

<http://dx.doi.org/10.7236/JIIBC.2013.13.4.149>

JIIBC 2013-4-20

스마트 CamRuler 계측 알고리즘 설계

Design of Measurement Algorithms in the Smart CamRuler

오선진*

Sun-Jin Oh

요약 최근 스마트 폰 기술의 급속한 성장과 더불어, 이를 위한 다양한 어플들의 개발과 보급이 활발하게 이루어지고 있다. 특히 스마트 기기의 카메라 모듈을 이용한 흥미로운 어플들이 지속적으로 개발되고 있어 우리들은 다양한 모바일 서비스를 사용할 수 있게 되었다. 본 논문에서는 스마트 기기의 카메라 모듈을 이용하여 촬영되는 피사체의 실물 크기를 정확하게 측정할 수 있는 어플에 사용되는 계측 알고리즘을 설계하고 구현하였다. 스마트 폰의 카메라 모듈로 피사체 이미지를 촬영할 때 3축 자이로 가속 센서를 이용하여 피사체간 거리와 기울기, 회전각을 실시간으로 측정하여 촬영 이미지 내의 피사체의 사이즈를 정확하게 계측하도록 하였다. 제안한 계측 알고리즘은 모의실험을 통해 그 계측 정확도를 비교 분석하였다.

Abstract With a rapid growth of smartphone technologies, various applications are developed and diffused actively nowadays. Especially, interesting applications using camera module in a smartphone are developed continuously, mobile users are able to use various useful mobile services in humdrum life. In this paper, we design and implement measurement algorithms which precisely measure the object taken by the camera module in a smartphone. We use 3-axis gyro accelerometer sensor in a smartphone to get the distance, incline and rotation angle in a real time when we take a picture of shooting object and can obtain precise size of it in the picture image. The measurement algorithms proposed in this paper are analyzed and evaluated by a simulation study.

Key Words : Smart CamRuler, Android, Smartphone App

1. 서론

스마트 폰의 대중화에 힘입어 폰에 간단히 설치하여 사용할 수 있는 다양한 어플들이 개발되어 빠르게 보급되고 있으며 이러한 어플들은 단순한 기능을 넘어 실생활에서 아주 유용하게 사용될 수 있는 광범위한 분야로 지속적으로 확대 발전되고 있다. 스마트 폰 어플들의 주요 응용 분야는 소셜 네트워크, 교육, 건강, 레저와 스포

츠, 유틸리티, 취미생활과 오락 등 매우 다양한 분야로 발전하고 있으며 앞으로도 그 발전 가능성은 무궁무진하다고 할 수 있다.^[1] 특히 개인의 취향이나 개성이 중시되는 최근의 사회 풍조에 발맞춰 개인의 라이프스타일이나 취향에 맞는 독특한 맞춤형 어플들이 많은 인기를 얻고 있고, 주로 실생활의 편의성과 효율성을 향상시키는데 도움을 주는 어플들이 주류를 이루고 있다.^[2] 아울러, 이런 요구를 수용할 수 있는 다양한 맞춤형 어플들이 용이하

*종신회원, 세명대학교 정보통신학부
접수일자 : 2013년 5월 21일, 수정완료 : 2013년 7월 13일
게재확정일자 : 2013년 8월 16일

Received: 21 May, 2013 / Revised: 13 July, 2013 /

Accepted: 16 August, 2013

**Corresponding Author: sjoh@semyung.ac.kr

Dept. of Computer & Information Science, Semyung University, Korea

게 개발될 수 있도록 인터페이스 역할을 할 수 있는 미들웨어에 관한 연구와 개발 또한 본격화 되면서 비록 스마트 폰 응용 소프트웨어 개발 플랫폼 환경이나 사양이 다양하더라도 최소의 수정과 기능 설정만으로 쉽게 개발하여 사용할 수 있는 미들웨어의 등장과 활용이 최근 활발하게 이루어지고 있다.^[3]

본 논문에서는 안드로이드 기반 스마트 기기의 카메라 모듈을 이용하여 촬영되는 대상 피사체의 실물 크기를 정확하게 측정할 수 있는 어플에 사용되는 계측 알고리즘을 설계하고 구현하였다. 피사체를 촬영하는 스마트 폰의 기종에 따라 크기 계측을 하는 대상 피사체의 결과 이미지에 다양하게 영향을 미치는 특성이 있어 계측에 어려움을 주게 되는데, 이 문제를 극복하고 계측 결과의 정확도와 신뢰성을 높이기 위해 0점 조절모듈, 높이 및 폭 측정모듈, 지정거리 측정모듈, 그리고 일반거리 측정모듈 등 4가지로 설계하였고, 구현한 알고리즘은 모의실험을 통해 계측 정확도를 비교 분석하였다. 모의실험 결과 본 논문에서 제안한 피사체 계측 알고리즘은 사용 기종이나 특징에 무관하게 97% 이상의 정확도를 나타냈으며, 특히 피사체와 동일 평면에서 기준자를 이용한 피사체 계측을 하는 지정거리와 일반거리 측정 모듈에서는 사용 스마트 폰의 기종에 상관없이 99%이상의 정확도를 보였으며 다만 피사체 촬영 시 촬영 자세가 계측결과에 영향을 미치는 것으로 판명되었다.

본 논문의 구성은 다음과 같다. 2장에서는 시판 중인 스마트 폰 카메라 모듈의 특성 관련 연구를 알아보고, 3장에서는 스마트 Camruler의 시스템 모델을 정의하였으며, 4장에서는 본 논문에서 제안한 스마트 Camruler의 피사체 계측 알고리즘을 서술하였다. 5장에서는 구현 결과와 주요 분석 결과를 고찰하였으며, 마지막으로, 6장에서 향후 연구과제와 함께 결론을 맺는다.

II. 관련 연구

스마트 CamRuler 앱을 이용한 정확한 피사체 계측을 위해서는 우선 시판 중인 스마트 폰들의 카메라 모듈 특성을 정확히 분석하고 적용할 수 있는 특성연구가 절실히 요구된다. 즉 스마트 폰 카메라 모듈의 기종과 시스템 버전에 따른 피사체 계측 결과에 영향을 미칠 수 있는 주요 요소로는 첫째로, 스마트 기기마다 장착된 카메라 모듈의 해상도가 상이하므로 촬영된 이미지를 구성하는 화

소 수가 다르고 위치를 인식하는 좌표 값 역시 매우 달라진다. 둘째로, 촬영 시 피사체와의 거리, 방향, 촬영 각도, 높이에 따라 이미지의 형태나 모양이 상이하므로 이로 인해 이미지 내 피사체의 모양이나 크기 또한 크게 달라진다. 셋째로, 촬영 시 조명의 위치, 밝기, 개수에 따라 피사체의 외곽선의 위치, 경계가 불분명하게 나타나는 문제를 가지고 있다. 마지막으로 터치스크린을 기반으로 하는 스마트 기기의 액정화면에 보이는 화상 이미지는 손가락 제스처에 의해 자유롭게 이동, 축소 및 확대가 가능하므로 계측하고자 하는 피사체의 크기가 고정되지 않아 이들의 길이 계측에 어려움이 있다. 따라서 이런 문제를 극복할 수 있는 방안이 마련되어야 정밀한 피사체 크기 계측이 가능하다.^[4, 5] 최근 이와 관련된 연구 결과를 보면, 시판되는 스마트 폰 카메라 모듈들의 특성을 알아보기 위해 피사체를 촬영하기 전과 후의 촬영 이미지 간 차이를 분석한 결과, 거의 모든 기종에서 촬영 전 액정에 표시되는 피사체의 이미지가 촬영 후 결과 이미지와 매우 차이가 남을 알 수 있었다. 이들에서는 대부분 촬영 전 이미지의 세로 길이가 가로 길이에 비해 짧게 왜곡되어 전체적으로 위에서 아래로 눌린듯한 이미지를 보였다. 그러나 촬영 후 결과 이미지에서는 이러한 현상이 모든 기종에서 나타나지 않고 가로와 세로의 비율이 비교적 정확하게 나타남을 알 수 있었다. 이는 촬영 전에 액정을 통해서 나타난 이미지는 안드로이드 이미지 최적화 과정을 거친 촬영 후 결과 이미지에서와는 전혀 다르게 보여짐을 알 수 있다.^[4, 5] 두 번째 카메라 특성 연구에서는 스마트 폰 기종 별 같은 거리와 높이에서 동일한 피사체를 촬영하였을 때의 카메라 모듈의 기울기 각도와 회전각도가 결과 영상 이미지의 사이즈에 미치는 영향을 분석한 결과, 같은 크기의 피사체를 촬영 기울기와 회전 각도를 달리하여 촬영했을 때 결과 이미지는 그 각도에 따라 크게 왜곡되어 나타남을 알 수 있고, 이로 인해 정확한 피사체의 크기 계측에 큰 영향을 미치는 것을 알 수 있었다.^[5] 이어서 촬영 이미지 내의 특정 피사체를 계측할 때 터치스크린을 통해 손가락제스처로 계측 시 오차 정도를 관찰한 결과, 손가락을 사용하여 계측할 때 액정의 크기가 작아 피사체의 정확한 포인팅이 쉽지 않아 스타일러스 펜을 이용했을 때와 오차 범위 내에서 차이를 보였으며 이를 보정 할 수 있는 방안이 필요하다고 판단되었다.^[5, 6]

스마트 폰 카메라 모듈로 촬영한 피사체 크기에 미치는 영향에 관한 특성 연구 결과를 종합해 보면, 우선 겐

력시 기종을 제외한 거의 대부분의 스마트 폰 기종에서 촬영 전과 후의 피사체의 크기 변화가 있음을 알게 되었고 또한, 피사체를 촬영하는 기종에 따라 결과 이미지에 서의 피사체 크기를 측정할 좌표 값이 매우 상이하다는 것을 알았다.^[7] 따라서 피사체 크기 계측을 위해서는 사전에 0점 조절이 필요하며, 촬영 기율기와 회전각도 역시 촬영 결과 이미지의 크기에 직접적으로 영향을 주는 것으로 나타났으며 이를 한꺼번에 보정 할 수 있는 함수식 도출이 매우 어렵다고 결론을 내렸다. 따라서 피사체 촬영은 기율기와 회전 각도를 고려하여 촬영을 해야 정밀한 계측결과를 기대할 수 있다. 촬영거리에 대한 피사체의 결과 이미지 계측 결과는 규칙적이고 이에 대한 함수식 도출이 가능하여 피사체간의 거리 계측 역시 가능함을 알 수 있었다.^[7]

III. 스마트 CamRuler 시스템 모델

스마트 폰의 급속한 발전과 보급으로 많은 사용자들이 스마트 폰 관련 응용에 많은 관심과 참여가 이루어지고 있다. 본 논문에서 제안하는 스마트 CamRuler 앱의 시스템 모델은 모바일 컴퓨팅 환경에서 스마트 폰이나 태블릿 PC에 내장된 카메라 모듈과 정전식 멀티 터치 가능한 터치스크린을 기반으로, 스마트 기기의 컴퓨팅 기능을 이용한 어플이다. 이 앱은 구글의 안드로이드 OS를 기반으로 하는 스마트 기기에 내장된 카메라 모듈을 통해 촬영된 화상 이미지 안에 특정 피사체의 크기와 피사체간 거리를 계측하는 어플로 다양한 분야에서 보조 계측장비로 사용될 수 있다.

스마트 CamRuler 앱 모델의 주요 동작원리는 스마트 폰에 내장되어 있는 카메라 모듈과 3축 자이로 가속도 센서를 이용하여 촬영 피사체까지의 거리와 피사체의 사이즈를 측정하는 계측 앱이며, 이때 가속 센서를 이용하여 피사체까지의 기율기와 회전 각도를 측정하고 측정하는 스마트 폰의 높이 등을 사용하여 거리와 크기를 측정하게 된다. 이때 사용하는 3축 가속도 센서의 좌표 시스템과 동작원리는 다음과 같다.

그림 1은 스마트 폰에 내장된 3축 자이로 가속도 센서의 좌표계를 보여준다. 그림에 보인바와 같이, 센서는 X, Y, Z 의 좌표에 대해서 각각에 해당하는 값을 배열로 받아와서 처리할 수 있게 되어있다. 이때 각 좌표축의 정의는 다음과 같다.^[8]

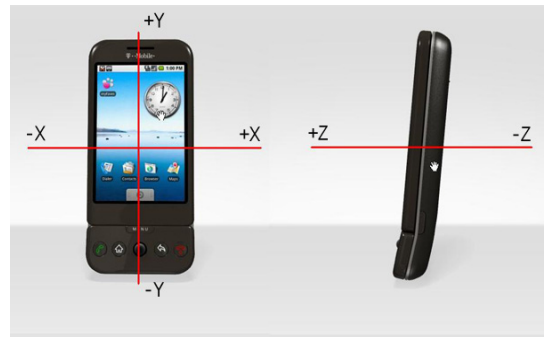


그림 1. 3축 가속도 센서의 좌표계
Fig. 1. Coordinates of 3-axis Accelerometer Sensor

- X축 : 화면에 수평축(portrait 모드에서 짧은 에지를 landscape 모드에서 긴 에지)을 나타내며 오른쪽을 가리킨다.
- Y축 : 화면에 수직축을 나타내며 화면 위쪽을 가리킨다. (원점은 왼쪽 밑 코너)
- Z축 : 단말이 화면을 위로해서 테이블에 올려져 있다고 생각했을 때 하늘을 가리킨다.

여기서 SENSOR_ORIENTATION (방향 센서)의 각 배열의 값은 각도를 나타낸다. 즉, 단말기 화면이 하늘을 향한 상태로 테이블 위에 수평으로 놓여있는 상태를 기준으로 각 축을 중심으로 회전시키는 것을 생각하면 된다. 따라서,

- values[0] : Z 축을 중심으로 회전 ($0 \leq azimuth < 360$)
0 = 북, 90 = 동, 180 = 남, 270 = 서
- values[1] : X축을 중심으로 회전 ($-180 \leq pitch \leq 180$)
Z축이 Y축 방향으로 향하면 0 보다 큰 값
화면이 하늘을 향하고 테이블 위에 수평으로 놓여있는 상태 0, 화면이 아래를 향하면 -180 or 180, 똑바로 세우면 -90, 거꾸로 세우면 +90
- values[2] : Y축을 중심으로 회전 ($-90 \leq roll \leq 90$)
Z축이 X축 방향으로 향하면 0보다 큰 값

가속도 센서는 압력센서와 더불어 반도체 공정 기술과 마이크로 머시닝 기술의 발달에 힘입어 상용화가 빠르게 진행되는 분야로 공장 자동화 및 계측기기, 자동차, 스마트 폰과 가전제품 등의 응용 분야로 점차 확대되고 있다.^[8]

IV. 스마트 CamRuler의 계측 알고리즘

본 논문에서 제안하는 스마트 폰 CamRuler의 피사체 계측 알고리즘은 그 측정 대상과 방법에 따라 거리 측정 모듈, 높이·폭 측정모듈, 지정거리 측정모듈, 그리고 일 반거리 측정모듈 등 크게 네 가지로 나눌 수 있다. 우선 스마트 폰의 카메라 모듈을 이용한 피사체 사이즈 측정에 앞서 스마트 폰 기종마다 제각각 가지고 있는 카메라 모듈의 특성으로 인한 오차를 줄이기 위한 0점 조절 모듈을 이용하게 되는데, 이 모듈에서는 현재 사용자가 스마트 폰을 들고 촬영을 할 때의 스마트 폰의 지상으로부터의 정확한 높이를 입력하게 하고, 피사체와의 약속된 거리(즉, 일반적으로 2m 또는 3m 정도)를 측정하고, 피사체의 사이즈를 측정하기 위한 기준자가 되는 길이가 알려진 객체(예를 들어, 30cm 자 등)를 알려진 거리에서 촬영한 후 그 기준자의 액정 화면상 좌표 값을 이에 대응하는 길이 값으로 환산하는 알고리즘을 실행하게 된다. 그림 2는 본 논문에서 제안하는 0점 조절 모듈의 주요 알고리즘을 보여준다.

```

<0점조절 알고리즘>
input height  $h$  of CamRuler taking shooting object;
// 피사체 촬영 시 스마트 폰 카메라 높이  $h$  입력
fulfilling distance measurement algorithm;[7]
// 피사체 간 거리 측정 알고리즘 실행
measure incline angle  $\theta$  using Accelerometer
// 스마트 폰 가속 센서를 이용 기울기 각도 측정
get the distance  $d$ 


$$d = \frac{h}{\tan(\text{RAD}(90 - \theta))}$$
 (1)

let  $d$  be a determined value (2 or 3 m)
// 거리가 약속된 거리가 되도록 앞뒤로 이동
take a picture of a standard ruler
// 길이가 알려진 기준자를 촬영
get 2 coordinates of a standard ruler
// 액정 화면에서 이 기준자의 양쪽 두 좌표 구함
input real size of a standard ruler
// 기준자의 실제 길이를 입력
calculate a real size per unit coordinate
// 단위 좌표 당 실제 길이  $U$ 를 계산하여 저장
    
```

그림 2. 0점 조절 알고리즘
Fig. 2. Zero Point Adjustment Algorithm

피사체의 높이와 폭을 측정하는 모듈은 촬영 대상이 되는 피사체가 수직 또는 수평으로 배치되어 있을 때 측

정하는 알고리즘으로 우선 높이를 측정하는 모듈에서는 피사체간 거리를 거리 측정 알고리즘을 이용하여 구한다 음, 피사체의 꼭대기를 향해 CamRuler를 지향하고 그 기울기 각도 θ_2 를 구하여 정확한 피사체의 높이를 구하게 된다.^[7] 한편, 피사체의 폭을 측정하는 알고리즘에서는 우선 피사체간 거리를 거리 측정 알고리즘을 이용하여 구한다음, 피사체의 왼쪽 끝과 오른쪽 끝을 각각 CamRuler를 지향하고 그 회전각도 θ' 와 θ'' 를 구하여 정확한 피사체의 폭을 구할 수 있다.^[7] 그림 3은 본 논문에서 제안하는 피사체의 높이와 폭을 구하는 모듈의 주요 알고리즘을 보여준다.

```

<높이 폭 측정 알고리즘>
fulfilling distance measurement algorithm;[7]
// 피사체 간 거리 측정 알고리즘 실행
measure incline angle  $\theta$  using Accelerometer
// 스마트 폰 가속 센서를 이용 기울기 각도 측정
get the distance  $d$ 


$$d = \frac{h}{\tan(\text{RAD}(90 - \theta))}$$


get incline angle  $\theta_2$  of the highest point of object
// 피사체의 꼭대기 점까지의 기울기 각도 측정
calculate object's height  $H$  // 피사체의 높이 계산

$$H = h + d \times \tan(\text{RAD}(\theta_2 - 90))$$
 (2)

get rotation angle  $\theta'$  and  $\theta''$  from left & right end
// 피사체의 왼쪽과 오른쪽 끝을 향해 회전각 측정
calculate object's width  $W$  // 피사체의 폭 계산

$$W = d \times \tan(\text{RAD}(360 - \theta')) + d \times \tan(\text{RAD}(\theta''))$$
 (3)
    
```

그림 3. 높이와 폭 측정 알고리즘
Fig. 3. Height & Width Measurement Algorithm

지정거리 측정 모듈은 일정하게 정해진 거리(2 또는 3m)에서 촬영한 피사체의 크기를 측정하는 모듈로 0점 조절 모듈에서 이미 계산해 놓은 단위 좌표 당 환산한 실제 길이를 이용하게 된다. 우선 CamRuler를 피사체로부터 정해진 거리까지 이동하면서 거리 측정 알고리즘을 이용하여 정확한 거리 위치에 서서, 길이 측정 대상의 피사체를 촬영한 후, 그 피사체의 측정하고자 하는 부분의 양쪽 끝단의 두 점에 대한 좌표를 찍어 이들 점간의 액정 상 좌표 값을 측정한다. 이어서 이 좌표에 대한 단위 좌표 당 환산된 실제 길이 U 를 곱해서 이 피사체의 실제 길이를 구할 수 있다. 이와 같은 방법을 사용하면 스마트

폰의 기종에 따른 액정 화면의 특징이나 오차에 대한 문제를 고려하지 않아도 피사체의 정확한 사이즈의 측정이 가능하게 된다. 그림 4는 본 논문에서 제안한 지정거리 측정 모듈의 주요 알고리즘을 보여준다.

```

<지정거리 측정 알고리즘>
position at determined distance using a distance
measurement algorithm;
// 거리측정 알고리즘 사용 정해진 거리의 촬영지점 위치
take a picture of a shooting object;
// 촬영 대상 피사체를 촬영한다.
get coordinates of each both end point of the object;
// 대상 피사체의 양단의 좌표를 점을 찍어 지정한다.
calculate the size of object  $T$  on a color LCD screen;

$$T = \sqrt[2]{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

// 액정 화면상의 피사체의 크기를 좌표값 사용하여 계산
calculate the real size  $S$  of a object using real size per unit
coordinate;

$$S = U \times \sqrt[2]{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4)$$

// 단위 좌표 당 실제 길이  $U$ 를 액정화면 상 피사체 크기
에 곱하여 실제 피사체의 크기를 계산
    
```

그림 4. 지정거리 측정 알고리즘
Fig. 4. Predetermined Distance Measurement Algorithm

```

<일반거리 측정 알고리즘>
take a picture of a shooting object with a standard ruler;
// 촬영 대상 피사체를 기준자와 함께 촬영한다.
get coordinates of each both end point of the STD ruler;
// 기준자의 양단의 좌표를 점을 찍어 지정한다.
calculate the size of STD ruler  $T$  on a color LCD screen;

$$T = \sqrt[2]{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

// 액정 화면상의 기준자의 크기를 좌표값 사용하여 계산
input real size of a standard ruler
// 피사체 실제 길이를 입력
calculate a real size per unit coordinate
// 단위 좌표 당 실제 길이  $U$ 를 계산하여 저장
get coordinates of each both end point of the object;
// 대상 피사체의 양단의 좌표를 점을 찍어 지정한다.
calculate the size of object  $T'$  on a color LCD screen;

$$T' = \sqrt[2]{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

// 액정 화면상의 피사체의 크기를 좌표값 사용하여 계산
calculate the real size  $S$  of a object using real size per unit
coordinate  $U$ ;

$$S = U \times T' = U \sqrt[2]{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5)$$

// 단위 좌표 당 실제 길이  $U$ 를 액정화면 상 피사체 크기
에 곱하여 실제 피사체의 크기를 계산
    
```

그림 5. 일반거리 측정 알고리즘
Fig. 5. General Distance Measurement Algorithm

마지막으로 일반거리 측정 모듈은 지금까지 피사체의 정확한 사이즈 측정에 장애요인이 되었던 스마트 폰 카메라의 물리적인 특징, 촬영 영상을 보여주는 액정 화면의 해상도나 기종에 따른 오차 발생 등으로 인한 문제점들을 모두 해소할 수 있는 계측 알고리즘으로 어떠한 기종의 스마트 폰 카메라나 액정 화면을 사용하던지 그 기종이나 물리적인 특징과는 상관없이 정확한 피사체의 크기를 측정할 수 있다. 이를 위해 피사체 촬영 전 한 가지 요구사항이 있는데 피사체를 촬영할 때 사이즈의 기준이 될 수 있는 기준자(base)를 피사체와 동일 거리에서 동시에 촬영해야 하는 것이다. 이전 과정과는 달리 이 모듈에서는 촬영 시 피사체간 거리는 고려하지 않고 기준자와 동일 평면상에서 촬영하여 우선 기준자의 양단간의 좌표를 액정 화면상에서 선택하여 좌표 값을 구하고, 이를 통해 액정 화면상 좌표 거리를 계산한다. 이어서 기준자의 실제 길이를 입력받아 단위 좌표 당 실제 길이를 환산한다. 대상 피사체의 실제 길이를 구하기 위해 측정 피사체의 양단간의 좌표를 액정 화면상에서 선택하여 좌표를 구하고, 이를 통해 액정 화면상 대상 피사체의 좌표 거리를 계산한다. 대상 피사체의 실제 길이는 계산된 좌표 거리에 단위 좌표 당 실제 길이를 곱하여 구해진다. 다음의 그림 5는 본 논문에서 제안한 일반거리 측정모듈의 주요 알고리즘을 보여준다.

V. 구현 결과 및 고찰

이 장에서는 본 논문에서 제안한 안드로이드 기반 스마트 폰 카메라 모듈을 통해 촬영된 피사체 사이즈를 측정하는 알고리즘의 구현결과를 보여주고, 이를 모의실험을 통해 국내에서 시판되는 주요 스마트 폰 기종을 대상으로 계측 정확도를 분석·고찰한 결과를 보여준다. 제안한 알고리즘의 구현은 안드로이드의 기본 개발 언어인 자바의 JDK를 자바 실행 환경인 Eclipse를 기반으로, 여기에 ADT 프러그 인을 설치하고, 안드로이드 응용 개발을 위한 Android SDK 플랫폼 상에서 진행하였으며, 구현 결과를 안드로이드 폰 에뮬레이터인 AVD를 통해 테스트 하였다. 그림 6은 본 논문에서 제안한 0점 조절 모듈의 구현 결과 일부를 보여준다. 여기서 스마트 폰 카메라의 촬영 높이를 입력 받고, 피사체까지의 거리를 측정하며 액정 단위 좌표 당 실제 길이를 계산한다.

```

Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        // 제한시간을 알기 쉽게 하기 위하여 프로그래스바 사용
        // 1초간격으로 31초에서 1초까지 카운트
        mHandler.sendEmptyMessageDelayed(0, 1000);
        // 시간 감소
        timervalue++;
        if (timervalue == 10) {
            // *****영점 조절 높이 가져옴
            // 발표
            AlertDialog.Builder alertDlg = new AlertDialog.Builder(
                zeromeasure.this);
            alertDlg.setTitle("현재 스마트폰 높이 입력하세요.");
            alertDlg.setView(linear);
            // TODO Auto-generated method stub
            -중략-
        }
        // 실제적인 센서값들을 받아옴
        public void onSensorChanged(SensorEvent event) {
            // TODO Auto-generated method stub
            float[] v = event.values;
            // 최종적으로 출력되는 거리값
            double max;
            String strNumber = String.format("%.2f", v[1]);
            String strNumber2 = String.format("%.2f", v[2]);
            // 타일이 맞을시 값을 출력
            switch (event.sensor.getType()) {
                // 방향센서일시 동작
                case Sensor.TYPE_ORIENTATION:
                    // 거리구하는 공식
                    max = maxheight / (Math.tan(90 * Math.PI
                    / 180 - Math.abs(v[1]) * Math.PI / 180));
                    String strNumber3 = String.format("%.1f", max);
                    measure.setText("거리 : " + strNumber3);
                    mTxtOrient.setText(" pitch(기울기) : " + strNumber
                    + " , roll(회전) : " + strNumber2);
                    -중략-
            }
        }
    }
}
    
```

그림 6. 0점 조절 알고리즘 구현결과
 Fig. 6. Implementation Result of a Zero Point Adjustment Algorithm.

```

AlertDialog.Builder alertDlg = new AlertDialog.Builder(
    zeromeasure_result.this);
alertDlg.setTitle("두 점 사이의 cm를 입력하세요.");
alertDlg.setView(linear);
alertDlg.setPositiveButton("확인",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            name = edt1.getText().toString();
            if (edt1.length() != 0) {
                if (value.equals("")) {
                    cv.put("point", name);
                    cv.put("pitanum", pitanum1);
                    db.insert("cameradatabasetable", null, cv);
                    db.close();
                    finish();
                }
                if (value.length() != 0) {
                    cv.put("point", name);
                    cv.put("pitanum", pitanum1);
                    db.update("cameradatabasetable", cv, null, null);
                    db.close();
                    finish();
                }
            } else {
                XYcount = 0;
                count = 0;
                Toast.makeText(getBaseContext(),
                    "다시 점을 찍고 값을 입력하여 주세요.",
                    Toast.LENGTH_SHORT).show();
            }
        }
    });
    
```

그림 7. 일반거리 측정 알고리즘 구현 결과
 Fig. 7. Implementation Result of the General Distance Measurement Algorithm

그림 7은 일반거리 측정 알고리즘을 구현한 결과의 일부분을 보여준다. 그림에 보인바와 같이, 촬영한 이미지의 사이즈를 측정하기 위해 기준자가 되는 대상 피사체의 양단에 터치스크린을 통해 두 점을 찍어 좌표 값을 추출하고, 그 기준자의 실제 길이를 입력하여 액정 화면의 좌표 단위 당 실제 길이를 계산하는 알고리즘의 일부를 보여 준다. 이때 터치스크린 상의 정확한 양단의 좌표를 정확히 구하도록 방향키를 이용하여 목표 지점으로 정확하게 이동시킬 수 있도록 하였다.

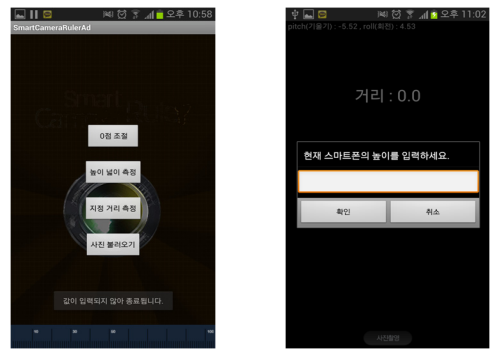


그림 8. 스마트 CamRuler 실행 결과
 Fig. 8. Execution Results of the Smart CamRuler.

그림 8은 본 논문에서 제안한 스마트 Camruler를 실행한 결과의 일부를 보여준다, 그림에 보인바와 같이 왼쪽의 실행 화면은 처음 스마트 Camruler 앱이 실행되어 보여지는 메뉴 화면으로 4가지의 측정 모듈 중에서 하나를 선택할 수 있도록 하였고, 오른쪽의 그림은 0점 조절 모듈이 실행되고 있는 가운데 피사체의 촬영 시 스마트폰의 촬영 높이를 입력하도록 하는 실행 화면을 보여 준다.

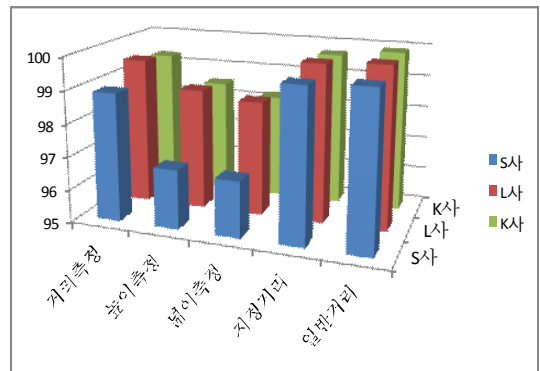


그림 9. 모의실험 결과
 Fig. 9. Results of the Simulation Study.

본 논문에서 제안하는 스마트 폰의 카메라 모듈을 이용한 대상 피사체 크기 계측 알고리즘의 사이즈 계측 정확도를 알아보기 위해 국내에서 시판되고 있는 국내 3사의 안드로이드 기반 주요 스마트 폰 기종을 대상으로 피사체 간 거리 측정, 높이 측정, 폭 측정, 지정거리 측정, 그리고 일반거리 측정 모듈에 대한 모의실험을 실시하여 그 계측 정확도를 알아보았다. 거리는 1m, 2m, 3m의 거리를 각각 10회씩 반복하여 측정하였고, 높이와 폭 그리고 지정거리 측정은 1m, 2m, 3m 거리에서 각각 10회씩 측정하여 정확도를 비교하였으며, 일반거리 측정 모듈에서는 임의의 거리에서 30cm의 기준자를 사용하여 각각의 기종에 대해 총 10회씩 측정하여 그 계측 정확도를 구하였다. 그림 9는 이러한 각 모듈에 대한 기종별 계측 정확도를 그림으로 보여준다. 그림에서 보인바와 같이 전 기종에 걸쳐 피사체 계측에 대한 정확도는 97% 이상으로 매우 높게 나타났으며, 기종별로는 S사제품 보다는 L사와 K사 기종에서 99%가 넘는 정확도를 나타내고 있음을 알 수 있다. 아울러, 각 모듈별로 계측 정확도를 비교해 보면 상대적으로 지정거리와 일반거리 측정 모듈이 다른 기타 모듈에 비해 그 정확도가 한층 높게 나타나고 있음을 알 수 있었다.

VI. 결 론

최근 스마트 폰 어플의 급속한 개발과 보급으로 일상 생활에서 많은 유용한 모바일 서비스를 이용하는 것이 가능하게 되었고, 흥미로운 응용들이 속속 개발되고 있다. 본 논문에서 우리는 안드로이드 기반 스마트 폰의 카메라 모듈을 이용하여 피사체의 크기를 측정하는 알고리즘을 설계하고 구현하였다. 피사체 사이즈 측정 알고리즘은 촬영하는 스마트 폰의 기종에 따른 특성에 따라 다양한 결과 이미지에 영향을 미치는 특징이 있어 계측에 어려움을 주게 되는데, 이 문제를 극복하고 계측 결과의 정확도와 신뢰성을 높이기 위해 0점 조절모듈, 높이와 폭 측정모듈, 지정거리 측정모듈, 그리고 일반 거리 측정모듈 등 4가지로 설계하였으며, 구현한 알고리즘은 모의실험을 통해 그 정확도를 비교 분석하였다.

모의실험 결과 본 논문에서 제안한 피사체 계측 알고리즘은 사용 기종이나 특성에 무관하게 97% 이상의 계측 정확도를 나타냈으며, 특히 피사체와 동일 평면에서

기준자를 이용한 피사체 계측을 하는 지정거리 모듈과 일반거리 측정 알고리즘에서는 사용한 스마트 폰의 기종에 상관없이 99% 이상의 정확도를 보였다. 다만 피사체 촬영 시 촬영 자세가 계측 결과에 영향을 미치는 것으로 판명되었다. 향후 연구 과제는 피사체 계측 결과의 정확도를 높일 수 있도록 촬영 자세의 영향을 최소화 하는 방안과 터치스크린에서의 정확한 포인팅 방법 등의 알고리즘 개발에 관한 것이다.

References

- [1] John, A., Adamic, L., Davis, M., Nack, F., Shamma, D. A., and Seligmann, D. D., "The future of online social interactions: what to expect in 2020", Proceedings of the 17th International Conference on WWW, 2008.
- [2] D. K. Lee, J. H. Kwon, "Social Search Algorithm considering Recent Interests of User", Journal of Korean Institute of Information Technology, vol. 9, issue 4, pp. 187-194, Apr 2011.
- [3] Oh, S., "Design of a Middleware for Android-based Smartphone Applications", Journal of the Institute of Webcasting, Internet and Telecommunication, Vol., 12, No. 2, pp. 111 - pp. 118, 2012.
- [4] Alan C. Bovik, Essential Guide to Image Processing, Elsevier, 2009.
- [5] Oh, S., "A Study on Characteristics of Smart Phone Camera Module for Measuring a Shooting Object", The Journal of the Institute of Webcasting, Internet and Telecommunication, Vol. 12, No. 5, pp. 99 - pp. 105, October 2012.
- [6] Kim, H., Lee, Y., "Haptic Modeling Method based on Multi-layered Images", Journal of Korean Institute of Information Technology, Vol. 8, No. 7, 163-169, 2010.
- [7] <https://play.google.com/store/apps/details?id=kr.sira.measure>
- [8] <http://www.androidpub.com/1305>

저자 소개

오 선 진(중신회원)



- 제 6권 제2호 참조
- 현재 : 세명대학교 정보통신학부 교수
<주관심분야 : 스마트 응용, 그린IT, MANETs, 모바일컴퓨팅, USN 등>