JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 26 No. 6, pp. 37-46, June 2021
https://doi.org/10.9708/jksci.2021.26.06.037

A Design and Implementation of Yoga Exercise Program Using Azure Kinect

Jong Hoon Park*, Dae Han Sim*, Young Pyo Jun**, Hongrae Lee**

*Student, Division of Computer and Telecommunications Engineering, Yonsei University, Wonju, Korea
*Student, Division of Computer and Telecommunications Engineering, Yonsei University, Wonju, Korea
**Professor, Software Division, Yonsei University, Wonju, Korea
**Professor, Software Division, Yonsei University, Wonju, Korea

[Abstract]

In this paper, we designed and implemented a program to measure and to judge the accuracy of yoga postures using Azure Kinect. The program measures all joint positions of the user through Azure Kinect Camera and sensors. The measured values of joints are used as data to determine accuracy in two ways. The measured joint data are determined by trigonometry and Pythagoras theorem to determine the angle of the joint. In addition, the measured joint value is changed to relative position value. The calculated and obtained values are compared to the joint values and relative position values of the desired posture to determine the accuracy. Azure Kinect Camera organizes the screen so that users can check their posture and gives feedback on the user's posture accuracy to improve their posture.

▸ **Key words**: Computer Vision, Home Training, Joint, Posture Measurement, Azure Kinect

[요 약]

본 논문에서는 Azure Kinect를 사용하여 요가 자세의 정확도를 측정하고 판단하는 프로그램을 설계하고 구현하였다. 이 프로그램은 Azure Kinect Camera와 센서를 통해 사용자의 모든 관절 위치를 측정한다. 측정한 관절의 값은 두 가지 방법으로 정확도를 판단하는 데이터로 사용된다. 측정된 관절 데이터는 삼각법과 피타고라스의 정리를 통하여 관절의 각도를 구한다. 또한, 측정된 관절 값은 상대적인 위치 값으로 변경한다. 각각 계산하여 구한 값은 목표하고자 하는 자세의 관절 값 및 상대적 위치 값과 비교하여 정확도를 판단한다. Azure Kinect Camera를 통해 사용자가 본인의 자세를 확인할 수 있도록 화면을 구성하고 사용자의 자세 정확도를 피드백으로 전달해 사용자의 자세 향상을 유도한다.

▸ **주제어**: 컴퓨터 비전, 홈 트레이닝, 관절, 자세측정, 에저 키넥트

• First Author: Jong Hoon Park, Co-Authors: Young Pyo Jun, Hongae Lee, Corresponding Author: Dae Han Sim
*Jong Hoon Park (jongh@yonsei.ac.kr), Division of Computer and Telecommunications Engineering, Yonsei University
*Dae Han Sim (dhsim@yonsei.ac.kr), Division of Computer and Telecommunications Engineering, Yonsei University
**Young Pyo Jun (ypjun@yonsei.ac.kr), Software Division, Yonsei University
**Hongrae Lee (vim0614@gmail.com), Software Division, Yonsei University
• Received: 2021. 04. 20, Revised: 2021. 05. 19, Accepted: 2021. 05. 19.

I. Introduction

현대 사회에서 운동은 사람들의 삶의 만족도를 높일 수 있는 주요 요인 중 하나로 여겨진다[1]. 코로나19의 영향으로 요가, 헬스, 필라테스, 수영장 같은 다중이용시설을 방문하는 것이 어려움에도 유튜브, 앱 등 다양한 홈 트레이닝 플랫폼을 이용하여 비대면을 통해 집에서 혼자 운동을 하는 사람들이 늘고 있다[2]. 그러나, 시장조사 기관 엠브레인 트렌드모니터(trendmonitor.co.kr)의 '운동'을 하는 전국 만 19~59세 성인남녀 1,000명을 대상으로 '운동'에 대한 인식 및 '홈트레이닝'에 대한 설문조사에 따르면, 65.8%의 사람들은 홈 트레이닝만으로 일정부분 운동효과를 볼 수 있지만, 59%의 사람들은 전문가의 도움이 없는 운동에는 한계가 있다고 생각하는 것으로 조사되었다[3]. 특히 요가 및 필라테스와 같은 자세 교정 운동을 홈 트레이닝만으로 할 경우, 운동 자세에 대한 피드백이 없기 때문에 잘못된 자세로 운동을 지속하는 경우가 발생하며 이에 따른 운동 효과 저하 및 근육 손상 등의 부상이 유발되기도 한다.

이러한 한계점을 개선하기 위해 본 논문은 Kinect 센서를 사용하여 사용자의 요가 자세를 인식하고, 받아온 데이터를 통해 평가하여 정확도를 산출하는 프로그램을 설계 및 구현하였다. 사용자는 각 자세에 대하여 정확도를 설정할 수 있고, 설정한 정확도 이상으로 자세를 5초간 유지하면 다음 자세로 넘어가도록 구현하여 운동 자세에 대한 피드백을 받을 수 있게 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 프로그램에서 사용하는 Kinect Camera의 사용법과 자세 인식에 관한 기존 연구를 소개하고, 3장에서는 구현한 프로그램에서 사용한 자세 인식 방법 및 정확도 측정 방법을 설명한다. 4장에서는 실험 결과를 설명하며 5장에서는 결론과 향 후 연구에 대해 기술한다.

II. Preliminaries

1. Review of Azure Kinect

Azure Kinect는 Microsoft 사에서 개발한 Camera의 하나로 RGB camera, Depth camera, IR emitters, microphone array, gyroscope, accelerometer를 탑재하고 있다.

그림 1은 Azure Kinect의 하드웨어 구성을 나타낸다. Azure Kinect의 RGB camera (Color camera)는 최대 4K (3840 x 2160) 해상도에서 30 FPS(Frame per Second)를 지원하며, 다른 해상도에 따른 Ratio와 FPS는 표 1과 같다.

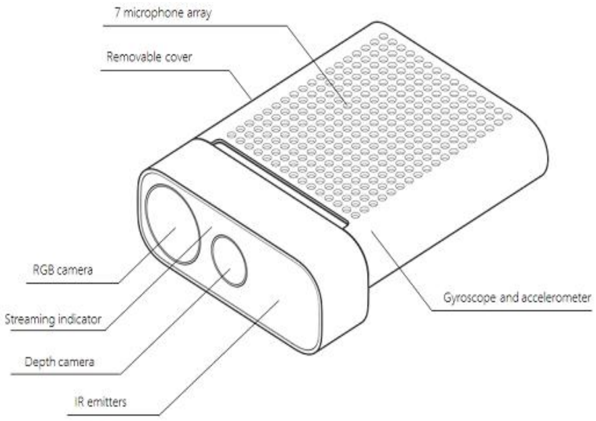

Fig. 1. Azure Kinect Hardware Configuration

Table 1. RGB Camera FPS by Resolution

Resolution	Ratio	FPS
3840 x 2160	16:9	0, 5, 15, 30
2560 x 1440	16:9	0, 5, 15, 30
1920 x 1080	16:9	0, 5, 15, 30
1280 x 720	16:9	0, 5, 15, 30
4096 x 3072	4:3	0, 5, 15
2048 x 1536	4:3	0, 5, 15, 30

Azure Kinect의 Depth camera는 물체 대상까지의 거리를 확인할 수 있는 Camera이다. Depth camera는 IR emitters와 함께 작동하며, ToF(Time-of-Flight) 방식을 사용한다. Depth camera의 모드에 따른 해상도와 FPS는 표 2와 같다. 그림 2는 Azure Kinect Camera를 통해 측정 가능한 사용자의 관절 위치를 나타낸다. 사용자의 관절은 Body Tracking SDK를 사용해 위치를 확인할 수 있다.

Table 2. Resolution and FPS by Depth Camera Mode

Mode	Resolution	FPS
NFOV unbinned	640×576	0, 5, 15, 30
NFOV 2x2 binned	320×288	0, 5, 15, 30
WFOV 2x2 binned	512×512	0, 5, 15, 30
WFOV unbinned	1024×1024	0, 5, 15
Passive IR	1024×1024	0, 5, 15, 30

본 논문의 프로그램은 Microsoft에서 제공하는 SDK[4]를 사용하여 구현하였다. SDK는 개발자가 Camera에 접근할 수 있는 함수, Camera의 초기 설정을 진행할 수 있는 함수, Color camera 데이터에 접근할 수 있는 함수, 사람이 탐지되었을 때 사람의 수와 사람의 관절 데이터를 불러오는 함수 등을 제공한다.

그림 3은 Azure Kinect Camera의 처리 과정을 나타낸다. OPEN 함수를 통해 Kinect와 컴퓨터가 연결되며,

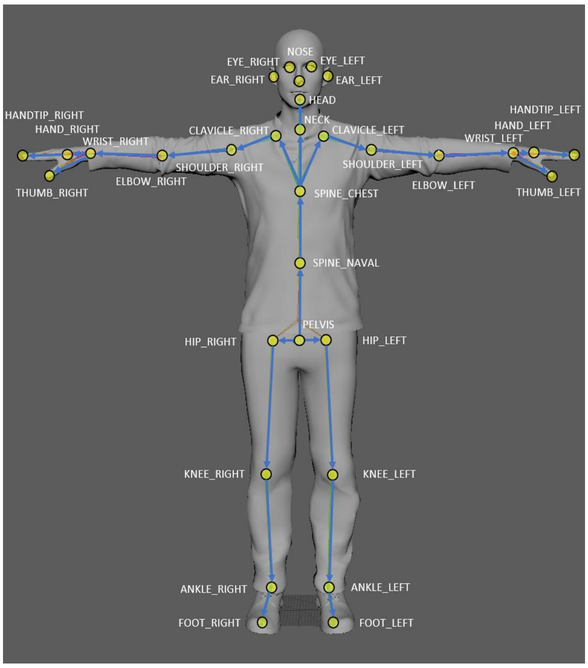

Fig. 2. Measurable joint position by using Azure Kinect Camera

Kinect는 프로그램 내에서 하나의 객체로 사용된다. CONFIGURE 과정에서는 Color Format, Color Resolution, Depth Mode, Synchronized Images Only, Camera FPS를 설정한다. 설정값에 따라서 Camera는 초당 5~30장씩 사진을 연속적으로 촬영한다. Camera를 통해 한 장의 사진을 촬영하는 동작을 Capture라고 한다. Capture는 Start에서 시작하여 Stop 까지 계속 작업을 진행한다. Camera에서 캡처된 사진은 Depth to color, Segmentation, Draw 등을 통해, 개발자가 필요한 다양한 작업을 진행할 수 있다.

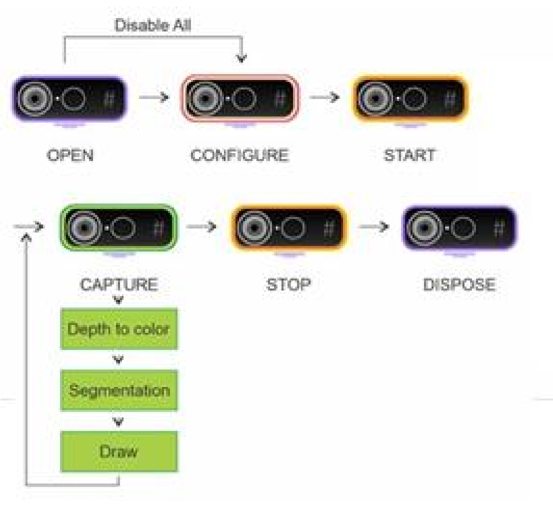

Fig. 3. Azure Kinect Camera Process

2. Recognizing the user's posture

Kinect Camera로 사용자의 자세를 인식하는 연구들은 다양하게 진행되었다. [5]에서는 고령자를 위한 프로그램을 개발하였다. 목표 자세는 만세, 양팔 벌리기, 차렷, 무릎들기, 다리 옆으로 들기 등이 있으며 다소 난이도가 높은 스쿼트, 런지와 같은 다양한 운동 자세도 포함되어 있다. 사용자의 관절 좌표를 3D 좌표로 받아온 후 X-Y축 각도, X-Z축 각도를 구하여 사용자의 자세를 인식하였다. 그러나 [5]에서는 전신이 나오지 않는 경우 오류가 발생하기 때문에 전신 자세 인식만 가능하다는 제약이 있다. [6]에서는 사용자가 스마트 거울 앞에서 운동할 때 Kinect Camera로 사용자의 자세를 인식하여 피드백을 준다. [6]에서는 Capture를 통한 데이터를 사용하여 사용자의 관절 각도를 계산하여 Lateral Raise, Lat Pull Down 등의 동작을 인식하였다. [7]에서는 데이터베이스에 운동 동작에 따른 관절의 Quaternion(X, Y, Z, Rotation) 값을 저장한 후 사용자의 운동 자세와 실시간으로 비교하여 각각의 값을 유클리드 거리공식에 대입하여 정확도를 측정하였다.

앞서 소개한 세 개의 논문은 Azure Kinect의 이전 버전인 Kinect v2를 사용하며, 사용자의 자세 인식을 위해 한 가지 방법만을 제시하였다. 본 논문에서는 Azure Kinect를 사용하여 Kinect v2보다 사용자의 관절 포인트를 더 많이 사용할 수 있어 정확하게 사용자의 자세를 인식할 수 있으며 사용자의 자세 측정에 두 가지의 방법을 동시에 적용하여 보정 적용하여 더욱 객관적인 자세 인식과 정확도를 측정할 수 있다.

3. Assessing the user's posture

사용자의 자세를 정확하게 평가하면 피드백을 전달할 수 있다. 사용자에게 정확한 피드백은 자세교정에 있어 도움이 된다. 사용자에게 피드백을 주는 방법으로는 시각 청각 등 감각기관을 사용할 수 있다. 청각으로는 사용자의 잘못된 자세를 소리로 알려줄 수 있고, 시각으로는 사용자의 자세와 목표하는 자세를 비교해서 보여줌으로써 사용자가 잘못된 자세를 인지할 수 있도록 한다. 홈 트레이닝 사용자에게 피드백은 운동 효과 향상에 도움이 된다.

[8]에서는 웨어러블 디바이스와 Kinect를 동시에 사용한다. 두 개의 기기는 사용자의 관절 각도를 계산하며 사용자의 관절 데이터의 평균과 목표 자세 관절 데이터의 평균 간 차이를 계산하여 계량화한다. 평균 간 차이를 계산하는 방식은 사용자가 잘못된 방식으로 운동하는 것은 파악할 수 있지만, 어느 관절이 바르게 위치하지 못하고 있는지는 파악할 수 없다.

[9]에서는 사용자의 자세 평가를 위해 15개 관절 사이의 거리를 구해 105개의 데이터를 사용한다. 각 목표 자세마다 준비된 105개의 데이터는 정규화 과정을 거친 후, K-Means 알고리즘을 적용하여 군집화하며 군집화된 값과 사용자의 각 관절의 차이 값을 비교하여 사용자의 자세를 평가한다. [9]에서는 사용자의 자세에서 잘못된 부위를 알려주지만, 정확도를 표시하지는 못한다. 본 논문은 사용자에게 정확도를 함께 알려줌으로서 사용자가 정확한 자세로 운동할 수 있도록 유도한다.

III. Implementation of posture recognition using Azure Kinect

사람마다 키, 몸무게, 자세 등 체형이 모두 다르기 때문에 같은 자세에 대한 운동하는 사람의 특정 관절을 2D 좌표상에서의 신체 특정 부위 위치로 판별하면 사용자마다 관절 위치 측정값은 다르게 인식되어 정확하게 판단하기 어렵다. 본 장에서는 사용자의 체형에 따른 자세를 정확히 판별하기 위해 사용한 방법들을 설명한다.

1. Measuring joint angles using trigonometry

체형이 다른 두 사람이 같은 자세를 취했을 때, 동일한 관절 사이 각도는 체형의 영향을 받지 않고 동일하게 측정된다. 이러한 특징을 사용하면, 사용자의 관절 각도와 목표한 자세의 관절 각도를 비교해 정확하게 자세를 판별할 수 있다. 사용자의 관절 각도는 Kinect Camera로 측정한 각 관절의 위치값으로 측정할 수 있다.그림 4는 운동하는 자세에서 각 관절 각도를 측정하는 예를 나타낸다. 무릎 각도를 구하기 위해서는 Kinect Camera로 골반, 무릎, 발목 세 점의 위치를 각각 측정한다. 측정된 값을 사용하여 골반과 무릎, 발목과 무릎의 길이를 구하고, 마지막으로 측정된 관절 위치와 관절 간의 거리를 삼각법과 피타고라스 정리를 적용하면 무릎 각도를 구할 수 있다.

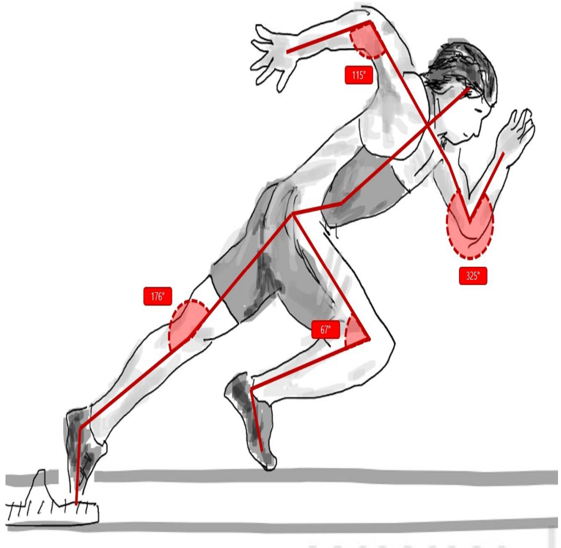

Fig. 4. Example of a measurable joint in the body

그림 5는 삼각법을 사용한 관절의 각을 계산하는 과정을 나타낸다. 각도를 구할 관절의 위치를 점 Q로 나타낸다. 점 Q를 원점으로 이동하면 점 Q‘로, Q’를 기준으로 점 P는 점 P’로, 점 R은 점 R‘로 이동된다. 점 P’와 X축이 이루는 각을 θ_1로 나타내며 이는 수식 1을 통해 구할 수 있다.

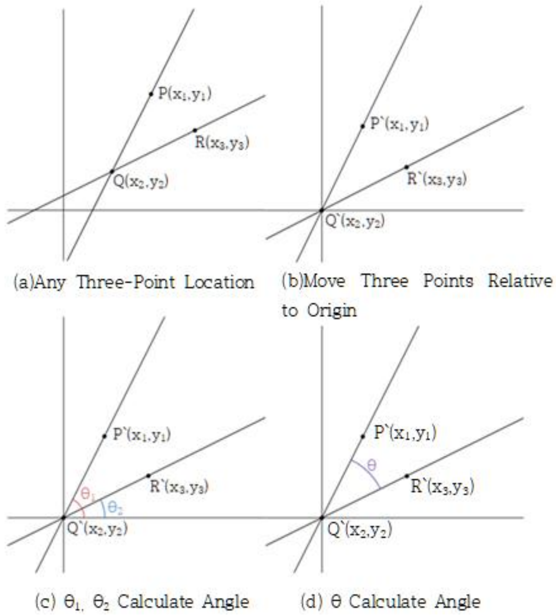

Fig. 5. calculating joint angles using trigonometry

$$\theta_1 = \tan^{-1}\left(\frac{y_1 - y_2}{x_1 - x_2}\right) \qquad \text{(Formula 1)}$$

위와 같은 방식으로 점 R’와 X축이 이루는 각을 θ_2로 나타내며 이는 수식 2를 통해 구할 수 있다.

$$\theta_2 = \tan^{-1}\left(\frac{y_3 - y_2}{x_3 - x_2}\right) \qquad \text{(Formula 2)}$$

구하고자 하는 관절의 각 θ는 θ_1에서 θ_2을 뺀 값으로 구할 수 있으며 이를 수식 3으로 나타낼 수 있다.

$$\theta = \tan^{-1}\left(\frac{y_2 * (x_1 - x_3) + y_1 * (x_3 - x_2) + y_3 * (x_2 - x_1)}{(x_2 - x_1) * (x_1 - x_3) + (y_2 - y_1) * (y_1 - y_3)}\right)$$

(Formula 3)

수식3에 사용되는 Arctangent로 관절 각도를 구하면, θ의 범위는 -90도에서 90도 사이의 값을 반환하게 된다. 하지만 θ가 -45도인 경우, 135도인지 315도인지가 구별하기 어려워진다. 이러한 문제를 해결하기 위해서는 θ의 범위를 0도에서 360도로 변환이 필요하다.

그림 6은 Acrtangent 결과의 범위에 따른 θ의 네 가지 구역을 나타낸다. 구역 A는 0도에서 90도, 구역 B는 90도에서 180도, 구역 C는 180도에서 270도, 그리고 구역 D는 270도에서 360도로 정의된다. 수식 3의 결과값이 -90도에서 0도로 반환되면, 관절의 각은 구역 B와 D의 θ_2 또는 θ_4가 되고 0도에서 90도로 반환되면 구역 A와 C의 θ_1 또는 θ_3이 된다. θ_1, θ_2, θ_3, θ_4는 피타고라스의 정리를 사용하여 관절 각도가 예각과 둔각인지에 따라 결정된다. 예각과 둔각을 결정하는 피타고라스 정리는 아래 수식 4와 5로 나타낸다.

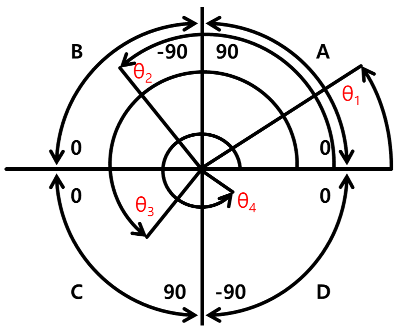

Fig. 6. Four zones of θ according to the range of Arctangent results

$$(last\_side)^2 < (upper\_side)^2 + (lower\_side)^2$$

(Formula 4)

$$(last\_side)^2 > (upper\_side)^2 + (lower\_side)^2$$

(Formula 5)

위의 수식에서 upper_side는 윗변, lower_side는 아랫변, last_side는 나머지 한 변으로 정의한다. 수식 4는 윗변과 아랫변 길이의 제곱의 합이 나머지 한 변의 제곱보다 큰 경우를 나타내며 이 조건을 만족하면 예각으로 θ값이 양수이면 θ_1로 음수이면 θ_4로 결정된다. 수식 5는 윗변과 아랫변 길이의 제곱의 합이 나머지 한 변의 제곱보다 작은 경우를 나타내며 이 조건을 만족하면 둔각으로 θ값이 양수이면 θ_3으로 음수이면 θ_2로 결정된다.

그림 7은 삼각법과 피타고라스 정리를 사용한 각 관절 각도를 측정하는 과정을 소스 코드로 나타냈다. Angle 함수는 각도를 측정할 관절의 위치와 연결되는 관절들의 위치를 매개변수로 입력받는다. 입력받은 변수들을 사용해 수식 3을 수행하며, Math.Atan함수로 인해 Theta는 라디안의 값을 가진다. Theta_angle은 Theta 값에 (180 / Math.PI)를 곱하여 라디안에서 각도로 변환한 값을 갖는다.

```
Angle(x1, y1, x2, y2, x3, y3){
/* Trigonometry */
Theta=Math.Atan(y2*(x1-x3)+y1*(x3-x2)+y3*(x2-x1)/(x2-x1)*(x1-x3)+(y2-y1)*(y1-y3))  //Formula (3)
Theta_angle = Theta * (180/Math.PI)

/* Pythagorean Theorem */
upper_side=math.Pow((x1-x2),2) + math.Pow((y1-y2),2)
lower_side=math.Pow((x3-x2),2) + math.Pow((y3-y2),2)
last_side=math.Pow((x1-x3),2) + math.Pow((y1-y3),2)

/*Result ∈ [0, 360]°*/
if( last_side < upper_side + lower_side ){ //Formula (4)
  if (Theta_angle>0){
    return Theta_angle          //A Zone
  }elseif(Theta_angle<0){
    return Theta_angle + 360; //D Zone
  }
}
elif(last_side > upper_side + lower_side){ //Formula (5)
    return Theta_angle+180;    //B Zone, C Zone
}
```

Fig. 7. Joint angle measurement code using trigonometry

Theta_angle의 범위를 -90~90도에서 0~360도로 변환하기 위해서는 Theta_angle이 예각인지 둔각인지 판별해야 하는데 이를 위해 upper_side, lower_side, last_side의 길이를 피타고라스의 정리를 사용하여 계산한다. 구해진 세 변의 길이가 수식 4를 만족하고 Theta_angle이 양수이면 구역 A에 포함되므로 Theta_angle의 값은 유지된다. 구해진 세 변의 길이가 수식 4를 만족하고 Theta_angle이 음수이면 구역 D에 포함되므로 Theta_angle에 360을 더한 값으로 변환된다. 구해진 세 변의 길이가 수식 5를 만족하면 Theta_angle은 구역 B, C에 포함되므로 Theta_angle의 값에 180을 더한 값으로 변환된다. 0~360도 값으로 변환된 Theta_angle은 사용자의 관절 각도와 목표한 자세의 관절 각도를 비교하는데 사용된다.

2. Determining posture using relative positions

Kinect Camera에서 절대적 위치를 사용한 사용자의 자세 판별은 개인의 체형 차이에 따라 정확도가 달라진다. 예를 들어 목표 자세를 취한 사람의 키가 180cm의 성인

인 반면, 사용자는 130cm 이하의 어린아이라면, 사용자의 올바른 자세에도 정확도가 떨어지게 된다. 이러한 문제점을 해결하기 위해 사용자의 체형에 맞는 자세의 기준점을 정하여 목표 자세와 상대적으로 비교하는 방법이 필요하다. 이러한 방법을 사용하면, 목표 자세의 관절 위치를 사용자의 상대적 관절 위치와 비교하여 정확하게 자세를 판별할 수 있다.

그림 8은 자세에 따른 관절의 상대적 위치를 분석하는 방법을 보여준다. (a)-1은 차렷 자세, (b)-1은 양팔을 벌린 자세를 나타낸다. 자세에 따라 관절의 상대적 위치를 분석하기 위해 (a)-2와 같이 차렷 자세는 사용자의 관절 위치 중 머리, 발, 어깨 혹은 팔꿈치의 위치를 사용하여 사각형이 그려진다. (b)-2와 같이 양팔을 벌린 자세는 사용자의 관절 위치 중 머리, 발, 양 손 끝의 위치를 사용하여 사각형이 그려진다. 마지막으로 (a)-3, (b)-3과 같이 사각형은 가로세로로 각 3등분으로 균일하게 나누어지며, 왼쪽 위를 1번으로 시작하여 가로 방향으로 차례로 번호를 매긴다. 나누어진 작은 사각형은 사용자의 자세에 따라 포함되는 사용자의 관절이 달라진다. 예를 들어 (a)-3의 사용자 오른쪽 팔꿈치는 4번 사각형에 포함되고 왼쪽 손목은 6번 사각형에 포함되지만, (b)-3의 사용자 오른쪽 팔꿈치는 1번 사각형에 포함되고 왼쪽 손목은 3번 사각형에 포함된다. 이와 같이 나누어진 사각형은 자세에 따른 관절의 상대적 위치를 판별하는데 사용된다.

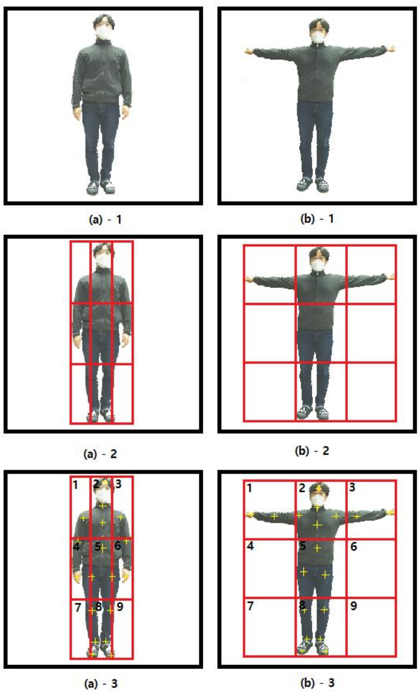

Fig. 8. Analysis of relative position of joints according to posture

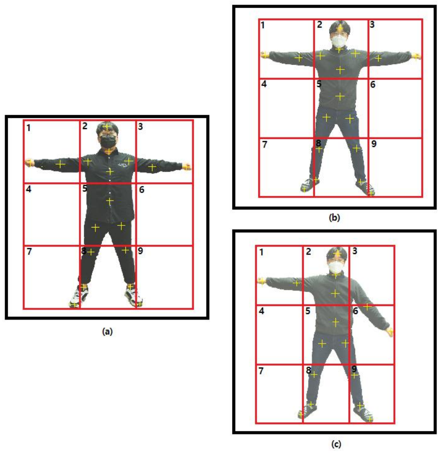

Fig. 9. Example of User Posture Analysis with Relative Position

그림 9는 관절의 상대적 위치를 사용하여 사용자 자세의 정확도를 판별하는 예를 보여준다. (a)는 목표 자세의 관절을 상대적 위치로 분석한 결과를 나타내는 것으로 4번과 6번 사각형을 제외한 나머지 사각형들에는 머리, 어깨, 무릎 등 목표 자세의 주요 관절들이 위치한다.

(b),(c)는 목표 자세를 따라하는 자세의 예를 나타낸다. (b)의 사용자가 취한 자세에 대한 관절의 상대적 위치를 (a)와 비교하면, 양팔의 팔꿈치 및 머리 어깨 등의 주요 관절들이 (a)와 동일하게 위치하며, 4번과 6번 사각형에는 어떠한 관절도 위치하지 않고 있다. 이를 통해 목표 자세와 사용자의 자세가 일치하는 것으로 판단한다. 이와 같은 방법으로 (c)의 사용자가 취한 자세에 대한 관절의 상대적 위치를 (a)와 비교하면, 사용자의 오른쪽 팔꿈치와 손, 머리 및 가슴의 위치는 (a)와 동일하게 위치하지만, 6번 사각형을 비교하였을 때에는 (a)는 관절이 위치하지 않았지만, (c)는 왼손 관절이 위치하고 있다. 또한, 목표 자세의 왼쪽 무릎은 (a)의 8번 사각형에 위치하고 (c)의 9번 사각형에 위치되어 각각 다른 사각형에 위치되는 것을 볼 수 있다. 이를 통해 목표 자세와 사용자의 자세가 일치하지 않은 것으로 판단한다. 이러한 판단 방법은 자세의 정확도를 계산하는데 사용되는데, 목표 자세에서 확인할 관절의 수와 사용자의 자세의 관절 중 목표 자세와 동일한 위치에 있는 관절의 수를 비교하여 정확도를 측정한다.

IV. Experiments and Results

본 논문에서는 사용자의 체형에 따른 자세를 정확히 판별하기 위해 삼각법을 사용한 관절 각도 측정 방법과 상대적 위치를 사용한 자세 판별 방법을 제안하였다. 제안하는

방법의 성능을 평가하기 위하여 사용된 실험 환경은 표 3과 같고 실험에 참여한 실험자 체형정보는 표 4와 같다.

Table 3. Testing Environment

TEST ENVIRONMENT	SYSTEM	VERSION
OS	windows 10 education	ver1909
CPU	ryzen 3700x	
GPU	rtx2070 super 8gb	
RAM	DDR4 16gb	
PROGRAM	visual studio 2019	ver16.6.3
Camera	Azure Kinect	v.1.4.0

Table 4. Experimental body shape information

USERS	GENDER	HEIGHT(cm)	WEIGHT(Kg)
USER1	male	175cm	90kg
USER2	female	160cm	58kg
USER3	female	167cm	48kg
USER4	male	175cm	78kg

본 논문에서 제안한 방법의 성능을 측정하기 위해 요가에서 사용되는 네 가지 자세를 목표 자세로 사용하였다. 그림 10은 실험에 목표 자세로 사용한 네 가지 요가 자세를 나타낸다.

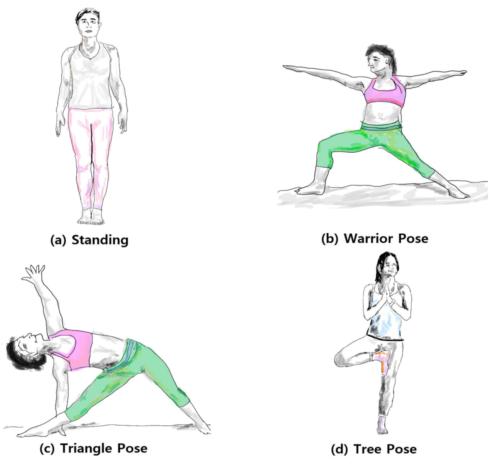

Fig. 10. Target Yoga Position

그림 11은 Azure Kinect를 사용하여 사용자의 요가 자세를 인식하고, 목표 자세와 비교하여 정확도를 산출하는 프로그램의 실행 화면을 보여준다. 왼쪽 창은 사용자의 실시간 영상을 보여주며, 오른쪽 창은 목표 자세를 보여준다. 프로그램 하단 좌측에 제안한 방법에 따라 계산된 정확도 결과를 출력하고, 중앙에 정확도에 따른 피드백, 우측에 목표 자세의 명칭을 보여준다.

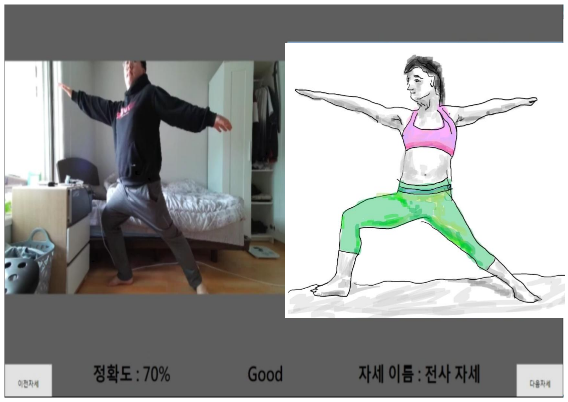

Fig. 11. Example of program execution

표 5는 삼각법을 사용한 관절 각도 측정방법을 사용하여 사용자 자세의 정확도를 실험한 결과이다. 실험에는 원형운동(Circumduction), 모음운동(Adduction), 벌림운동(Abduction)의 측정이 가능한 양 팔꿈치, 양 무릎, 척추 등 5개의 관절을 사용하였다. 사용자의 관절 각도와 목표한 자세의 관절 각도의 차이를 이용하여 실험에서 정확도를 계산하였고 이에 대한 수식은 아래와 같다.

$$Accuracy_{ang.} = \left(\sum_{i=1}^{5}\left(\left(1 - \frac{|G_i - M_i|}{180}\right) \times 100\right)\right) / 5\ (\%)$$

(Formula 6)

수식 6에서 i는 5개의 관절에 대한 인덱스이며, Gi는 목표 자세의 i-관절 각도를, M_i는 사용자의 자세의 i-관절 각도를 나타낸다. i-관절의 정확도는 1에서 오차율을 뺀 값에 100을 곱한 값으로 사용하며, $Accuracy_{ang.}$은 관절 5개에 대한 정확도 값의 평균을 사용한다.

Table 5. Posture Accuracy (%)

	(a)	(b)	(c)	(d)
USER1	95	92	95	93
USER2	97	94	82	86
USER3	97	97	78	92
USER4	100	97	91	94
AVERAGE	97.25	95	86.5	91.25

표 5를 보면, 목표 자세가 그림 10의 (a)인 경우 다리가 모이지 않았거나 팔을 완전히 펴지 않을 경우 양 팔꿈치의 각도 및 무릎에서 오차가 발생하며 척추의 구부정함 정도가 평균 약 3%의 오차를 발생시켰다. (b)의 경우, 무릎과 양팔을 편 모양에 따라 평균 5%의 오차가 발생하였다. (c)의 경우, 상급 요가 자세로 몸의 기울임에 따른 척추의 각도와 양팔 펼침의 정도가 사람마다 달라 많은 오차를 발생

시켰으며, 한쪽 손목이 발목에 가려져 팔꿈치의 각도를 정확하게 측정할 수 없는 한계로 인하여 평균 약 14%의 오차가 나타난다. (d)의 경우, 가슴 쪽 척추가 손목에 가려져 척추 각도를 정확하게 측정할 수 없는 한계로 인하여 평균 약 9%의 오차를 발생시켰다.

Table 6. Relative position accuracy (%)

	N = 20				N = 32			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
USER1	100	95	80	90	100	96	81	90
USER2	100	90	75	90	100	96	78	90
USER3	100	90	85	90	100	90	81	93
USER4	100	85	80	100	100	93	84	93
AVERAGE	100	90	80	92.5	100	93.75	81	91.5

표 6은 상대적 위치를 사용한 자세 판별 방법을 사용하여 사용자 자세의 정확도를 실험한 결과이다.

$$Accuracy_{RP} = \frac{M}{N} \times 100(\%) \qquad \text{(Formula 7)}$$

상대적 위치를 사용한 자세 측정에 대한 정확도는 수식 7과 같이 계산한다. N은 목표 자세에서 확인할 관절들의 개수로 최대 32개의 관절을 목표 관절 수로 설정할 수 있다. M은 사용자 자세의 관절 중 목표 자세와 동일한 위치에 있는 관절의 수로 목표 자세와 일치하는 자세를 취할수록 정확도가 높아진다. 목표 자세에서 확인할 관절의 수에 따른 정확도의 차이를 구별하기 위해 N은 20과 32로 설정하여 실험하였다. N이 20일 경우 목표 관절은 얼굴에 있는 눈, 코, 귀 5개의 관절을 머리 관절 1개로, 양손의 엄지와 끝 4개의 관절을 양손 관절 2개로 대체하고 쇄골 2개와 골반 1개의 관절이 제외되어 설정된다. N이 32일 경우, 목표 관절은 Azure Kinect에서 측정 가능한 모든 관절로 설정된다.

표 6을 보면, 목표 자세가 그림 10에서 (a)의 경우 사용자의 관절 위치가 상대적으로 계산한 사각형의 범위를 벗어나지 않는 자세이므로, N과 관계없이 모든 사용자의 정확도가 100%로 나타난다. (b)의 경우, 사용자의 하체 위치에 따라 골반의 위치가 달라져 사격형 위치의 불일치가 발생하여 N이 20일 경우 평균 90%의 정확도로, N이 32일 경우 평균 93.75%의 정확도로 나타난다. (c)의 경우, 상급 요가 자세로 사용자의 오른쪽 무릎의 위치와 왼손의 위치가 사각형의 경계선에 걸쳐지는 경우가 발생하여 N이 20일 경우 평균 80%의 정확도를, N이 32일 경우 평균 81%의 낮은 정확도로 나타난다. (d)의 경우, 상대적 위치를 사용해 구한 사각형의 좌우 폭이 좁게 계산되어 사용자의 손의 위치가 오차를 발생시켜 N이 20일 경우 평균 92.5%의 정확도로, N이 32일 경우 평균 91.5%의 정확도로 나타난다.

N이 20일 경우와 32일 경우를 전반적으로 비교하면, N이 32일 경우에 정확도 값이 높게 측정되는데 이는 각 관절에 대한 정확도는 N이 20일 경우에는 5%, N이 32일 경우에는 약 3%의 비율을 갖기 때문이다. 하지만 N이 32일 경우 머리와 양 손에 대한 관절의 수가 세밀하게 나누어져 손의 위치가 다른 관절과 겹치게 되는 경우 오차가 발생할 수 있다.

삼각법을 사용한 관절 각도 측정 방법은 5개의 관절(양쪽 팔꿈치, 양쪽 무릎, 척추)의 각도만을 사용하므로 다른 관절의 위치를 반영하지 못하며, 상대적 위치를 사용한 자세 판별 방법은 사용자의 관절의 각도를 피드백하지 못한다. 따라서 두 방법을 같이 사용하여 사용자에게 피드백하면 하나의 방법만으로 피드백하는 것보다 정확한 피드백을 할 수 있다.

Table 7. Integrated accuracy of angle and relative position (%)

	(a)	(b)	(c)	(d)
USER1	98	94	88	92
USER2	98	95	80	88
USER3	99	94	80	93
USER4	100	95	88	94
AVERAGE	98.75	94.5	84	91.75

표 7은 32개의 관절에 대한 상대적 위치를 사용한 자세 판별 방법과 삼각법을 사용한 관절 각도 측정 방법을 사용하여 사용자 자세 정확도를 계산한 결과이다. 두 방법을 같이 사용한 정확도는 아래의 식으로 계산한다.

$$Accuracy(\%) = \frac{1}{2}Accuracy_{ang.}(\%) + \frac{1}{2}Accuracy_{RP}(\%)$$

(Formula 8)

수식 8은 삼각법을 사용한 관절 각도 측정 방법의 결과와 상대적 위치를 사용한 자세 판별 방법의 결과를 50대 50의 가중치를 두어 사용자의 자세 정확도를 측정한 식이다.

표 7을 보면, 목표 자세가 그림 10의 (a)의 경우 관절들의 상대적 위치는 상대적으로 구한 사각형의 범위를 벗어나지 않지만, 양 팔꿈치와 무릎관절의 각도가 완전히 펴지지 않은 사용자의 자세가 피드백 되었다. (b)의 경우 양팔꿈치 및 무릎의 각도에 대한 피드백 뿐 아니라 상대적 위치를 통한 사용자의 하체 위치까지 피드백 되었다. (c)와

(d)의 경우 척추, 양 팔꿈치 및 무릎의 각도에 대한 피드백에 양손의 위치에 대한 상대적 위치까지 피드백 되었다. 이를 통해 각각의 방법으로 발견하지 못했던 오차를 정확도에 반영함으로 더 정확한 피드백이 사용자에게 제공됨을 알 수 있다.

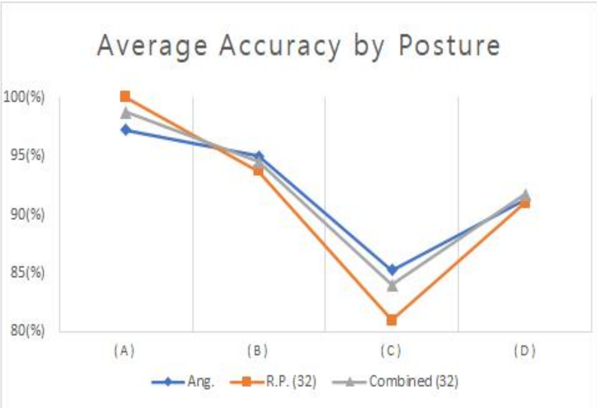

Fig. 12. Average accuracy by posture

그림 12는 삼각법을 사용한 관절 각도 측정 방법, 상대적 위치를 사용한 자세 판별 방법 두가지를 동시에 사용하는 방법에 따른 자세별 정확도를 비교하는 그래프로, (a)와 (d)의 자세에서는 상대적 위치를 사용한 자세 판별 방법이 높은 정확도를 피드백 하였고, (b)와 (c)의 자세에서는 삼각법을 사용한 관절 각도 측정 방법이 높은 정확도를 피드백 하였음을 보여준다. 두 가지를 통합하여 측정한 경우 각 측정 방법에서 측정할 수 없는 부분을 상호보완하여 사용자에게 피드백 함으로써 각 자세에 대한 정확한 피드백을 줄 수 있다.

V. Conclusions

본 논문에서 구현한 프로그램은 운동하는 사용자의 자세 정확도를 피드백으로 전달함으로 사용자의 운동 성능 향상을 도모하였다. 삼각법을 사용한 관절 각도 측정 방법과 상대적 위치를 사용한 자세 판별 방법을 동시에 사용함으로써 각 방법에서 측정할 수 없는 부분을 상호 보완한 결과를 사용자에게 제공하여 향상된 정확한 피드백을 전달하도록 했다. 카메라로 운동 중인 사용자의 모습과 표준 요가 자세를 동시에 화면에 보여주고 피드백과 함께 자세의 정확도를 실시간으로 보여줌으로써 점진적으로 정확한 자세를 만들어 갈 수 있도록 프로그램을 구현하였다.

ACKNOWLEDGEMENT

"This research was supported by the MISP (Ministry of Science,ICT), Korea, under the National Program for Excellence in SW (2019-0-01219) supervised by the IITP (Institute of Information & communications Technology Planning & Evaluation)" (2019-0-01219)

REFERENCES

[1] Yongwoon Chung, "The more you exercise, the more satisfied you are.", Dong-A Ilbo, https://www.donga.com/news/It/article/all/20200714/101963779/2

[2] Maeil Business News Korea, "Corona19's Fast-growing 'Hompt' Market", August, 2020 https://www.mk.co.kr/news/business/view/2020/08/792107/

[3] Trend Monitor, Can I get closer to exercise with home training, an exercise that always ends with mincing?, https://www.trendmonitor.co.kr/tmweb/trend/allTrend/detail.do?bIdx=1677&code=0501&trendType=CKOREA

[4] Microsoft, Microsoft Azure Kinect SDK Manual, https://docs.microsoft.com/ko-kr/azure/kinect-dk/

[5] Kyoung Shin Park. "Development of Kinect-Based Pose Recognition Model for Exercise Game," KIPS. Transactions on Computer and Communication Systems, vol. 5, No. 10, pp. 303-310, Oct. 2016.

[6] Jae-sic Seo, Min-ho Shin, Mi-kyeong Moon, "Fitness System for Correcting Real-time Posture Using Smart Mirror," Journal of IKEEE, vol. 23, No. 1, pp. 774-79, Mar. 2019.

[7] Back Seung-jin, Kuk Geun, Rhee Chan-ho, Kim Sung-hoon, Chae Ok-sam, "Motion Recognition System using Kinect," Journal of KIISE, Jeju, Korea, pp. 1851-1852, Jun. 2017.

[8] Da-won Kim, Hee-jo Nam, Seung-yeon Lee, You-kyung Haam, O-Seok Seo, HyungJune Lee, "Multi-User Home-Training Healthcare System Using Kinect Sensor and Wearable Devices," The Journal of Korean Institute of Communications and Informations Scieences, Vol. 44, No. 4, pp. 719 - 727, Apr. 2019.

[9] Joowon Lee, Kyungsoo Oh, "Yoga learning content by using Kinect," Journal of The HCI Society of Korea, Pyeongchang, Korea, pp. 221-223, Jan. 2012.

Authors

Jong Hoon Park entered the Division of Computer and Telecommunications Engineering at Yonsei University, Wonju, Korea, in 2018. He is currently a student in the Division of Computer and Telecommunications Engineering, Yonsei University. He is interested in artificial intelligence, application software, and system software.

Dae Han Sim entered the Division of Computer and Telecommunications Engineering at Yonsei University, Wonju, Korea, in 2018. He is currently a student in the Division of Computer and Telecommunications Engineering, Yonsei University. He is interested in artificial intelligence, optimization theory, and machine learning.

Young Pyo Jun received the B.S. degree in Computer Science from Kwangwoon University, Korea in 1985 and received M.S. and Ph.D. degrees in Computer Science from KAIST, Korea, in 1987 and 1994, respectively. Dr. Jun joined the faculty of the Division of Computer and Telecommunications Engineering at Yonsei University, Wonju, Korea, in 2020. He is currently an Associate Professor in the Software Division, Yonsei University. He is interested in artificial intelligence, application software, and system software.

Hongrae Lee received the B.S., M.S. and Ph.D. degrees in Computer Science from Yonsei University, Korea, in 2010, 2012 and 2019, respectively. Dr. Lee joined the faculty of the Division of Computer and Telecommunications Engineering at Yonsei University, Wonju, Korea, in 2020. He is currently an Assistant Professor in the Software Division, Yonsei University. He is interested in image processing, image comunication technology and computer vision system.