

인체모션 데이터 획득 장치와 최적화 기법을 사용한 로봇운동 데이터 생성과 애니메이션

배태영^{a*}, 김영석^b

Generation and Animation of Optimal Robot Joint Motion data using Captured Human Motion data

Tae Young Bae^{a*}, Young Seog Kim^b

^a Seoul National University of Science and Technology, 317 Fronier Hall,
232 Gongneung-ro, Nowon-gu, Seoul, 139-743, Republic of Korea

^b Seoul National University of Science and Technology, 918 Fronier Hall, Republic of Korea

ARTICLE INFO

Article history:

Received	1	April	2013
Revised	9	May	2013
Accepted	14	May	2013

Keywords:

Motion capture keyword
Android robot
Marker
Motion
Motion builder

ABSTRACT

This paper describes a whole-body (human body's) motion generation scheme for an android robot that uses motion capture device and a nonlinear constrained optimization method. Because the captured motion data are based on global coordinates and the actors have different heights and different upper-lower body ratios, the captured motion data cannot be used directly for a humanoid robot. In this paper, we suggest a method for obtaining robot joint angles, which allow the resultant robot motion to be as close as possible to the captured human motion data, by applying a nonlinear constrained optimization method. In addition, the results are animated to demonstrate the similarity of the motions.

1. 서론

최근 로봇분야는 구동요소 및 컴퓨터와 영상처리 등의 기술적 진보를 바탕으로 국내외 연구가 활발히 이루어지고 있다. 그 중에 이족 보행 로봇에 대한 연구로서 인체와 흡사한 모션을 생성하는 방법에 대한 연구가 활발하게 진행되고 있다^[1]. 이족 보행 로봇은 인간형 로봇으로 개발하기 위하여 센서, 비전, 제어, 지능 등의 다양한 분야의 연구^[5-7]가 필요하지만, 기본적으로는 안정된 모션의 구현이 가장 주요한 연구 과제이다. 안정된 모션을 구현하기 위하여 다양한 방법으로 연구되고 있으며, 최근 인체모션을 캡처하여 로봇에 적용하는 방법이 활발히 진행 중이다^[8].

그동안 발표된 논문들에서 제시된 기술을 살펴보면 로봇 모션 패턴은 구조와 수학적 계산을 토대로 한 데이터를 이용하여 추출한다^[2].

계산을 토대로 한 데이터는 연결성의 부정확함과 자연스럽지 못하는 현상이 생긴다^[3-4]. 따라서 본 논문에서는 인체의 움직임을 추적하고 추적된 움직임의 데이터를 이용한다.

본 논문에서는 motion capture 장비를 이용하여 인체 motion에 대한 데이터를 추출하는 과정과 각 관절의 데이터 변환 및 robot에 적용하기 적합한 데이터로 변환하는 과정을 제시하였다. 특히 본 논문에서는 robot 관절에 맞는 데이터로 변환하기 위하여, 캡처된 인체운동데이터와 robot의 운동데이터사이의 error를 최소화하는 최적화 기법을 제시하였다. 최적화 기법 알고리즘을 적용함으

* Corresponding author. Tel.: +82-70-8656-3505

Fax: +82-2-944-6352

E-mail address: tybae@robocube.co.kr (Tae Young Bae).

로서 인체 모션에 보다 흡사한 로봇 관절데이터를 얻을 수 있었으며, Matlab 코드 및 OpenGL과 inventor 시뮬레이션을 통하여 최적화된 data를 확인하였다.

2. 본 론

2.1 Motion capture장비를 통한 인체 동작 Data의 추출

Fig. 1의 optitrack camera는 motion technologies (社)에서 개발한 motion 캡처용 camera이다. 이 camera는 인체를 촬영하는 용도로 사용되는데, 총 12대가 marker를 부착한 인체의 주위에 놓여져서 초당 100장의 영상을 PC로 전송한다. PC로 전송된 영상은 Motion Builder Program에 입력되어 인체의 각 관절에 대한 data로 추출된다. 이때 추출된 인체관절데이터는 global 좌표계에 대한 변위와 오일러 angle 값으로 구성된다. 이와 같이 추출된 인체관절 데이터를 robot관절에 적용하기 위해서는, pelvis center에 고정된 좌표계에 대한 각 관절의 상대변위와 상대 각 data를 얻어야 하는데, Motion Builder Program에서는 제공하지 않고 있다.



Fig. 1 Optitrack Camera

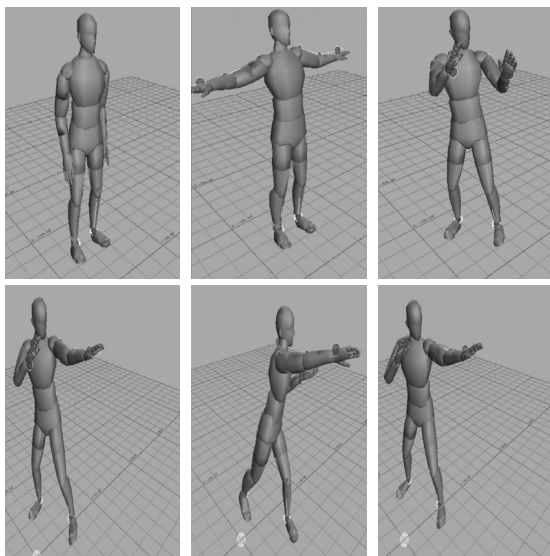


Fig. 2 Animation motion capture

2.2 인체 Motion 선택

인체 motion의 선택으로 복싱동작을 선택하였는데, 이는 모든 관절을 사용하며, dynamic stepping과 같은 난이도 높은 동작을 포함하고 있기 때문이다 (Fig. 2 참조).

인체에서 추출한 data를 robot에 적용하기 위해서는 robot에 맞는 scaling과 data변환이 필요하다. 그 이유는 robot이 동작데이터를 캡처한 인체와 키, 상-하체 비율에서 매우 다르기 때문이다.

2.3 Homogeneous transformation

로봇은 골반을 중심으로 각 관절의 각을 변화시킴으로써 원하는 동작을 만들게 된다. 그러나 모션캡처 장비를 통하여 추출된 데이터는 Global 좌표에 대한 각 관절의 변위와 오일러 angle 값이므로, 로봇에 적용하려면 골반좌표계에 대한 각 관절의 상대변위와 상대 각 data로 변환하는 과정이 필요하다. 이를 위하여 homogeneous transformation을 사용한다. transformation은 Fig. 3과 같이 사용하였고 식 (2)에서 Trans (x, y, z)는 global 좌표계(x_0, y_0, z_0)의 원점에서 골반 좌표계(x_1, y_1, z_1)의 원점을 가리키는 벡터이며, Rot (x, y, z)는 global 좌표계로부터 골반 좌표계(x_1, y_1, z_1)로의 Euler angle이다. 따라서 global 좌표계로 표현된 공간상의 임의 한 점을 골반 좌표계(x_1, y_1, z_1)로 표현하기 위한 좌표변환 행렬은 식 (2)의 A_0^1 matrix와 같다 (Fig. 3 참조).

$$R = Rot_{z,\gamma} Rot_{y,\beta} Rot_{x,\alpha}$$

$$= \begin{bmatrix} c_\gamma & -s_\gamma & 0 & c_\beta & 0 & s_\beta \\ s_\gamma & c_\gamma & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix} \quad (1)$$

$$= \begin{bmatrix} c_\gamma c_\beta & -s_\gamma c_\alpha + c_\gamma s_\beta s_\alpha & s_\gamma s_\alpha + c_\gamma s_\beta c_\alpha \\ s_\gamma c_\beta & c_\gamma c_\alpha + s_\gamma s_\beta s_\alpha & -c_\gamma s_\alpha + s_\gamma s_\beta c_\alpha \\ -s_\beta & c_\beta s_\alpha & c_\beta c_\alpha \end{bmatrix}$$

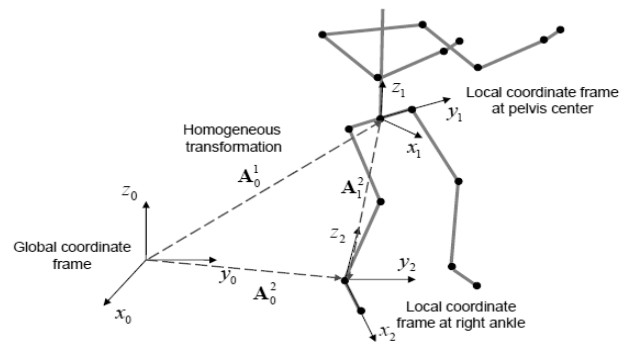


Fig. 3 Coordinate transformation of the right ankle joint from global frame to local frame

$$A_0^1 = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} = \text{Trans}_{z,c} \cdot \text{Trans}_{y,b} \cdot \text{Trans}_{x,a} \cdot \text{Rot}_{z,\gamma} \cdot \text{Rot}_{y,\beta} \cdot \text{Rot}_{x,\alpha}$$

$$= \begin{bmatrix} c_\gamma c_\beta & -s_\gamma c_\alpha + c_\gamma s_\beta s_\alpha & s_\gamma s_\alpha + c_\gamma s_\beta c_\alpha & a \\ s_\gamma c_\beta & c_\gamma c_\alpha + s_\gamma s_\beta s_\alpha & -c_\gamma s_\alpha + s_\gamma s_\beta c_\alpha & b \\ -s_\beta & c_\beta s_\alpha & c_\beta c_\alpha & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

식 (2)에서 S와 C는 각각 Sine과 Cosine 함수를 약자로 가리키며 a, b, c는 Translation, α, β, γ 는 Rotation이다.

본 논문에서 사용한 모션캡처 장비를 사용하면 global 좌표계로부터 골반 좌표계(x_1, y_1, z_1)로의 원점 간 상대위치벡터와 Euler angle을 얻을 수 있으므로, 이를 식 (2)에 대입하면 좌표변환 행렬 A_0^1 을 정할 수 있다. 마찬가지로 방법으로 global 좌표계로부터 발목에 고정된 좌표계(x_2, y_2, z_2) 로의 좌표변환 행렬 A_0^2 를 정할 수 있다.

이때 A_0^1 과 A_0^2 를 사용하면 골반 좌표계(x_1, y_1, z_1)로부터 발목에 고정된 좌표계(x_2, y_2, z_2)로의 좌표변환 행렬 A_1^2 를 다음과 같은 식 (3)으로 얻을 수 있다.

$$A_1^2 = A_0^2(A_0^1)^{-1} \quad (3)$$

식 (3)을 이용하면 골반 좌표계(x_1, y_1, z_1)로부터 발목에 고정된 좌표계(x_2, y_2, z_2) 로의 상대위치 값과 상대관절각 값을 계산할 수 있는데, 식 (3)을 성분으로 표시하면 식 (4)와 같으며

$$A_1^2 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

식 (4)로부터 골반 좌표계(x_1, y_1, z_1)와 발목에 고정된 좌표계(x_2, y_2, z_2)사이의 관절각 값을 식 (5), 식 (6), 식 (7)에 따라서 구할 수 있다.

$$\theta_x = \tan^{-1}\left(\frac{n_y}{n_x}\right) \quad (5)$$

$$\theta_y = \tan^{-1}\left(\frac{-n_z}{n_x \cos \theta_x + n_y \sin \theta_x}\right) \quad (6)$$

$$\theta_z = \tan^{-1}\left(\frac{a_x \sin \theta_x - a_y \cos \theta_x}{o_y \cos \theta_x + o_x \sin \theta_x}\right) \quad (7)$$

본 논문에서는 pelvis center에 고정된 좌표계(x_1, y_1, z_1)가 모든 관절의 위치와 orientation을 나타내는 기준으로 하였다. 이에 따라서 pelvis center를 제외한 모든 관절에 대하여 식 (3)~식 (7)을 적용하여 pelvis center에 대한 상대 좌표변환 행렬을 얻도록 하였다.

2.4 Robot 관절 DATA의 생성을 위한 Scaling

robot은 (社)로 보큐브테크에서 개발 중인 안드로이드 robot의 size를 이용하여 인체data를 robot data로 scaling을 하였다.

robot의 이름은 SARA이고 Slim Ankle Robot Android의 약어다. SARA는 여자 인체를 모방하여 설계되어졌다.

Fig. 4는 3 D model과 관절의 위치와 방향을 나타낸다. robot의 키는 160 cm이며 46 kg이다. SARA Robot은 slim 발목이 특징이며 발목의 roll, pitch 관절이 분리되어 있다.

인체 data를 robot데이터 변환을 위해 2단계 변환 방법이 필요하다. 먼저 인체와 robot의 신장 비율을 사용하여 인체의 motion data를 스케일링을 하고, 인체와 robot의 팔 길이의 비율이 다르므로 목을 중심으로 팔 길이 비율을 사용하여 팔 motion data를 스케일링 한다 (Table 1 참조).

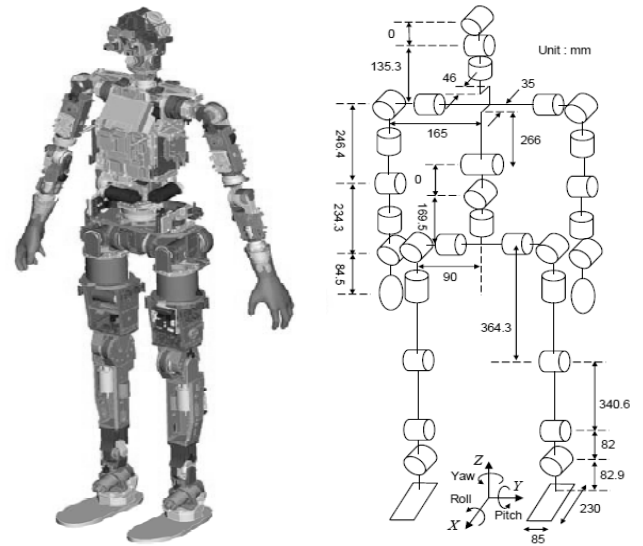


Fig. 4 3D model robot and joint structure and dimensions

Table 1 Height Ratio and Arm Ratio

	Height[mm]	Arm length[mm]
HUMAN	1,800	740
SARA	1,602	499
Height Ratio	1,602/1,800 = 0.89	
Arm Ratio	$\frac{499}{740 \times 0.89} = 0.757668$	

2.5 Robot 관절 DATA의 최적화 과정

비선형 최적화 기법은 capture된 관절 운동 데이터에 가장 근접한 운동데이터를 생성하는 로봇관절의 움직임에 대한 값을 얻을 수 있게 해준다. 이를 위하여 본 논문에서는 비선형 최적화 기법을 사용하였으며, 다리, 몸통, 팔의 관절을 구분하여 error의 비용함수 값을 정의하고, 모든 관절에 대한 전체 비용함수는 식 (8)과 같이 설정하였다. 식 (8)에서 k는 Time frame이며 다음 프레임으로 넘어갈 경우 k+1로 표현된다. 비선형최적화 문제를 해결하기위해서 1/100초 단위로 시간을 증가시키면서 반복적으로 계산을 하였다. 계산시간을 짧게 하기 위하여 이전의 결과를 time step에서 초기값으로 사용했다.

$$J_{total}^k = c_{leg} \cdot J_{leg}(\theta_1^k, \dots, \theta_{12}^k) + c_{torso} \cdot J_{torso}(\theta_{13}^k, \theta_{14}^k, \theta_{15}^k) \quad (8)$$

$$+ c_{arm} \cdot J_{arm}(\theta_{16}^k, \dots, \theta_{23}^k)$$

다리, 몸통, 팔에 대한 각각의 비용함수를 3단계로 나누어서 식 (9)와 같이 순차적으로 최적화를 수행하였다.

$$1^{st} \text{ step: } c_{leg} = 1, c_{torso} = 0, c_{arm} = 0 \quad (9)$$

$$\min_{\theta_1^k, \dots, \theta_{12}^k} J_{total}^k(\theta_1^k, \dots, \theta_{23}^k) = J_{total}^k(\theta_1^{k*}, \dots, \theta_{12}^{k*}, \theta_{13}^k, \dots, \theta_{23}^k)$$

$$2^{nd} \text{ step: } c_{leg} = 1, c_{torso} = 1, c_{arm} = 0$$

$$\min_{\theta_1^k, \dots, \theta_{15}^k} J_{total}^k(\theta_1^k, \dots, \theta_{12}^k, \theta_{13}^k, \dots, \theta_{23}^k)$$

$$= J_{total}^k(\theta_1^{k*}, \dots, \theta_{15}^{k*}, \theta_{16}^k, \dots, \theta_{23}^k)$$

$$3^{rd} \text{ step: } c_{leg} = 1, c_{torso} = 1, c_{arm} = 1$$

$$\min_{\theta_1^k, \dots, \theta_{23}^k} J_{total}^k(\theta_1^k, \dots, \theta_{15}^k, \theta_{16}^k, \dots, \theta_{23}^k) = J_{total}^k(\theta_1^{k*}, \dots, \theta_{23}^{k*})$$

식 (9)에서 superscript*는 최적화된 관절각의 값을 나타내며, $\theta_1^{k*}, \dots, \theta_{12}^{k*}$ 는 다리관절각, $\theta_{13}^{k*}, \dots, \theta_{15}^{k*}$ 는 몸통관절각, $\theta_{16}^{k*}, \dots, \theta_{23}^{k*}$ 는 팔 관절각을 가리킨다.

1st step에서는 총 비용함수 중에서 다리부분에 대해서만 비용함수를 최소화하도록 다리관절각 값을 정한다. 2nd step에서는 총 비용함수 중에서 몸통부분에 대해서만 비용함수를 최소화하도록 몸통관절각 값을 정한다. 단, 2nd step에서는 1st step에서 구한 최적화된 다리관절각 값을 사용한다. 3rd step에서는 총 비용함수 중에서 팔부분에 대해서만 비용함수를 최소화하도록 팔 관절각 값을 정한다. 단, 3rd step에서는 1st, 2nd step에서 구한 최적화된 다리관절각 및 몸통관절각을 사용한다. 이와 같이 단계적으로 최적화함으로써 최적화 대상이 되는 관절수를 줄여서 최적화 알고리즘

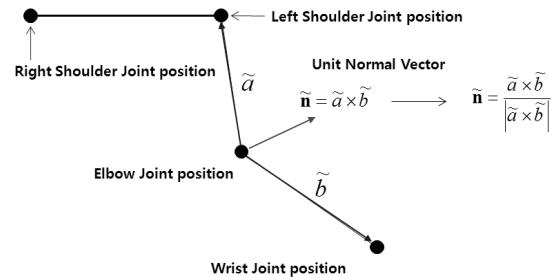


Fig. 5 Definitions of the vectors in the upper body

의 수렴성(convergence)을 높일 수 있다.

Fig. 5와 식 (10), 식 (11), 식 (12)을 사용하여 식 (8)의 비용함수를 구체적으로 나타내었다. 식 (10)은 다리 비용함수이고 식 (11)은 몸통 비용함수, 식 (12)는 팔 비용함수에 관한 내용이다.

Fig. 5에서 \hat{n} 은 어깨, 팔꿈치, 손목으로 구성되는 평면의 단위벡터이며 팔꿈치의 방향을 나타낸다.

$$J_{leg} = \sqrt{k_1 \cdot e_{A.P}^2 + k_2 \cdot e_{A.O}^2 + k_3 \cdot e_{K.P}^2 + k_4 \cdot e_{K.O}^2} \quad (10)$$

$$J_{torso} = \sqrt{e_{R.SP}^2 + e_{N.P}^2 + e_{L.SP}^2} \quad (11)$$

$$J_{arm} = \sqrt{e_{W.P}^2 + e_{E.P}^2} + k_t(A+B) \quad (12)$$

$$A = k_\theta(1 - \hat{n}_a \cdot \hat{n}_r) / 2$$

$$B = (1 - k_\theta)(k_p|\theta_{SY}| + k_d|\hat{\theta}_{SY}|)$$

$$k_\theta = |\hat{a}_a \cdot \hat{b}_a|$$

식 (10)과 식 (11), 식 (12)에서 비용함수 J가 작을수록 인체 운동 data와 robot data간의 모션이 일치한다.

식 (10)에서 k_1, k_2, k_3, k_4 는 가중치 상수이다. k_1 는 발목 위치에 대한 가중치이고 k_2 는 발목 방위에 대한 가중치, k_3 는 무릎 위치에 대한 가중치, k_4 는 무릎 방위에 대한 가중치이다. 다수의 시행착오를 거쳐서 k_1 는 1.0, k_2 는 0.2, k_3 는 1.0, k_4 는 1.0으로 선정하였다. $e_{A.P}, e_{A.O}, e_{K.P}, e_{K.O}$ 는 발목과 무릎에 대한 변위와 각 오차로서 식 (13)과 같이 정의하였다.

$$e_{A.P} : \text{ankle position error}, \quad (13)$$

$$e_{A.P}^2 = (P_{A.Px}^{ref} - P_{A.Px})^2 + (P_{A.Py}^{ref} - P_{A.Py})^2 + (P_{A.Pz}^{ref} - P_{A.Pz})^2$$

$e_{A.O}$: ankle orientation error,

$$e_{A.O}^2 = (R_{A.O.x}^{ref} - R_{A.O.x})^2 + (R_{A.O.y}^{ref} - R_{A.O.y})^2 + (R_{A.O.z}^{ref} - R_{A.O.z})^2$$

$e_{K.P}$: knee position error,

$$e_{K.P}^2 = (P_{K.P.x}^{ref} - P_{K.P.x})^2 + (P_{K.P.y}^{ref} - P_{K.P.y})^2 + (P_{K.P.z}^{ref} - P_{K.P.z})^2$$

$e_{K.O}$: knee orientation error,

$$e_{K.O}^2 = (R_{K.O.x}^{ref} - R_{K.O.x})^2 + (R_{K.O.y}^{ref} - R_{K.O.y})^2 + (R_{K.O.z}^{ref} - R_{K.O.z})^2$$

식 (11)에서 $e_{R.SP}$, $e_{N.P}$, $e_{L.SP}$ 는 어깨와 목의 위치 에러로서 식 (14)의 정의와 같다.

$e_{R.SP}$: Right shoulder position error, (14)

$$e_{R.SP}^2 = (P_{R.SP.x}^{ref} - P_{R.SP.x})^2 + (P_{R.SP.y}^{ref} - P_{R.SP.y})^2 + (P_{R.SP.z}^{ref} - P_{R.SP.z})^2$$

$e_{L.SP}$: Left shoulder position error,

$$e_{L.SP}^2 = (P_{L.SP.x}^{ref} - P_{L.SP.x})^2 + (P_{L.SP.y}^{ref} - P_{L.SP.y})^2 + (P_{L.SP.z}^{ref} - P_{L.SP.z})^2$$

$e_{N.P}$: Neck position error,

$$e_{N.P}^2 = (P_{N.P.x}^{ref} - P_{N.P.x})^2 + (P_{N.P.y}^{ref} - P_{N.P.y})^2 + (P_{N.P.z}^{ref} - P_{N.P.z})^2$$

여기서, $e_{R.SP}$ 는 오른쪽 어깨 위치 error이고 $e_{N.P}$ 는 목의 위치 error, $e_{L.SP}$ 는 왼쪽 어깨 위치에 관한 error이다.

식 (12)에서 $e_{W.P}$, $e_{E.P}$ 은 손목과 팔꿈치에 대한 변위 오차로서 식 (15)의 정의와 같다.

$e_{W.P}$: wrist shoulder position error, (15)

$$e_{W.P} = e_{R.W.P} + e_{L.W.P}$$

$$e_{R.W.P}^2 = (P_{R.W.P.x}^{ref} - P_{R.W.P.x})^2 + (P_{R.W.P.y}^{ref} - P_{R.W.P.y})^2 + (P_{R.W.P.z}^{ref} - P_{R.W.P.z})^2$$

$$e_{L.W.P}^2 = (P_{L.W.P.x}^{ref} - P_{L.W.P.x})^2 + (P_{L.W.P.y}^{ref} - P_{L.W.P.y})^2 + (P_{L.W.P.z}^{ref} - P_{L.W.P.z})^2$$

$e_{E.P}$: Elbow position error,

$$e_{E.P} = e_{R.E.P} + e_{L.E.P}$$

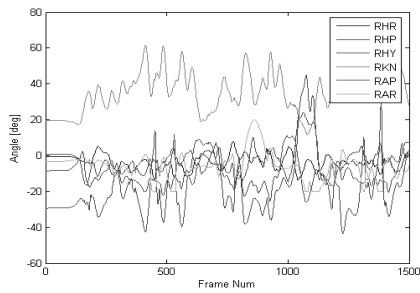
$$e_{R.E.P}^2 = (P_{R.E.P.x}^{ref} - P_{R.E.P.x})^2 + (P_{R.E.P.y}^{ref} - P_{R.E.P.y})^2 + (P_{R.E.P.z}^{ref} - P_{R.E.P.z})^2$$

$$e_{L.E.P}^2 = (P_{L.E.P.x}^{ref} - P_{L.E.P.x})^2 + (P_{L.E.P.y}^{ref} - P_{L.E.P.y})^2 + (P_{L.E.P.z}^{ref} - P_{L.E.P.z})^2$$

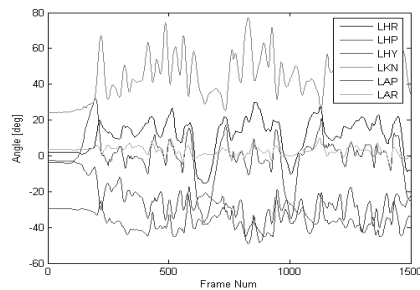
식 (12)에서 첨자 a, r은 actor (인체)와 robot을 가리키며, $e_{W.P}$ 는 손목관절 위치 error이고 $e_{E.P}$ 는 팔꿈치 위치 error이다. 식 (12)에는 손목과 팔꿈치 위치 이외에 term A, term B를 추가되었는데, term A는 인체와 robot간의 팔꿈치 방향을 일치시키기 위한 항이며, term B는 인체 법선벡터 \hat{n}_a 와 로봇 법선벡터 \hat{n}_r 가 동일할 경우 0이 되게 하는 역할을 함으로서, 팔을 수평으로 펼쳤을 때 어깨와 팔에 자유도가 겹쳐서 발생하는 수치적 문제를 해결하는 목적으로 사용된다.

Table 2 Name of the Joint Angle

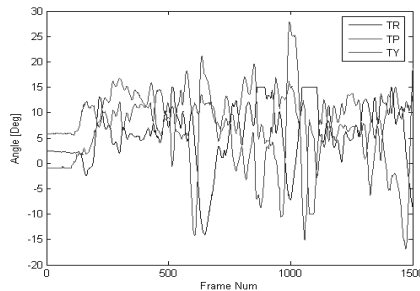
Fig2.6	part	Name	Joint
(a)	leg	RHR	Right Hip Roll
		RHP	Right Hip Pitch
		RHY	Right Hip Yaw
		RKN	Right Knee Pitch
		RAP	Right Ankle Pitch
		RAR	Right Ankle Roll
		LHR	Left Hip Roll
		LHP	Left Hip Pitch
		LHY	Left Hip Yaw
		LKN	Left Knee Pitch
		LAP	Left Ankle Pitch
		LAR	Left Ankle Roll
(c)	Torso	TR	Torso Roll
		TP	Torso Pitch
		TY	Torso Yaw
(d)	Arm	RSR	Right Shoulder Roll
		RSP	Right Shoulder Pitch
		RSY	Right Shoulder Yaw
		REB	Right Elbow Pitch
		RWY	Right Wrist Yaw
		RWR	Right Wrist Roll
		LSR	Left Shoulder Roll
		LSP	Left Shoulder Pitch
		LSY	Left Shoulder Yaw
		LEB	Left Elbow Pitch
		LWY	Left Wrist Yaw
		LWR	Left Wrist Roll
(e)			



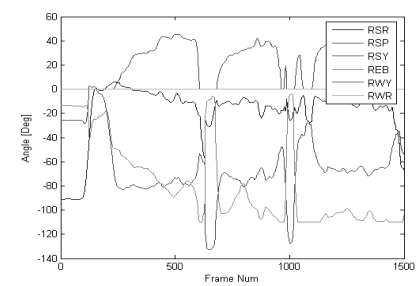
(a) right leg joint angle



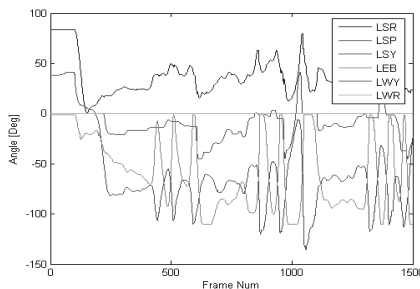
(b) left leg joint angle



(c) torso joint angle



(d) Right arm joint angle



(e) Right arm joint angle

Fig. 6 Optimized joint angle trajectories

2.6 최적화 결과

최적화 기법은 MATLAB에서 제공하는 fmincon 함수를 사용하였다. fmincon를 사용하면 등식 또는 부등식 형태의 제약조건을 갖는 비선형 목적함수를 최소화하도록 하는 관절각을 찾을 수 있다. 이를 위하여 2.5절에서 서술한 비용함수를 matlab 코드를 작성하고 fmincon을 호출하여 최적화 해를 구하였다. 이때 시간은 1/100단위로 반복적으로 해로 구하였고 그 결과를 Fig. 6에 나열하였다.

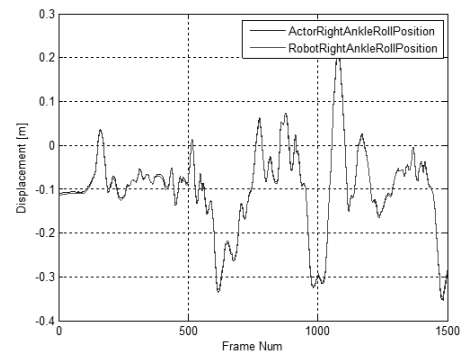
Fig. 6은 최적화된 관절각의 시간에 따른 변화 값을 보여준다. Table 2에서 Fig. 6의 도시된 관절각 이름을 정의하였다.

2.7 최적화 된 Robot 관절각 Data의 Error

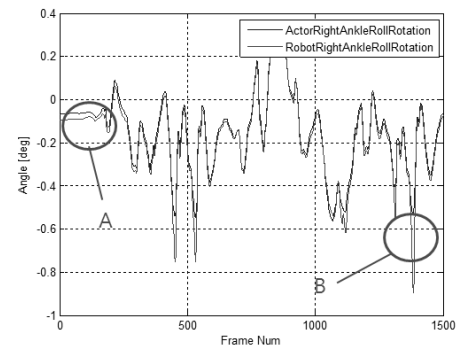
식 (8)의 비용함수를 최소화 하는 관절각을 로봇에 적용한 다음, capture되고 scaling된 인체 운동데이터 data와 robot 운동데이터 사이의 오차를 분석한 결과 다음과 같다.

본 논문에서는 많은 관절 중에 대표적으로 하체의 오른쪽 발목과 왼쪽 / 오른쪽 손목을 선정하였고, scaling된 인체와 로봇에 대하여 position과 orientation 값을 Fig. 7과 Fig. 8에 나타내었다.

Fig. 7(a)에서 Actor_Right_Ankle_Roll_Position은 모션빌더



(a) Right ankle joint position error of human and robot



(b) Right ankle joint Rotation error of human and robot

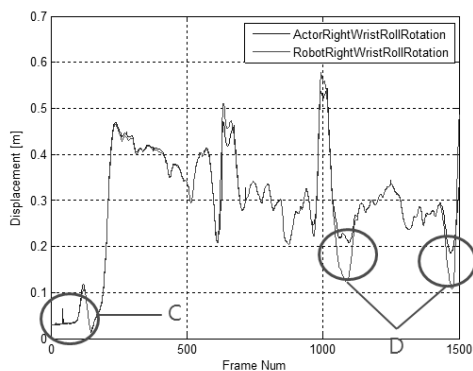
Fig. 7 Right ankle joint error of human and robot

에서 추출한 인체 오른쪽 발목 roll 관절 position data이고 Robot_Right_Ankle_Roll_Position은 로봇의 오른쪽 발목 roll 관절 position data이다. Fig. 7(a)에서 그래프는 error를 미세하게 볼 수 있고 최대 5 mm의 오차가 관찰되었다.

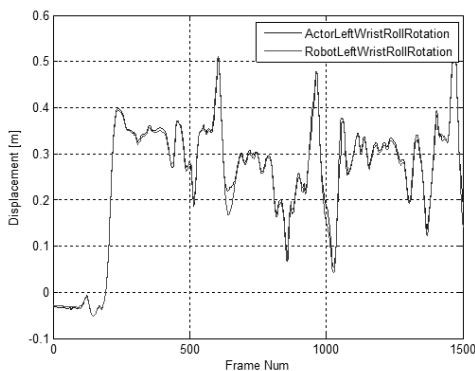
Fig. 7(b)는 Actor_Right_Ankle_Roll_Rotation은 모션빌더에서 추출한 인체 오른쪽 발목 roll 관절 Rotation data이고 Robot_Right_Ankle_Roll_Rotation은 로봇 오른쪽 발목 roll 관절 Rotation data이다. Fig. 7(b)에서 그래프에는 error가 Term A에서 0.03 deg, Term B에서 0.26 deg 오차가 발생하였다. termB에서의

오차는 simulation에서 하체가 빠른 움직임을 할 때 나타났다.

Fig. 8는 양쪽 손목의 position error대한 궤적이다. Fig. 8(a)에서 Actor_Right_Wrist_Roll_Position은 모션빌더에서 추출한 인체 오른쪽 손목 roll 관절 position data이고 Robot_Right_Wrist_Roll_Position은 로봇 오른쪽 손목 roll 관절 position data이다. Fig. 8(a)에서 영역 C부분에 0.04 m 오차가 발생하였고, 영역 D에서는 손을 뺄 때 0.1 m 오차가 발생하였다. 오차가 다소 크지만 로봇의 팔, 상체 길이비가 인체와 달라서, 팔을 뺄었을 때 발생하는 불가항력적 오차이며, 잠깐의 순간에만 나타나고 있다.



(a) Right wrist joint position error of human and robot



(b) Left wrist joint position error of human and robot

Fig. 8 Right and left wrist joint position error

Fig. 8(b)에서 Actor_Right_Wrist_Roll_Position은 모션빌더에서 추출한 인체 왼쪽 손목 roll 관절 position data이고 Robot_Right_Wrist_Roll_Position은 로봇 왼쪽 손목 roll 관절 position data이다. 왼쪽 손목에서 오차가 2 mm이하로 나타났다.

Fig. 7, Fig. 8의 결과를 보면 전반적으로 인체운동을 잘 추종하고 있으며, 부분적으로 차이가 나는 부분이 있지만 짧은 구간에서만 존재하여 로봇에 적용하기에 적합한 데이터로의 변환이 이루어졌다고 판단된다.

2.8 변환된 DATA에 대한 Kinematic Simulation

Kinematics simulation을 Fig. 9와 같이 진행을 하여 결과를 도출하였다. 최적화된(관절) data를 matlab에서 추출한 다음, matlab으로 작성된 Animation program, OpenGL로 작성된 Animation program을 확인한다. Fig. 10은 inventor simulation software에 입력하여 각각 kinematic simulation을 실시하였다. matlab, OpenGL과 inventor의 3가지 animation결과가 일치함을 확인 했고 향후 동적 simulation을 하는데 사용될 것이다.

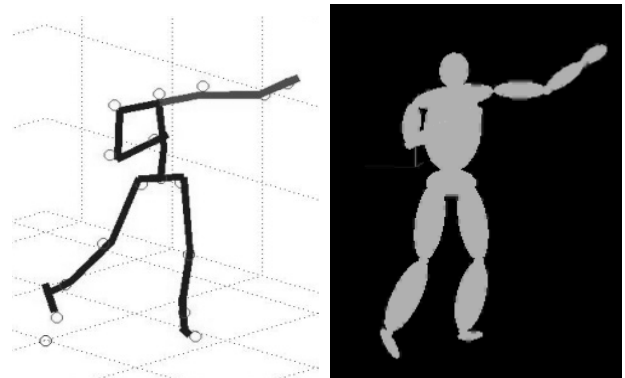


Fig. 9 OpenGL from the joint direction convert



Fig. 10 inventor simulation results

3. 결 론

본 논문은 motion capture 장비에서 추출된 인체 데이터를 추출하여 robot 관절 data로 변환하고, 비선형 최적화 기법을 적용하여, capture된 인체 관절 운동 데이터에 가장 근접하게 하는 로봇관절 각데이터를 얻었다. 그리고 최적화된 관절각 data를 바탕으로 matlab, OpenGL과 inventor의 3가지 animation을 이용하여 robot과 인체의 motion을 근접성을 확인하였으며, 향후 동적 simulation을 하는데 사용될 것이다.

후 기

이 연구는 서울과학기술대학교 교내학술 연구비 일부 지원으로 수행되었습니다.

References

- [1] Miura, H., Shimoyama, I., 1984, Dynamic Walk of a Biped, International Journal of Robotics Research, 3:3 60-74.
- [2] Kajita, S., Tani, K., 1996, Experimental Study of Biped Dynamic Walking, IEEE Control Systems. 16 13-19.
- [3] LEE, Y. S., JUNG, S. Y., PARK, I. M., KIM, M. S., HAN, S. H., 2012, A Study on Robust Walking Control of Biped Robot, KSMTE, 4 115-115.
- [4] GUN, S. J., 2008, On the Estimation of the center of mass an autonomous Bipedal Robot, robot - system, 14 886-892.
- [5] JO, Y. G., Eom, J. H., 2012, Dynamic Modeling for Stable Walking of Two-Leg Biped Robot, KSMTE. 4 136-136.
- [6] Hirai, K., Hirose, M., Haikawa, Y., Takenaka, T., 1998, The development of Honda humanoid robot , IEEE, 2 1321-1326.
- [7] Gubina, F., Hemami, H., McGhee, R. B., 1974, On the dynamic stability of biped locomotion, IEEE. 21 102-108.
- [8] Channon, P. H., Hopkins, S. H., Phan, D. T., 1992, Derivation of optimal walking motions for a biped walking robot, Robotica. 10:2 165-172