

작업자-로봇 상호작용을 위한 컴퓨터 비전 기반의 충돌 감지 시스템

이주석* · 이준수* · 민병권*

* 연세대학교 기계공학부

Collision detection system based on computer vision for Human-Robot Interaction

Ju Suk Lee*, Joon Soo Lee* and Byeong Kwon Min*

* School of Mechanical Engineering, Yonsei Univ.

Key Words: Collision Detection(충돌 감지), Computer Vision(컴퓨터 비전), Human Pose Estimation(사람 자세 추정), Human-Robot Cooperation(사람-로봇 협동)

초록: 4 차 산업 혁명 이후 더 유연하고 효율적인 생산을 위해 로봇과 작업자가 작업 공간을 공유하게 되었다. 그로 인해 로봇과 작업자 간 충돌이 발생할 수 있기에 이를 방지할 방법이 필요하다. 따라서 본 연구는 RGB 이미지와 깊이 이미지를 얻을 수 있는 Depth 카메라를 활용하여 작업자와 로봇의 충돌을 감지하고 신호를 반환하는 시스템을 제안한다. 제안한 충돌 감지 시스템은 Depth 카메라의 촬영된 이미지로부터 Human Pose Detector 를 통해 작업자의 주요 관절의 위치 정보를 구한다. 또한 로봇의 조인트 각도 데이터로부터 정기구학을 통해 각 로봇 링크들의 위치 정보를 구한다. 이후 구한 위치 정보를 기반으로 작업자와 로봇의 각 링크들에 OBB(Oriented Bounding Box) 방법으로 충돌을 감지할 기준인 경계 볼륨을 씌운다. SAT(Separated Axis Theorem)을 활용하여 경계 볼륨들 간 중첩을 감지하고, 이를 충돌로 간주하여 피드백 신호를 반환한다. 제안한 시스템의 정확도를 확인해보기 위해 작업자와 가상의 로봇 간 충돌 시뮬레이션을 진행하였으며 약 90%의 정확도를 나타냈다

Abstract: After the fourth industrial revolution, robots and operators have shared workspaces for more flexible and efficient production. As a result, collision between robots and the operators may occur, and a method to prevent such collision is needed. Therefore, this study proposes a system that detects collision between an operator and a robot and returns a signal by using a Depth camera that obtains RGB images and depth images. Through Human Pose Detector, the proposed collision detection system obtains the positional information, which key points of the operator, from the images taken by the Depth camera. In addition, the positional information of each robot link is obtained from the joint angle data of the robot through forward kinematics. Then a bounding volume, which is a criterion for detecting collisions, is applied to the operator and each link of robot using the OBB(Oriented Bounding Box) method. The SAT(Separated Axis Theorem) is used to return a feedback signal when bounding volumes are overlap. A collision simulation is conducted between an operator and a virtual robot to check the accuracy of the proposed system, which yields approximately 80%.

1. 서 론

산업용 로봇은 큰 작업 공간을 가지고, 목적에 맞게 로봇의 말단 장치를 자유롭게 교체할 수 있기에 유연성이 뛰어나다.⁽¹⁾ 그리고 작업자를 대신해 반복적이고 위험한 일을 할 수 있다.⁽²⁾ 따라서 산업용 로봇은 제조업 분야만 아니라 산업 전반에 걸쳐 점차 다양하게 활용되고 있다. 하지만 산업용 로봇이 보급됨에 따라 작업자가 로봇의 가동

범위에 들어가 부딪히는 등 관련 사고가 발생하고 있으며 다른 산업 사고들에 비해 작업자들이 당하는 부상의 강도가 2 배 이상 크다.⁽³⁾ 이에 작업자와 로봇 간의 안전한 협업 환경을 구축하기 위해서 작업자와 로봇의 작업 공간을 분리시키는 시스템들이 등장했다. 하지만 독일에서 주도한 제 4 차 산업 혁명과 미국에서 육성한 스마트 공장의 등장 이 더 유연하고 효율적인 생산을 추구하도록 요구 했다.⁽⁴⁾ 이러한 변화는 산업용 로봇들을 단순히 물건의 이동, 반복적인 행위에만 사용하는 것이 아

니라 사람과 로봇이 협업을 하는 방식으로 바뀌었다.

전통적으로 충돌을 감지하기 위해 사용한 방법은 작업자와 로봇의 작업공간을 분리 시키는 것이다. 대표적인 방식으로 GPS 와 mini INS(Inertial Navigation System)가 결합된 시스템을 활용한 방식⁽⁵⁾과 RFID(Radio-Frequency Identification)를 활용하는 방식⁽⁶⁾이 있다. GPS & mini INS 는 현재 위치를 2 차원적으로 인식하여 작업자와 로봇의 위치 정보를 얻고 둘 사이의 거리를 기준으로 충돌을 판단한다. 하지만 위치를 2 차원적으로 인식하기에 물체의 높이를 알 수 없다는 단점이 있다. RFID 는 신호를 발생하는 tag 장비와 이를 인식하는 reader 장비로 구성되어 있다. Tag 를 작업자들에게 부착시킨 후 충돌 가능 구역으로 들어오게 되면 reader 에서 이 신호를 인식하고 충돌을 파악한다. 하지만 RFID 는 작업자 수만큼 tag 장비를 확보해야 한다는 단점이 존재한다.^(5,6)

작업자와 로봇이 협업을 하게 되면서 전통적인 충돌 감지 시스템은 한계가 있어 작업자와 로봇이 작업 공간을 공유하는 환경에서 쓰일 충돌 감지 시스템이 필요했다. 그 중에 한 가지 방식은 카메라를 활용하는 것이고 여러 가지 연구가 존재한다. 여러 개의 카메라로 작업자와 로봇을 촬영한 뒤 RGB 이미지 상에서 둘의 충돌 여부를 판단하는 연구가 있다.⁽⁷⁾ 하지만 이 연구는 2D 이미지만을 이용하기 때문에 여러 대의 카메라가 필요하고 그로 인해 설치에 번거로움이 있다. 그리고 RGB 이미지와 픽셀의 깊이를 나타내는 깊이 이미지를 얻을 수 있는 2 대의 Depth 카메라로 작업자와 로봇을 각각 찍은 후, 깊이 이미지를 이용해 3D visualization 을 통해 충돌 여부를 판단하는 연구가 존재한다.⁽⁸⁾ 하지만 이 방식은 깊이 이미지만을 이용하기 때문에 햇빛이 너무 세면 감지가 잘 안되는 등 외란에 취약하다. 또한 한 표면을 여러 개의 depth 카메라로 비추면 그 표면의 감지 능력에 영향을 줄 수 있다. 그리고 3D visualization 하는데 시간이 오래 걸리기에 응답속도가 느리다는 단점이 존재한다.⁽⁸⁾

본 연구는 앞선 연구들의 한계점을 극복하고자 1 개의 Depth 카메라와 python 을 활용하여 작업자와 로봇의 충돌을 감지하고 충돌 시 로봇을 정지하며 신호를 반환하는 시스템을 구현하고자 한다. RGB 이미지와 depth 이미지를 동시에 사용하기에 외란의 영향을 덜 받을 수 있고 작업자의 3 차원 위치 정보도 얻을 수 있다. 또한 작업자가 별도의

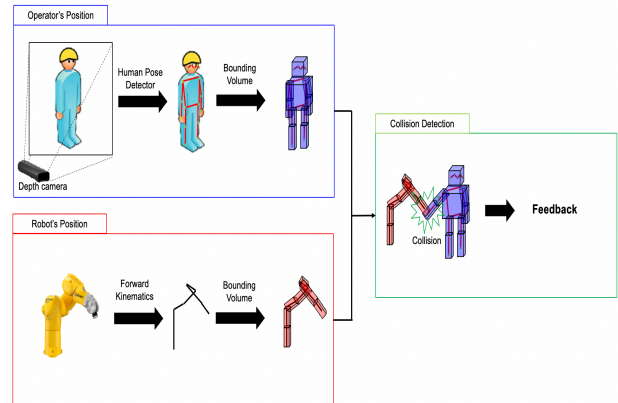


Fig. 1 Schematic modeling for collision detection

장비를 필요로 하지 않는다. 그리고 1 개의 Depth 카메라만 사용하기에 설치에 편의성도 존재한다. 충돌 여부를 visualization 과정 없이 python 내에서 계산하고 약 50 FPS 로 빠른 pose detector 모델을 사용하기에 응답속도가 빨라 기존 방식의 단점들을 극복할 수 있다.

2. 이론 및 연구방법

2.1 작업자 식별 및 주요 관절 추출

본 연구에서는 촬영된 이미지에서 사람을 식별하고 주요 관절을 추출하는 pose detection 을 위하여 human pose detector 모델로 MoveNet 오픈 소스를 활용하였다.

Human pose detector에는 사람을 탐지하고 관절의 위치를 찾고 자세를 추정하는 top-down 과 사람의 관절을 탐지하고 자세를 추정하는 bottom-up 방식이 존재한다. MoveNet 은 bottom-up 방식으로 사람을 탐지하는 과정이 생략되었기 때문에 AlphaPose 와 같은 다른 모델들에 비해 속도(FPS)가 빠르다.⁽⁹⁾ MoveNet 은 2 가지(Lightning, Thunder) 모델이 존재한다. Lightning 은 속도가 약 50FPS 로 빠르지만 정확도는 약 89.1%이다. 반대로, Thunder 는 속도가 약 30FPS 로 상대적으로 느리지만 정확도는 약 93.3%로 높다.⁽⁹⁾

본 연구는 충돌을 감지하고 로봇을 정지시키는 것으로 안전을 목적으로 하기에 실제 충돌과 충돌 감지 사이의 지연을 최소화 하기 위하여 연산속도가 빠른 MoveNet Lightning 모델을 선택하였다.

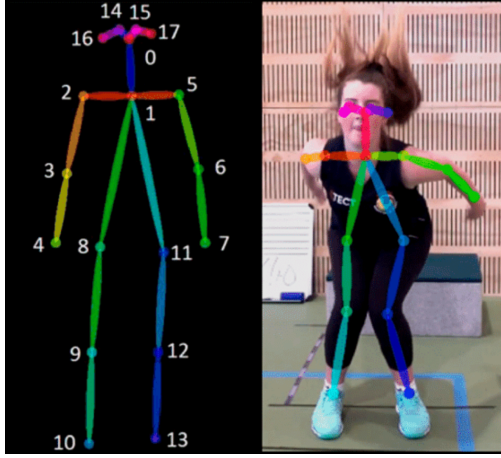


Fig. 2 Extraction of 17 keypoints in human using MoveNet Lightning

2.2 RGB 이미지와 Depth 이미지 맞추기

Depth 카메라는 RGB 카메라와 깊이 카메라 2개가 존재한다. 두 카메라의 다른 특성(광각, 위치, 해상도 등)으로 인해 깊이 이미지와 RGB 이미지 간에 1:1로 픽셀이 대응하지 않아 깊이 정보를 바로 읽어올 수 없다. 따라서 RGB 이미지에서 특정 픽셀이 갖는 깊이 값을 알아내기 위해서는 alignment 작업이 반드시 필요하다. 깊이 이미지와 RGB 이미지를 맞추는 alignment 작업을 거치게 되면 해당 픽셀에 대한 거리 정보를 정확하게 얻을 수 있다. 본 연구에선 Intel RealSense SDK의 라이브러리내 정의된 align 함수를 이용했다. 깊이 이미지를 align 함수를 이용해 RGB 이미지에 맞춘 후 불러왔다.

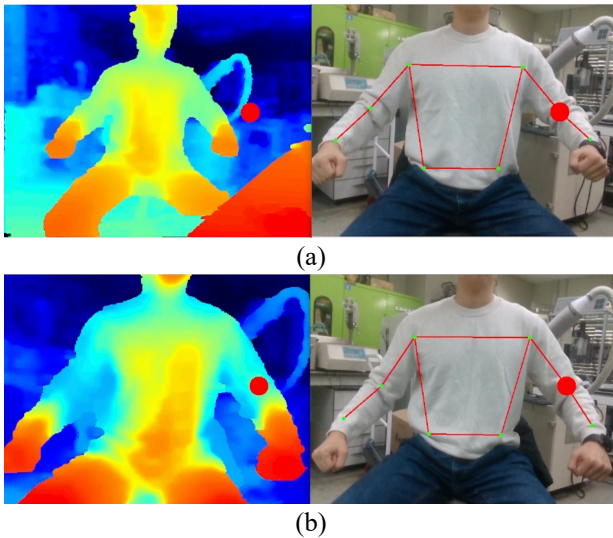


Fig. 3 (a) Without alignment (b) With alignment

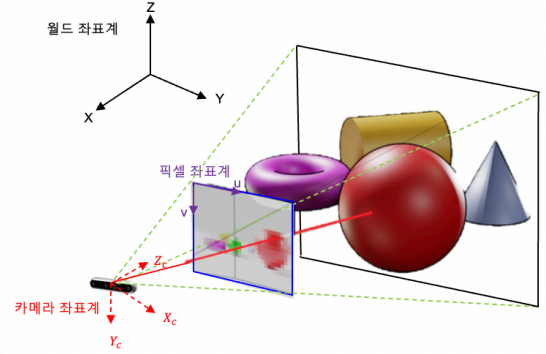


Fig. 4 Comparison of three coordinates system

2.3 픽셀 좌표계에서 카메라 좌표계로의 변환

작업자의 3D 위치 정보를 얻기 위해선 2D 좌표계인 픽셀 좌표계에서 3D 좌표계인 카메라 좌표계로의 변환이 필요하다. 실제 이미지는 사용된 렌즈, 렌즈와 센서와의 거리 등 카메라 내부의 기구적인 부분에 의해서 크게 영향을 받는다. 이러한 내부 요인을 제거해야만 정확한 3차원 좌표를 얻을 수 있다.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2)$$

여기서 f_x, f_y 는 focal length로 렌즈의 중심으로부터 카메라의 센서 촬상면까지의 거리, c_x, c_y 는 principal point로 카메라 렌즈 중심의 좌표, r 은 rotation으로 카메라의 회전, t 는 translation으로 카메라의 이동을 의미한다. 그리고 $[u, v, 1]$ 은 픽셀 좌표계, $[X, Y, Z, 1]$ 은 월드 좌표계, $[X_c, Y_c, Z_c, 1]$ 은 카메라 좌표계를 뜻한다. 본 연구에선 카메라가 고정되어 있기에 (1)식에서 카메라의 회전, 이동을 의미하는 Extrinsic properties matrix는 고려하지 않아도 된다. 따라서 초점거리와 주점을 의미하는 intrinsic properties matrix만 구하면 (2)식을 통해서 픽셀 좌표계를 카메라 좌표계로 변환할 수 있다.

2.4 정기구학을 통한 로봇의 위치 정보

로봇의 위치 정보는 정기구학을 통해서 얻을 수 있다. 로봇 매니퓰레이터 내의 연결된 링크 간의 관계를 수식적으로 나타내기 위해 D-H 매개변수를 구해 변환행렬(Transformation matrices)을 찾는다.⁽¹⁰⁾

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

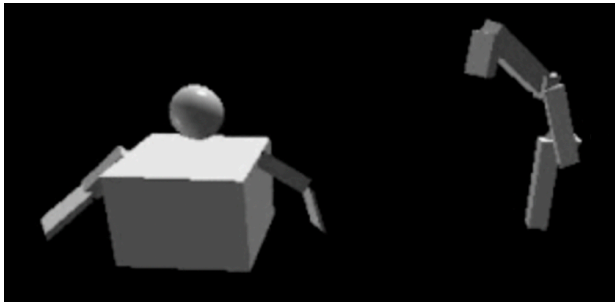
$${}^0T_N = {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{N-2}T_{N-1} {}^{N-1}T_N \quad (4)$$

(4)번 식을 구하게 되면 로봇의 베이스 좌표계 기준 링크들의 3 차원 좌표를 얻을 수 있다. 마지막으로 카메라와 베이스의 상대적 위치 정보만 주어진다면 카메라 좌표계 기준 로봇의 각 링크들의 3 차원 위치 정보들을 알 수 있다.

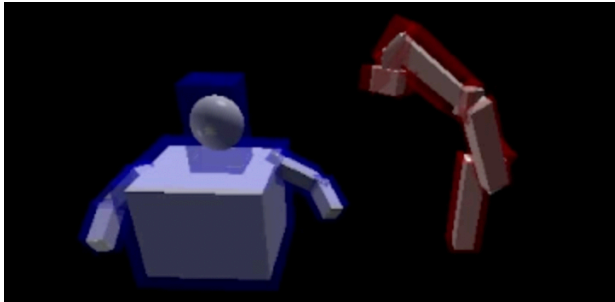
2.5 충돌 알고리즘

카메라 좌표계 기준 작업자와 로봇의 3 차원 위치 정보들을 얻은 후 각 위치들에 경계 볼륨을 씌운 후 볼륨들 간의 중첩 여부로 충돌을 판단할 것이다. 경계 볼륨이란 물체를 둘러싸고 있는 경계면으로 볼륨들 간의 중첩 여부를 판단해 collision detection에 쓰인다.⁽¹¹⁾ 경계 볼륨을 씌우는 방법에는 AABB(Axis-Aligned Bounding Box), OBB(Oriented Bounding Box) 등이 존재한다. OBB는 축이 회전하는 경계 볼륨을 씌우는 방식이다. 본 연구에서는 작업자의 주요 관절들과 로봇의 링크들은 모두 회전하기에 OBB 방식을 사용하였다.

경계 볼륨들 간의 충돌 판단은 분리축 이론 (SAT - Separating Axis Theory)을 기본으로 한다. 분리축 이론이란 '임의의 두 볼록 다면체 A, B에 대해 어떤 축이 존재해서 그 축으로의 다면체들의



(a)



(b)

Fig. 5 (a) Without a collision box (b) A blue collision box on the operator, a red collision box on robot

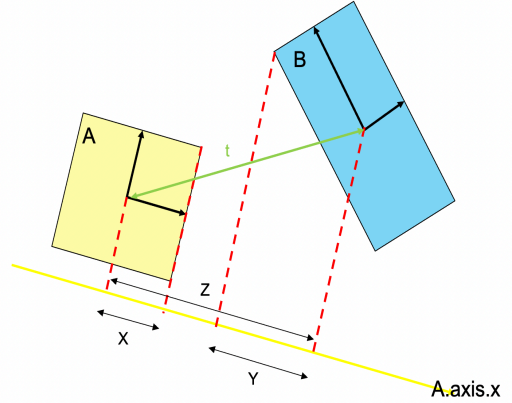


Fig. 6 Separating Axis Theorem

Table 1 Size of bounding volume

Unit: m	Width	Depth	Height
Head	0.2	0.2	0.3
Arm	0.15	0.15	0
Body	0.1	0.1	0

투영들의 구간이 서로 겹치지 않는다면 A, B는 서로 분리되어 있다'는 이론이다.⁽¹²⁾

$$X = \vec{A}_x \quad (5)$$

$$Y = |\vec{A}_x \cdot (\vec{B}_x + \vec{B}_y)| \quad (6)$$

$$Z = |\vec{A}_x \cdot \vec{t}| \quad (7)$$

Fig. 6에선 $X+Y < Z$ 이므로 A와 B가 A의 x축 방향에선 겹치지 않은 것이다. OBB 방식이므로 위와 같은 방식을 분리축 15개 각각에서 계산하여 직사각형 투영들의 구간이 겹쳐 있는지 확인하여 충돌을 판단한다.

그리고 생체 역학적 하중을 견뎌낼 수 있는 임계치는 신체 부위별로 다르다. 얼굴이 65로 등과 어깨가 210인 것에 비해 현저히 낮다.⁽¹³⁾ 따라서 본 연구에선 OBB 방식과 SAT를 python으로 구현하였고, 충돌이 발생하면 입게 될 피해 강도를 고려하여 **Table 1**에서 볼 수 있듯이 머리 부분에 경계 볼륨을 더 두껍게 설정하였다.

3. 시뮬레이션 및 결과

3.1 시뮬레이션 환경

본 연구에서는 노트북(Intel i7-9750H 2.6GHz, 16GB DDR4 RAM, NVIDIA GeForce RTX 2070 with MAX-Q Design 8GB)과 tensorflow-python=2.7.0 환경에서 시뮬레이션을 진행하였다. 시뮬레이션 상에서 가상의 로봇을 구현한다. 작업자와의 충돌 여부를 판단 후 충돌 시 로봇을 정지시키고 신호를

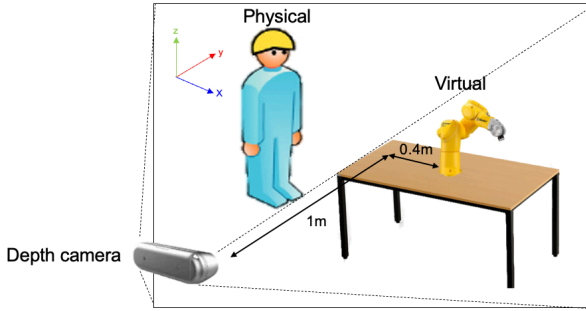


Fig. 7 Experimental setup for collision detection

반환할 것이다. 그리고 가상의 로봇이 카메라로부터 일정한 거리만큼 떨어져 pick and place 동작을 반복한다고 가정한다.

3.2 작업자의 위치 정보

카메라로 작업 환경과 작업자를 찍고 RGB 이미지에서 작업자를 식별하고 주요 관절 부위를 추출한다. RGB 이미지와 깊이 이미지를 맞춰 해당 픽셀에 대한 정확한 깊이 값을 구한다. 주요 관절의 픽셀 좌표 값을 카메라 좌표계 기준 값으로 변환한다.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 917.23 & 0 & 651.377 \\ 0 & 915.625 & 365.354 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (8)$$

카메라 좌표계 기준 주요 관절들의 위치 정보들을 이용해 작업자의 형상을 3D 시각화 시뮬레이션에 반영한다.

3.3 로봇의 위치 정보

MATLAB Simulink 에서 6 자유도 로봇 매니퓰레이터(TX2-40)를 로봇 형상, 제한 각도 등을 고려하여 모델링했다. Table 2 에 있는 r 은 공통 범선 사이의 거리, α 는 $x(i+1)$ 축을 따라서 $z(i)$ 에서 $z(i+1)$ 까지 회전한 각도, d 는 i 번째 좌표 시스템에서 $z(i)$ 축 방향으로 공통범선 까지의 거리, $\theta = x(i)$ 에서 $x(i+1)$ 까지 $z(i)$ 축을 따라서 회전한 각도이다.

D-H table 을 이용해 각 링크마다 식 (3), 식 (4) 를 세운다면 베이스 좌표계 기준 링크들의 3 차원 위치 정보를 얻을 수 있다. 이와 같은 정기구학을 python 에 구현하여 실시간으로 계산되게 했다. 다음으로 카메라와 로봇 베이스의 위치 차이를 고려하여 카메라 좌표계 기준 로봇의 3 차원 위치 정보를 얻었고 Fig. 9 에 나타냈다.

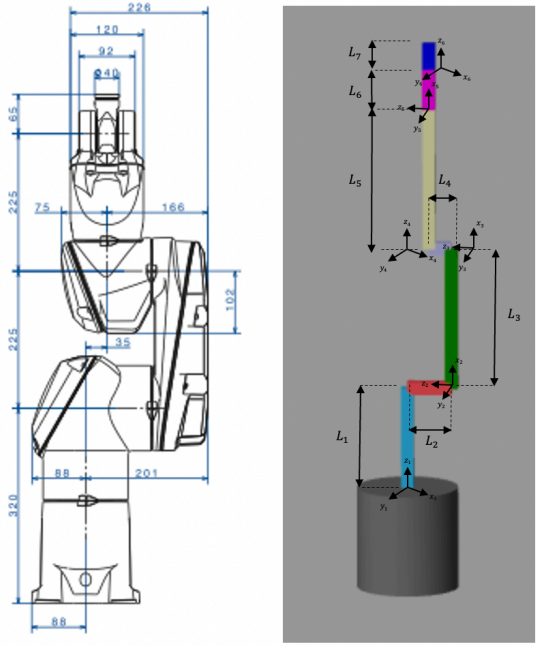


Fig. 8 TX2-40 modeling

Table 2 D-H table

i	r	α	d	θ
1	$L_2=0.07m$	90°	$L_1=0.24m$	θ_1
2	$L_3=0.225m$	0°	0	θ_2
3	0	-90°	$L_4=0.035m$	θ_3
4	0	90°	$L_5=0.225m$	θ_4
5	$L_6=0.065m$	-90°	0	θ_5
6	0	0	$L_7=0.04m$	θ_6

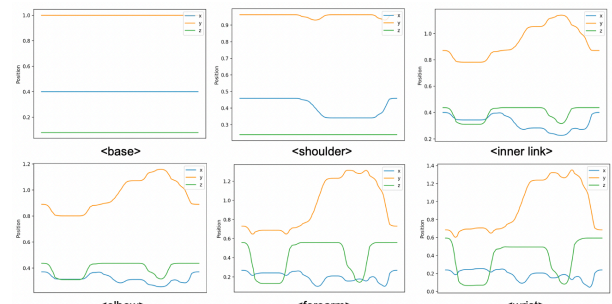


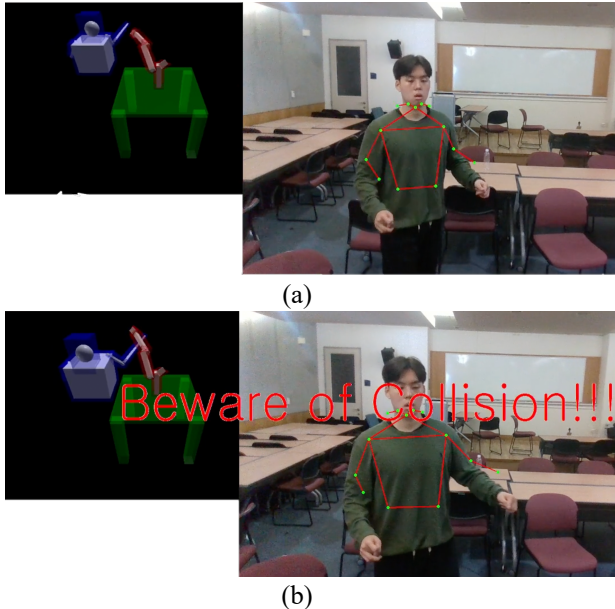
Fig. 9 The position of each link relative to camera coordinates

3.4 충돌 여부 판단

카메라 좌표계 기준 작업자와 로봇의 3 차원 위치 정보들을 바탕으로 경계 볼륨을 형성한다. 그리고 python 으로 구현한 OBB 방식을 통해 경계 볼륨들을 찍우고 SAT 를 활용해 볼륨들 간의 중첩을 감지하고 충돌 여부를 확인한다.

Table 3 Collision tests in various situations

	Accuracy
Case 1	47/50
Case 2	44/50

**Fig. 10** (a) Before colliding with a virtual robot (b) Detecting collision, stop a virtual robot and give a feedback signal

3.5 시뮬레이션 결과

Pick and place 동작을 수행하는 가상의 로봇과 작업자와의 충돌을 잘 탐지하는 것을 확인했다. 본 논문에서 제시한 방식의 정확도를 나타내기 위해 여러 가지 상황에서 충돌 여부를 정확하게 파악하는 실험을 해보았다. 발생 재해 형태별 재해 발생 건수를 보면 끼임, 부딪힘이 가장 많다.⁽¹⁴⁾ 따라서 Case 1은 구조물 옆에서 멈춰 있는 작업자에게 로봇이 접근했을 때, Case 2는 움직이는 작업자가 로봇에 접근했을 때로 나누어서 실험을 진행하였다. 각 case 별로 50 번 진행하여 충돌 여부를 정확히 감지 했는지를 정확도로 나타냈다. 시뮬레이션 결과 약 90%의 정확도를 나타냈다.

4. 결 론

인간-로봇 협업 시스템에선 무엇보다 안전이 중요하다. 그렇기에 본 연구에서는 1 개의 Depth 카메라를 활용하여 작업자가 로봇의 구동 범위 내로 들어왔을 때 작업자와 로봇의 충돌을 감지하고 신호를 반환하는 시스템을 구현하였다. 이 시스템은

노트북 환경에서 진행하였음에도 약 14 FPS 정도의 응답속도를 나타냈으며, 필요에 따라 경계 볼륨의 크기를 조절할 수 있어 유연성이 높고, 1 개의 카메라의 위치만 조절하여 사용하면 되기에 편의성도 높다.

5. 한계 및 후속 연구

시뮬레이션 결과 정확도가 약 90%가 나왔다. 이는 정확도보다 속도를 우선시하여 사람의 주요 관절을 추출하는 MoveNet Lightning 모델의 한계로 인해 사람 말단 부위의 위치 정보를 항상 정확하게 얻을 수 없었기 때문이다. 이로 인해 일어나지 않은 충돌을 인식해 신호를 반환하는 경우가 존재하였다. 특히 case 2 와 작업자가 움직일 때 말단 부위의 길이가 원래보다 늘어나는 현상이 발생하였다. 그렇기에 충돌 하지 않았을 때 충돌로 인식하는 경우는 있지만, 충돌을 인식하지 못하는 경우는 존재하지 않았다. 불필요한 가동 중지 시간을 유발 했으나 안전상의 문제는 없었다. 향후에 이것은 필터를 한 층 적용함으로써 해결할 수 있을 것이다. 사람의 주요 관절 위치들의 속도를 계산함으로써 일정 값 이상의 속도를 가진 위치를 표시 하지 않으며 충돌 계산에서 제외하는 방식이다. 마지막으로 본 연구에선 실제 로봇이 아닌 가상의 로봇으로 충돌 시뮬레이션을 진행하였다. 따라서 이후에 제안된 시스템을 실제 환경에 적용해 보고 싶다.

참고문헌

(References)

- (1) Kim, S. H., Nam, E., Ha, T. I., Hwang, S. H., Lee, J. H., Park, S. H., & Min, B. K. (2019). Robotic machining: A review of recent progress. *International Journal of Precision Engineering and Manufacturing*, 20(9), 1629-1642.
- (2) Robla-Gómez, S., Becerra, V. M., Llata, J. R., Gonzalez-Sarabia, E., Torre-Ferrero, C., & Perez-Oria, J. (2017). Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access*, 5, 26754-26773.
- (3) Korea Occupational Safety and Health Agency Occupational Safety and Health Research Institute.
- (4) Thoben, K. D., Wiesner, S., & Wuest, T. (2017). "Industrie 4.0" and smart manufacturing-a review of research issues and application examples. *International journal of automation technology*, 11(1), 4-16.
- (5) Wang, J., & Razavi, S. N. (2016). Low false alarm rate

- model for unsafe-proximity detection in construction. *Journal of Computing in Civil Engineering*, 30(2), 04015005.
- (6) Chae, S., & Yoshida, T. (2010). Application of RFID technology to prevention of collision accident with heavy equipment. *Automation in construction*, 19(3), 368-374.
- (7) Ferraguti, F., Landi, C. T., Costi, S., Bonfè, M., Farsoni, S., Secchi, C., & Fantuzzi, C. (2020). Safety barrier functions and multi-camera tracking for human-robot shared environment. *Robotics and Autonomous Systems*, 124, 103388.
- (8) Schmidt, B., & Wang, L. (2014). Depth camera based collision avoidance via active robot control. *Journal of manufacturing systems*, 33(4), 711-718.
- (9) Jo, B., & Kim, S. (2022). Comparative Analysis of OpenPose, PoseNet, and MoveNet Models for Pose Estimation in Mobile Devices. *Traitement du Signal*, 39(1), 119-124.
- (10) Paden, B. E. (1985). Kinematics and control of robot manipulators. Ph. D. Thesis.
- (11) Safeea, M., Mendes, N., & Neto, P. (2017). Minimum distance calculation for safe human robot interaction. *Procedia Manufacturing*, 11, 99-106.
- (12) Huynh, J. (2009). Separating axis theorem for oriented bounding boxes.
- (13) Hyungsoo Ahn. "Safety Study According to Collaborative Robot Collision Speed." Domestic master's thesis Korea University of Technology and Education, 2018. Chungcheongnam-do