

多関節ロボットによるピッキング自動化

中山 恵介* 富田 洋文* 前島 崇宏* 平間 毅** 若生 進一* 木村 匠***

1. はじめに

株式会社昭芝製作所では、製造工程において人手で行っている部品のピッキングを省力化するため、ロボットによる自動化を検討している。そこで、多関節ロボットによるピッキングを自動化するための要素技術について、研究開発を行った。

2. 目的

本研究開発のシステムは、大別すると、ロボット、カメラ、ワークで構成される(図1)。ピッキングを自動化するには、最適化計算などにより、それぞれの位置姿勢関係を明確にする必要がある。

よって、図1のシステムを構築し、ロボット座標系から見たワークの位置姿勢を求め、実機にてキャリブレーションの精度を検証した。

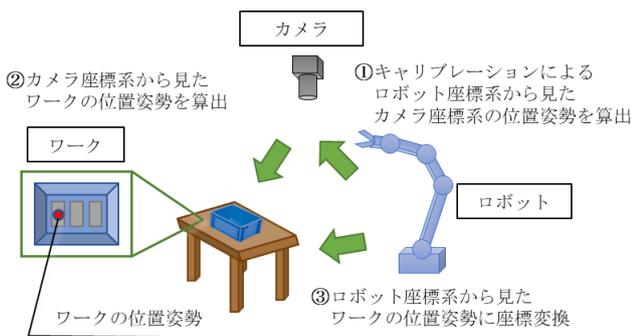


図1 多関節ロボットによるピッキングシステム

3. 研究内容

3.1 キャリブレーションによるロボット座標系から見たカメラ座標系の位置姿勢

ロボットとカメラの位置姿勢関係をキャリブレーションするため、図2に示すシステムを構築した。ロボット、カメラ及びキャリブレーションボードで構成され、それぞれに図2に示す座標系を定義した。また、キャリブレーションボードのパターンには、サークルグリッドを使用した。

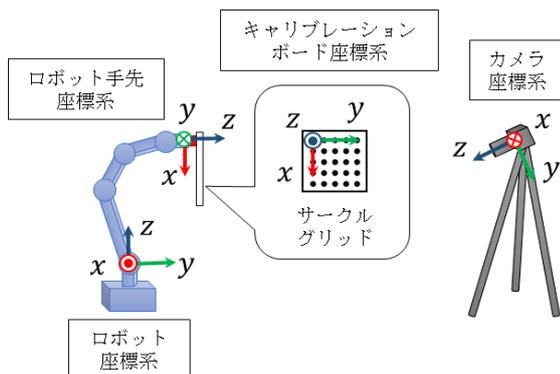


図2 キャリブレーションシステムの構成

キャリブレーションの手順を示す。

- A) カメラにキャリブレーションボードが写るように、ロボットを手動操作し、静止状態でのキャリブレーションボードの画像とロボット座標系から見たロボット手先座標系の位置姿勢関係を多数取得する。
- B) 図2に示す座標系を座標変換し、カメラ座標系から見た円の位置(x_{p1}, y_{p1}, z_{p1})を求める。
- C) 3次元位置情報(x_{p1}, y_{p1}, z_{p1})を透視投影変換し、カメラ座標系から見た画像平面上における2次元位置情報(x_p, y_p)に変換する(図3)。
- D) カメラ内部パラメータを用いて2次元位置情報(x_p, y_p)をウインドウ上の指定領域に変換(ビューポート変換)し、ピクセル値(u_p, v_p)を求める。
- E) OpenCVのライブラリである`cv::findCirclesGrid1)`を用いて円を検出し、ピクセル値(u_{obs}, v_{obs})を求める。式(1)、(2)に示すピクセル値の差分をコスト関数に設定し、Ceres-Solverのライブラリ`ceres::AutoDiffCostFunction2)`を用いて、非線形最小二乗法により最適化計算を行う。

$$u_{cost} = u_p - u_{obs} \quad (1)$$

$$v_{cost} = v_p - v_{obs} \quad (2)$$

この最適化計算により、ロボット手先座標系から見たサークルグリッド座標系の位置姿勢関係と、ロボット座標系から見たカメラ座標系の位置姿勢関係の推定値を求めることができる。

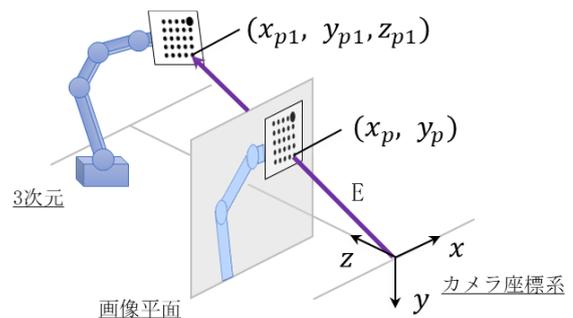


図3 透視投影変換の概要

3.2 カメラ座標系から見たワークの位置姿勢算出

簡易的にキャリブレーションボードをワークの代用品として、キャリブレーション精度を検証する(図4)。カメラ座標系から見たワーク座標系(サークルグリッド原点)の位置姿勢は、撮影したボード画像1枚とOpenCVのライブラリ`cv::solvePnP1)`を使用して求めた。

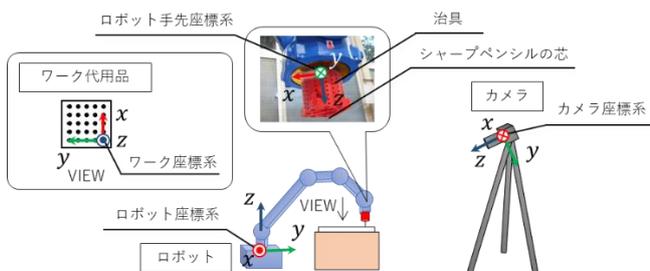


図4 キャリブレーション精度検証の構成

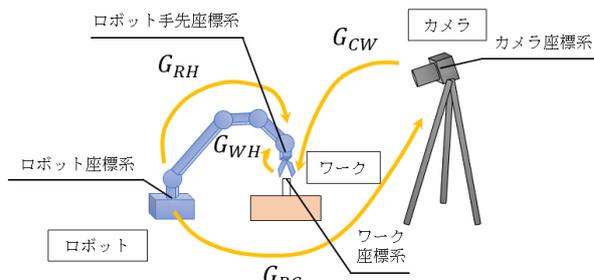


図5 座標系同士の関係

3.3 ロボット座標系から見た手先座標系の算出

ロボット座標系から見たロボット手先の位置姿勢を目標値として、ロボットコントローラに入力することで、ロボット手先を目標値へ移動することができる。

ロボット座標系からロボット手先座標への座標変換行列 G_{RH} は、カメラ座標系からワーク座標系への座標変換行列 G_{CW} 、ロボット座標系からカメラ座標系への座標変換行列 G_{RC} 、ワーク座標系からロボット手先座標系への座標変換行列 G_{WH} を用いて式(3)で表される(図5)。

$$G_{RH} = G_{RC}G_{CW}G_{WH} \quad (3)$$

ここで、座標系 A から座標系 B への座標変換行列 G_{AB} は、座標系 A を z 軸周りに ψ 度回転(新たにできる座標系を A'とする)、座標系 A'を y 軸周りに θ 度回転(新たにできる座標系を A''とする)、座標系 A''を x 軸周りに φ 度回転(新たにできる座標系を A'''とする)した後、座標系 A'''を x 軸方向へ T_x 、y 軸方向へ T_y 、z 軸方向へ T_z 並進する操作を行うことで座標系 B に一致するとき、式(4)で定義される。

$$G_{AB} = T(T_x, T_y, T_z)R_x(\varphi)R_y(\theta)R_z(\psi) \quad (4)$$

$$T(T_x, T_y, T_z) = \begin{pmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\psi) = \begin{pmatrix} \cos \psi & \sin \psi & 0 & 0 \\ -\sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3.4 キャリブレーション精度検証結果

キャリブレーション精度検証は、ティーチングペンダントを用いてシャープペンシル芯の先端をワーク座標系(ボードの原点)に手で移動させた時、ティーチングペンダントに表示される値を目標値とした(図4)。目標値と計算結果を比較した際の誤差を、キャリブレーション精度とする。

「キャリブレーションボードの画像」及び「ロボット座標系から見たロボット手先座標系の位置姿勢」のデータ 168 セットを用いてロボットとカメラの位置姿勢関係をキャリブレーションした。ロボット座標系から見たロボット手先座標系の位置姿勢の最大誤差は、y 方向の並進で 14.1mm、Yaw 方向の回転で 2.74°であった(表1)。

表1 キャリブレーション精度検証の結果

		ロボット座標系から見た ロボット手先座標系の位置姿勢		
		目標値	計算結果	差
位置	x (mm)	140.10	142.00	1.90
	y (mm)	913.34	927.42	14.1
	z (mm)	22.096	23.424	1.33
姿勢	Roll (deg)	179.92	180.93	1.01
	Pitch (deg)	0.0002	0.97974	0.98
	Yaw (deg)	90.004	87.261	2.74

4. まとめ

実機にてキャリブレーションの精度を検証し、ロボット座標系から見たロボット手先座標系の位置姿勢の最大誤差は、y 方向の並進で 14.1mm、Yaw 方向の回転で 2.74°であった。

キャリブレーション精度をさらに向上させるためには、y 方向(カメラ座標系から見て奥行方向)のキャリブレーション用データを増やすこと等が必要である。

5. 参考文献

- 1) http://opencv.jp/opencv-2svn/cpp/calib3d_camera_calibration_and_3d_reconstruction.html
- 2) http://ceres-solver.org/nlns_modeling.html