 않은 대기상태에서는 X축과 Y축은 서로 분리되어 있다. 따라서 터치 인터럽트를 발생시키기 위해 X축의 풀업과 YM의 스위치만을 활성화 하여, 풀업은 전원, YM은 접지로 연결되게 한다. 터치가 입력되면 X축과 Y축은 서로 접촉되어 회로가 구성되어 전압이 생기며 그 전압에 의해 터치스크린 인터럽트가 발생한다.

터치스크린은 스크린 표면을 눌렀을 때 생기는 전압차로 상대적인 좌표를 계산하며 이 좌표 값은 아날로그 신호이므로 ADC를 통해 디지털 값으로 변환해야 한다. S3C2440A가 제공하는 변환 모드는 터치스크린을 사용하지 않고 일반적인 ADC로 사용하는 보통(normal) 변환 모드, X/Y 위치를 자동으로 변환해주는 자동(auto) X/Y 위치 변환 모드가 있으며, 터치스크린 인터럽트를 대기하는 인터럽트 대기 모드가 있다.



[그림 4] 터치스크린 인터페이스

3. 구현

3.1 터치스크린 구동을 위한 초기화 방법

터치스크린을 구동하려면 ADC와 터치스크린의 제어 레지스터들을 설정해야 한다. 터치스크린의 터치 인터럽트를 대기하도록 하고, 인터럽트의 발생 시 터치된 위치에 대한 아날로그 신호의 디지털 변환을 위해 다음과 같이 설정한다.

① 자동 X/Y 위치 변환 모드 설정
② 인터럽트 대기 모드 설정
③ 풀업(pull-up) 스위치와 YM 스위치 활성화

터치스크린 인터럽트를 대기하기 위해 인터럽트 대기 모드를 설정하고, 인터럽트 발생 시 터치된 X, Y의 아날로그 위치 값의 디지털 값으로의 변환을 위해 자동 X/Y 위치 변환 모드를 설정한다. 인터럽트 대기상태에서 터치 인터럽트의 발생여부의 확인을 위해, 터치스크린에서 사용하는 ADC의 4개 채널(YP, YM, XP, XM)에 연결된 스위치와 XP에 연결된 풀업 스위치의 활성화 여부를 설정해야 한다. 인터럽트 대기상태에서는 풀업과 YM 스위치만 활성화하여 스크린이 터치 될 경우, X축과 Y축이 접촉되어 인터럽트를 발생하도록 설정한다.

인터럽트가 발생하여, 인터럽트 서비스 루틴(ISR)이 실행 될 경우, 다음과 같이 설정한다.

① 풀업 스위치 비활성화
② 자동 X/Y 위치 변환 비활성화
③ 인터럽트 대기 모드 비활성화

터치 인터럽트의 마스킹을 위해 풀업 스위치를 비활성화 하고, 자동 X/Y 변환 모드와 인터럽트 대기모드를 해제 한 후, 데이터 레지스터로부터 변환된 X, Y 좌표를 읽어 온다. 모든 처리가 끝난 후 다시 인터럽트를 대기하도록 설정한다.

ADC에 의해 디지털 신호로 변환된 좌표(XP, YP)의 범위와 LCD 픽셀좌표(x,y)의 범위는 서로 다르므로, 터치스크린 패널의 좌표를 LCD의 픽셀 좌표에 대응 되도록 하는 보정 기능이 필요하다. 두 점의 좌표를 알면, 1차 함수 y=f(x)를 구할 수 있다. 따라서 좌측 상단의 픽셀(x1,y1)과 우측 하단의 픽셀 좌표(x2,y2)에 대응되는 터치스크린 패널의 위치((xp1,yp1),(xp2,yp2))를 구하고, LCD 픽셀 좌표와 터치스크린 패널의 좌표는 비례 관계이므로, 이 좌표들을 이용하여 터치스크린 패널의 위치에 대응되는 픽셀 좌표 x와 y를 식 (1)과 같이 구할 수 있다:

$$x=mx*xp+Bx \quad y=my*yp+By \tag{1}$$

여기서 mx와 my는 기울기이고 Bx와 By는 절편이다. 기울기는 (y의 증분/x의 증분) 이므로 mx와 my는 식 (2)와 같이 구할 수 있다:

$$mx=lx/(xp2-xp1) \quad my=ly/(yp2-yp1) \tag{2}$$

여기서, lx는 두 픽셀 간의 X축 길이이고 ly는 두 픽셀 간의 Y축 길이이다. 또한 절편은 식 (1)의 수식에 (x1,y1)에 대응되는 (xp1,yp1)를 대입하여 식(3)과 같이 구할 수 있다:

$$Bx=x1-(mx*xp1) \quad By=y1-(my*yp1) \tag{3}$$

3.2 터치 영역 인식 기법 구현

다음은 터치된 위치의 픽셀 좌표가 그 명령을 의미하는 도형 내에 존재하는지의 여부를 판단하기 위해 사용되는 알고리즘의 구현을 설명한다.

정사각형 혹은 직사각형일 경우, 터치된 픽셀의 X, Y 좌표가 사각형의 최소 X, Y 좌표와 최대 X, Y 좌표 범위 내에 존재하는지 검사함으로써 간단히 판단 할 수 있다. 그러나 오각형 같은 다각형의 경우처럼, 범위 검사 방법으로는 판단 할 수 없는 경우에는 교차 수 방법을 사용한다.

다음은 교차 수 검사법의 의사 코드이다. 인자로 다각형의 꼭지점들의 좌표와 꼭지점 수, 터치된 픽셀 좌표를 입력 받는다. 우선 터치된 픽셀의 오른쪽으로 가상의 선을 생성하고, 다각형의 변을 이루는 모든 직선과 가상의 선의 교차 여부를 검사한다. 교차된 변의 수가 짝수면 터치된 픽셀은 도형의 내부에, 홀수면 외부에 존재한다.

```
Crossing_Num(꼭지점들의좌표, 꼭지점수, 터치된좌표)
begin
  가상선=가상선생성(터치된 좌표)
  while(교차여부검사횟수<=꼭지점수)
  begin
    다각형의변=다각형의변구하기(이웃한 두 꼭지점 좌표)
    교차여부검사(다각형변, 가상선)
  end
  if 교차된변의수==짝수 then 도형의 외부에 존재
  else 도형의 내부에 존재
end
```

다음은 원의 거리 측정법의 의사 코드를 보여준다. 인자로 원의 중심점 좌표와 반지름 그리고 터치된 픽셀 좌표를 입력 받는다. 피타고라스 정의에 의해 터치된 픽셀과 원점의 거리를 구하며, 이 두 픽셀의 거리가 반지름보다 작거나 같은 경우에는 터치된 픽셀이 원의 내부에, 큰 경우에는 외부에 존재한다.

```
RangeChecking(중심점좌표, 반지름, 터치된좌표)
begin
  거리=거리구하기(중심점, 터치된좌표)
  if 거리=<반지름 then 도형의 내부에 존재
  else 도형의 외부에 존재
end
```

다음은 색상 비교법의 의사코드를 보여준다. 인자로 도형 내부의 색과 터치된 픽셀의 좌표를 입력 받는다. 색상 비교법은 현재 LCD의 화면 정보가 저장된 프레임버퍼에서 터치된 픽셀의 색상 값을 읽어와 도형 내부의 색과 비교하여, 도형 내부의 색과 같으면 내부에, 다르면 외부에 존재하는 것으로 판단한다.

```
ColorComparison(도형내부색, 터치된좌표)
begin
  터치된지점의색=픽셀색구하기(터치된좌표)
  if 터치된지점의색==도형내부의색
  then 도형의 내부에 존재
  else 도형의 외부에 존재
end
```

3.3 단순 이차원 그래픽스 라이브러리 함수

도형 영역의 터치 여부를 판단하기 위한 방법들의 성능평가를 위해, 선, 사각형을 포함한 다각형, 원 같은 기

본적인 이차원 도형들과 그림을 출력 할 수 있는 함수들을 구현한다. 표 1은 본 논문에서 구현한 이차원 그래픽스 라이브러리 함수들을 보여준다.

[표 1] 이차원 그래픽스 라이브러리 함수

함수명	인 자	기 능
DrawLine	• 시작점 좌표 • 끝점 좌표 • 선 색상 값	선을 그리는 함수
DrawPolygon	• 꼭지점들의 좌표 • 꼭지점의 수 • 선 색상 값	다각형을 그리는 함수
DrawCircle	• 중심점 좌표 • 반지름 • 선 색상 값	원을 그리는 함수
DrawBMP	• 출력 시작좌표 • 비트맵 파일의 파일 포인터	BMP 이미지를 그리는 함수

선을 그리는 DrawLine 함수와 원을 그리는 DrawCircle 함수는 Bresenham의 선, 원 알고리즘[7,8]을 사용한다. 방정식을 이용하여 선과 원을 그릴 경우 발생하는 실수 연산의 느린 연산속도 문제를 해결하기 위해 고안된 Bresenham 알고리즘은 정수 계산을 이용해 실제 직선과 원에 가까운 좌표를 찾아 그리는 효율적인 선, 원 생성 알고리즘이다.

사각형을 포함한 다각형을 그리는 DrawPolygon함수는 도형의 꼭지점들의 좌표와 꼭지점의 수 그리고 색상 값을 인자로 입력받아 각 꼭지점을 DrawLine 함수를 이용하여 연결시킴으로서 쉽게 구현할 수 있다.

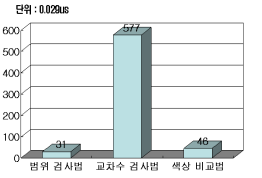
비트맵 이미지를 출력하는 DrawBMP 함수는 비트맵 파일 포인터를 이용하여 비트맵 파일의 헤더 영역에서 그림의 수직, 수평 크기를 구하여 전체 픽셀의 개수를 계산하고, 첫 픽셀부터 마지막 픽셀까지 픽셀의 색상 값을 읽어와 해당 좌표의 픽셀에 출력 시킴으로써, 그림 전체를 LCD 화면에 출력 시킨다.

4. 성능평가

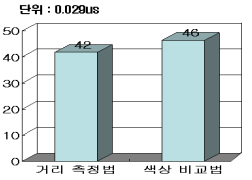
성능 평가는 터치스크린 모듈인 LP35가 부착된 LN2440SBC 임베디드 보드에서 측정한다. 코어의 속도는 최고속도인 405Mh로 설정하였다. 측정은 PWM(pulse width modulation) 타이머를 이용하며, 세밀한 측정을 위해 타이머 클럭을 35Mh로 설정하여, 1틱 당 0.029us의 시간이 소요 되도록 하였다.

터치스크린을 이용하여 임의의 위치를 터치한 후, 터치된 픽셀 좌표를 획득한 시점부터 터치된 픽셀이 도형의 영역에 존재하는지를 판단하는데 소요되는 시간을 측정했다. 사각형은 범위 검사법, 색상 비교법, 교차 수 검사법을 비교하고, 원형은 거리측정법과 색상 비교법을 비교하며, 다각형은 교차 수 검사법과 색상 비교법을 비교했다.

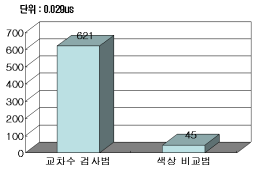
그 결과 알고리즘별 판단 소요시간은 그림 5와 같다. a는 사각형, b는 원, c는 오각형, d는 다각형의 변이 증가함에 따른 소요시간을 보여준다. 즉, 사각형의 판단 소요시간은 범위 검사법이, 원형은 거리 측정법이. 다각형은 색상 비교법의 속도가 가장 빠르게 나타났다. 교차 수 검사법은 색상 비교법과 달리 교차 여부를 검사해야하는 다각형의 변의 수가 증가함에 따라 판단 소요시간이 증가 했다. 따라서 사각형은 범위 검사법이, 원형은 거리 측정법이, 다각형 및 화살표와 같은 그림은 색상 비교법이 적합하다.



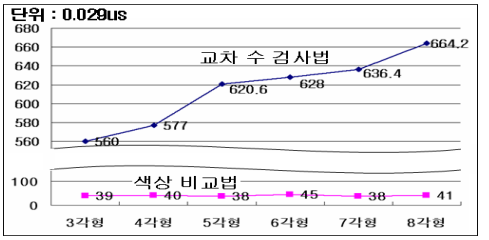
a. 사각형



b. 원



c. 다각형(오각형)



d. 다각형 변의 증가에 따른 소요시간

[그림 5] 알고리즘별 소요시간

5. 결론

본 논문은 터치스크린 모듈인 LP35가 부착된

LN2440SBC 임베디드 보드를 사용하여 터치스크린의 구동을 위한 초기화 방법 및 좌표 보정 방법을 설명했다. 또한, 제한된 컴퓨팅 환경을 가지는 임베디드 시스템에서 사용자 명령 입력을 위해 도형들을 사용할 경우, 명령 입력 판단 방법에 따라 성능에 영향을 끼칠 수 있으므로, 각 상황에 적합한 판단 방법을 제시하기 위해 도형으로 표현된 특정 영역의 터치여부의 판단 방법으로 사용되고 있는 사각형의 범위 검사법, 다각형의 교차 수 검사법, 원의 거리 측정법, 그리고 이런 모든 도형들에 적용 가능한 색상 비교법을 구현하고, 이 방법들의 성능 평가를 위해 이차원 그래픽스 함수들을 구현 했으며, 이를 이용하여 도형을 생성한 후, 각 방식에 따라 도형들의 영역 터치 여부를 판단하는데 소요되는 시간을 측정했다. 이 측정 결과 사각형은 범위 검사법이, 원은 거리 측정법이, 다각형 및 그림을 이용한 도형일 경우에는 색상 비교법이 가장 적합한 것으로 나타났다.

참고문헌

[1] Samsung Electronics, "S3C2440A 32-BIT CMOS MICROCONTROLLER USER'S MANUAL Revision 1", Samsung Electronics, 2004.

[2] 홍건표, 하동호, "ARM920T S3C2440A를 이용한 개발자들을 위한 ARM 프로세서", OHM, 2006.

[3] ROBERT SEDGEWICK, "Algorithms in C++", Addison Wesley, 1992.

[4] 이준호, 박영진, 박윤식, "보 보강재 배치 최적화 문제에서의 기하구속조건 처리기법 ", 한국소음진동공학회 춘계학술발표대회 논문집, 단일호, pp. 870-875, 2004.

[5] John Catsoulis, "Designing Embedded Hardware, 2nd ED의 역서", 한빛미디어, 2007.

[6] Samsung Electronics, "Product Information LTV350QV-F04", Samsung Electronics, 2005.

[7] Donald Hearn, M.Pauline Baker, “COMPUTER GRAPHICS의 역서”, 아진, 2002.

[8] 최윤철, 임순범, "컴퓨터 그래픽스 배움터(개정판)", 생능, 2006.

오 삼 권(Sam-Kweon Oh) **[종신회원]**

- 1994년 5월 : Queen's University (Canada)(컴퓨터과학박사)
- 1995년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 교수

<관심분야>
Embedded Real-Time Systems, OS, Communication Protocol, Fault Tolerance

박 근 덕(Geun-Duk Park) **[정회원]**

- 2005년 8월 : 서울대학교 전기컴퓨터공학부 (공학박사)
- 2006년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 조교수

<관심분야>
임베디드소프트웨어공학, 서비스 지향 컴퓨팅, XML 응용

김 병 국(Byoung-Kuk Kim) **[준회원]**

- 2009년 3월 ~ 현재 : 호서대학교 일반대학원 컴퓨터공학과 (컴퓨터공학 석사과정)

<관심분야>
Embedded System, Real Time Operating System