Journal of Institute of Control, Robotics and Systems (2025) 31(8):927-933
https://doi.org/10.5302/J.ICROS.2025.25.0094 ISSN:1976-5622 eISSN:2233-4335

RGB 카메라를 이용한 로봇 매니퓰레이터 및 핸드-아이 캘리브레이션

Calibration of Robot Manipulator and Hand-eye Using RGB Camera

이 지 용, 유 범 재*
(Jiyong Lee[1] and Bum-Jae You[1,*])
[1]Center for humanoid research, Korea Institute of Science and Technology

Abstract: This paper proposes a robot manipulator and hand–eye kinematic calibration method using an RGB camera attached to an end effector. For in-hand configuration, sensing objects with the camera allows for simultaneous kinematic calibration of the manipulator and hand–eye calibration, which is more effective than performing them separately. Thus, the camera measures the chessboard's 2D corners at different robot arm postures. These corners are reprojected to all images, and a calibration is performed with a bundle adjustment method by reducing 2D reprojection errors. Additionally, the chessboard is placed at multiple positions so that the manipulator can have any posture without limitation. The transformation between chessboard positions is included in the optimization process. This method is validated through exploratory experiments, with results compared to a laser tracker, demonstrating its effectiveness.

Keywords: kinematic calibration, hand-eye calibration, bundle adjustment, RGB camera, robot manipulator

I. 서론

로봇은 위험한 환경에서나 단순 반복 작업에서 사람을 대체할 수 있는 수단으로써 공장 자동화의 도구로써 활용되어 생산성을 높이기 위해 사용되었다. 최근에는 반복적인 pick & place와 같은 전통적인 작업을 벗어나 일상 생활에서 사람을 대신해 사물을 인식하고 사람과 같이 환경에 유연하게 대응하며 작업을 수행하는 단계에 이르렀으며, 이와 같은 연구 방법이 전세계적인 관심을 받고 있다[1].

작업의 복잡도에 따라 수행하는 방법은 다르겠지만, 작은 오차에도 물체를 잡거나 조작하는데 실패할 수 있는 가능성이 있기 때문에 작업의 성공률을 높이는 가장 기본적인 요소는 로봇 매니퓰레이터(robot manipulator)의 엔드이펙터(end effector)가 작업을 수행하기 위한 위치로 정확하게 이동할 수 있게 하는 것이다. 로봇은 이상적으로 원하는 위치로 움직이도록 설계되지만, 가공 및 조립하는 중에서 오차가 발생될 수 있으며, 동작할 현장에 설치하는 과정이나 오랜 사용으로 인한 노후화로 인해 추가적인 오차가 발생할 수 있다. 이러한 오차는 캘리브레이션 과정을 통해 보정될 수 있다.

로봇 캘리브레이션은 위에서 언급한 여러가지 이유로 발생할 수 있는 계산상의 오차를 줄이기위해 로봇의 파라미터를 조정하는 과정이다. 일반적으로 로봇의 엔드이펙터의 위치를 측정한 후, 로봇의 파라미터로 계산된 위치와 일치하도록 최적화 과정을 통해서 파라미터를 조정하게 된다. 이때, 엔트이펙터의 위치를 정확하게 측정하기 위해 레이저 트래커(laser tracker)를 사용하는 캘리브레이션 방법론이 많이 사용된다[2]. 이 방법은 측정 정밀도가 우수하고 작은 크기의 마커를 측정하기 때문에 상대적으로 넓은 작업 공간에 대한 측정이 가능하지만, 고가의 장비이고 산업현장 등 공간이 확보되지 않는 곳에서는 사용하기 어렵다.

이를 해소하기 위해 카메라와 체스보드를 이용한 캘리브레이션 방법들이 연구되었다[3,4]. 체스보드 패턴과 로봇의 엔드이펙터에 부착된 카메라의 사용은 넓은 공간을 요구하지 않으며, 비용적인 부담도 적다. 하지만, 고정된 위치의 체스보드는 카메라가 체스보드를 응시해야 하므로 로봇의 움직임을 제한할 수 있다. 이를 극복하기 위해 체스보드를 여러 곳에 부착하여 캘리브레이션을 수행하기도 하지만, 같은 위치의 체스보드에 대해 측정된 데이터 간 오차만을 고려하여 로봇의 큰 움직임 사이의 오차는 고려되지 않는 단점이 있다.

본 연구에서는 임의의 여러 위치로 체스보드를 옮기면서 측정한 데이터를 모두 함께 사용하는 로봇 팔 캘리브레이션 방법을 제안한다. 로봇의 카메라가 응시할 수 있는 곳에 체스보드를 옮겨 놓음으로써 로봇 팔이 움직일 수 있는 범위에 대한 제한을 없애고, 서로 다른 위치에서 측정된 체스보드 간의 오차를 계산할 수 있어서, 모든 작업 공간을 고려한 캘리브레이션이 가능하다는 장점이 있다.

II. 관련 연구

로봇 캘리브레이션은 카메라 캘리브레이션, 핸드-아이 캘리브레이션, 로봇 팔 캘리브레이션으로 나눌 수 있다[5].

* Corresponding Author
Manuscript received April 14, 2025; revised June 2, 2025; accepted July 1, 2025
이지용: 한국과학기술연구원 휴머노이드연구단 박사후연구원(jiyonglee@kist.re.kr, ORCID 0000-0003-3156-6501)
유범재: 한국과학기술연구원 휴머노이드연구단 책임연구원(ybj@kist.re.kr, ORCID 0009-0008-7749-5570)
※ 본 연구는 과학기술정보통신부(MSIT) 한국연구재단의 지원을 받아 연구되었음(No. 2022M3C1A3098746).

카메라 캘리브레이션은 카메라 렌즈에 있는 왜곡된 정보를 보정하고 카메라의 내부 파라미터와 여러 개의 카메라를 사용할 경우 카메라 간의 관계인 외부 파라미터를 찾는 절차이다. 특별한 패턴을 사용하는 방법[6]이 대표적이며, 파라미터를 추가하여 더 복잡한 카메라 모델을 이용한 방법들이 제안되었다[7,8].

로봇을 움직이는 명령들은 일반적으로 로봇의 기준 좌표계를 기준으로 계산된다. 카메라를 이용하여 물체를 측정하고 로봇을 이용하여 측정된 물체를 조작하기 위해서는 카메라 좌표계에서 측정된 정보를 로봇 좌표계의 정보로 변환이 가능해야 한다. 카메라는 로봇의 외부나 로봇의 엔드이펙터에 근처에 부착하는 것이 일반적이다. 외부에 카메라를 배치할 경우에는 카메라에서 측정된 정보를 로봇의 기준 좌표계로 변환하는 행렬이 필요하고, 로봇의 엔드이펙터에 근처에 부착할 경우에는 카메라에서 엔드이펙터의 좌표계로 변환하는 행렬이 필요하다. 핸드-아이 캘리브레이션은 카메라 좌표계에서 측정된 정보를 로봇 좌표계로 변환할 수 있도록 로봇과 카메라 간의 변환 관계를 찾는 절차이다. 일반적으로 로봇에 핸드-아이 캘리브레이션을 위한 링크를 추가하고 카메라 캘리브레이션과 같이 외부 패턴을 이용하여 수행한다[9,10].

로봇은 설계상의 공칭 파라미터(nominal parameter)로 가공되어 조립되지만, 실제로는 가공 과정 중에 발생된 오차, 조립과정에서 발생된 오차, 운반 및 노후화로 인한 오차 등 다양한 이유로 인해 의도된 동작과 다르게 움직일 수 있다. 로봇 팔 캘리브레이션에서는 이러한 오차를 보정하고 의도된 동작을 하도록 설계상의 공칭 파라미터(nominal parameter)를 수정하여 로봇 엔드이펙터의 실제 자세와 계산된 자세가 일치하도록 하는 파라미터를 찾는 과정이다. 가장 널리 사용되는 방법은 레이저 트래커를 사용하는 방법이다[11].

III. 배경 지식

로봇 캘리브레이션을 위한 카메라 모델링, 핸드-아이 모델링, 로봇 팔 기구학 모델링 및 최적화 과정에서 구해야 할 파라미터들을 다음과 같다.

1. 카메라 모델링

카메라의 왜곡(distortion)은 방사왜곡(radial distortion)과 접선왜곡(tangential distortion)으로 이루어진다. 방사왜곡은 렌즈의 굴절률에 의해 발생하고, 접선왜곡은 카메라의 제조 과정에서 렌즈와 CCD센서의 수평이 맞지 않거나 렌즈 자체의 중심이 맞지 않아 발생하게 된다. 카메라 캘리브레이션을 위해서 Brown-Canrady 왜곡 모델을 사용하였으며, 식 (1)과 같이 표현된다.

$$x_d = (k_1 r^2 + k_2 r^4 + k_3 r^6)x_u + (2p_1 x_u y_u + (p_2(r^2 + 2x_u^2))$$
$$y_d = (k_1 r^2 + k_2 r^4 + k_3 r^6)y_u + (2p_1 x_u y_u + (p_2(r^2 + 2y_u^2))$$
$$where\ r = \sqrt{x_u^2 + y_u^2} \tag{1}$$

x_d, y_d는 왜곡이 있는 이미지 좌표이며, x_u, y_u는 왜곡이 없는 이미지 좌표이다. k_1, k_2, k_3 은 방사왜곡 파라미터이며, p_1, p_2는 접선왜곡 파라미터이다. 렌즈 왜곡이 없다고 가정했을 경우 카메라 모델은 식 (2)와 같이 표현할 수 있다.

$$x_u = f_x \frac{X}{Z} + c_x$$
$$y_u = f_y \frac{Y}{Z} + c_y \tag{2}$$

f_x, f_y는 초점거리(focal length), c_x, c_y는 이미지 중심(principle point), X, Y, Z 는 3차원 좌표이다. 최적화과정을 통해 $f_x, f_y, c_x, c_y, k_{1,2,3}, p_{1,2}$를 구하게 된다.

2. 핸드-아이 모델링

로봇 팔의 말단 좌표계와 카메라 좌표계 사이의 관계 (T_c^{ee})는 일반적으로 하나의 확장된 링크로 표현된다. 그러므로 두 좌표계 간의 관계는 식 (3)과 같이 6개의 파라미터로 표현된다.

$$T_c^{ee} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$
$$R = R_z(\gamma)R_y(\beta)R_x(\alpha), t = (t_x, t_y, t_z) \tag{3}$$

α, β, γ는 각각 x축, y축, z축에 대한 회전 각도이고, t_x, t_y, t_z는 x축, y축, z축에 대한 이동한 정도를 나타낸다. 최적화 과정에서 6개의 파라미터를 구하게 된다.

3. 로봇 팔 기구학 모델링

로봇 팔을 캘리브레이션하기 위해서 각 조인트에 좌표계를 설정하고 링크를 수학적으로 표현하여야 한다. 이를 위해 수정 D-H (modified Denavit-Hartenberg) 파라미터 표기법을 사용하였다[12]. 4개의 파라미터를 사용하여 각 조인트와 링크 간의 상대적인 변환 관계 (T_n^{n-1})를 식 (4)와 같이 표현할 수 있다.

$$T_n^{n-1} = \begin{bmatrix} c\theta_n & -s\theta_n & 0 & a_{n-1} \\ s\theta_n c\alpha_{n-1} & c\theta_n c\alpha_{n-1} & -s\alpha_{n-1} & -d_n s\alpha_{n-1} \\ s\theta_n s\alpha_{n-1} & c\theta_n s\alpha_{n-1} & c\alpha_{n-1} & d_n c\alpha_{n-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$
$$, where\ c = cos, s = sin$$

각 링크는 (a, α, d, θ)의 4개의 파라미터로 구성된다. 로봇의 설계도나 명세서에는 이 값들의 설계상의 수치(nominal parameters)가 정해져 있으나, 조립이나 운반 등의 이유로 로봇에 비틀림이 발생하여 실제로는 약간의 오차가 발생한다. 그러므로 실제 파라미터의 값은 식 (5)와 같이 표현할 수 있다.

$$\tilde{a} = a + \Delta a$$
$$\tilde{\alpha} = \alpha + \Delta\alpha$$
$$\tilde{d} = d + \Delta d$$
$$\tilde{\theta} = \theta + \Delta\theta \tag{5}$$

로봇이 정확하게 동작하기 위해서는 설계상의 수치(nominal parameter (a, α, d, θ))에 추가로 비틀림 정도를 나타내는 오프셋($\Delta a, \Delta\alpha, \Delta d, \Delta\theta$)을 더해주어야 원하는 동작을 할 수 있다. 그러므로 캘리브레이션 과정에서 구해야 할 파라미터는 $(\Delta a, \Delta\alpha, \Delta d, \Delta\theta)$이다.

IV. 제안하는 방법

카메라가 로봇 팔의 엔드이펙터 근처에 부착되어 있는 경우, 한 곳에 놓여 있는 체스보드를 측정하기 위해서는 카메

라가 체스보드를 응시하도록 유지해야 되므로 로봇 팔의 움직임을 제한하게 되고, 로봇 팔의 작업공간(work space)을 모두 포함할 수 없게 된다. 이를 극복하기 위해 여러 다른 곳에 체스보드를 놓고 측정하여 작업공간을 최대한 포함시켜 이러한 문제를 해결하려고 하지만, 이 경우에도 한 이미지로 서로 다른 위치에 있는 체스보드를 동시에 측정할 수 없기 때문에, 직접적으로 대응점을 찾을 수가 없어 최적화 과정에서 서로 다른 위치에 있는 체스보드 간의 관계를 활용하지 못하게 된다. 이러한 단점을 극복하기 위하여, 하나의 이미지로 동시에 측정이 불가능한 곳에 놓여 있는 체스보드 간의 변환 관계를 찾고 이를 최적화에 활용할 수 있는 방법을 제안한다.

전체 캘리브레이션은 [5]에서와 같이, 카메라 캘리브레이션, 핸드-아이 캘리브레이션, 로봇 팔 캘리브레이션 등 세 가지 캘리브레이션을 수행한다. 최적화는 번들 조정(bundle adjustment)를 사용하여 수행하였다.

1. 여러 위치에서 수집된 데이터의 활용 방법

체스보드를 하나의 특정 위치에 놓고 수집된 데이터를 이용한 캘리브레이션은, 그림 1의 왼쪽 그림과 같은 경우로, 로봇의 서로 다른 자세에서 측정된 체스보드의 코너 포인트의 값은 모두 동일한 위치의 값이므로 식 (6)과 같은 방법으로 최적화를 통해 수행할 수 있다.

$$\Delta a, \Delta\alpha, \Delta d, \Delta\theta = argmin \sum_i \sum_j \left(T_{ee}^{b}(q_i) T_c^{ee} P^{c_i} - T_{ee}^{b}(q_j) T_c^{ee} P^{c_j}\right)^2 \quad (6)$$

즉, 로봇의 여러 자세에서 얻어진 조인트 값(q)과 그 자세에서 획득한 체스보드의 카메라 좌표계에서의 3차원 포인트(P^c)가 주어졌을 때, 카메라에서 엔드이펙터로의 변환행렬(T_c^{ee}) 과 엔드이펙터에서 로봇좌표계로의 변환행렬($T_{ee}^{b}(q)$)을 이용하여 로봇 좌표계로 변환했을 경우에 모두 같은 값을 가져야 한다. 하지만, 이 경우에는 카메라가 체스보드를 주시해야 하므로 로봇 자세의 범위가 제한될 수 있다. 로봇 자세에 대한 제한을 없애기 위해서는 그림 1의 오른쪽 그림처럼 체스보드의 위치를 움직이면 되는데 이 경우에는 식 (6)과 같은 최적화를 수행하지 못 한다. 서로 다른 위치에 놓인 체스보드 간의 변환 행렬은 동일한 이미지 내에 존재하지 않기 때문에 직접적으로 구할 수 없다. 이 문제를 해결하기 위해서, 그림 2의 왼쪽 그림과 같이 로봇의 위치는 동일한데 체스보드의 위치가 달라진 것은 그림 2의 오른쪽 그림과 같이 체스보드는 동일한 위치에 놓고 서로 다른 로봇의 위치에서

그림 1. 체스보드 위치 설정 (한 곳(왼쪽), 여러 곳 (오른쪽)).
Fig. 1. The configuration of chessboard positions (one position (left), multiple position (right)).

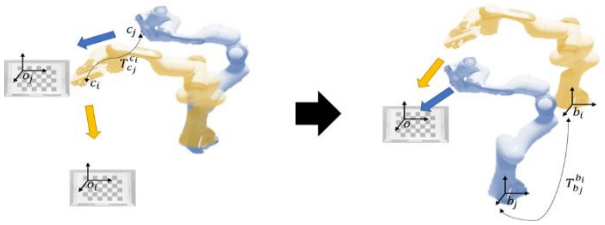

그림 2. 체스보드 위치와 로봇 기준 좌표계의 관계.
Fig. 2. The relation between multiple chessboard locations and multiple robot locations.

체스보드를 측정하는 것과 같다는 점에 착안하였다. 그에 따라 서로 다른 체스보드 간의 변환행렬은 동일한 체스보드를 측정하는 서로 다른 로봇 위치 간의 변환행렬로 대체할 수 있다.

이상적인 상황에서 α위치에 놓인 체스보드를 i번째 로봇 자세($q_{\alpha,i}$)에서 측정된 카메라 좌표계의 코너 포인트($p^{c_{\alpha,i}}$)와 β위치에 놓인 체스보드를 j번째 로봇 자세($q_{\beta,j}$)에서 측정된 카메라 좌표계의 코너 포인트 ($p^{c_{\beta,j}}$)가 주어졌을 때 두 데이터 간의 변환은 식 (7)과 같이 표현할 수 있다.

$$T_{ee}^{b}(q_{\alpha,i}) T_c^{ee} P^{c_{\alpha,i}} = T_{b_\beta}^{b_\alpha} T_{ee}^{b}(q_{\beta,j}) T_c^{ee} P^{c_{\beta,j}} \quad (7)$$

실제로는 서로 다른 위치의 체스보드의 값을 이용하지만, 위의 식을 계산할 때는 같은 체스보드를 서로 다른 위치에서 측정한다고 생각하기 때문에 각각의 경우의 로봇 좌표계(b_α, b_β)간의 변환 행렬을 통해 하나의 공통된 좌표계로 변환이 가능하다. 위 식을 활용하여 서로 다른 위치의 체스보드 측정 값을 포함한 최적화는 식 (8)과 같이 수행할 수 있다.

$$\begin{aligned} \Delta a, \Delta\alpha, \Delta d, \Delta\theta &= argmin \sum_\alpha \sum_\beta \sum_i \sum_j (P^{c_{\alpha,i}} - M P^{c_{\beta,j}})^2 \\ M &= \left(T_{ee}^{b}(q_{\alpha,i}) T_c^{ee}\right)^{-1} T_{b_\beta}^{b_\alpha} T_{ee}^{b}(q_{\beta,j}) T_c^{ee} \end{aligned} \quad (8)$$

로봇 위치 간의 변환행렬인 $T_{b_\beta}^{b_\alpha}$는 로봇 위치 b_α의 i번째 자세의 체스보드를 로봇 좌표계로 변환한 값 ($P^{b_{\alpha,i}} = T_{ee}^{b}(q_{\alpha,i}) T_c^{ee} P^{c_{\alpha,i}}$)과 로봇 위치 b_β의 j번째 자세에서의 체스보드를 로봇 좌표계로 변환한 값($P^{b_{\beta,j}} = T_{ee}^{b}(q_{\beta,j}) T_c^{ee} P^{c_{\beta,j}}$)을 이용하여 구한다. 식 (7)을 통해서 구하는 것도 가능하지만, 이 연산에서 생기는 에러 값을 최적화에 더 잘 반영하기 위해서 일대다 관계를 활용하였다. 한 곳의 체스보드에서 수집된 체스보드 데이터($I = \{P^{b_{\alpha,i}} | i = 1 \dots n\}$)와 또 다른 위치에서의 체스보드 데이터($J = \{P^{b_{\beta,j}} | j = 1 \dots m\}$)가 주어지면, 하나의 체스보드 코너 값 $P^{b_{\alpha,i}}$와 여러 개의 체스보드 코너 값 J로 Umeyama 알고리즘[13]을 사용하여 변환 관계를 계산하였다. 일대일 관계로 계산할 경우에는 오차가 존재하지 않고, 다대다 관계로 계산할 경우 여러 데이터 간의 오차가 상쇄되어 전체 오차가 완화된다. 서로 다른 위치에서 측정된 데이터를 이용하여 로봇 위치 간의 변환행렬에 포함되어 있는 오차를 효과적으로 최적화 과정에 반영하기 위해 일대다 관계를 이용한 계산 방식을 제안하였으며, 이 방법을 사용하면

여러 개의 데이터에 대한 개수 제한이 없기 때문에 각 위치에서 측정된 데이터의 수가 같지 않아도 최적화를 수행할 수 있게 된다.

2. 번들 조정(bundle adjustment)을 이용한 캘리브레이션

번들 조정은 여러 개의 2D 이미지에서 추출한 특징점과 이들 특징점에 대응하는 3D 공간 상의 점들 사이의 일관성을 유지하면서 최적화를 수행하는 방법이다. 본 연구에서 최적화 과정을 통해서 구해야 할 파라미터는 카메라 내부 파라미터, D-H 파라미터 오프셋($\Delta a, \Delta\alpha, \Delta d, \Delta\theta$), 카메라와 엔드이펙터 간의 변환 행렬(T_c^{ee})로써, 내부 파라미터 9개 ($f_x, f_y, c_x, c_y, k_1, k_2, k_3, p_1, p_2$), D-H 파라미터 오프셋 23개 ($4 \times (\text{링크수} - 1) - 1$), T_c^{ee} 파라미터 6개(회전 3개, 이동 3개), 그리고 카메라와 체스보드의 변환 행렬($6 \times K$(이미지 수))로 총 $38 + 6 \times K$개이다. D-H 파라미터는 하나의 링크에 대한 변환 행렬을 4개의 파라미터로 구성하는데 중복성(Redundancy) 문제로 첫 번째 링크와 마지막 링크의 마지막 파라미터를 제외하였다. 체스보드와 카메라 간의 변환 행렬을 구하기 위해서 체스보드 3D모델을 사용하였다. 최종적으로 최적화에 사용할 오차함수는 식 (9)와 같다.

$$
\begin{aligned}
&f_x, f_y, c_x, c_y, k_1, k_2, k_3, p_1, p_2, \Delta a, \Delta\alpha, \Delta d, \Delta\theta, T_{0\ldots K} \\
&= argmin \sum_{b_a} \sum_{b_b} \sum_{i} \sum_{j} \left(f\left(x_d^{c_{\alpha,i}}\right) - Mf\left(x_d^{c_{\beta,j}}\right) \right)^2 \\
&M = \left(T_{ee}^{b}\left(q_{b_\alpha,i}\right) T_c^{ee}\right)^{-1} T_{b_\beta}^{b_\alpha} T_{ee}^{b}\left(q_{b_\beta,j}\right) T_c^{ee} \\
&P^{b_a,c_i} = f\left(x_d^{b_a,c_i} | f_x, f_y, c_x, c_y, k_1, k_2, k_3, p_1, p_2\right)
\end{aligned}
\tag{9}
$$

$x_d^{c_{\alpha,i}}, x_d^{c_{\beta,j}}$는 왜곡이 있는 이미지에서의 체스보드 코너의 이미지 좌표이고, $f(\cdot)$는 이미지 좌표에서 3차원 좌표로 변환해주는 함수이다.

최적화해야 할 파라미터들은 미리 알고 있는 값들로 초기화하였다. 즉, 각 파라미터는 문서상의 값(nominal parameter)으로 초기화하고, 식 (9)를 최소화하는 파라미터들을 찾기 위해 LM (Levenberg-Marquardt) 알고리즘[14]을 사용하여 최적화를 수행하였다.

V. 실험 및 결과

1. 실험 환경

Franka Emika Panda 로봇 팔을 사용하여 실험을 진행하였다. 그림 3과 같이 로봇 팔은 7자유도를 가지며, Realsense D435 카메라를 Gripper의 아래쪽에 부착하였다. D435 카메라는 RGB-D 카메라이지만, 제안된 방법에서는 RGB 이미지만을 사용하고 Depth는 사용하지 않았다.

데이터는 그림 4와 같이 7x5 코너 포인트를 갖는 체스보드를 사용하였으며, 로봇 주변 여섯 위치에 체스보드를 놓아 데이터를 수집하였다. 각 위치에서 랜덤한 로봇 자세로 7장의 이미지와 해당 자세의 조인트 각도를 저장하였다. 각 이미지에서 OpenCV 라이브러리의 findChessboardCorners 함수를 사용하여 체스보드 코너를 추출하였다. $f(\cdot)$는 undistort 함수로 왜곡이 없는 이미지를 만들어 체스보드의 코너를 추출 후, solvePnPRansac 함수를 적용하여 초기 카메라와 체스보드의 변환 행렬 ($T_{0\ldots K}$)를 구하였다. 최적화는 scipy라이브러리의 least_squares 함수를 사용하여 해를 구했으며 유한차분법(finite difference method)을 사용하여 수행하였다.

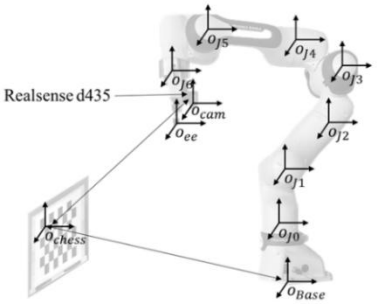

그림 3. 실험 환경 구성.

Fig. 3. The configuration of experimental setting.

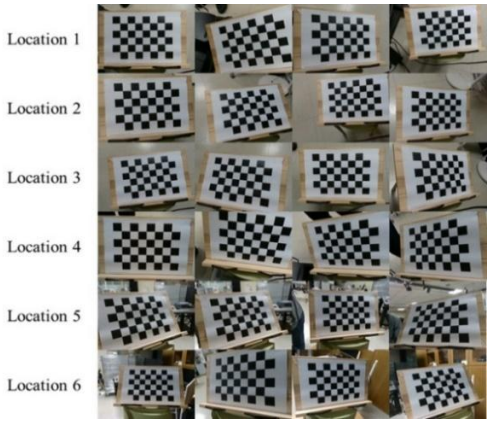

그림 4. 여러 위치에서 저장된 체스보드 이미지.

Fig. 4. The captured chessboard images at multiple locations.

성능 평가를 위해 네 가지의 비교실험을 수행하였다: (1) 핸드-아이 캘리브레이션만 수행한 경우와 핸드-아이 캘리브레이션과 로봇 팔 캘리브레이션을 모두 수행한 경우, (2) RGB 데이터를 사용한 경우와 RGB-D 데이터를 사용한 경우, (3) 레이저 트래커를 이용한 로봇 팔 캘리브레이션과 제안한 방법을 사용한 경우, (4) 파라미터의 수에 따른 성능을 비교, 실험하였다.

첫 번째 실험은 로봇 팔에 내재되어 있는 오차의 유무 파악과 결과에 미치는 영향을 보기 위한 실험이고, 두 번째 실험은 Depth 이미지에 내재되어 있는 거리 오차가 캘리브레이션 결과에 미치는 영향을 파악하고, RGB 데이터만으로도 우수한 캘리브레이션 정확도를 얻을 수 있는지 검증하기 위한 것이다. 이 실험에서는 RGB-D 데이터를 사용한 경우에는 sovlePnPRansac 함수 대신 Depth 값을 이용하여 계산한 코너의 3차원 값을 활용하여 체스보드 기본 모델과 3차원 코너 값 사이의 3D-3D 변환행렬을 구하여 실험하였다. 세 번째 실험은 가장 정확하다고 알려진 레이저 트래커와 결과를 비교하기 위한 것이다. 60개의 랜덤 포즈에서 측정된 마커의 위치 값을 사용하여 로봇 팔 캘리브레이션을 먼저 수행한 후, 수집된 이미지를 활용하여 제안한 알고리즘과 동일한 방법으로 핸드-아이 캘리브레이션을 추가로 수행하였다. 마지막으로 파라미터의 수에 따른 정확도와 최적화 시간을 측정하였다.

표 1. 핸드-아이 캘리브레이션만 한 경우와 전체 캘리브레이션 경우의 성능 비교.

Table 1. Performance comparison between only Hand-Eye calibration and Whole (Hand-eye + Arm) calibration.

	Only Hand-eye calibration	Hand_eye + Arm calibration
2D error (px)	9.15 ± 4.51	0.76 ± 0.46
3D error (mm)	7.39 ± 3.18	0.64 ± 0.38
2D RMSE (px)	10.20	0.89
3D RMSE (mm)	8.05	0.73

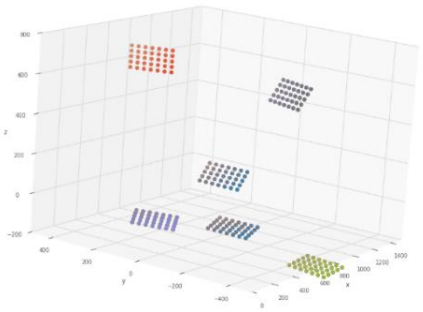

그림 5. 체스보드 코너의 3차원 공간 투영.

Fig. 5. Projection of chessboard corners into 3D space.

위의 모든 실험에서 동일한 이미지 데이터가 사용되었고, 두 번째 실험을 위해서 모든 이미지는 카메라로부터 체스보드가 1m이내에 존재하도록 하였다.

2. 실험 결과

캘리브레이션용 데이터와 같은 방법으로 테스트용 데이터를 추가로 수집하여 성능 테스트를 수행하였다.

표 1에 핸드-아이 캘리브레이션을 수행한 경우와 핸드-아이 및 로봇 팔 캘리브레이션을 수행한 경우의 성능을 비교하였다. 각 로봇 자세에서 측정한 체스보드의 코너를 동일한 위치의 다른 로봇 자세에서 측정된 값을 각각 2D 이미지 공간과 3D 공간 상으로 투영하여 오차를 측정하였다. 표 1에서 보인 바와 같이 핸드-아이 캘리브레이션만 수행했을 때보다 두 가지의 캘리브레이션을 모두 수행했을 때 오차가 1mm 이하로 크게 감소하였다. 그림 5는 3D 공간으로 투영된 체스보드의 코너를 시각화한 것이다. 여섯 위치의 체스보드 코너 포인트를 표시하였다. 각 위치의 포인트는 10개의 데이터가 겹쳐져 있는 것으로 캘리브레이션 결과 하나의 체스보드로 보일 정도로 잘 정렬되었다.

기존의 RGB-D 데이터를 사용한 캘리브레이션 방법들이 존재한다[3,4]. 하지만, Depth 이미지는 거리에 따른 잡음(noise)이 존재하여 Depth 이미지에서 얻어진 거리 값을 사용하였을 때, 캘리브레이션의 정확도에 영향을 줄 수 있다. 사용된 RealSense D435 카메라는 2m 거리에서 최대 2%의 Depth 오차 (약 4cm)를 갖고 거리가 멀어질수록 오차는 점점 커진다[15]. 그 영향을 알아보기 위해 RGB-D 데이터를 사용했을 때와 본 연구에서 제안한 RGB 데이터만을 사용했을 때의 결과를 비교하였다. RGB-D 데이터를 사용할 경우에는 체스보드 코너의 3차원 좌표를 Depth 이미지에서 얻은 거리 값을 사용하여 계산하였으며, 카메라의 내부 파라미터는 RGB 이미지와 Depth 이미지와의 카메라 캘리브레이션 정보를 유지하기 위해 내부 파라미터를 제외하고 최적화 과정을 수행하였다. 실험결과를 표 2에 보였다.

표 2. D435 카메라의 RGB-D와 RGB 영상을 사용한 경우의 성능 비교.

Table 2. Performance comparison between using RGB-D images and using RGB images from D435 camera.

	RGB-D 사용	RGB 사용
2D error (px)	1.10 ± 0.76	0.76 ± 0.46
3D error (mm)	1.37 ± 1.03	0.64 ± 0.38
2D RMSE (px)	1.34	0.89
3D RMSE (mm)	1.71	0.73

표 3. 레이저 트래커와의 성능 비교.

Table 3. Performance comparison with Laser Tracker.

	Laser Tracker + Hand-Eye Calib.	Proposed Method
2D error (px)	1.81 ± 1.02	0.76 ± 0.46
3D error (mm)	1.81 ± 1.05	0.64 ± 0.38
2D RMSE (px)	2.08	0.89
3D RMSE (mm)	2.09	0.73

RGB-D 데이터를 사용하였을 때의 오차가 RGB 데이터만을 사용했을 때보다 2D 이미지 공간에서는 약 0.5 픽셀, 3D 공간에서는 약 1 mm 만큼 더 크게 측정되었다. 이는 카메라의 Depth 정확도가 전체 캘리브레이션의 결과에 부정적으로 작용하지만, RGB 데이터만을 사용하는 경우 Depth 오차의 영향을 받을 가능성을 없앰으로써 오차 1 px 이하, 1 mm 이하의 더 정확한 결과를 보였다. 성능 테스트를 위한 데이터는 RGB-D 카메라가 작은 오차로 Depth 데이터를 안정적으로 획득할 수 있도록 카메라의 원점과 체스보드 간 거리가 1m 이내에서 수집하였다. 더 먼 거리에서 측정된 체스보드 데이터를 이용하였을 경우에는 Depth 오차에 의한 영향이 더 크게 나타날 수 있다.

3. 레이저 트래커와의 결과 비교

제안한 방법의 정확도를 레이저 트래커를 사용한 로봇 팔 캘리브레이션 결과와 비교하였다. 레이저 트래커로는 핸드-아이 캘리브레이션을 수행할 수 없기 때문에, 먼저 로봇 팔을 캘리브레이션 하여 D-H 파라미터를 결정하고, 체스보드를 사용하여 핸드-아이 캘리브레이션을 추가로 수행하였다. 실험을 위해 그리퍼에 마커를 부착하여 Hexagon AT960 Leica Laser Tracker로 측정하였고, RoboDyn 소프트웨어를 이용하여 캘리브레이션 하였다. 실험 결과는 표 3에 보였다.

제안하는 방법이 레이저 트래커를 이용한 캘리브레이션에 비해, 두 가지의 캘리브레이션에서 발생한 누적 오차를 이용하여 최적화를 수행함으로써 전체 오차를 최소화하여 1 mm 이하의 작은 오차를 보였다.

4. 파라미터 수에 따른 성능 및 실행시간

제안하는 방법은 IV.2에서 언급한 것과 같이 $38 + 6 \times K$개의 파라미터를 가지고 있다. 이미지 수(K)에 따라 파라미터의 수가 선형적으로 증가하게 되어 최적화 완료까지의 시간이 달라진다. 파라미터의 수에 따른 성능 분석을

표 4. 파라미터 수에 따른 성능 및 최적화 실행시간.
Table 4. Performance based on the number of parameters and optimization time.

	Small	Medium	Full
2D error (px)	1.30 ± 0.81	1.07 ± 0.79	0.76 ± 0.46
3D error (mm)	1.35 ± 0.73	1.32 ± 0.73	0.64 ± 0.38
2D RMSE (px)	1.54	1.40	0.89
3D RMSE (mm)	1.65	1.51	0.73
Opt. Time	약 3분	약 30분	약 10시간

위해 두 가지의 변형 모델을 추가하여 총 세 가지의 경우 (‘Small’, ‘Medium’, ‘Full’)에 대해 실험하였다.

‘Small’은 Kinematic Parameter (Hand-eye (6), DH-parameters (23))만을 최적화 변수로 설정한 버전으로 29개의 파라미터를 사용하고, ‘Medium’은 ‘Small’에 Camera Parameter (Intrinsic (4), Distortion (5))를 포함하여 38개의 파라미터를 사용한다. 그리고 ‘Full’은 제안하는 방법으로 ‘Medium’에 각 이미지당 6개의 변환행렬을 구하기 위한 파라미터를 포함하여 290 (38 + 6 × (7장 × 6위치))개의 파라미터를 사용한다. 표 4와 같이, 파라미터의 수가 증가할수록 성능은 향상되었으나 최적화에 필요한 시간은 늘어나는 것을 알 수 있다.

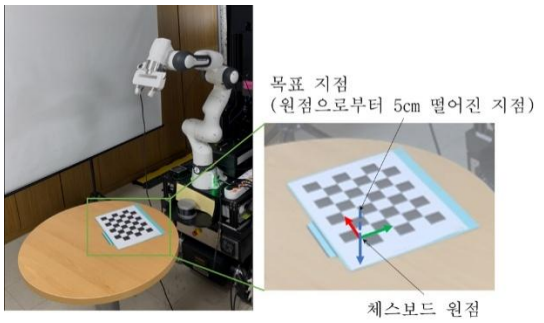

그림 6. 실제 로봇 실험 환경.
Fig. 6. The experimental setting for the real robot.

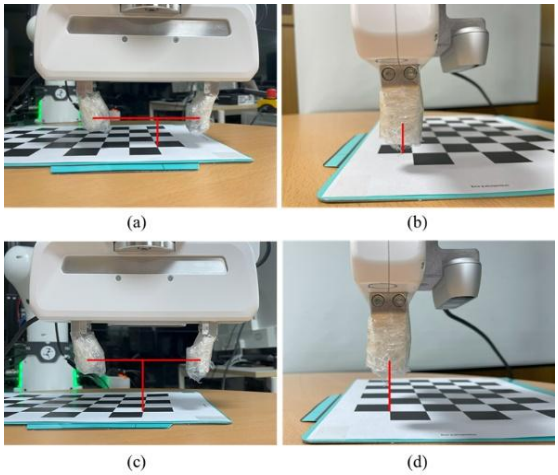

그림 7. 실제 로봇을 이용한 실험 결과.
Fig. 7. The experimental result of using the real robot.

5. 로봇 실험

캘리브레이션 결과를 실제 로봇에 반영하여 목표 위치로 정확히 움직이는가를 알아보기 위해, 그림 6과 같이 테이블 위에 체스보드를 놓고 로봇 팔에 장착된 카메라로 인식 후 체스보드의 원점 위치에서 수직방향으로 5 cm 떨어진 곳으로 로봇 엔드이펙터의 중심이 이동하는 실험을 수행하였다.

실험 결과는 그림 7과 같다. 첫 번째 줄 그림 (a)와 (b)는 핸드-아이 캘리브레이션만 수행한 결과이고, 두 번째 줄 그림 (c)와 (d)는 전체 캘리브레이션을 수행한 결과이다. 수직방향 붉은 색 라인이 그리퍼 중심의 위치를 표시한다. 핸드-아이 캘리브레이션만 수행했을 경우에는 그리퍼의 중심이 체스보드의 원점에서 한 쪽으로 치우친 반면, 전체 캘리브레이션을 수행한 경우는 체스보드의 원점이 그리퍼의 중심에서 수직으로 연장된 붉은 선과 정면과 측면 모두에서 일치하도록 움직인 것을 확인할 수 있다. 제안된 방법을 통해 캘리브레이션을 수행한 결과 카메라로 인식한 위치로 로봇 팔의 엔트이펙터가 정확히 이동하였다.

VI. 결론

본 논문에서는 여러 위치에서 측정된 체스보드를 RGB 카메라로 측정하여 핸드-아이 캘리브레이션과 로봇 팔 캘리브레이션을 동시에 수행하는 방법을 제안하였다. 한 곳에 놓인 체스보드를 이용할 경우에 로봇 팔의 운동 범위가 제한될 수 있다는 문제점을 체스보드의 위치를 움직이면서 측정함으로써 해결하였다. 또한, 서로 다른 체스보드의 데이터를 최적화하는데 사용하기 위해서 체스보드 간의 변환 행렬을 로봇이 움직인 변환 행렬로 대체하여 해결함으로써, 모든 데이터 간의 오차 계산이 가능해지게 하였다. 최적화는 번들 조정을 사용하여 핸드-아이 캘리브레이션, 로봇 팔 캘리브레이션뿐만 아니라, 카메라 내부 파라미터, 체스보드의 변환 행렬도 최적화 파라미터에 포함시켜서, 거리 값을 이용하지 않고 RGB 이미지만을 이용한 캘리브레이션이 가능하게 하였다. 최적화 파라미터를 최소화하여 연산시간을 단축하기 위한 연구가 필요할 것이다.

REFERENCES

[1] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, “A survey of robot manipulation in contact,” *Robotics and Autonomous Systems*, vol 156, 2022.

[2] A. Nubiola and I. A. Bonev, “Absolute calibration of an ABB IRB 1600 robot using a laser tracker,” *Robotics and Computer-Integrated Manufacturing*, vol 29, no 1, pp. 236-245, 2013.

[3] J. Li, A. Ito, H. Yaguchi, and Y. Maeda, “Simultaneous kinematic calibration, localization, and mapping (SKCLAM) for industrial robot manipulators,” *Advanced Robotics*, vol. 33, no. 23, pp. 1225-1234, 2019.

[4] W. B. Jang, J. Y. Lee, S. H. Park, S. Y. Chung, M. Jin, and M. J. Hwang. “Kinematic calibration of robot manipulator using RGB-D camera,” *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 29, no. 3, pp. 264-271, 2023.

[5] R. Y. Tsai and R. K. Lenz, “Overview of a unified calibration trio for robot eye, eye-to-hand, and hand calibration using 3D

machine vision," *Sensor Fusion: Spatial Reasoning and Scene Interpretation*, vol. 1003, pp. 202–213, Jan. 1989.
[6] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 22, pp.1330–1334, 2000.
[7] J. Kannala, and S.S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1335–1340, 2006.
[8] J. Wang, F. Shi, J. Zhang, and Y. Liu, "A new calibration model of camera lens distortion," *Pattern Recognition*, vol. 41, pp. 607–615, 2008.
[9] Z. Zhang, L. Zhang, and G. Z. Yang, "A computationally efficient method for hand-eye calibration," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, pp. 1775–1787, 2017.
[10] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 345–358, 1989b.
[11] R. Judd, and A. Knasinski, "A technique to calibrate industrial robots with experimental verification," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 20–30, 1990.
[12] J. J. Craig, "Introduction to robotics: Mechanics and control (3rd ed.)," *Pearson*, 2018.
[13] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376-380, April 1991.
[14] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp.431–441, 1963.
[15] Intel RealSense Computer Vision – Depth and Tracking cameras, https://www.intelrealsense.com/depth-camera-d435/

이 지 용

2005년 전남대학교 컴퓨터정보통신공학부(공학사). 2007년 과학기술연합대학원대학교 HCI 및 로봇응용공학전공(공학석사). 2023년 University of Manchester Computer Science (공학박사). 2023년~현재 한국과학기술연구원 휴머노이드연구단 박사후연구원. 관심분야는 컴퓨터비전, 기계학습, 로보틱스.

유 범 재

1985년 서울대학교 제어계측공학과(공학사). 1987년 KAIST 전기및전자공학과(공학석사). 1991년 KAIST 전기및전자공학과 (공학박사). 1991년~1994년 ㈜터보테크 연구실장. 1994년~현재 한국과학기술연구원 휴머노이드연구단 책임연구원. 관심분야는 휴머노이드 로봇, 공존현실, 공간 인터랙션, 로봇비전, 임베디드시스템.