

仮想化環境にて学習したビジュアルサーボによる机上物体の把持

○岩崎拓也（信州大学） 山崎公俊（信州大学）

1. 緒言

ロボットの主要な作業の一つにピッキングがある。組み立て作業などでは、部品をピッキングすることから始まる。ピッキングの際、ロボットはエンドエフェクタを把持対象に近づけ、対象に対して正しい姿勢で把持動作を実行しなければならない。対象物に対して、目標の姿勢をとる方法としてビジュアルサーボがある。これは、現在のカメラの情報から特徴点を抽出し、あらかじめ保持していた目標姿勢時の特徴点の位置と、実際の特徴点の位置が合致するようにロボットを動作させる手法である。ビジュアルサーボは、現在画像が入力されれば、即時に動作が可能という利点がある。一方で、カメラに映る環境が変化すると、特徴点の抽出が正しく行えないなどの欠点もある。

徳田ら[1]は、ロボットの動作と動作前後での画像のペアを学習データとして学習することで、深層学習によるビジュアルサーボを実装した。このビジュアルサーボにより、正確な位置決めを達成しているが、学習した環境以外での位置決めは難しいと考えられる。Sergeyら[2]は、多数のロボットを用いて把持動作を試行し、大量の学習データを収集することで、カメラの情報から、ロボットの動作に対する把持成功率を算出できるようにした。そして、このニューラルネットを用いた手法により、新規の物体に対しても把持を成功させた。しかし、学習には大量の学習データが必要とされており、実装には大きなコストがかかる。

本研究の先行研究として、川越ら[3]は、シミュレーション環境にて位置決め動作を生成する手法を提案した。この手法では、シミュレーション環境に現実の環境を再現し、再現したシミュレーション環境上で、ビジュアルサーボを実行する。これにより、現実環境での背景や目標物に付与された模様の変化に影響されずに位置決めをすることを可能とした。また、ビジュアルサーボにCNNを用いることで、目標物の形状の変化にも対応可能にしている。筆者らはこれに加え、障害物が存在する場合にも対応可能にする手法を検討してきた[4]。

本研究の目的は、先行研究の手法[4]を実環境での物体把持に適用するための手法の拡張である。川越らが提案した机上の再現手法では、把持対象物を人の手により識別しなければならなかった。提案手法では、把持目標物とそれ以外の障害物とを自動で識別し、それに対応した環境をシミュレーション環境に再現することができる。また、先行研究のビジュアルサーボ手法[4]に改良を加えることで、より多くの環境に対応できるようにする。

物体把持までの流れ

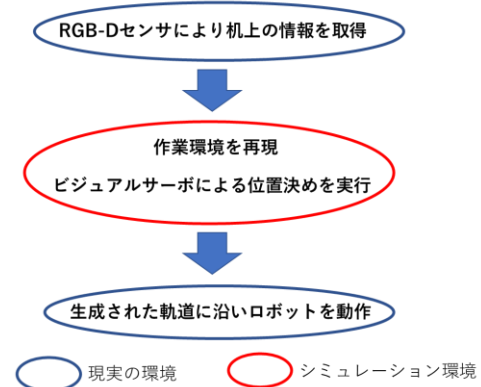


図1 提案手法の流れ

2. 問題設定とアプローチ

2.1 問題設定

本研究では、机上に置かれた対象物を把持することを目的とする。机上には、対象物のほかにも物体が存在することを想定する。

2.2 アプローチ

上記の問題設定にて対象物を把持するためには、机上の多様な環境に対応可能な位置決めが必要である。提案手法はこれを解決するために図1に示すアプローチをとる。図1の青枠で囲まれた部分は、現実の環境にて実行され、赤枠で囲まれた部分は、シミュレーション上で実行される。机上の環境をシミュレーション上で再現し、そこでおこなうビジュアルサーボにより、対象物把持のための位置決め動作を生成する。最後に、生成された軌道に沿って実機のロボットを動作させることで位置決めを実現する。

3. 提案手法

3.1 ビジュアルサーボ

3.1.1 ビジュアルサーボの全体構成

図2にビジュアルサーボの全体構成を示す。このビジュアルサーボでは、二つのCNNを用いている。ひとつめのCNNは、現在のカメラ画像と目標姿勢時のカメラ画像を入力として受け取り、目標姿勢までの手先の姿勢変位を出力する。二つ目のCNNでは、現在のカメラ画像を入力として受け取り、球面上に一樣に配置されたそれぞれの点の方向について、障害物が存在するかどうかの評価値を算出する。次にこれらの出力から、ステップ毎の目標姿勢変位 \mathbf{v}_g を決定する。しかし、この \mathbf{v}_g は、ロボットにより実現可能な量とは限らない。また、手先の動きにより、

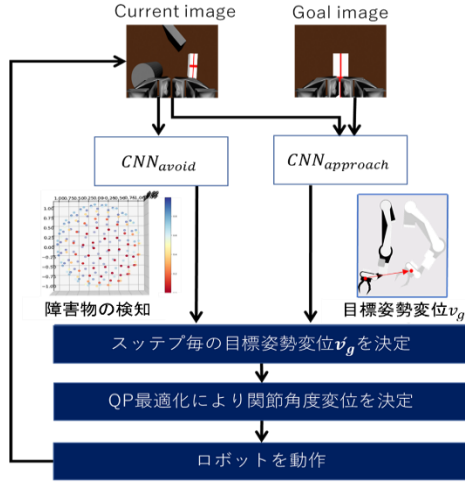


図2 ビジュアルサーボの全体図

対象物がカメラの視界から外れてしまい、ビジュアルサーボを破綻させてしまう可能性がある。そこで、各関節角の可動範囲と、対象物をカメラの視界に収めるための制約を満たすように、二次計画法を適用し最終的な関節角度の変位を算出する。

3.1.2 CNNによる目標姿勢変位の生成

最終的な目標姿勢変位を算出する $CNN_{approach}$ は、手先の姿勢変位 $\mathbf{v}_g = (v_x, v_y, v_z, v_\phi, v_\theta, v_\psi)$ を出力する。ネットワークの構造は、川越ら[3]の研究において使用されたものと同じものを用いた。CNNの学習は、シミュレーション上で手先をランダムに動かし、その前後の画像と手先の姿勢変位を取得する。このようにして得た画像ペアを入力とし、手先の姿勢変位を出力とし学習をおこなう。

3.1.3 CNNによる障害物の回避

障害物の方向を検出する CNN_{avoid} は、手先を中心とした、球面上に一樣に配置された点の方向に障害物があるかどうかの評価値を算出する。手先位置から n 番目の観測点に向かうベクトルを \mathbf{g}_n とする。そして、各障害物要素に対して、手先位置からそれぞれの要素へのベクトルを \mathbf{v}_{om} (m は障害物要素のシリアル番号) とする。これらを用いて、 \mathbf{g}_n における評価値 G_n を以下の式で算出する。

$$G_n = \max \left\{ \mathbf{g}_n \cdot \frac{\mathbf{v}_{o1}}{\|\mathbf{v}_{o1}\|}, \dots, \mathbf{g}_n \cdot \frac{\mathbf{v}_{om}}{\|\mathbf{v}_{om}\|}, 0 \right\} \quad (1)$$

上式は、 \mathbf{g}_n と障害物方向のベクトルのコサイン類似度を計算し、その中で最大のものが0を選んでいる。評価値が小さい観測点の方向ほど、障害物の衝突を起こしにくい。

3.1.4 ステップ毎の目標姿勢変位 \mathbf{v}_g の決定

手順は次のとおりである。まず評価値 G_n が所定の閾値 t_g を下回る観測点の中から、 $\mathbf{g}_n \cdot \mathbf{v}_{g(xyz)}$ の計算結果が最大となるものを選ぶ。ここで、 $\mathbf{v}_{g(xyz)}$ は $CNN_{approach}$ の出力のうちの空間変位成分である。こうして選ばれた点を \mathbf{g}_{max} とする。次に、以下のように姿勢変位ベクトル \mathbf{v}_g を構成する。

$$\mathbf{v}_g = \begin{pmatrix} |\mathbf{v}_{g(xyz)}| \mathbf{g}_{max} \\ \mathbf{v}_{g(\phi\theta\psi)} \end{pmatrix} \quad (2)$$

ここで、 $\mathbf{v}_{g(\phi\theta\psi)}$ は \mathbf{v}_g の方位変位成分である。(2)式は、手先の回転量と手先の平行移動量については、 \mathbf{v}_g の値を維持する。一方で、手先の動く方向は \mathbf{g}_{max} に入れ替えること意味している。ただし、 \mathbf{g}_{max} は障害物がない方向の中で、一番対象物に近づく方向を示しているの、障害物回避を優先したうえで、目標物に近づくことを可能にする。筆者らは、この考え方が有効であるという知見を経験から得た。

しかし、障害物の配置によっては、目標姿勢の方向の点の評価値が所定の閾値 t_g を下回り続け、デッドロックが発生してしまう現象が起こり得る。これを防ぐため、本実験では手法[4]に改良を加えてこの問題を解決した。具体的には、ステップ毎に t_g を以下のように更新するようにした。

$$t_g^{n+1} = \begin{cases} t_g^n + \varepsilon, & \text{if } \mathbf{v}_{g(xyz)} \cdot \mathbf{g}_{max} \leq 0 \\ t_g^n, & \text{if } \mathbf{v}_{g(xyz)} \cdot \mathbf{g}_{max} \geq 0 \end{cases} \quad (3)$$

ε は小さな正の定数であり、 t_g^n は n ステップ目での閾値を表している。上式は、ステップ毎の目標姿勢変位が最終的な目標姿勢の逆向きだった場合に、次ステップに障害物による制約を緩めている。この制約の緩和は、障害物に衝突する危険性を高めるが、初期の閾値 t_g^0 を小さく設定することで、多くの場合に障害物を確実に避けることが可能であることが確認されている。

3.1.5 QP最適化を用いた関節角度変位の決定

前章までの手順により得られたステップ毎の手先の姿勢変位 \mathbf{v}_g は、ロボットにより実現可能な量とは限らない。そこで、QP最適化を用いて制約を考慮した調整を行う。

QPの基本形は以下のとおりである。

$$\text{minimize } f_o = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \quad (4)$$

$$\text{s.t. } \mathbf{f}_c = \mathbf{A} \mathbf{x} - \mathbf{b} \leq \mathbf{0} \quad (5)$$

文献[4]では目的関数を次のように設定している。

$$\text{minimize } f_o = |\mathbf{v}_g - \mathbf{v}|^2 \quad (6)$$

ここで、 \mathbf{v} は求めたい姿勢変位である。また、制約として、ロボットの関節角度の可動域と、目標物をカメラの視界から外れないことを設定している。これらの制約は線形の不等式で表現可能であり、式(5)に適用することが可能である。

3.2 机上の再現

3.3.1 物体の検出

実環境を再現する際に、把持目標物とそれ以外の障害物を識別する必要がある。本実験では、YOLACT[5]を用いてインスタンスセグメンテーションをおこなうことで、物体を識別する。また、深度

画像とインスタンスセグメンテーションの結果を用いて、環境の点群から対象物の点群を抽出する。

3.3.2 物体の姿勢推定

事前に対象物の寸法を計測し、直方体や球体に近似したテンプレートとYOLACT[5]により分類された点群とでInteractive Closest Point(ICP)をすることで、姿勢推定をおこなう。

机の上にはあらかじめ想定されている対象物のほかにも物体が存在する場合が考えられる。そこで、対象物以外のものは、主成分分析により大まかに大きさの推定と姿勢推定を行い、直方体としてシミュレーション環境に出現させる。

4. 実験

4.1 実験環境

ロボットは、6自由度の腕部を持つカワダロボティクス製のHIROを使用した。また仮想環境には、オープンソースロボットシミュレータのGazebo[7]を利用した。また、三次元点群の取得に、RGB-Dセンサー (Microsoft 社製 KINECT)を用いた。

4.2 物体の姿勢推定

図3に実際に環境を再現した様子を示す。各物体が正しく配置されていることが分かる。ICPにはPoint Cloud Library[6]を利用した。

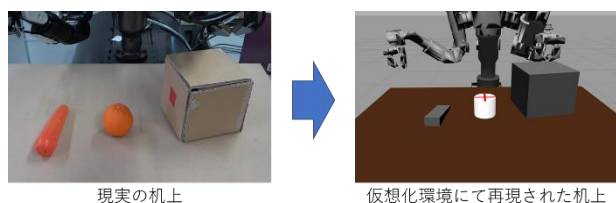


図3 机上の再現結果

4.3 把持実験

今回の実験では、把持対象物としてバナナ、オレンジ、ニンジン、食品サンプルと、携帯電話を用いた。これらの選定理由は、ロボットのハンドで把持が可能であり、姿勢推定が容易な単純な形状であることと、公式にて配布されているYOLACTの学習済みモデルにより判別可能であることである。位置決め手法が把持するのに十分な精度を持っているのかを調べるため、5種類の対象物に対して、4回ずつの計20回の実験を行った。実験方法は、まず仮想環境にて、把持動作直前の位置までの位置決め動作を生成し、生成された動作を実機のロボットに再生させる。次に、あらかじめ手動で設計された把持動作により把持を実行する。実験の結果、すべての実験で把持が成功した。このことから、物体を把持するのに十分な位置決め精度があることが確認できた。

次に、障害物がある場合の実験を行った。図4は物体の把持が成功するまでの様子を表している。ロボットの動きが(1)から(6)の順で進み、障害物を避けて把持が成功していることが分かる。

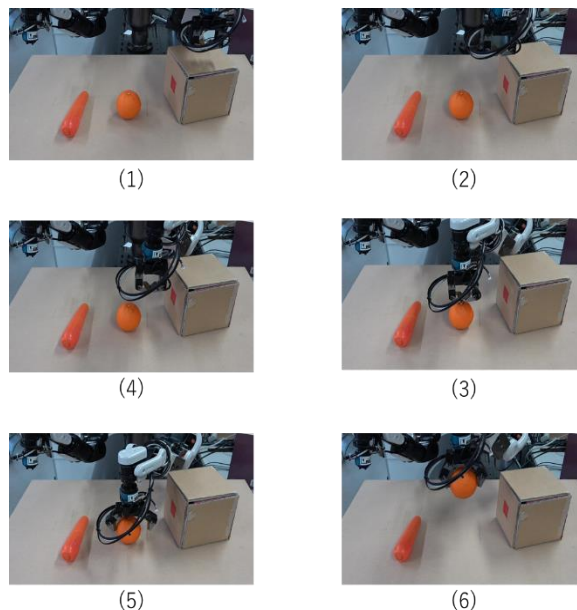


図4 実機による物体把持の様子

5. 結言

本稿では、筆者らが提案した手法[4]の改善と、実環境に応用した実験を行った。結果として、単純な形状の対象物に対して、十分な精度で把持のための位置決めが可能であることが分かった。また、障害物が存在する環境でも、障害物を避け位置決めすることに成功した。

謝辞 本研究の一部は、科研費およびJST【ムーンショット型研究開発事業】 Grant 番号【JPMJMS2034】の支援を受けた。

参考文献

- [1] 徳田冬樹, 荒井翔悟, 小管一弘: “イメージャコピアンを用いないビジュアルサーボによる位置決め,” 第36回日本ロボット学会学術講演会, 2E205, 2018.
- [2] S. Levine, P. Pastor, A. Krizhevsky and D. Quillen: “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” arXiv preprint arXiv:1603.02199, 2016
- [3] T. Kawagoshi, S. Arnold, K. Yamazaki: “Visual Servoing Using Virtual Space for Both Learning and Task Execution,” in Proc. of the 2021 IEEE/SICE International Symposium on System Integration, pp. 292-297, 2021.
- [4] 岩崎拓也, 山崎公俊: “仮想化環境での学習による多様な障害物配置および目標物形状に対応可能なビジュアルサーボ,” ロボティクス・メカトロニクス講演会, 2P2-H12, 2021.
- [5] D. Bolya, C. Zhou, F. Xiao, Y. j. Lee: “YOLACT: Real-Time Instance Segmentation,” IEEE/CVF International Conference on Computer Vision, pp9156-9165, 2019.
- [6] Point Cloud Library: <http://pointclouds.org/>
- [7] Gazebo: <http://gazebo.org/>