
고도를 고려한 정밀도 높은 운동거리 측정시스템

김대호* · 정인범**

The exercise-distance measuring system with high precision considering of altitude

Daeho Kim* · Inbum Jung**

본 연구는 지식경제부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음(NIPA-2012-H0401-12-1002)

요 약

운동자들의 운동 정보를 측정하기 위하여 스마트 폰의 운동 애플리케이션들은 GPS장치에서 얻은 정보들을 사용한다. 이러한 애플리케이션에서 사용자에게 제공하는 운동정보는 뛰거나 걸은 거리, 시간, 칼로리 소모량, 평균 속도 등이다. 이들 중 운동한 거리는 모든 정보의 바탕이 되므로 정확한 측정이 필요하다. 그러나 현재 사용되는 거리 측정법은 지구를 구 또는 타원으로 가정하여 계산하기 때문에 실제 운동거리와는 오차가 생기게 된다. 실제로 지구의 표면은 경사로 이루어져 있기 때문이다. 본 논문에서는 이 오차를 보정하여 지형의 경사도를 반영한 새로운 거리측정 알고리즘을 제안한다. 제한된 알고리즘의 성능을 평가하기 위하여 스마트 폰을 기반으로 한 운동 라이프가이드 시스템을 구축하였다. 실험을 통하여 제안된 알고리즘이 보다 정확한 운동 거리 측정 정보를 제공함을 보인다.

ABSTRACT

To measure the athletic information of exercisers, the applications of smartphone are programmed based on the sensing data from GPS device. These applications provide exercisers for running or walking distance, exercising time, calorie consumption, average speed, and so on. Among them, the exercising distance should measure accurately because it directly affects the other athletic information for exercisers. However, the existing methods for measuring the exercising distance makes errors because they are worked on the simple sphere or ellipse earth models. Actually, the surface of real earth is composed of inclined ground like hills and valleys. In this paper, a new exercising distance measuring algorithm is proposed to compensate the errors of existing method. It considers the altitude of slopes in exercising routes. To evaluate exercising distance measuring algorithms, we implement the athletic life-guide system based on the smartphone platform. In experiments, the proposed method shows that it provides more accurate distance measurement.

키워드

운동 거리, 고도, GPS, 스마트폰 애플리케이션

Key word

Exercising Distance, Altitude, GPS, Smartphone application

* 정회원 : 강원대학교 컴퓨터정보통신공학과 석사과정
(kim_dh@snslab.kangwon.ac.kr)

접수일자 : 2011. 12. 07

심사완료일자 : 2012. 01. 04

** 정회원 : 강원대학교 컴퓨터정보통신공학과 교수 (교신저자)

Open Access <http://dx.doi.org/10.6109/jkiice.2012.16.3.615>

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서 론

최근 사회복지가 발달함에 따라 조깅, 자전거, 농구, 인라인 스케이트 등 다양한 운동을 즐길 수 있는 시설이 많이 늘어났다. 이러한 변화는 사람들의 삶의 질이 향상됨에 따라 건강에 대한 관심이 크게 늘어나고 있기 때문이다. 예전 사회에서는 생계유지에 집중하던 삶이었지만 현재는 소득의 증가와 문화생활의 발전 등으로 인해 점차 건강에 대해 신경을 쓰고 있다. 식생활 변화도 비슷한 맥락으로 풀이된다. 이전에 비해 먹을거리가 풍부해졌으며 바쁘게 돌아가는 사회와 편리함을 추구하는 삶 때문에 패스트푸드와 인스턴트 음식을 쉽게 접할 수 있다.

[그림 1]은 최근 5년간 10대들의 성인병 진단 횟수를 나타낸 그래프이다.[1] 해를 거듭할수록 3대 질병의 진단수가 늘어나는 것을 확인 할 수 있다. 이러한 성인병 발병의 증가는 10대 만의 문제가 아니라 전 연령대에서 나타나고 있는 문제이다. 그러나 성인병은 식습관과 운동 등 생활환경 변화를 통하여 예방 및 치료가 가능하기 때문에 많은 사람들이 꾸준한 운동을 하고 있는 중이다.[2, 3]

운동의 중요성이 늘어남에 따라서 스포츠 관련회사들은 새로운 시장을 개척하기 시작하였고 작은 기구를 만들거나 스마트폰과 연동시켜서 서비스를 제공하는 등 다양한 운동 보조제품들을 만들기 시작했다. 또한, 스마트폰 애플리케이션 제작 회사에서는 스마트폰으로 운동의 도움을 줄 수 있는 다양한 애플리케이션을 앱마켓에 출품하기 시작했다. 특히, 조깅이나 걷기, 자전거 등 유산소 운동관련 보조 시스템들이 각광을 받고 있는 중이다.[4]

유산소 운동관련 보조 시스템의 가장 중요한 요소는 정확한 운동거리 측정이다. 운동거리와 운동시간을 이용하여 사용자에게 각종 정보를 알려주기 때문이다. 그러나 현재 운동 보조 시스템들은 고도정보를 이용하지 않은 거리측정 알고리즘을 사용하므로 실제 운동거리와 오차가 생기게 된다. 본 논문에서는 이 오차를 줄일 수 있는 DMACA(Distance Measuring Algorithm based Considering Altitude)을 제안한다. 또한 이 알고리즘을 사용하는 스마트폰 애플리케이션을 설계 및 구현한다. 이로써 다른 운동 보조 시스템보다 좀 더 운동 정보의 정확성을 높일 수 있다.

II. 관련연구

2.1. 기술 동향

현재 다양한 운동 보조 시스템들은 크게 보조 기구를 사용하는 시스템과 스마트폰 애플리케이션으로 배포되는 방식으로 나눌 수 있다. 보조 기구는 운동용으로 최적화 되어 있기 때문에 소형 기구로 이루어져 있다. 시계만한 크기의 디스플레이 장치로 정보를 제공하거나 스마트폰, mp3 등 주변장치와 연동, 이어폰을 이용하는 방법 등 다양한 방법으로 운동정보를 제공한다.

[그림 2]는 대표적인 운동보조 기구인 Nike 사의 Nike+ 와 Adidas 사의 MiCoach 이다.[5, 6] 이 제품들은 달리기 운동을 대상으로 하며 운동 시 소모 칼로리와 운동 시간, 평균 속도 등의 정보를 제공한다. Nike+ 제품의 특징으로는 실내 환경에서 사용이 가능하다는 점이다. 그렇기 때문에 트레드 밀을 이용하여 운동하는 사람들도 사용할 수 있다.

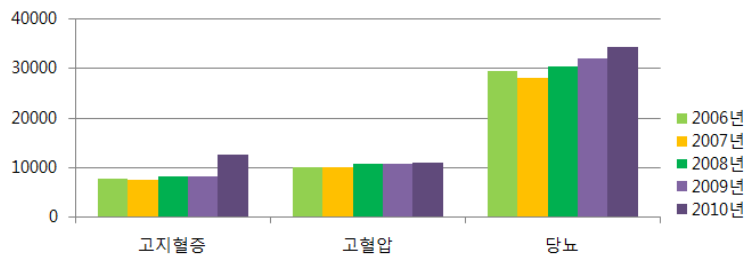


그림 1. 10대 성인병 진단횟수
Fig. 1 Teenage geriatric disease



그림 2. 운동보조 시스템. Nike+(좌) MiCoach(우)
Fig. 2 Exercise support system

마이코치의 특징은 1:1 개인 코칭 시스템을 지원한다. 운동 구간별 스케줄을 미리 정하고 운동 시 이어폰을 통하여 구간정보 및 속도 조절 메시지를 알려준다. 이러한 특징뿐만 아니라 두 제품들은 다른 사용자들과의 경쟁 요소를 집어넣어 사용자의 흥미를 이끌어 내고 있다.

2.2. 운동 시 칼로리 측정방법

2.2.1. METs(Metabolic equivalent)

사람이 운동하는데 소비된 에너지를 간단한 방식으로 나타내는 것을 MET이라 한다. 1MET는 성인 남자가 앉아서 쉬고 있는 상태의 필요한 에너지나 몸에서 필요로 하는 산소의 양($VO_{2rest} = 3.5 \text{ mL/kg/min}$)을 의미한다 [7, 8]. 즉, 1MET란 휴식하고 있을 때 1분 동안 몸무게 1킬로그램 당 3.5mL의 산소가 필요함을 의미한다.

운동 생리학자들은 여러 가지 운동을 할 때 소모되는 산소를 측정했다. 그리고 해당 운동이 휴식상태를 대비하여 얼마나 많은 양의 산소를 사용했는지 계산했다. 힘든 일을 할수록 더 많은 산소가 필요 하며, 따라서 MET가 높아짐을 알 수 있다. 이렇게 계산된 MET를 사용하여 운동을 하는 동안 소모된 칼로리를 측정할 수 있는데 공식은 다음과 같다.

$$\text{에너지 소모량 (Kcal/min)} = 0.0175 \times \text{MET} \times \text{체중 (Kg)}$$

예를 들어 체중 65Kg인 수영선수가 수영을 80분간 했다면 해당 공식에 의하여 약 819칼로리를 사용했다는 것을 알 수 있다.

하지만 MET는 사람에 따라서 그 범위와 내용이 상당히 다를 수도 있기 때문에 아주 정확하게 과학적인 결과를 판정하는 데에는 부적합한 면이 있다. 같은 범위의 MET 안에서, 한 사람의 체질지수, 체력수준 및 수 없이 많은 주변 환경적인 요소에 의해서 에너지 소모가 크게 달라질 수 있다. 따라서 MET는 신체활동에 대한 하나의 지수에 지나지 않으며, 이를 에너지의 단위는 아니라고 생각하면 된다.

2.2.2. 지구성 운동 중 소비열량 계산

지구성 운동 중 소비열량 계산 방법은 사용자가 걷기 또는 달리기 운동을 했을 때 정확한 소모 칼로리를 측정하기 위한 계산식이다[9]. MET의 경우 여러 가지 다양한 운동을 대략적인 방법으로 운동을 할 때 필요한 산소량을 결정 했다. 그러므로 정확한 에너지 소모량을 파악할 수 없었다.

METs 계산법 보다 정밀하게 소모 칼로리를 측정하기 위하여 사용자는 트레드 밀에서 직접 산소발생 측정기를 장착하고 걷기 또는 달리기 운동을 하면서 산소 발생량을 측정했다. 또한, 측정을 할 때 속도, 시간, 트레드 밀의 기울기 등 다양한 요인을 변경하면서 산소 발생량을 측정했고 공식화 했다. 공식은 아래와 같다.

1) 걷기 운동일 경우

$$VO_2 \text{수평적요소 (ml/kg/min)} = 0.1 \times \text{걷는속도 (m/min)}$$

$$VO_2 \text{안정적요소} = 3.5 \text{ (ml/kg/min)}$$

$$\text{총 } VO_2 \text{ (ml/kg/min)} = VO_2 \text{수평적요소} + VO_2 \text{수직적요소} + VO_2 \text{안정적요소}$$

$$VO_2\text{수직적요소}(ml/kg/min) = 1.8 \times \text{걸는속도}(m/min) \times \text{경사도}(\%)$$

2) 달리기 운동일 경우

$$VO_2\text{수평적요소}(ml/kg/min) = 0.2 \times \text{달리는속도}(m/min)$$

$$VO_2\text{수직적요소}(ml/kg/min) = 0.9 \times \text{달리는속도}(m/min) \times \text{경사도}(\%)$$

$$VO_2\text{안정적요소} = 3.5(ml/kg/min)$$

$$\text{총 } VO_2(ml/kg/min) = VO_2\text{수평적요소} + VO_2\text{수직적요소} + VO_2\text{안정적요소}$$

3) 공통부분

$$\text{총 } VO_2(\ell/min) = \frac{\text{총 } VO_2(ml/kg/min) \times \text{체중}}{1000}$$

$$\text{에너지소비량}(Kcal) = VO_2(\ell/min) \times 5.0 \times \text{운동시간}(min)$$

III 고도를 이용한 거리계산 알고리즘

사용자가 운동한 칼로리를 측정할 때 가장 중요한 요소는 운동거리이다. 운동한 거리에 따라서 소모하는 칼로리가 다르기 때문이다. 그렇기 때문에 정확한 운동거리를 알아야지만 더 정확한 칼로리 소모량을 측정할 수 있다. 운동거리를 측정하는 방법은 GPS를 이용하는 방법과 가속도 센서를 사용하여 발걸음수를 측정하고 운동거리를 예측하는 방법 등이 있는데 이 둘의 정확성을 위해 본 논문에서는 GPS를 사용하여 운동거리를 측정할 것이다. 이번 장에서는 GPS를 사용하여 거리계산을 하는 알고리즘을 살펴보고 해당 알고리즘을 사용할 때의 문제점을 파악한 후 본 논문에서 제안하는 DMACA(Distance Measuring Algorithm based on Considering Altitude)에 대하여 설명한다.

3.1. 기존 거리계산 알고리즘 문제점

운동거리를 측정하기 위해서 GPS를 사용한다. GPS는 NMEA(The national Marine Electronics Association) 규격을 통하여 위도, 경도, 시간, 속도, 고도 등 다양한 정보를 제공한다. GPS가 제공하는 정보 중 위도와 경도 데이터를 이용하면 자신의 위치를 알아낼 수 있다. 운동 중에 GPS 기기를 사용하여 위치정보를 주기적으

로 받아오고 이전 위치와 현재 위치를 이용하여 운동거리를 측정한다.

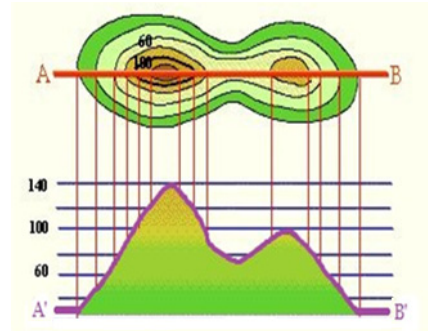


그림 3. 고도 문제의 예
Fig. 3 Example of Altitude problem

일반적으로 거리를 측정하는 알고리즘은 지구를 구 또는 타원체로 가정하거나 평면으로 투영한 후 각각 위치의 위도와 경도 값을 사용하여 계산한다. 지구의 표면은 높낮이가 다르기 때문에 경사가 존재하는데 구 또는 타원체로 가정하는 이유는 지구의 높낮이 정보를 모두 가지고 있을 수 없기 때문이다.

[그림 3]을 보면 좀 더 쉽게 이해할 수 있다. [그림 3]은 사용자가 경사가 있는 지형을 운동했을 때 등고선을 이용한 평면도와 단면도를 이용해 이동경로를 나타낸 것이다. A 지점과 B 지점은 운동 시작점과 운동 종료점을 의미한다. \overline{AB} 는 A 지점과 B 지점을 지형을 무시한 채 연결한 직선이고 $\overline{A'B'}$ 는 두 지점을 지형을 고려하여 연결한 선이다.

사용자가 A 지점에서 B 지점까지 운동을 했을 때 칼로리 측정을 하기 위해서는 운동 거리를 계산해야 한다. 단순히 두 지점 사이의 거리를 측정할 때는 두 지점을 연결한 직선인 \overline{AB} 의 길이를 구하면 된다. 하지만 사용자가 운동했을 때의 운동거리는 \overline{AB} 의 길이가 아니다. 왜냐하면 운동한 장소는 평지가 아닌 경사가 있는 곳이기 때문에 사용자는 경로 \overline{AB} 가 아닌 경로 $\overline{A'B'}$ 로 운동을 하기 때문이다. \overline{AB} 와 $\overline{A'B'}$ 의 차이는 경사가 심할수록 또는 경사가 수시로 바뀔수록 더 커지게 된다. 이러한 거리측정 할 때의 오류를 보완하기 위하여 다음 장에서는 DMACA를 제안한다.

3.2. DMACA (Distance Measuring Algorithm based Considering Altitude)

DMACA는 사용자의 이동거리를 좀 더 정확하게 파악하기 위해 제안하는 거리측정 알고리즘이다. 뒷장에서 살펴보았듯이 단순한 거리계산이 아닌 이동거리를 측정할 경우 오차가 생기게 되며 이러한 오차는 지구가 평평하지 않기 때문이란 것을 알 수 있었다. 이를 해결하기 위해 DMACA는 GPS 데이터중 하나인 고도 값을 이용한다. 고도는 평균해수면을 기준으로 하여 자신의 높이를 나타내는데 이 데이터를 이용하여 운동 중 지형의 높낮이를 감지 할 수 있다. DMACA의 알고리즘은 다음과 같으며 지구를 타원체로 가정한 거리측정 알고리즘을 기반으로 구현했다[10].

$$ML = (LAT1_{dd} + LAT2_{dd}) \div 2$$

$$KPD_{lat} = 111.13209 - 0.56605 \times \cos(2 \times ML) + 0.00120 \times \cos(4 \times ML)$$

$$KPD_{lon} = 111.41513 \times \cos(ML) - 0.09455 \times \cos(3 \times ML) + 0.00012 \times (5 \times ML)$$

$$NS = KDP_{lat} \times (LAT1_{dd} - LAT2_{dd})$$

$$EW = KPD_{lon} \times (LON1_{dd} - LON2_{dd})$$

$$DISTANCE_{direct} = \sqrt{NS^2 + EW^2}$$

$$DIFF_{alt} = ALT1 - ALT2$$

$$DISTANCE_{move} = \sqrt{DISTANCE_{direct}^2 + DIFF_{alt}^2}$$

그림 4. DMACA 알고리즘
Fig. 4 DMACA Algorithm

$LAT1_{dd}$ 와 $LON1_{dd}$ 는 첫 번째 좌표의 위도와 경로의 degree-decimal format 이고 $LAT2_{dd}$ 와 $LON2_{dd}$ 는 두 번째 좌표의 위도와 경로의 degree-decimal format 이다. $ALT1$ 는 첫 번째 좌표의 고도이고 $ALT2$ 는 두 번째 좌표의 고도이다. (1) $LAT1_{dd}$ 와 $LAT2_{dd}$ 를 이용하여 위도의 중간 값인 ML 을 구한다. (2) KPD_{lat} (Kilometer per degree)는 해당 위치에서 위도 1°의 거리이다. (1)번 식에서 구한 ML 을 이용하여 구한다. (3) KPD_{lon} (Kilometer per degree)는 해당 위치에서 경도 1°의 거리이다. (1)번 식에서 구한 ML 을 이용하여 구한다. (4) KPD_{lat} 를 이용하여 두 지점의 NS (North-South) 거리를 구한다.

(5) KPD_{lon} 를 이용하여 두 지점의 EW (East-West) 거리를 구한다. (6) 피타고라스 정리를 이용하여 두 지점의 거리인 $DISTANCE_{direct}$ 를 구한다. 이 거리는 지형을 고려하지 않은 평평한 지구로 가정했을 때의 직선거리이다. 다음 과정을 통해 더 정확한 이동거리를 측정할 수 있다. (7) 두 지점의 고도의 차이인 $DIFF_{alt}$ 를 구한다. (8) (6)번 식의 결과인 $DISTANCE_{direct}$ 와 (7)번 식의 결과인 $DIFF_{alt}$ 를 이용하여 $DISTANCE_{move}$ 를 구한다. 결과 값인 $DISTANCE_{move}$ 는 본 논문에서 제안하는 DMACA의 결과 값으로서 사용자가 이동한 거리이고 단위는 km 이다.

IV. 성능평가

4.1. 실험환경

제안하는 DMACA 알고리즘을 평가하기 위해 구글 맵을 이용하여 실험하였다. 구글맵은 평면도, 위성지도와 함께 지형도를 제공하고 있으며 지형도를 이용하면 각 지점들의 위도, 경도, 고도를 알아낼 수 있다[11]. 지형정보는 JSON 또는 XML로 제공하기 때문에 어떠한 언어를 사용하여 프로그램을 구현하더라도 지형정보를 받을 수 있다는 장점이 있다. 실험 환경은 자바스크립트로 구현하였고 구글에서 제공하는 Google Maps API ver.3를 사용하였다. 이로서 실험을 할 때 직접 사용자가 이동을 하면서 실험을 하지 않아도 거리측정이 가능하고 경사가 없는 평지 또는 완만한 경사, 일방적인 경사, 혼합 지형 등 보다 다양한 환경에서 손쉽게 실험이 가능하다. 실험 방법은 지도상에서 임의의 운동루트를 설정하고 해당 운동루트를 구성하고 있는 지점의 위도, 경도, 고도 정보를 받아온다. 받아온 정보를 이용하여 DMACA 알고리즘과 다른 거리측정 알고리즘을 적용하여 운동거리를 측정하고 어느 정도 성능향상이 있는지 알아본다.

4.2. 경사에 따른 성능평가

DMACA 알고리즘은 기존 거리측정 알고리즘과 달리 경사가 있는 지형을 고려한 알고리즘이다. 이번 장에서는 경사가 있는 지형에서 기존의 거리측정 알고리즘과 측정값의 차이를 알아본다. 실험방법은 동일한 두 지점

을 설정하고 경사도를 변화시키면서 이동거리를 측정한다. [그림 5]는 경사도에 따른 거리측정 결과를 나타내는 그래프이다. x축은 경사도로서 단위는 °이며 0°에서 35°까지 5° 단위로 변화시켰다. y축은 두 지점의 직선거리와 각각의 알고리즘으로 측정된 운동거리의 비율을 나타낸다. y축 값이 1.5라고 가정하면 두 지점의 직선거리와 해당 알고리즘으로 계산한 운동거리는 1.5배 차이가 난다. 파란색 막대 그래프는 Ellipsoidal Earth 알고리즘으로 측정된 값이고 빨간색 막대 그래프는 DMACA 알고리즘으로 측정된 값이다.

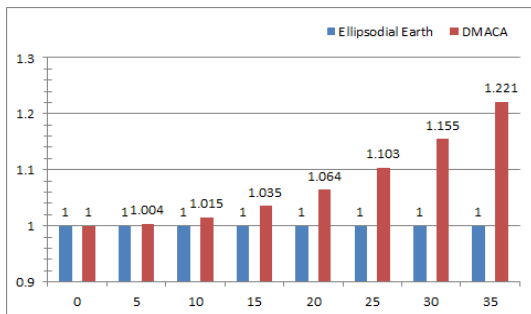


그림 5. 경사도에 따른 거리측정 결과 그래프
Fig. 5 Distance measurement graphs on slopes

결과에서 나타나듯이 Ellipsoidal Earth 알고리즘은 경사도를 고려하지 않기 때문에 경사도가 증가와 상관없이 결과 값은 항상 1이다. 반면에 DMACA 알고리즘은 y축 값은 최소 1부터 최대 1.221까지 나온다. 0°에서는 경사도가 없기 때문에 Ellipsoidal Earth와 같이 1이 나오며 경사도가 증가 할수록 지수적으로 값이 증가한다. 즉, 두 지점의 직선거리가 동일하여도 경사도에 따라 이동 거리에 영향을 미치며 경사도가 높을수록 이동거리가 상승하는 것을 알 수 있다.

4.3. 실제지형에서 성능평가

앞장에서 보았듯이 지형의 경사도에 따라 직선거리와 실제 이동거리가 차이가 나는 것을 알 수 있었다. 이번실험에서는 실제 지형에서 DMACA 알고리즘과 기존의 거리측정 알고리즘을 사용하여 이동거리를 측정한다.

측정방법은 완만한 경사 및 평지, 산악지형, 이 두 가지 유형이 복합적으로 나타나는 혼합지형 등 총 3가지

유형으로 나눈 후 DMACA, Spherical Earth, Ellipsoidal Earth 알고리즘을 사용하여 각각의 이동거리를 계산한다. 이동경로 측정방법은 각 군에 특성에 맞는 임의의 루트를 설정하고 루트 내에서 임의의 지점을 샘플링 한다. 그 후 각 지점간의 거리를 3 각 유형은 경사도를 이용하여 구분한다. 경사도는 도로가 기울어진 정도를 의미한다.

$$\text{경사도} = \frac{\text{두 지점의 높이의 차}}{\text{두 지점의 직선거리}} \times 100$$

경사도가 0%일 경우 두 지점의 고도가 같을 때이며 평지를 의미하고 경사도가 높을수록 두 지점 간의 고도 차이가 큰 것이다. 따라서 경사도에 따라 각 지형의 유형을 나누었다.

완만한 경사 및 평지로 이루어진 유형은 경사도가 최소 -5% 이상 최대 5%이하인 곳으로 도심지 또는 논과 같은 평지를 의미한다. 산악지형은 경사도에 따로 제한이 없으며 도로가 있는 일반적인 산을 의미한다. 혼합지형은 두 가지 유형이 복합적으로 나타나되 경사도에 제한은 없지만 평균 경사도가 ±10% 인 곳을 의미한다.

[그림 6]의 (a), (b), (c)는 각 유형의 지형도와 유형의 고도를 나타내는 그래프이다. 이동경로는 지형도에서 빨간선으로 표시하였으며 이동거리는 DMACA, Ellipsoidal Earth, Spherical Earth 총 3가지 알고리즘을 이용하여 계산하였다[12].

그래프의 X축은 각 지점의 번호를 의미하고 Y축은 고도를 의미하며 단위는 m(미터)이다. 각 지점마다 고도를 영역 그래프로 그렸기 때문에 단면도와 유사한 그래프로 해당 이동경로의 대체적인 유형을 파악할 수 있다. 또한 각 알고리즘의 결과를 쉽게 알아볼 수 있도록 이동거리는 비슷한 거리로 설정하였다.

[그림 6]의 (a)는 춘천 시내의 지형도이며 완만한 경사 및 평지지형을 실험하기 위한 곳이다. 고도는 약 80~120m 정도로 나타나며 등고선의 변화가 거의 없고 경사도는 최대 4.3%이다. 총 51개의 지점에서 샘플링 후 거리를 계산했다. [그림 6]의 (b)는 춘천 - 양구간 도로의 일부의 지형도이며 산악 지형을 실험하기 위한 곳이다. 고도는 약 220~520m 사이로 나타나며 산악 지형이기 때문에 등고선 변화가 심하며 경사도는 최소 -26.9%에서 최대 32%까지 다양하게 나타난다. 총 64개 지점에서 샘플

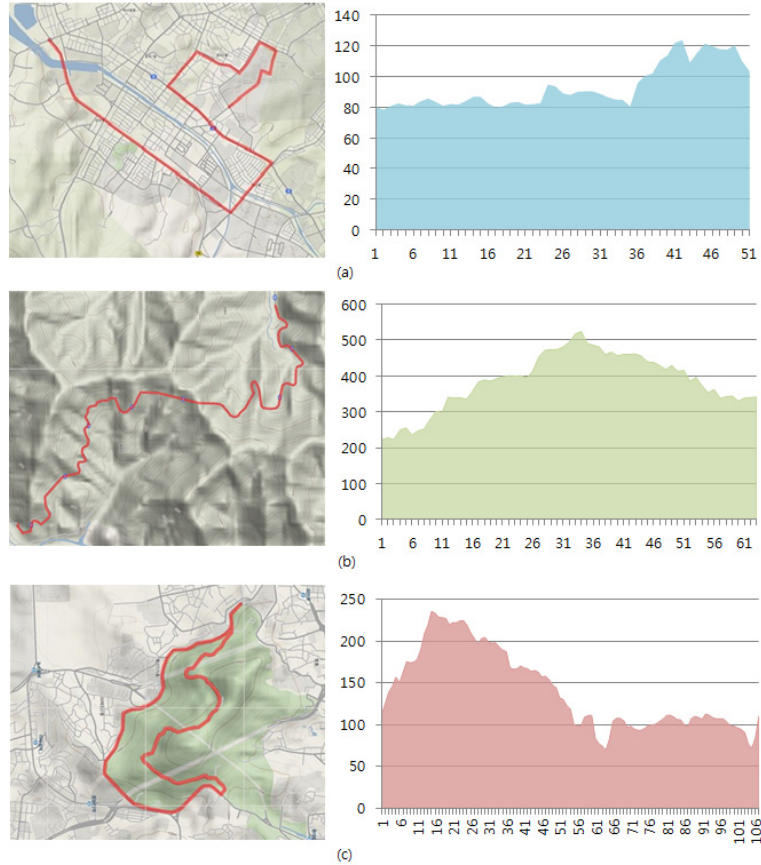


그림 6. 지형도와 유형의 고도를 나타내는 그래프
 Fig. 6 Graphs of topographic map and altitude types

링 후 거리를 계산했다. [그림 6]의 (c)는 남산 일대의 도로의 지형도이며 혼합 지형을 실험하기 위한 곳이다. 남산 및 주변도로가 적당한 경사로 이루어져 있기 때문에 설정하였다. 그래프를 보면 산악 유형에 비해 경사가 심한 것처럼 보인다. 하지만 Y축의 범위 및 보조선의 간격 때문에 일어난 현상으로 약 두 배 정도 가파르게 보인다. 경사도는 최대 경사도는 30.4%였지만 대부분 적당한 경사도였으며 평균 경사도는 7.6% 정도다. 총 106개 지점에서 샘플링 후 거리를 계산했다.

[그림 7]은 본 실험의 결과를 나타내는 그래프이다. x축은 총 이동거리를 나타내며 단위는 m(미터)이다. Y축은 유형에 따라 평지, 혼합지형, 산악지형 총 세 가지로 나누었으며 각 유형별로 초록색은 DMACA, 빨간색은

Ellipsoidal Earth, 파란색은 Spherical Earth 알고리즘의 결과를 나타낸다.

완만한 경사 및 평지일 때 DMACA의 결과 값은 7,938.67m 로 가장 큰 값이 나왔으며 Ellipsoidal Earth 는 7935.33m, Spherical Earth는 7,931.93m 로 나타났다. DMACA 알고리즘은 타 알고리즘과 3~7m 정도로 비교적 적은 차이를 보이고 있다. DMACA의 기초 알고리즘인 Ellipsoidal Earth와 DMACA가 결과 값이 차이나는 이유는 평지유형에도 작지만 어느 정도 경사도가 존재하기 때문이다. 그러나 경사도가 적기 때문에 결과 값 역시 미묘하게 나타난다.

혼합 지형일 때 DMACA의 결과 값은 7,937.67m 로 가장 큰 값이 Ellipsoidal Earth 는 7903.25m, Spherical Earth

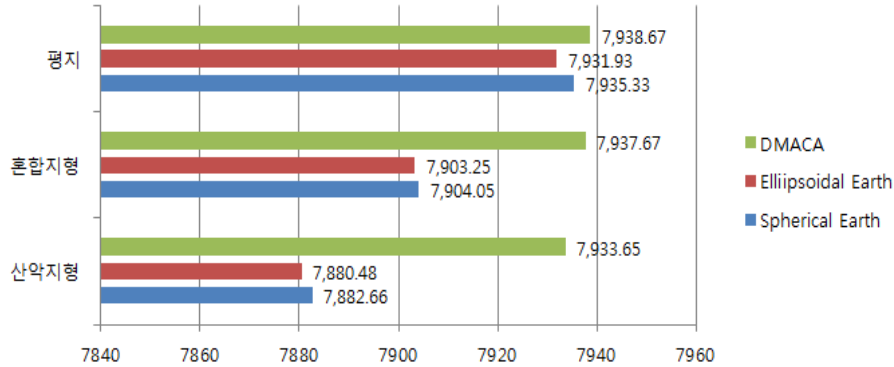


그림 7. 각 알고리즘의 결과 값 그래프
Fig. 7 Result of algorithm

는 7,904.05m 로 나타났다. DMACA 알고리즘은 타 알고리즘과 35m 정도로 평지유형과 비슷한 거리에서 많은 차이를 보이고 있다. 이러한 이유는 경사도가 평지유형에 비해 높았기 때문이다. 그러나 타 알고리즘은 고도를 고려하지 않기 때문에 1m 정도로 매우 적은 차이를 보이고 있다.

산악 지형일 때 DMACA의 결과 값은 7,933.65m 로 가장 큰 값이 나왔으며 Ellipsoidal Earth 는 7880.48m, Spherical Earth는 7,882.66m 로 나타났다. DMACA 알고리즘은 타 알고리즘과 약 53m 정도로 평지유형 뿐만 아니라 혼합 지형과도 차이를 보이고 있다. 경사도가 혼합 지형에 비해 차이가 많이 나기 때문이다.

실험을 통해 DMACA 알고리즘은 지형에 따라 결과 값이 다른 것을 알 수 있으며 경사도가 높을수록 그 차이가 많이 나는 것을 알 수 있다.

라이벌의 실황을 볼 수 있으며 라이벌 뿐만 아니라 근처 사용자들도 볼 수 있게 설계 하였다. 또한, 지도 내에 경쟁구간을 두어 랭킹 서비스를 제공하였다. 이 경쟁 구간에서 가장 기록이 좋은 사용자 3명을 항상 확인 할 수 있게 하여 모든 사용자로 하여금 경쟁의식을 가지도록 하였다.



그림 8. 라이프가이드 시스템의 배경도
Fig. 8 Context diagram of life-guide System

V. 라이프가이드 시스템 설계 및 구현

기존 운동 보조기구들의 특징은 운동 정보제공 뿐만 아니라 경쟁 시스템 도입으로 인한 사용자의 흥미를 이끈다는 점이었다. 본 시스템에도 사용자 경쟁 시스템을 도입하였다. 기존 시스템은 간접적인 경쟁방식으로 운동을 마치고 홈페이지에서만 기록을 사용한 경쟁이었지만 본 시스템에서는 라이벌 기반의 직접적인 경쟁 방식을 사용한다. 운동 도중에도 자신의 운동 정보와

표 1. 시스템의 세부 사양
Table. 1 System specifications

사양	구성 요소	
	서버	스마트폰
CPU	Intel i7 860	Qualcomm QSD8250 snapdragon
OS	Windows 7	Android 2.2
개발 언어	C#	JAVA(Andorid)

[그림 8]과 같이 여러 사용자를 두고 서비스를 제공해야 하므로 서버와 스마트폰으로 구성했다. [표 1]은 각 시스템의 사양이다. 서버의 CPU는 i7이고 윈도우즈7을 설치했으며 언어는 C#을 이용하여 만들었다. 다양한 사용자의 접속 및 서비스 제공이 가능하다. 스마트폰은 HTC의 Desire를 사용하였다.

[그림 9]는 운동을 하고 있는 상황을 캡처한 화면이다. 기본적인 구성을 지도와 운동 정보로 나뉜다. 지도는

자신이 있는 좌표를 중심으로 나타나며 지도의 줌업 및 줌다운 그리고 이동이 가능하기 때문에 다른 지역의 지도를 확인 할 수도 있다. 운동 정보는 시간과 거리, 소모 칼로리 등을 알려준다.

[그림 9]의 왼쪽 그림은 사용자 및 다른 사용자를 그룹 모드를 이용했을 때 모습이다. 사용자는 초록색 아이콘을 이용하여 자신의 위치를 표시하며 빨간색 라인을 이용하여 자신이 뮌 경로를 지도를 통하여 확인 할 수 있

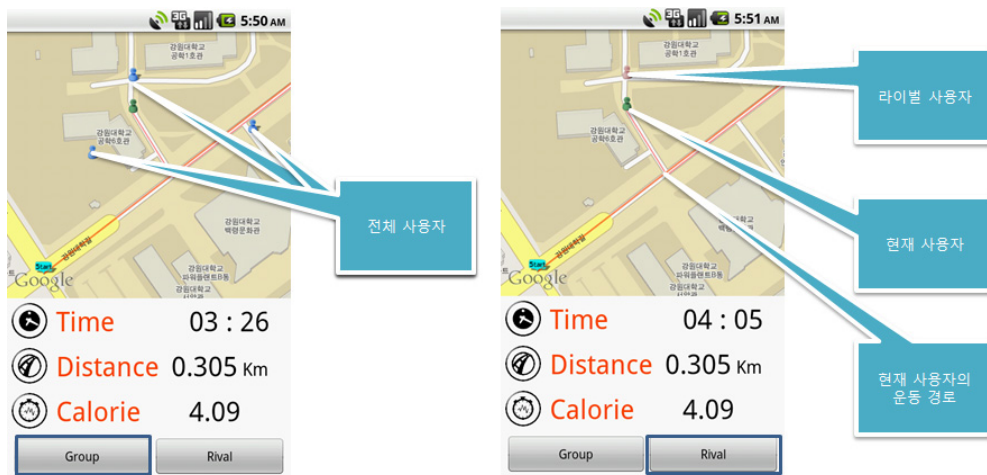


그림 9. 운동 시 실황 중계 화면
Fig. 9 Real-Time exercise demonstration

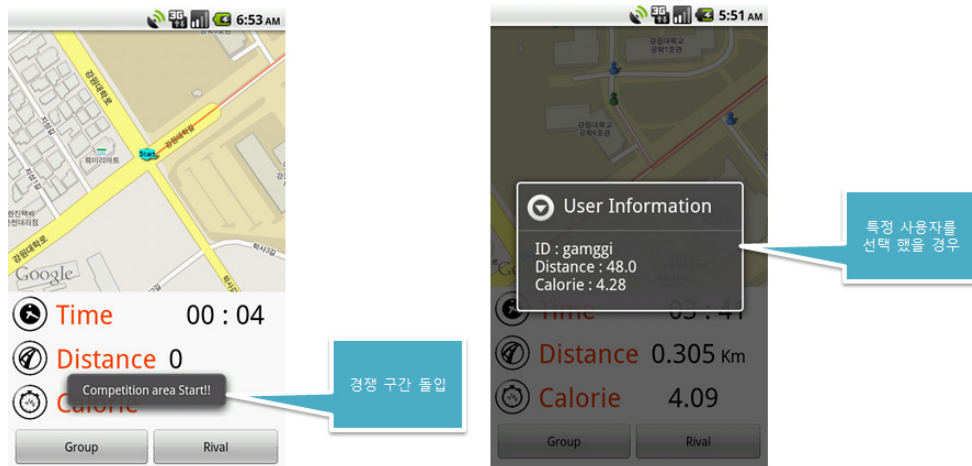


그림 10. 랭킹 서비스 및 타 사용자 정보
Fig. 10 Ranking service and other client information

다. 또한 그룹 내 사용자들은 파란색 아이콘으로 표시하여 근처에 어느 정도 사용자가 있는지 알 수가 있다. [그림 9]의 오른쪽 그림은 라이벌 모드로 설정했을 경우의 화면이다. 그룹모드와 다르게 라이벌의 아이콘만 표시하며 빨간색으로 나타낸다.

[그림 10]은 랭킹 서비스와 다른 사용자 정보를 알려주는 화면이다. 왼쪽 그림을 보면 하늘색 아이콘이 보이는데 그 아이콘이 랭킹 구간의 시작을 의미한다. 사용자가 시작구간에 진입을 하면은 진동과 함께 토스트 메시지를 띄우면서 사용자에게 알려준다. 또한 아이콘을 터치할 경우 해당 랭킹 구간의 베스트 3의 기록을 알려준다. 오른쪽 화면은 다른 사용자의 아이콘을 눌렀을 때 나오는 화면으로서 상대방의 운동 거리와 소모 칼로리를 확인 할 수 있게 하였다.

VI. 결 론

사회복지가 발달함에 따라 다양한 운동을 즐길 수 있는 시설들이 생기게 되었다. 이러한 변화는 사람들의 삶의 질 향상으로 인해 건강에 대한 관심이 크게 늘어난 것이라고 볼 수 있다. 이러한 사회변화에 맞춰 스포츠 관련 기업들은 운동 보조 시스템들을 출시하기 시작했다. 특히 서비스에 강점을 가지고 있는 걷기, 조깅 등 유산소 운동관련 보조 시스템들이 각광을 받고 있는 중이다. 운동 보조 시스템의 가장 중요한 요소는 정확한 운동거리를 측정해야 한다. 운동거리를 기반으로 다양한 정보를 산출해 낼 수 있기 때문이다. 그러나 지금 사용되는 보조 시스템은 경사면을 이용하지 않는 거리측정 알고리즘을 사용하기 때문에 실제 운동거리와 오차가 생기게 된다.

본 논문에서는 실제 운동거리와 거리 측정 알고리즘간의 거리오차를 줄일 수 있는 DMACA 알고리즘을 제안한다. 이 알고리즘은 기존 거리측정 알고리즘과 GPS 데이터의 고도 정보를 사용하여 거리를 산출하기 때문에 더 정확한 거리 측정을 할 수 있다.

실험 결과에 따르면 동일 지점에서 거리를 측정할 때 경사도가 늘어날수록 거리는 지수적으로 증가하는 것을 확인하였고 실제 지형별 실험에서 산악 지형의 경우 거리측정 알고리즘과 최대 0.8%까지 차이나는 것을 확인 할 수 있다.

사용자에게 더 정확한 운동 보조 시스템을 이용할 수 있도록 DMACA 알고리즘을 기반으로 하는 라이프가이드 시스템을 구현했다. 라이프가이드 시스템은 직접적인 경쟁 시스템 도입으로 인해 사용자가 흥미를 느끼면서 운동을 할 수 있게 보조해주는 애플리케이션이다. 그러기 위해 라이벌 선정 및 경쟁구간을 도입하였다. 사용자는 운동 도중에 자신이 위치한 지도, 운동 거리, 소모 칼로리 등 다양한 정보를 확인 할 수 있으며 다른 사용자의 정보도 실시간으로 받아 올 수 있다. 또한 경쟁구간에서는 최고기록을 확인 할 수가 있어서 사용자에게 경쟁심을 유발한다는 장점이 있다.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음(NIPA-2012-H0401-12-1002)

참고문헌

- [1] 건강보험심사평가원, <http://www.hira.or.kr>
- [2] ML Pollock, "Health and Fitness Through Physical Activity", 1978.
- [3] ML Pollock, "Exercise prescription for the cardiac patient: an update.", Clin Sports Med 3(2), pp 425-442, 1984
- [4] Elizabeth D. Mynatt, "IT in healthcare: a body of work", Magazine interactions, Vol.18 Issue 3, pp 22-25, May + June 2011
- [5] Nike+. <http://nikerunning.nike.com/>
- [6] Adidas Mi Coach 2009. <http://www.adidas.com/us/micoach/#HowMiCoachCoaches/sdf/mdf>
- [7] Ainsworth BE, "Compendium of Physical Activities: an update of activity codes and MET intensities", Med Sci Sports Exerc, Vol.32, pp 498 - 516, 2000
- [8] Nuala M. Byrne, "Metabolic equivalent: one size does not fit all", Journal of Applied Physiology, Vol.99 no.

- 3, pp 1112-1119, 2005
- [9] Lippincott Williams & Wilkins, "ACSM's Guidelines for Exercise Testing and Prescription", 2009
- [10] Geographical distance.
"http://en.wikipedia.org/wiki/Geographical_distance"
- [11] Google Maps,
"http://code.google.com/intl/en/apis/maps/index.html"
- [12] Spherical Earth,
"http://en.wikipedia.org/wiki/Spherical_Earth"

저자소개



김대호(Daeho Kim)

2010년 ~ 2012년 강원대학교
컴퓨터정보통신공학
공학석사

※ 관심분야: 스마트폰, 임베디드 S/W, 센서네트워크



정인범(Inbum Jeong)

2001년 ~ 현재 강원대학교
컴퓨터정보통신공학
전공교수

※ 관심분야: 운영체제, 소프트웨어공학, 멀티미디어
시스템, 센서네트워크