

制 御 工 学 実 験 報 告 書

実験テーマ：(5) 多自由度マニピレーターの運動学

実験日 : 令和 6 年 11 月 08 日

11 月 22 日

共同実験者 : 3 番 蘆田 修平
 1 1 番 岡本 陵平
 1 9 番 近藤 慧始
 3 6 番 宮武 駿

提出日 : 令和 6 年 11 月 29 日

再提出日 : 令和 年 月 日

再々提出日 : 令和 年 月 日

5 S 2 8 番

氏名 : 野口 史遠

コメント

(5) 多自由度マニピレーターの運動学

1 実験目的

産業用ロボットの多くは、複数のリンクと回転関節から構成されるシリアルリンク型である。本実験では、6 自由度垂直多関節型のマニピレータを用いて、ティーチングと呼ばれる産業用ロボットのプログラムを作成する方法を学ぶ。また、マニピレータの制御で用いられる運動学の基礎知識を習得する。さらに、カメラを用いた物体検知と 3 次元位置計測を通じて、ロボットビジョンの基礎技術を学ぶ。

2 多自由度マニピレータ

2.1 ティーチング

マニピレータ（産業用ロボット）の制御は、運動学や軌道決定を計算し、モータを制御するなどのマニピレータの動きを含めた制御に分けて考える必要がある。前者については、4 年後期の講義「ロボティクス II」にて基礎知識を学ぶ。後者のマニピレータへの動作入力方法には、ティーチングとよばれる表示方法が用いられている。

ティーチングは、ダイレクトティーチング、オンラインティーチング、オフラインティーチングに分類される。ダイレクトティーチングはマニピレータ本体に直接（ダイレクト）に触れながらマニピレータの動作を記録し、記録した動作をプレイバック（再現）する。オンラインティーチングはティーチングペンダントと呼ばれるインターフェイスを使用して、マニピレータを手動操作し、動作を記録、プレイバックさせる。オフラインティーチングは、コンピュータ上でシミュレータを操作し、動作を記録する。そして、実際のマニピレータにデータを送信し、プレイバックさせる方法である。

2.2 システム構成

本実験で用いる多自由度マニピレータは、xArm 6（UFACTORY 製）である。6 個の回転関節を有するため、自由度は 3 次元空間の位置姿勢を任意に決めることができる 6 自由度である。各関節はサーボモータによって駆動され、関節角度はアブソリュート型エンコーダによって取得される。

本実験システムは、マニピレータ、空圧駆動型ロボットグリッパおよび制御 PC、ノート PC から構成される。図 2.1 にマニピレータのシステム構成を示す。制御 PC は Linux と呼ばれる OS が搭載されており、制御周期 10ms でマニピレータを制御している。制御 PC 内では、マニピレータの運動学、姿勢、軌道の計算、モータの制御などが行われており、マニピレータの動きを統合的にコントロールしている。ノート PC は専用の制御ソフトウェアを用いることで、マニピレータの手動操作、状態の確認やティーチング、各種設定を行うことができる。つまり、ティーチングペンダントと同等の操作ができる。また、Python を用いることで、ロボットビジョンや AI を活用したマニピレータへの動作指示の送信や、マニピレータの状態を受信することも可能である。

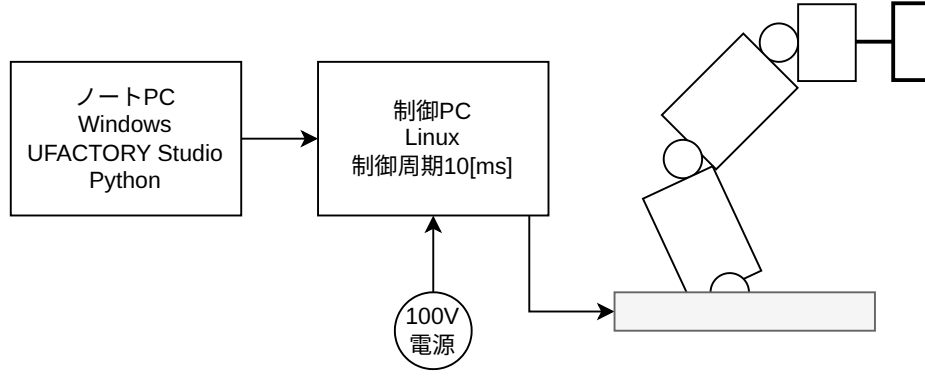


図 2.1 システム構成

2.3 マニピュレータの運動学

各関節角度を入力として、手先の位置姿勢を求める問題を順運動学という。一方で、手先の位置姿勢を入力として各関節角度を求める問題を逆運動学という。本実験では、マニピュレータの第 4～第 6 関節を固定し、3 自由度マニピュレータとして運動学を考える。これは、マニピュレータを簡単な構造として、運動学を幾何学計算するためである。

2.3.1 順運動学

グローバル座標系 Σ_0 の原点から z_0 軸方向に l_1 平行移動した座標系 Σ からみた手先位置を計算する。幾何学的に考える場合、まず $x'z'$ 平面で考える。三角関数を用いると、手先位置 $[x' \ z']^T$ は、

$$x' = l_2 \cos(\varphi_1 - \theta_2) - l_3 \cos\{(\varphi_1 - \theta_2) + (\varphi_2 - \theta_3)\} \quad (2.1)$$

$$z = l_2 \sin(\varphi_1 - \theta_2) - l_3 \sin\{(\varphi_1 - \theta_2) + (\varphi_2 - \theta_3)\} \quad (2.1)$$

となる。次に、座標系 Σ の原点から手先位置を xy 平面に射影すると、式 (2.1) より大きさ $|x'|$ を用いて、

$$x = |x'| \cos \theta_1 \quad (2.2)$$

$$y = |x'| \sin \theta_1 \quad (2.2)$$

と計算することができる。また、グローバル座標系 Σ_0 からみた手先位置は、式 (2.1)、式 (2.2) を用いて、

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z + l_1 \end{bmatrix} \quad (2.3)$$

となる。

2.3.2 逆運動学

逆運動学について説明する。グローバル座標系からみた手先位置 $\begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$ が既知の場合、

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 - l_1 \end{bmatrix} \quad (2.4)$$

と座標系 Σ からみた手先位置を求めることができる。つまり、手先位置 $[x \ y \ z]^T$ が既知となった。次に、座標系 Σ の xy 平面を考えると、関節角度 θ_1 は、

$$\theta_1 = \tan^{-1} \frac{y}{x} \quad (2.5)$$

と x, y を用いて計算できる。なお、 $x^2 + y^2 = 0$ のときは、 θ_1 は一意に決まらず、任意の値になる。ここで、アークタンジェントは、プログラミング言語において atan と atan2 の 2 種類存在する。今回は、戻り値の範囲が広い atan2 を使用する。

$$\theta_1 = \text{atan2}(y, x) \quad (2.6)$$

以下、アークタンジェントは atan2 で表記する。

次に関節角度 θ_3 を余弦定理から求める。先に計算に必要な x' について求める。 xy 平面の原点から手先位置を射影した点までの長さは

$$|x'| = \sqrt{x^2 + y^2} \quad (2.7)$$

と x, y を用いて計算できる。絶対値を外すと、

$$x' = \pm \sqrt{x^2 + y^2} \quad (2.8)$$

である。この値を用いて余弦定理を適用する。余弦定理は、三角形 ABC に対し、辺 AB, AC のなす角を θ とすると、

$$BC^2 = AB^2 + AC^2 - 2AB \cdot AC \cos \theta \quad (2.9)$$

という関係式が成立する定理である。 $x'z'$ 平面の三角形に注目すると、余弦定理を適用して

$$\cos(\varphi_2 - \theta_3) = \frac{l_2^2 + l_3^2 - x'^2 - z^2}{2l_2l_3} \quad (2.10)$$

が成立する。

また、 $\sin^2 \theta + \cos^2 \theta = 1$ の関係から、

$$\sin(\varphi_2 - \theta_3) = \pm \sqrt{1 - \cos^2(\varphi_2 - \theta_3)} \quad (2.11)$$

となるため、式 (2.6) と同様に atan2 を用いると、

$$\theta_3 = \varphi_2 - \text{atan2}\{\sin(\varphi_2 - \theta_3), \cos(\varphi_2 - \theta_3)\} \quad (2.12)$$

となる。

最後に、関節角度 θ_2 を求める。上記で求めた関節角度 θ_3 の値が既知になったと考え、順運動学で求めた式 (2.1) を $\cos(\varphi_2 - \theta_2), \sin(\varphi_2 - \theta_2)$ の連立方程式として考えることができる。連立方程式を解くと、

$$\cos(\varphi_1 - \theta_2) = \frac{K_c x' + K_s z}{K_c^2 + K_s^2}, \quad \sin(\varphi_1 - \theta_2) = \frac{K_s x' + K_c z}{K_c^2 + K_s^2} \quad (2.13)$$

と解を導出できる。atan2 を用いると、

$$\theta_2 = \varphi_1 - \text{atan2}\{\sin(\varphi_1 - \theta_2), \cos(\varphi_1 - \theta_2)\} \quad (2.14)$$

となる。なお、本実験では逆運動学の式の中に正負が混在している場合はすべて正の場合で考えるものとする。

2.4 マニピュレータの姿勢表現

マニピュレータの手先姿勢は2つの座標系 Σ_0 と Σ_e を用いて表現することができる。グローバル座標系 Σ_0 はマニピュレータのベースに固定された静止座標系であり、手先座標系 Σ_e はロボットアームの手先に固定された座標系である。

手先の姿勢を表現する際は、座標系 Σ_0 からみた座標系 Σ_e の各軸 (e_x, e_y, e_z) の単位ベクトル ${}^0\mathbf{e}_x, {}^0\mathbf{e}_y, {}^0\mathbf{e}_z$ を用いる。単位ベクトルとは、大きさ(長さ)が1のベクトルである。つまり、 ${}^0\mathbf{e}_x, {}^0\mathbf{e}_y, {}^0\mathbf{e}_z$ はそれぞれ座標系 Σ_e の軸 (e_x, e_y, e_z) 方向を向く大きさ1のベクトルを意味している。この3つのベクトルをまとめて、 3×3 の行列で表記されたものを回転行列と呼ぶ。

$${}^0\mathbf{x}_P = \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix}, \quad {}^0\mathbf{e}_y = \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix}, \quad {}^0\mathbf{e}_z = \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \quad (2.3)$$

$${}^0\mathbf{R} = \begin{bmatrix} {}^0\mathbf{e}_x & {}^0\mathbf{e}_y & {}^0\mathbf{e}_z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.16)$$

回転行列よりもわかりやすく姿勢を表現する方法として、固定角やオイラー角と呼ばれる方法がある。本実験では xArm 6 に採用されている固定角を用いる。固定角は、座標系 Σ_0 を基準に次の3つの連続する回転で表現する。(1)の軸まわりの「roll」回転、(2)の軸まわりの「pitch」回転、(3)の軸まわりの「yaw」回転である。3つの連続回転により、任意の姿勢を表現することができる。回転行列では3つの角のパラメータが必要であったが、固定角では ψ, θ, ϕ の3つのパラメータで姿勢を表現することができる。ここで、xArm 6 ではなくロボットアーム上の ψ, θ, ϕ をピッチ角 P, ϕ をヨー角 Y, R と呼称している。

また、固定角の ψ, θ, ϕ が既知のとき、回転行列 \mathbf{R} は、

$${}^0\mathbf{R} = R_z R_y R_x = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.17)$$

となり、 ψ, θ, ϕ を軸まわりにそれぞれ単位回転した際の回転行列 \mathbf{R} で計算することができる。また、回転行列の各要素が既知のときは、

$$\theta = \text{atan2} \left(-r_{31}, \pm \sqrt{r_{11}^2 + r_{21}^2} \right) \quad (2.18)$$

$$\psi = \begin{cases} \text{atan2}(r_{21}, r_{11}) & \text{if } \cos \theta > 0, \\ \text{atan2}(-r_{21}, -r_{11}) & \text{if } \cos \theta < 0 \end{cases} \quad (2.19)$$

$$\phi = \begin{cases} \text{atan2}(r_{32}, r_{33}) & \text{if } \cos \theta > 0, \\ \text{atan2}(-r_{32}, -r_{33}) & \text{if } \cos \theta < 0 \end{cases} \quad (2.20)$$

と計算できる. この計算は, 式 (2.16) と式 (2.17) の各要素を比較し, 方程式を立式することで求めることができる. 式 (2.16) の値を反映して, それぞれの求め方は以下の通りである.

- ピッチ角 θ : (1.1) 要素を乗じて和を取る. また, (3.1) 要素を用いて, atan2 を用いる.
- ヨー角 ψ : (1.1) 要素および (2.1) 要素を比較して, atan2 を用いる.
- ロール角 ϕ : (3.2) 要素および (3.3) 要素を比較して, atan2 を用いる.

3 ロボットビジョン

3.1 概要

マニピュレータは, カメラによる画像処理を組み合わせることで, 用途の幅を広げることができる. ステレオカメラ (RGB-D カメラ) を取り付けすることで, 対象物体の 3 次元位置情報を得ることができる. そして, その位置情報を利用することで, 把持対象の検出や手先位置の微調整が可能となる. この一連の作業をリアルタイムで実行できるとすれば, カメラが介在したフィードバックシステム (ビジュアルフィードバックシステムと呼ばれる) が完成する.

本実験では画像処理による対象物体の 3 次元位置計測を行う. そして, 位置情報からマニピュレータの手先位置を決定し, オンラインティーチングによる動作の入力を行う.

3.2 画像処理

本実験では, 次の手順で画像処理を行い, 3 次元位置を計測する.

- (1) 画像の取得: カメラから取得される画像は, カラー画像であり, 赤色成分 (R), 緑色成分 (G), 青色成分 (B) から構成されている. それぞれの成分は, 0 から 255 の 256 階調の値を持つため, 1 画素 (pixel) は 3 バイトである. 色は RGB 色空間と HSV 色空間と呼ばれる 2 つの色の表現方法がある. 本実験では HSV 色空間で画像を処理する.
- (2) 空間フィルタリング: 画像には外乱光などの影響により色の明るさやコントラストなどが変化する. その変化を低減するために, 画像フィルタを通す. 本実験では, 空間フィルタリングと呼ばれる「平均化フィルタ」, 「ガウシアンフィルタ」, 「メディアンフィルタ」, 「双方向フィルタ」をそれぞれ用いる.
- (3) 2 値化処理: HSV 色空間から特定の色を検出し, 白黒画像を生成する. これは 2 値化と呼ばれる処理を行う. 2 値化は任意の値 (しきい値) を定め, それを基準として, 各ピクセルを白または黒に割り当てる. この操作によって, 抽出したい物体をそれ以外と区別する. 実際, しきい値の決定方法には, P タイル法, 判別分析法, 傾分といった手法など多岐にわたるが, ここではヒストグラムを参考に試行錯誤して決定するものとする. この 2 値化処理によって対象物体が特定できる画像が得られる.

- (4) 特徴パラメータの抽出：対象物体の特徴となる値は、面積、周囲長、重心位置、形状などが考えられる。本実験では、画像全体を 1 つの物体と仮定し、2 値化された白部分の面積と重心位置を求める。この重心位置を 3 次元位置計測に利用する。
- (5) ステレオ法による 3 次元位置の計測：2 台のカメラ画像それぞれにある (4) で取得した画像上の物体の重心位置から三角測量の原理を用いると、3 次元位置を計測できる。また、本実験で使用するステレオカメラ (Realsense D435f) にはプロジェクタによるパターンが照射されており、そのパターンを 2 台のカメラが検出することでカメラ画像同士のマッチングを行っている。

3.3 座標変換

カメラから取得された画像はマニピュレータのグローバル座標系とは異なる視点を持っている。そこで、カメラに固定された座標系からみた対象物体の位置を、グローバル座標系に変換して考える必要がある。本実験では、マニピュレータの手先にカメラが固定されているため、カメラに固定された座標系は手先座標系 Σ_e と等しいものとして考える。

座標を変換するには、グローバル座標系 Σ_0 の原点から手先座標系 Σ_e の原点までの位置ベクトル ${}^0\mathbf{p}_{0,e}$ と、手先の姿勢を表す回転行列 ${}^0\mathbf{R}$ を用いて、

$${}^0\mathbf{x}_P = {}^0\mathbf{p}_{0,e} + {}^0\mathbf{R}\mathbf{x}_P \quad (3.1)$$

と計算する。ここで、 ${}^0\mathbf{x}_P$ はグローバル座標系からみた対象物体、 \mathbf{x}_P は手先座標系からみた対象物体の 3 次元位置座標を示している。本実験では、 ${}^0\mathbf{p}_{0,e}$ がマニピュレータの手先位置、 \mathbf{x}_P が求めたい対象物体の 3 次元位置、 ${}^0\mathbf{x}_P$ がカメラから取得した対象物体の 3 次元位置、 ${}^0\mathbf{R}$ がマニピュレータの手先姿勢となる。ただし、マニピュレータの制御用ソフトウェアでは、XYZ オイラー角で姿勢が表現されるため、 ${}^0\mathbf{R}$ に変換する必要がある。

4 実験 1-1：ティーチング実験

4.1 実験概要

本実験では、図 4.1 中に示された番号の位置で試験管を模したホワイトボードマーカを抜き差しするマニピュレータの動作をダイレクトティーチングおよびオンラインティーチングを用いて制御 PC に入力する。P1～P3 の位置には部品が配置されており、その部品をそれぞれ S1～S3 に運搬する。また、運搬の途中で試験管を 3 回振る動作も行う。

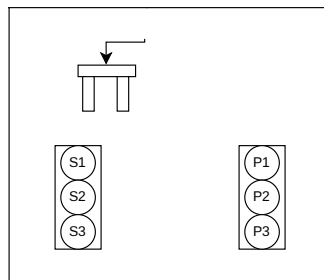


図 4.1 実験環境 (実験 1)

4.2 ダイレクトティーチング実験

4.2.1 実験手順

実験は次の手順で行う。

- (1) 空圧駆動型ロボットグリッパの電磁弁を操作し、ホワイトボードマーカーを把持させる。
- (2) マニピュレータを初期位置姿勢付近に移動させる。
- (3) マニピュレータ制御用ソフトウェアからダイレクトティーチングを実行する。
- (4) 初期姿勢から S1 上部に移動する。
- (5) S1 と P1 の中間地点で、ホワイトボードマーカーを 3 回振る。
- (6) P1 の挿入口にホワイトボードマーカーを指す。
- (7) マニピュレータを初期位置姿勢付近に戻し、ダイレクトティーチングを終了する。
- (8) 記録した動作をプレイバックする。プレイバックの様子は動画撮影する。
- (9) P2 と S2, P3 と S3 の組み合わせでもダイレクトティーチングを実行する。ただし、実行する人を交代させる（全員 1 度は実行すること（重複してもよい））。実行中の様子を適宜写真、動画撮影し、ティーチングに苦労している点および改善すべき点などを観察する。撮影した写真は考察などに利用する。利用する場合は図として載せること。

4.2.2 実験結果

ダイレクトティーチングの実行結果について説明する。図 4.3 にダイレクトティーチングのプレイバックの様子を示す。

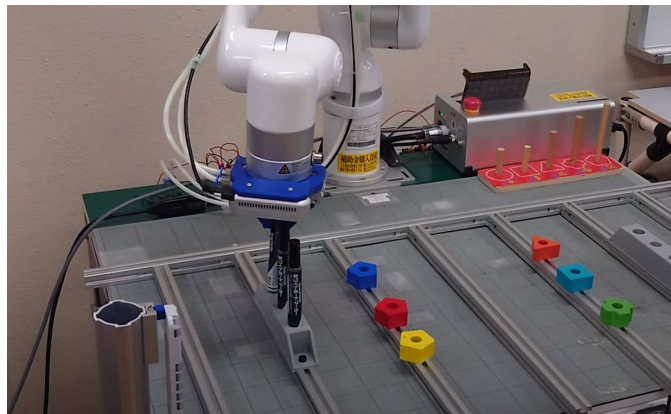


図 4.2 ダイレクトティーチング

ダイレクトティーチングでは、誤差が大きく、穴の大きさが小さくなるに連れ入りずなくなった。

また、マーカーの挿入に失敗した際、図 4.2 のようにマーカーが穴に対して垂直ではなく、斜めになっている。

4.2.3 考察

次に、実行結果について考察する。

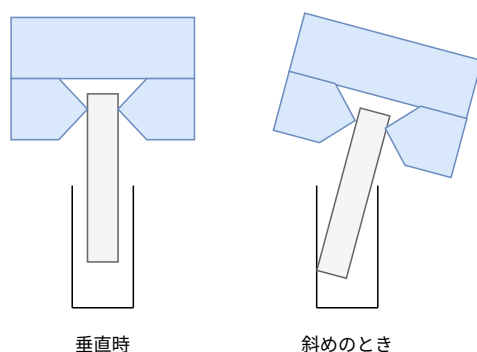


図 4.3 成功例と失敗例

本実験では、マニピュレータの第 4～第 6 関節を固定し、3 自由度マニピュレータとして運動学を考える。実験結果から、マーカーが挿入口の穴に対して垂直でない場合、特に斜めの姿勢で挿入を試みた際に失敗することが多いことが確認された（図 4.2 参照）。これは、図 4.3 に示すように、斜めの場合には穴の奥でマーカーが壁に衝突し、進入が妨げられることが原因である。また、マニピュレータの制御特性上、動作時の最大トルクが限定されているため、壁に衝突した際に動作が停止しやすいことも失敗の一因と考えられる。

加えて、今回のティーチングでは、マニピュレータに直接触れて動作を記録する方式であり、人の操作による微小な誤差が記録された動作にも反映される。特に、マーカーの直径と挿入口の穴の大きさの関係が非常にタイトである場合、この誤差が失敗の主因となる可能性が高い。

3 自由度の運動学に基づいて考えると、マニピュレータの先端の位置と姿勢を同時に制御することが困難である。特に、穴の位置や角度に対して、マーカーを正確に挿入するには先端の角度制御が求められるが、今回は第 4～第 6 関節を固定しているため、これが運動学的な制約となり、マーカーを正確に挿入できない要因の一つとなった。

4.3 オンラインティーチング実験

4.3.1 実験手順

マニピュレータ制御用ソフトウェアを用いて、オンラインティーチングを行う。オンラインティーチングはビジュアルプログラミングモードを用いて行う。実験は次の手順で行う。

- (1) マニピュレータを初期位置姿勢に移動させる。
- (2) マニピュレータ制御用ソフトウェアからオンラインティーチングを実行する。
- (3) 初期姿勢から S1 に移動し、ホワイトボードマーカーを把持する動作を行う。その後、ホワイトボードマーカーを S1 から抜く。
- (4) S1 と P1 の中間地点で、ホワイトボードマーカーを 3 回振る。なお、マーカーを振る動作のティーチングは最初の 1 名が行い、その後はティーチングのデータを複製して使用してよい。
- (5) P1 の挿入口にホワイトボードマーカーを指す。
- (6) マニピュレータを初期位置姿勢に戻す。

- (7) P2 と S2,P3 と S3 の組み合わせでも同様にティーチングを行う。ただし、実行する人を交代すること。そして、全員 1 度は実行すること。人数が 4 人以上いる場合は、挿入口を P1,P2 のホワイトボードマーカを S1,S2 にそれぞれ異なる動作にする。実行中の様子を適宜撮影し、ティーチングに苦労している点などを観察する。撮影した写真は考察などに利用する。利用する場合は図として載せること。
- (8) オンラインティーチングを終了する。
- (9) 記録した動作をプレイバックする。プレイバックの様子は動画撮影する。

4.3.2 実験結果および考察

オンラインティーチングの実行結果について説明する。図??にオンラインティーチングのプレイバックの様子を示す。

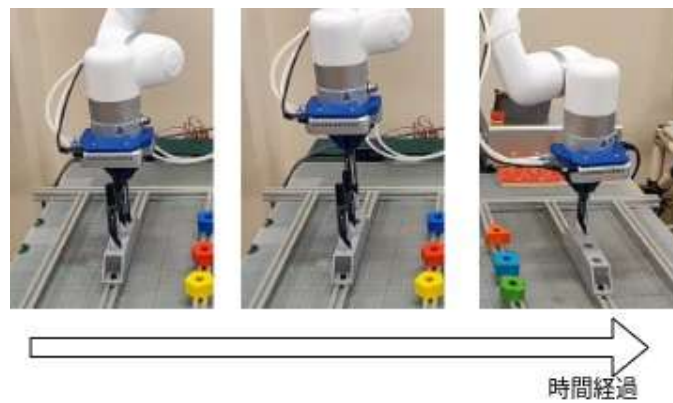


図 4.4 オンラインティーチング

実験結果として、オンラインティーチングにより記録された動作が正確に再現されていることが確認された。特に、以下の点で実験は成功したと評価できる。

マーカ保持：ホワイトボードマーカをグリッパで正確に把持できており、マーカの真ん中を保持する動作が安定して行われた。

マーカの抜き取り：マーカを垂直に抜き取る動作が実現できた。(図 4.4 中央)

経路の正確性：マーカを運ぶ際の手先の経路が直線的であり、予め設定した座標通りの動作が再現された。

マーカの挿入：マーカを挿入口に挿入する動作もスムーズに行われ、挿入角度が垂直であることが確認された。(図 4.4 右)

4.3.3 考察

次に、実行結果を考察する。

まず、オンラインティーチングは、ダイレクトティーチングに比べて精度の高い動作記録が可能である点が確認された。オンラインティーチングでは、手先の位置と経路がプログラムを通じて割り出されるため、動作記録に人為的な誤差が入りにくい。また、記録したデータをプレイバックすることで、同じ動作を高い再現性で繰り返すことができるため、複雑な動作の設定にも適している。手先の位置や運動学を基に動作を記録・再現するため、リンク構造や関節角度の特性を正確に反映した動作が実現できた。

ロボットのグリッパについては、エア駆動型の特徴が動作の安定性に寄与している。エア圧を用いた把持は、マーカに適切な力を加えることで、破損のリスクを回避しつつ正確に把持できることが実証された。また、高速な開閉動作が可能で、マーカを保持する動作や離脱動作がスムーズに行われた。この特徴により、マーカの真ん中を正確に保持し、垂直に抜き取る動作が安定して再現された。

さらに、実験環境での測定が重要であると考え、特に、目標座標（S1,P1 など）を事前に正確に測定し、プログラムに反映したことで、記録と再現の間に生じる誤差を最小限に抑えることができた。また、ロボットアームの初期位置を統一し、マーカや挿入口の固定を行ったことで、環境全体の安定性を確保し、繰り返し動作の精度を高めることができた。

5 実験 1-2：運動学（3 自由度）実験

5.1 実験概要

本実験では、制御 PC 内で実行されている運動学をエクセルを用いて計算する。そして、指定された手先位置にマニピュレータを移動させる実験を行う。マニピュレータは第 4～第 6 関節が 0[deg] に固定された 3 自由度マニピュレータとして扱う。マニピュレータへの動作入力は関節角度の数値指定によるオンラインティーチングを用いる。

また、図 5.1 に示すように本実験における逆運動学は目標位置においてのみ計算し、実験 1-2 とは異なり途中経路では計算を行わないものとする。

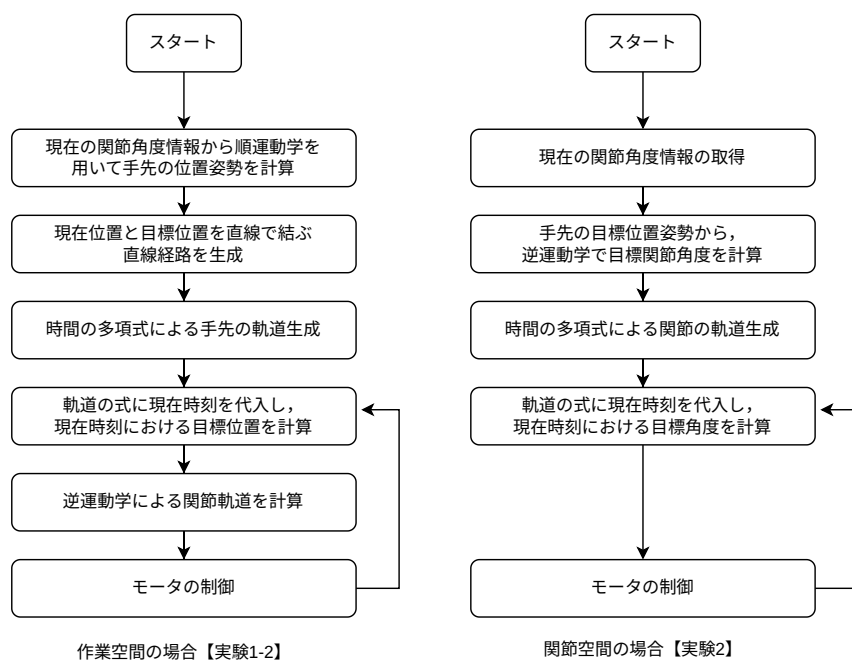


図 5.1 フローチャート

5.2 実験手順

実験は以下の手順で行う。

- (1) 逆運動学（式 (2.6), 式 (2.12)～式 (2.14)）を計算するエクセルファイルを作成し, 表 5.2 に記された手先位置から関節角度を計算する. 数値は小数点第 1 位まで表記するものとする.
- (2) 初期位置, No.1～3, 初期位置の順に手先位置が到達するようにオンラインティーチングを実行する.
- (3) 記録した動作をプレイバックする. プレイバックの様子は撮影する.

表 5.1 手先位置

Position No.	x [mm]	y [mm]	z [mm]
1	168.1	168.1	368.8
2	210.6	210.6	196.1
3	541.3	0	182.8

5.3 実験結果

表 5.3 に逆運動学の計算結果を示す. また, 図 5.2 に, 手先位置が移動した際の経路の模式図を示す.

図 5.2 に示すように, 手先位置が移動する際の経路は, 直線ではなく弧を描いている. この経路は, 手先の直線的な移動を行う実験 1 と異なり, 本実験では関節空間での動作が主な要因となっている.

具体的には, 実験 1 では直線経路が生成されており, 手先の位置を目標座標系に沿って直接制御していた. 一方で, 本実験では関節角度を指定する形式で制御を行ったため, 手先の動きは関節空間での動作結果として自然に弧を描く形となった.

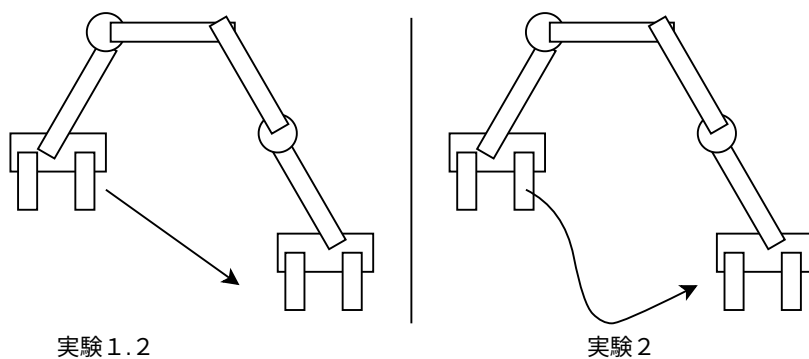


図 5.2 手先の経路

表 5.2 各関節角度の最適計算結果

Position No.	θ_1 [deg]	θ_2 [deg]	θ_3 [deg]
1	45.0	-60.0	0.0
2	-45.0	-10.0	-10.0
3	0.0	30.0	-60.0

表 5.3 計算結果

ポジション 1 の解	θ_1 [deg]	θ_2 [deg]	θ_3 [deg]
解 1	45.0	-60.0	0.0
解 2	45.0	-15.2	59.8
解 3	225.0	38.7	59.8
解 4	225.0	166.4	0.0

5.4 考察

次に, 実行結果の考察を述べる.

本実験における手先経路が実験 1 と異なる理由は, 運動学の考え方と動作制御方法の違いに起因する. 実験 1 では手先位置を直線的に移動させるために, 手先の軌道を直接指定する「デカルト空間」での制御を用いていた. 一方で, 本実験では「関節空間」における動作が採用され, 指定された関節角度に基づいて動作が行われる. このため, 関節の回転運動に依存する形で手先経路が弧を描く形となった.

また, マニピュレータの可動範囲を考慮した逆運動学の解についても注意が必要である. 本実験では, 関節角度を指定する際に複数の解が存在することが表 5.4 の計算結果から確認できる. 例えば, ポジション 1 の解では, 複数の角度組み合わせが得られるが, それぞれが異なる経路を生成する可能性がある. このうち, マニピュレータの物理的な可動範囲や不要な動きを回避するために適切な解を選択することが重要である.

さらに, 特定の解を選択しない場合, マニピュレータが関節の限界位置や奇妙な経路を辿る可能性があり, 作業精度や安全性に影響を及ぼす. 例えば, 表 5.4 に示されるように, 解 4 では関節角度が大きく変化するため, 動作が不自然になる可能性がある. このような問題を回避するため, 逆運動学の解を選択する際には, マニピュレータの可動範囲や作業条件を十分に考慮する必要がある.

6 実験 2-1：画像処理実験

6.1 実験概要

本実験では, 手先に装着した RGB-D カメラから取得した画像を処理し, 対象物体の検出および 3 次元位置情報の取得を実験する. マニピュレータの卓上には, 右角度の積み木 (赤色, 青色, 黄色) および収納台が設置されており, 実験では積み木および各収納位置を対象に実験を行う.

6.2 実験手順

実験手順は以下の手順で行う.

- (1) 卓上に積み木を 1 個設置し, 卓上の目盛りから各物体の $0^\circ xy$ 座標を直接計測し, 記録する.
- (2) Python の開発環境 Spyder を起動し, 画像処理プログラムから空間フィルタリングを OFF にする.
- (3) 画像処理プログラムを実行後, マニピュレータの制御用ソフトウェアを用いてマニピュレータを手動操作し, 画像に積み木が現れる位置まで手先を移動する. この際, 手先の位置姿勢を記録する.
- (4) ヒストグラムを参考にしながら積み木の HSV 色空間のしきい値 (最大値, 最小値) を設定し, 記録する. また, スクリーンショット等を用いて, 画像処理結果を記録する.
- (5) 得られた 3 次元位置情報を記録する.
- (6) 「平均化フィルタ」, 「ガウシアンフィルタ」, 「メディアンフィルタ」, 「双方向フィルタ」において, それぞれ (1) ~ (5) を繰り返す.
- (6) 残りの積み木および収納位置を適所について, (1) ~ (6) を繰り返す.
- (7) 式 (3.1) を用いて, 座標変換を行い, 物体の 3 次元位置をグローバル座標系で表現するエクセルファイルを作成し, 計算する.

6.3 実験結果

表 6.1 ~ 表 6.3 に実験結果を示す.

表 6.1 積み木 (赤) の結果

	フィルタ無	平均フィルタ	ガウシアン F	メディアン F	双方向フィルタ
実位置 (x_0°, y_0°) [mm]	488.5, -130.0				
手先位置 (x_e°, y_e°, z) [mm]	401.1, 0.0, 418.2				
手先姿勢 (R, P, Y) [deg]	180, 0, 0				
H(max, min)	(3,0)	(10,0)	(10,0)	(10,0)	(10,0)
S(max, min)	(255,60)	(232,61)	(255,60)	(255,60)	(255,60)
V(max, min)	(255,0)	(255,0)	(255,0)	(255,0)	(255,95)
計測位置 (x_e°, y_e°, z) [mm]	(85.1, 120.1, 371)	(84.5, 130.3, 379.9)	(85.0, 128.6, 377.9)	(84.3, 124.3, 375.9)	(85.0, 126.9, 375.9)
計測位置 (x_0°, y_0°, z) [mm]	(486.2, -120.1, 47.2)	(485.6, -130.3, 38.3)	(486.1, -128.6, 40.3)	(485.4, -204.3, 42.3)	(486.1, -126.9, 42.3)

表 6.2 積み木（青）の結果

	フィルタ無	平均フィルタ	ガウシアン F	メディアン F	双方向フィルタ
実位置 (x_0^o, y_0^o) [mm]	588.5, -130.0				
手先位置 (x_e^o, y_e^o, z) [mm]	401.1, 0.0, 418.2				
手先姿勢 (R, P, Y) [deg]	180, 0, 0				
H(max, min)	(160,140)	(160,140)	(160,140)	(160,140)	(160,140)
S(max, min)	(255,60)	(255,60)	(255,60)	(255,60)	(255,60)
V(max, min)	(255,0)	(255,0)	(255,0)	(255,0)	(255,0)
計測位置 (x_e^o, y_e^o, z) [mm]	(181.6,128.1,374.9)	(182.7,131.5,377.9)	(182.1,129.7,375.9)	(182.4,129.0,375.9)	(180.7,127.4,375.9)
計測位置 (x_0^o, y_0^o, z) [mm]	(582.7,-128.1,43.3)	(583.1,-131.5,40.3)	(583.2,-129.7,42.3)	(583.5,-129.0,42.3)	(581.8,-127.4,42.3)

表 6.3 積み木（黄）の結果

	フィルタ無	平均フィルタ	ガウシアン F	メディアン F	双方向フィルタ
実位置 (x_0^o, y_0^o) [mm]	388.5, -130.0				
手先位置 (x_e^o, y_e^o, z) [mm]	401.1, 0.0, 418.2				
手先姿勢 (R, P, Y) [deg]	180, 0, 0				
H(max, min)	(50,30)	(50,30)	(50,30)	(50,30)	(50,30)
S(max, min)	(255,60)	(255,60)	(255,60)	(255,60)	(255,60)
V(max, min)	(255,0)	(255,0)	(255,0)	(255,0)	(255,0)
計測位置 (x_e^o, y_e^o, z) [mm]	(-12.3,119.9,365.9)	(-15.0,124.7,370.9)	(-13.0,121.9,366.9)	(-13.0,117.2,365.9)	(-13.0,121.2,365.9)
計測位置 (x_0^o, y_0^o, z) [mm]	(338.8,-119.9,52.3)	(386.1,-124.7,47.3)	(388.1,-121.9,51.3)	(388.1,-117.2,52.3)	(-13.0,121.2,365.9)

次に、図 6. 1 に各フィルタの誤差率のグラフを示す。

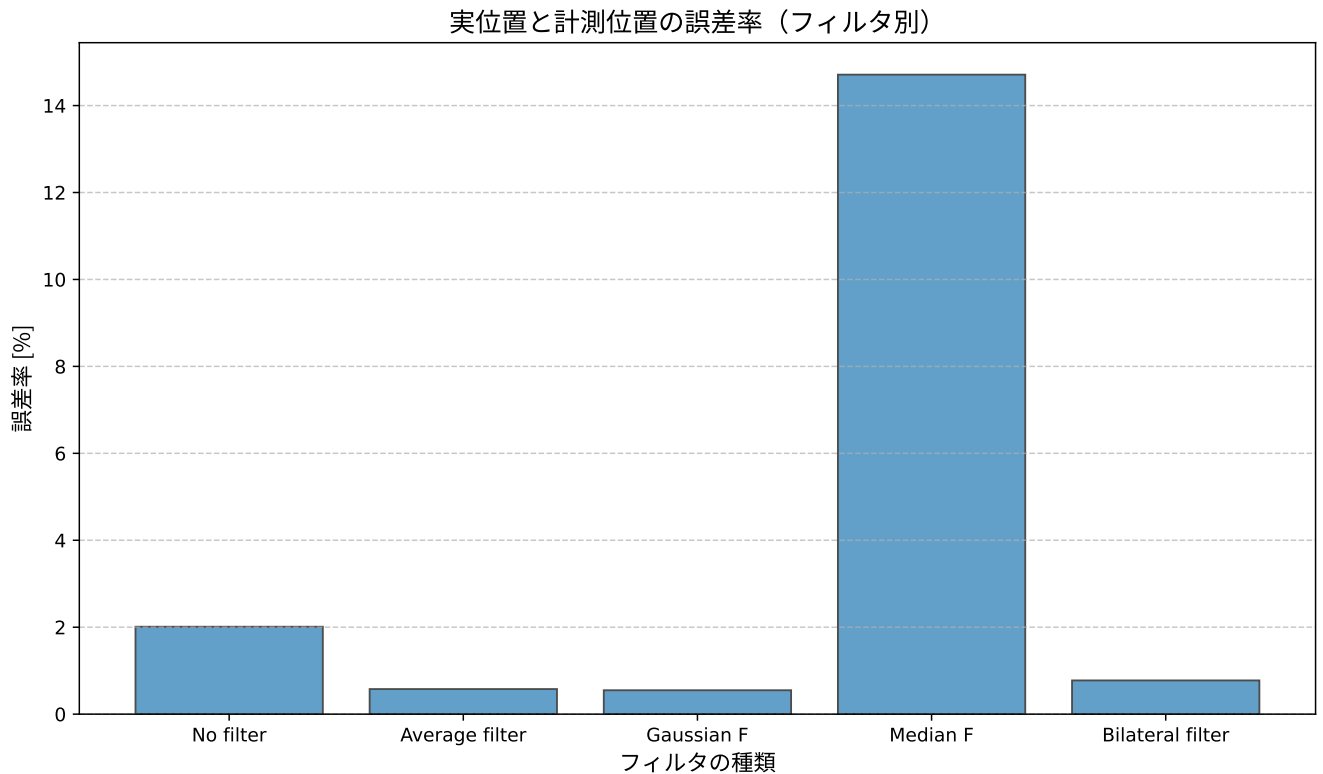


図 6.1 各フィルタの誤差率のグラフ

次に、図 6.5 から図 6.6 に赤ブロックのフィルタ後の画像を示す。

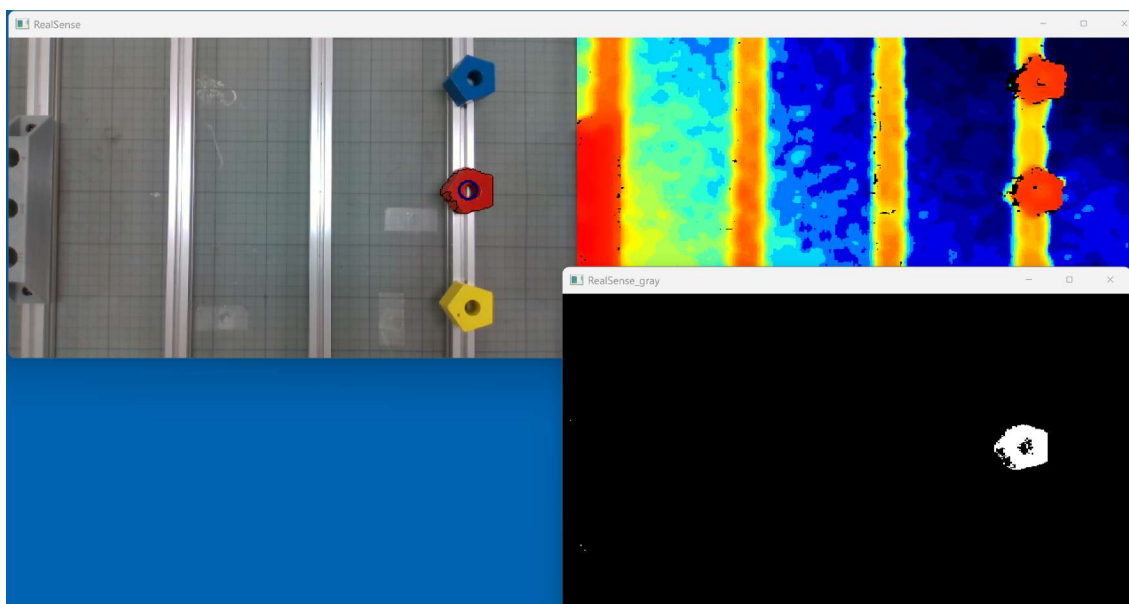


図 6.2 赤フィルタなし

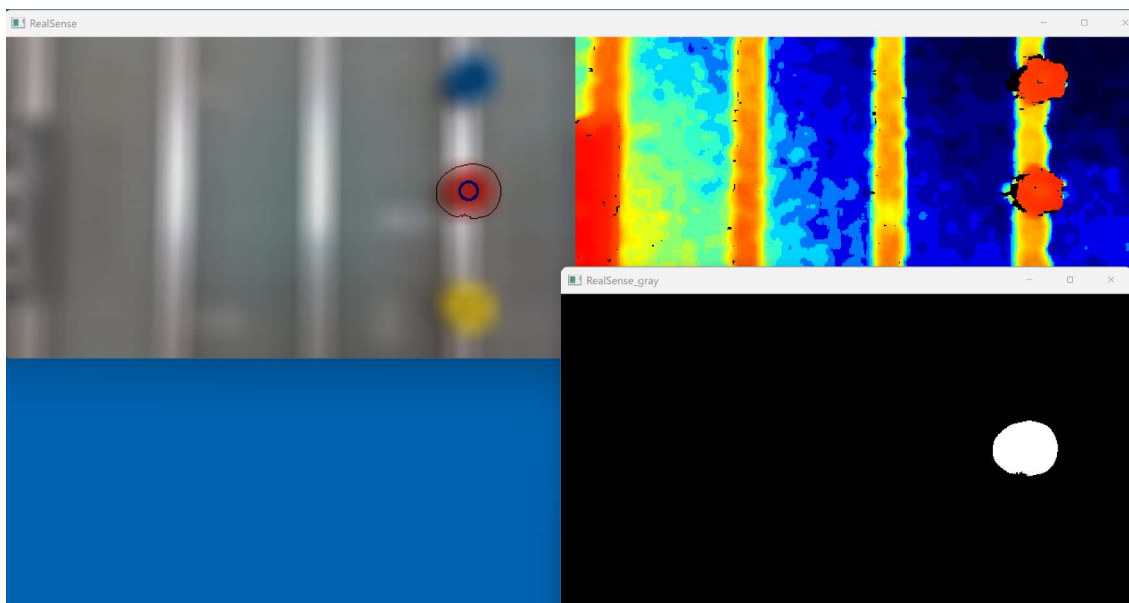


図 6.3 赤平均フィルタ

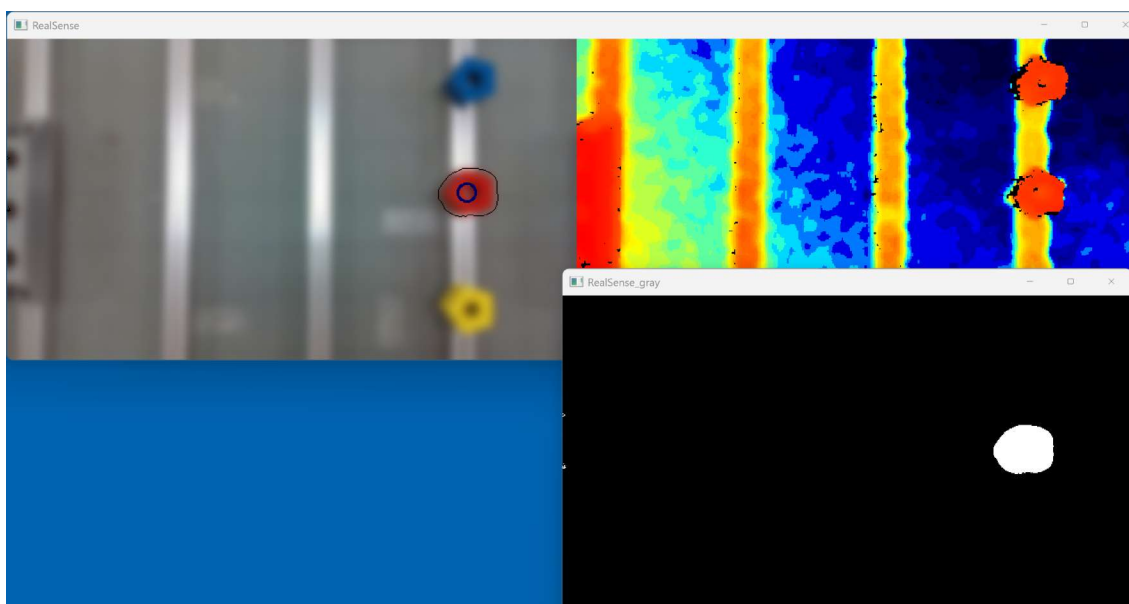


図 6.4 赤ガウシアン

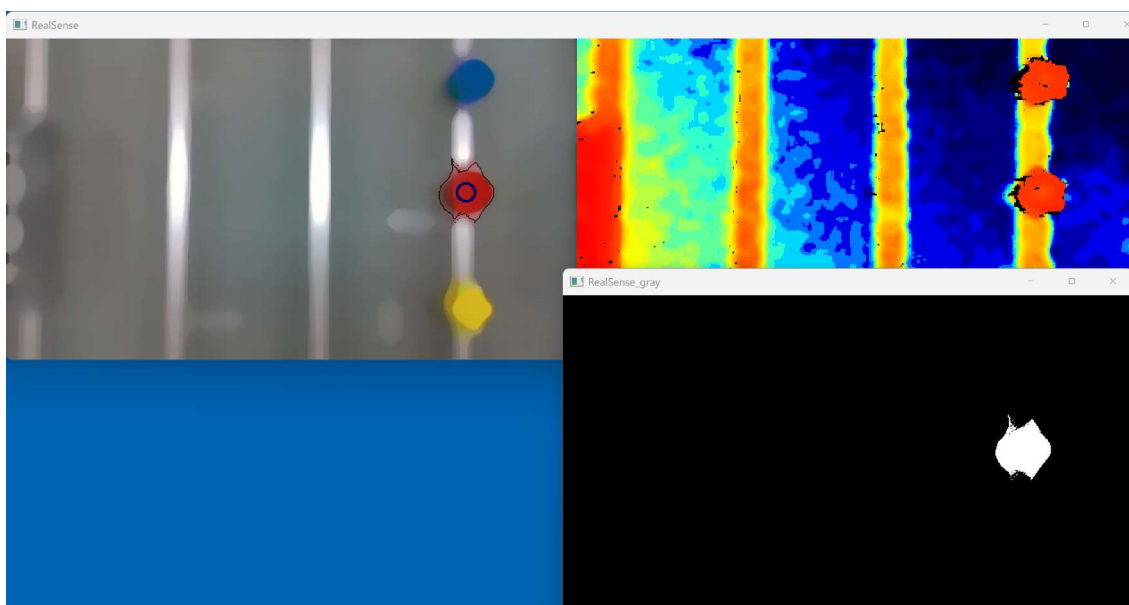


図 6.5 赤メディアン

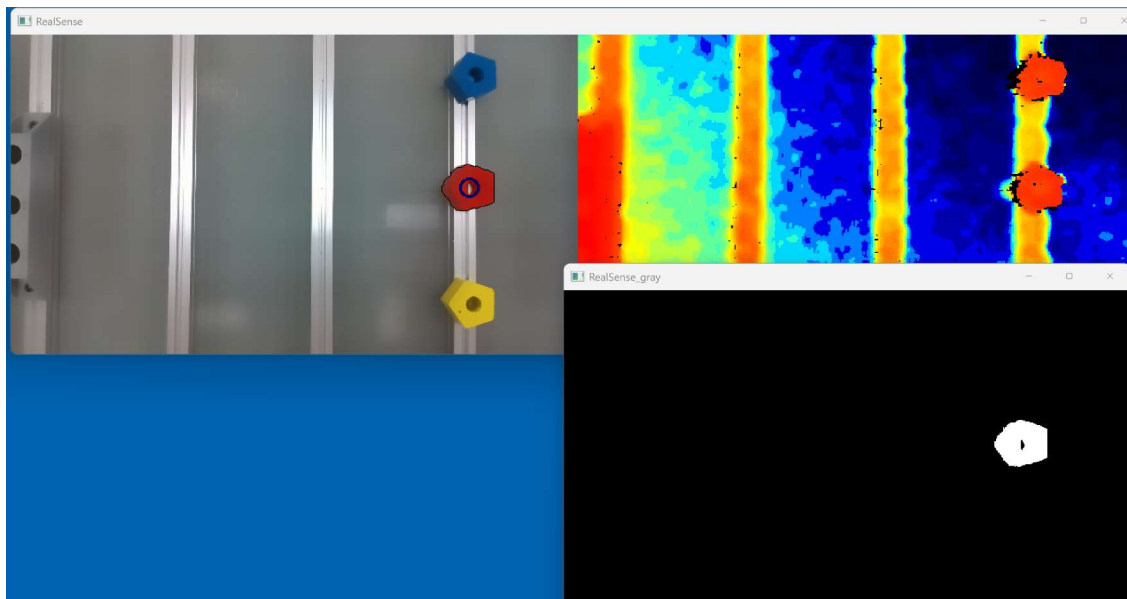


図 6.6 赤双方向

6.4 考察

本実験において、実位置と計測結果の誤差が生じた要因を以下に考察する。

フィルタリング処理の種類により、画像の平滑化やノイズ除去の効果が異なるため、計測位置に差異が生じたと考えられる。たとえば、メディアンフィルタはランダムノイズの除去に優れる一方で、エッジ部分を損なう可能性がある。一方、ガウシアンフィルタや双方向フィルタは全体的な平滑化に優れるが、細かなエッジ情報を維持しにくい場合がある。これらの特性が、計測位置の差異を引き起こした要因と考えられる。

他には、カメラの位置によってブロックの見え方や影響が変わり、二値化画像の生成に影響を与えた可能性がある。例えば、カメラがブロックに対して角度を持って設置されている場合、画像上の歪みが発生し、計測位置に誤差が生じたと推測される。また、光源の位置や反射が二値化結果に影響を及ぼし、HSV 空間での閾値設定にも影響を与えた可能性がある。

空間フィルタリングは、二値化処理の前段階でノイズを除去し、画像のエッジを強調する役割を持つ。フィルタ処理を行わない場合、ノイズがそのまま画像に残り、二値化処理の際にエッジが不正確になる可能性が高い。平均フィルタは全体的な平滑化効果があるが、エッジ部分も平滑化されるため、ブロックの輪郭がぼやける可能性がある。ガウシアンフィルタは画像全体を滑らかにするが、エッジのぼやけは平均フィルタより少ない。ただし、ノイズの除去が不完全な場合がある。メディアンフィルタはランダムノイズの除去に優れるため、二値化画像のエッジが比較的是っきりする。ただし、エッジ付近で異常値を検出する可能性がある。双方向フィルタは、ノイズ除去とエッジ保持のバランスが良い。本実験では平均フィルタを用いた例が1番誤差が少なかったが、これはぼやかすことにより、より測定した中心がより中心に近づいたからだと考えられる。

7 実験 2-2 : 3 次元位置計測と物体の把持収納実験

7.1 実験概要

本実験では, 手先に装着した RGB-D カメラから把持物体および収納位置の 3 次元位置情報を取得し, 物体の把持および収納する実験を行う. 図 7.1 に実験環境の模式図を示す. マニピュレータの卓上には, 把持する五角形の積み木 (赤色, 青色, 黄色) と収納位置用の積み木 (黄, 青, 緑) が置かれている. マニピュレータは五角形形の積み木を検出・把持して収納位置に配置する動作を行う.

7.2 実験手順

実験手順は以下の手順で行う.

- (1) 卓上に積み木を 3 個, 収納台座をすべて無造作に配置する.
- (2) Python の開発環境 Spyder を起動し, 実験 3 において最も優れた結果となった空間フィルタリングを選択する.
- (3) 画像処理プログラムを実行後, マニピュレータの制御用ソフトウェアを用いてマニピュレータを手動操作し, 画像に各物体が現れる位置まで手先を移動する. この際, 手先の位置姿勢を記録する.
- (4) 実験 3 で記録した HSV 色空間のしきい値を設定し, 各物体の検出を行う. また, 実験 3 で作成したエクセルを用いて, 各物体の 3 次元位置をグローバル座標系で表現する.
- (5) マニピュレータの制御用ソフトウェアを用いて, 手先位置の数値指定によるオンラインティーチングを行う. 動作は, 赤積み木の把持収納, 青積み木の把持収納, 黄積み木の把持収納をすべて連続で行う. ただし, 手先の姿勢 $(R, P, Y) = (180, 0, 0)$ [deg] とし, 手先の 0° 位置を 80[mm] (把持時), 110[mm] (収納時) とすること. また, 動作の様子を動画撮影する.

7.3 実験結果

表 7.1 に実験結果を示す. また, 図 7.2 にマニピュレータが積み木を把持・収納する様子を示す.

表 7.1 物体の位置

	赤積み木	青積み木	黄積み木	赤収納	青収納	黄収納
$(x_0^\circ, y_0^\circ, z)$ [mm]	401.1, 0.0, 418.2					
(R, P, Y) [deg]	180, 0, 0					
$(x_e^\circ, y_e^\circ, z)$ [mm]	84.5, 130.3, 37	182.7, 131.5, 377.9	-150, 124.7, 370.9	(177.7, 126.6, 375.9)	(88.7, -129.5, 372.9)	(-5.7, -128.8, 365.9)
$(x_0^\circ, y_0^\circ, z)$ [mm]	482.6, -130.3, 38.3	583.1, -131.5, 41.3	386.1, -124.7, 47.3	(578.8, -126.6, 42.3)	(489.9, 129.545.3)	(395.4, 128.852.3)
成功	成功	成功	成功	成功	成功	失敗



つかみ動作



収納動作

図 7.1 保持する様子

平均フィルタで得られた位置をもとにロボットアームを動作させた。少し掴むものからずれていても、アームの手を閉じると掴むことができていた。

7.4 考察

次に、考察を行う。本実験で使用したロボットアームのグリップは、把持時に「手」を閉じる動作を通じて対象物を固定する構造を持っている。この構造により、対象物の位置に多少の誤差が生じた場合でも、把持を成功させることが可能である。特に、赤積木と青積木は手先位置が完全に一致しない場合でも、グリップが自動的に物体を挟む動作により、成功率を高める結果となった。

8 課題

8.1 逆運動学の導出

8.1.1 式 (2.6) の導出

手先位置 $\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$ がグローバル座標系で与えられているとき、座標系 Σ における手先位置は以下のように変換される：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 - l_1 \end{bmatrix} \quad (2.4)$$

ここで, xy 平面に注目すると, 関節角度 θ_1 は, x 軸と手先位置 $\begin{bmatrix} x \\ y \end{bmatrix}$ がなす角度として定義される. $\tan^{-1}(y/x)$ では範囲が $[-\pi/2, \pi/2]$ に制限される場合があるため, atan2 を用いて次の式を得る:

$$\theta_1 = \text{atan2}(y, x) \quad (2.6)$$

8.1.2 式 (2.12) の導出

xy 平面における原点から手先位置への距離 (射影距離) を計算する. これはピタゴラスの定理を適用することで下のように求められる:

$$|x'| = \sqrt{x^2 + y^2} \quad (2.7)$$

正負の符号を考慮して絶対値を外すと,

$$x' = \pm \sqrt{x^2 + y^2} \quad (2.8)$$

次に, $x'z'$ 平面に注目する. l_2, l_3, x', z からなる三角形を考えると, 余弦定理を適用できる. 余弦定理の一般形は次の通り:

$$c^2 = a^2 + b^2 - 2ab \cos \theta \quad (2.9)$$

ここで,

$$a = l_2 \quad (\text{第2関節から第3関節までのリンク長})$$

$$b = l_3 \quad (\text{第3関節から手先位置までのリンク長})$$

$$c = \sqrt{x'^2 + z^2} \quad (x'z' \text{ 平面上の手先位置から原点までの距離})$$

$$\theta = \varphi_2 - \theta_3 (\text{角度})$$

を代入する. このとき, 次の式が成立する:

$$x'^2 + z^2 = l_2^2 + l_3^2 - 2l_2l_3 \cos(\varphi_2 - \theta_3)$$

これを変形すると,

$$\cos(\varphi_2 - \theta_3) = \frac{l_2^2 + l_3^2 - x'^2 - z^2}{2l_2l_3} \quad (2.10)$$

さらに, 三角関数の関係式,

$$\sin^2 \theta + \cos^2 \theta = 1$$

を用いて, $\sin(\varphi_2 - \theta_3)$ を次のように表せる:

$$\sin(\varphi_2 - \theta_3) = \pm \sqrt{1 - \cos^2(\varphi_2 - \theta_3)} \quad (2.11)$$

ここで, $\cos(\varphi_2 - \theta_3)$ と $\sin(\varphi_2 - \theta_3)$ が求まったので, atan2 を適用して θ_3 を計算する:

$$\theta_3 = \varphi_2 - \text{atan2}\{\sin(\varphi_2 - \theta_3), \cos(\varphi_2 - \theta_3)\} \quad (2.12)$$

8.1.3 式 (2.13) ~ 式 (2.14) の導出

関節角度 θ_2 を求めるために, 順運動学の式を逆に利用する. リンク 2 とリンク 3 の動作を順運動学で記述すると, x' 軸方向と z 軸方向の動きは次のように表される :

$$x' = K_c \cos(\varphi_1 - \theta_2) - K_s \sin(\varphi_1 - \theta_2) \quad (2.13a)$$

$$z = K_s \cos(\varphi_1 - \theta_2) + K_c \sin(\varphi_1 - \theta_2) \quad (2.13b)$$

ここで, 以下の定義を用いる :

$$K_c = l_2 + l_3 \cos \theta_3, \quad K_s = l_3 \sin \theta_3$$

これらを用いて, $\tan(\varphi_1 - \theta_2)$ を計算するために正弦と余弦の比を取る.

式 (2.13a) を $\cos(\varphi_1 - \theta_2)$ で整理する :

$$x' = K_c \cos(\varphi_1 - \theta_2) - K_s \sin(\varphi_1 - \theta_2)$$

$$\Rightarrow x' + K_s \sin(\varphi_1 - \theta_2) = K_c \cos(\varphi_1 - \theta_2)$$

$$\Rightarrow \cos(\varphi_1 - \theta_2) = \frac{x' + K_s \sin(\varphi_1 - \theta_2)}{K_c}$$

次に, 式 (2.13b) を $\sin(\varphi_1 - \theta_2)$ で整理する :

$$z = K_s \cos(\varphi_1 - \theta_2) + K_c \sin(\varphi_1 - \theta_2)$$

$$\Rightarrow z - K_s \cos(\varphi_1 - \theta_2) = K_c \sin(\varphi_1 - \theta_2)$$

$$\Rightarrow \sin(\varphi_1 - \theta_2) = \frac{z - K_s \cos(\varphi_1 - \theta_2)}{K_c}$$

これらを用いて, 正弦と余弦の比を計算する :

$$\tan(\varphi_1 - \theta_2) = \frac{\sin(\varphi_1 - \theta_2)}{\cos(\varphi_1 - \theta_2)}$$

$\sin(\varphi_1 - \theta_2)$ と $\cos(\varphi_1 - \theta_2)$ をそれぞれ代入する :

$$\tan(\varphi_1 - \theta_2) = \frac{\frac{z - K_s \cos(\varphi_1 - \theta_2)}{K_c}}{\frac{x' + K_s \sin(\varphi_1 - \theta_2)}{K_c}}$$

分母と分子を整理する :

$$\tan(\varphi_1 - \theta_2) = \frac{z - K_s \cos(\varphi_1 - \theta_2)}{x' + K_s \sin(\varphi_1 - \theta_2)}$$

ここで, $\sin(\varphi_1 - \theta_2)$ と $\cos(\varphi_1 - \theta_2)$ を具体的に代入する代わりに, 上記を簡略化して次の形で整理する :

$$\tan(\varphi_1 - \theta_2) = \frac{K_s x' + K_c z}{K_c x' - K_s z}$$

$\tan(\varphi_1 - \theta_2)$ を atan2 を用いて直接表現すると：

$$\theta_2 = \varphi_1 - \text{atan2}(K_s x' + K_c z, K_c x' - K_s z) \quad (2.14)$$

8.2 回転行列とオイラー角の変換式

課題：式 (2.17) ~ 式 (2.20) を導出せよ。手書き可。ただし、枠や枠内の文章は報告書に記載不要とする。参考文献は報告書最後の参考文献の欄に記載すること。

固定角を用いて姿勢を表現する際、回転行列 \mathbf{R} は x 軸 (roll, ϕ)、 y 軸 (pitch, θ)、 z 軸 (yaw, ψ) の回転を順に適用して得られる。各軸周りの回転行列を以下のように定義する：

z 軸周りの回転 (yaw, ψ)：

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

y 軸周りの回転 (pitch, θ)：

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

x 軸周りの回転 (roll, ϕ)：

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

これらを順に掛け合わせることで、全体の回転行列 \mathbf{R} が得られる：

$${}^0\mathbf{R} = R_z R_y R_x$$

まず、 R_z と R_y を掛け合わせる：

$$R_z R_y = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

行列の積を成分ごとに計算する：

$$(1,1) \text{ 成分} : \cos \psi \cdot \cos \theta + (-\sin \psi) \cdot 0 + 0 \cdot (-\sin \theta) = \cos \psi \cos \theta$$

$$(1,2) \text{ 成分} : \cos \psi \cdot 0 + (-\sin \psi) \cdot 1 + 0 \cdot 0 = -\sin \psi$$

$$(1,3) \text{ 成分} : \cos \psi \cdot \sin \theta + (-\sin \psi) \cdot 0 + 0 \cdot \cos \theta = \cos \psi \sin \theta$$

以下同様に計算を進めると：

$$R_z R_y = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi & \cos \psi \sin \theta \\ \sin \psi \cos \theta & \cos \psi & \sin \psi \sin \theta \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

次に、この結果に R_x を掛け合わせる：

$${}^0\mathbf{R} = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi & \cos \psi \sin \theta \\ \sin \psi \cos \theta & \cos \psi & \sin \psi \sin \theta \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

行列積を成分ごとに計算すると：

$$(1, 1) \text{ 成分} : \cos \psi \cos \theta \cdot 1 + (-\sin \psi) \cdot 0 + \cos \psi \sin \theta \cdot 0 = \cos \psi \cos \theta$$

$$(1, 2) \text{ 成分} : \cos \psi \cos \theta \cdot 0 + (-\sin \psi) \cdot \cos \phi + \cos \psi \sin \theta \cdot \sin \phi = \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi$$

以下同様に計算を進め、最終的に次の結果を得る：

$${}^0\mathbf{R} = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.17)$$

8.2.1 式 (2.18) の導出

回転行列の (3, 1) 成分に注目すると：

$$\sin \theta = -r_{31}$$

また、(1, 1) および (2, 1) 成分を用いると：

$$\cos \theta = \sqrt{r_{11}^2 + r_{21}^2}$$

よって、 θ は以下のように表される：

$$\theta = \text{atan2} \left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2} \right) \quad (2.18)$$

8.2.2 式 (2.19) の導出

ヨー角 ψ は回転行列の (1, 1) および (2, 1) 成分を用いて次のように定義される：

$$\tan \psi = \frac{r_{21}}{r_{11}}$$

これを atan2 を用いて表現すると：

$$\psi = \begin{cases} \text{atan2}(r_{21}, r_{11}) & \text{if } \cos \theta > 0, \\ \text{atan2}(-r_{21}, -r_{11}) & \text{if } \cos \theta < 0 \end{cases} \quad (2.19)$$

8.2.3 式 (2.20) の導出

ロール角 ϕ は、回転行列の (3, 2) および (3, 3) 成分を用いて次のように定義される：

$$\tan \phi = \frac{r_{32}}{r_{33}}$$

これを atan2 を用いて表現すると：

$$\phi = \begin{cases} \text{atan2}(r_{32}, r_{33}) & \text{if } \cos \theta > 0, \\ \text{atan2}(-r_{32}, -r_{33}) & \text{if } \cos \theta < 0 \end{cases} \quad (2.20)$$

8.3 空間フィルタリング

平均化フィルタ

平均化フィルタは、注目画素を中心に一定範囲内のピクセルの輝度値の平均を計算し、それを新しい値として置き換える方法である。このフィルタはノイズの低減に有効ですが、エッジ部分も平滑化されるため、画像の細部が失われやすいという欠点がある。適用範囲が広いほど画像がぼやける傾向があります。

ガウシアンフィルタ

ガウシアンフィルタは、ガウス関数に基づく重みを用いて近傍画素の平均値を計算する方法である。平均化フィルタよりも自然な平滑化が得られるため、ノイズ低減とエッジの保持をある程度両立できる特性を持つ。ただし、画像全体を滑らかにするため、エッジ部分がぼやけることがある。

メディアンフィルタ

メディアンフィルタは、注目画素の近傍領域のピクセル値の中央値を新しい値として置き換える方法である。このフィルタは、ランダムノイズ（特にスパイクノイズ）の除去に非常に有効である。一方で、画像のエッジ部分を比較的保持する特徴があるが、大規模な平滑化には適していない。

双方向フィルタ 双方向フィルタは、空間距離と輝度の類似性を考慮して加重平均を行うフィルタである。ノイズを除去しながらエッジを保持する性能が高く、画像の細部を維持したまま平滑化可能である。

8.4 色空間

RGB 色空間は、光の三原色である赤（Red）、緑（Green）、青（Blue）の成分を組み合わせる色を表現する加法混色のモデルである。各成分は通常 0～255 の範囲で表され、すべての成分が最大値の場合は白、最小値の場合は黒を表す。RGB 色空間はモニターやカメラなどのディスプレイデバイスで直接使用されるため、画像処理で広く採用されているが、人間の視覚的な感覚に直感的ではない点が欠点である。

HSV 色空間は、人間の色覚に基づいて設計された色空間であり、色相（Hue）、彩度（Saturation）、明度（Value）の 3 つのパラメータで色を表現する。色相は 0～360 度で色の種類を表し、彩度は色の鮮やかさ、明度は色の明るさを示す。HSV 色空間は、色相と明るさを分離できるため、画像処理において物体検出や色分類に適している。特に、背景の影響を受けにくいという利点がある。

8.5 ロボットビジョンの実用例

画像処理技術とロボットアームを複合させたものとして、例えば手術支援ロボット「ダヴィンチ」、定置型イチゴ収穫ロボット、電王手くんなどが挙げられる。

手術支援ロボット「ダヴィンチ」は、単眼カメラやステレオカメラを搭載し、外科手術の際に高精度かつ微細な操作を可能にするロボットである。カメラによる 3 次元視覚情報を用いて、執刀医にリアルタイムで視覚フィードバックを提供する。また、ロボットアームを使用して細かな縫合や切開を行うことで、人間の手では困難な作業を実現している。特に、内視鏡手術などの低侵襲手術で活用されており、患者の回復時間短縮や負担軽減に寄与している。

定置型イチゴ収穫ロボットは、ステレオカメラを用いて果実の位置や熟度を判定し、自動で収穫作業を行うロボットである。カメラで得られた 3 次元位置情報に基づき、ロボットアームが正確にイチゴを把持して収穫する。果実を傷つけないように柔らかく握るグリッパや刃物を用いる点が特徴である。

電王手くんは、ロボットアームと単眼カメラを組み合わせた将棋対局用のロボットである。カメラが盤面の駒の配置

を認識し、ロボットアームが指示された手を実行する仕組みを備えている。この技術により、人間との対局や AI 同士の試合の可視化が可能となる。駒を正確に把持して適切な位置に配置する高度な制御技術が使用されており、将棋ファンや教育用途での活用が進んでいる。

参考文献

- 1) 細田耕：実践ロボット制御, オーム社,2019.
- 2) 株式会社アールティ：3 自由度ロボットの順運動学と逆運動学, < <https://rt-net.jp/humanoid/archives/2652> > (2024 年 4 月 1 日閲覧).
- 3) 酒井幸市：デジタル画像処理入門, コロナ社,2001.
- 4) 北山直洋：Python で始める OpenCV 4 プログラミング, 株式会社カッタシステム,2019.
- 5) 画像解析におけるフィルタ処理 - ノイズ軽減フィルタのまとめ
<<https://www.mitani-visual.jp/mivlog/imageprocessing/filter-summary.php>> (2024 年 11 月 28 日閲覧).
- 6) LITTLE HEROES:RGB って何? HSV,HSL との違いは?
<<https://www.littleheroes.jp/w/articles/ac8b3376-5775-42a7-8bce-5f5f0442b877>>(2024 年 11 月 28 日閲覧).
- 7) スマート農業 360：定置型イチゴ収穫ロボット< <https://smartnogyo.com/archives/383.html> > (2024 年 11 月 28 日閲覧).
- 8) DENSO: < <https://www.denso.com/jp/ja/about-us/sponsor/denoute/denoute-kun/> > (2024 年 11 月 28 日閲覧).
- 9) 東京医科大学病院：手術支援ロボット< <https://hospinfo.tokyo-med.ac.jp/davinci/> > (2024 年 11 月 28 日閲覧).