Vol.2021-CVIM-225 No.20
2021/3/4

VRインタフェースを用いたタスク分割デモンストレーションによる6自由度把持の学習

川上 大智[1,a)]　石川 涼一[1,b)]　ローハス メナンドロ[1,c)]　佐藤 啓宏[3,d)]　大石 岳史[1,e)]

概要：ロボットの自由度が増加するにつれ，ロボットの動作の実装は複雑かつ困難になる．本研究では，6自由度把持動作の学習に注目し，強化学習の報酬設計が容易になるように把持動作を複数のタスクに分割することを考える．模倣学習の教師データとしてのデモンストレーションデータを収集するため，人間が直感的にロボットを操作できるVRインタフェースを作成した．タスクに応じて人間が動作のデモンストレーションを行い，模倣学習を行うことで，強化学習と組み合わせた学習の効率化を実現した．

1. はじめに

複雑な動作を実現することが可能であるヒューマノイドロボットの開発が盛んになり，より高度な知能を持ったロボットが今後普及していくと思われる．ロボットが人間の代わりに，給仕や片付け，医療や災害時の救助活動など，様々な作業を行なうとき，ロボットが物体を把持する動作は，必要不可欠な動作である．

現在広く用いられている工業用のロボットは，特定の物体を掴むことを目的とした特定のロボットが，その物体を掴むことに特化した動作で物体の把持行う．特定のロボットが把持を実現するためには，把持する物体の種類ごとに，ロボットの関節の動作を実装する必要がある．これは把持する物体の種類が増えるごとに，実装の労力も増えるため，実装上の負担が大きい．また，このような実装は特定のロボットにのみ可能であり，全く別のロボットを用いる場合，またはロボットのハードウェアの一部が変化した場合，一般的に把持動作を実装し直す必要がある．

今後ますます普及していくと考えられる家庭用ロボットが扱う物体の種類には制限がなく，日々，新たな物体を扱う可能性がある．無数にある物体の一つ一つについて，物体をどのようにして掴むかということを予め実装することは不可能であると言える．このため，未知の物体に対する把持動作の自動化や，把持動作の実装の労力を最小限に抑えることが望まれる．

物体の把持を行う際に人間が利用できる情報のうち，ロボットも利用できるものは限られてくる．物体の視覚情報は，ロボットに取り付けられる光学センサーから取得可能であるが，把持対象の物体が，ロボットの手によって見えなくなるオクルージョンが発生する場合がある．また，触覚情報に関しては，触覚センサがロボットに取り付けられていない場合が多い．質量・重心の推定は，RGB画像を用いて行う試みがある [1]．しかし，例えば水筒などの，内部が不可視の物体について，視覚情報のみから質量・重心を推定することには限界がある．また，物体の摩擦力の推定についても，物体の表面状態によって摩擦力は変化するため，困難である．さらに，物体をどうやって使うかによっても，把持方法を考える必要がある．

把持を実現する既存の手法において，ロボットは，主に物体のRGBD情報と，認識した物体の姿勢情報を用いて把持を行う．これらの情報は人間が用いている情報と比較して，非常に限られたものであり，これらの情報から，ロボットが人間と同じような把持動作を実現するのは非常に困難である．

また，把持点推定の学習に用いるデータセットは，主要なものにCornell Grasp Dataset [2]があるが，これは数百種類の物体に対し，物体のRGBD情報，点群情報，及び把持点を表現した長方形が人間の手によってアノテーションされているものであり，ハードウェアの情報は含まれていない．データセットを，全てのハードウェアに対して使用可能とするためには，ハードウェアの情報を含まないように抽象化する必要がある．ハードウェアの情報を含まないデータセットを用いて，複雑な構造を持つ多指ハンドや，

[1] 東京大学
[2] LINE株式会社
[3] 京都先端科学大学
[a)] kawakami@cvl.iis.u-tokyo.ac.jp
[b)] ishikawa@cvl.iis.u-tokyo.ac.jp
[c)] menandro.roxas@linecorp.com
[d)] sato.yoshihiro@kuas.ac.jp
[e)] oishi@cvl.iis.u-tokyo.ac.jp

ⓒ 2021 Information Processing Society of Japan

柔らかい指を持つハンドなど，ハンドの把持点と角度の他に考慮しなければならない要因を含む様々なハードウェアが把持を実現することは現状困難であると言える [3,4].

データセットが入手困難である中、ロボットに把持動作を学習させる手法として，強化学習を用いることが有効であると考えられる．しかし，強化学習における報酬の設計は複雑であり，また，実際にロボットが動作する 1 回の試行にある程度時間がかかるため，十分な試行回数を重ねるためには数日から数十日単位の時間がかかる．そこで，人間によるロボット操作のデモンストレーションを，ロボットに学習させる模倣学習が提案されている [5,6].

デモンストレーションデータを収集するためには，まず人間がロボットを操作する必要がある．人間がロボット操作を行うインターフェースで実用化されているものには，Vinci Surgical System [7] などのマススタースレーブ方式のものが多く存在し，精度の良い 6 自由度のロボットハンドの操作を実現している．しかし，マススタースレーブ方式の操作インターフェースは操作者側に高価な機器を必要とし，また，操作対象のロボットと操作側の機器が対応付けられているため，多様なロボットに対応することが困難である．今後ロボットがますます普及していくことを考えると，より安価な操作インターフェースが必要となってくる．そこで，VR 空間（Virtual Reality Space）で，操作者がロボットを操作することを考える．VR 空間で 3 次元情報を操作者が認識しながら，ロボットを直感的に操作することで，6 自由度把持のデモンストレーションデータの収集を目指す．

本研究では，VR インタフェースを用いて 6 自由度把持のデモンストレーションデータを収集し，模倣学習と強化学習を組み合わせて学習を行うことを考える．

2. 関連研究

物体をどの位置で，どの方向から掴むかといった把持点推定に関する研究の多くは，把持対象の物体を真上から見た 2 次元空間において，2 次元の点，及び 1 次元の角度を与える 3 自由度把持（3 Degree of Free Grasp; 3-DOF Grasp / 2D Grasp）を扱っている [8,9]．一方，6 自由度の把持（6 Degree of Free Grasps; 6-DOF Grasps / 3D Grasp）では，3 次元の点，及び 3 次元の方向を扱う．6 自由度把持では，3 自由度把持と異なり，真上からでは掴みにくい物体を様々な方向から把持することができる．人間が自然に行っている把持動作は 6 自由度把持であり，ロボットが人間らしい動作を実現するためには，6 自由度の把持動作が必要である．

画像処理の分野で深層学習を用いた手法が成功を収めてから，3 次元の点群情報を扱う分野でも深層学習が検討されてきた．2 次元の画像情報は，深層学習モデルの入力として，画像の縦と横のピクセルを用いるため，入力のサイズは，数百×数百次元となる．対して 3 次元の点群情報は，ピクセルと同様に扱うことができない．3 次元の点群情報を深層学習モデルの入力として扱うために，ピクセルのように表現したボクセル表現や，2 次元の画像情報と深度情報を用いた 2.5 次元表現が用いられてきたが，画像処理の分野と同様に成功を収めることは困難であった．

物体の把持点推定に初めて深層学習を用いた例は，2015 年の 2D 把持点推定の研究である [10]．入力画像を 24 × 24 ピクセルと，極めて粗くすることで，モデルの入力の次元を 4000 次元程度とし，学習に成功している．これ以降，物体把持の研究で深層学習を用いた手法が検討されるようになってきた．三次元の点群情報を効率的に扱う PointNet [11] の登場後，物体の把持点推定の性能は，特に 2D 把持に関して，従来よりも非常に向上した [12]．しかし，6 自由度把持に関しては，深層学習モデルの出力の次元も 3 次元情報であるため，依然として学習は難しい．また，教師データの作成も困難である．

2D 把持の把持点推定の学習に用いるデータセットとしては，はじめに述べた Cornell Grasping Dataset を用いる研究が多い．このデータセットは実際にハードウェアを用いて把持を行っておらず，ロボットが把持把持を実現できる可能性が高い点を人間の推測によってアノテーションしている．6 自由度把持の把持点のアノテーションでは，VR インターフェースを用い，VR 空間で人間が物体の把持点を設定することで 6 自由度の把持点情報を含むデータセットを作成した例がある [13]．このデータセットは，ある特定のロボットハンドに依存しており，他の種類のロボットハンドを用いる場合に関しては検討されていない．データセットを用いた把持点推定の問題点として，データセットにハードウェアの情報が含まれていない場合や，データセットが特定のハードウェアに依存している場合，データセットを用いて多様なハードウェアに把持点を学習させることが困難であることが挙げられる．

Mousavian らは，206 種類の物体の 3D モデルを用いた，FleX [14] による 10 万回程度の物理演算シミュレーションにより，把持点を推定する試みを行った [15]．物理演算シミュレーションを用いると，人間によるアノテーションが不要である．しかし，物理演算シミュレーションでは，精度良く物理演算を行おうとすればするほど，大量の計算資源が必要である．また，実環境とシミュレーションでは，環境の変化によって差異が生じる可能性がある．

データセットが不要な把持の学習の手法として，強化学習がある．強化学習を用いたロボットの把持学習に関する既存研究では，実際にロボットに物体を把持させる試行を数十日単位の長期間行ったものがある [16,17]．このように，実環境でのデータの収集には長い時間がかかる．また，人間によるアノテーションされたデータを用いない強化学習では，人間らしくない動作を学習する可能性がある [18].

ⓒ 2021 Information Processing Society of Japan

人間によるロボット操作のデモンストレーションを，ロボットに学習させる模倣学習では，強化学習と比較して，ロボットの状態と動作の対応付けを短い時間で学習することができる [19] ．しかし，様々な物体の把持を学習するのに要するデモンストレーションのデータは，大きくなる傾向にある．把持の模倣学習に関する既存研究では，1 種類の物体の把持に要するデモンストレーションのデータは 10 分程度必要である [20]．模倣学習には，単に状態と行動を対応付ける Behavioral Cloning [21] や，デモンストレーションデータから報酬を推定する逆強化学習 (Inverse Reinforcement Learning) [22] があるが，より効率的に方策と報酬を学習することができる手法に，Generative Adversarial Imitation Learning (GAIL) がある [23] ．

3. タスクの分割

把持動作において，ロボットの初期姿勢から，把持位置までの各関節の一連の動作について，強化学習の報酬設計を設定することは困難である．本節では，ロボットアームの目的動作を，複数のタスクに分割することについて述べる．

目的動作のタスク T を n 個に分割することを考える．分割されたタスクを $T_i \quad (1 \leq i \leq n)$ とする．タスク T_i で満たすべきロボットハンドの制約を C_i とする．分割されたタスクについて，次が成り立つ場合，C_i は C_j と比較して制限の強い制約となる．

$$C_i \rightarrow C_j \qquad (i < j) \tag{1}$$

一般に，タスクの制約が弱い場合，タスクの制約が強い場合と比較して，動作の学習は容易である．このため，タスクの制約が徐々に強くなっていくようにタスクを分割することで，最終的な目的動作を効率的に学習できると考えられる．

また，ハンドの位置を特定の位置に移動させるといった場合は，あるステップで，ハンドの位置が条件を満たせばよいのみであるが，ハンドを用いて物体とインタラクションを行うといった場合，複数のステップにまたがって，条件を満たす動作を考える必要がある．あるステップで条件を満たせばよい場合と比較して，複数のステップにまたがって，条件を満たす場合の動作の学習は，著しく困難である．このため，複数のステップにまたがって動作を考える必要のあるタスクは，タスクを分割した場合，後段に設定することで，全体としての動作の学習を速めることができると考えられる．

3.1 タスク間での最適化

複数のタスクを別々に学習したのち，複数のタスク全体を通しての動作を最適化することを考える．ここでの最適化とは，タスク全体でかかる総ステップ数を減少させることとする．強化学習において，目的の動作の実現にかかるステップ数を小さくするために，1 ステップごとに微小な負の報酬を与える手法がある．本研究では，タスクごとにニューラルネットワークを構成することを考える．タスクごとにニューラルネットワークを構成した場合，それぞれのタスクで，1 ステップごとに負の報酬を与えることで，各タスクの最適化を図ることができる．しかし，各タスクが，1 つのタスク内で最適化されていても，複数のタスク間で最適化されているとは限らない．そこで，それぞれのタスクについて，次に行うタスクの完了時にその報酬和を加えることで，タスク間での最適化を図る．

3.2 段階的学習

タスクを分割することで，分割する前と比較して，強化学習の報酬設計が容易になる場合がある．考慮すべき報酬の種類数が少なるなるほど，報酬設計は容易になると考えられるが，依然として，複数の報酬が衝突し，目的動作の学習を妨げる可能性がある．目的動作の学習を行うため，与える報酬の値を，段階的に変更することが有効である．目的動作を実現するためには，優先度の高い動作をまず先に学習させる必要がある．このような場合，学習の初期段階で，優先度の高い動作に関する報酬の値を大きく設定しておくことで，優先度の低い動作の学習が不十分となっても，優先度の高い動作を学習することができる．学習の途中で，優先度の高い動作を実現することができた場合，優先度の低い動作に関する報酬の値を大きくすることで，段階的に目的動作の学習を行うことが可能になる．

4. 6 自由度把持のための VR インタフェース

本節では，人間が VR 空間でロボットを操作し，ロボットが把持動作を行うことで，デモンストレーションデータを収集することについて述べる．ロボットの操作インターフェースは主に，コマンド入力方式，マスタースレーブ方式，モーションマッピングを用いた方式に大別される．また，これらを操作者が VR 空間で行う手法が提案されている [24, 25]．本研究では，物体の 6 自由度把持操作に用いるインターフェースとして，安価な機器を用いて直感的に操作できるモーションマッピングを用いた方式のものを使用する．

4.1 逆運動学の数値的解法によるロボットの操作

人間がロボットを操作し，物体の把持動作を行う際，人間の与える入力に応じてロボットハンドの 3D 空間上の位置 $\boldsymbol{p}$，及び姿勢 $\boldsymbol{\phi}$ が決定される必要がある．人間が与えるロボットハンドの 3D 空間上の目標位置を $\boldsymbol{p}_{target}$，目標姿勢を $\boldsymbol{\phi}_{target}$ とする．現在のロボットハンドの位置姿勢 $\boldsymbol{p}$，$\boldsymbol{\phi}$ から，目標位置姿勢である $\boldsymbol{p}_{target}$，$\boldsymbol{\phi}_{target}$ に変化させるために，逆運動学（Inverse Kinematics）を用いる．目標

点の次元が 6 次元であり，ロボットのアームの次元が 6 次元以上である場合，関節の角度は解析的に求まるが，わずかな目標点の変化で，その目標点を実現する関節の角度が大きく異なる場合がある．そこで，逆運動学の数値的解法であるヤコビアンを用いた逆運動学を用いる．ロボットの各関節の角度を $\boldsymbol{\theta}$ とし，$\boldsymbol{\theta}$ が微小に変化した場合の $\boldsymbol{p}$，$\boldsymbol{\phi}$ の微小変化を $\Delta\boldsymbol{q}$ とする．$\Delta\boldsymbol{q}$ は，ヤコビアン J を用いて表せる．

$$J = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{\theta}} \tag{2}$$

$$\Delta\boldsymbol{q} = \begin{bmatrix} \Delta\boldsymbol{p} \\ \Delta\boldsymbol{\phi} \end{bmatrix} = J\Delta\boldsymbol{\theta} \tag{3}$$

このとき，関節の移動方向 $\Delta\boldsymbol{\theta}$ は，ヤコビアン J の一般化逆行列 $J^\sharp$ を用いて表せる．

$$\Delta\boldsymbol{\theta} = J^\sharp \Delta\boldsymbol{q} \tag{4}$$

このようにして計算された $\Delta\boldsymbol{\theta}$ を，アームの現在の関節の角度に加算することを繰り返し，ロボットのハンドを目標点に到達させる．また，ロボットハンドの開閉動作は，ロボットのハンドの目標位置姿勢とは別の入力を与える．入力方式としては，操作者によるキー入力や，操作者の手をトラッキングして，人差し指と親指の距離から，ハンドの開閉動作を行うといった方式が挙げられる．

4.2 ロボットの操作を行う VR インタフェース

人間がロボットを操作する際，VR 空間を用いることで，直感的な効率的な学習ができるとされている [25, 26]．VR 空間では，予めロボットの 3D モデルを作成しておき，現実空間のロボットの各関節角に応じて，対応する VR 空間のロボットの各関節角を設定することで，現実空間と VR 空間の対応付を行う．また，ロボットに取り付けられた 1 つまたは複数の RGBD センサにより，ロボットを取り巻く環境を取得し，VR 空間に投影する．ロボットが扱う物体は，位置姿勢を認識するシステムを持ちることで，予め 3D モデルを作成しておくことで，VR 空間上に表示することができる．このような VR 空間を用いる利点としては，次のものが挙げられる．

- 操作者は手に持つコントローラによる目標位置姿勢と，ロボットのハンドの現在の位置姿勢を，三次元的に把握しながら直感的な操作が可能になる．
- 操作者が自身の頭部を動かすことで，視点の変更を直感的に行うことができる．このため，ロボットによるオクルージョンにより，物体の位置姿勢を操作者が把握できなくなるような視点の状況を解決しやすい．
- 予めロボットや物体の 3D モデルを作成しておくことで，操作者が操作しているロボットや，ロボットが扱う物体の現在の形状を，ノイズなく描画することができる．

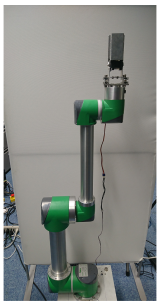


図 1: 使用したロボットアーム及びハンド

欠点としては，次のものが挙げられる．

- VR 空間と現実空間で，空間的な誤差が生じてしまい，操作者が意図どおりの操作を行うことができない可能性がある．
- 操作者は，VR ゴーグルを装着する必要がある．

近年では VR 機器がますます普及してきており，より安価で高性能な機器が入手可能になっている．現在では，数万円程度の VR 機器と，それに接続する 10 万円程度の計算機があれば，操作者側のシステムを構成することが可能である．

5. 実験

5.1 ハードウェア構成

本研究で使用したロボットアーム及びハンドを図 1 に示す．ロボットアームとして，6 自由度のマニピュレータを用いた[*1]．また，ロボットのハンドとして，平行二指ハンドを用いた．RGBD センサとしては，Realsense SR305 [*2] を用いた．

5.2 タスクの分割

本研究では，把持動作のタスクを次の 3 つに分割した．

- タスク 1 : ロボットのハンドの方向を，把持対象について予め設定した把持点に向かう方向へ向ける．
- タスク 2 : タスク 1 の制約を満たしながら，ロボットのハンドの位置を把持点へ近づける
- タスク 3 : ロボットのハンドを閉じる動作を行い，物体を把持する

分割したタスクを行う状態は，図 2 に示す流れで，状態遷移する．タスク 1 からタスク 3 に移るにつれ，タスクの学習の困難さは増加していく．

5.3 段階的学習における報酬設計

本節では，タスクごとの報酬設計について述べる．学習状況に応じて報酬の値を変更するほか，報酬の種類を変更する段階的学習を行う．

[*1] 日本電産株式会社, i611 ロボット
[*2] https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

ⓒ 2021 Information Processing Society of Japan

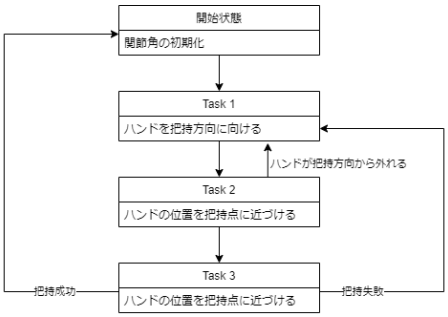


図 2: タスクの状態遷移

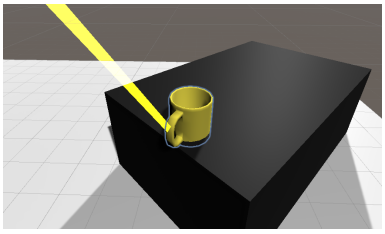


図 3: 把持方向の表示

タスク 1

本研究では，把持対称の物体に対し，把持方向を設定する．把持方向は，図 3 に示すような，予め物体に設定した把持点へ向かう特定の向きとする．本研究では，把持対象の物体について，予め実際のロボットを操作して，把持動作を実現したときの，ハンドの位置姿勢を用いて把持方向を設定した．

タスク 1 における強化学習の報酬を与えるタイミングは次のように設定した．

- ロボットのハンドの方向が，把持方向に近づいたとき
- ロボットのハンドの位置が，把持方向に近づいたとき
- ロボットのハンドが，把持方向を向いたとき
- 一定以上のステップ数が経過したとき
- ロボットが物体，及び作業台に衝突したとき
- ロボットのハンドが物体から一定以上の距離だけ離れたとき

また，学習段階に応じて次のように報酬を変更する．

- タスク 1 の成功確率が一定以上になったとき
  動作の効率化のため，1 ステップごとに罰として一定の負の報酬を与える．
- 全てのタスクの学習が完了したとき
  タスクを通しての動作の効率化のため，1 ステップとのに罰として一定の報酬を与えていたのをやめる．また，タスク 2 終了時にタスク 2 の報酬和を与える．

タスク 2

タスク 2 における強化学習の報酬を与えるタイミングは次のように設定した．

- ロボットのハンドが把持点に近づいたとき
- ロボットのハンドが把持方向を向いていないとき
- 一定以上のステップ数が経過したとき
- ロボットが物体，及び作業台に衝突したとき
- ロボットのハンドが物体から一定以上の距離だけ離れたとき

タスク 2 では，ロボットのハンドが把持対象の物体と衝突することを回避することが学習の過程で求められる．これを考慮して，学習段階に応じて次のように報酬を変更する．

- 一定以上のステップ数が経過したとき
  ロボットのハンドが物体に衝突しないように，衝突時の報酬によって学習している可能性があるため，物体と衝突したときの報酬を増加させる．これは，衝突時の罰の影響を軽減させることにあたる．
- タスク 2 の成功時
  タスク 2 では，物体と衝突した際に罰としての報酬を与えるが，この影響がタスク 2 の成功時に与えられる報酬と比較して小さい場合，衝突を回避することを学習しない可能性がある．このため，物体と衝突したときの報酬を減少させる．これは，衝突時の罰の影響を増加させることにあたる．
- タスク 2 の成功確率が一定以上になったとき
  動作の効率化のため，1 ステップごとに罰として一定の負の報酬を与える．
- 全てのタスクの学習が完了したとき
  タスクを通しての動作の効率化のため，1 ステップとのに罰として一定の報酬を与えていたのをやめる．また，タスク 3 終了時にタスク 3 の報酬和を与える．

タスク 3

タスク 3 における強化学習の報酬を与えるタイミングは次のように設定した．

- 一定以上のステップ数が経過したとき
- ロボットのハンドが物体から一定以上の距離だけ離れたとき
- 把持が成功した場合

学習段階に応じて次のように報酬を変更する．

- タスク 3 の成功確率が一定以上になったとき
  動作の効率化のため，1 ステップごとに罰として一定の負の報酬を与える．
- 全てのタスクの学習が完了したとき
  タスクを通しての動作の効率化のため，1 ステップとのに罰として一定の報酬を与えていたのをやめる．

全体を通しての学習の流れは，図 4 に示す．分割されたタスクごとに，GAIL での模倣学習を行う．各タスクの成功確率が一定以上になった場合，各タスクでの強化学習に移行する．タスク 1 の強化学習が完了したとき，タスク 1 の学習を停止し，タスク 2 の学習に移行する．タスク 2 の学習時には，タスク 1 の学習結果を用いてタスク 2 の初期状態を生成する．同様に，タスク 3 の学習時についても，タスク 1・タスク 2 の学習結果を用いて初期状態を生成す

ⓒ 2021 Information Processing Society of Japan

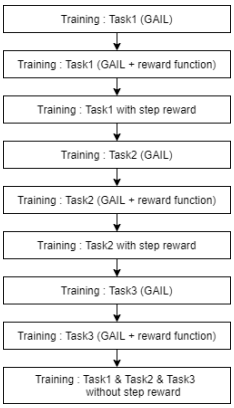


図 4: 学習の流れ

る．タスク 3 の強化学習が終了したとき，タスク全体を通しての動作の最適化に移行する．

5.4 構成したインタフェース

本研究では，模倣学習を行う際に必要となるデモンストレーションデータを収集するため，VR インタフェースを作成した．使用した VR ヘッドセットは，Oculus Quest [*3]である．

操作者は，図 5 のように，位置姿勢をトラッキングできるコントローラを用いて，ロボットを操作する．トラッキングした位置姿勢を用いて，ロボットハンドの目標位置，目標姿勢，指の開閉動作を設定する．設定した目標位置，目標姿勢から，逆運動学を用いてロボットが動かすべき各関節の角度を設定する．

本研究では，把持対象の物体に AR マーカを貼り付けることで，物体の位置姿勢を認識する．タスクの状態の表示は，ロボットのハンドの方向ベクトルを示す線の色によって行う．

VR 空間では，ロボットの目標位置姿勢を操作者が与えたのち，ロボットが実際に動く位置が計算され，3D モデルの表示に反映される．実際のロボットは，表示された 3D モデルの位置姿勢に同期させるように，各関節を動作させる．これは，疑似的なマスタースレーブ方式での操作であるといえる．本研究では，同期速度を 10Hz とした．

強化学習および模倣学習の入力特徴量として，1 ステップごとに観測する状態は，物体の位置姿勢，ロボットの各関節の位置姿勢，エンドエフェクタの位置姿勢である．また，出力特徴量として，各関節の角速度を用いる．作成したインタフェースを用いて，実際にデモンストレーションデータを収集した．

5.5 DNN アーキテクチャ

図 6 に，タスク 1 からタスク 3 で用いるネットワークアーキテクチャを示す．行動を出力するモデルは，3 層の隠れ層を持ち，各層に LSTM を用いた再帰型ニューラルネットワークとした．隠れ層ごとのノード数は 256 とした．学習アルゴリズムは，PPO（Proximal Policy Optimization）アルゴリズム [28] を用いる．

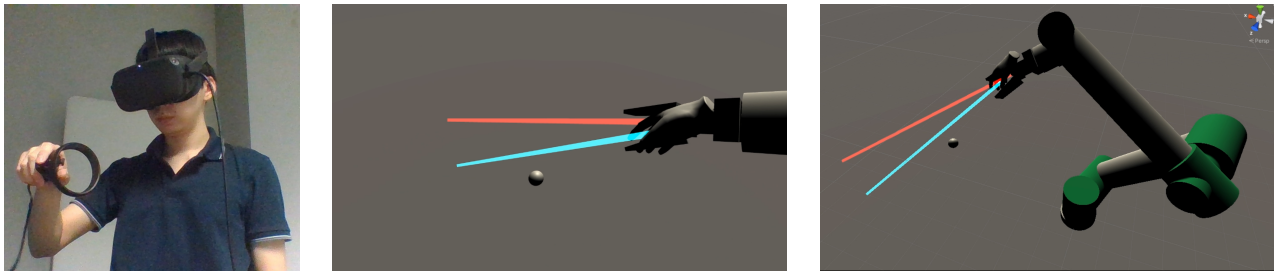
[*3] https://www.oculus.com/quest/

5.6 結果

まず，タスク 1 について，GAIL による模倣学習を行ったのち，強化学習を行った．図 7 に，学習過程における，タスク 1 について設計した報酬の，20000 ステップごとの 1 エピソードあたりの平均報酬和を示す．また，図 10 に，学習途中での 1 エピソードあたりの平均総ステップ数を示す．20 万ステップの学習ののち，GAIL による報酬に加え，設計した報酬を用いた学習に移行した．186 万ステップの学習ののち，タスク 1 の動作の最適化を行うため，1 ステップごとに微小な負の報酬を与えるように，報酬設計を変更した．266 万ステップ程度で，1 エピソードあたりの平均報酬和，平均ステップ数は増減しなくなった．ここで，タスク 1 の学習を停止し，タスク 2 の学習に移行する．図 8 に，学習過程における，タスク 2 について設計した報酬の，20000 ステップごとの 1 エピソードあたりの平均報酬和を示す．図 10 では，タスク 2 に要するステップ数が加算されている．10 万ステップの学習ののち，GAIL による報酬に加え，設計した報酬を用いた学習に移行した．136 万ステップの学習ののち，タスク 2 の動作の最適化を行うため，1 ステップごとに微小な負の報酬を与えるように，報酬設計を変更した．172 万ステップ程度で，1 エピソードあたりの平均報酬和，平均ステップ数は増減しなくなった．ここで，タスク 2 の学習を停止し，タスク 3 の学習に移行する．図 8 に，学習過程における，タスク 3 について設計した報酬の，20000 ステップごとの 1 エピソードあたりの平均報酬和を示す．図 10 では，タスク 3 に要するステップ数が加算されている．10 万ステップの学習ののち，GAIL による報酬に加え，設計した報酬を用いた学習に移行した．60 万ステップ程度で平均報酬和は上昇を留めた．最後に，タスク全体を通して動作の最適化を行うため，次のタスクの報酬和を，前のタスクの報酬に加えるように報酬設計を変更し，学習を行なった．タスク全体を通して動作の最適化を開始してから 50 万ステップ程度で，1 エピソードあたりの平均ステップ数は減少しなくなった．

以上の学習に要した総ステップ数は，460 万ステップ程度となった．最終的な 1 エピソードあたりの平均ステップ数は 80 程度となった．

タスク 1 からタスク 3 までに関して，およそ 10 万から 20 万ステップ程度で，デモンストレーションデータに存在するような動作の模倣が学習され始める．模倣学習を用いず，強化学習のみで目標の動作を学習させようとすると，数倍から数十倍のステップ数を要する場合や，そもそも学習が進まないといった問題があるため，学習の初期段階に



(a) operator (b) first-person-view (c) third-person-view

図 5: VR 空間でのロボットの操作

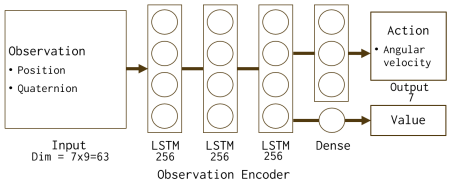


図 6: 行動を生成するネットワークアーキテクチャ

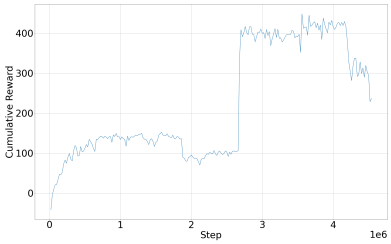


図 7: タスク 1 について 1 エピソードあたりの平均報酬和

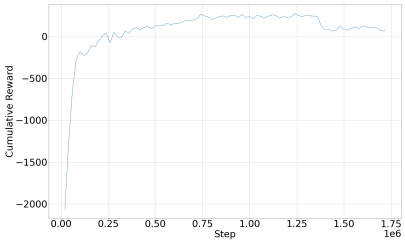


図 8: タスク 2 について 1 エピソードあたりの平均報酬和

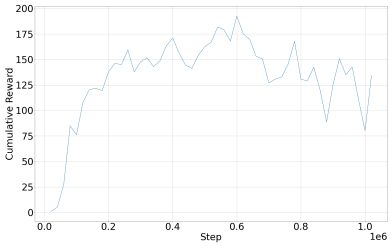


図 9: タスク 3 について 1 エピソードあたりの平均報酬和

模倣学習を用いることは，非常に有効であると考えられる．

また，デモンストレーションデータでは，1 エピソード

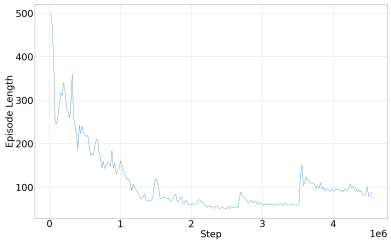


図 10: 1 エピソードあたりの平均ステップ数

を完了するためには，少なくとも 200 ステップ程度を要した．デモンストレーションデータと比較して，今回の実験で得られた最終的な動作のステップ数は，80 程度と大幅に減少した．今回の実験のように，模倣学習のみの学習で動作を学習するのではなく，強化学習を組み合わせて用いることで，目標動作に要するステップ数を減少させることが可能であることが示された．

6. まとめ

6.1 本研究の成果

本研究では，VR 空間で人間が直感的にロボットを操作し，物体把持のデモンストレーションを取得できるインタフェースを作成した．実際に，6 自由度のアーム，及びその先端に取り付けられた 1 自由度のハンドを操作するインターフェースを構築した．また，物体把持のタスクを，把持動作の流れが人間にとって理解しやすいように，また，強化学習の報酬設計が容易になるように，3 つのタスクに分割し，学習を行った．作成した VR インタフェースを用いて，操作者が分割されたタスクに応じてロボットを操作することにより，デモンストレーションデータを収集した．得られたデモンストレーションデータを用いて模倣学習を行ったのち，強化学習により把持動作の学習を行うことで，把持動作の学習を実現した．強化学習の報酬設計については，段階的に報酬を変更することで，安定した把持の学習を実現した．また，分割されたタスクについて，次のタスクの報酬和を，前のタスクの報酬として加えることで，タスク全体を通して要するステップ数を削減すること

を示した．学習された動作に要するステップ数は，デモンストレーションデータのものと比較して小さく，強化学習によって，より最適化された動作の学習を実現した．

6.2 今後の展望

本研究では，アームの基点を固定して問題を扱った．実用上，ロボット本体は移動可能であると仮定すべきである．その場合，ロボット本体やカメラの位置を，物体を把持しやすいように移動させることを考える必要がある．

また，今回の実験では，デモンストレーションデータを取得する際，ロボットを操作する人間は，著者ひとりのみであった．複数人がデモンストレーションデータを取得した場合，ロボットの操作に，操作者の癖が表れることにより，模倣学習がうまく進まない可能性が存在する．複数人で効率的にデモンストレーションデータを取得する手法についても検討すべきであると考えられる．

参考文献

[1] T. Standley, O. Sener, D. Chen, S. Savarese, “ image2mass: Estimating the Mass of an Object from Its Image,” Conference on Robot Learning, pp. 324-333, 2017.
[2] Y. Jiang, S. Moseson, and A. Saxena, “Efficient grasping from RGBD images: Learning using a new rectangle representation,” ICRA, 2011.
[3] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, “Planning multi-fingered grasps as probabilistic inference in a learned deep network,” arXiv preprint arXiv:1804.03289, 2018.
[4] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, “Learning object grasping for soft robot hands,” IEEE Robotics and Automation Letters, 3(3):23702377, 2018.
[5] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz, “Keyframe-based learning from demonstration,” International Journal of Social Robotics, vol. 4, no. 4, pp. 343355, 2012.
[6] J. Schulman, J. Ho, C. Lee, and P. Abbeel, “Learning from demonstrations through the use of non-rigid registration,” in Proceedings of the 16th International Symposium on Robotics Research (ISRR), 2013.
[7] M. Talamini, K. Campbell, and C. Stanfield, “Robotic gastrointestinal surgery: early experience and system description,”Journal of laparoendoscopic & advanced surgical techniques, vol. 12, no. 4, pp. 225232, 2002.
[8] Sulabh Kumra, Shirin Joshi and Ferat Sahin, “Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network,” arXiv:1909.04810v1, 2019.
[9] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” Robotics: Science and Systems (RSS), 2017.
[10] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” IJRR, 2015.
[11] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” CVPR, 2017.
[12] H. Liang, X. Ma, S. Li, M. Ġorner, S. Tang, B. Fang, F. Sun, and J. Zhang. “Pointnetgpd: Detecting grasp configurations from point sets,” IEEE International Conference on Robotics and Automation (ICRA), 2019.
[13] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, “Learning 6-dof grasping interaction via deep geometry-aware 3d representations,” arXiv preprint arXiv:1708.07303, 2017.
[14] M. Macklin, M. Muller, N. Chentanez, and ¨ T. Kim, “Unified particle physics for real-time applications,” ACM Transactions on Graphics (TOG), 33(4):153, 2014.
[15] A. Mousavian, C. Eppner, and D. Fox, “6-dof graspnet: Variational grasp generation for object manipulation,” arXiv preprint arXiv:1905.10520, 2019.
[16] 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.
[17] L. Pinto and A. Gupta, “Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours,” IEEE International Conference on Robotics and Automation (ICRA), pages 34063413, 2016.
[18] N. Heess, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, A. Eslami, M. Riedmiller, et al.“Emergence of Locomotion Behaviours in Rich Environments,” In: arXiv preprint arXiv:1707.02286, 2017.
[19] J. S. Dyrstad, E. Ruud ye, A. Stahl, J. Reidar Mathiassen, “Teaching a Robot to Grasp Real Fish by Imitation Learning from a Human Supervisor in Virtual Reality,” International Conference on Intelligent Robots and Systems (IROS), pp. 7185-7192, 2018.
[20] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. International Conference on Robotics and Automation (ICRA), 2018.
[21] S. Ross, G. J. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning.” in AISTATS, vol. 1, no. 2, 2011.
[22] A. Y. Ng, S. J. Russell, et al., “Algorithms for inverse reinforcement learning.” in Icml, pp. 663670, 2000.
[23] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in NIPS, pp. 45654573, 2016.
[24] C. Stanton, A. Bogdanovych, and E. Ratanasena,“Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning,”in Proc. Australasian Conference on Robotics and Automation, 2012.
[25] L. Fritsche, F. Unverzag, J. Peters, and R. Calandra, “First-person teleoperation of a humanoid robot,” in Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on. IEEE, pp. 9971002, 2015.
[26] J. I. Lipton, A. J. Fay, and D. Rus,“Baxter’s homunculus: Virtual reality spaces for teleoperation in manufacturing,”IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 179186, 2018.
[27] S. Garrido-Jurado, R. Muoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marn-Jimnez, ”Automatic generation and detection of highly reliable fiducial markers under occlusion,” Pattern Recognition, vol. 47, pp. 2280-2292, 2014.
[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

ⓒ 2021 Information Processing Society of Japan