

사용자의 동작인식 및 모사를 구현하는 로봇시스템 설계

최용욱^a, 윤상현^a, 김준식^a, 안영석^a, 김동환^{a*}Robot System Design Capable of Motion Recognition and
Tracking the Operator's MotionYonguk Choi^a, Sanghyun Yoon^a, Junsik Kim^a, YoungSeok Ahn^a, Dong Hwan Kim^{a*}^a Department of Mechanical System Design Engineering, Seoul Nat'l Univ. of Science and Technology,
232, Gongreung-ro, Nowon-gu, Seoul 01811, Korea

ARTICLE INFO

Article history:

Received	5	June	2015
Revised	31	October	2015
Accepted	4	December	2015

Keywords:

Motion recognition image processing
3D depth sensor camera
Motion following
Skeleton joint information

ABSTRACT

Three dimensional (3D) position determination and motion recognition using a 3D depth sensor camera are applied to a developed penguin-shaped robot, and its validity and closeness are investigated. The robot is equipped with an Asus Xtion Pro Live as a 3D depth camera, and a sound module. Using the skeleton information from the motion recognition data extracted from the camera, the robot is controlled so as to follow the typical three mode-reactions formed by the operator's gestures. In this study, the extraction of skeleton joint information using the 3D depth camera is introduced, and the tracking performance of the operator's motions is explained.

1. 서론

기존의 정형화된 작업을 수행하는 산업용 로봇으로부터 사용 목적에 맞는 다양한 유형의 비산업용 로봇이 지속적으로 출현하고 있다. 로봇과 인간이 상호 교감하여 사용자인 사람에게 즐거움을 주는 엔터테인먼트 로봇^[1]이 각광을 받고 있으며 인간을 모사하는 형태로까지 발전하고 있다. 엔터테인먼트 로봇은 메카니즘의 구조, 기능, 성능에 따라 다양한 형태를 가지고 있다. 예로 노인을 대상으로 하는 엔터테인먼트 로봇의 경우 직접 몸을 움직여 사용할 수 있도록 제작되어 즐거움과 동시에 운동 효과를 가져다줄 수 있다^[2]. 또한, 귀여운 모습의 로봇이 여러 동작을 하게 된다면 노인의 심적 우울증을 해소시킬 수 있다^[3].

본 연구에서는 펭귄 모양의 로봇 기구를 설계하여 사용자에게

친근감을 주게 로봇을 제작하였다. 이 로봇에서 중점을 두고 있는 부분은 사람의 동작을 인식하고 인식된 동작에 따라 로봇이 각기 다른 반응을 하도록 제작하였다. 또한 펭귄 모양의 로봇의 전체 시스템을 소개하고 로봇의 몸체에 해당하는 하드웨어와 영상처리를 진행하는 소프트웨어를 제안하고 로봇이 작동하는 동작에 대한 알고리즘을 설명한다. 본 연구에서는 기존의 감성 로봇 PARO^[4]와 Rabi^[5]와는 다르게 비교적 간단한 구조로 설계하였으며 3차원 깊이 측정 카메라^[6]를 사용하여 사용자의 동작을 인식하여 따라하거나 사용자의 이동을 감지하여 따라올 수 있게 제작되었다.

기존에 연구된 이동감지 방식^[7]과 달리 동작인식을 위하여 골격 이미지를 형성하여 인체 부위별 관절을 연결한 이미지^[8]로 재구성하여 동작 모드를 정확히 인식하는 방안을 제시하였다. 또한 거리 측정데이터의 오차를 보상하기 위한 방법을 사용하여 실제 거리와

* Corresponding author. Tel.: +82-2-970-6362

Fax: +82-2-974-8270

E-mail address: dhkim@seoultech.ac.kr (Dong Hwan Kim).

측정거리간의 오차를 최소화 하여 로봇을 효과적으로 작동하였다. 제작된 로봇을 사용하여 사용자의 모션을 인지하고 따라하는 실험을 통하여 로봇의 효용성 및 제한한 방안의 유효성을 검증하였다.

2. 로봇 시스템

2.1 시스템 개요

전체 로봇의 시스템 구조는 Fig. 1과 같다. 3D 거리 측정 카메라인 Xtion Pro Live와 PC의 역할을 수행하는 임베디드 보드, 음성 인식을 담당하는 음성 모듈, 데이터를 받아들여 모터를 제어하는 마이크로프로세서(DSP-28335)로 구성되어 있다. Xtion Pro Live는 USB 2.0으로 연결되는 3D 거리 측정 카메라이기 때문에 실제적인 영상처리를 진행하는 PC의 역할은 임베디드 보드인 PCM-3363에서 처리하였다. 처리된 데이터는 RS-232 시리얼 통신을 통해 DSP-28335로 전송된다. 전송된 데이터를 통하여 로봇의 각 관절에 위치한 모터를 제어하여 로봇의 동작을 수행한다. 사용된 음성 모듈에서는 10가지 음성을 인식하며 각각의 음성에 따라서 동작이 제어되고 이 데이터는 마이크로프로세서로 전송된다. Fig. 2는 로봇이 동작되는 흐름도를 나타낸다. 로봇이 실행되면 Xtion Pro Live나 음성 모듈로부터 여러 데이터를 받게 된다. 동작 인식 데이터를 받기 위해서는 Xtion Pro Live에서 관절 데이터를 전송받으며 이를 통해 사람을 따라가는 추적 모드, 사람의 동작을

따라하는 모션 미러링(motion mirroring) 모드, 모든 동작이 중지되는 정지 모드로 나뉘어 사람이 취하는 동작에 따라 모드를 실행하게 된다. 또한, 음성 모듈에서 10가지 음성을 기억하여 기억된 음성이 실행되면 그에 따른 동작이 실행되게 된다.

2.2 하드웨어 구성

2.2.1 기구 구성

본 로봇의 기구적 특징은 크게 두 가지이다. 첫째, 로봇의 외형은 마사히로 이론을 토대로 설계하였다⁹⁾. 이 이론은 로봇이 사람의 외형에 가까이 근접할수록 친근함을 주지만, 그 이상으로 사람의 외형과 비슷할 경우 사람에게 혐오감을 준다는 내용이다. 이 이론을 반영하여 Fig. 3과 같이 펭귄 동물의 외형을 채택하여 사람들에게 귀엽고 다가가기 쉽게 로봇의 구조를 제작하였다. 둘째, 본 로봇은 2족 보행을 하는 로봇으로 제작하였다.

본 로봇은 기존의 휴머노이드 로봇과는 다른 메커니즘을 제안했다. 기존에 개발된 휴머노이드 로봇은 보통 한쪽 다리에 5개의 모터를 설치하여 작동한다. 본 연구에서는 무게를 줄이기 위해 새로운 기구적 메커니즘을 설계하여 기존의 로봇보다 전체적으로 적은 모터를 이용하여 이동, 회전, 모사를 가능하게 했다. Fig. 4는 제작된 펭귄로봇의 다리 와 상체 구조를 보여준다. 다음 장에서 설명하겠지만 영상처리를 통해 사용자로 하여금 간단하게 모드 선택 동작을 가능하게 하고 사람동작을 모사하기 위하여 Fig. 4와 같이 한쪽

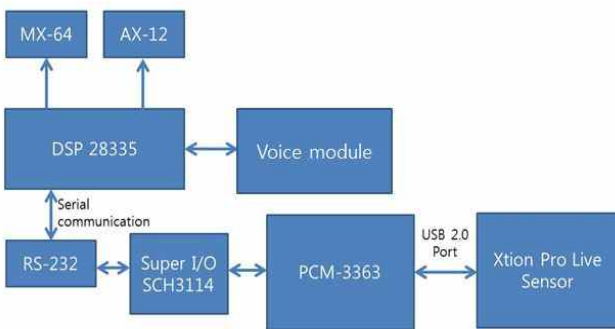


Fig. 1 Robot control system structure

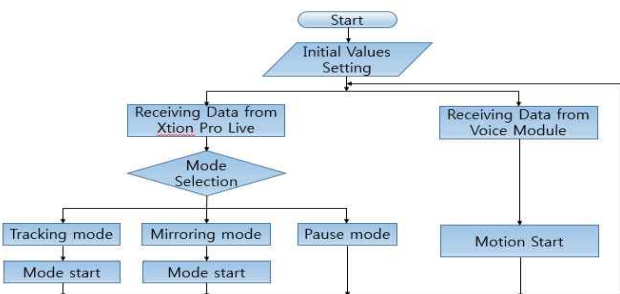


Fig. 2 Flow chart of robot operation



Fig. 3 Developed robot appearance employing uncanny valley theory

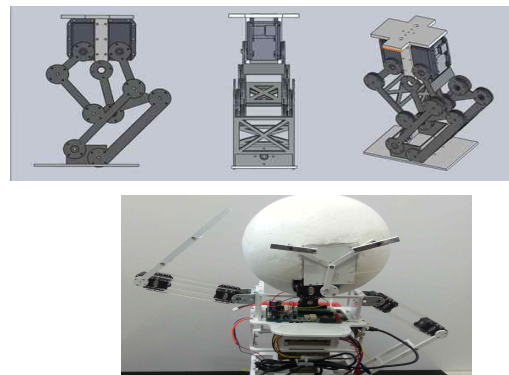


Fig. 4 Mechanism of robot leg and arm

팔에 2개의 모터와 허리에 1개의 모터를 사용하였다. 또한 사람의 동작을 비슷하게 표현하기 위해 사람의 상박과 하박의 길이비와 비슷하게 로봇을 상박과 하박으로 나누고 허리의 모터를 이용해 사람의 상체 동작인식으로 인사, 경례, 상체의 좌우 움직임 등 간단한 모사 기능만 가능하게 하였다.

2.3 소프트웨어 구성

2.3.1 소프트웨어 개요

본 로봇에서 동작을 인식하여 처리하기 위한 시스템 환경은 Table 1과 같다. Window 7(×86) 기반의 임베디드 보드에서 프로그래밍을 하였다. 기본적인 프로그래밍 언어로는 C언어 및 C++언어이다. OpenCV를 통해 영상을 입, 출력하게 되며 OpenNI를 통해 사람을 인식하고 깊이 거리를 측정한다. 또한, NITE를 통해 동작 인식을 완료하게 된다.

Xtion Pro Live를 구성하는 구조는 Fig. 5와 같다. 이 디바이스는 기본적인 OpenNI라이브러리를 이용하여 프로그래밍하게 되며 OpenNI 라이브러리는 해상도(resolution), 메타데이터(metadata) 등을 담은 그릇으로 기본 정보를 가지고 있으며 본 로봇에 사용되는 사용자 감지 및 거리 측정을 가능하게 한다. NITE는 동작인식을 지원하는 OpenNI의 미들웨어 라이브러리이다. 여기서 각 관절의 골격 데이터(skeleton data)를 추출할 수 있는 함수들을 제공한다.

본 로봇에는 위에 소개된 개발 환경이 구축되어 있고 이 개발

Table 1 Software development environment of Xtion Pro Live

Operation system	Window 7 ×86
Programming Language	C/C++
Additional Libraries	OpenCV, OpenNI, NITE
PC Environment	PCM-3363

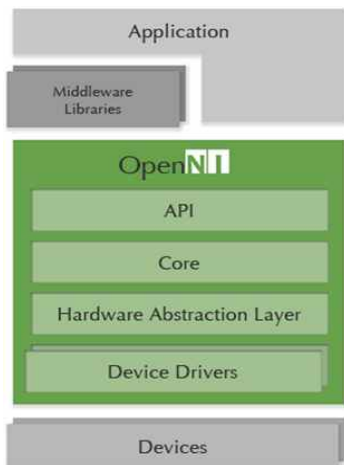


Fig. 5 Xtion Pro Live structure

환경을 이용하기 위해 이동형인 임베디드 PC^[10]가 이용되었다. 이것은 본 로봇에서 사용하는 Xtion Pro Live의 영상처리 및 RS-232 시리얼 통신을 이용하는데도 적합하다. 임베디드 PC에서 USB 2.0 포트를 통해 Xtion Pro Live와 연결되며 데이터는 RS-232 포트에서 마이크로프로세서로 통신된다.

2.3.2 영상처리 알고리즘

사용자의 모션을 인식하고 따라하는 과정에서는 거리지도(depth map)을 생성하여 거리영상을 출력하고 NITE를 이용해 동작을 인식하는 영상처리 과정을 거치게 된다.

영상처리에서 첫 번째 과정인 거리지도를 생성하기 위해서는 다음과 같은 과정을 거치게 된다. Xtion Pro Live를 구성하는 기본 구조인 OpenNI의 Context를 생성한다. 생성된 Context에서 거리정보를 구하기 위하여 DepthGenerator를 호출한다. DepthGenerator에서 호출된 거리 정보들은 OpenNI Context로 보내져 거리 지도를 생성한다. 생성된 거리 지도에서 추가로 출력 지도 모드(map mode)를 설정한다. 세로 480, 가로 640의 해상도를 지정한 후 FPS(frame per second)는 30이라는 VGA 모드를 설정하게 되면 설정된 지도 정보를 Context에 적용하고 지도 생성을 시작한다. 지도가 생성된 후 OpenCV의 입출력 헤더파일인 Highgui.h에서 영상 입출력 함수를 이용하여 창을 생성한다. 생성된 창에서 각각의 픽셀 RGB값을 설정하고 Context의 정보를 창에 적용하여 출력하게 되면 거리지도 생성이 완료된다. 생성된 지도에서는 동작인식은 가능하지 않고 Xtion Pro Live의 측정 가능 범위인 0.8 m~3.5 m까지의 영역의 거리값(depth)을 측정할 수 있게 된다.

영상처리 두 번째 과정인 동작인식을 이용하기 위해서는 OpenNI의 미들웨어 라이브러리인 NITE를 이용해야 한다. NITE에서 동작인식 함수들을 담고 있는 xml 파일을 이용하여 NITE의 정보들을 추출한다. 생성된 파일에서 노드(node)를 생성하고 동작인식을 지원하는 함수인 UserGenerator에서 사람을 인식하는 PoseDetection, 사람 관절을 인식하는 골격(skeleton)을 선언한다. 선언된 각각의 함수들에서 callback 함수를 이용해 UserDetection, Calibration, PoseDetection의 event를 확인한다. 이후 사람의 관절 데이터를 추출해 주는 함수인 GetSkeleton JointPosition 함수를 이용하여 사용자인 유저(user)의 해당 관절의 데이터를 추출한다. 이때 추출되는 데이터는 사람의 총 24개의 관절의 데이터를 표현하며 x 와 y 는 픽셀 좌표, z 는 거리로 표현된다. Fig. 6에서는 생성된 거리 지도위에 사람의 관절을 나타내는 골격 관절(skeleton joint)을 OpenCV를 이용하여 선으로 그려 나타내었다.

또한, 관절의 방향을 나타내는 회전(orientation)은 quaternion



Fig. 6 Skeleton image with depth map

으로 표현되지만 본 로봇에 적용하지 않았기 때문에 여기서는 설명하지 않는다.

2.3.3 동작 모드의 완성

영상을 활용한 세 가지 동작모드가 본 로봇에 활용되었다. 토르소 관절(torso joint)의 거리 값을 구하여 사용하는 추적모드가 있으며 각 관절에서 양팔의 상, 하박과 몸통의 각도를 벡터의 내적을 통해 계산하여 각도 값을 구하고, 사람의 위치를 판단하여 눈을 맞추는 미러링 모드가 두 번째 모드이고 마지막은 정지 모드이다. 추적 모드를 위해 토르소 관절의 거리를 구하는 것은 OpenNI의 UserGenerator 함수를 이용하였다. 이 함수에서는 사용자 변수, 관절 이름 및 관절 변수를 설정하면 거리 값을 mm단위로 얻을 수 있다.

모션 미러링 모드에서 사람의 동작을 따라 하기 위하여 설정한 각도는 오른팔 상박, 오른팔 하박, 왼팔 상박, 왼팔 하박 그리고 몸통 등 총 5가지이다. 각각의 각도는 벡터의 내적을 이용하였다.

$$\theta = \cos^{-1}\left(\frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|}\right) \quad (1)$$

식 (1)은 벡터의 내적으로부터 두 벡터 사이 각을 나타내는 수식이다. 식 (1)을 계산할 경우 왼팔 상박을 예를 들면 \vec{A} 는 목 관절과 왼쪽 어깨 관절을 연결한 벡터, \vec{B} 는 왼쪽 어깨 관절과 왼쪽 팔 관절을 연결한 벡터이다. 이와 같이 두 벡터의 사이 각을 구함으로써 관절의 각도 값을 구할 수 있다. 본 연구에서는 소스 코드로 이를 표현하였고 총 5가지의 관절 각도를 구하였으며 각각의 각도 값을 구하기 위한 관절 벡터의 설정 값은 Table 2에서 볼 수 있다. 여기서 구한 관절각도 값은 DSP로 전송되고 전송된 각도 값을 토대로 로봇이 사람의 모습을 따라하는 동작을 진행하게 된다.

모션 미러링에서 눈 접촉을 위해 사람의 목 관절을 기준으로 얻어지는 데이터는 거리와 2차원 평면의 x 와 y 의 값이다. x 와 y 의 값을 구하기 위해서는 픽셀 좌표와 거리 정보를 이용해야 한다. Fig. 7은 Table 2를 그림으로 나타낸 것이다.

Fig. 8에서 볼 수 있듯이 Xtion Pro Live의 FOV(field of view)

Table 2 Adjacent joint vectors to calculate each joint angle

Angle	Vector 1	Vector 2
Left upper arm [θ_1]	Neck ~ Left shoulder	Left shoulder ~ Left elbow
Left lower arm [θ_2]	Left elbow ~ Left shoulder	Left shoulder ~ Left elbow
Right upper arm [θ_3]	Neck ~ Right shoulder	Right shoulder ~ Right elbow
Right lower arm [θ_4]	Right elbow ~ Right shoulder	Right shoulder ~ Right elbow
Whole body [θ_5]	Torso ~ Neck	Left hip ~ Left knee

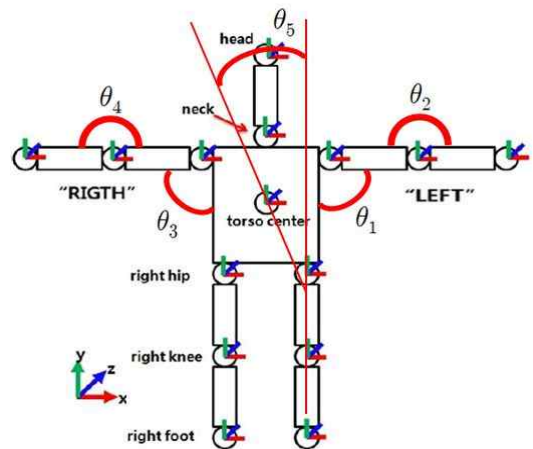


Fig. 7 Skeleton Definitions of human body

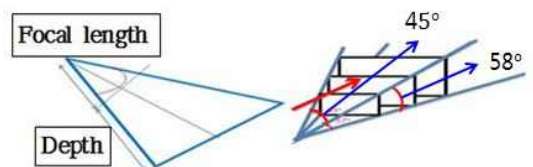


Fig. 8 Calculation for real x, y coordinates

는 45° (수직 방향 관측 각도)와 58° (수평 방향 관측 각도)이며 초점 거리는 1.083 mm이다. 이 값들을 식 (2)와 (3)에 적용하면 실제 x 와 y 의 값을 구할 수 있다.

$$x = (D + F) \tan\left(\frac{45^\circ}{2}\right) \frac{x_i - 320}{640} \quad (2)$$

$$y = (D + F) \tan\left(\frac{58^\circ}{2}\right) \frac{240 - y_i}{480} \quad (3)$$

여기서 D 는 거리이며 F 는 초점거리이고, x_i 와 y_i 는 수평 및 수직 방향의 픽셀좌표 값이다. 픽셀 좌표로 얻어지는 x 와 y 는 거리가 달라져도 같은 픽셀 정보를 가질 수 있기 때문에 시각 접촉에서는

많은 오차가 생긴다. Fig. 9에서 표현된 a_1 과 a_2 는 실제 x 와 y 값을 이용하여 거리에 따른 실제 사람의 위치를 판단할 수 있는 정보가 되며 depth와 실제 x 와 y 와의 tangent 값으로 계산된다. 구해진 a_1 과 a_2 는 DSP로 전송되어 로봇이 제어된다.

본 로봇에서 사용한 모드는 위의 설명과 같이 3가지 모드이다. 3가지 모드 각각의 동작을 만들어서 사용자가 선택할 수 있다. 로봇이 작동되면 사용자를 인식하고 보정과정이 끝나게 되면 모드선택 동작을 통해 모드를 고르게 되고 골라진 모드의 동작이 이루어지게 되는 것이다. 동작을 만드는 과정은 특정 관절각도 및 수직방향 픽셀 좌표를 통해 만들었으며 Fig. 10과 같이 3가지 동작을 통해 모드를 선택할 수 있다. 각각의 동작을 만들기 위한 조건은 다음과 같다. 추적 모드를 위한 조건에서는 관절각도의 범위 선정과 관절 수직 픽셀 좌표를 통해 설정했다. 관절각도 범위는 왼팔 하박의 각도가 75°에서 105°사이에 위치하고 오른팔 하박의 각도는 75°에서 105°사이에 위치하며, 관절 수직픽셀좌표는 왼손의 수직좌표가 왼쪽 어깨의 수직좌표보다 클 경우 및 오른손의 수직좌표가 오른쪽 어깨의 수직좌표보다 작을 경우이다. 이처럼 단순히 관절각도 및 수직 픽셀좌표를 토대로 만들었으며 나머지 2개의 동작을 생성하는 것 또한 위의 과정과 같다.

2.3.4 깊이 오차 보정

본 로봇에서 사용되는 추적과 모션 미러링은 Xtion Pro Live에서 측정되는 거리를 기반으로 동작이 이루어진다. 따라서 거리가 신뢰할 수 있는 값인지를 판단하는 것이 가장 중요한 문제이다. 거리를 측정하는 방식은 Xtion Pro Live에서 특정 패턴의 적외선이 송출되고 물체에 반사된 적외선이 카메라에 감지되는 원리를 이용

한다. 특정한 패턴이 몇 픽셀 이동했는가를 측정하여 이를 거리로 환산하며 거리를 측정하게 되는 것이다. Fig. 11과 같이 적외선은 특정 패턴을 가지고 있으며 측정 결과 1 cm의 거리변화 시 1픽셀이 이동하며 이는 최소 해상도가 1 cm라는 결과이다.

따라서 거리를 직접 측정하는 실험을 거쳐 오차를 분석하였다. 측정은 Xtion Pro Live의 인식범위인 0.8 m와 3.5 m 사이의 거리를 이용하였으며 적외선에 영향을 받지 않는 물체인 종이상자를 이용하여 카메라와 0.8 m 거리에서부터 100 mm씩 거리를 늘려가며 거리를 측정하였다. 측정은 각 거리마다 24개의 데이터를 평균하여 나타내었으며 이를 그래프로 나타내면 Fig. 12와 Table 3과 같다.

Table 3에서 데이터가 일정한 규칙 없이 늘어서 있기 때문에 측정된 거리 오차를 보정하여 처리할 필요가 있다. 여기서는 다항식 보간법(polynomial interpolation)을 이용했다. 다항식 보간법은 고차 다항식의 함수로 근사시켜 수치적인 방법으로 얻을 수 있다. 본 로봇에서 다항식 보간법을 통해 오차 값을 보정하였으며 보정된 오차 값을 통해 오차를 비교적 단순한 함수로 하여 보정하였다. 다항식 보간법은 식 (4)와 같다^[11].

$$y = a_0 + a_1x + \dots + a_{k-1}x^{k-1} + a_kx^k \quad (4)$$

식 (4)에서 $[k]$ 는 k 번째 거리 값을 의미하며 마찬가지로 $[k-1]$ 은 이전 측정 거리 값이다. 다항식 보간법을 통해 Matlab tool을 사용하여 보정된 오차 값과 보정전의 오차 값은 Fig. 13에 나타내었다. Fig. 13처럼 오차 값들은 식 (5)로 삼차함수로 간단히 표현될 수 있고 다항식 보간법으로 보정된 값에서 식 (5)를 감산하여 오차를

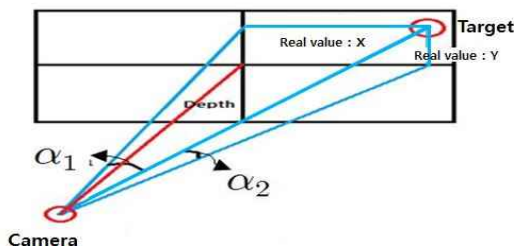


Fig. 9 Setting α_1 , α_2 for eye contact

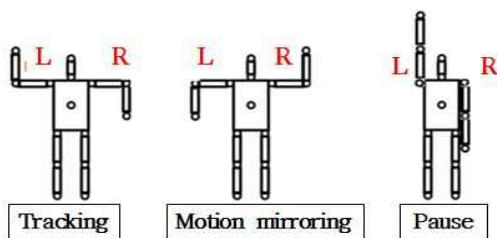


Fig. 10 Three Modes selections

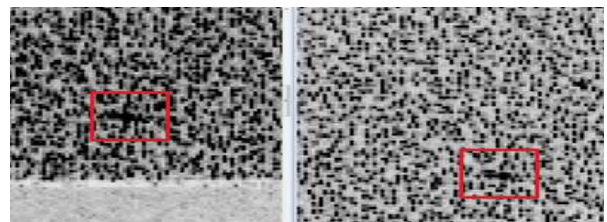


Fig. 11 Infrared ray pattern shapes for different depth

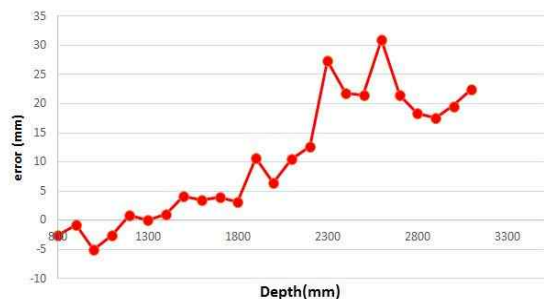


Fig. 12 Measured depth error graph

Table 3 Depth measurement error

Real distance [mm]	The measured values [mm]	Error [mm]
800	797.4	-2.6
900	899.2	-0.8
1000	995	-5
1100	1097.5	-2.5
1200	1200.8	0.8
1300	1300	0
1400	1401	1
1500	1504.1	4.1
1600	1603.5	3.5
1700	1704.7	4.7
1800	1803.1	3.1
1900	1910.7	10.7
2000	2006.4	6.4
2100	2110.5	10.5
2200	2212.6	12.6
2300	2327.4	27.4
2400	2421.8	21.8
2500	2521.4	21.4
2600	2631	31
2700	2721.5	21.5
2800	2818.4	18.4
2900	2917.6	17.6
3000	3019.6	19.6
3100	3122.5	22.5

Table 4 The depth measurement error correction

Real distance [mm]	The measured values [mm]	Error [mm]
800	798.4	-1.6
900	901.3	1.3
1000	997.7	-2.3
1100	1100.2	0.2
1200	1203	3.0
1300	1301.3	1.3
1400	1401	1
1500	1502.5	2.5
1600	1600.1	0.1
1700	1699.3	-0.7
1800	1795.7	-4.3
1900	1900.8	0.8
2000	1994.5	-5.5
2100	2096.3	-3.7
2200	2196.4	-3.6
2300	2309	9
2400	2401.9	1.9
2500	2500.2	0.2
2600	2608.8	8.8
2700	2698.8	-1.2
2800	2795.7	-4.3
2900	2895.4	-4.6
3000	2998.5	-1.5
3100	3103.1	3.1

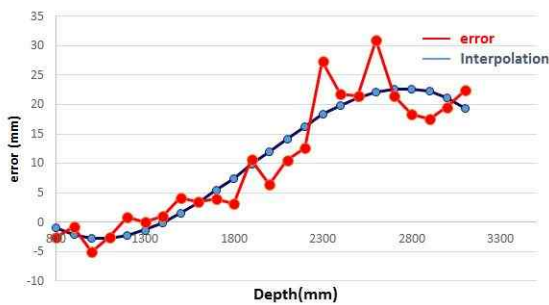


Fig. 13 Adjusted depth error compensation by polynomial interpolation method

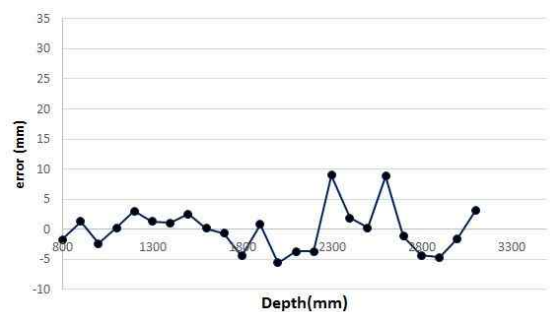


Fig. 14 Finally adjusted depth error compensation

보정하였으며 이를 Table 4에서 결과 값을 나타내었다. 또한 보정된 오차 값들을 Fig. 14에서 볼 수 있다. 최종적으로 보정된 Fig. 15의 오차 값들은 일부분을 제외하면 오차가 거의 없으며 이는 보정된 거리 값들을 신뢰할 수 있다고 판단할 수 있었으며 이를 코딩으로 변환하여 적용하였다.

$$y = -9.865 \times 10^{-9}x^3 + 5.674 \times 10^{-5}x^2 - 0.08667x + 37.11 \quad (5)$$

위에서 설명된 내용을 토대로 진행한 영상처리 중 추격모드와 모션 미러링 모드 Fig. 15와 Fig. 16에 표시하였다. Fig. 15에서는 추격모드를 실행하여 토르소 관절의 x와 y의 실 좌표 및 거리를

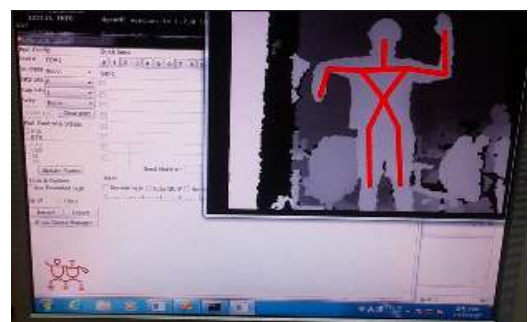


Fig. 15 Image of tracking mode

DSP로 전송하는 화면이다. Fig. 16에서는 모션 미러링 모드를 실행하여 왼팔 상박, 왼팔 하박, 오른팔 상박, 오른팔 하박, 몸통 각도

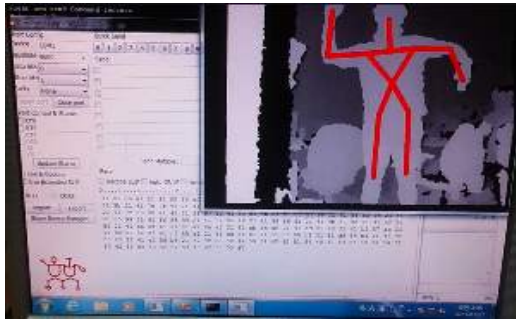


Fig. 16 Image of motion mirroring mode

Table 5 The depth up to torso joint

Real distance [mm]	The measured values [mm]	The corrected values [mm]
1900	1910.700024	1900.859075
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245
1901	1910.800000	1900.800025
1900	1910.700024	1900.859075
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245
1901	1910.800000	1900.800025
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245
1900	1910.700024	1900.859075
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245
1901	1910.800000	1900.800025
1900	1910.700024	1900.859075
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245
1901	1910.800000	1900.800025
1902	1910.899976	1900.750245

그리고 시각 접촉을 위한 두 가지의 각도 값을 DSP로 전송하는 화면이다. DSP로 전송하기 위한 통신 방법은 RS-232 직렬 통신방식을 이용하였으며 각 데이터는 16진수로 전송하였다. 이 과정을 토대로 로봇은 각각의 동작에 따라 제어된다. Table 5를 보면 알 수 있듯이 위의 과정을 통하여 로봇과 사람의 거리를 20회에 걸쳐 측정값과 오차를 보정한 값을 볼 수 있다. 보정 과정을 통하여 오차를 최소화하고 로봇의 사람 동작모사 테스트를 통하여 팔 상박의 각도 결과 값을 Table 6에 표시 하였고. 이를 바탕으로 Table 6의

Table 6 The angle values of right upper arm

Count	Region	Right upper arm (degree)	
		The measured values	Real angle value
1		8°	10°
2		19°	20°
3		33°	30°
4		44°	40°
5		52°	50°
6		60°	60°
7		73°	70°
8		83°	80°
9		91°	90°
10		102°	100°
11		111°	110°
12		121°	120°
13		133°	130°
14		142°	140°
15		152°	150°

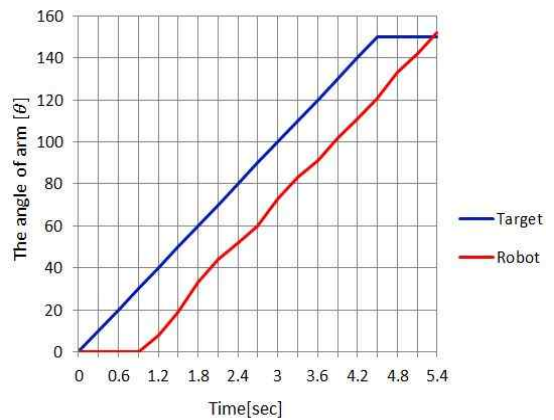


Fig. 17 The histories of the target and the arm angle

결과 값을 Fig. 17과 같이 실험을 통해서 Table 6의 결과 값을 통해 목표인 사람의 동작을 로봇의 오른쪽 팔의 상박 각도를 그래프로 표시하였다. Table 6과 Fig. 17을 보면 알 수 있듯이 오차는 0~4°차이로 근소하게 나타났다. 그리고 실제 로봇 동작시간과 타겟을 실시간으로 측정한 Fig. 17을 보면 1~2초 정도 지연되어 움직임을 알 수 있다. 초기 임베디드 보드를 이용하며 거리 창을 생성하고 동작을 인식하기까지 약 20~25초라는 시간이 소요된다. 더 좋은 사양의 임베디드 보드의 선택과 인식알고리즘이 보완되면 보다 빠른 시간 내에 동작이 구현될 것이다.

3. 결론

본 논문에서 가장 중요하게 다룬 내용은 동작 인식 기능이며 이

를 개발한 2족 보행 펭귄 로봇에 적용하여 효용성을 검증하였다. 동작인식의 경우 본 로봇의 구현목표에서 사용자를 인식하는 사용자 감지, 원하는 동작을 생성하는 모드 선택 그리고 관절의 수평 및 수직 길이와 물체까지의 거리를 추출하는 방법을 사용하여 원하는 관절의 데이터를 제어부인 DSP로 전송할 수 있었다.

본 연구에서 펭귄 로봇은 엔터테인먼트 목적으로 개발되었다. 하지만 로봇 자체가 가지는 동작의 한계성 때문에 엔터테인먼트 로봇으로서의 충분한 기능을 가지기에는 아직은 미흡한 상태이다. 동작의 다양화, 로봇의 지능화를 통한 로봇 개발의 발전은 여전히 로봇 산업의 과제이다. 또한 로봇의 움직임에 따라 Xtion Pro Live에서 사용자의 관절 값이 깨지는 경우가 발생하기도 하는데 로봇이 움직이는데 치명적인 결함이 되므로 로봇의 움직임에도 관절 값이 잘못 계산되지 않는 방법 및 관절 값의 이상발생 시 이를 사전에 예방하거나 동작을 재조정하는 등의 연구가 계속 필요하다.

후 기

이 연구는 서울과학기술대학교 교내 연구비 지원으로 수행되었습니다.

References

- [1] Kim, S., Choi, C., Lim, G., 2013, KISTEP 10 Future Technologies for Smart Ageing, KISTEP Issue Paper 2013-02, Korea Institute of Science & Technology Evaluation and Planning, 12-35.
- [2] Park, T., Yang, E., Hwang, W., 2012, Human Motion Analysis for Silver Robots, *Journal of Korean Society of Design Science*, 25:3 3-8.
- [3] Ryu, J., Park, C., Kim, J., Kang, S., 2008, KOBIE: A Pet-type Emotion Robot, *Journal of Korea Robotics Society*, 3:2 154-163.
- [4] Shibata, T., 2011, Research on Interaction between Human and Seal Robot, PARO, *Robotics Society of Japan*, 29:1 31-34.
- [5] Yang, S., Byun, J., Seo, J., 2008, The Design of Kids Robot: Rabie, *Journal of Korean Society of Design Science*, 21:6 79-82.
- [6] Kang, Y., Ho, Y., 2011, Depth Generation Method Using Multiple Color and Depth Cameras, *Journal of the Institute of Electronics of Korea*, 48:3 13-18.
- [7] Kim, J., Kim, Y., Suga, Y., 2008, Recognition of Moving Objects in Mobile Robot with an Omnidirectional Camera, *Journal of Robotics Society*, 3:2 91-97.
- [8] Jang, W., Kim, J., and Lee, D., 2013, A Study on Estimating Finger Joint and 3D Modelling Hand Using Depth Camera, *Journal of The Korea Information Science*, 40:12 794-798.
- [9] Mori, M., 1970, The Uncanny Valley, *Energy(in Japanese)*, 7:4 33-35.
- [10] PCM-3363, 2011, PCM-3363 User Manual, ADVANTECH Co. Ltd., 5.
- [11] Weisstein, Eric, W., n.d., viewed 20 September 2015, Least Squares Fitting—polynomial, <<http://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html>>.
- [1] Kim, S., Choi, C., Lim, G., 2013, KISTEP 10 Future Technologies for Smart Ageing, KISTEP Issue Paper 2013-02, Korea Institute of Science