

<학술논문>

DOI <http://dx.doi.org/10.3795/KSME-A.2013.37.10.1219>

ISSN 1226-4873(Print)
2288-5226(Online)

시선위치 추적기법 및 3차원 위치정보 획득이 가능한 사지장애인 보조용 웨어러블 로봇 시스템

서형규* · 김준철* · 정진형* · 김동환**

* 서울과학기술대학교 기계시스템디자인공학과

Wearable Robot System Enabling Gaze Tracking and 3D Position Acquisition for Assisting a Disabled Person with Disabled Limbs

Hyoung Kyu Seo*, Jun Cheol Kim*, Jin Hyung Jung*
and Dong Hwan Kim**

* Dept. of Mechanical System Design Engineering, Seoul Nat'l Univ. of Science and Technology

(Received March 6, 2013 ; Revised August 3, 2013 ; Accepted August 12, 2013)

Key Words: Wearable Robot(웨어러블 로봇), Gaze Tracking(시선위치추적), Legs and Arms Disabled Person(사지장애자), 3D Position Identification(3차원 위치정보 확인)

초록: 눈 움직임만으로 물건을 집고자 하는 사지장애자를 위한 웨어러블 로봇을 소개한다. 이 로봇에서는 시선위치추적 알고리즘을 적용하여 파지하고자 하는 물체를 보는 동공의 움직임을 확인하여 물체의 2차원 정보를 구하고 물체까지의 깊이는 로봇 어깨위에 올려져 있는 Kinect라는 장치를 사용하여 구한다. 물체와 로봇, 그리고 카메라간의 좌표변환과 매칭을 통하여 최종 물체의 3차원 정보를 추출하고 이 정보는 로봇제어기인 DSP로 전송되어 물체를 잡을 수 있도록 제어하게 되어 궁극적으로 사용자가 물체를 정확히 잡을 수 있도록 한다.

Abstract: A new type of wearable robot is developed for a disabled person with disabled limbs, that is, a person who cannot intentionally move his/her legs and arms. This robot can enable the disabled person to grip an object using eye movements. A gaze tracking algorithm is employed to detect pupil movements by which the person observes the object to be gripped. By using this gaze tracking 2D information, the object is identified and the distance to the object is measured using a Kinect device installed on the robot shoulder. By using several coordinate transformations and a matching scheme, the final 3D information about the object from the base frame can be clearly identified, and the final position data is transmitted to the DSP-controlled robot controller, which enables the target object to be gripped successfully.

1. 서론

현재 사회적으로 급속한 고령화와 산업재해나 교통사고, 질병 등의 이유로 장애인의 수가 점점 증가하고 있는 것이 큰 문제점이다. 따라서 노인과 장애인들에 대한 복지과 서비스 산업이 점차 확대되고 있고, 특히 거동이 불편한 노인과 장애인을 위한 여러 유형의 서비스 제품들이 개발되고 있는데, 그 중에서도 사용자가 직접 입고 사용할 수 있는 웨어러

블(wearable) 로봇이 크게 증가 하고 있다.^(1,2)

이러한 웨어러블 로봇 분야에서 거동이 불편한 사람을 대상으로 만들어지는 제품들은 대부분 근력 지원형 로봇인데 이것은 사용자의 몸을 지탱 해주면서 일상적인 생활을 할 수 있도록 도와주는 역할을 한다.^(3,4)

하지만, 보통의 장애인과 달리 몸을 전혀 움직일 수 없는 전신마비 장애인들에게는 기존의 근력 지원형 웨어러블 로봇을 사용하기에는 한계가 있다. 이들은 몸의 근육을 사용할 수 없어서 근전도 신호(EMG)를 만들어 낼 수 없다. 또한 부분적인 신체의 움직임을 이용하지 못하는 문제 역시 기존 웨어러

† Corresponding Author, dhkim@seoultech.ac.kr

© 2013 The Korean Society of Mechanical Engineers

블 로봇을 사용하기에 어려운 점이 많다. 따라서 기존의 시스템과 다른 방법으로 재활 목적을 겸비한, 스스로의 의지에 따라 자신의 신체를 움직일 수 있는 웨어러블 로봇의 개발이 필요하다.

뿐만 아니라 기존에 개발되고 있는 많은 웨어러블 로봇들의 방식에는 여러 가지 문제점이 발생한다. 우선, 근육에서 발생하는 생체 신호를 감지하기 위하여 OP-AMP를 이용한 신호 증폭 센서를 이용하는데 이때, 증폭 과정에서 생기는 수많은 노이즈를 처리하고 분석하는데 많은 노력과 시간이 소요된다.⁽⁵⁾ 또한 사지마비 장애인처럼 사용자에게 따라 출력되는 신호의 정도가 다르기 때문에, 이를 처리 하는데 여러 어려움이 있었다. 따라서 카메라를 통한 비전방식을 이용하면 주변 빛 밝기 이외에 별다른 외부 조건에 대해 제한을 받지 않고 영상처리를 통해서 밝기에 따른 노이즈 또한 처리할 수 있기 때문에 사용자의 의도대로 동작시키는데 용이한 면이 있다.

본 연구에서 제안하는 로봇은 몸을 움직일 수 없는 사지마비환자들이 오직 눈만을 움직일 수 있다는 점에서부터 착안하였다. 환자 즉, 사용자는 로봇 팔을 입고 눈동자를 움직여 줌으로써 사용자가 원하는 방향으로 사용자의 팔에 부착된 기구를 움직이는 것이다. 구체적으로는 Gaze Tracking^(6,9) 알고리즘을 통해 카메라로 동공의 움직임을 추적하여 사용자가 응시하는 곳의 평면 좌표를 도출하고, 모션 감지센서인 Kinect⁽¹⁰⁾를 이용하여 응시 점에 대한 깊이를 측정하여 착용한 로봇팔로 환자가 물체를 집을 수 있도록 한다.

Gaze Tracking은 사람의 시선을 추적하기 위한 방법이며, 사람의 심리를 연구하기 위한 방법으로 주로 사용되어왔다. 현재 알려진 Gaze Tracking 기술로는 시선의 3차원 위치를 모르는 상태에서 사용자의 머리 움직임이나 시선의 이동에 따른 상대적 방향만을 측정하는 2차원 추적기술과 모델기반의 3차원 추적 기술이 알려져 있다.⁽⁶⁾ 2차원 추적기술의 예로는 전극을 눈 주위에 부착해 그 전위차를 이용한 EOG 방식⁽⁷⁾으로 눈의 신경근육을 이용하여 측정하는 방식이며, 신경근육이 약간의 움직임에도 민감하므로 오동작률이 높다. 다른 방식으로는 카메라 광축 주위에 부착된 광원과 광축에서 떨어진 광원을 교차로 사용하여 동공이 밝고 어둡게 나오는 현상을 이용해 동공을 추적하는 방식⁽⁸⁾도 있다. 또한 동공과 각막에 반사된

반사점을 이용해 시선을 추적하는 방식도 있다.⁽⁶⁾

본 연구에서는 사지마비 장애인의 경우를 대상으로 하여 기구의 탈부착이 용이하게 하고 사용자의 편의를 위하여 등 쪽으로 액츄에이터를 배치하지 않는 방안 등을 강구하였으며 전체 무게를 줄여 사용자가 피로감을 느끼지 않도록 설계하였다. 근전도 신호(EMG)를 사용하지 않고 장애자가 원하는 물체를 로봇 팔을 통하여 집을 수 있게 하기 위하여 물체의 2차원 정보를 동공의 움직임을 추적하여 이용하는 Gaze Tracking 기법을 활용하여 구하고 Gaze tracking으로 얻은 화면상의 물체를 모션 감지센서인 Kinect를 통하여 물체까지의 거리를 측정한다. 이러한 방법을 통하여 원하는 물체의 3차원 정보를 로봇에 입력하여 환자에 부착된 로봇을 원하는 물체까지 이동시켜 물체를 집을 수 있게 하였다.

2. 시스템 구성

2.1 기구부의 특징

Fig. 1의 로봇 구조도에서는 각 부위의 역할 및 기능이 상호 독립성을 유지하고 기구 간 간섭을 최소화 할 수 있게 설계하였다. 먼저 머리에 달린 고글형태의 Gaze tracking 카메라가 안면부에 부착되어 응시점의 x-y좌표 값을 읽어 들이며, Kinect 센서는 z값(물체와의 거리)을 읽어 측정한다. 로봇팔은 어깨, 팔꿈치에 세 개의 모터를 사용하여 3 DOF 시스템으로 팔이 움직일 수 있도록 하였다. 우선, 어깨에 달린 모터가 팔 전체를 들 수 있도록 설치되어 있으며, 팔꿈치 바로 위에 달린 반원 형태의 고리가 팔의 회전을 돕는다. 반원 고리 앞에 설치된 모터에 의해 하박을 들어올리며, 마지막으로 손에 달린 장갑이 물건을 잡는데 도움을 준다.

어깨부위에 달린 모터가 팔꿈치 위에 달린 링기어를 잡아당겨 팔을 들어 올리는 메커니즘을 구현하였다. 그리고 여러 사람들도 착용할 수 있게 길이 조

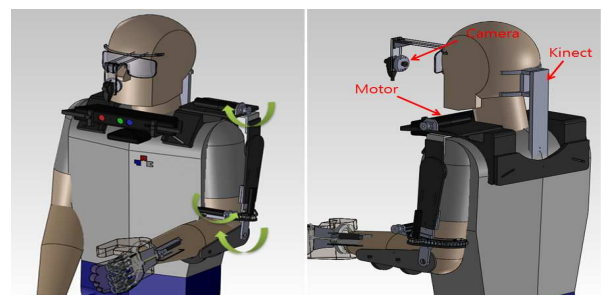


Fig. 1 Modeling equipped with robot components

정이 가능하도록 슬라이드 블록에 Hole을 일정한 간격으로 내서 핀을 삽입하면 1cm간격으로 길이 조정이 가능하다. 나머지 부위에도 slot Hole가공을 통해 필요한 경우 다양한 신체 크기에 따라 미세한 길이 조정이 가능하도록 설계하였다. Fig. 2는 웨어러블 로봇 기구부의 크기를 나타낸다. 사람의 체형에 따라 변동이 가능하도록 설계된 점이 특징이다.

팔꿈치부분의 가장 핵심은 링기어를 사용하여 팔의 Yaw방향으로 회전을 구현한 점이다. Fig. 3에서 보이는 수직으로 세워져 있는 모터에 연결된 스퍼기어가 C자 모형의 고리를 따라 회전하는 링기어를 회전시키면서, 하박에 부착된 가이드도 같이 회전하게 되면서 팔이 같이 따라 움직인다. 팔의 회전 중심과 기어의 회전 중심이 일치하도록 설계되어 회전이 자연스럽다. C자 모형으로 한 이유는 앞에 달린 와이어 모터를 안정적으로 지지해주면서, 입고 벗기 편하기 위해 제작하였다. 왼쪽의 모터가 와이어를 감게 되면 옆에 달린 가이드가 올라가면서 팔이 자연스럽게 올라간다. 그리고 팔을 안정적으로 들어올리기 위한 U자 모양의 가이드가 설치되어 있다.

2.2 Gaze Tracking

본 연구의 핵심은 Gaze Tracking을 이용하여 사지마비 장애인이 눈을 움직임으로 기준 축에서 물체의 좌표를 구하여 착용한 로봇 팔을 제어하

여 원하는 물건을 잡는 것이다. 카메라와 Kinect 센서를 통해 물체의 영상을 취득하고 영상처리를 하여 MCU(마이컴제어기)에 물체의 좌표를 전달하도록 시스템을 구성하였다. Fig. 4는 본 연구에서 수행한 Gaze tracking을 실현하는 시스템이다.

Gaze Tracking을 위해서는 안정적으로 눈 영상을 취득하는 것이 중요하며 이를 위해 카메라와 얼굴을 고정시킬 수 있게 Fig. 5와 같은 시선추적기를 설계하였다. 지지대에 카메라 A를 고정하였고 카메라 내부에 가시광선필터를 부착하여 외부 빛에 의한 노이즈를 차단하여 영상처리 시 용이하도록 하였다. 또한 사람의 눈에 무해한 적외선 발광부 B를 이용하여 Fig. 6과 같이 가시광선 필터에 영향을 받지 않고 홍채와 동공영역을 구분할 수 있도록 하였다.

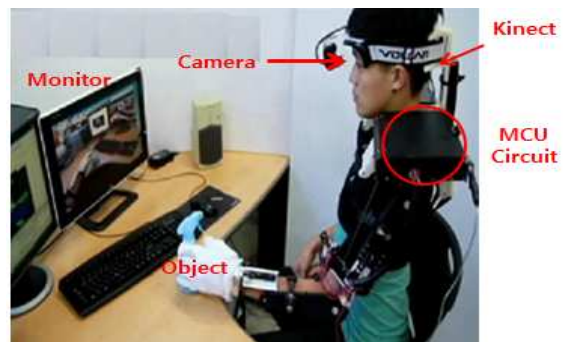


Fig. 4 Gaze Tracking System

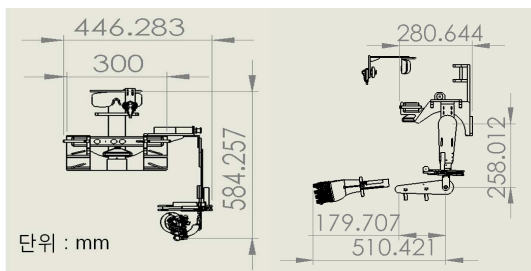


Fig. 2 Dimension of robot mechanism



Fig. 5 Camera and infrared light for Gaze tracking

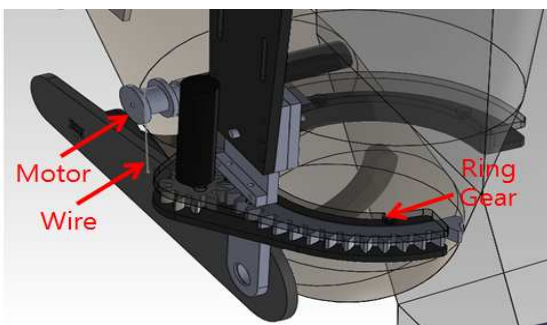


Fig. 3 Side view of elbow



(a) Eye image by regular camera (b) Eye image by gaze tracking camera

Fig. 6 Eye image taken from camera

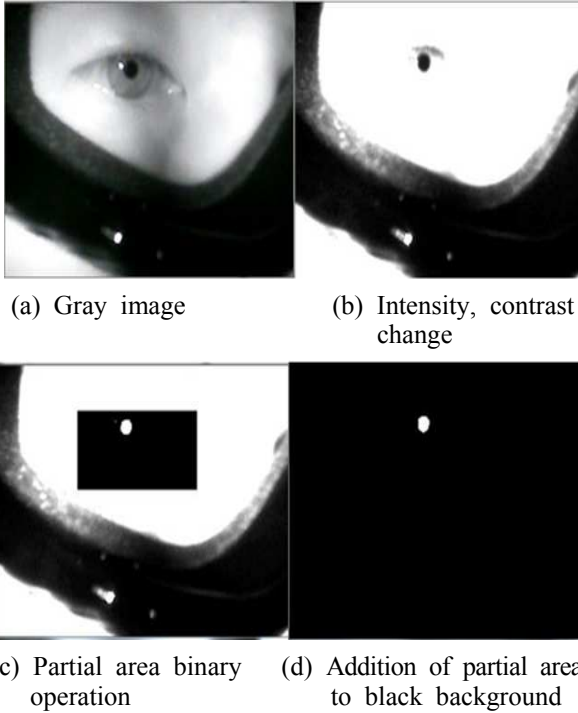


Fig. 7 Process for extracting eyeball area



Fig. 8 After center point detection

소프트웨어를 통해 카메라에서 받은 눈 영상을 영상처리를 이용하여 Fig. 7과 같이 (a)~(d)까지 동공영역만을 추출하여 중심점을 얻었다. 하지만 실제 눈 영상에 동공의 중심점을 표시해 보면 그림 8과 같이 동공을 나타내는 원의 위치가 일치하지 않는 오차가 발생하는데 이는 영상처리과정 때문에 생기는 원 영상의 변형 때문이다. 이러한 문제를 해결하기 위해 별도의 보정 알고리즘을 구성하였다.

Fig. 9(a)의 눈 영상에서 경계선(edge)를 검출하면 Fig. 9(b)처럼 나타나며 동공의 경계부분만을 확대하면 Fig. 9(c)의 빨간색 원형영역이라고 가정할 수 있다. Fig. 9(c)에서 보정 전 중심좌표를 (x_c, y_c) 이라 하면 빨간색 원형영역까지의 위쪽 픽셀 개수 A와 아래쪽 픽셀 개수 B를 구할 수

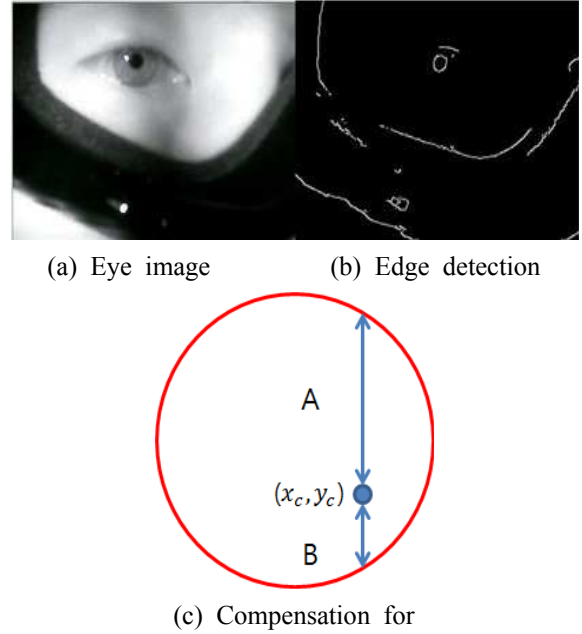


Fig. 9 Eye image processing sequence

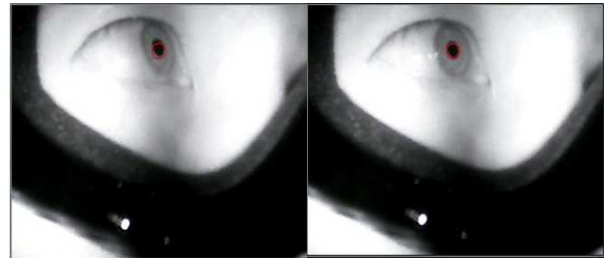


Fig. 10 Center of pupil detection

있으며 이를 식 (1)에 대입해 보정된 중심점의 y 좌표 y'_c 를 구할 수 있다.

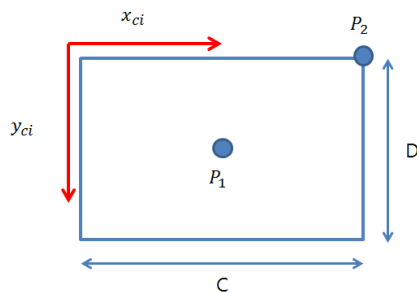
$$y'_c = y_c + \frac{(A+B)}{2} \tag{1}$$

같은 방식으로 x좌표도 보정하였다. 그 결과 실제 검출된 중심점은 Fig. 10과 같다.

Fig. 11과 같이 눈동자가 모니터의 중앙부분 A와 좌측 상단 부분 B를 응시 하였을 때의 눈 영상을 각각 Fig. 11(a)와 (b)에 나타내었고 이 영상을 이용하여 시선 mapping을 위해 동공의 중심좌표를 이용하는 Calibration 과정을 거친다. A부분을 바라보았을 때 카메라 상의 동공 중심좌표는 $P_1 = ({}^cP_{1x}, {}^cP_{1y})^T$ 이고, B를 바라보았을 때의 동공 중심좌표는 $P_2 = ({}^cP_{2x}, {}^cP_{2y})^T$ 이다. 이때 P_2 는 카메라 좌표에서 좌측 상단이 아니라 반대쪽인 오른쪽 상단에 해당한다. 이 두 개의 좌표를 이용하여 모니터의 전 영역을 바라보았을 때 동



(a) Center of monitor (b) Left top



(c) Pupil range

Fig. 11 Range of eye motions

공의 범위를 (c)의 사각형이라고 가정할 수 있고 다음의 수식을 이용하여 사각형의 가로 세로인 C와 D를 구할 수 있다.

$$C = 2({}^cP_{2x} - {}^cP_{1x}) \quad (2)$$

$$D = 2({}^cP_{2y} - {}^cP_{1y}) \quad (3)$$

Calibration 과정을 거친 후 Fig. 12와 같이 Kinect로부터 입력 받은 영상에 시선을 mapping시켜야 한다. 이는 카메라 영상과 Kinect 영상의 비율을 계산하여 구할 수 있다.⁽⁹⁾

$$x_{ki} = \frac{x_{ci}I}{C} \quad (4)$$

$$y_{ki} = \frac{y_{ci}H}{D} \quad (5)$$

식 (4)와 식 (5)의 x_{ki} , y_{ki} 는 목표 지점을 쳐다보았을 때의 Kinect 영상에서의 픽셀좌표이고, I와 H는 Kinect의 전체영상의 가로와 세로 픽셀의 크기이다.

카메라와 눈까지의 거리를 가깝게 하거나 더 높은 화소수의 카메라를 이용하면 동공이 모니터에 대응되는 범위가 넓어지고 더욱 정밀하게 대응시킬 수 있다.

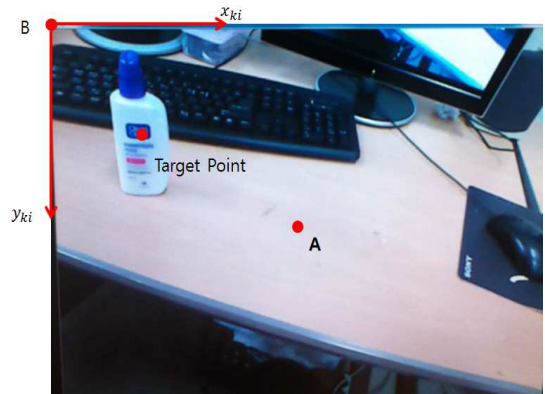


Fig. 12 Image taken by Kinect

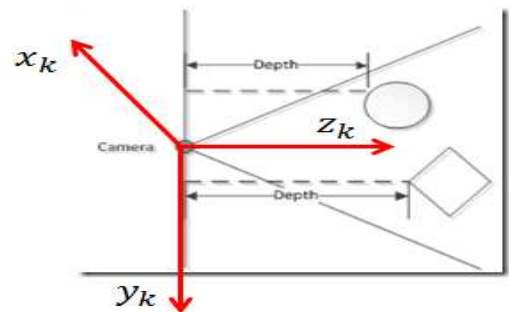


Fig. 13 Depth measurement method of Kinect



Fig. 14 Kinect coordinate system

2.3 좌표 측정

Kinect에서의 물체까지의 거리측정의 경우 Fig. 13와 같이 실제 물체와 Kinect까지의 최단거리가 아닌 x_k 나 y_k 좌표에 상관없이 z_k 값에 따라 거리가 측정된다.⁽¹⁰⁾ 이런 방식의 장점은 좌표를 측정할 때 어느 한 축으로 고정시키면 공간좌표를 측정하기가 수월해지기 때문이다.

Kinect에서 거리 데이터를 측정하면 계산을 통해 Kinect에서 목표지점까지의 거리 좌표를 구할 수 있다. Kinect 렌즈의 수평 시야각 43°와 수직 시야각 57°를 이용하여 거리를 계산하며, 수평시야각은 Fig. 14에서 Kinect영상의 x_{ki} 축 좌표에서 실제 x_k 좌표를 계산할 때, 수직시야각은 y_{ki} 축 좌표에서 실제 y_k 를 계산할 때 사용한다.⁽¹⁰⁾

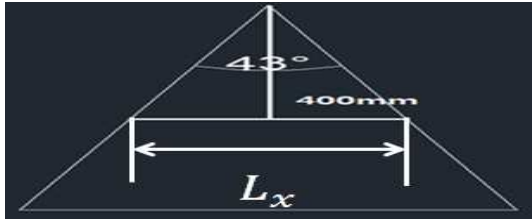


Fig. 15 The relationship between horizontal angle of view and image by Kinect

Fig. 15에서 L_x 는 Kinect 영상에서 z_k 가 400mm인 부분의 전체 x_{ki} 픽셀 수에 해당하는 실제 거리이다. Kinect 영상의 크기가 640*480 일 때, L_x 를 이용하여 한 픽셀 당 실제 거리를 계산하면 다음과 같다.

$$L_x = 2 \times 400 \tan(43^\circ / 2) \quad (6)$$

$$L_x / 640 = 0.49(mm) \quad (7)$$

식 (6), (7) 과 같은 방식으로 y_{ki} 좌표의 한 픽셀 당 길이를 구하면 다음과 같다.

$$L_y = 2 \times 400 \tan(57^\circ / 2) \quad (8)$$

$$L_y / 480 = 0.905(mm) \quad (9)$$

따라서 사용자가 바라본 물체의 실제 좌표는 다음과 같다.

$${}^k P_x = (z_k + FocalLength) \times \tan(43^\circ / 2) \times (320 - x_{kinect}) / I \quad (10)$$

$${}^k P_y = (z_k + FocalLength) \times \tan(57^\circ / 2) \times (y_{kinect} - 240) / H \quad (11)$$

$${}^k P_z = z_k + FocalLength \quad (12)$$

식 (10)과 (11), (12)에서 *FocalLength*는 Kinect 렌즈의 초점거리를 나타낸다. 이것을 바탕으로 영상에서의 단위를 실제 우리가 사용하는 단위로 환산해 줄 수 있다.

Fig. 16은 실제 사용자가 착용하는 로봇과 Kinect의 좌표계를 보여준다. 사용자의 왼쪽 어깨는 로봇의 어깨와 고정되어 좌표계의 원점에 위치하여 움직이게 되고 Kinect는 로봇 팔이 부착된 왼쪽 손이 닿는 범위의 영상을 얻기 위해 오른쪽 어깨 위에 pitch와 yaw에 대해 일정 각도만큼 회전한 후 부착되어 있다. 이 때문에 어깨의 좌표축과 Kinect 좌표축이 달라지는데, 이를 위해서 Kinect의 위치에서 바라본 물체 P 의 좌표인 ${}^k P$ 를 어깨의 좌표축을 기준으로 한 ${}^s P$ 좌표로 변환하는 작업을 수행해야 한다.

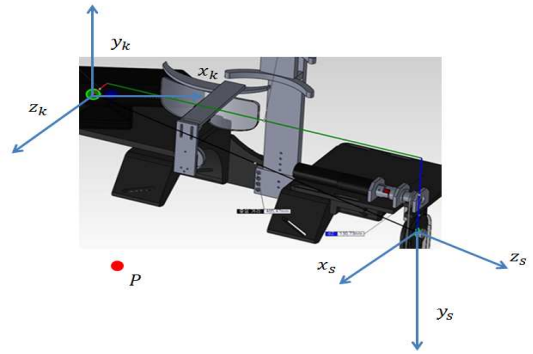


Fig. 16 Coordinates system

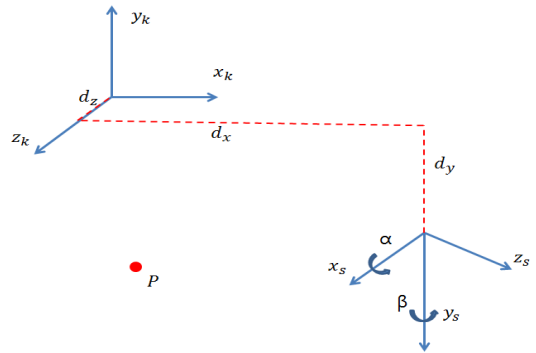


Fig. 17 Coordinate transformation process

Fig. 17처럼 좌표변환 행렬인 ${}^s T_k$ 를 이용하여 Kinect에서 물체까지의 좌표를 어깨에서 물체까지의 좌표로 변화해야 한다. 따라서 변환과정은 다음과 같다.

$${}^s P = {}^s T_k {}^k P \quad (13)$$

$${}^s P = \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^k P \quad (14)$$

여기서 (n, o, a) 는 회전 변환 값이고 (d_x, d_y, d_z) 는 두 좌표 중심간 직선이동 값이다. 그리고 α 와 β 는 두 좌표사이의 회전각도 값이다. 두 축간의 좌표변환 행렬은 다음과 같이 구해진다.

$${}^s P = \begin{bmatrix} \cos\beta & 0 & \sin\beta & d_x \\ \sin^2\beta & \cos\alpha - \cos\beta\sin\beta & d_y & \\ -\cos\alpha\sin\beta & \sin\alpha & \cos\alpha\cos\beta & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^k P \quad (15)$$

식 (15)를 통하여 최종적으로 구한 ${}^s P$ 는 PC 프로그램에서 통신을 통해 DSP로 전송하고, DSP에

서는 이 데이터를 이용하여 역기구학(Inverse Kinematics) 연산을 통해 계산한 각도만큼 각 관절의 모터들을 회전하여 팔을 이동시킨다.

2.4 제어 시스템

로봇의 구동은 DSP내에서 모든 것이 이루어지는 것이 아니라, 여러 가지 외부 장치와의 통신을 기반으로 하여 이루어진다. 즉, 외부 기기(Kinect)로 부터는 필요한 데이터를 전송 받고, 다른 기기로 필요한 정보를 전송하게 된다. 우리 로봇에서는 SCI(Serial Communications Interface) 통신을 기반으로 통신이 이루어진다. DSP는 PC 소프트웨어와 Interface가 이루어지며 Gaze Tracking과 Kinect로 구해진 삼차원의 위치좌표를 전달 받는데 사용된다.

최종적으로 전달받은 좌표 데이터는 모터에 요구되는 각도 값으로 환산하여 모터를 회전시켜 물건을 집게 된다. 우리는 Gaze Tracking과 Kinect를 통하여 어깨 축부터 물체 P까지의 좌표 sP 를 생성한 후 통신을 통해 DSP로 전송하게 된다. 이때 DSP 내에서는 전달받은 데이터에 대하여 로봇의 역기구학을 계산하여 각 모터에 필요한 변위 값을 명령하게 된다.⁽¹¹⁾

설계한 로봇 기구의 D-H 파라미터값⁽¹¹⁾을 Fig. 19와 같은 로봇의 좌표 시스템에 적용하고 좌표 변환행렬 (Transformation matrix)을 계산한다.



Fig. 18 Communication system

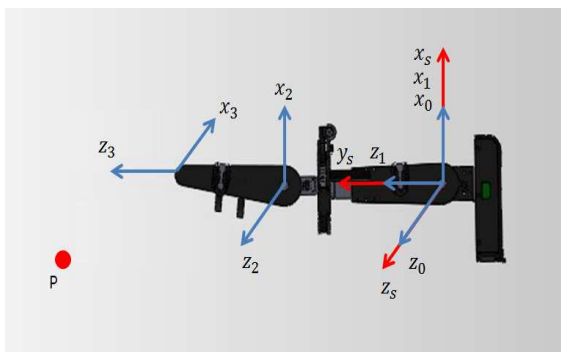


Fig. 19 Shoulder coordinate system

Fig. 19에서 End - effector의 좌표인 (x_3, y_3, z_3) 가 물체 P에 위치하도록 3개의 로봇 팔을 제어하면 물체를 잡을 수 있게 된다. 각 축의 좌표 변환행렬을 이용하여 어깨 축 좌표로부터 최종 End-effector 좌표간의 변환행렬이 주어지면 $({}^sT_3)$ 이를 만족하는 각 좌표 간 변환행렬 $T_1(\theta_1), T_2(\theta_2), T_3(\theta_3)$ 로부터 회전각도 $(\theta_1, \theta_2, \theta_3)$ 를 계산한다. 계산 결과는 아래와 같이 표현된다.

$${}^sT_3 = T_1 T_2 T_3 \tag{16}$$

역기구학을 구하면 다음과 같다.

$$\theta_1 = \sin^{-1} \left(\frac{a_3^2 - P_x^2 - {}^sP_y^2 - {}^sP_z^2 - d_1^2 - d_2^2 + 2d_1 {}^sP_x}{\sqrt{(2d_2 {}^sP_x)^2 + (-2d_2 {}^sP_y)^2}} \right) - \beta \tag{17}$$

$$\beta = \tan^{-1} \left(\frac{-2d_2 {}^sP_y}{2d_2 {}^sP_x} \right) \tag{18}$$

$$\therefore \theta_3 = \sin^{-1} \left(\frac{{}^sP_y \cos \theta_1 - {}^sP_x \sin \theta_1 - d_2}{a_3} \right) \tag{19}$$

$$\theta_2 = \sin^{-1} \left(\frac{d_1 - {}^sP_z}{a_3 \cos \theta_3} \right) \tag{20}$$

3. 실험 결과

실제 과제 동작과 함께 Gaze Tracking을 이용하여 모니터 상에서 실제 시선이 어느 정도로 정확하게 mapping 되는지를 실험하였고, Kinect센서의 깊이와 그것을 통한 삼차원좌표를 반복 측정하여 Table 1에 실험결과를 정리하였다.

Table 1 Gaze tracking test

실험종류	거리(cm)	평균오차(cm)	실험횟수	
Gaze Tracking	40	0.2	100	
	50	0.8	100	
Kinect를 통한 공간좌표	x	50	1.2	100
	y	50	1.1	100
	z	50	0.11	100

Table 2 Grip motion test

분류	실험 조건	End-effector 도달오차(cm)	End-effector 파지성공률(%)	실험 횟수
물체 위치	50cm	3.8	80	10
	40cm	3.4	82	10
	30cm	3.2	85	10

Gaze Tracking에서 거리가 50cm 떨어질 경우 오차가 많이 나는 걸 볼 수 있는데, 이것의 가장 큰 이유는 사용자의 흔들림이다. 흔들림은 기구가 동작하면서 많이 일어나며 모니터에 사용자의 시선을 mapping시키는 Calibration 과정에서 오차를 발생시킨다. 장애자의 경우 실제 오차가 다소 커질 가능성이 있다고 판단된다. 이러한 오차의 발생에도 불구하고 실제 물체의 파지 동작에는 심각한 문제가 발생하지 않았으며 높은 파지 성공률을 가지는 것으로 확인되었다.

또한 Kinect를 통하여 삼차원 좌표 측정 시, z 좌표의 오차에 비해 x, y의 오차가 크게 발생하는 것을 볼 수 있는데 이것은 좌표 측정 과정에서 x, y의 데이터는 Gaze Tracking과 z 값을 이용하여 계산되기 때문에 두 데이터의 오차가 쌓여서 더 큰 오차가 발생 되는 것이다.

실제 물건을 집는 동작의 경우 일상적인 생활에서 사람이 물건을 집는 상황을 고려하여 테스트 하였다. 집는 물체의 기준은 컵, 과일, 휴대폰 등과 같이 어느 정도의 크기를 가지는 것으로 정하였고, 실제 테스트에선 80*80*80(mm)의 크기를 가진 물체를 기준으로 손바닥의 가운데 점이 물체에 얼마나 접근하는지 측정하였다. 실험에서는 사지를 사용하지 못하는 대상인을 찾기 어려워 정상적인 사람을 대상으로 실험을 수행하였고 장애자의 경우는 본 실험 결과가 약간의 차이가 발생할 가능성이 있다. Table 2에 실험결과를 정리하였다.

우선, 물체 위치 변화 실험은 어깨 축을 기준으로 $y = 30cm, z = 0cm$ 으로 구속하고 x좌표만을 변화시키면서 실행하였다. Table 2의 실험에서 물체의 위치가 멀어질수록 파지 성공률이 떨어지

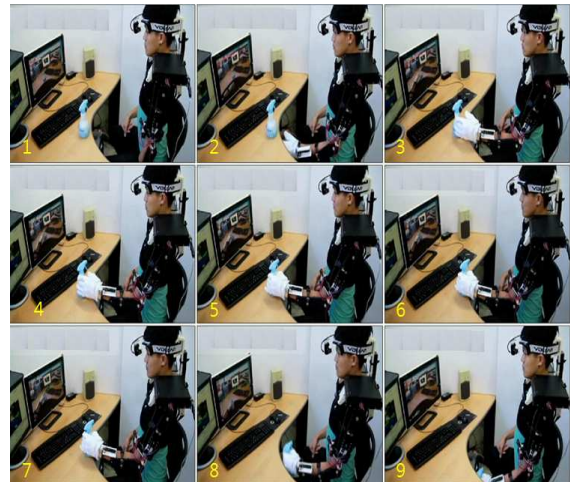


Fig. 20 Robot operation

는 것을 볼 수 있는데 이러한 현상의 원인 또한 사용자 신체의 흔들림으로 인하여 나타나는 것이다. 거리가 멀어짐에 따라 사용자의 몸의 흔들림이 커지게 되고, 좌표계가 틀어지면서 오차가 발생하였다.

Fig. 20은 제작한 웨어러블 로봇을 사람이 실제로 착용하여 동작하는 화면이며 원활하게 동작하는 것을 확인할 수 있다. 사람이 집을 수 있을 만큼의 영역내로 로봇이 구동하긴 하지만 오차는 존재하였다. 이 오차들에 대하여 여러 가지 원인을 알아보았다.

첫째, 신체의 흔들림이다. 기존의 웨어러블의 경우 로봇의 몸체를 벽이나 기둥에 고정시켜 흔들림이 없도록 하는데 반해 우리 로봇은 사용자가 직접 입기 때문에 로봇이 움직일 때 발생하는 모션의 흔들림으로 인하여 오차가 발생할 수 있다. 이 과정에서 기준좌표계가 흔들리면서 오차가 발생하게 된다.

둘째, 센서의 오차이다. Kinect의 경우 적외선 센서처럼 깊이 값을 측정하기 때문에 매번 일정하지 않고 빛의 조건에 따라 1mm에서 2mm정도의 오차가 발생한다.

4. 결론

본 연구에서는 사지를 사용하기 어려운 장애자를 위하여 고안된 웨어러블 로봇을 개발하였다. 사용자의 눈동자의 움직임만을 비전처리를 통하여 파지하고자 하는 물체를 정확하게 파악하고, 이를 로봇의 목표구동 명령으로 처리하여 사용자

의 팔을 물체까지 움직여 물체를 집을 수 있도록 하였다.

비전처리를 이용한 사용자 의도를 파악하는 방법의 장점은 물체 정보처리가 비교적 간단하게 이루어질 수 있다는 것이다. 적외선센서를 사용하여 외부 빛에 의한 노이즈를 최소화 하여 훨씬 신뢰성 있고 간단하게 구현하였고, 물체를 보는 사용자의 눈동자의 움직임을 파악하여 물체의 2차원 위치 정보를 구하고 Kinect를 사용하여 물체까지의 거리 정보를 구하여 최종적으로 파지하고자 하는 물체까지 로봇 팔을 이동시켜 사용자가 물체를 집을 수 있도록 제작하였다. Gaze tracking 기법을 적용하였으며, 물체를 보는 사용자의 동공의 움직임을 정확히 측정하여 물체의 2차원 정보를 구하였다. 또한 물체 위치정보를 보다 명확히 하기 위하여 사용자가 모니터를 주시하게 하여 집고자 하는 물체를 가리키게 하였다.

이러한 웨어러블 로봇에서 가장 중요한 부분은 사용자와의 호환성이다. 로봇의 외형을 인체공학적인 설계를 통하여 착용감과 동작 등을 개선하고, 신체의 구조를 자세히 연구하여 로봇과 신체의 결합력을 높였고 다양한 사용자가 착용할 수 있도록 신체구조에 따라 조정이 가능하도록 설계하였다. 웨어러블 로봇의 실험 결과 사용자의 흔들림으로 인한 물체 파지 및 물체위치 계산에 약간의 오차가 발생하였으며 이를 극복할 수 있는 기구의 보완과 물체 움직임을 기반한 제어 알고리즘의 도입이 필요하다.

참고문헌

- (1) Nam, M. K., 2011, "An Analysis on the Design Development Status of Wearable Goos for Measuring the Physical Function and Position," *Korean Society of Design Culture*, Vol. 17, No. 2, pp. 161~171
- (2) Norisugu, T., Daisuke, S. and TakaiwaM., 2009, "Development of Power Assist Wear Using Pneumatic Rubber Artificial Muscles," *Robotics and Mechatronics*, Vol. 21, No. 5, pp. 607~608
- (3) Kiguchi, K., Tanaka, T. and Fukuda, T., 2004, "Neuro-Fuzzy Control of a Robotic Exoskeleton With EMG Signals," *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 4, pp. 481~490
- (4) Nalto, J., Obinata, G., Nakayama, A. and Hase, K., 2007, "Development of a Wearable Robot for Assisting Carpentry Workers," *ARS*, Vol. 4, No. 4, pp. 431~436
- (5) Kundu, A. S., Mazumder, O. and Bhaumik, S., 2011, "Design of Wearable, Low Power, Single Supply Surface EMG Extractor Unit for Wireless Monitoring," *International Conference on Nanotechnology and Biosensors IPCBEE*, Vol. 25, pp. 69~74
- (6) Park, J., Kwon, Y.-M. and Shon, K., 2006, "Gaze Tracking System Using Feature Points of Pupil and Glints Center," *The Korean Society of Broadcast Engineers*, Vol. 11, No. 1, pp. 80~90
- (7) Barea, R., Boquete, L., Mazo, M., Lopez, E., 2002, "Wheelchair Guidance Strategies Using EOG," *Journal of Intelligent and Robotic System*, Vol. 34, pp. 279~299
- (8) Park, K. R., 2004, "Gaze Detection System by IR-LED based Camera," *KICS*, Vol.29, No. 4C, pp. 494~504
- (9) Lee, W. O., Lee, E. C., Park, K. R., Lee, H. K., Park, M., Lee, H. K. and Hong, J. W., 2009, "A Study on Controlling IPTV Interface Based on Tracking of Face and Eye Positions," *KICS*, Vol. 35, No. 6, pp. 930~939
- (10) <http://en.wikipedia.org/wiki/Kinect>
- (11) Saeed B. Niku, 2011, Introduction to Robotics, 2nd Ed, Prentice Hall, pp. 33~132
- (12) <http://blogs.msdn.com/b/msroboticsstudio/archive/2011/11/29/kinect-for-robotics.aspx>