

東京大学
情報理工学系研究科 創造情報学専攻
博士論文

タンジブルインタフェースを用いた
操縦自律間横断学習ロボットシステムの構成法
Robot system architecture of transversal learning
between maneuvering and autonomy using tangible interface

北川 晋吾
Shingo Kitagawa

指導教員 岡田 慧 教授

2021年12月

概要

ロボット工学の発展により産業用ロボットに代表されるように物体操作作業を行うロボットシステムが社会に導入されているが、多くは単純な物体操作作業を対象とした単腕ロボットシステムであり、人のように複雑な物体操作作業を行うことをロボットは期待されている。しかし、手順があり接触の多いような複雑な物体操作作業において、ロボットが自律的にこれら作業を行う際には作業や動作の複雑さから失敗や新たな作業などに対応するための作業手順の変更が課題となっている。そこで本研究では複雑な物体操作作業における作業手順の変更という課題を、操縦・自律の間を切り替えながら横断的に作業を学習する操縦自律間横断学習を行うことで解決する。したがって本研究では手順があり接触の多い複雑な物体操作作業において、操縦と自律の2つの異なる実行形態間を切り替えながら作業に関する知識を横断的に学習するロボットシステムの構築を目指し、操縦自律間横断学習ロボットシステムの構成法について論じる。

本研究では物体操作作業を行う操縦自律間横断学習ロボットシステムに必要な構成要素として、タンジブルインタフェースを用いた動作手順の教示学習手法と、実世界適応を用いた動作戦略の自律学習手法、ユーザ入力に応じた操縦自律間切替実行手法を提案する。本研究では物体操作作業について対象物の状態に基づいて作業を動作順序と動作に分割し、操縦実行において作業手順を表す動作順序を教示学習し、自律実行において動作を制御する動作戦略を自律学習し、これらを操縦自律間で共有することで操縦と自律の2つ作業実行形態間で作業に関する横断学習を実現する。まず1つ目のタンジブルインタフェースを用いた動作手順の教示学習手法は、ロボットを用いて物体操作作業の操縦実行する際に、操縦者からの接触や回転を接触できる対象物を模した形のタンジブルインタフェースを用いることで、手順があり接触の多い複雑な物体操作作業の操縦実行と作業手順を表す動作順序の教示学習を実現する手法であり、本研究では箱を用いた双腕物体操作作業において本教示学習手法の有効性を示した。そして2つ目の実世界適応を用いた動作戦略の自律学習手法は、ロボットを用いて物体操作作業の自律実行する際に、シミュレーションによる事前学習と実世界での動作試行に基づいた再学習を組み合わせた実世界適応を用いることで接触の多い動作を制御する動作戦略の自律学習を実現する手法であり、本研究では倉庫環境での双腕物体操作作業において本自律学習手法の有効性を示した。そして3つ目のユーザ入力に応じた操縦自律間切替実行手法は、上記の2つの学習手法で学習された動作順序と動作戦略を操縦自律間で共有しながらユーザの入力に応じて操縦と自律を切り替えながら作業を実行する手法であり、本研究では箱の回転積み重ね作業において本切替実行手法の有効性を示した。最後に統合実験において、上記の3手法のロボットシステムとしての統合可能性と、本研究で提案する操縦自律間横断学習ロボットシステムの構築可能性、そして作業の自律実行の途中で操縦に切り替えることによる作業手順の教示学習の効率性について箱の回転積み上げ実験と物体ダンボール箱収納作業実験にて評価した。

本研究は、操縦と自律という2つの作業実行形態を切り替えながら物体操作作業を横断的に学習するロボットシステム構成法により、操縦と自律といった異なる作業実行形態間で横断的

ii

に物体操作作業を学習するシステムが構成可能であることを示し、異なる作業実行形態を切り替えながら作業を横断的かつ効率的に学習することで、失敗や新たな作業などに対応するために作業手順の変更を即応的に学習しながら物体操作作業を実行できるロボットシステムの可能性を示したものとなっている。

目次

第 1 章	序論	1
1.1	本研究の背景と目的	1
1.2	本研究の構成	2
第 2 章	物体操作作業の操縦自律間横断学習ロボットシステム	4
2.1	はじめに	4
2.2	ロボットによる物体操作作業	4
2.2.1	物体操作作業の定義	4
2.2.2	物体操作作業の接触部位と接触点に基づく分類	4
2.2.3	本研究で取り組む複雑な物体操作作業	5
2.3	物体操作作業における作業実行形態	6
2.3.1	物体操作作業における作業実行形態の分類	6
2.3.2	物体操作作業の操縦と自律	7
2.3.3	本研究で取り組む操縦自律間で学習するロボットシステム	8
2.4	物体操作作業の操縦自律間横断学習ロボットシステム	9
2.4.1	対象物の状態に基づいた操縦自律間横断学習手法	10
2.4.2	操縦実行における作業手順の教示学習	14
2.4.3	自律実行における動作戦略の自律学習	16
2.4.4	物体操作作業の操縦自律間切替実行	17
2.4.5	操縦自律間横断学習ロボットシステムの構成	18
2.5	おわりに	21
第 3 章	操縦自律間横断学習ロボットシステムのハードウェア・ソフトウェア構成	22
3.1	はじめに	22
3.2	操縦自律間横断学習ロボットシステムのハードウェア構成	22
3.2.1	双腕ロボット Baxter-JSK のハードウェア構成	22
3.2.2	双腕移動ロボット PR2-JSK のハードウェア構成	30
3.2.3	単腕移動ロボット Fetch-JSK のハードウェア構成	32
3.3	操縦自律間横断学習ロボットシステムのためのソフトウェア基盤	35
3.3.1	分散型ロボットミドルウェア	35

iv 目次

3.3.2	動作生成・計画	36
3.3.3	環境認識	37
3.3.4	有限状態機械管理	38
3.4	おわりに	38
第4章	タンジブルインタフェースによる作業手順の教示学習	39
4.1	はじめに	39
4.2	対象物の状態に基づいた動作順序グラフによる作業手順記述	40
4.3	物体操作作業の操縦におけるユーザインタフェース	41
4.4	物体操作作業の操縦におけるタンジブルインタフェース	42
4.4.1	対象物指向型操縦と手先指向型操縦の比較	43
4.4.2	タンジブルインタフェースによる対象物指向型操縦の利点と欠点	44
4.4.3	複雑な物体操作作業におけるタンジブルインタフェース	45
4.4.4	タンジブルインタフェースを用いた対象物指向型操縦システム	46
4.5	対象物の接触・回転状態を表現するタンジブルインタフェース Miniature Tangible Cube の実装	47
4.5.1	Miniature Tangible Cube による実装の概要	47
4.5.2	近接センサによる Miniature Tangible Cube の接触計測	48
4.5.3	IMU による Miniature Tangible Cube の回転計測	50
4.5.4	Miniature Tangible Cube の回路構成	51
4.5.5	Miniature Tangible Cube の動力源と計算機	51
4.5.6	Miniature Tangible Cube のネットワーク構成とセンサデータ通信	53
4.6	センサデータに基づくタンジブルインタフェースの接触・回転状態認識	53
4.6.1	対象物とタンジブルインタフェースの回転状態	54
4.6.2	対象物とタンジブルインタフェースの周囲との接触状態	54
4.6.3	Miniature Tangible Cube の IMU に基づく回転状態認識	56
4.6.4	Miniature Tangible Cube の近接センサに基づく接触状態認識	58
4.6.5	Miniature Tangible Cube のセンサデータに基づくタンジブルインタフェース状態認識	59
4.7	タンジブルインタフェースの ObjectStateNet を用いた動作計画	59
4.7.1	ObjectStateNet の定義	60
4.7.2	ObjectStateNet におけるエッジとなる動作集合の設定	61
4.7.3	ObjectStateNet を用いた動作計画	63
4.8	タンジブルインタフェースによる動作順序の教示学習と自律実行	64
4.8.1	タンジブルインタフェースによる動作順序の教示学習	65
4.8.2	教示学習された動作順序による作業の自律実行	66
4.9	評価実験	67
4.9.1	Miniature Tangible Cube の状態認識実験	67

4.9.2	Miniature Tangible Cube による単一物体の操作作業の操縦実行実験 . . .	68
4.9.3	Miniature Tangible Cube による複数物体の操作作業の操縦実行実験 . . .	72
4.9.4	Miniature Tangible Cube による複雑形状物体の操作作業の操縦実行実験 . . .	81
4.9.5	Miniature Tangible Cube による把持動作を伴う物体操作作業の操縦実行実験	83
4.9.6	Miniature Tangible Cube による教示学習に基づく箱操作作業の自律実行実験	92
4.10	議論	99
4.10.1	対象物指向型操縦の全身物体操作動作への拡張	99
4.10.2	より器用な物体操作動作への拡張	99
4.10.3	形状変化可能なタンジブルインタフェース	99
4.11	おわりに	100
第 5 章	実世界適応による動作戦略の自律学習	101
5.1	はじめに	101
5.2	物体操作作業における動作戦略	102
5.3	動作戦略の実世界試行に基づく実世界適応	102
5.4	CNN 作業失敗予測器を用いた学習型動作戦略	103
5.4.1	倉庫環境における棚への収納作業	103
5.4.2	CNN 作業失敗予測器	106
5.4.3	CNN 作業失敗予測器の学習	106
5.4.4	CNN 作業失敗予測器を用いた動作戦略	109
5.5	CNN 把持点予測器を用いた学習型動作戦略	109
5.5.1	倉庫環境におけるピッキング作業	110
5.5.2	CNN 把持点予測器	111
5.5.3	シミュレーションを用いた CNN 把持点予測器の事前学習	116
5.5.4	CNN 把持点予測器を用いた動作戦略	129
5.6	CNN 把持点予測器の実世界適応	135
5.6.1	実世界把持経験に基づく再学習による実世界適応	135
5.6.2	双腕把持作業における実世界把持経験収集システム	136
5.6.3	CNN 把持点予測器を用いた実世界把持経験収集とデータセット生成	137
5.6.4	実世界把持経験に基づく CNN 把持点予測器の再学習による実世界適応	141
5.7	評価実験	145
5.7.1	作業失敗予測に基づく棚への双腕収納作業実験	145
5.7.2	把持点予測に基づく倉庫環境での双腕ピッキング作業実験	147
5.8	議論	167
5.8.1	再学習による新たな把持点の獲得	167
5.8.2	実世界適応による動作戦略の改善	168

vi 目次

5.8.3	実世界適応における学習サンプルの偏りとその影響	168
5.9	おわりに	170
第 6 章	物体操作作業の操縦自律間切替実行システム	171
6.1	はじめに	171
6.2	物体操作作業の実行システム	172
6.3	物体操作作業の操縦自律間切替実行システム	172
6.4	ユーザの入力に応じた操縦自律間切替実行	173
6.4.1	操縦実行から自律実行への切替	173
6.4.2	自律実行から操縦実行への切替	174
6.4.3	ユーザの音声入力に応じた操縦自律間切替実行	175
6.5	物体操作作業の作業実行管理システム	179
6.5.1	物体操作作業の作業管理システム	180
6.5.2	物体操作作業の作業実行システム	181
6.6	物体操作作業のための視聴覚統合遠隔操縦システム	183
6.6.1	VR デバイスを用いた手先指向型遠隔操縦システム	184
6.6.2	ロボット 状態・周囲環境情報の視聴覚・振動情報提示システム	185
6.7	物体操作作業における実世界経験収集システム	186
6.7.1	作業経験収集システム	187
6.7.2	日常経験収集システム	188
6.8	物体操作作業のための深層学習モデルによる認識	189
6.8.1	物体操作作業のための物体・環境認識	191
6.8.2	物体操作作業のための人認識	191
6.9	評価実験	194
6.9.1	冷蔵庫物品取り出し作業操縦実行実験	194
6.9.2	冷蔵庫物品取り出し作業自律実行実験	194
6.9.3	毎日キッチン巡回作業自律実行実験	194
6.10	おわりに	196
第 7 章	物体操作作業の操縦自律間横断学習ロボットシステムの評価	198
7.1	はじめに	198
7.2	箱の回転積み重ね作業の操縦自律間切替実行における作業手順の教示学習実験	199
7.2.1	箱の回転積み重ね作業で検証するユーザ入力に応じた操縦自律間切替 実行	200
7.2.2	箱の積み重ね作業の作業手順の教示学習実験	201
7.2.3	箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業 の作業手順の教示学習実験	203
7.2.4	作業手順の教示学習結果に基づく箱の積み重ね作業の自律実行実験	212

7.2.5	作業手順の教示学習結果に基づく箱の Z 軸 90° 回転積み重ね作業の自律実行実験	213
7.2.6	箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験	218
7.2.7	作業手順の教示学習結果に基づく箱の Z 軸 180° 回転積み重ね作業の自律実行実験	226
7.3	自律学習型把持動作を伴う物体ダンボール箱収納作業の操縦自律間切替実行における作業手順の教示学習実験	231
7.3.1	物体ダンボール箱収納作業で検証するユーザ入力に応じた操縦自律間切替実行	233
7.3.2	ダンボール積み重ね作業の作業手順の教示学習実験	233
7.3.3	作業手順の教示学習結果に基づくダンボール積み重ね作業の自律実行実験	235
7.3.4	ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験	240
7.3.5	作業手順の教示学習結果に基づく自律学習型把持動作を伴う物体ダンボール箱収納作業の自律実行実験	241
7.4	おわりに	245
第 8 章	結論	250
8.1	本研究の総括	250
8.2	本研究の成果	251
8.3	本研究の結論	252
8.4	今後の展開	253
	発表文献と研究活動	255
	参考文献	259
付録 A	Miniature Tangible Cube の実装	275
A.1	Miniature Tangible Cube の基板実装	275
付録 B	Miniature Tangible Cube による物体操作作業の操縦実行実験	284
B.1	Miniature Tangible Cube によるぬいぐるみ詰め寄せ実験	284
B.2	Miniature Tangible Cube によるぬいぐるみ積み重ね実験	284
B.3	Miniature Tangible Cube による袋入り靴下の箱収納作業実験	285
付録 C	コンビニ物品操作操縦実験	291
C.1	コンビニ物品品出し操縦実験	291
C.2	コンビニ物品陳列操縦実験	291

viii 目次

C.3	コンビニペットボトル取り出し操縦実験	292
C.4	コンビニゴミ分別操縦実験	293
付録 D	日本科学未来館・東京大学間遠隔物体ピッキング操縦実験	296
D.1	実験参加者・実験手順	296
D.2	遠隔物体取り出し実験結果	298
D.3	視聴覚統合型遠隔操縦システムのユーザビリティ評価	298
D.3.1	遠隔操縦物体操作システムのユーザビリティ評価アンケート	298
D.3.2	遠隔操縦物体操作システムのユーザビリティ評価軸の設計	299
D.3.3	遠隔操縦物体操作システムのユーザビリティ評価アンケートの設計	300
D.3.4	ユーザビリティ評価アンケート結果	302
D.3.5	ユーザビリティ評価アンケートの議論	303

第 1 章

序論

1.1 本研究の背景と目的

ロボット工学の発展により産業用ロボットに代表されるように物体操作作業を行うロボットシステムは多く社会に導入されている [1-3]. これらのロボットシステムは単腕ロボットかつエンドエフェクタを用いた物体操作作業であり, 人と同じように両腕や身体全体を使うことで更に多様な物体操作作業の実現ができるロボットシステム [4-8] が社会に導入されることが期待されている. 人は箱の蓋をあげ, 右・左から抱えあげ, 箱の中身を出すなどといった複数の接触がある動作を順序通りに行う必要があるような物体操作作業を容易に行なえるため, ロボットにも同様な能力を期待されており, このような機能を有したロボットシステムの研究が自律実行 [9, 10] と操縦実行 [11, 12] の 2 つの分野でこれまで盛んに行われているが, このような手順があり接触の多い物体操作作業では手順による作業の複雑さと接触の多い動作の複雑さから作業の対応性・安定性が低く, 失敗や新たな作業などへの対応のための作業手順の変更が課題となっている. そこで本研究では手順があり接触の多い物体操作作業に取り組み, 自律と操縦を切り替えながら作業の知識を横断的に学習することで作業を継続しながら作業手順の変更を学習するようなロボットシステムを構築することを目指す. 目標とするロボットシステムを構成するために, 本研究では作業に関する知識の中でも作業手順に着目し, 作業実行を継続しながら自律と操縦を切り替えることで作業の自律実行中に操縦実行に切り替えて人からその場で作業手順の変更を教示学習でき, 操縦実行にて変更された作業手順を自律実行できるような操縦自律間横断学習を実現する.

それでは物体操作作業の操縦・自律を切り替えながら, 作業に関する知識を操縦自律間で横断的に学習するロボットシステムはどのように構成することができるのだろうか? ここでロボットシステムにおける作業の実行形態の観点から物体操作作業をみると, 物体操作作業を行うロボットシステムは操縦実行分野では遠隔操縦 [12], 自律実行分野では自律学習ピッキングシステム [13] など, 作業の実行形態に応じて多種多様なシステムが提案・構成されている. これらのシステムは人が遠隔地からロボットを操縦することや, ロボットが自律的にピッキング作業を行うことといった作業に応じて構成されている. しかし操縦であっても自律であっても, ロボットが物体を操作することには変わりなく, ロボットシステムとして作業に

2 第1章 序論

関する知識を共有することができ、かつ操縦と自律間で作業に関する知識を共有することで作業の学習を効率的に学習することができる。そこで本研究では対象物の状態に着目し物体操作作業を対象物の状態に基づいて動作順序と動作に分割することで、操縦実行において作業手順を表す動作順序を教示学習し、自律実行において動作を制御する動作戦略を自律学習し、これらを操縦自律間で共有することで操縦自律間横断学習を実現する。よって本研究では自律と操縦の2つの作業実行形態を実行しながら切替可能、かつ動作順序と動作戦略を横断学習可能なロボットシステムを構築し、このロボットシステムを用いて複雑な物体操作作業を実行することを目的とし、操縦・自律の2つの作業実行形態間を切り替えながら、作業を横断的かつ効率的に学習するロボットシステムの構成法を明らかにする。

1.2 本研究の構成

本論文は全8章からなる。本論文の構成を図1.1に示す。以下に各章の概要を述べる。

第1章「序論」では、本研究の背景と目的について述べた。

第2章「物体操作作業の操縦自律間横断学習ロボットシステム」では、物体操作作業とその実行システムについて作業実行形態の観点から議論し、本研究で取り組む操縦・自律の2つの作業実行形態を横断的に学習する操縦自律間横断学習と、これを実現する操縦自律間横断学習ロボットシステムの構成法について述べる。

第3章「操縦自律間横断学習ロボットシステムのハードウェア・ソフトウェア構成」では、本研究で用いるロボットシステムのハードウェア・ソフトウェア構成について説明する。

第4章「タンジブルインタフェースによる作業手順の教示学習」では、物体操作作業の操縦実行において、対象物の接触・回転状態を表現するためにタンジブルインタフェースを用いることを提案し、このインタフェースの実装手法と動作計画手法、そして操縦実行に基づく動作順序の教示学習手法について述べる。

第5章「実世界適応による動作戦略の自律学習」では、物体操作作業の自律実行において、自律学習するための学習型動作戦略の構成手法とそのシミュレーションと実世界での実世界試行を用いた実世界適応手法について述べる。

第6章「物体操作作業の操縦自律間切替実行システム」では、操縦自律間横断学習ロボットシステムを実現するための操縦自律間切替実行システムの構成手法について述べる。

第7章「物体操作作業の操縦自律間横断学習ロボットシステムの評価」では、本研究で構築する操縦自律間横断学習ロボットシステムの有効性をロボットによる実世界での物体操作作業実験で評価する。

第8章「結論」では、本研究の結論を述べる。

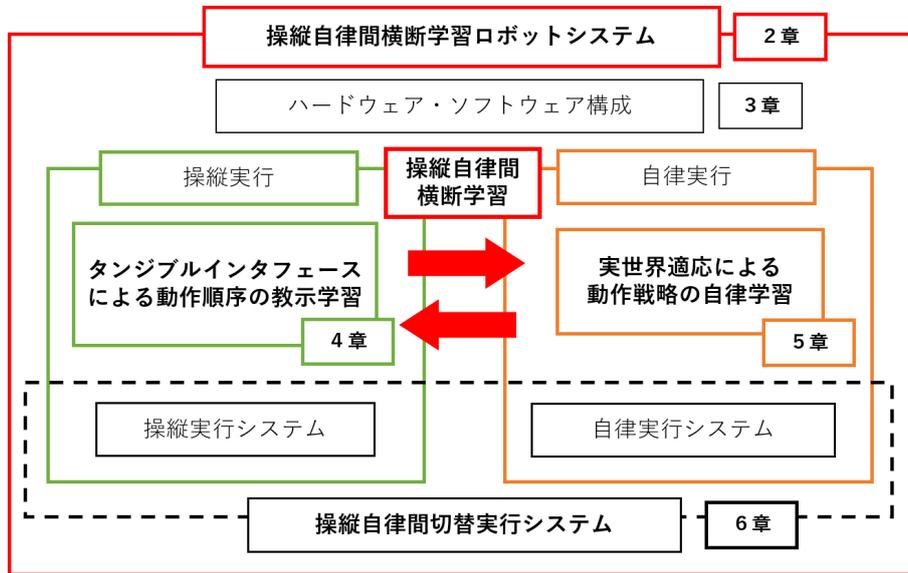


図 1.1: 操縦自律間横断学習ロボットシステムと本論文の構成

第 2 章

物体操作作業の操縦自律間横断学習 ロボットシステム

2.1 はじめに

本章では本研究で取り組む物体操作の操縦自律間横断学習ロボットシステムの構成について説明する。まず本研究で取り組む物体操作作業と操縦・自律という 2 つの作業実行形態を横断的に学習することの利点について説明し、本研究で目指す物体操作作業の操縦・自律を切り替えながら作業を学習する操縦自律間横断学習と、それを実現するための操縦自律間横断学習ロボットシステムの構成について説明する。

2.2 ロボットによる物体操作作業

本節では本研究で取り組む物体操作作業について説明する。まず物体操作作業について定義し、それを接触部位と接触点に基づいて分類し、本研究で取り組む複雑な物体操作作業について説明する。

2.2.1 物体操作作業の定義

まず物体操作作業について定義する。物体操作作業とは、ロボットが腕や身体を用いて物理的接触を行うことで、対象となる物体の状態を目標の状態に物理的に変更する作業と本研究では定義する。このとき作業には対象物について目標となる状態があり、対象物を現在の状態から目標の状態までロボットを用いて遷移させることが物体操作作業であるといえ、例としては把持や箱の回転・抱え上げなどが挙げられる。

2.2.2 物体操作作業の接触部位と接触点に基づく分類

次に物体操作作業の接触部位と接触点に基づく分類について説明する。まずロボットのハードウェア構成について分類する。ロボットのハードウェア構成は四肢の数で分類することがで

き、腕が1本しかないロボットを単腕ロボット [14–16]、腕が2本あるロボットを双腕ロボット [17–21]、人と同じように腕が2本・脚が2本あるロボットをヒューマノイド [22–24] と区別する。このとき物体操作作業ではハードウェア構成に基づいて分類することができ、本研究では物体操作作業を行う接触部位と接触点の数に着目して以下の3つに分類する。

- 単腕物体操作作業
- 双腕物体操作作業
- 全身物体操作作業

1つ目は単腕のエンドエフェクタを用いて行える物体操作作業を単腕物体操作作業と呼び、物体の把持やそれに伴うピッキング作業が例として挙げられる [25, 26]。2つ目は双腕のエンドエフェクタを用いて行える物体操作作業を双腕物体操作作業と呼び、双腕を用いることで持ち替えや抱え上げなどによって、単腕物体操作よりもさらに器用な物体操作を行うことができる [9, 27, 28]。3つ目はロボット全身の部位を用いて行える接触点が3点以上の物体操作作業を全身物体操作作業と呼び、等身大ロボットが必要となるような大きな物体を取り扱うことができる [7, 8]。このように物体操作作業は接触部位と接触点に分類でき、この分類からロボットがどのようなハードウェア構成であるかが重要となり、同じ作業でもあってもどのようなハードウェア構成を有しているかに応じて実行可能性が決まることがわかる。例えば単腕ロボットはエンドエフェクタを用いた単腕物体操作と腕全体を用いた全身接触動作を行うことができ、双腕ロボット、ヒューマノイドは単腕・双腕・全身作業の全てを行うことができる。本研究では上記の3つの単腕・双腕・全身物体操作作業の全てについて双腕ロボットを用いて取り組む。

2.2.3 本研究で取り組む複雑な物体操作作業

本項では本研究で取り組む複雑な物体操作作業について説明する。本研究で取り組む物体操作作業の複雑性は以下の2つである。

- 手順がある
- 多くの物理接触がある

手順のある物体操作作業とは、物体操作作業において作業を実行するために複数の動作を順序に沿って実行する必要がある作業のことであり、ピッキング作業 [25, 26] とは異なり複数の動作を組み合わせることや複数の物体を取り扱うことが求められる。例えばものを掴んで箱に収納して片付けるなどの作業が挙げられ、一般に人間は日常生活や工場・倉庫でもこのような手順のある作業を行っており、ロボットも同様の物体操作作業を行うことが期待されている。これまで多くの研究 [9, 10] が行われてきているが、作業の複雑さからロボットによる自律的な作業実行では作業の対応性が低く、失敗や新たな作業に対応するための作業手順の変更が課題となっている。

多くの物理接触がある物体操作作業とは、物体に対してエンドエフェクタのみではなく全身

や双腕を用いて物体操作を行う作業のことであり、ピッキング作業 [25, 26] とは異なり把持以外の動作を行うことでより多くの物体に様々な作業を行うことが求められる。例えば箱を両手で抱えて回転させて持ち上げるなどの作業が挙げられ、こちらも同様に一般に人間は日常生活や工場・倉庫でもこのような手順のある作業を行っており、ロボットも同様の物体操作作業を行うことが期待されている。これまで多くの研究 [7, 8] が行われてきているが、動作の複雑さからロボットによる自律的な動作実行では動作の安定性が低く、失敗や新たな作業に対応するための作業手順の変更が課題となっている。

以上より、本研究では手順があり接触の多い物体操作作業に取り組み、失敗や新たな作業手順に対応するための作業手順の変更という課題に対して、作業実行を継続しながら自律と操縦を切り替えることで作業を学習するようなロボットシステムの実現に取り組む。このように操縦・自律の間を切り替えながら横断的に作業を学習することで、自律実行で複雑な物体操作作業を行っている際に失敗が発生した場合や新たな作業に変更された際にも操縦実行に切り替えることで作業を継続しながら対応でき、かつ操縦実行で行った作業を学習する、ような知能的なロボットシステムが実現できると本研究では考える。

2.3 物体操作作業における作業実行形態

本項では物体操作作業について作業実行形態の視点から分類しながら、物体操作の操縦・自律の切り替えと横断的な学習について説明する。まず物体操作作業における作業実行形態とは何かについて説明した後、作業実行形態に基づいて物体操作作業を分類し、物体操作作業の操縦自律の切替について説明する。

2.3.1 物体操作作業における作業実行形態の分類

まず物体操作作業における作業実行形態について説明する。作業実行形態とはロボットが作業を実行する際の方式であり、物体操作作業においては図 2.1 に示すように以下の3つの作業実行形態を定義することができる。

- 操縦: Maneuvering
- 自律: Autonomy
- 協調: Collaboration

操縦 (Maneuvering) とは、人がロボットを作業中に指示を送りながら作業を行うことであり、認識や計画はロボットが補助をする場合 [29, 30] もあるが、最終的な判断は人が行う方式である。自律 (Autonomy) とは、ロボットが自ら判断しながら作業を行うことである。協調 (Collaboration) とは、ロボットと人が同じ作業目的のもと作業を行うことであり、人と協力しながら、ロボットは自らの判断のもとで作業を行うことである。本研究では操縦・自律・協調の中でも操縦と自律に着目する。物体操作作業の操縦実行とは、人がロボットを操縦して作業を実行することであり、これまで宇宙などの人が立ち入れない場所にある遠隔ロボットによる

物体操作作業に関する研究が盛んに行われてきた [11, 12, 31]. 物体操作の自律実行とは, プログラムがロボットを管理し作業を実行することであり, これまで組み立てや倉庫・工場作業などの産業における物体操作作業の効率化に関する研究が盛んに行われてきた [9, 10, 26]. 本研究ではこの中でも特に操縦・自律の2つの作業実行形態に着目し, この2つの作業実行形態を切り替えながら実行し, 作業に関する知識を横断的に共有・学習することを目指す.

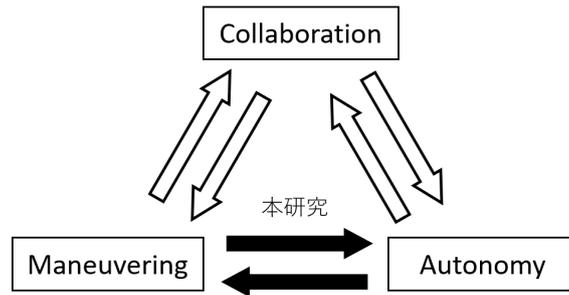


図 2.1: 操縦・自律・協調の作業実行形態の遷移とその効果: 物体操作作業は操縦・自律・協調の3つの作業実行形態に分類することができる. 本研究では操縦・自律の2つの作業実行形態に着目し, 操縦・自律の作業実行形態が切り替えながら実行でき, かつ操縦・自律の間で作業に関する知識を横断的に学習することができる知能的なロボットシステムを構成することを目指す.

2.3.2 物体操作作業の操縦と自律

物体操作作業には 2.3.1 項に示すように操縦・自律・協調の3つの作業実行形態があるが, 本研究ではこの中でも特に操縦・自律の2つに着目する. 2.2 節でも説明したように本研究で取り組む手順があり接触の多い物体操作作業の自律実行では作業手順の変更が課題となり, 本研究ではこの課題を以下の2つの学習手法で解決する.

- 自律における作業の動作の自律学習
- 自律と操縦を切り替えながらの作業手順の教示学習

まず1つめの自律における作業の動作の自律学習は, 作業手順の変更に対して新たな作業手順で必要とされる動作をロボットがシミュレーションや実世界で作業を試行しながら, 成功・失敗などの試行結果に基づいて動作を自律的に学習することである. この作業の動作の自律学習は, 新たな作業において必要となる新たな接触の多い複雑な物体操作動作をロボットが自律的に獲得することができるため, 作業手順の変更には効果的である. 2つめの自律と操縦を切り替えながらの作業手順の教示学習については, 手順があり接触の多い物体操作作業においてロボットに対してどのような手順で対象物に接触するかを学習する必要であり, この作業手順の学習はロボットが自律的に学習することは複数の対象物を扱うような作業では探索範囲が組み合わせ的に増大してしまい困難となる. しかし人は作業の状況に対して柔軟に対応できるた

8 第2章 物体操作作業の操縦自律間横断学習ロボットシステム

め自律から操縦に切り替えることで人がロボットに対して作業手順を教示することは効果的である。このとき人とロボットが協力しながら作業を行う協調では教示することは困難であるため、本研究では操縦に着目し、自律から操縦に切り替えてロボットを直接操縦者が動かすことで複雑な作業の教示学習を行う。この自律と操縦を切り替えながらの作業手順の教示学習は、新たな作業において手順を知らないため自律的に作業を行えないロボットに対して人はその場で操縦に切り替えて作業手順を教示することができるため、作業手順の変更に効果的である。また本研究では対象物の接触状態を表現する操縦実行手法を提案し、この操縦実行手法を用いることで手順があり接触の多い物体操作作業の教示学習を実現する。

以上より本研究では物体操作作業の操縦・自律の2つの実行形態に着目し、自律においては作業の動作の自律学習を行い、操縦においては作業手順の教示学習を行うことで、この2つの実行形態を切り替えながら学習することができるロボットシステムの構築を目指す。

2.3.3 本研究で取り組む操縦自律間で学習するロボットシステム

では操縦・自律など複数の作業実行形態間で学習するロボットシステムにはどのような機能が必要となるだろうか？まず1つめの機能として、ある作業において作業実行形態を切り替えて実行した際に、複数の作業実行形態間で学習した作業に関する知識を共有しながら効率的に学習することが求められる。例えば作業の操縦実行において教示した作業手順を自律実行に共有することで同じ作業を再現することや、作業の自律実行の途中で操縦に切り替えて作業手順を教示することで自律実行の際に使われていた作業の知識を再利用して教示学習することが挙げられる。これは本研究で目指す作業実行形態間での学習を実現するために必要な機能である。そして2つめの機能として、ある作業を行っている最中に作業を中断することなく作業実行形態を切り替えることが求められる。例えば作業を自律実行している最中に作業手順の変更を実現するために操縦実行に切り替えることが挙げられる。このような作業手順の変更は本研究で取り組む手順があり接触の多い物体操作作業では必要であり、本研究のシステムには必要となる機能であると言える。

つまり、本研究で取り組む操縦自律間で学習するロボットシステムはかつ作業実行形態間で作業の知識を横断的に学習するべきであり、作業実行形態を切り替えながら継続的に作業を実行できるべきである。よって本システムに必要な機能は以下の2つとなる。

- 操縦自律間横断学習機能: 操縦自律間で作業に関する知識を横断的かつ効率的に学習できること
- 操縦自律間切替実行機能: 操縦と自律を切り替えながら作業を実行できること

操縦自律間横断学習機能とは、ロボットシステムが操縦・自律の2つの作業実行形態を遷移しながら物体操作作業に関する知識を横断的かつ効率的に学習できることである。この操縦・自律の作業実行形態を切り替えながら作業に関する知識を共有し学習することを本研究では操縦自律間横断学習と呼び、これの実現に取り組む。操縦自律間横断学習では操縦実行における教示学習、自律実行における自律学習と操縦・自律についてそれぞれ異なる手法を用いて作業

を学習するが、学習された作業の知識を操縦自律間で共有することで過去の実行・学習結果を再利用した横断的かつ効率的な作業の学習を実現する。なお本研究では作業に関する知識の中でも作業手順を表す動作順序と動作を制御する動作戦略に着目し、操縦実行における教示学習において動作順序を、自律実行における自律学習において動作戦略を学習することで操縦自律間横断学習を実現する。操縦自律間切替実行機能とは、ロボットシステムが操縦・自律型の物体操作作業が実行可能であることである。操縦自律間切替実行機能については本研究ではユーザの入力に応じて操縦自律間切替実行を行うことができる作業実行システムを構成することで実現する。

したがって本研究では、以上の2つの機能を有して作業実行形態間の横断学習と切替実行を行えるロボットシステムの構成を行うことを目的とする。作業実行形態間の切替実行と横断学習が可能なロボットシステムは作業に関する知識を横断的に学習することができるのであれば、異なる作業実行形態の間で相互に影響をあたえながら学習していく知能的なロボットシステムが構成できると考え、本研究では物体操作作業を行うロボットシステムにおいて作業実行形態間の横断学習機能と切替実行機能を実現することを目指す。

2.4 物体操作作業の操縦自律間横断学習ロボットシステム

2.2節では本研究で取り扱う手順があり接触の多い複雑な物体操作作業の必要性と課題について説明し、2.3節では本研究で取り扱う操縦・自律とその間を切り替えながら作業を横断的に学習する操縦自律間横断学習の利点について説明した。ここで2.3節より本研究で目指す複雑な物体操作作業において操縦自律間横断学習ができる可能であるロボットシステムは、操縦自律間横断学習機能と操縦自律間切替実行機能の2つの機能を備えているべきであるといえる。これらの2つの機能を有したロボットシステムを操縦自律間横断学習ロボットシステムと呼び、本研究ではこのロボットシステムの構成に取り組む。本研究で構成する操縦自律間横断学習ロボットシステムは作業を対象物の状態の変化に応じて動作と動作順序に分割し、図2.2に示すように、作業に関する知識として作業手順を表す動作順序と動作を制御する動作戦略をそれぞれ操縦実行における教示学習と自律実行における自律学習を用いて学習することで操縦自律間横断学習を実現する。

したがって本研究の目的は、複雑な物体操作における操縦自律間横断学習ロボットシステムの構成であり、本節ではこのシステムを実現するための構成について説明していく。本節ではまず対象物の状態に基づいた作業の分割とその分割に基づく横断学習手法について説明し、操縦自律間横断学習機能を構成するための操縦実行における教示学習手法と自律実行における自律学習手法を説明し、操縦自律間切替実行機能を構成するための操縦自律間切替実行手法について説明し、最後に本研究で構成するシステムの全体像について説明する。

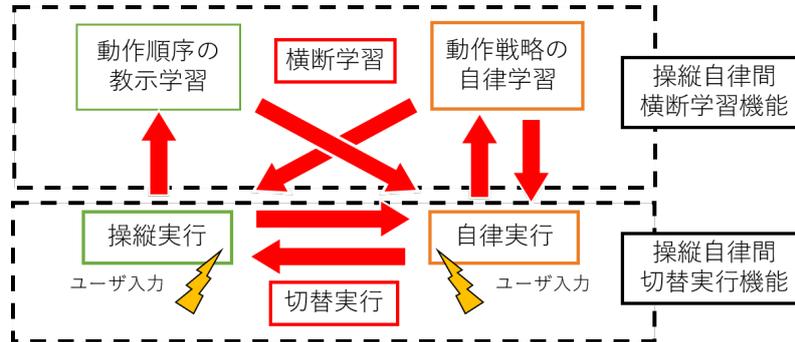


図 2.2: 本研究で構成する操縦自律間横断学習ロボットシステムのコンセプト：操縦自律間横断学習ロボットシステムは物体操作作業に必要な操縦自律間横断学習機能と操縦自律間切替実行機能を有し，この2つの機能を有することで，操縦実行と自律実行を切り替えながら作業に関する知識を学習することができるロボットシステムを構成することができる．本研究では作業手順を表す動作順序と動作を制御する動作戦略に着目し，動作順序を操縦実行において教示学習し，動作戦略を自律実行において自律学習し，ユーザ入力に応じて操縦と自律を切り替えながら実行することで操縦自律間横断学習を実現する．

2.4.1 対象物の状態に基づいた操縦自律間横断学習手法

本研究では作業に関する知識として作業手順を表現する動作順序と作業の動作を制御する動作戦略をそれぞれ操縦実行において教示学習，自律実行において自律学習することで，作業の操縦自律間横断学習を実現する．このとき作業を操縦自律間で横断学習するために作業を複数の動作とその連なりである動作順序に分割する必要がある，本研究ではこの作業の分割を対象物の接触・回転状態に基づいて行う．本項ではまずこの対象物の接触・回転状態に基づいた作業の分割について説明し，そのあとに作業を構成する動作と動作戦略をどのように作業自律間で横断学習するかについて説明する．

2.4.1.1 対象物の接触・回転状態に基づいた作業の分割とその利点

まず本研究における作業の分割について説明する．本研究では対象物の接触・回転状態に着目して作業の分割を行い，図 2.3 に示すように作業を動作とその連なりである動作順序に分割し，動作の動作戦略を自律実行において自律学習し，動作順序を操縦実行において教示学習することで作業の操縦自律間横断学習を実現する．具体的には，対象物の接触・回転状態が変化しその状態を保てる場合に，以前の接触・回転状態から今の接触・回転状態へ遷移するために行ったロボットの操作を1つの動作とし，作業をこのような複数の動作の順序とする．対象物の接触・回転状態に基づいて作業を分割することで，動作手順を対象物の接触・回転状態の遷移として表現することができ，この手法には以下の2つの利点がある．

- 新たな作業の教示学習への再利用性

- 接触の多い複雑な作業の教示可能性

以下ではこの2つの利点について順に説明する。

まず1つめの利点である新たな作業への再利用性について説明する。2.2節で説明したように物体操作作業とは対象物の状態を目的の状態に変更する作業であるため、作業手順を求めることは現在の対象物の状態から作業の目標の状態への経路を探索する問題であると言い換えることができる。このとき本研究では作業を対象物の接触・回転状態に基づいて分割し、動作順序を対象物の接触・回転状態の遷移として教示学習するため、新たな作業であってもこれまでに学習された動作順序から追加・分岐する形で教示学習することができる。例えば先に箱を90°回転する作業の動作順序を教示したあと箱を更に180°回転する作業の動作順序を新たに教示学習する場合には、箱が90°回転している対象物の状態までは再利用可能であり、箱が90°回転している対象物の状態から動作順序を分岐して更に90°回転させるような動作順序を教示することで効率的な動作順序の教示学習が可能である。また新たな作業の目標状態が、これまでに学習された動作順序の途中が目標状態である可能性もあり、例えば先に箱を180°回転する作業の動作順序を教示したあと箱を90°回転する作業を行う場合には、箱を180°回転する作業の動作順序の途中で箱が90°回転している対象物の状態が存在しているため、新たな教示を行う必要なく箱を90°回転させる作業を実現することができる。この新たな作業の教示学習への再利用性は本研究で目指す操縦自律間横断学習の効率性に寄与しており、作業の自律実行の途中で操縦に切り替えて作業手順の教示学習する場合にはこれまで学習された動作順序に追加・分岐した形で動作順序を学習することができ、これによって過去の実行・学習結果を再利用した効率的な作業手順の教示学習を実現することができる。

2つめの利点である接触の多い複雑な作業の教示可能性について説明する。本研究では作業を対象物の接触・回転状態に基づいて分割し、動作順序をその対象物の接触・回転状態の遷移として表現するため、教示学習においてロボット・対象物間の接触状態のみではなく、対象物・対象物間の接触状態や対象物・環境間の接触状態を表現することが可能となる。例えば箱を異なる箱の上に積み重ねる作業の動作順序として、箱ともう1つの箱の間の積み重なりによる接触と箱と床の接触を表現することができ、このような対象物を中心とした作業の動作順序を教示学習することでロボットは作業手順について対象物・対象物間と対象物・環境間の状態の変化も含めて学習することができる。操縦実行ではロボット・対象物間の接触を表現するユーザインタフェース [32, 33] はこれまで研究されてきたが、操縦実行における教示学習のための対象物・対象物間や対象物・環境間の接触を表現するユーザインタフェースは提案されておらず、対象物・環境・ロボット間の接触を教示学習できるような操縦インタフェースは実現できていない。そこで本研究ではこのようなロボット・対象物間のみではなく対象物・対象物間や対象物・環境間の接触を考慮した操縦を対象物指向型操縦と呼び、この対象物指向型操縦において動作順序の教示学習を実現する。

以上より本研究では作業を対象物の状態に着目して動作と動作順序に分割し、動作順序については操縦実行において教示学習し、動作の動作戦略を自律実行において自律学習することに取り組む。また操縦実行における動作順序の教示学習の際に、対象物・対象物間や対象物・環

境間の接触状態の遷移を教示するためにこれらの接触を考慮した対象物指向型操縦とそれを用いた動作順序の教示学習に取り組む。

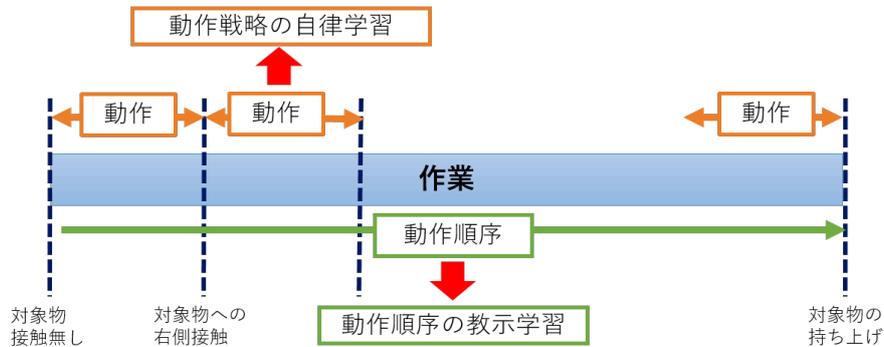


図 2.3: 対象物の状態に基づいた作業の分割: 本研究では対象物の状態に着目し作業を対象物の状態の変化に応じて動作と動作順序に分割する. このとき操縦実行においては動作順序を教示学習し, 自律実行においては動作の動作戦略を自律学習し, 各々で学習された動作順序と動作戦略を操縦自律間で共有することで操縦自律間横断学習を実現する.

2.4.1.2 物体操作作業の操縦自律間横断学習

本目では操縦自律間で横断的に作業を学習する操縦自律間横断学習について説明する. 本研究では物体操作作業において操縦自律間を切り替えながら横断的かつ効率的に作業を学習する操縦自律間横断学習を実現することを目的とし, これを実現した操縦自律間横断学習ロボットシステムと呼び, このシステムの構成を目指す. このとき本研究で構成する操縦自律間横断学習を図 2.4 に示す. 図 2.4 に示すように, ロボットは操縦実行において動作順序を教示学習し, 自律実行において動作戦略を自律学習する. そして操縦実行において教示学習された動作順序を自律実行に共有し, 自律実行において自律学習された動作戦略を操縦実行に共有し, 共有された動作順序と動作戦略を用いて作業を操縦自律切替実行することで, 操縦自律間横断学習を実現する.

次に操縦自律間横断学習の意義について説明する. 本研究で取り組む操縦自律間横断学習の利点は 2.3.2 項で説明した以下の利点が増えられる.

- 作業手順の変更の即応性: 人がその場で教示することで作業手順の変更ができること
- 作業手順の変更の効率性: 操縦と自律で効率的に作業手順の変更ができること

まず 1 つ目の作業手順の変更の即応性について説明する. 作業手順の変更の即応性とは, 失敗や新たな作業などの元々想定されている作業以外のことに対応するために作業手順をその場で変更することができることである. この利点は作業手順の変更によってロボットは作業手順を知らないため自律的に作業を行えない場合に対して, 人がその場でロボットを操縦して作業を行うことで作業手順を即応的に教示学習できることであり, 2.2.3 項で説明したように手順

があり接触の多い複雑な物体操作作業では失敗や新たな作業への対応のための作業手順の変更が発生しうるため、作業途中で人がその場で作業手順の変更ができることは有用である。つまり、操縦自律間横断学習を行うことで、作業手順の変更について操縦を用いて手順をその場で教示し、教示した手順をすぐに自律実行することができるため、この特徴は作業手順の変更に効果的である。

次に2つ目の作業手順の変更の効率性について説明する。これは2.4.1.1目で説明した対象物の接触・回転状態に応じた作業の分割によって再利用性の高く作業手順を表す動作順序を記述することができるため、作業の自律実行の途中で操縦に切り替えて作業手順の教示学習を行うことで過去の実行・学習結果を再利用して作業手順を効率的に学習することができ、これは作業手順の変更の教示学習においては有用である。つまり、操縦自律間横断学習を行うことで、作業手順の変更について自律実行の途中から操縦を用いてその場で教示することで、自律実行された動作順序から分岐する形で作業手順を教示学習することができ、過去の実行・学習結果を再利用して効率的に作業手順を教示学習ができるため、この特徴は作業手順の変更に効果的である。

以上より、操縦自律間横断学習の2つの利点として作業手順の変更の即応性と効率性が挙げられ、これらの利点は作業手順の変更に効果的である。よって本研究では操縦自律間横断学習を実現し、これを用いて手順があり接触の多い複雑な物体操作作業における作業手順の変更に取り組む。

2.4.1.3 操縦自律間横断学習における学習手法

ここまで本研究で取り組むと操縦自律間横断学習とその意義について説明した。本目ではこの操縦自律間横断学習を構成する学習手法について説明する。本研究では操縦自律間横断学習の実現のために、表2.1に示す2つの学習手法を組み合わせる。

まず1つめの作業手順の教示学習について説明する。本研究では作業手順を動作順序に表現し、操縦実行における動作順序の教示学習においてタンジブルインタフェース [34] を用いることで2.4.1.1目で説明したような対象物・対象物間と対象物・環境間の接触状態に基づいて手順があり接触が多い複雑な物体操作作業の動作順序を教示学習する手法を提案する。タンジブルインタフェース [34] とは、触ることで操作するインタフェースのことであり、本研究では対象物を模したタンジブルインタフェースを複数用いることで対象物・対象物間と対象物・環境間の接触を表現し、タンジブルインタフェースを用いて表現された対象物の状態に基づいて動作順序を教示することで複雑な物体操作作業の動作順序の教示学習を実現する。

次に2つめの作業の動作の自律学習について説明する。本研究では作業の動作を制御する動作戦略に着目し、自律実行における動作戦略の自律学習において実世界適応 [35] を用いることで、実世界にて実行可能な物体操作作業の動作の動作戦略を自律学習する手法を提案する。実世界適応とはシミュレーションにおける事前学習と実世界における動作試行に基づく再学習の2段階で動作戦略を自律学習する手法であり、本研究では把持動作について少ない実世界での動作試行でより安定的な把持動作を自律学習することで複雑な物体操作作業の動作戦略の自律学習を実現する以下ではこの2つの学習手法と操縦自律切替実行の本研究における特色・実

装について順に説明する。

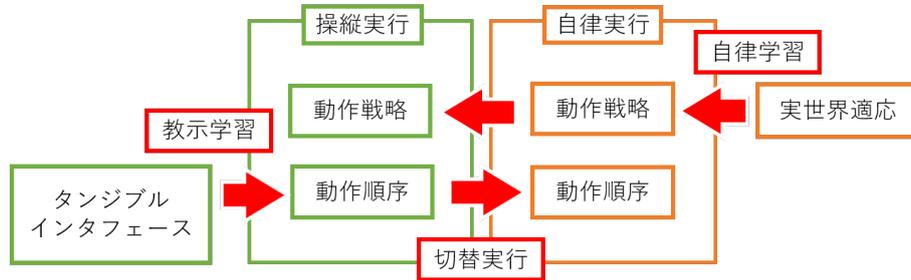


図 2.4: 操縦自律間横断学習: 本研究では操縦実行における教示学習にて動作順序を学習し, 自律実行における自律学習にて動作戦略を学習し, これらをそれぞれ操縦自律間で共有し, 共有された動作順序と動作戦略を用いて作業の操縦自律切替実行することで操縦自律間横断学習を実現する。

表 2.1: 操縦自律間横断学習ロボットシステムにおける操縦自律間横断学習と各学習手法

	作業手順の教示学習	動作戦略の自律学習
作業実行形態	操縦実行	自律実行
学習対象	作業手順を表す動作順序	動作を制御する動作戦略
手法	タンジブルインタフェースによる作業手順の教示学習 (Miniature Tangible Cube [37])	実世界適応による動作戦略の自律学習 (Self-supervised learning [35, 36])

2.4.2 操縦実行における作業手順の教示学習

本項では操縦実行における作業手順の教示学習について説明する。まず作業手順を表す動作順序の教示学習について定義した後に、本研究で取り組むタンジブルインタフェースによる動作順序の教示学習について説明する。

2.4.2.1 作業手順を表す動作順序の教示学習

本目では作業手順を表す動作順序の教示学習について説明する。教示学習とはロボットが人から作業や動作について教えてもらうことで学習する手法であり、近年では特に操縦データを活用した手法 [31, 38, 39] が取り組まれており、単一の動作については成果を示しているが、手順があり接触の多い複雑な物体操作作業を行うことは困難であり、操縦データから動作を分割認識し、その動作順序を作業手順として学習することが課題となっている。また対象物・対象物間や対象物・環境間などの接触状態の変化についても作業手順として認識する必要があるが、これはロボットの手先を動かすような教示学習 [31, 38, 39] では作業の分割点を認識する

ことが困難である。つまり、操縦における教示学習 [31, 38, 39] は、人の操縦データから動作や作業を効率的に学習することが目的であり、複雑な物体操作作業において操縦者の操縦実行で収集されたデータから作業を動作に分割し、作業手順としての動作順序を認識・学習することが課題となる。よって動作順序の教示学習は以下のように定義することができ、本研究ではこれの実現に取り組む。

- ある環境におかれたある対象物群に対して実行可能であり、かつ操縦者が実行を求めると動作列を操縦者の操縦実行から学習すること

2.4.2.2 動作順序の教示学習の構成要素

本目では本研究で着目する動作順序の教示学習の構成要素について説明する。本研究では動作順序の教示学習において、以下の2つの要素に着目する。

- 操縦ユーザインタフェース
- 動作順序の教示学習

操縦ユーザインタフェースとは、人がロボットを操縦するために用いる入力装置であり、操縦しやすいユーザインタフェースに関する研究は多くなされている [32, 33, 40–42]。しかしこれらの研究ではエンドエフェクタのみを用いた物体操作作業が主であり、双腕や全身を用いた物体操作動作が行われておらず、対象物・対象物間や対象物・環境間に接触が発生するような複雑な物体操作作業は実現できていない。また物体操作作業の教示学習において、人がロボットを操縦したデータを元に物体操作作業を学習する研究 [31, 38, 39] であるが、多くが単腕把持など簡単な動作のみが対象であり、動作順序があるような複雑な物体操作作業に適用されていない。そこで本研究では接触点の多く動作順序のあるような、より複雑な物体操作動作を行うための操縦ユーザインタフェースとそれを用いた動作順序の教示学習に取り組み、対象物・対象物間や対象物・環境間の接触を表現できるタンジブルインタフェース [34] を操縦インタフェースとして教示学習に用いることを提案する。

2.4.2.3 タンジブルインタフェースによる動作順序の教示学習

本目では動作順序の教示学習における本研究の提案手法について説明する。操縦による動作順序の教示学習では操縦データを収集する際に動作順序や異なる動作を明確に判別し学習することが重要であり、これを実現するために本研究では、タンジブルインタフェース [34] を用いて対象物の接触・回転状態を表現することを提案し、この手法を物体操作作業の操縦実行と操縦による動作順序の教示学習に用いる。対象物の接触・回転状態を表現するタンジブルインタフェースとは、対象物を中心とした形でロボットを操縦するユーザインタフェースを実装する手法であり、対象物・環境・ロボット間の接触状態を表現することができる。よってタンジブルインタフェースを用いることで複数の物体や環境が接触し合うような複雑な物体操作作業を操縦実行できると同時に、対象物の接触や回転といった状態の変化を明確な動作の分割点として認識しながら、動作順序を効率的に学習することができる。したがって本研究ではタンジブル

ルインタフェースによる操縦実行を行いながら、この操縦実行結果に基づいて動作順序を学習することに取り組む。具体的には操縦者や環境への接触・回転を赤外光の物体からの反射を計測することで近接・接触を検知する近接センサと地磁気も計測することができる IMU を用いて計測することでタンジブルインタフェースの接触・回転状態を認識し、その認識した状態を対象物で再現するようにロボットが動作を実行し、実行された動作を学習することでタンジブルインタフェースによる動作順序の教示学習を実現する。

2.4.3 自律実行における動作戦略の自律学習

本項では自律実行における動作戦略の自律学習について説明する。まず動作戦略の自律学習について定義した後に、本研究で取り組む実世界適応による動作戦略の自律学習について説明する。

2.4.3.1 動作戦略の自律学習

本目では動作戦略の自律学習について説明する。自律学習とは学習システムにおいて学習対象を実世界での経験に適応させるために学習を行う手法であり、実世界でロボットが自律的に動作を行い続ける手法 [13, 43–46] などが挙げられるが、いずれも単腕把持などの単純な動作にのみ適応されており、双腕動作などの接触の多い動作を行うことは困難である。また実世界において事前知識無しの状態からロボットが接触の多い複雑な動作を試行錯誤的に自律学習することが困難であり、少ない実世界の試行から効率的に複雑な動作を学習することが課題となる。つまり、自律学習は実世界の経験を効率的に収集し適応させることが目的であり、接触の多い複雑な動作を効率的に自律学習することが課題となる。よって動作戦略の自律学習は以下のように定義することができ、本研究ではこれの実現に取り組む。

- ある環境におかれたある対象物群に対して実行可能な動作の戦略をロボットが自律的に学習すること

2.4.3.2 動作戦略の自律学習の構成要素

本目では本研究で着目する動作戦略の自律学習の構成要素について説明する。本研究では動作戦略の自律学習において、以下の2つの要素に着目する。

- 学習型動作戦略
- 動作戦略の自律学習

学習型動作戦略とは学習手法で構成された状況に応じて最適な動作を選ぶための戦略であり、近年の深層学習技術の発展から盛んに研究されているが [13, 25, 43, 44, 47–49]、単腕把持など一部の限られた動作のみを取り扱っており、複雑な接触のある動作については適応されていない。動作戦略の自律学習とは、前述の学習型動作戦略をロボットが自律的に学習する手法であり、近年ではロボットシステムの安定化に伴って実機による試行やシミュレーションを用い

た手法 [13, 43, 44] が提案されているが、こちらでも単腕把持など一部の限られた動作のみを取り扱っており、複雑な接触のある動作については適応されていない。そこで本研究では、双腕などの接触の多い複雑な物体操作動作を取り扱える学習型動作戦略の自律学習に取り組み、シミュレーションと実世界試行を組み合わせた実世界適応によって学習型動作戦略の自律学習を行うことを提案する。

2.4.3.3 実世界適応による動作戦略の自律学習

本目では動作戦略の自律学習における本研究の提案手法について説明する。本研究では自律学習においてどのように動作を行うかを決定する動作戦略に着目し、この動作戦略のシミュレーションを用いた事前学習と実機による動作試行に基づく学習を用いて多様な動作への適応を実現し、このような事前学習と実世界での動作試行に基づく再学習の組み合わせる手法を提案し、この手法を実世界適応 [35] と呼ぶ。このとき動作戦略の実世界適応とは、動作戦略をシミュレーションによる事前学習を元に少ない実世界での動作試行の経験から学習する手法であり、この手法を用いることで双腕などの接触の多い物体操作動作を少ない実世界の経験から効率的に学習することができる。したがって本研究では実世界適応を用いて学習型動作戦略を自律学習することに取り組む。具体的には本研究では CNN 把持点予測器の自己教師あり学習 (Self-supervised learning) において、シミュレーションによる自動データセット生成による事前学習と、実世界での把持試行とその結果の自動アノテーションに基づく再学習を行うことで、CNN 把持点予測器の実世界適応を実現する。自己教師あり学習 (Self-supervised learning) とは教師あり学習の一種であり、教師あり学習で用いるデータセットのアノテーションをロボットが自ら行うことで自律学習を実現する手法である。

2.4.4 物体操作作業の操縦自律間切替実行

本項では物体操作作業の操縦自律間切替実行について説明する。2.3 節で説明した操縦・自律の物体操作作業に基づいて物体操作作業の操縦・自律の実行システムと、操縦と自律を切り替えながら実行するための物体操作作業の操縦自律間切替実行システムについて説明する。

2.4.4.1 物体操作作業の操縦・自律の実行

まず物体操作作業の操縦実行を行うための実行システムである操縦実行システム、自律実行を行うための実行システムである自律実行システムについて説明する。操縦実行システムとは、図 2.5a に示す操縦型の物体操作作業を行う実行システムであり、ロボットは認識・計画・実行ループを行うが、認識・計画については人が判断・決定を行うことで操縦を実現する。自律実行システムとは、図 2.5b に示す自律型の物体操作作業を行う実行システムであり、ロボットのみで認識・計画・行動ループを行う自律を実現する。本研究で取り組む操縦自律間切替実行はこの2つのシステムを切り替えながら作業を継続的に実行することができるシステムであるといえる。

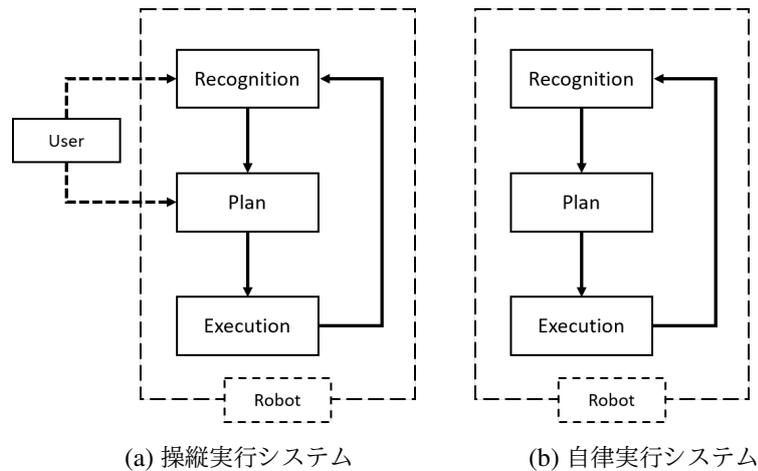


図 2.5: 物体操作作業の操縦・自律の実行システム: 本研究で取り組む操縦・自律の実行システムを示す。操縦実行システムは認識・計画について人が介入するのに対して、自律実行システムではロボットが自ら判断し認識・計画・行動を行う。

2.4.4.2 ユーザの入力に応じた操縦自律間切替実行システム

操縦自律間切替実行システムとは、操縦と自律を切り替えながら物体操作作業を継続的に行うことができる実行システムである。この操縦自律間切替実行システムは図 2.6 に示すように図 2.5a に示す操縦実行システムと図 2.5b に示す自律実行システムを切り替えながら作業を実行することができるシステムといえ、このときユーザの入力に応じて 2 つのシステムを切り替える必要がある。例えば自律実行で作業を行っていた際に失敗や対象物が想定している状況と異なっている場合には、ユーザがロボットのカメラ画像などの情報から判断してシステムに対して音声などで自律実行から操縦実行への切り替えを指示することで、作業手順を修正することで対応することが可能であり、このユーザ入力に基づく操縦・自律の切り替えは失敗の起こりやすい複雑な物体操作作業を行う場合には必要となる。本研究ではこの切り替えにはユーザからの音声入力を用い、各動作を実行した後にユーザの音声入力を受け付けることで、ユーザの音声入力に応じて操縦と自律を切り替えながら作業を継続的に実行する実行システムを構築することを目指す。

2.4.5 操縦自律間横断学習ロボットシステムの構成

ここまで操縦自律間横断学習ロボットシステムについて、操縦自律間横断学習と操縦自律間切替実行の 2 つに分けて説明してきた。本項ではこれらを統合した本研究の目的である操縦自律間横断学習ロボットシステムの構成について説明する。本研究で構成する操縦自律間横断学習ロボットシステムの構成について図 2.7 に示す。本研究は操縦自律間横断学習ロボットシステムの構成を目的とし、図 2.7 に示す各要素について図 1.1 に示す章構成のもとで説明する。

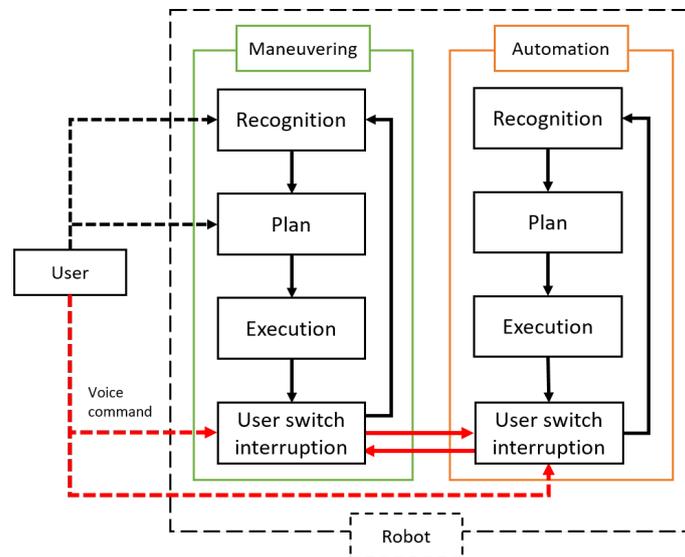


図 2.6: 物体操作作業の操縦自律間切替実行システム：本研究で取り組む操縦自律間切替実行システムを示す。操縦自律間切替実行システムではユーザの入力に応じて操縦実行システムと自律実行システムを切り替えながら作業を継続的に実行する。本研究では各動作を実行した後にユーザの音声入力を受け付けることで、ユーザの音声入力に応じた操縦自律間切替実行を実現する。

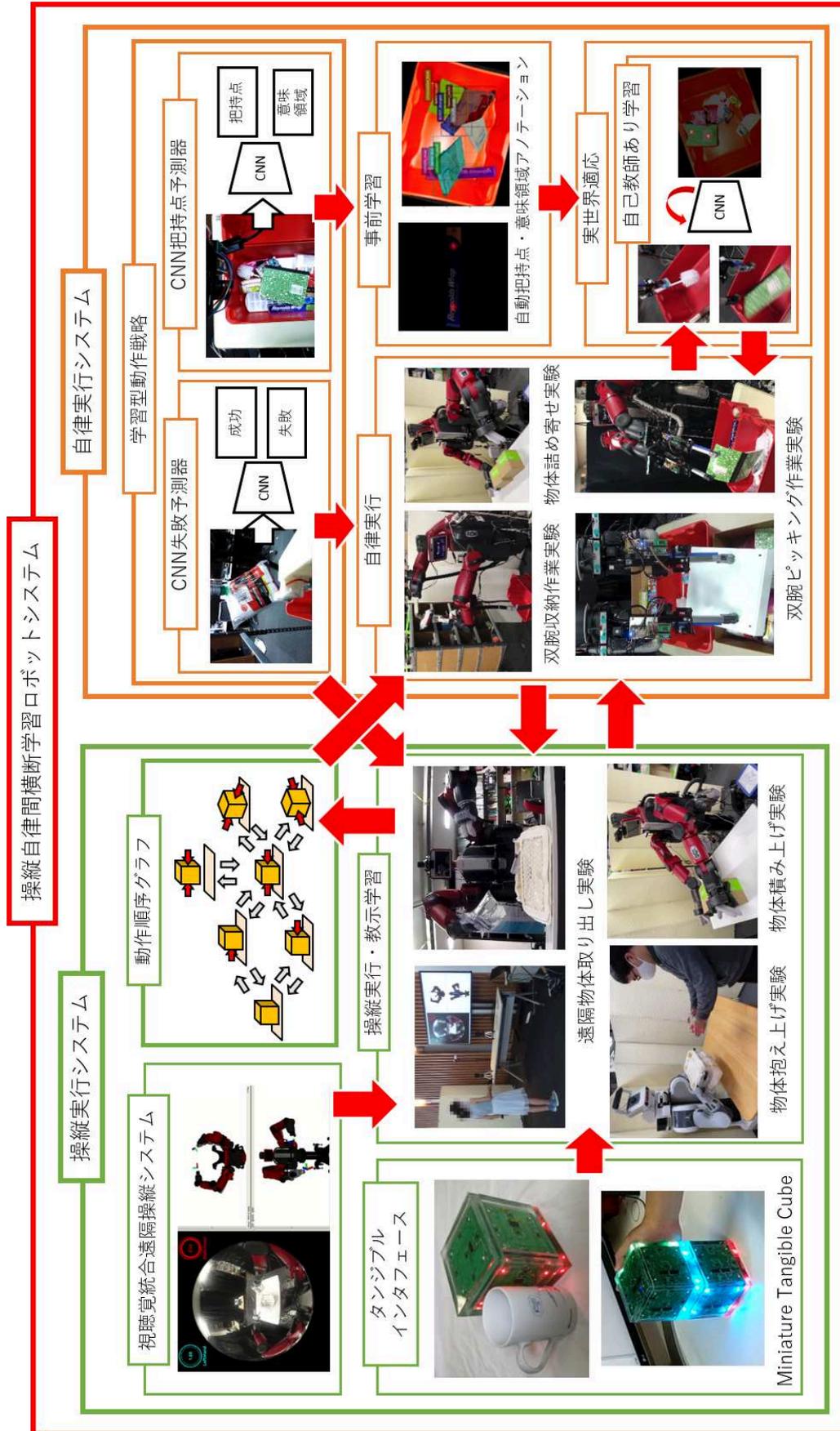


図 2.7: 操縦自律間横断学習ロボットシステム構成

2.5 おわりに

本章では本研究で取り組む手順があり接触の多い複雑な物体操作作業のための操縦自律間横断学習ロボットシステムの構成について説明し，そのシステムの操縦自律間横断学習と操縦自律間切替実行システムについて議論した．以下では図 1.1 に示すようにこのシステムを対象物の接触・回転状態を表現するタンジブルインタフェースによる操縦実行とそれに基づく作業手順の教示学習，学習型動作戦略を用いた自律実行とその実世界適応による動作戦略の自律学習，操縦自律間切替実行システムの各々について詳しく説明する．

第 3 章

操縦自律間横断学習ロボットシステムのハードウェア・ソフトウェア構成

3.1 はじめに

本章では、本研究で用いるロボットシステムについてハードウェアとソフトウェアという 2 つの観点から説明する。ハードウェア構成としては、双腕ロボット Baxter [17, 18] にカスタムグリッパを導入したロボット Baxter-JSK, 双腕移動ロボット PR2 [19] に計算機・センサなどを交換・追加したロボット PR2-JSK, 単腕移動ロボット Fetch [15] に計算機・センサなどを交換・追加したロボット Fetch-JSK を用い、ソフトウェア構成としては分散型ロボットシステム ROS [50] を用いて複数台の計算機とロボット実機をつなぐロボットシステムを構築する。

3.2 操縦自律間横断学習ロボットシステムのハードウェア構成

本節では本研究の実験で用いるロボットシステムのハードウェア構成について説明する。

3.2.1 双腕ロボット Baxter-JSK のハードウェア構成

本研究で用いる双腕ロボット Baxter-JSK には 2 種類のカスタムグリッパがあり、双腕ロボット Baxter [17, 18] の双腕のそれぞれの手先に、吸引多指グリッパ [51] を改造した吸引グリッパを導入したものを Baxter-JSK-APC, ソフトハンド [52] を導入したものを Baxter-JSK-Softhand と呼び、それぞれの全体像を図 3.1, 図 3.2 に示す。Baxter-JSK の基本となる Baxter は直列弾性アクチュエータ (Series Elastic Actuator) をグリッパを除く各関節に採用することで、ハードウェア的に衝撃に強く、環境や物体に馴染むロボットであり、接触点が多く拘束の多い物体操作作業や、環境との接触の可能性が高い遠隔操縦による物体操作作業に適している。Baxter-JSK-APC は吸引グリッパ上には片腕 2 つの両腕で計 4 つの RGB-D カメラセンサを搭載しており、これらのカメラを用いて実験環境の認識を行う。また Baxter-JSK-Softhand は頭部に魚眼カメラ (図 3.10) を配置することでユーザに広視野角の視覚情報を提供し、腹部

に導入された RGB-D カメラを用いて実験環境の認識を行う。さらに Baxter-JSK-Softhand の腹部設置されたマイク（図 3.11）を用いることでロボットの動作音と、ロボットと環境や対象物との衝突音をユーザに提供する。また、これらの視聴覚情報と各関節位置・トルク情報は ROS のメッセージとして遠隔地に送信される。以下ではこれらのロボットのハードウェア構成についてそれぞれ詳しく説明する。



図 3.1: 吸引グリッパ [51] を装備した双腕ロボット Baxter-JSK-APC

3.2.1.1 関節構成

本研究で用いるロボットは双腕ロボット Baxter [17, 18] は Rethink Robotics 社から発売されたロボットプラットフォームであり、人と同じように 2 本の腕を有したロボットである。以下で説明するロボットのスペックについては [53] より引用したものである。各腕に 7 つの回転関節と頭部に 1 つの回転関節があり全身で計 15 の回転関節を持つ。Baxter の左腕の回転関節の配置は図 3.3 に、左腕のリンク寸法に関しては図 3.4 に、各関節の可動範囲、最大角速度およびピークトルクに関しては表 3.1 に示すとおりである。また右腕に関しても左右対称であるものの同様の仕様となっており、本研究では図 3.3 に映されている Baxter 標準の並行グリッパは用いない。Baxter は各関節について Serial Elastic Actuator (SEA) が使われているため大

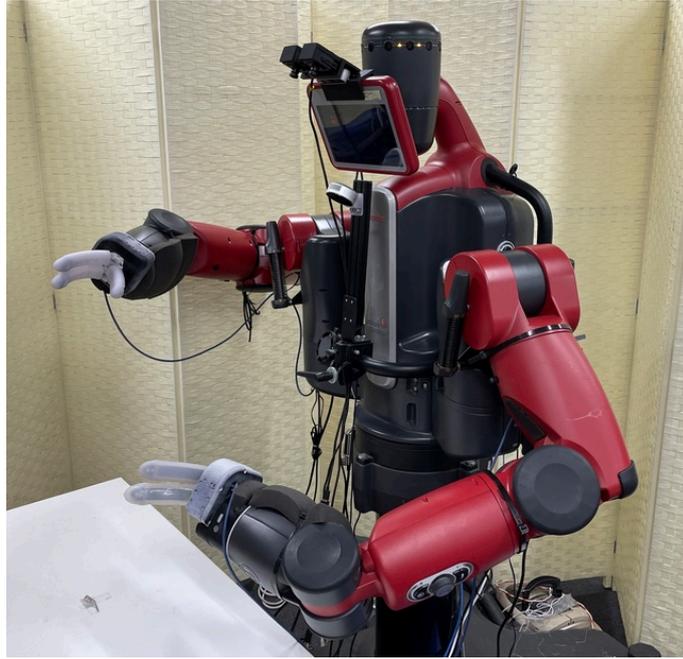


図 3.2: ソフトハンド [52] を装備した双腕ロボット Baxter-JSK-SoftHand

きな衝撃力に強く，本研究で行うロボット実機を用いるマニピュレーション作業の自律学習に適しており，先行研究でも用いられている [43]．Baxter の制御計算機はロボット体内に存在し Intel 社の Core i7-3770 という CPU を搭載しており関節制御や重力補償などを行っているが，本研究では画像処理・点群処理の高速化を行うための外部計算機を導入し，さらにその外部計算機に NVIDIA 社の Titan X という GPU を搭載することで深層学習計算の高速化を行う．

表 3.1: 双腕ロボット Baxter の片腕 7 関節の性能 ([53] より引用，改変)

Joint	Type	Min Limit	Max Limit	Max Speeds	Peak Torque
S0	revolute	-1.7016 rad	1.7016 rad	2.0 rad s^{-1}	50 N m
S1	revolute	-2.147 rad	1.047 rad	2.0 rad s^{-1}	50 N m
E0	revolute	-3.0541 rad	3.0541 rad	2.0 rad s^{-1}	50 N m
E1	revolute	-0.05 rad	2.618 rad	2.0 rad s^{-1}	50 N m
W0	revolute	-3.059 rad	3.059 rad	4.0 rad s^{-1}	15 N m
W1	revolute	-1.5707 rad	2.094 rad	4.0 rad s^{-1}	15 N m
W2	revolute	-3.059 rad	3.059 rad	4.0 rad s^{-1}	15 N m

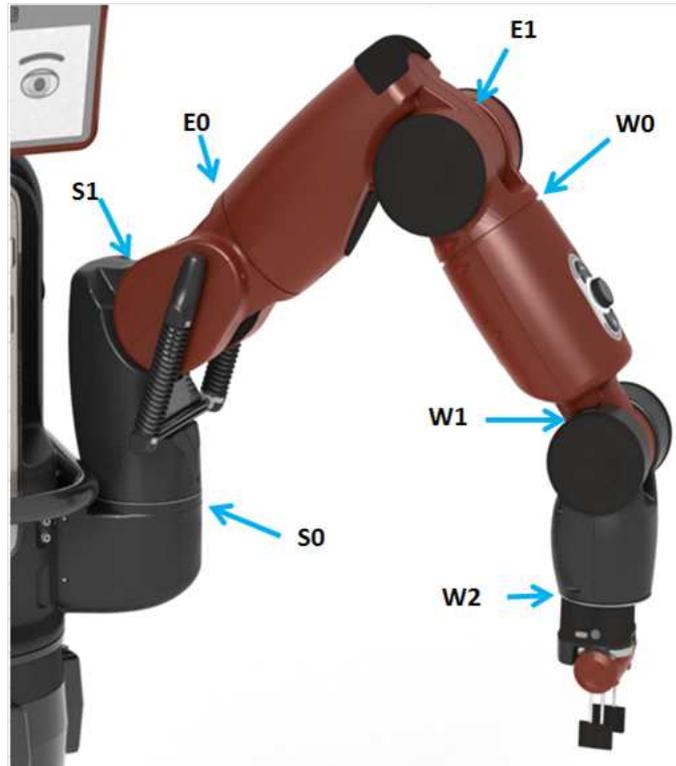


図 3.3: 双腕ロボット Baxter の片腕 7 関節 ([53] より引用)

3.2.1.2 直動・手先関節をもつ Baxter-JSK-APC の吸引グリッパ

本研究では双腕ロボット Baxter [17, 18] の標準の並行グリッパではなく、図 3.5 に示すような 1つの直動関節と 1つの回転関節を有した吸引グリッパを導入する。この吸引グリッパは倉庫内でのピッキングタスクのために開発されたグリッパ [51] の改造系であり、ロボットは直動関節と手先の回転関節を用いて対象物品の把持を行う。吸引グリッパの手先には図 3.7 に示すように直動関節があり、最大約 120 mm 前方に手先を移動させることができる。また図 3.8 に示すような回転関節もあり、ワイヤ駆動で手先を 180° 回転させることができる。この 2 関節を駆使することで複雑環境での単腕・双腕把持を実現し、安定な把持収納作業を行うことが可能となる。使用した吸引グリッパの回転関節はワイヤ駆動であるため衝撃力に強くマニピュレーション作業の自律学習に適している。また直動関節はラックアンドピニオン駆動であるがモータ負荷割合やモータ回転速度が計測できるため、物体や環境に接触したことを検出し物品へのアプローチ動作を停止することができる。

吸引装置としては吸引グリッパと吸引管を繋いで家庭用掃除機を設置し、この家庭用掃除機で吸引することで物体の把持を行う。家庭用掃除機は常に空気を吸引することで真空吸着方式では把持できない微小な穴の空いた物体や、グリッパ手先が物体に密着しない物体を把持することができる。また本研究では、この吸引グリッパの吸引管内に気圧センサを取り付け、この気圧センサの変化から物体を把持しているかどうかを認識する。これらの吸引グリッパの現在

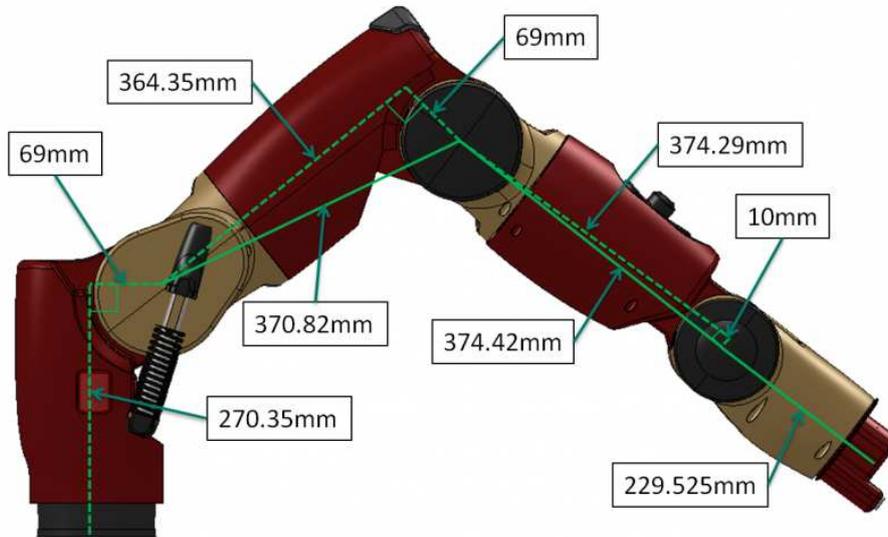


図 3.4: 双腕ロボット Baxter の片腕のリンク寸法 ([53] より引用)

の関節角度や関節角度指令，気圧センサの値などは，グリッパ上に搭載されたマイクロコンピュータ Arduino を介して制御計算機とのシリアル通信 [54] で送受信される。

3.2.1.3 接触力を分散する Baxter-JSK-Softhand のソフトハンド

本研究で用いるソフトハンド [52] は図 3.9 に示すように，ワイヤ駆動を採用することで蛇腹構造である指をそれぞれゴム外装することで，衝撃に強くこわれにくいハンドとなっている。3 本対向に配置されたゴム外装の指がワイヤ駆動によって曲がることによって，物体に対して接触力を分散して加えることができ，安定的な把持を行うことができる。3 本の指のワイヤはハンド根本にある 3 つのサーボモータでそれぞれ別々に駆動している。またソフトハンドはやわらかく物体や環境に馴染むことで物体を安定的に把持することができるため，物体操作作業の遠隔操縦に適しており，本研究では操縦実験で主に用いる。

3.2.1.4 物体認識のための RGB-D ハンドカメラセンサ

本研究では物体認識のための RGB-D カメラセンサとして Orbbec 社の Astra mini S [55] を各腕 2 つの計 4 つを図 3.5 と図 3.6 に示すように吸引グリッパ上に導入する。RGB-D カメラセンサとは RGB 画像と深度情報である深度画像を同時に撮影することができるカメラセンサであり，本研究で使用する際の Astra mini S カメラセンサの性能は表 3.2 に示すとおりであり，このカメラセンサは赤外光でパターンを投射しその歪みを計測することで深度情報を得るが，この手法では赤外光が吸収される物体（黒い物体）や乱反射しない物体（鏡面）の深度情報を得ることが難しい。そのため本研究ではこのカメラを 2 台横に並べて 2 台のカメラからの RGB 画像を用いるステレオによる深度情報を計算し，IR パターン投射による深度情報と組み合わせることでより正確な深度情報を得る。

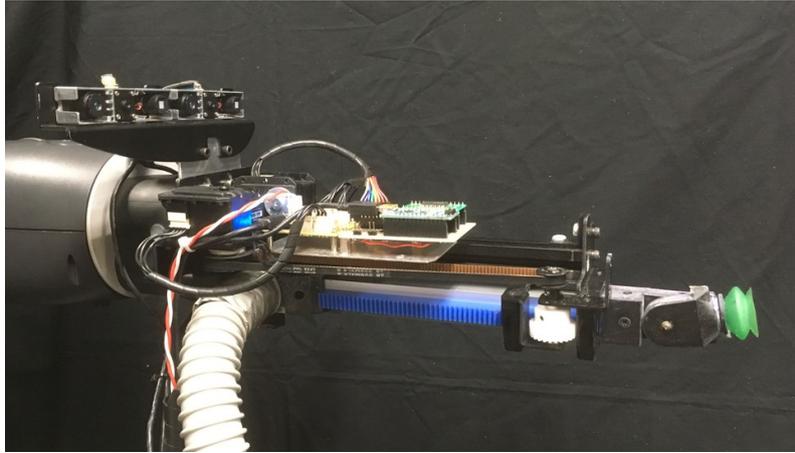


図 3.5: Baxter-JSK-APC の吸引グリッパと RGB-D カメラの全体像

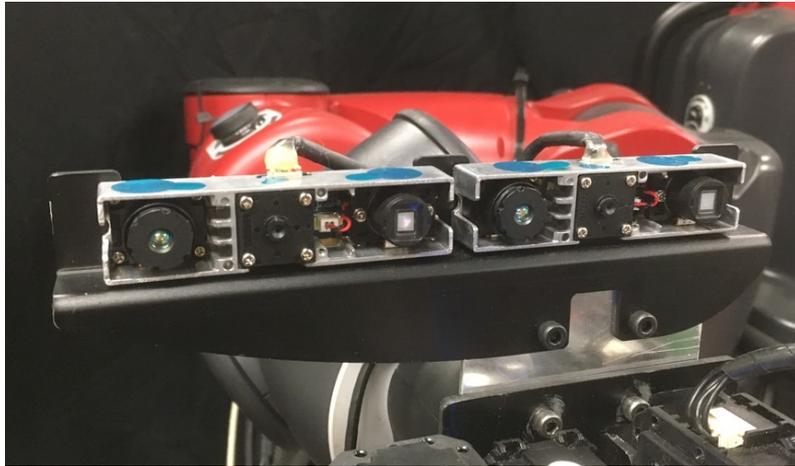


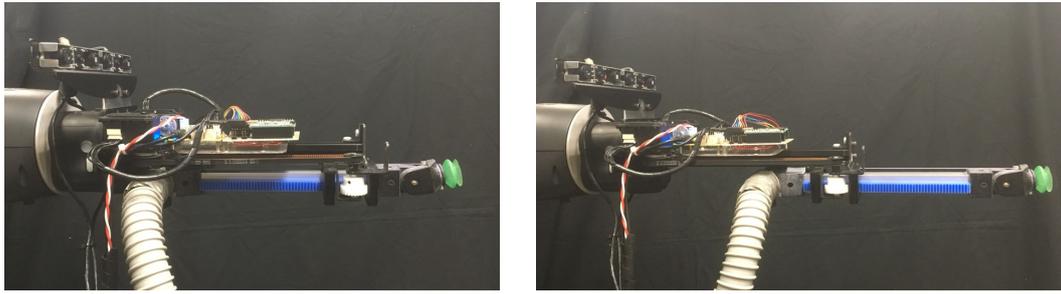
図 3.6: 吸引グリッパに取り付けられた Astra mini S カメラセンサ

3.2.1.5 物体認識のための RGB-D 腹部カメラセンサ

本研究では物体認識のための RGB-D カメラセンサとして Intel 社の Realsense L515 [56] を腹部に 1 つ導入する．本研究で使用する際の Realsense L515 カメラセンサの性能は表 3.3 に示すとおりであり，このカメラセンサは赤外光の反射光の位相差から距離を計測する IR Time of Flight 方式である．

3.2.1.6 遠隔操縦のための頭部魚眼カメラ

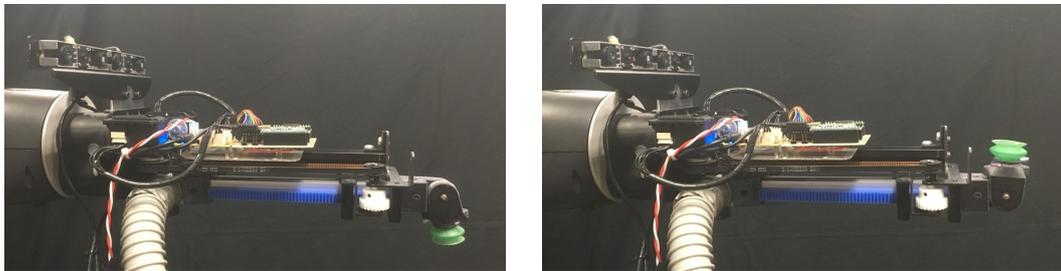
本研究では遠隔操縦の際に，Baxter-JSK 頭部に設置した図 3.10 に示す 4K 魚眼カメラ JK Imaging 社の Kodak PIXPRO SP360 4K [57] もしくは Shenzhen Ailipu Technology 社の ELP-USB4KHDR01-BL170 [58] を用いて操縦者に対して遠隔地の視覚情報を伝える．本研究で使用する際の Kodak PIXPRO Sp360 4K カメラセンサと ELP-USB4KHDR01-BL170 カメラセ



(a) 手先直動関節の収納時

(b) 手先直動関節の伸長時

図 3.7: 吸引グリッパの手先直動関節



(a) 手先回転関節の下向き 90° 回転時

(b) 手先回転関節の上向き 90° 回転時

図 3.8: 吸引グリッパの手先回転関節

表 3.2: Astra mini S カメラセンサの性能 [55]

	Astra mini S
RGB image resolution	1280 × 960 (QVGA)
RGB image horizontal field of view	60.0°
RGB image vertical field of view	49.5°
Depth image capture principle	IR Pattern
Depth image capture range	0.35 m – 2.5 m
Depth image resolution	640 × 480 (VGA)
Depth image horizontal field of view	60.0°
Depth image vertical field of view	49.5°

ンサの性能はそれぞれ表 3.4 と表 3.5 に示す。操縦者はヘッドマウントディスプレイ (HMD) やディスプレイに投影されたこの魚眼カメラの RGB 画像を元に物体操作作業の遠隔操縦を行う。



図 3.9: 双腕ロボット Baxter-JSK-Softhand のソフトハンド [52]

表 3.3: Intel Realsense L515 カメラセンサの性能 [56]

	Realsense L515
RGB image resolution	960 × 540
RGB image horizontal field of view	69.0°
RGB image vertical field of view	42.0°
Depth image capture principle	IR Time of flight
Depth image capture range	0.25 m – 3.9 m
Depth image resolution	640 × 480 (VGA)
Depth image horizontal field of view	70.0°
Depth image vertical field of view	55.0°

3.2.1.7 遠隔操縦と人との対話のための腹部全方位マイク

Baxter-JSK は腹部に図 3.11 に示す全方位マイク Respeaker Mic Array [59] を搭載しており、遠隔操縦の際には操縦者に対して遠隔地の聴覚情報を伝える。操縦者はこの全方位マイクで計測された聴覚情報を、イヤホンを通して知覚し、この情報を元に物体操作作業の遠隔操縦を行う。また人からの音声による作業実行の際にはこの腹部全方位マイクを用いて人の発話をロボットが計測・認識し、この聴覚情報を元に人との対話を行う。



図 3.10: Baxter-JSK の頭部魚眼カメラ [57]

表 3.4: Kodak PIXPRO SP360 4K カメラセンサの基本性能 [57]

	Kodak PIXPRO SP360 4K
RGB image resolution	2880 × 2880
RGB image horizontal field of view	235.0°
RGB image vertical field of view	235.0°

3.2.2 双腕移動ロボット PR2-JSK のハードウェア構成

本研究で用いる双腕移動ロボット PR2-JSK(図 3.12) は双腕移動ロボット PR2 [19] に対して 計算機やセンサを交換・追加したものである。高さは腰部にある直動リンクにより 1330 mm から 1645 mm に変更可能であり, 両腕はグリッパを除いて 7 自由度であり, 1.8 kg までのものを抱えることができる [60]。PR2 全体の関節自由度は移動台車部分を除いて以下のように なっており, 詳細については表 3.6 に記載する [60, 61]。

- 腰部: 1 自由度
- 頭部: 2 自由度
- 腕部: 7 自由度 (両腕合計 14 自由度)

表 3.5: ELP-USB4KHDR01-BL170 カメラセンサの基本性能 [58]

	ELP-USB4KHDR01-BL170
RGB image resolution	3840 × 2160
RGB image horizontal field of view	170.0°
RGB image vertical field of view	170.0°



図 3.11: Baxter-JSK の腹部マイク [59]

- グリッパ: 1 自由度 (両腕合計 2 自由度)
- 腰部距離センサ駆動台座: 1 自由度

3.2.2.1 物体認識のための RGB-D カメラセンサ

PR2-JSK は頭部に多くのカメラセンサを搭載しているが、本研究では主に頭部上方に取り付けられた RGB-D カメラセンサである Microsoft 社製 Kinect を用いる [62] を用いる。この RGB-D カメラセンサは赤外光のパターンを投射し、そのパターンの歪みから深度を計測し、RGB 画像と位置合わせされた深度画像を生成するものである。

3.2.2.2 遠隔操縦と人との対話のため頭部全方位マイク

PR2-JSK は Baxter-JSK 同様に、頭部に全方位マイク Respeaker Mic Array [59] を搭載しており、遠隔操縦の際には操縦者に対して遠隔地の聴覚情報を伝える。操縦者はこの全方位マイクで計測された聴覚情報を、イヤホンを通して知覚し、この情報を元に物体操作作業の遠隔操縦を行う。また人からの音声による作業実行の際にはこの頭部全方位マイクを用いて人の発話



図 3.12: 双腕移動ロボット PR2-JSK

をロボットが計測・認識し，この聴覚情報を元に人との対話を行う．

3.2.2.3 移動のためのレーザ測距センサ

PR2-JSKは移動のために Hokuyo 社製のレーザ測距センサである Top-URG UTM-30LX [63] を用いる．PR2-JSKは腰部と移動台車部分に2つの UTM-30LX を搭載しているが，移動の際には主に移動台車のセンサデータを用いて環境を認識し，予めあたえられた地図情報と照らし合わせることで地図情報との自己位置を推定する．

3.2.3 単腕移動ロボット Fetch-JSK のハードウェア構成

本研究で用いる単腕移動ロボット Fetch-JSK(図 3.13)は単腕移動ロボット Fetch [15] に対して計算機やセンサを交換・追加したものである．高さは腰部にある直動リンクにより 1096 mm から 1491 mm に変更可能であり，腕はグリップを除いて 7 自由度であり，6 kg までのものを抱えることができる [15]．Fetch 全体の関節自由度は移動台車部分を除いて以下のようになっ

表 3.6: 双腕移動ロボット PR2 の各関節の性能 ([61] より引用, [60] に基づいて一部修正)

Joint	Type	Min Limit	Max Limit	Max Speeds	Peak Torque
torso lift joint	prismatic	0 mm	310 mm	0.013 m s^{-1}	10 000 N
head pan joint	revolute	-168°	168°	6.00 rad s^{-1}	2.65 N m
head tilt joint	revolute	-30°	60°	5.00 rad s^{-1}	15.00 N m
left shoulder pan joint	revolute	-40°	130°	2.10 rad s^{-1}	30.00 N m
right shoulder pan joint	revolute	-130°	40°	2.10 rad s^{-1}	30.00 N m
left/right shoulder lift joint	revolute	-30°	80°	2.10 rad s^{-1}	30.00 N m
left upper arm roll joint	revolute	-44°	224°	3.27 rad s^{-1}	30.00 N m
right upper arm roll joint	revolute	-224°	44°	3.27 rad s^{-1}	30.00 N m
left/right elbow flex joint	revolute	0°	133°	3.30 rad s^{-1}	30.00 N m
left/right forearm roll joint	continuous	-	-	3.60 rad s^{-1}	30.00 N m
left/right wrist flex joint	revolute	0°	130°	3.10 rad s^{-1}	10.00 N m
left/right wrist roll joint	continuous	-	-	3.60 rad s^{-1}	10.00 N m
left/right gripper joint	prismatic	0 mm	86 mm	0.2 m s^{-1}	1000 N
laser tilt joint	revolute	-45°	85°	10.00 rad s^{-1}	0.65 N m

ており, 詳細については表 3.7 に記載する [64].

- 腰部: 1 自由度
- 頭部: 2 自由度
- 腕部: 7 自由度
- グリッパ: 1 自由度

3.2.3.1 移動のためのレーザ測距センサ

Fetch-JSK は移動のために SICK 社製のレーザ測距センサである SICK TIM571 [65] を用いる. Fetch-JSK は移動の際には移動台車のセンサデータを用いて環境を認識し, 予めあたえられた地図情報と照らし合わせることで地図情報との自己位置を推定する.

3.2.3.2 物体認識のための RGB-D カメラセンサ

本研究では Fetch-JSK の頭部に搭載されている RGB-D カメラセンサである ARTCREATION 3D TECHNOLOGY 社製 Primesense Carmine 1.09 [66] を用いる. この RGB-D カメラセンサは赤外光のパターンを投射し, そのパターンの歪みから深度を計測し, RGB 画像と位置合わせされた深度画像を生成するものである.



図 3.13: 単腕移動ロボット Fetch-JSK

3.2.3.3 遠隔操縦と人との対話のため頭部全方位マイク

Fetch-JSK は PR2-JSK 同様に、頭部に全方位マイク Respeaker Mic Array [59] を搭載しており、遠隔操縦の際には操縦者に対して遠隔地の聴覚情報を伝える。操縦者はこの全方位マイクで計測された聴覚情報を、イヤホンを通して知覚し、この情報を元に物体操作作業の遠隔操縦を行う。また人からの音声による作業実行の際にはこの頭部全方位マイクを用いて人の発話をロボットが計測・認識し、この聴覚情報を元に人との対話を行う。

表 3.7: 単腕移動ロボット Fetch の各関節の性能 ([64] より引用, 改変)

Joint	Type	Min Limit	Max Limit	Max Speeds	Peak Torque
torso lift joint	prismatic	0 mm	400 mm	0.1 m s^{-1}	450 N
head pan joint	revolute	-90°	90°	1.57 rad s^{-1}	0.32 N m
head tilt joint	revolute	-45°	90°	1.57 rad s^{-1}	0.68 N m
shoulder pan joint	revolute	-92°	92°	1.25 rad s^{-1}	33.82 N m
shoulder lift joint	revolute	-70°	87°	1.45 rad s^{-1}	131.76 N m
upper arm roll joint	continuous	-	-	1.57 rad s^{-1}	76.94 N m
elbow flex joint	revolute	-129°	129°	1.52 rad s^{-1}	66.18 N m
forearm roll joint	continuous	-	-	1.57 rad s^{-1}	29.35 N m
wrist flex joint	revolute	-125°	115°	2.26 rad s^{-1}	25.70 N m
wrist roll joint	continuous	-	-	2.26 rad s^{-1}	7.36 N m
gripper finger joint	prismatic	0 mm	50 mm	0.05 m s^{-1}	60 N

3.3 操縦自律間横断学習ロボットシステムのためのソフトウェア基盤

3.3.1 分散型ロボットミドルウェア

3.3.1.1 分散型ロボットシステム ROS

ROS [50] とは Robot Operating System の略称であり, 分散型ロボットシステムフレームワークの 1 つである. ROS は roscore というネームサーバを立てることで, ネットワーク上の計算機の複数のプロセスをノードとして管理・実行し, 各プロセス間通信を Publish-Subscribe 方式で行うフレームワークである. また ROS はオープンソース開発が行われており, 全世界的に改良・改善が行われている. 本研究で構築したロボットシステムでは, Baxter-JSK, PR2-JSK, Fetch-JSK それぞれの内部計算機に roscore のネームサーバを立て, 2 つの外部計算機にて画像処理やロボットの動作生成を行い, ロボットの動作指令をロボット内部計算機に送信する. このように複数計算機で並列処理を行うことで 1 計算機の負担を軽減するとともに, ロボットシステムの拡張が容易なシステムを構築する. また Baxter-JSK-SoftHand の場合には, 3.2.1.2 目で説明した吸引グリッパ上のマイクロコンピュータも, ROS の通信を介して吸引グリッパ上のセンサの読み取りや動作の制御を行う.

3.3.2 動作生成・計画

3.3.2.1 ロボット動作生成ソフトウェア Euslisp

Euslisp [67]とは Lisp 言語をインタフェースとしたロボットモデリング・動作生成ソフトウェアである。モデリング機能としては図 3.14 に示すようなロボットモデルや図 3.15 に示すような環境モデルを生成することができ、これらを用いてロボットの動作をシミュレーションすることや ROS を介してロボット実機の関節角度を再現することができる。またロボット動作生成機能としては、逆運動学計算を行うことでロボットの手先を目標姿勢に移動するための関節角度列を計算し、関節角度列の遷移を時系列補間してロボット実機に動作司令することができる。

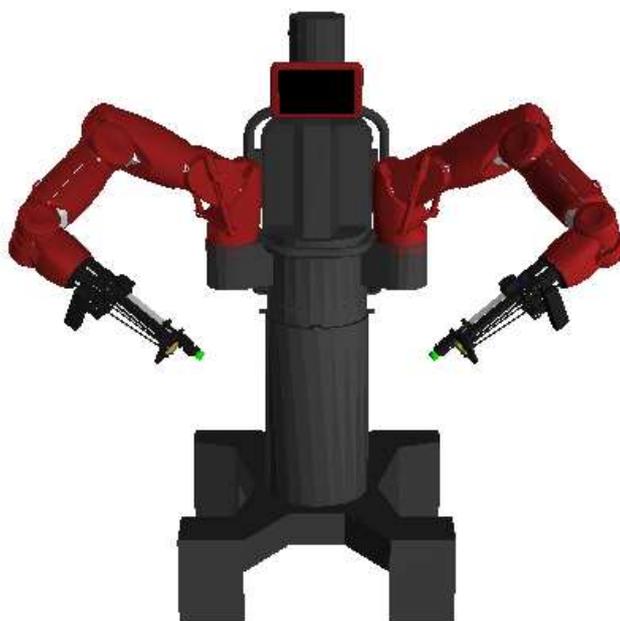


図 3.14: Euslisp による Baxter-JSK のロボットモデリング

3.3.2.2 ロボットモーションプランニングフレームワーク MoveIt!

MoveIt! [68]とはオープンソースで開発されているロボット動作計画フレームワークの1つであり、筆者も改良・改善に参加している。このフレームワークは逆運動学やロボットと環境の干渉計算や干渉を回避した関節角度動作列の生成を行う。またロボットモデルと環境を考慮した動作計画には The Open Motion Planning Library (OMPL) [69]を用いている。本研究では逆運動学計算は 3.3.2.1 目で説明した Euslisp を用いて計算し、単腕・双腕把持動作などの自己干渉や環境との接触回避動作のために MoveIt! を介して OMPL [69] を利用する。



図 3.15: Euslisp による研究室のロボットモデリング

3.3.3 環境認識

3.3.3.1 点群・画像処理ソフトウェア

本研究では三次元点群処理ソフトウェアとして Point Cloud Library (PCL) [70] を用いる。PCL は深度情報である三次元点群を高速に処理するオープンソース開発されたソフトウェアであり、本研究では興味領域抽出やユークリッド距離クラスタリングなどの処理に用いる。また深層学習以外の画像処理では主に OpenCV [71] というオープンソースの画像処理ソフトウェアを用いる。このソフトウェアも様々な画像処理関数が用意されており、本研究ではマスク画像処理や画像のサイズ変換などの処理に用いる。

3.3.3.2 深層学習フレームワーク Chainer

Chainer [72] とは Preferred Networks 社 [73] で主に開発されているオープンソースの深層学習フレームワークであり、筆者も開発に参加している。このフレームワークでは深層学習に必要な畳み込み層、正規化層、活性化関数や学習率の最適化手法などの様々な関数やアルゴリズムが Python アプリケーションとして提供されており、動的計算グラフを構築することで逆伝播計算を動的にかつ高速に行うことができる。また CuPy [74] を介して NVIDIA 社の GPU にて計算を行えるため深層学習計算を高速に並列計算することができる。さらに本研究では Mask-RCNN を元にした CNN の計算の高速化のために、深層学習並列分散ソフトウェア ChainerMN [75] を用いて複数 GPU による並列学習を行う。

3.3.3.3 深層学習画像処理フレームワーク ChainerCV

ChainerCV [76] とは Preferred Networks 社 [73] で主に開発されているオープンソースの深層学習画像処理フレームワークであり、筆者も開発に参加している。このフレームワークは物

体認識や物体インスタンス矩形検出，意味領域分割，インスタンス意味領域分割の学習済みモデルを提供しており，これらの認識タスクのための可視化関数などが実装されている．本研究では意味領域分割の評価関数や Mask-RCNN の特徴量抽出器の構造などに本ソフトウェアを用いる．

3.3.4 有限状態機械管理

3.3.4.1 有限状態機械管理フレームワーク SMACH

SMACH [77] とはオープンソース開発されている有限状態機械管理フレームワークであり，ロボットの作業動作を有限状態機械として管理することができる．SMACH [77] は ROS との通信を行えるため，本研究では双腕ロボットの認識や把持などの状態管理に用いる．本研究では左腕と右腕の認識や作業動作の管理として状態を管理する必要があり，有限状態機械の管理として SMACH を用いてロボットシステムを構築する．

3.4 おわりに

本章では本研究の検証実験で用いるロボットシステムをハードウェアとソフトウェアという観点から説明した．

第 4 章

タンジブルインタフェースによる作業手順の教示学習

4.1 はじめに

本章では対象物の接触・回転状態を表現するタンジブルインタフェースと、それを用いた物体操作作業の操縦実行とその操縦による作業手順を表す動作順序の教示学習について説明する。本研究における動作順序の教示学習の目的は手順があり接触の多い複雑な物体操作作業において操縦を通じて作業手順を動作順序として教示学習することであり、図 4.1 に示すように教示学習された動作順序を用いて作業を自律実行することで操縦自律間横断学習を実現する。まず本研究における対象物の状態に基づいた作業手順の記述法について説明する。そして物体操作作業におけるユーザインタフェースについて先行研究と比較しながら議論した後に、タンジブルインタフェースによる物体操作作業の操縦の有用性について説明する。そしてタンジブルインタフェースの実装である Miniature Tangible Cube [37] とそれを用いた物体操作作業の操縦実行について説明したあとに、このタンジブルインタフェースによる操縦実行における動作順序の教示学習について説明する。最後に双腕移動ロボット PR2-JSK と双腕ロボット Baxter-JSK-Softhand を用いて、タンジブルインタフェースによる物体操作作業の操縦実行とそれによる動作順序の教示学習の有効性の評価実験を行う。

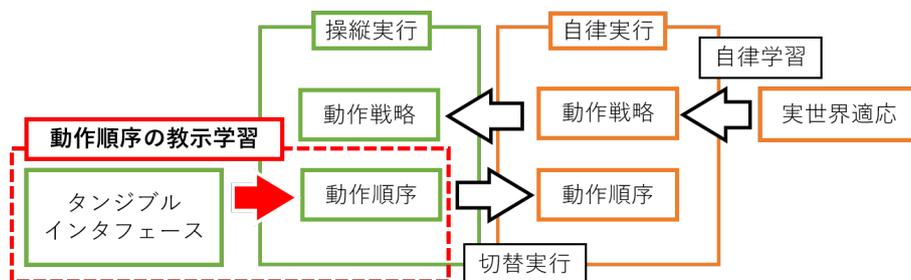


図 4.1: 操縦自律間横断学習における動作順序の教示学習の位置づけ

4.2 対象物の状態に基づいた動作順序グラフによる作業手順記述

本節では本研究における作業手順を表すための動作順序の記述法について説明する．本研究では作業を対象物の状態に基づいて動作とその連なりである動作順序に分割し，動作順序を作業手順として教示学習を行う．このとき動作順序の記述として対象物の状態をノード，ロボットの動作をエッジとする有向グラフを用い，この動作順序を示す有向グラフを本研究では動作順序グラフと呼ぶ．よって V をノード集合， E をエッジ集合， S^{obj} を対象物の状態集合， A をロボットの動作集合とする式 (4.1) に示す有向グラフを動作順序グラフ G とし，本研究ではこの動作順序グラフを用いて動作順序の教示学習を行う．

$$\begin{aligned} G &= (V, E) \\ V &= \{s^{obj} \mid s^{obj} \in S^{obj}\} \\ E &= \{a \mid a \in A\} \end{aligned} \quad (4.1)$$

本研究で作業手順の教示学習を行う動作順序グラフの一例を図 4.2 に示す．図 4.2 に示すように対象物がロボットや環境とのどのように接触しているかの状態をノードとし，そのノードの間をロボットの動作をエッジとする有向グラフによって動作順序を記述する．なお動作順序グラフのノード名とエッジ名は Lisp によるリストを用いて記述しており，作業で対象とする対象物の数に応じて Lisp のリストに追加することで複数の対象物への動作順序を記述する．例えば図 4.2 に示す一例では 2 つの対象物に対する動作順序を記述しており，エッジである動作名にリストを用いることでどちらの対象物に動作を行ったかを記述することができ，図 4.2 では上のノードから 1 つめの対象物に対して，(dualarm-left-right-top-approach-push-hold-action) と (dualarm-left-right-push-hold-action) の 2 種類の動作を行った場合のノードの遷移を示している．本研究ではこのリストの順番から対象物を区別する ID を設定し，1 つめの対象物については対象物 0，2 つめの対象物については対象物 1 といったように動作順序グラフに記述された対象物の状態ノード名のリストの順番を ID として区別する．図 4.2 の動作順序グラフの左側の分岐では 2 つの対象物が置かれている状態を示す上のノードから，1 つの対象物の左側面上方に対して左手で，右側面上方に対して右手で接触する動作 ((dualarm-left-right-top-approach-push-hold-action)) を行うことで左下のノードである対象物に対して左側面上方と右側面上方に接触している状態に遷移することができることを示している．また図 4.2 の動作順序グラフの右側の分岐では 2 つの対象物が置かれている状態を示す上のノードから，1 つの対象物の左側面に対して左手で，右側面に対して右手で接触する動作 ((dualarm-left-right-push-hold-action)) を行うことで右下のノードである対象物に対して左側面と右側面に接触している状態に遷移することができることを示している．

次に動作順序の記述に対象物の状態をノード，ロボットの動作をエッジとする有向グラフで

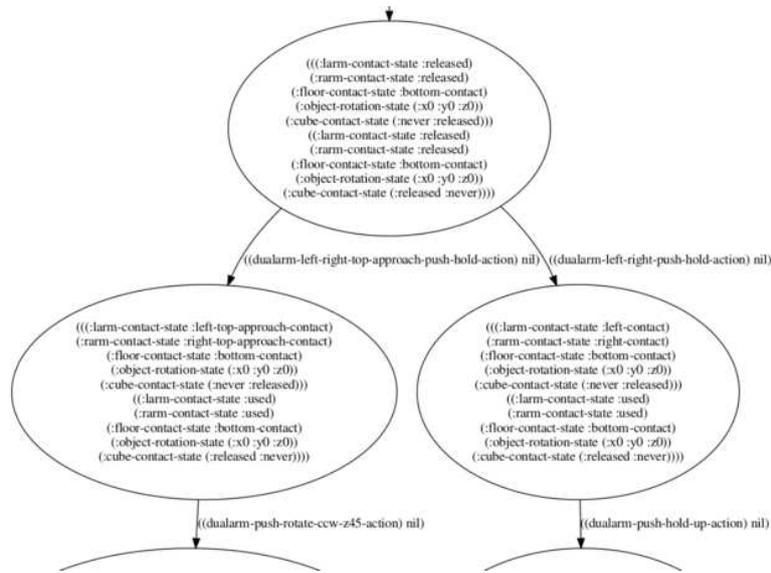


図 4.2: 作業手順を表す動作順序グラフの一例

ある動作順序グラフを用いる利点について説明する．動作順序グラフを用いて動作順序を記述する 1 つ目の利点は，操縦実行において新たな作業の動作順序を教示学習する際に，これまで学習されている動作順序に追加することが容易であるためである．これは 2.4.1.1 目で説明した利点の 1 つである再利用性と同じであり，新たな作業を教示学習する際に対象物の状態に基づいて新たな動作順序をこれまで学習された動作順序グラフに追加・分岐する形で追加することで操縦実行における動作順序の教示学習を実現できるためである．2 つ目の利点は自律実行において対象物の状態に基づいて作業実行に必要な動作列を計画できるためである．つまり作業の自律実行において，動作順序グラフのノードが対象物の状態でありエッジがロボットの動作であるため，この動作順序グラフ上で現在の対象物の状態ノードから目標の状態ノードへの経路を計画することで，この計画された経路上のエッジを作業の遂行に必要な動作列として求めることができ，これを用いて教示学習された動作順序による作業の自律実行が実現できるためである．

4.3 物体操作作業の操縦におけるユーザインタフェース

ロボットの操縦は危険な環境での遠隔操縦技術やテレエグジスタンス技術として研究されており [12, 32, 33, 78–85]. 視覚情報や力覚・触覚情報などのロボットが感知するセンサ入力をリアルタイムで提示することで，操縦者はロボットを巧みに動作できる操縦システムが提案されてきた．近年ではこれらの技術を用いた操縦ロボットによる物体操作作業が注目されており，操縦しやすいユーザインタフェース [32, 33, 40–42] や操縦者への様々なセンサフィードバック [81, 82]，操縦者・ロボット間の通信などによる時間遅延 [83] について様々な研究が行われている．しかし，これらの操縦システムはグリッパ手先座標などロボットに着目したシス

テムであり、操縦者はロボット手先の姿勢と力の指令しか出来ないため、双腕物体操作などの物体や環境との物理接触の多い作業の実行は困難である。これらは手先指向型インタフェースと分類することができ、これまでリアルタイムなセンサフィードバックを有したインタフェース [12, 32, 33, 78–83] や、VR デバイスを用いた手法 [85]、画像情報から操縦者の姿勢を認識する手法 [33, 84] が提案されてきた。本研究ではこのような手先指向型インタフェースを用いた操縦のことを手先指向型操縦と呼ぶ。

そこで本章ではロボットではなく操作する対象物に着目し、対象物の接触・回転状態を表現するためにタンジブルインタフェースを物体操作作業の操縦実行に用いることを提案する。タンジブルインタフェース [34] とは、触ることで操作するインタフェースのことであり、ユーザが直接接触れる物体を模したインタフェース [86, 87] としてヒューマンコンピュータインタラクション (HCI) の分野では広く用いられている。本研究ではタンジブルインタフェースを用いて対象物の回転と接触を表現することで、対象物があたかも操縦者の目の前にあるような対象物を模した (Miniature な) インタフェースを作成する。操縦者がユーザインタフェースをあたかも対象物であるかのように物体操作をすることで、ロボットが同様の物理拘束・幾何拘束を実現できるような動作を実行するユーザインタフェースである。つまり、タンジブルインタフェースを操縦者が触り、持ち上げ、回転させることでロボットの物体操作操縦ができるため、図 4.3 に示すようにこれまでの手先指向型インタフェースを用いた先行研究と比較して、応答性は低い対象物・環境・ロボットの間接触を表現できるため手順があり接触の多い複雑な物体操作作業を操縦するのに適しているといえる。また対象物の接触・回転状態を表現するインタフェースとして、タンジブルインタフェースの他にも HMD と VR デバイスを用いた実装も考えられるが、物体同士の重畳・重なり合いのより直感的な表現、プロジェクションを行うことで XR 技術との融合の可能性から本研究ではタンジブルインタフェースを用いて対象物の接触・回転状態を表現する。本研究ではこのようにタンジブルインタフェースで対象物を模すことを対象物指向と呼び、対象物指向のタンジブルインタフェースを用いた操縦を対象物指向型操縦と呼ぶ。本研究ではこのタンジブルインタフェースを用いて対象物の接触・回転状態を表現することで、手順があり接触の多いような物体操作作業の操縦実行とその動作順序の教示学習を実現する。

4.4 物体操作作業の操縦におけるタンジブルインタフェース

本研究で提案するタンジブルインタフェースを用いた、対象物指向型操縦のコンセプトは図 4.4 に示す。図 4.4 の左側が操縦者側、右側が遠隔地のロボットであり、左側の赤・黄・青・緑・紫色の立方体で表されているものが本研究で提案するタンジブルインタフェースである。図 4.4 のタンジブルインタフェースは同じ色の遠隔地の対象物と対応づけすることで、紫色のタンジブルインタフェースのように操縦者はこれを触り、持ち上げ、回転させることで、そのタンジブルインタフェースの姿勢と接触状態をロボットが対象物に対して再現することで遠隔操縦による物体操作を達成する。また赤・黄色のタンジブルインタフェースのように隣接しておくことで物体同士の隣接接触や、青・緑色のタンジブルインタフェースのように重ねておく

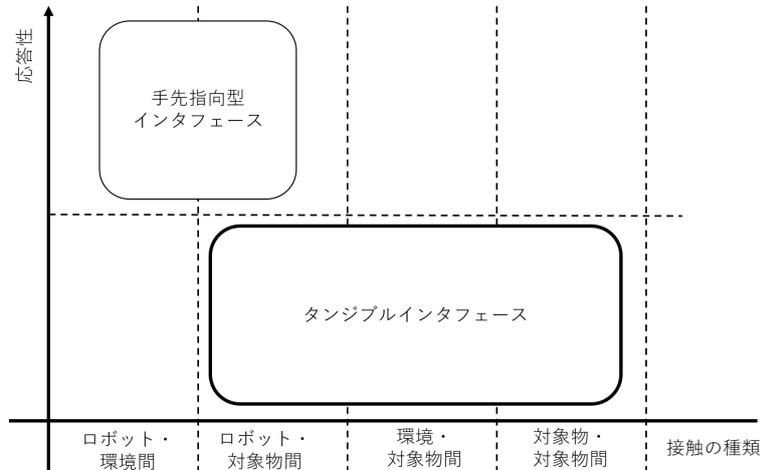


図 4.3: タンジブルインタフェースと手先指向型インタフェースの接触の種類と応答性の比較

ことで物体同士の重畳接触を表現でき、ロボットはこの姿勢と接触状態を元に接触状態を重視した物体配置作業を行うことができる。

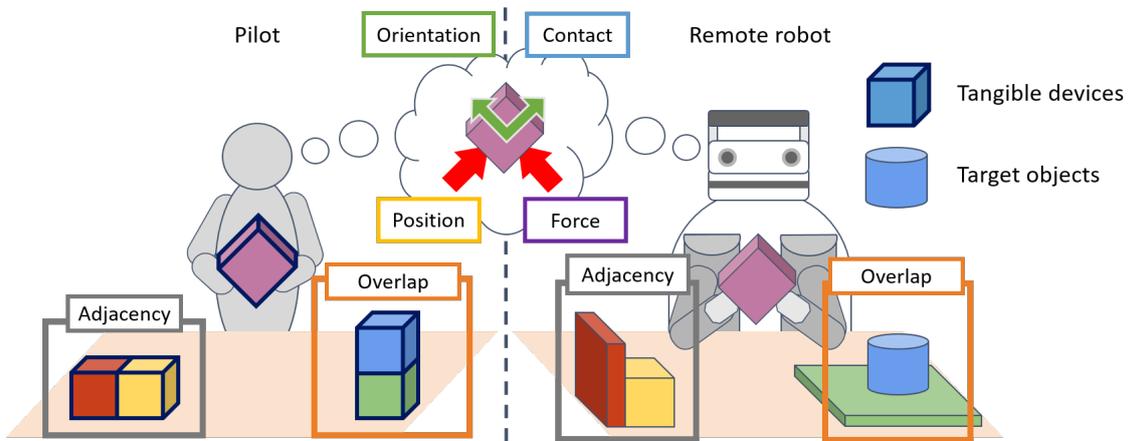


図 4.4: 物体操作作業の操縦実行におけるタンジブルインタフェースのコンセプト図 [37]

4.4.1 対象物指向型操縦と手先指向型操縦の比較

本項では本研究で提案するタンジブルインタフェースを用いた対象物指向型操縦と、従来のユーザインタフェースを用いた手先指向型操縦について比較して説明する。対象物指向型操縦と手先指向型操縦の違いについて図 4.5 に示す。先行研究の多く [32, 33, 78–85] は手先指向型操縦(図 4.5 右)に分類されるように、ロボットのエンドエフェクタの姿勢・力を操縦し、そのエンドエフェクタに対する力覚・触覚をリアルタイムで操縦者にフィードバックするものである。一方、本研究で提案するタンジブルインタフェースを用いた対象物指向型操縦は、図 4.5

左に示すように，対象物の姿勢と接触状態を直接的に表現するインタフェースを用いた操縦となる．つまり，手先指向型操縦は手先指向型インタフェースを通してロボットのエンドエフェクタの幾何・物理拘束を表現しているのに対して，タンジブルインタフェースを用いた対象物指向型操縦は対象物の幾何拘束と物理拘束を表現するような操縦手法といえる．

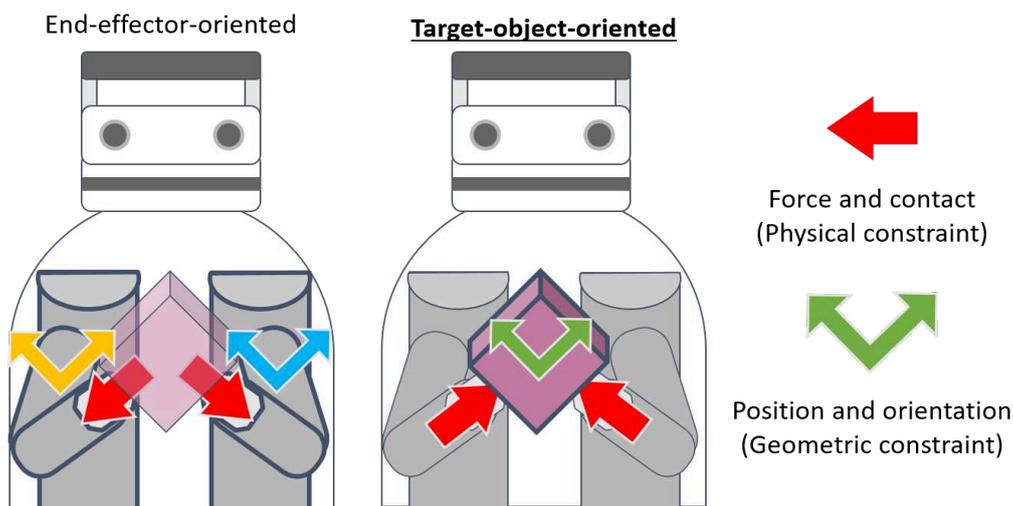


図 4.5: タンジブルインタフェースによる対象物指向型操縦と手先指向型インタフェースによる手先指向型操縦の比較 [37]

4.4.2 タンジブルインタフェースによる対象物指向型操縦の利点と欠点

次に物体操作作業の操縦におけるタンジブルインタフェースによる対象物指向型操縦の長所・短所について，先行研究である手先指向型インタフェースによる手先指向型操縦と比較しながら説明する．タンジブルインタフェースの長所を以下に示す．

複雑な接触のある物体操作動作の表現が容易である

手先指向型インタフェースはロボット・対象物間の接触のみを表現するが，タンジブルインタフェースはロボット・対象物間，対象物・対象物間，環境・対象物間の接触を表現することができるため，図 4.4 に示すように対象物同士や対象物・環境間の重畳，隣接を表現することができる．

操縦者とロボットの身体構造の違いを解決することができる

手先指向型インタフェースはロボットのエンドエフェクタを指定して操縦するが，タンジブルインタフェースは対象物に対する物理・幾何拘束を表現しているため，それを満たすロボットの姿勢を探索することができる．つまりエンドエフェクタ以外の接触点を探索範囲に含めることでロボットが全身接触動作を行うことが出来，人とロボットの身体構造の違いの解消が期待できる．

リアルタイムなセンサフィードバックが必要ない

手先指向型インタフェースはロボットのエンドエフェクタに対する物理拘束を表現するためリアルタイムな力覚フィードバックが求められるが、タンジブルインタフェースは対象物に対する物理拘束を直接的に表現しているため操縦者に対してリアルタイムに力覚をフィードバックする必要がない。しかし、これは後述するがリアルタイム性を失っているため短所の一つでもある。

以上が長所となる。次に短所を以下に示す。

想定されている物理・幾何拘束しか扱えない

タンジブルインタフェースは対象物に対する物理・幾何拘束を表現するために、動作を探索する動作プランニングが必須となるため、想定されている物理・幾何拘束とそれに対応する動作を入念に設計する必要がある。

扱える対象物の形状と剛性に制限がある

タンジブルインタフェースは対象物を模したインタフェースであるために、そのインタフェースの形状・剛性から大きくかけ離れた物体を扱うことは出来ない。特に流体・粉状の物体など形状が大きく変化するものは対応できない。

リアルタイム性が低い操縦システムになる

タンジブルインタフェースは対象物に対する物理・幾何拘束を表現するために、どのような動作が適しているかロボットが計画・実行するため、手先指向型インタフェースを用いた操縦実行に比べてリアルタイム性が低くなってしまう。

以上の長所・短所から、物体操作作業の操縦実行に置いてタンジブルインタフェースは想定された物体に対して想定された動作を行う場合に効果を発揮するものであり、特に双腕物体操作などの手順があり接触の多い物体操作において、手先指向型インタフェースよりも直感的に操縦ができるため、真価が発揮されるものである。特に宇宙空間などリアルタイム性が欠けてしまう作業環境では想定されたタスクを行うという点では適していると考ええる。

4.4.3 複雑な物体操作作業におけるタンジブルインタフェース

ここまでタンジブルインタフェースと手先指向型インタフェースの違いについて説明してきた。タンジブルインタフェースは複雑な接触があるような物体操作作業において、対象物に対する物理・幾何拘束を表現することに適している。そのため、タンジブルインタフェースのアプリケーションとして、複数の物理・幾何拘束のある双腕物体操作作業 [88, 89] などに適していると考ええる。そこで、本研究ではタンジブルインタフェースを手順があり接触の多い複雑な物体操作作業の操縦に用いることを提案する。

4.4.4 タンジブルインタフェースを用いた対象物指向型操縦システム

本研究では近接センサと IMU を用いてタンジブルインタフェースである Miniature Tangible Cube を実装し，図 4.6 に示す Miniature Tangible Cube を用いた対象物指向型操縦システムを構築する．まず Miniature Tangible Cube というタンジブルインタフェースに対して操縦者が対象物に対して行いたい接触状態や回転を表現し，このタンジブルインタフェースの近接センサと IMU のセンサ値からどのような動作を行いたいかの動作シンボルを選択することで動作プランニングを行い，選択させた動作シンボルを実行することで遠隔ロボットが物体操作を行うシステムを構築する．

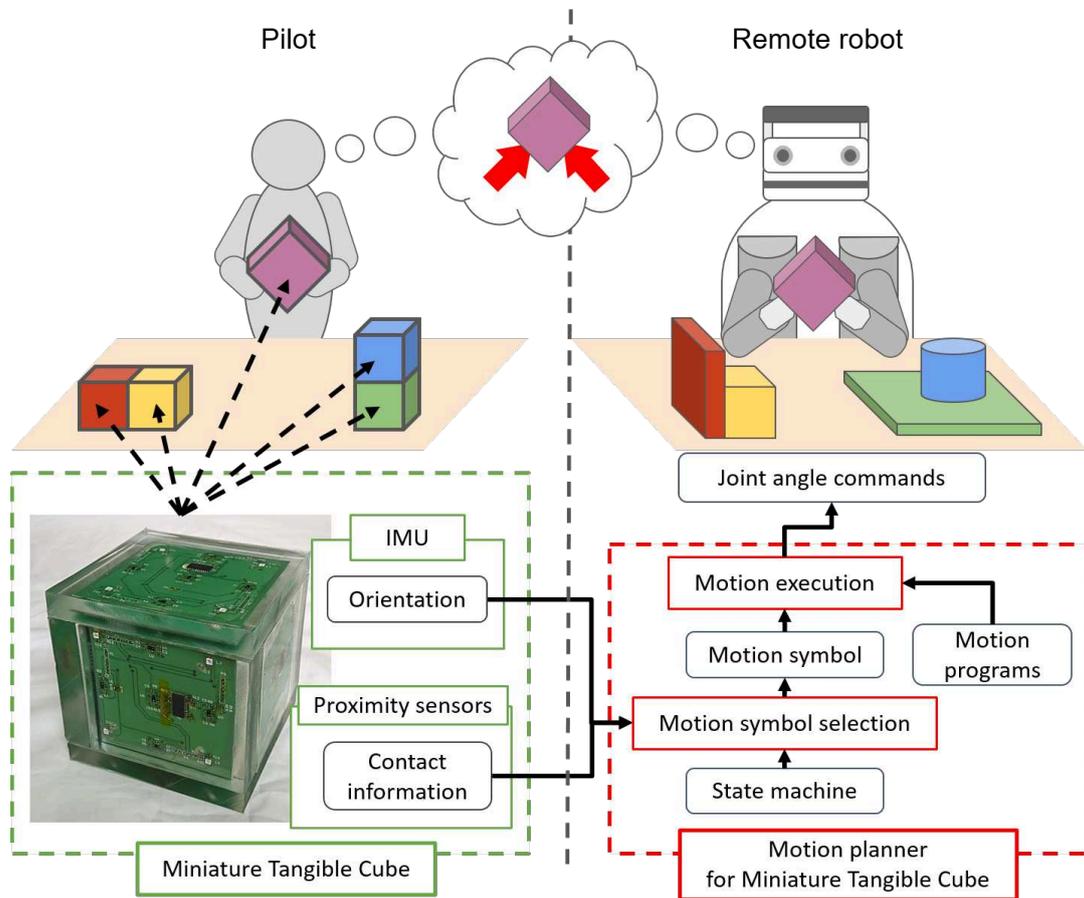


図 4.6: Miniature Tangible Cube を用いた対象物指向型操縦システム [37]

4.5 対象物の接触・回転状態を表現するタンジブルインタフェース Miniature Tangible Cube の実装

本項ではタンジブルインタフェース Miniature Tangible Cube の実装について説明する。本研究では Miniature Tangible Cube の表面に近接センサを導入することで接触を、内部に IMU を導入することで姿勢を認識し、これらのセンサデータから操縦者や環境、他の Miniature Tangible Cube からの接触を認識することができるタンジブルインタフェースを目指す。

4.5.1 Miniature Tangible Cube による実装の概要

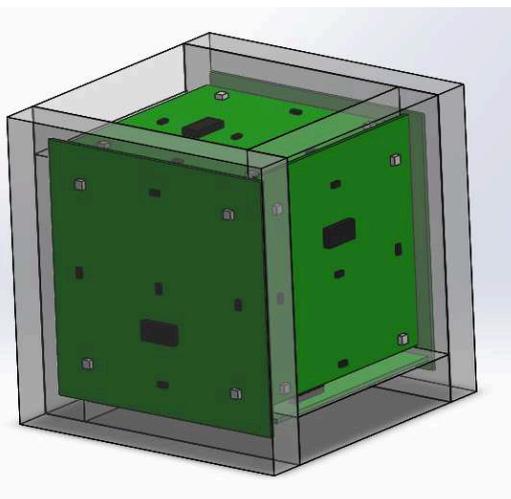
本研究で設計・作成した Miniature Tangible Cube について説明する。本研究で作成した Miniature Tangible Cube の外観を図 4.7 に示す。本研究では Miniature Tangible Cube を 2 種類の大きさを実装した。1 つは図 4.7a に示すデバイスであり各辺約 120 mm の立方体形状であるが、2 つめは図 4.7c に示すデバイスであり各辺約 100 mm の立方体形状と少し小さなデバイスとなっている。区別をする場合には前者を Miniature Tangible Cube Rev.1、後者を Miniature Tangible Cube Rev.2 と呼び、基本的サイズ以外にはセンサや機能、内部構成については大きな違いはないが、外形については Miniature Tangible Cube Rev.1 よりも小型であり、表面のシリコンゴムがユーザの接触によって回路基板から剥がれていくことを防ぐために各側面の基板はシリコンゴムが固定されるように切れ込みが入った手裏剣形状になっている。また Miniature Tangible Cube Rev.2 は小型化にともない、作成する際に必要となるシリコンゴムが少なくなったため軽量化にも成功している。

デバイスの内部には制御計算機と 9 軸 IMU, LiPo バッテリが配置されており (図 4.12), 全てのセンサ, LED, 制御計算機はこのバッテリーで駆動しており, 全てのセンサ情報は制御計算機を介して Wi-Fi を通じて遠隔地のロボットへ送信されている。デバイスの各面に近接センサを 5 つずつの計 30 個を表面に配置し (図 4.8a), これらはデバイス内部にある制御計算機と通信している (図 4.8b)。近接センサはデバイス各面に対して 5 つずつ十字上に設置されており, どの面が接触しているかだけでなく面上の位置も認識することができる。デバイス内部には制御計算機と 9 軸 IMU とリチウムポリマ電池 (図 4.12) が配置されており, デバイスの全てのセンサ, LED, 制御計算機はこの電池によって駆動している。また全てのセンサデータは制御計算機で処理を行ったのちに, 外部計算機とロボットに Wi-Fi で送信される。

Miniature Tangible Cube では IMU を用いてデバイスの姿勢を認識し, 各面 5 つの近接センサを用いてデバイス各面の接触状態と接触位置を認識する。さらに, 接触状態の表示用にデバイス各面にフルカラー LED を 4 つずつ配置し (図 4.8a), デバイスが接触を検知していることを操縦者に提示する。本研究では Miniature Tangible Cube Rev.1 を 1 台, Miniature Tangible Cube Rev.2 を 2 台の計 3 台を作成する。



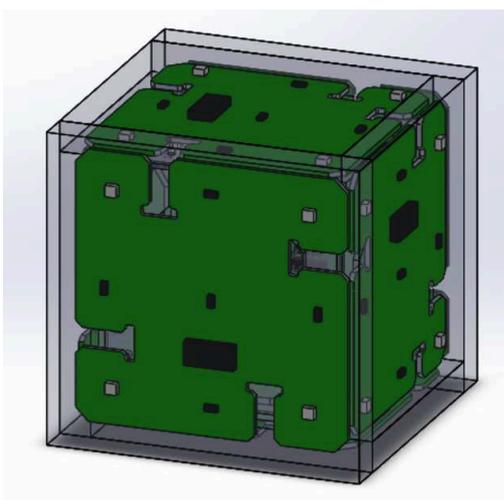
(a) Miniature Tangible Cube Rev.1 の外観図 [37]



(b) Miniature Tangible Cube Rev.1 の CAD 設計 [37]



(c) Miniature Tangible Cube Rev.2 の外観図



(d) Miniature Tangible Cube Rev.2 の CAD 設計

図 4.7: Miniature Tangible Cube の外観図と設計

4.5.2 近接センサによる Miniature Tangible Cube の接触計測

操縦者や環境との接触を認識するためのセンサとして近接センサを用いており、赤外光反射方式でセンサ付近 0 mm から 200 mm にある物体を非接触で検知する Vishay 社の VCNL4040 を使用する。赤外光反射方式とはセンサが物体に対して赤外光を照射し、物体から反射してきた赤外光の強度から物体の近接・接触を検知するものであり、赤外反射光の強度は物体の反射特性に応じて変化するものの近接から接触まで検知することができる。なお本研究ではこの近接センサを用いてタンジブルインタフェースを作成するため、環境・操縦者・他のタンジブル

表 4.1: Miniature Tangible Cube のハードウェアスペック [37]

Type	Product name
On-board computer	Raspberry Pi Zero WH
Proximity sensors	Vishay Intertechnology, VCNL4040
IMU	Bosch, BNO055
Silicone Rubber	Dow Inc., SYLGARD 184

インタフェースであり問題なく接触を検知することができる。これをシリコーンゴムモールドしてデバイス各面に配置することで、衝撃に強い接触センサとしてデバイスへの接触を認識することや、接触力の大きさをある程度認識することができ、この近接センサをゴムモールドして用いることで赤外光反射方式の変化を接触データとして計測することができる [90]。また近接センサを Miniature Tangible Cube の各面に 5 つ十字上に配置することで、各面のどの部分に接触していることが認識できる (図 4.8a)。そのためデバイス各面の中央、右、左、上、下の 5 点の接触を検出でき、デバイスへの接触状態のより詳しい認識に使うことができる。例えば、デバイスの底面の右側に接触が検出でき、左側にできない場合には、デバイスの左側がテーブルの端からでていることが認識できる。またデバイスの各面の裏側には I2C 通信と LED 通信のためのコネクタがデバイスの他面と制御コンピュータに接続されている。

シリコーンゴムモルディングは、図 4.9a に示すようなモールド型にデバイス各面を配置し、その上から低温で保存されたシリコーンゴムである DOW 社の SYLGARD 184 を流しこみ、減圧器内で十分に気泡を脱泡したあとにヒータで熱することで行う。このシリコーンゴムは透過性が高く衝撃を吸収するため、回路及び近接センサを保護しつつ近接センサの赤外光を透過させることができる。各面のシリコーンモルディング後の Miniature Tangible Cube のセンサ基板は図 4.9b に示す。なおシリコーンゴムモルディングの際にシリコーンゴムの中の気泡を十分に脱泡する必要があるため、また固まったシリコーンゴムが操縦者からの接触で動かないようにする必要もあるため、Miniature Tangible Cube Rev.2 の基板は四角形ではなく切れ込みが入った形に設計し、脱泡を効率的に行いかつシリコーンゴムが回路基板から動きにくくなるようにする。

近接センサデータは赤外反射光の強度であることから周囲の変化に対して安定ではないため、本研究では式 (4.2) に示すように、ある時刻 t の近接センサから取得されたセンサデータ $Prx_{raw}(t)$ に対して時系列的に重みをつけて平均値 $Prx_{average}(t)$ を取ることでセンサデータの安定化を行っている。

$$Prx_{average}(t) = 0.7 \cdot Prx_{average}(t-1) + 0.3 \cdot Prx_{raw}(t) \quad (4.2)$$

本研究で近接センサのセンサデータ値は式 (4.2) で計算された $Prx_{average}$ を用いる。

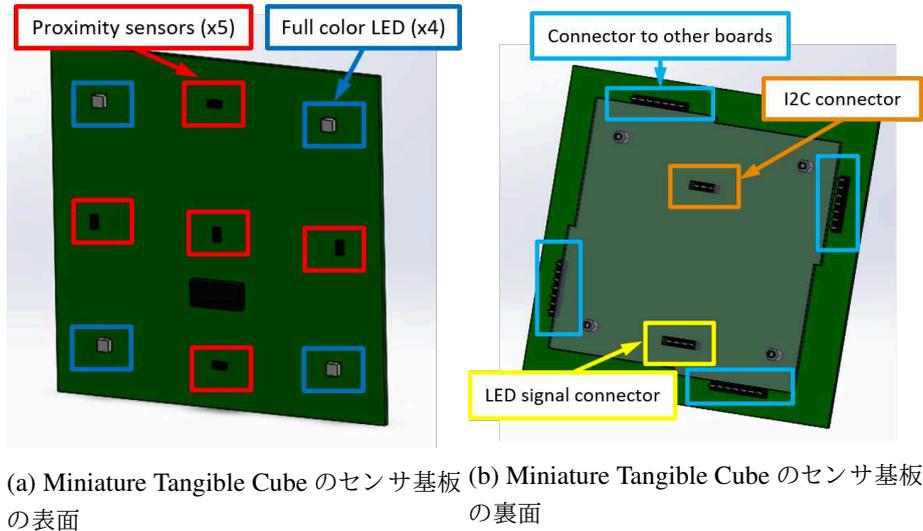


図 4.8: Miniature Tangible Cube の各面のセンサ基板とコネクタ [37]

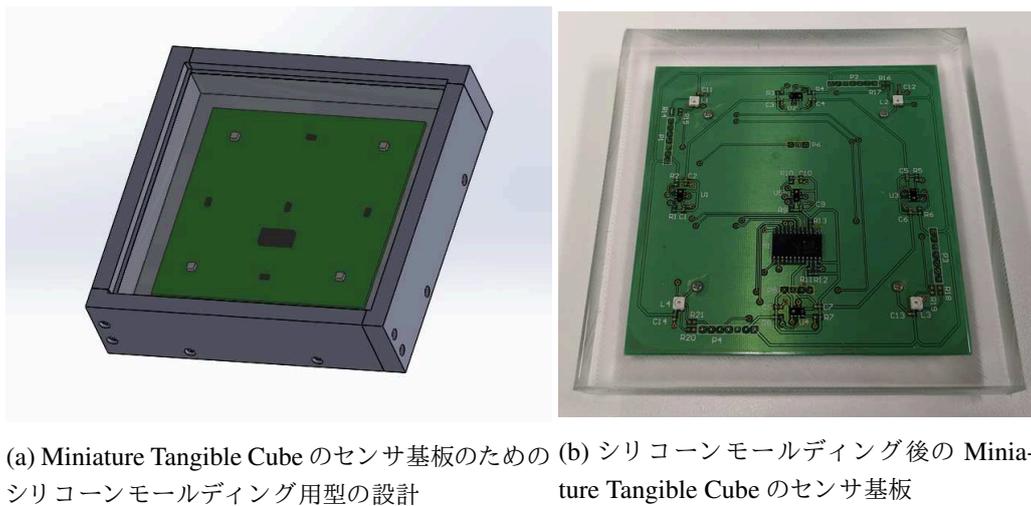


図 4.9: Miniature Tangible Cube のシリコンゴムモルディング [37]

4.5.3 IMU による Miniature Tangible Cube の回転計測

Miniature Tangible Cube では回転計測には IMU を用いており, Bosch 社の地磁気込み 9 DoF IMU の BNO055 が Miniature Tangible Cube の中央に配置されている. この IMU をは地磁気を用いることで重力方向のキャリブレーションを内部でおこなうことができるため, 回転をクォータニオン形式で I2C 通信で取得することができる. この回転を外部計算機に IMU データとして送信し, 外部計算機では IMU のセンサデータに対して Unscented Kalman Filter [91] を介してデバイスの姿勢を推定する. なお正面方向については起動時を正面方向と

してキャリブレーションされるが、現在推定されている回転を保存し、そこからの変位を計算することで起動後のキャリブレーションを可能にしており、GUI ボタンからもキャリブレーションを可能にしている。本研究では近接センサのセンサデータとこの推定した姿勢に基づいて 4.7.3 項で述べる動作プランニングを行っている。

4.5.4 Miniature Tangible Cube の回路構成

Miniature Tangible Cube の上下面の回路図は図 4.10、前後左右面の回路図は図 4.11 に示す。図 4.10、図 4.11 に示すように、各側面の回路基板は近接センサと LED と近接センサのための I2C マルチプレクサが配置されている。近接センサと LED は各 3.3V と 5V の別々の電源回路から電力を供給されている。Miniature Tangible Cube の制御計算機は各側面の個別の I2C アドレスを設定された I2C マルチプレクサに対して通信し、各側面の 5 つの近接センサと順番に通信することで、30 個の近接センサから 1 つずつシリアルに近接センサデータを取得する。LED については独自の通信規格に対応するために、制御計算機から制御通信回路が一筆書きの経路になるようにジャンパ抵抗を配置することで実装する。なお図 4.10、図 4.11 には接触センサの拡張性として NXP 社の気圧センサ MPL115A2 [92, 93] を各面 5 つずつとそのため I2C マルチプレクサを 1 つ配置できるように設計しているが本研究では使用していない。

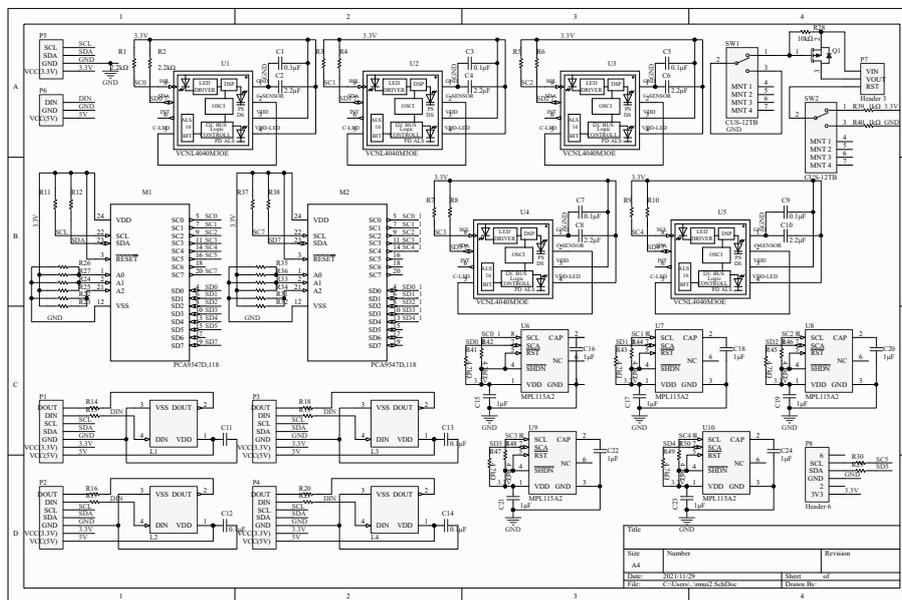


図 4.10: Miniature Tangible Cube Rev.2 の上下センサ基板の回路図

4.5.5 Miniature Tangible Cube の動力源と計算機

Miniature Tangible Cube の内部構造は図 4.12 に示すように、各面のボードがそれぞれ垂直にピンコネクタで繋がっている。またデバイス各面のセンサ基板の裏にはアクリル板が配置さ

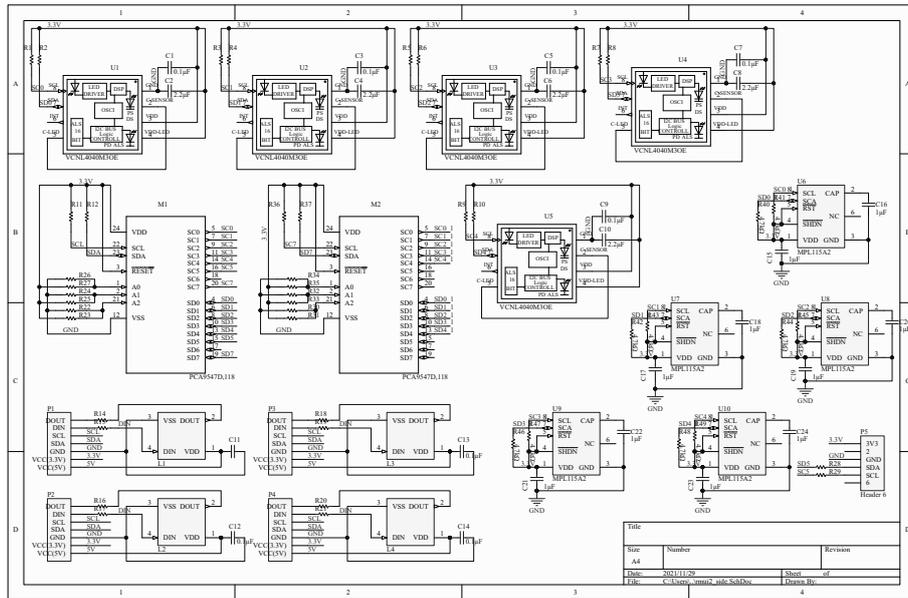
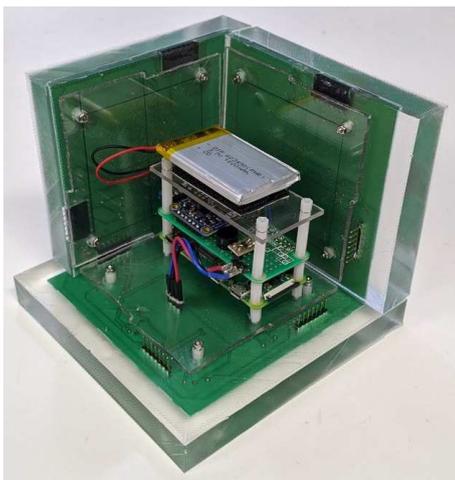
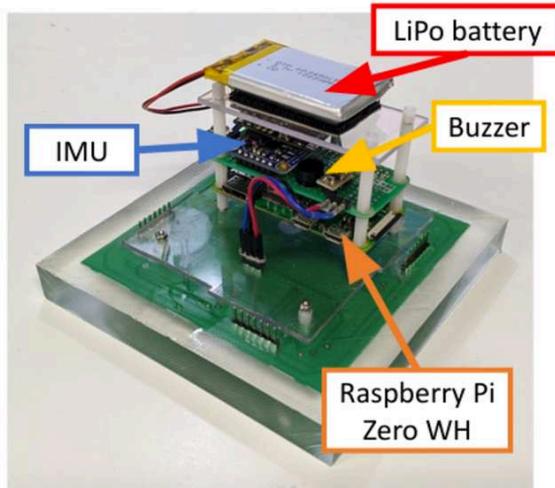


図 4.11: Miniature Tangible Cube Rev.2 の前後左右センサ基板の回路図

れており、これがデバイスの構造の補強を行っている。Miniature Tangible Cube の中央部には図 4.12b に示すように、計算機として Raspberry Pi Zero WH とデバイスの動力源として LiPo バッテリー、そして、Miniature Tangible Cube のセンサ・計算機を含む全ての動力はこの LiPo バッテリーで駆動しており、計算機は全てのセンサ・LED の制御を行い、外部計算機に ROS ネットワークを介してデータの通信を行っている。



(a) Miniature Tangible Cube の内部構造の外観



(b) Miniature Tangible Cube の内部の構成部品

図 4.12: Miniature Tangible Cube の内部構造 [37]

4.5.6 Miniature Tangible Cube のネットワーク構成とセンサデータ通信

Miniature Tangible Cube の通信については図 4.13 に示す。各近接センサは各センサボードの I2C マルチプレクサを介して、計算機と I2C でセンサデータの通信を行う。また IMU も I2C で計算機にセンサデータの通信を行う。そして I2C で通信したセンサデータを計算機で ROS メッセージとして、ロボットと外部計算機に無線ネットワークを介して送信される。またフルカラー LED はデバイス内部の計算機によって独自の通信規格で制御されている。

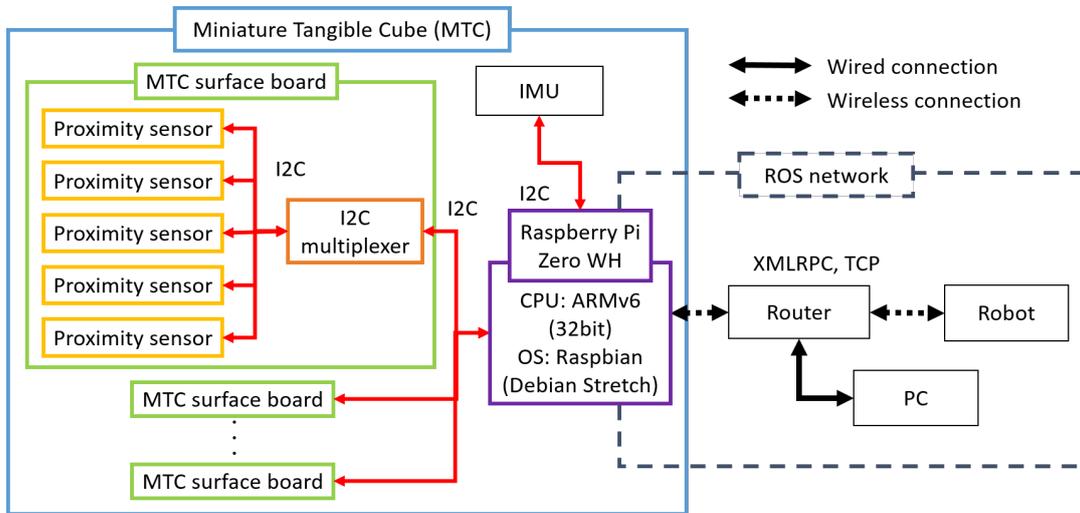


図 4.13: Miniature Tangible Cube のネットワーク構成とセンサデータ通信 [37]

また外部計算機ではデバイスの接触状態や姿勢の可視化も行っており、遠隔操縦時には操縦者はこの可視化された情報を見ることができる。

4.6 センサデータに基づくタンジブルインタフェースの接触・回転状態認識

本節では Miniature Tangible Cube のセンサデータに基づく接触・回転状態認識について説明する。Miniature Tangible Cube からロボットの操縦実行を行う際には、対象物と環境、ロボットの関係性として接触と回転に関するセンサデータに基づいて認識し、認識されたタンジブルインタフェース状態に基づいて動作計画を行う。4.5 節で説明したように、Miniature Tangible Cube は近接センサを用いて接触の計測では近接センサの物体に対する赤外反射光の強度を計測し、IMU を用いた回転の計測ではクォータニオン表記による回転を計測したが本節ではこれらのセンサデータを離散化しタンジブルインタフェース状態を回転状態と周囲との接触状態に記号化することでタンジブルインタフェース状態認識を行う。まず本研究で対象とする回転状態と周囲との接触状態について対象物とタンジブルインタフェースの両ケースについて定義した後に、これらの状態を近接センサと IMU のセンサデータを用いてどのように認識するか

について説明する。

4.6.1 対象物とタンジブルインタフェースの回転状態

本項では回転状態について説明する。本研究では回転状態について記号化して取り扱う。対象物の回転状態は IMU のセンサデータなどではクォータニオン表記で表現されるが、本研究では物体の回転を X 軸, Y 軸, Z 軸それぞれの軸に対する回転をそれぞれ 45° 単位で離散化することによって記号化し、本研究ではある対象物 O に対する上記の回転の記号化を $Rotation(O)$ と記述する。対象物の回転状態に対してもタンジブルインタフェースの回転状態同様に定義することができ、本研究では上記の回転状態の定義をタンジブルインタフェースと対象物の両方の場合で用い、ある対象物 O とそれに対応するタンジブルインタフェース D について以下の式 (4.3) が成立するような動作計画を行うことが対象物ユーザインタフェースによる操縦実行の目的となる。

$$Rotation(O) = Rotation(D) \quad (4.3)$$

なお異なる回転の記号による同じ回転状態の表現をさけるため、本研究では X 軸, Y 軸, Z 軸のどれか 1 つの回転軸についての回転操作を行う物体操作作業のみを取り扱う。

本研究において回転状態の記号化を 45° 単位で行う理由について説明する。まず回転状態をそもそもどうして記号化するのかについて説明する。回転状態を記号化する理由は 2 つある。1 つめはタンジブルインタフェースによる動作計画を行う際に 4.7 節において説明するように離散化されたタンジブルインタフェースの状態をノードとする有限状態機械を用いるためである。2 つめの理由としてタンジブルインタフェースによって動作順序を教示学習する際に、本研究では 4.2 節で説明した対象物の状態をノードとする動作順序グラフを学習するため、この動作順序グラフを形成するためには対象物の状態を離散化することでノードを形成する必要があるためである。次に回転状態の記号化を 45° 単位で行う理由について説明する。本研究では手順があり接触の多い物体操作作業に取り組むが、その中でも特に回転に関しては 4.9.2.3 目や 7.2 節で行うような双腕による大きな物体の回転動作の操縦実行を行う。このとき本研究で用いる PR2-JSK や Baxter-JSK-SoftHand では物体を双腕で同じ接触点のまま Z 軸 90° 回転させることは腕が箱もしくはロボット自身に衝突するなどのハードウェア構成上の理由から物理的に実行できない場合が発生する。そのため本研究では、物体を一度 45° 回転させたあとに接触点を接触が発生しないように変更したのちに再度 45° 回転させるという抱え直し動作 (Reholding) を物体の回転操作中に行うことで双腕による大きな物体の回転動作を実現する。この抱え直し動作を回転操作の途中に指示することができるために本研究では 45° 単位で回転状態を記号化する。

4.6.2 対象物とタンジブルインタフェースの周囲との接触状態

本項では周囲との接触状態に基づく関係性について説明する。本研究では周囲との接触状態に基づく関係性について記号化して取り扱う。まず対象物と環境とロボットの接触状態に関す

る関係性について説明する．複数の対象物がロボットの前にある場合に，ある1つの対象物 O の，ある他の対象物 O' と環境とロボットの関係性は，接触に着目すると以下のように記述することができる．

- 対象物と接触: 対象物のどの部分が他のある対象物のどの部分と接触しているか
- 環境との接触: 環境のどの部分 (壁, 床など) が対象物のどの部分が接触しているか
- ロボットとの接触: ロボットのどの部位 (右手, 左手など) に対象物のどの部分が接触しているか

上記の関係性は環境を E_r , ロボットを R , 接触部位を $Part$ で示すと以下のように記述できる．

- 対象物同士 O, O' の接触: $ObjContact(Part(O), Part(O'))$
- 環境 E_r と対象物 O の接触: $EnvContact(Part(E_r), Part(O))$
- ロボット R と対象物 O の接触: $RobotContact(Part(R), Part(O))$

次に上記の関係性に対応するタンジブルインタフェースと操縦環境と操縦者の接触状態に基づく関係性について説明する．Miniature Tangible Cube は前項で説明した対象物に関する関係性を操縦者側で再現することでロボットを操縦するものであり，対象物に関する関係性を，対象物ではなくタンジブルインタフェース，環境ではなく操縦環境，ロボットではなく操縦者に変換することで，同様のタンジブルインタフェースと環境と操縦者の関係性を記述することができる．つまり，複数のタンジブルインタフェースが操縦者の前にある場合に，ある1つのタンジブルインタフェース D の，ある他のタンジブルインタフェース D' と環境と操縦者との関係性は，接触に着目すると以下のように記述することができる．

- タンジブルインタフェースとの接触: タンジブルインタフェースのどの部分が他のあるタンジブルインタフェースのどの部分と接触しているか
- 操縦環境との接触: 操縦環境のどの部分 (壁, 床など) がタンジブルインタフェースのどの部分が接触しているか
- 操縦者との接触: 操縦者のどの部位 (右手, 左手など) にタンジブルインタフェースのどの部分が接触しているか

上記の関係性は操縦環境を E_p , 操縦者を D , 各接触部位を $Part$ で示すと以下のように記述できる．

- タンジブルインタフェース同士 D, D' の接触: $DevContact(Part(D), Part(D'))$
- 操縦環境 E_p とタンジブルインタフェース D の接触: $EnvContact(Part(E_p), Part(D))$
- 操縦者 P とタンジブルインタフェース D の接触: $PilotContact(Part(P), Part(D))$

よって上記の対象物の接触状態とタンジブルインタフェースの接触状態の定義から，あるタンジブルインタフェース D, D' とそれに対応する物体 O, O' に関して以下の式 (4.4), 式 (4.5), 式 (4.6) が成立するような動作計画を行うことが対象物ユーザインタフェースによる操縦実行

の目的となる.

$$ObjContact(Part(O), Part(O')) = DevContact(Part(D), Part(D')) \quad (4.4)$$

$$EnvContact(Part(E_r), Part(O)) = EnvContact(Part(E_p), Part(D)) \quad (4.5)$$

$$RobotContact(Part(R), Part(O)) = PilotContact(Part(P), Part(D)) \quad (4.6)$$

4.6.3 Miniature Tangible Cube の IMU に基づく 回転状態認識

本項では Miniature Tangible Cube の IMU データからどのように回転状態を認識するかについて説明する. 本研究では回転状態を X 軸, Y 軸, Z 軸それぞれの軸について 45° 単位で散化した記号によって表現し, X 軸, Y 軸, Z 軸の各回転軸に対して以下の 8 つの状態 (以下では X 軸の例を示す) を設定し, 各回転軸についてこのうちのどれであることを認識する.

- X0: 正面を向いている状態
- X45: 正面から X 軸 45° 回転した状態
- X90: 正面から X 軸 90° 回転した状態
- X135: 正面から X 軸 135° 回転した状態
- X180: 正面から X 軸 180° 回転した状態
- X-135: 正面から X 軸 -135° 回転した状態
- X-90: 正面から X 軸 -90° 回転した状態
- X-45: 正面から X 軸 -45° 回転した状態

X0 とは, Miniature Tangible Cube が正面を向いた場合の形状と同じ状態であり, X45, X90, X135, X180, X-135, X-90, X-45 とは Miniature Tangible Cube が X 軸を回転軸としてそれぞれ 45°, 90°, 135°, 180°, -135°, -90°, -45° 回転した状態を示す.

次に IMU のセンサデータに基づく姿勢推定からどのように上記の記号に落とし込むかについて説明する. なお異なる回転の記号による同じ回転状態の表現をさけるため, 本研究では X 軸, Y 軸, Z 軸のどれか 1 つの回転軸についての回転状態のみを取り扱う. 推定された姿勢 R からの X 軸の回転状態認識アルゴリズムを Algorithm 1 に示す. 推定された姿勢を用いて X 軸正負方向, Y 軸正負方向, Z 軸正負方向の 6 つの単位ベクトルを回転させ, それぞれの単位ベクトルと X 軸正方向, Y 軸正方向, Z 軸正方向の 3 つの単位ベクトルとのノルムを計算し, 以前の回転状態が X45, X135, X-135, X-45 であり, かつ全ノルムの値が閾値よりも小さかった場合と, 以前の回転状態が X0, X90, X180, X-90 であり, かつ X 軸正方向の単位ベクトルとのノルムの値が閾値よりも小さかった場合には以前の回転状態から 45° もしくは -45° 回転したものととして回転状態として認識する. このとき回転方向は回転状態のみでは一意に定まらないため, 接触状態の変化の合理性に基づいて回転方向を認識する. 例えばタンジブルインタフェースに対して左右に接触があった状態から Z 軸に対して 45° 回転した場合には接触は左前と右後になり, Z 軸に対して -45° 回転した場合には接触は左後と右前になる

ため、接触状態の変化をみることで回転方向を認識することができる。本研究では閾値として $threshold = 0.5$ を用いる。

Algorithm 1 Miniature Tangible Cube の X 軸の回転状態認識アルゴリズム

$Rotation(D)_{prev}$: Previous rotation state of Miniature Tangible Cube

R_{dev} : Estimated rotation of Miniature Tangible Cube

$threshold$: Threshold

$Rot(R, v)$: Rotate vector v with rotation R

$Rotate(Rotation(D), ax, angle)$: Get rotated $Rotation(D)$ by $angle$ degree in ax axis

$GetRotateDirection(Rotation(D))$: Get rotation direction from contact state change and return 1 or -1

$Norm(v1, v2)$: Calculate L2-norm of vector $v1$ and $v2$

procedure RECOGNIZEROTATIONSTATE($Rotation(D)_{prev}, R_{dev}, threshold$)

$Xp, Yp, Zp \leftarrow False$

$Vs \leftarrow [[1, 0, 0], [-1, 0, 0], [0, 1, 0], [0, -1, 0], [0, 0, 1], [0, 0, -1]]$

for $V \leftarrow Vs$ **do**

$V_{dev} \leftarrow Rot(R_{dev}, V)$

if $Norm(V_{dev}, [1, 0, 0]) < threshold$ **then**

$Xp \leftarrow True$

end if

if $Norm(V_{dev}, [0, 1, 0]) < threshold$ **then**

$Yp \leftarrow True$

end if

if $Norm(V_{dev}, [0, 0, 1]) < threshold$ **then**

$Zp \leftarrow True$

end if

end for

if (Xp and Yp and Zp and $Rotation(D)_{prev}$ in $[X45, X135, X-135, X-45]$) or (Xp and $Rotation(D)_{prev}$ in $[X0, X90, X180, X-90]$) **then**

Return $Rotate(Rotation(D)_{prev}, X, 45 * GetRotateDirection(Rotation(D)_{prev}))$

else

Return $Rotation(D)_{prev}$

end if

end procedure

4.6.4 Miniature Tangible Cube の近接センサに基づく接触状態認識

本項では Miniature Tangible Cube の近接センサのセンサデータからどのようにして接触状態を認識するかについて説明する。

まずセンサデータからどのように接触面認識を行うかについて説明する。接触面認識アルゴリズムは Algorithm 2 に示すように近接センサ値に閾値を設定し、Miniature Tangible Cube の上下左右前後の各 6 面について閾値を超えている近接データが 1 つでもあれば接触、なければ非接触状態と判定する。本研究では近接センサのセンサ値に対する接触認識の閾値を $threshold = 500$ と設定する。

Algorithm 2 Miniature Tangible Cube の接触面認識アルゴリズム

```

Sf: Surface of Miniature Tangible Cube
threshold: Threshold
GetProximities(Sf): Get proximity sensor data on surface Sf
procedure CONTACTSURFACESTATE(Sf, threshold)
  Contact  $\leftarrow$  False
  Prxs  $\leftarrow$  getProximities(Sf)
  for Prx  $\leftarrow$  Prxs do
    if Prx > threshold then
      Contact  $\leftarrow$  True
    end if
  end for
  Return Contact
end procedure

```

次に接触面認識からのタンジブルインタフェースの接触状態の記号化について説明する。本研究では接触状態を Miniature Tangible Cube の上下左右前後の各 6 面についてそれぞれ環境、操縦者、タンジブルインタフェースのどこに接触しているかどうかで記号化する。まず単一タンジブルインタフェースの場合について説明する。単一タンジブルインタフェースの場合には以下の仮定を置くことで接触状態を認識する。

- タンジブルインタフェースの開始時の底面の接触は床との接触である
- 接触面は操縦者であっても環境であっても 1 面もしくは正対する 2 面ずつしか増減しない
- 接触状態は一度に操縦者と環境のどちらかしか変化しない
- 操縦者は 2 面以上は接触しない
- 操縦者は右側の接触は右手、左側の接触は左手で行う

以上の仮定のもとで、接触が起こった面について環境か操縦者かを判定し、接触状態として認

識する．なお環境状況によっては上記の仮説のみでは接触状態を一意に判定できない場合が存在するが，そういった場合には作業に応じて接触状態認識の判定が一意になるように新たな仮定を置くことで解決する．

次に複数タンジブルインタフェースの場合について説明する．単一タンジブルインタフェースの場合と異なり複数のタンジブルインタフェースが存在する場合には，タンジブルインタフェース同士が接触すると複数のタンジブルインタフェースの接触面が2面同時に増加し，タンジブルインタフェース同士が離されると接触面が2面同時に減少する．よって複数タンジブルインタフェースの場合には以下の仮定を置くことで接触状態を認識する．

- タンジブルインタフェースの開始時の底面の接触は床との接触である
- 接触面が他の接触タンジブルインタフェースと同時に各1面増減する場合には，そのタンジブルインタフェースとの接触である
- 接触面は操縦者と環境の場合には1面もしくは正対する2面ずつしか増減しない
- 接触状態は一度に操縦者と環境のどちらかしか変化しない
- 操縦者は2面以上は接触しない
- 操縦者は右側の接触は右手，左側の接触は左手で行う

以上の仮定のもとで，接触が起こった面について環境か操縦者かを判定し，接触状態として認識する．なお複数タンジブルインタフェースの場合も同様に，環境状況によっては上記の仮説のみでは接触状態を一意に判定できない場合が存在するが，そういった場合には作業に応じて接触状態認識の判定が一意になるように新たな仮定を置くことで解決する．

4.6.5 Miniature Tangible Cube のセンサデータに基づくタンジブルインタフェース状態認識

本項では，これまで説明した回転状態認識と接触状態認識をまとめたタンジブルインタフェース状態認識について説明する．タンジブルインタフェース状態とはこれまで説明してきた回転状態認識と接触状態の組み合わせで表せるものとなるが，この組み合わせられた全てのタンジブルインタフェース状態が実現可能であるとは限らない．本研究では実現可能であるタンジブルインタフェース状態間について動作を設定することで ObjectStateNet を構築し，これを Miniature Tangible Cube の動作計画に用いる．

4.7 タンジブルインタフェースの ObjectStateNet を用いた動作計画

本節ではタンジブルインタフェース Miniature Tangible Cube と 4.7.1 項で説明する ObjectStateNet を用いた動作計画について説明する．本研究ではタンジブルインタフェースを用いて対象物の状態をタンジブルインタフェースに再現するため，タンジブルインタフェース状態を認識することでその状態を対象物で再現するようにロボットが動くような動作計画を行う．こ

のとき ObjectStateNet を用いることで対象物の状態を中心として動作を実行・記述できる, この2つを用いることで複数の物体や環境が接触し合うような複雑な物体操作作業の操縦実行が可能となる. まず ObjectStateNet とそのエッジを構成する動作集合について説明した後に, それを用いた ObjectStateNet による動作計画について説明する.

4.7.1 ObjectStateNet の定義

本項では ObjectStateNet について説明する. まず StateNet について説明したのちに ObjectStateNet について説明する.

StateNet [94] とは図 4.14a に示すような各ノードがロボットや対象物, 環境の状態を表し各エッジが動作を表す有限状態機械のことである. 一般的にはロボットの動作には時間がかかり, 各エッジの動作実行時間は無視できないため, 図 4.14b に示すような動作実行状態を追加した有限状態機械がより詳細ではあるが, 煩雑であるため本研究では図 4.14a の形式で StateNet を記述する. StateNet は有限状態集合 S , 動作集合 A , 状態遷移関数 $T: S \times A \rightarrow S$, 初期状態 s_0 , 終了状態集合 F の組 $M = \{S, A, T, s_0, F\}$ である. これは有限状態機械の一種であると解釈できるが, 一般的な有限状態機械の場合には「移動する」などの動作をしている状態を S とし, 「ものをみつける」などの入力信号集合を A とするが, StateNet はロボットや対象物, 環境の状態集合を S とし動作集合を A とする. StateNet はこのように表現することで, 複雑な環境に対して順序がある複雑な動作を実行することができる.

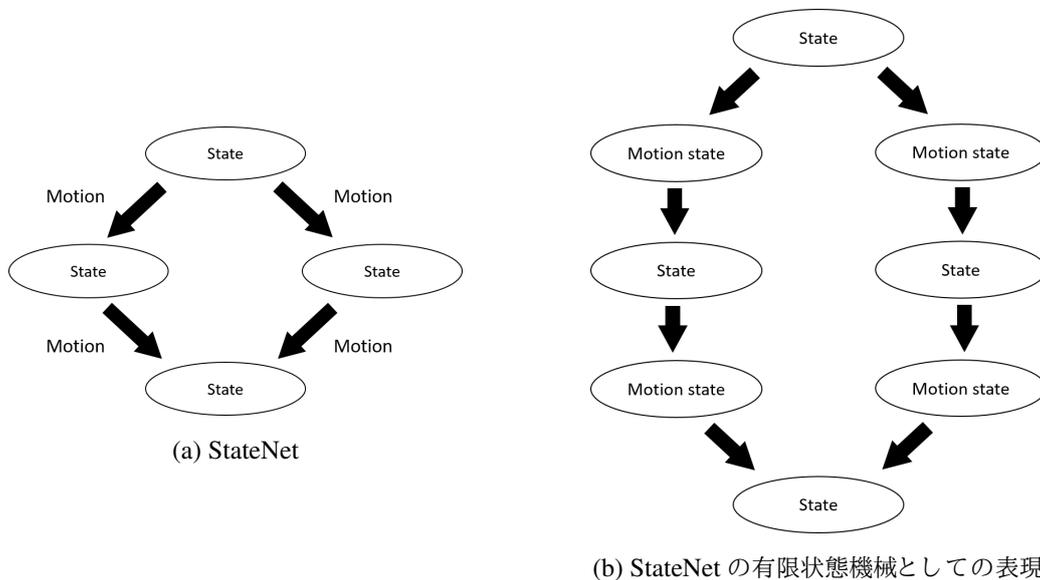


図 4.14: StateNet の構成: StateNet とは各ノードが状態を表し各エッジが動作を表す有限状態機械であり, 本研究では (a) のように記述する.

これまで StateNet は図 4.15a に示すようにロボットの姿勢を有限状態集合 S として, ヒューマノイドの転倒復帰制御 [94-96] などに用いられてきたが, 本研究では物体操作作業につい

て有限状態集合 S の観点から拡張し，図 4.15b に示すような対象物の接触・回転状態の集合 S^{obj} を有限状態集合とする ObjectStateNet を導入する．ObjectStateNet とは StateNet [94] の状態をロボットの姿勢などのロボットの状態ではなく対象物の接触や回転などの対象物の状態として記述したものであり，各ノードが対象物の回転や接触など対象物の状態で記述されており各ノードは動作エッジで接続されている．この時，対象物の ObjectStateNet は対象物の接触・回転状態の集合 S^{dev} ，動作集合 A ，状態遷移関数 $T^{obj} : S^{obj} \times A \rightarrow S^{obj}$ ，初期状態 s_0^{obj} ，終了状態集合 F の組 $M^{obj} = \{S^{obj}, A, T, s_0^{obj}, F\}$ と定義することができる．また ObjectStateNet は 4.2 節で説明した動作順序を表す動作順序グラフとノードとエッジの構成は同じであるが，動作順序グラフとは異なり有限状態機械であるため，本研究では対象物の接触・回転状態に基づく状態遷移関数 $T^{obj} : S^{obj} \times A \rightarrow S^{obj}$ を有している．

本研究では対象物の状態をタンジブルインタフェースを用いて操縦者が再現するため，ObjectStateNet の状態をタンジブルインタフェースの状態に変換することで，タンジブルインタフェースの ObjectStateNet も同様にタンジブルインタフェースの接触・回転状態の集合 S^{dev} ，動作集合 A ，状態遷移関数 $T^{dev} : S^{dev} \times A \rightarrow S^{dev}$ ，初期状態 s_0^{dev} ，終了状態集合 F の組 $M^{dev} = \{S^{dev}, A, T, s_0^{dev}, F\}$ と定義することができる．この時，対象物の ObjectStateNet とタンジブルインタフェースの ObjectStateNet には式 (4.7) の関係が成立するような対象物とタンジブルインタフェースに共通する状態遷移関数 T' が定義でき，これはタンジブルインタフェースが対象物の状態を表現していることを表す．

$$\begin{aligned} T^{dev} &= T'(S^{dev}) \\ T^{obj} &= T'(S^{obj}) \end{aligned} \quad (4.7)$$

タンジブルインタフェースは対象物の状態を表現しているため，タンジブルインタフェースの ObjectStateNet での状態遷移関数 T^{obj} は対象物の ObjectStateNet での状態遷移関数 T^{dev} に対応しており，この ObjectStateNet を用いて操縦実行を行うことで式 (4.3) と式 (4.4)，式 (4.5)，式 (4.6) を成立させながら操縦実行を行うことができる．

なお，これまでの研究で用いられていた StateNet はロボット・対象物の状態をノードとした StateNet であるため RobotStateNet (図 4.15a) と区別し，以降では StateNet は ObjectStateNet や RoboStateNet など全てを含んだ，状態をノードとして動作をエッジとした有限状態機械のことを示す．次に本研究で設計する ObjectStateNet の動作集合 A について説明する．

4.7.2 ObjectStateNet におけるエッジとなる動作集合の設定

本項では 4.7.1 項で説明した ObjectStateNet を構成するために，エッジとなる動作集合 A について設定する．そのためにはまず本研究で取り扱う物体操作動作を設定し，これらを分解することで ObjectStateNet のエッジとなる動作集合 A を設定する．本研究で対象とする物体操作動作は以下の通りである．

- 物体抱え込み動作: 対象物を抱え込む
- 物体持ち上げ動作: 対象物を持ち上げる

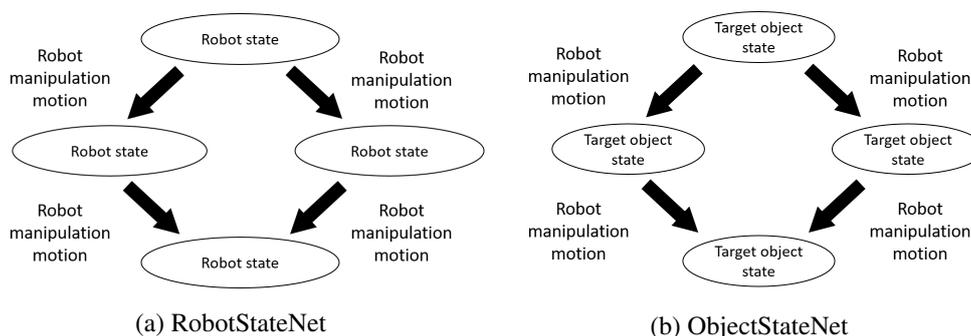


図 4.15: RobotStateNet と ObjectStateNet: 先行研究 [94–96] で用いられていた StateNet はロボットの姿勢などロボットの状態をノードとしており, 本研究ではこれを RobotStateNet と呼ぶ. 対象物の状態をノードとする ObjectStateNet を用いることで対象物を中心とした形で StateNet を記述する.

- 物体配置動作: 持ち上げた対象物を配置する
- 物体回転動作: 対象物を回転する
- 物体詰め寄せ動作: 2つの物体を隣接して配置する
- 物体積み重ね動作: 2つの物体を重畳して配置する
- 物体把持動作: 物体を把持する

そして, これらの操作を接触や姿勢について着目して分割すると, 以下の動作の組み合わせで実現できると考えられる.

- 接触: 対象物に接触する
- リリース: 対象物との接触をやめる, 離す
- 回転: 抱え込んだ対象物を回転する
- 持ち上げ: 抱え込んだ対象物を持ち上げる
- 配置: 抱え込んだ対象物を配置する
- 押し移動: 接触した状態で対象物を押して移動する
- 把持: 対象物を把持する
- 把持リリース: 対象物への把持をやめる, 開放する

ここで抱え込みとは2点以上の接触によって物体が双腕などのロボット身体で固定されている状態を指し示すため, 物体抱え込み動作は両腕による接触によって実現可能である. 同様に, 物体持ち上げ動作は物体抱え込み動作のあとに持ち上げを行うことで, 物体配置動作は持ち上げ動作のあとに配置を行うことで, 物体回転動作は抱え込み動作のあとに回転を実行することで実現できる. また物体詰め寄せ動作は物体に接触したあとに他の物体の隣まで押し移動動作を行うことで, 物体積み重ね動作は物体を持ち上げたあとに他の物体の上に配置することで実現できる.

次にこれらの動作集合は **Miniature Tangible Cube** のセンサデータから認識することができることを示す。接触とリリースは、タンジブルインタフェースの各面に配置されている近接センサのデータから認識でき、回転はタンジブルインタフェース内部の **IMU** のセンサデータから認識できる。持ち上げと配置についてはタンジブルインタフェース底面の接触を検知することで認識でき、対象物を抱え込んでいるときにタンジブルインタフェース底面の接触が検知できないときにロボットは対象物を持ち上げる。対象物を持ち上げているときにタンジブルインタフェース底面の接触が検知されたときには、他のタンジブルインタフェースの上面に接触が検知された場合にはそのタンジブルインタフェースに対応する対象物のうえに持ち上げている対象物をロボットは配置し、そうでない場合にはロボットは対象物を床の上に配置する。押し移動についてはタンジブルインタフェースの面が接触をしながら他のタンジブルインタフェースの面との接触があった場合には、タンジブルインタフェースが押されて移動したとして他のタンジブルインタフェースの隣まで移動したとして、ロボットは対象物を押して他の対象物と隣り合うように配置する。把持動作についてはタンジブルインタフェースの上面に接触があった場合にはロボットは対象物を上から把持する。把持リリースについてはタンジブルインタフェースの上面の接触を離れた場合には、ロボットは対象物への把持を開放する。具体的な動作計画については 4.7.3 項に示す。以上より本研究では上記の接触、リリース、回転、持ち上げ、配置、押し移動、把持、把持リリースを **ObjectStateNet** のエッジとなる動作集合 A とする。

4.7.3 ObjectStateNet を用いた動作計画

本項では 4.7.2 項で設定した動作集合 A をエッジとして用いた **ObjectStateNet** とそれを用いた動作計画について説明する。4.7.2 項で規定した動作集合 A を用いた物体操作の **ObjectStateNet** を図 4.16 に示す。本研究ではこの **ObjectStateNet** を用いてタンジブルインタフェースの接触・回転状態をノードとして、以前のタンジブルインタフェースの状態ノード $s_{prev}^{dev} \in S^{dev}$ から現在タンジブルインタフェースの目標状態ノード $s_{now}^{dev} \in S^{dev}$ をつなぐエッジである動作 $a \in A$ を選ぶことで動作選択し、選択した動作を実行することで物体操作作業を実現する。この動作選択は状態遷移関数 T_{dev} を用いて式 (4.8) のように定式化することができる。

$$\arg_{a \in A} T_{dev}(s_{prev}^{dev}, a) = s_{now}^{dev} \quad (4.8)$$

4.7.1 項で説明したように対象物の **ObjectStateNet** とタンジブルインタフェースの **ObjectStateNet** について共通の状態遷移関数 T' が定義できるため、タンジブルインタフェースの **ObjectStateNet** を用いて経路計画を行うことは対象物の **ObjectStateNet** を用いた経路計画と同じであり、タンジブルインタフェースを用いた操縦実行のための動作計画を行うことができる。このときタンジブルインタフェースの状態認識には 4.6 節に示した接触・回転状態認識をもちいることで複数物体の重なりや隣りあいを認識することができ、操縦者がタンジブルインタフェースを操作することでこのタンジブルインタフェースの接触・回転状態の遷移を認識し、その遷移に対応する動作を動作集合 A 内から選択・実行することで操縦実行を実現する。

動作集合 A 内から選択された動作について、それぞれ予めプログラムされた動作として逆運動学を解くことで実行すべき関節角度列を求め、これをロボットに送信・実行することで動作を実行できる。なお持ち上げ・回転・配置などの動作は、ロボットが物体を抱え込むことで物体が両腕の間に固定していることが実行条件として設定されているため、指令した動作を元に各動作が実行可能かどうかを検査する。

なお本研究の操縦実行で用いる ObjectStateNet はタンジブルインタフェースと対象物の共通の状態遷移関数 T' の条件を対象物もしくはタンジブルインタフェースの状態 s^{obj}, s^{dev} の変化に応じて記述することで実装する。この実装手法は複数の物体の状態や位置の組み合わせによって考慮すべき有限状態集合 S^{obj}, S^{dev} の全状態数が増える場合に対しても状態遷移関数 T' は対象物もしくはタンジブルインタフェース状態 s^{obj}, s^{dev} の変化について記述するため対応できるが、状態遷移関数 T' を用いた動作計画で選択された動作 $a \in A$ が実際に実行可能であるかは保証されない。そのため本研究では作業に応じて操作実行で実行された動作列を、実行可能な動順序として教示学習する。

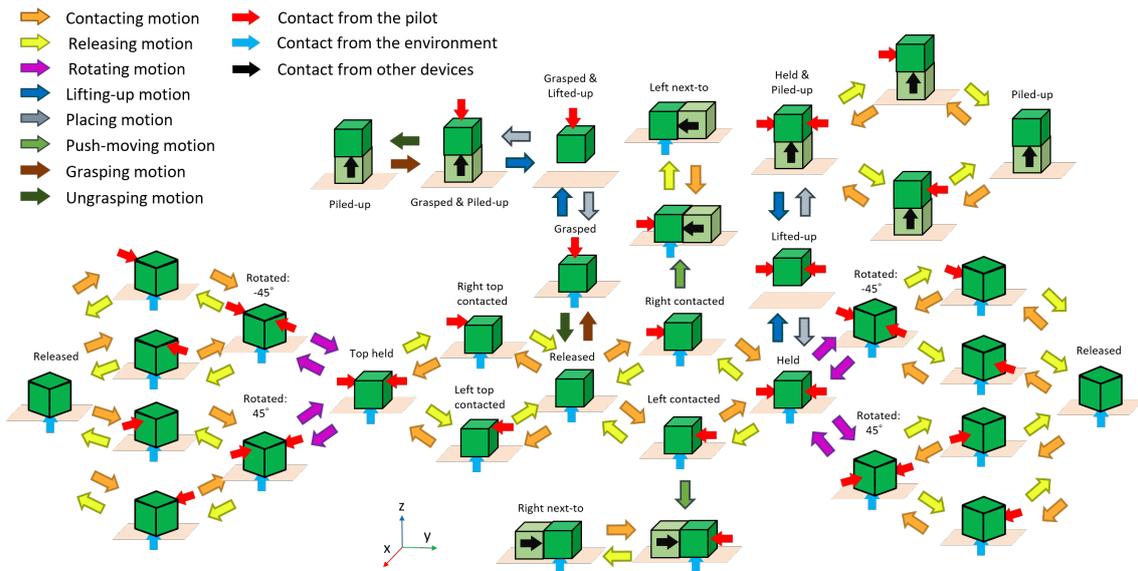


図 4.16: タンジブルインタフェースによる 操縦実行のための ObjectStateNet

4.8 タンジブルインタフェースによる動作順序の教示学習と自律実行

ここまでタンジブルインタフェースを用いた物体操作作業の操縦実行について説明してきた。本節ではタンジブルインタフェースの操縦実行に基づく動作順序の教示学習について説明する。本研究では図 4.17 に示すように 4.7 節で説明したタンジブルインタフェースによる操縦実行のための ObjectStateNet の実行経路を、動作順序の記述と記述し、保存することで動作順序の教示学習を実現する。このとき実行経路には ObjectStateNet のどの状態を遷移したか

と，遷移の際にどの動作を動作集合 A から選択したかを記述していき，本研究ではこれを対象物の接触・回転状態をノードとし動作をエッジとする有向グラフとして教示学習する．この有向グラフは作業を行うための実行可能な動作の順序とその動作によって対象物の状態がどのように変化するかを保存しており，本研究ではこの有向グラフを動作順序グラフと呼び，これを自律実行に共有することで作業の自律実行を実現する．

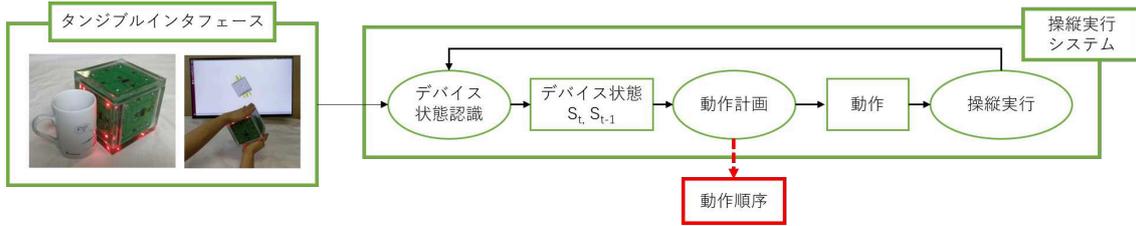


図 4.17: タンジブルインタフェースによる動作順序の教示学習

4.8.1 タンジブルインタフェースによる動作順序の教示学習

本項ではタンジブルインタフェースによる動作順序の教示学習について説明する．本研究では 4.6 節で説明したタンジブルインタフェースの接触・回転状態認識と 4.7 節で説明した動作計画によって，操縦者からの対象物への接触・回転状態変更を認識し，その状態変更を行うことができる動作を実行することでタンジブルインタフェースによる操縦実行を実現している．このとき t 回目の動作を $a_t \in A$ ， t 回目の動作を行ったあとの対象物の接触・回転状態 $s_t^{obj} \in S^{obj}$ とすると， t 回目の動作 a_t は対象物の接触・回転状態を s_{t-1}^{obj} から s_t^{obj} に変更していることになり，これは式 (4.9) に示すように a_t は s_{t-1}^{obj} を表すノードから s_t^{obj} を表すノードへの遷移を表すエッジであるといえる．

$$a_t = \overline{s_{t-1}^{obj} s_t^{obj}} \quad (4.9)$$

よって t 回目の動作のあとの動作順序グラフ $G_t = (V_t, E_t)$ とするとタンジブルインタフェースによる動作順序の学習は式 (4.10) のように記述することができる．

$$\begin{aligned} V_t &= V_{t-1} \cup \{a_t\} \\ E_t &= E_{t-1} \cup \{s_{t-1}^{obj}, s_t^{obj}\} \end{aligned} \quad (4.10)$$

以上より本研究で提案するタンジブルインタフェースによる動作順序グラフの教示学習は 4.7 節で説明した動作計画を含めて Algorithm 3 のように表すことができる．まずタンジブルインタフェースの接触・回転状態 s_t^{dev} をセンサデータから認識したのちに，1 ステップ前のタンジブルインタフェースの接触・回転状態 s_{t-1}^{dev} と現在のタンジブルインタフェースの接触・回転状態 s_t^{dev} の遷移示すエッジを動作 a_t を探索することで動作計画を行い，探索された動作 a_t を実行する．そして 1 ステップ前の対象物の接触・回転状態 s_{t-1}^{obj} と動作が行われたあとの対象物の接触・回転状態 s_t^{obj} ，動作 a_t を動作順序グラフ $G = (V, E)$ に追加することでタンジブルインタフェースを用いた操縦実行と動作順序の教示学習を実現する．

Algorithm 3 タンジブルインタフェースによる動作順序の教示学習アルゴリズム

V : Vertices of motion order graph G
 E : Edges of motion order graph G
 A : Robot manipulation actions
 s_0^{obj} : Initial contact and rotation state of target objects
 s_0^{dev} : Initial contact and rotation state of tangible interfaces
 $RecognizeDevice(s^{dev})$: Recognize contact and rotation state of tangible interfaces
 $ExecuteRobot(a)$: Execute robot manipulation action a
procedure LEARNMOTIONORDERGRAPH($V, E, A, s_0^{obj}, s_0^{dev}$)
 $t \leftarrow 0$
while maneuvering **do**
 $t \leftarrow t + 1$
 $s_t^{dev} \leftarrow RecognizeDevice(s_{t-1}^{dev})$
 $a_t \leftarrow \arg \min_{a \in A} T_{dev}(s_{t-1}^{dev}, a) = s_t^{dev}$
 $ExecuteRobot(a_t)$
 $s_t^{obj} \leftarrow T^{obj}(s_{t-1}^{obj}, a_t)$
 $V \leftarrow V \cup \{a_t\}$
 $E \leftarrow E \cup \{s_{t-1}^{obj}, s_t^{obj}\}$
end while
Return V, E
end procedure

4.8.2 教示学習された動作順序による作業の自律実行

本項では教示学習された動作順序グラフによる作業の自律実行について説明する．動作順序グラフを用いた作業の自律実行では，式 (4.11) に示すように動作順序グラフ $G = (V, E)$ 上で現在の対象物の状態 $s_{now}^{obj} \in S^{obj}$ から作業目標となる対象物の状態 $s_{goal}^{obj} \in S^{obj}$ への経路計画を行い，その経路上に存在するエッジ列である動作列 $\{a_0 \dots a_n \mid a_i \in A\}$ を順に実行することで作業の自律実行を実現する．

$$\begin{aligned}
 & \underset{\{a_0 \dots a_n \mid a_i \in A\}}{\operatorname{argmin}} && n \\
 & \text{subject to} && \operatorname{walk}(s_{now}^{obj}, \{a_0 \dots a_n\}, G) = s_{goal}^{obj}
 \end{aligned} \tag{4.11}$$

なおこのウォーク関数 $\operatorname{walk}(s, \{a_0 \dots a_n\}, G)$ は動作順序グラフ G 上においてノード s からエッジ列 $\{a_0 \dots a_n\}$ を順に遷移して到達するノードを返す関数である．以上より本研究で提案する教示学習された動作順序グラフによる作業の自律実行は Algorithm 4 のように表すことができる．まず現在の対象物の接触・回転状態 s_0^{obj} から目標の対象物の接触・回転状態 s_{goal}^{obj} への最短経路を計画し，その経路上にあるエッジ列である動作列 $\{a_0 \dots a_n\}$ を順に実

行することで教示された動作順序による作業の自律実行を実現する。なお Algorithm 4 における経路探索関数 $SolvePath(s, s', G)$ は式 (4.11) を解き経路上にあるエッジである動作列 $\{a_0 \dots a_n \mid a_i \in A\}$ 返す関数を表す。

Algorithm 4 動作順序による作業の自律実行アルゴリズム

G : Motion order graph

s_0^{obj} : Initial contact and rotation state of target objects

s_{goal}^{obj} : Goal contact and rotation state of target objects

$SolvePath(s, s', G)$: Solve path from node s to node s' on graph G and return edge set $\{a_0 \dots a_n\}$

$ExecuteRobot(a)$: Execute robot manipulation action a

procedure EXECUTEMOTIONORDERGRAPH($s_0^{obj}, s_{goal}^{obj}, G$)

$\{a_0 \dots a_n\} \leftarrow SolvePath(s_0^{obj}, s_{goal}^{obj}, G)$

for $a_i \leftarrow \{a_0 \dots a_n\}$ **do**

$ExecuteRobot(a_i)$

end for

end procedure

4.9 評価実験

本節では本研究で作成した Miniature Tangible Cube について、双腕移動ロボット PR2-JSK と双腕ロボット Baxter-JSK-SoftHand を用いた物体操作作業実験を通して評価する。

4.9.1 Miniature Tangible Cube の状態認識実験

本項では本研究で作成したタンジブルインタフェースである Miniature Tangible Cube の接触と回転認識、およびそのセンサデータの通信が正しく行われているかを確認した。

4.9.1.1 Miniature Tangible Cube における接触および回転認識

本研究で作成した Miniature Tangible Cube の動作実験を図 4.18 に示す。Miniature Tangible Cube の後ろのディスプレイには Miniature Tangible Cube の姿勢が灰色の立方体、物理接触が赤色と黄色の矢印で可視化されている。図 4.18 に示すように Miniature Tangible Cube と机、Miniature Tangible Cube と操縦者の手の接触状態が認識されており、持ち上げられたタンジブルインタフェースの姿勢変化も接触状態と同様に認識されていることがわかる。よって作成したタンジブルインタフェースを用いて対象物の姿勢と接触状態を表現できることを確認した。

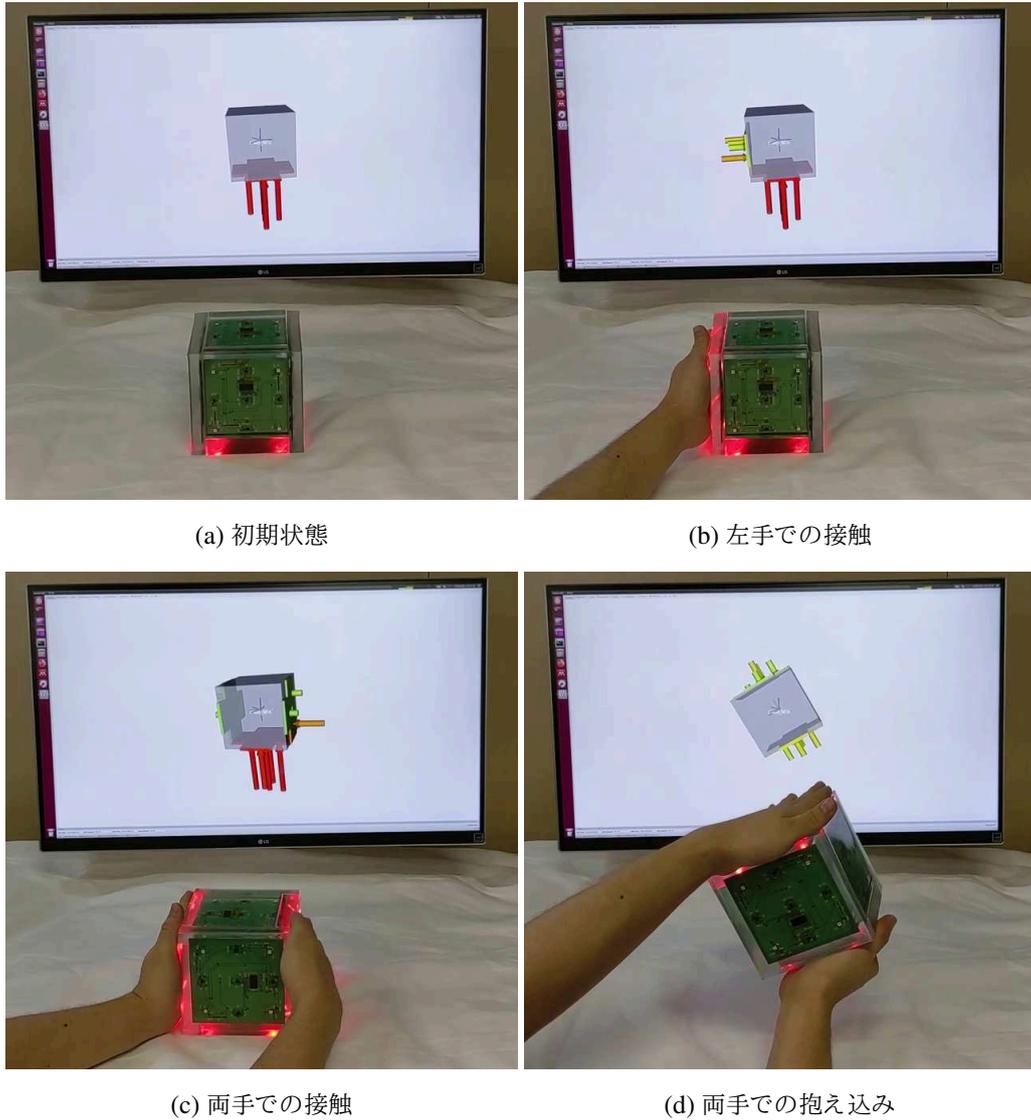


図 4.18: Miniature Tangible Cube における接触および回転認識実験 [37]

4.9.2 Miniature Tangible Cube による単一物体の操作作業の操縦実行実験

本項では本研究で作成したタンジブルインタフェースである Miniature Tangible Cube を用いた単一対象物に対する物体操作作業の操縦実行実験について説明する。本実験では単一対象物を対象とするため 1 台の Miniature Tangible Cube Rev.1 と 双腕移動ロボット PR2-JSK を用いて箱に対する箱操作作業を行った。本実験では対象物として図 4.19 に示す幅 400 mm, 奥行き 270 mm, 高さ 200 mm の布製の箱を用いた。



図 4.19: 単一物体の操作作業の操縦実行実験で用いる布製の箱

4.9.2.1 Miniature Tangible Cube による箱持ち上げ作業実験

本目では Miniature Tangible Cube による箱持ち上げ作業実験について説明する．本実験では図 4.19 に示す布製の箱を対象物として用いた．本研究で行った Miniature Tangible Cube による箱の持ち上げ作業実験について図 4.20 に示す．図 4.20 に示すように，Miniature Tangible Cube に対して左右に接触することで PR2-JSK が箱を抱え込み (図 4.20a, 図 4.20b)，抱え込んだ状態でタンジブルインタフェースを持ち上げることで PR2-JSK が箱を持ち上げ (図 4.20c)，タンジブルインタフェースを机に置くことで PR2-JSK が箱を机の上に置くことが出来た (図 4.20d)．

4.9.2.2 Miniature Tangible Cube による箱傾け作業実験

本目では Miniature Tangible Cube による箱傾け作業実験について説明する．本実験では図 4.19 に示す布製の箱を対象物として用いた．本研究で行った Miniature Tangible Cube による箱の箱傾け作業実験について図 4.21, 図 4.22 に示す．図 4.21, 図 4.22 に示すように Miniature Tangible Cube に対して左右から接触することで PR2-JSK が箱を抱え込み (図 4.21a, 図 4.22a)，タンジブルインタフェースを X 軸に 45° もしくは -45° に回転させることで PR2-JSK が箱を抱え込んで回転させることに成功した (図 4.21b, 図 4.22b)．

4.9.2.3 Miniature Tangible Cube による箱持ち替え回転作業実験

本目では Miniature Tangible Cube による箱持ち替え回転作業実験について説明する．本実験では図 4.19 に示す布製の箱を対象物として用いた．本研究で行った Miniature Tangible Cube による箱持ち替え回転作業実験について図 4.23, 図 4.24, に示す．図 4.23, 図 4.24 に示すように，Miniature Tangible Cube に対して左右から接触することで PR2-JSK が箱を抱え込み (図 4.23a, 図 4.24a)，タンジブルインタフェースを Z 軸 45° もしくは -45° に回転させることで PR2-JSK が箱を抱え込んだまま回転させ (図 4.23b, 図 4.24b)，タンジブルインタフェース

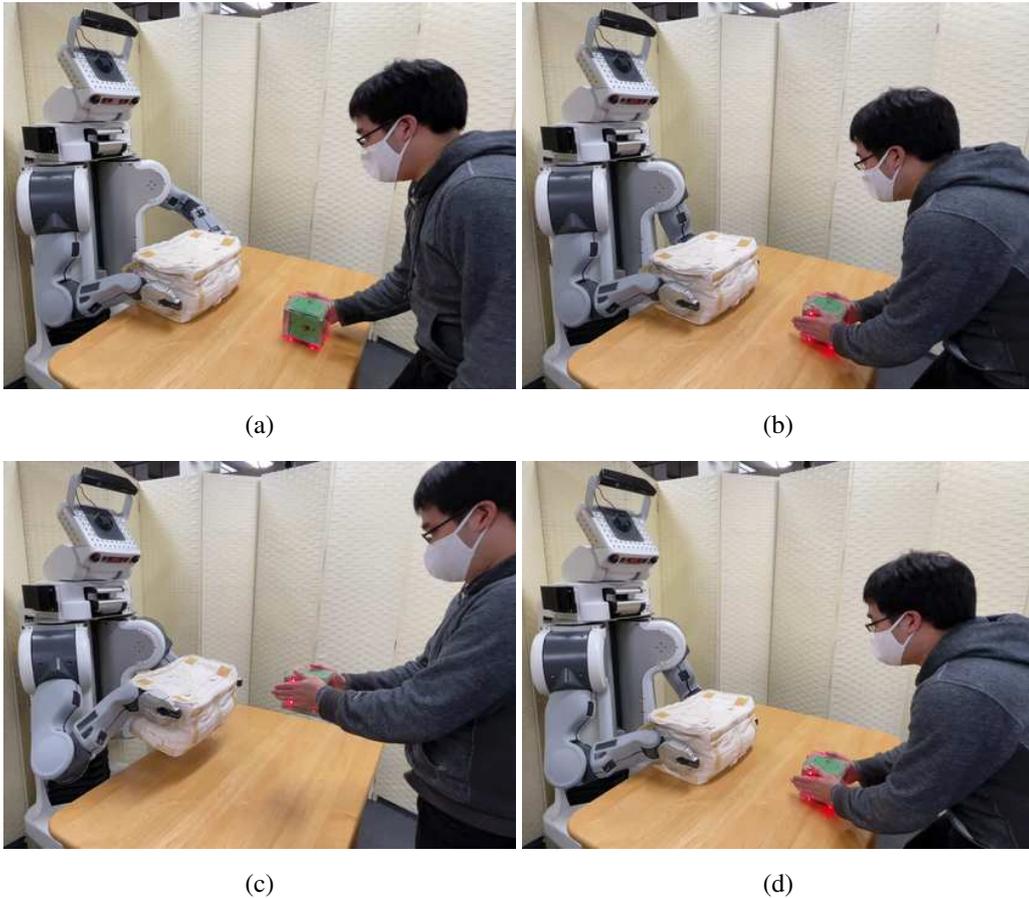


図 4.20: Miniature Tangible Cube による箱持ち上げ作業実験 [37]

を一旦離れたあとに持ち替えることで PR2-JSK も箱を持ち替え (図 4.23c, 図 4.23d, 図 4.24c, 図 4.24d,) 更にタンジブルインタフェースを Z 軸 45° もしくは -45° を回転させることで, PR2-JSK が箱を持ち替えながら Z 軸に 90° 回転させることに成功した (図 4.23e, 図 4.23f, 図 4.24e, 図 4.24f).

また, 対象物に対して AR マーカを取り付けて, PR2-JSK の頭部カメラから対象物の Z 軸回転計測を行った. 箱持ち替え回転作業実験 (Z 軸 90° 回転) における AR マーカによる対象物姿勢 Z 軸の時系列変化を図 4.25 に示す. 図 4.25 に示すように, 対象物の Z 軸は Miniature Tangible Cube の Z 軸変化に対して, 5 秒程度の遅れはありながらも追従し, 操縦者は Miniature Tangible Cube を用いて対象物を持ち替えながら Z 軸 90° 回転させることに成功した. さらに所要時間については, 6.6 節で説明する手先指向型インタフェースである VR コントローラによる手先指向型操縦システムを用いて, 同様の箱持ち替え回転作業実験を行い比較した. 結果は表 4.2 に示すように, VR コントローラを用いた場合には 131 秒所要したのに対して, Miniature Tangible Cube を用いた場合には 109 秒と 22 秒早く作業を行うことが出来た. 以上の結果から, Miniature Tangible Cube を用いて箱持ち替え回転操作を行えることが確認でき,

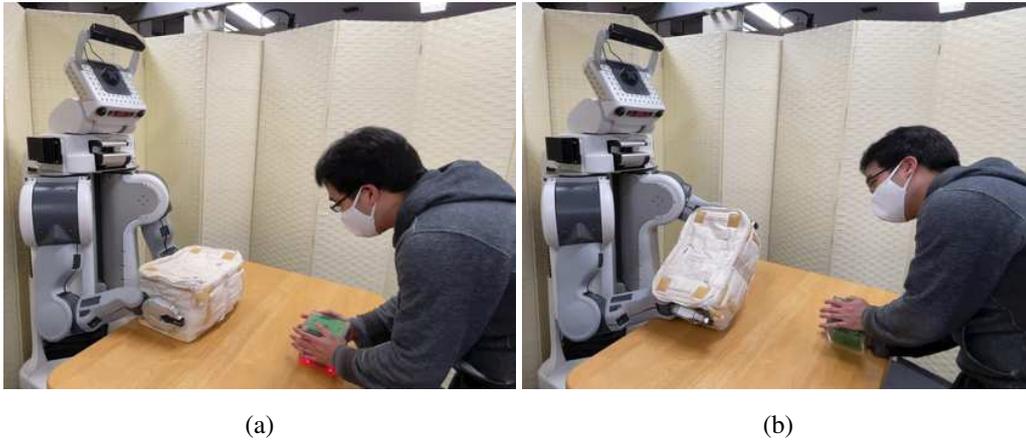


図 4.21: Miniature Tangible Cube による箱傾け作業実験 (X 軸 45°) [37]



図 4.22: Miniature Tangible Cube による箱傾け作業実験 (X 軸 -45°)

また VR コントローラを用いた場合よりも所要時間が短くなることで Miniature Tangible Cube の有効性を確認した。

表 4.2: Miniature Tangible Cube による箱持ち替え回転作業実験の所要時間 [37]

	Consuming time
Miniature Tangible Cube	109 s
VR controller	131 s

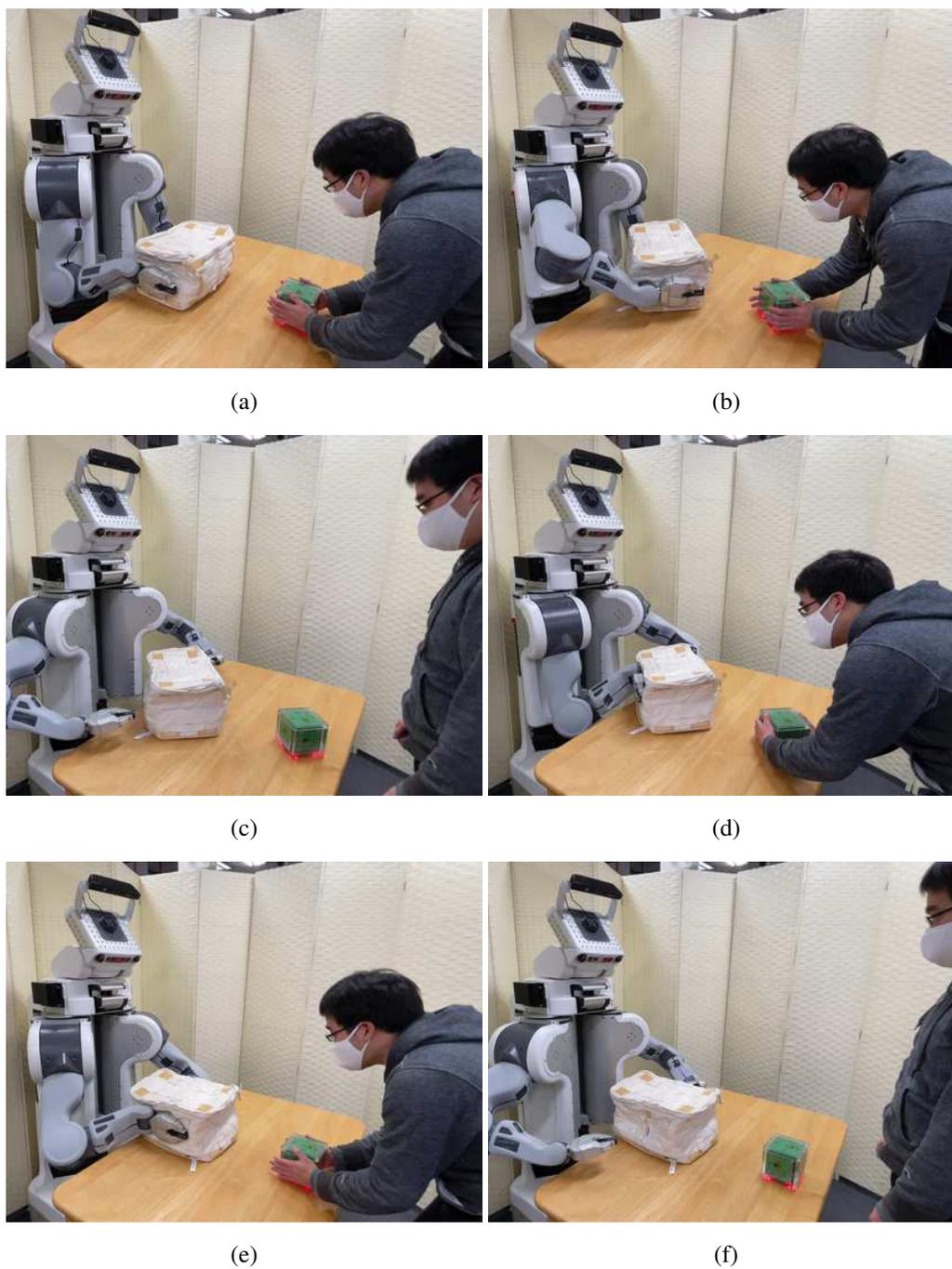


図 4.23: Miniature Tangible Cube による箱持ち替え回転作業実験 (Z 軸 90°) [37]

4.9.3 Miniature Tangible Cube による複数物体の操作作業の操縦実行実験

本項では本研究で作成したタンジブルインタフェースである Miniature Tangible Cube を用いた複数対象物に対する物体操作作業の操縦実行実験について説明する。本実験では複数

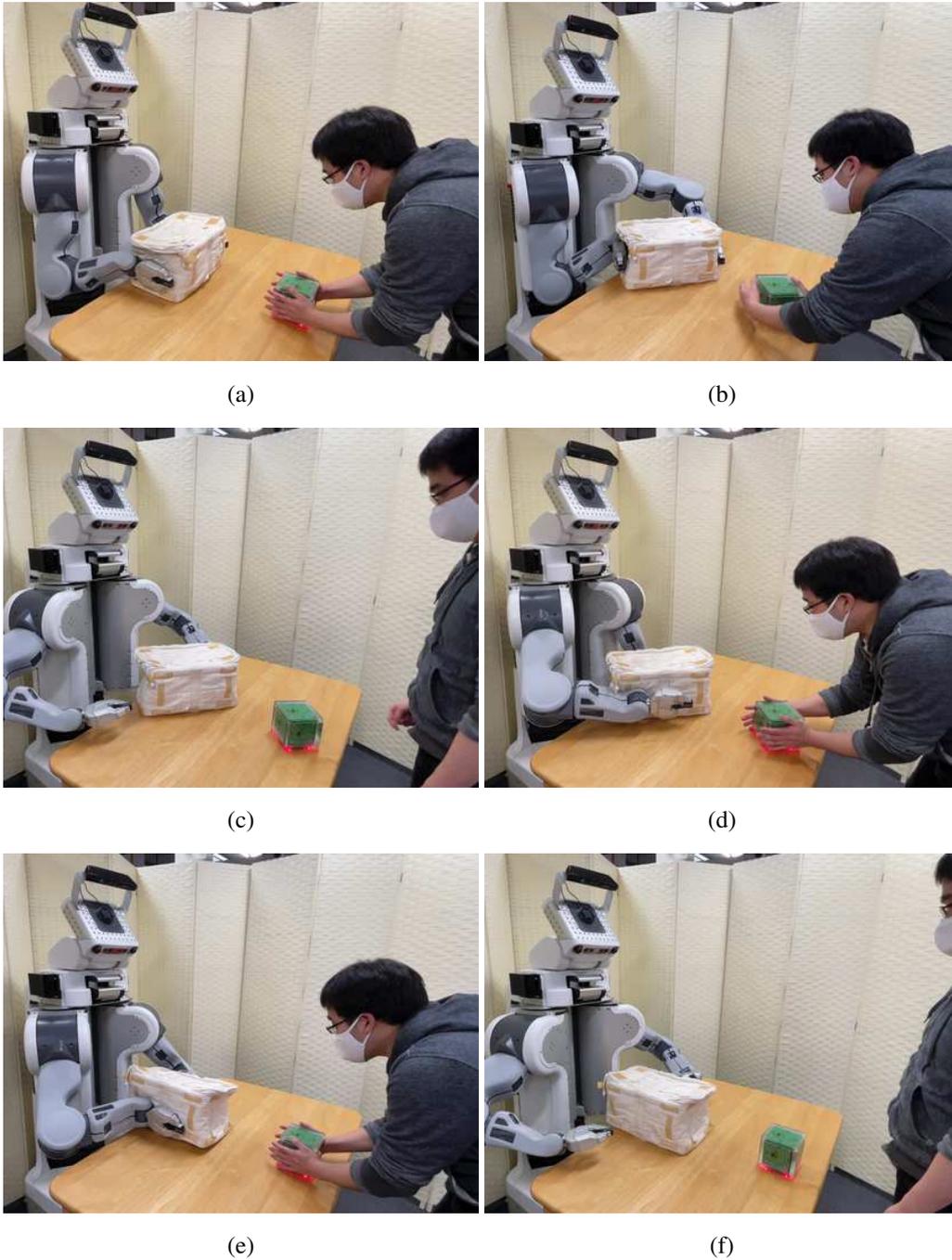


図 4.24: Miniature Tangible Cube による箱持ち替え回転作業実験 (Z 軸 -90°)

対象物を対象とするため 2 台の Miniature Tangible Cube Rev.2 と双腕ロボット Baxter-JSK-SoftHand を用いて箱に対する物体操作作業を行った。本実験では対象物として図 4.26 に示す幅 220 mm, 奥行き 210 mm, 高さ 130 mm のダンボール製の箱を用いた。なお本実験では操縦者は遠隔ロボットの頭部カメラの画像に基づいて操縦実行を行った。

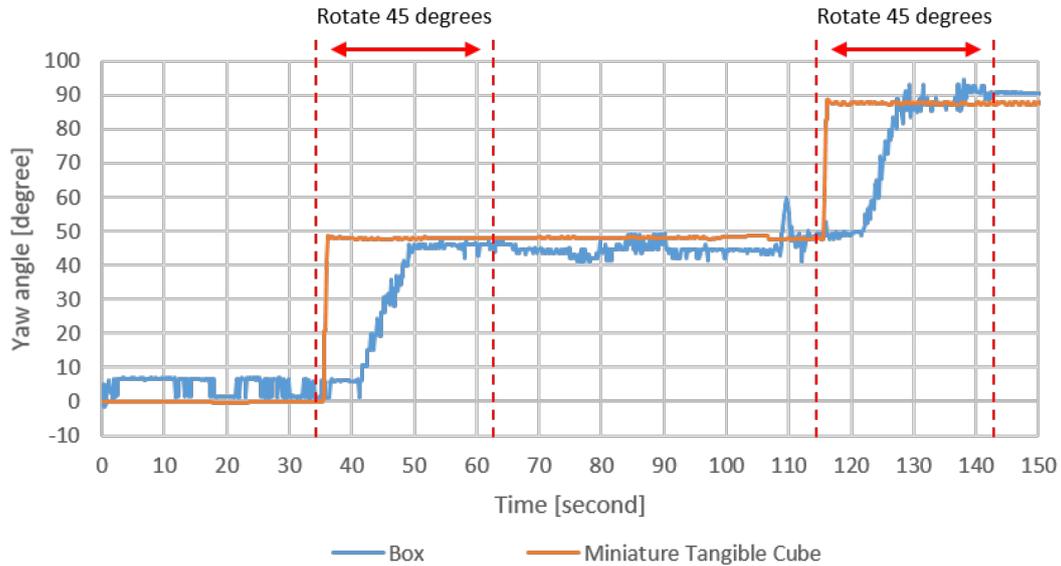


図 4.25: Miniature Tangible Cube による箱持ち替え回転作業実験 (Z 軸 90°) における AR マーカによる対象物姿勢 Z 軸の時系列変化 [37]



図 4.26: 複数物体操作作業の操縦実行実験で用いるダンボール製の箱

4.9.3.1 Miniature Tangible Cube による箱詰め寄せ作業実験

本目では Miniature Tangible Cube による箱詰め寄せ作業実験について説明する。本実験では図 4.26 に示すダンボール製の箱を対象物として用いた。本研究で行った Miniature Tangible Cube による箱詰め寄せ作業実験については、操縦者側の様子を図 4.27, ロボット側の様子を図 4.28, 操縦者側表示画面を図 4.29 に示す。図 4.27, 図 4.28, 図 4.29 に示すように、まず操縦者が右側に置かれた Miniature Tangible Cube に対して右側面に接触することで (図 4.27b),

ロボットが右側に置かれた箱を右腕で右側面に対して接触した(図 4.28b, 図 4.29b). そして操縦者が右側の Miniature Tangible Cube を左に押し進め左側の Miniature Tangible Cube と接触させることで(図 4.27c), ロボットが右側に置かれた箱を右腕で左に移動させて, 左側に置かれた箱と接触させた(図 4.28c, 図 4.29c). 最後に操縦者が右側の Miniature Tangible Cube への接触を離すことで(図 4.27d), ロボットが右側に置かれた箱への接触を離れた(図 4.28d, 図 4.29d). 以上の一連の動作列によってロボットが右側の箱を押して左側の箱に詰め寄せることに成功した.

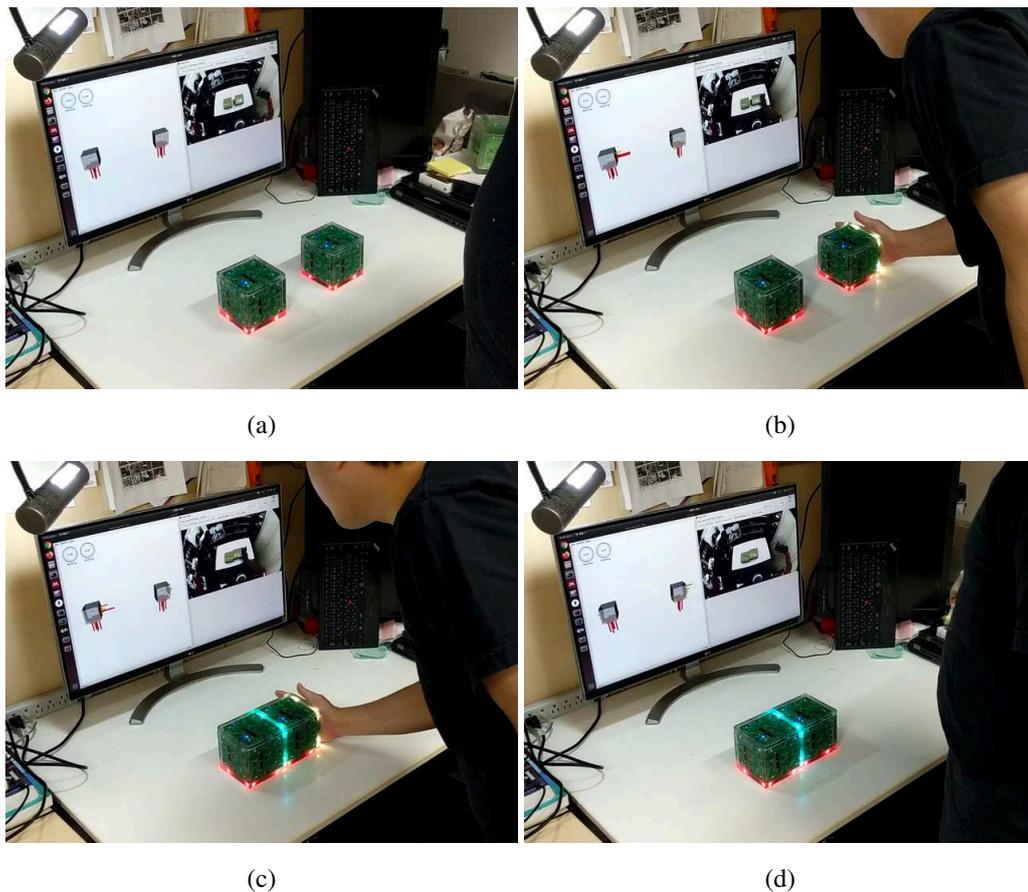


図 4.27: Miniature Tangible Cube による箱詰め寄せ作業実験 (操縦者側)

4.9.3.2 Miniature Tangible Cube による箱積み重ね作業実験

本目では Miniature Tangible Cube による箱積み重ね作業実験について説明する. 本実験では図 4.26 に示すダンボール製の箱を対象物として用いた. 本研究で行った Miniature Tangible Cube による箱積み重ね作業実験については, 操縦者側の様子を図 4.30, ロボット側の様子を図 4.31, 操縦者側表示画面を図 4.32 に示す. 図 4.30, 図 4.31, 図 4.32 に示すように, まず操縦者が手前に置かれた Miniature Tangible Cube に対して右側面に接触することで(図 4.30b), ロボットが手前に置かれた箱を右腕で右側面に対して接触し(図 4.31b, 図 4.32b), 操縦者が

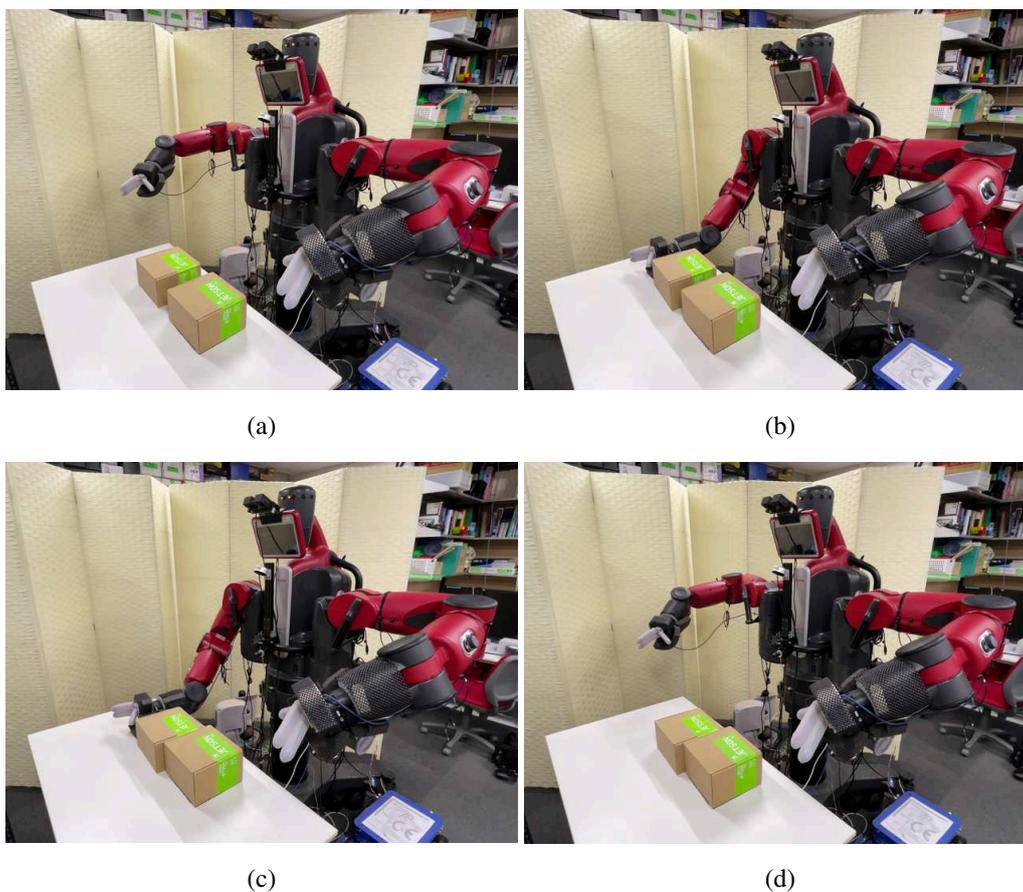


図 4.28: Miniature Tangible Cube による箱詰め寄せ作業実験 (ロボット側)

同じ Miniature Tangible Cube に対して左側面に接触することで (図 4.30c), ロボットが手前に置かれた箱を左腕で左側面に対して接触することで箱を双腕で抱えた (図 4.31c, 図 4.32c). そして操縦者が抱え込んだ手前の Miniature Tangible Cube を持ち上げることで (図 4.30d), ロボットが手前に置かれた箱を左右から抱え込んで持ち上げ (図 4.31d, 図 4.32d), 操縦者が持ち上げた手前の Miniature Tangible Cube を奥に置かれた Miniature Tangible Cube の上に置くことで (図 4.30e), ロボットが持ち上げた箱をロボットから見て奥にある箱の上に積み重ねた (図 4.31e, 図 4.32e). 最後に操縦者が積み重ねられた Miniature Tangible Cube に対する接触を左, 右と離すことで (図 4.30f, 図 4.30g, 図 4.30h), ロボットが積み重ねられた箱に対する接触を離れた (図 4.31f, 図 4.31g, 図 4.31h, 図 4.32f, 図 4.32g, 図 4.32h). 以上の一連の動作列によってロボットが手前の箱を持ち上げて奥の箱に積み重ねることに成功した.

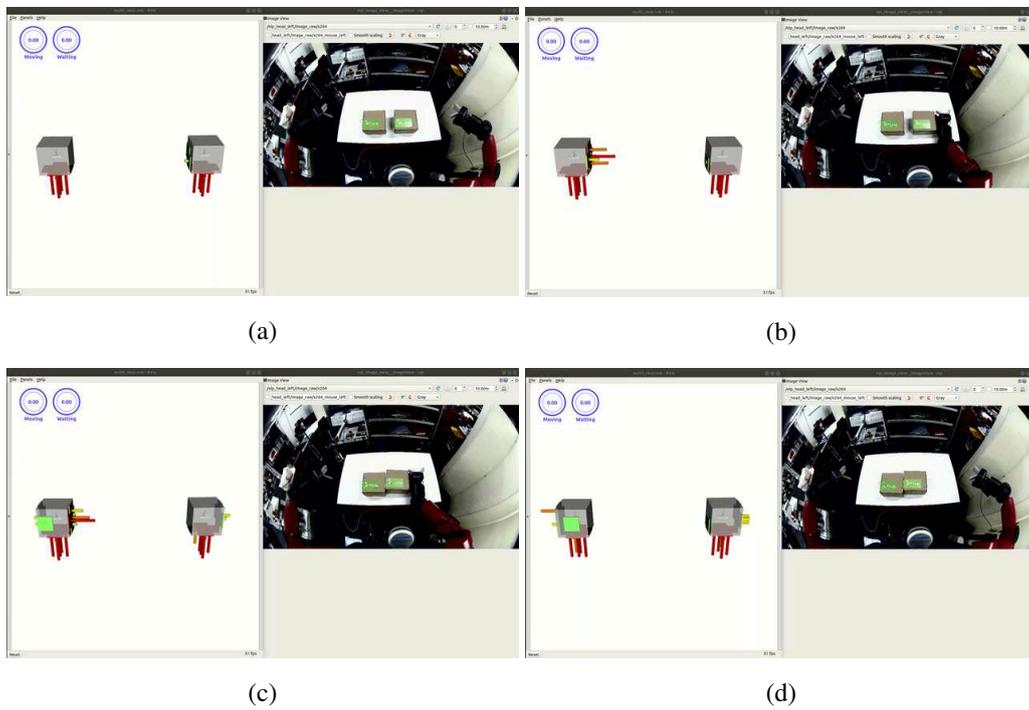


図 4.29: Miniature Tangible Cube による箱詰め寄せ作業実験 (操縦者側表示画面)

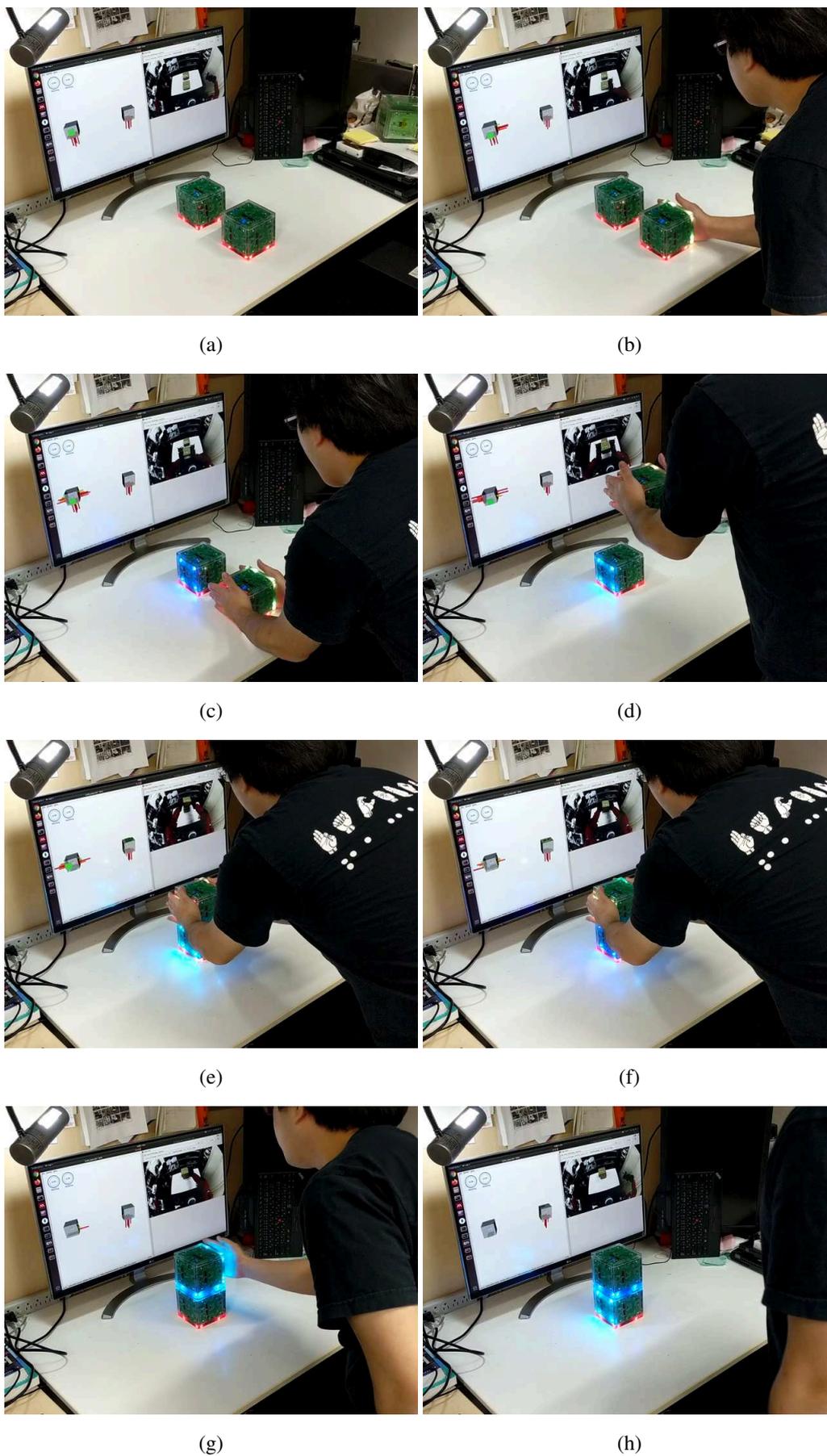


図 4.30: Miniature Tangible Cube による箱積み重ね作業実験 (操縦者側)

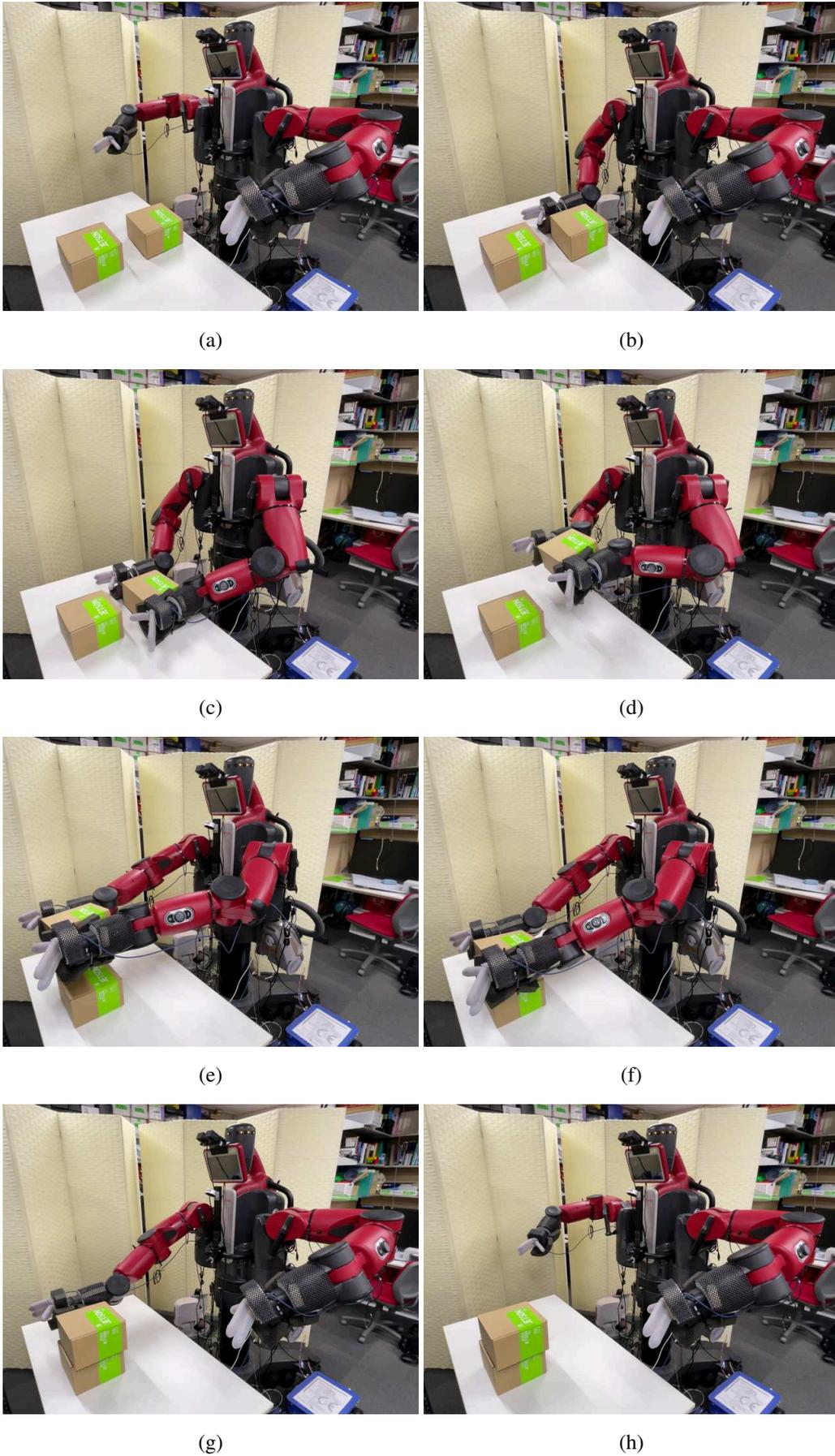


図 4.31: Miniature Tangible Cube による箱積み重ね作業実験 (ロボット側)

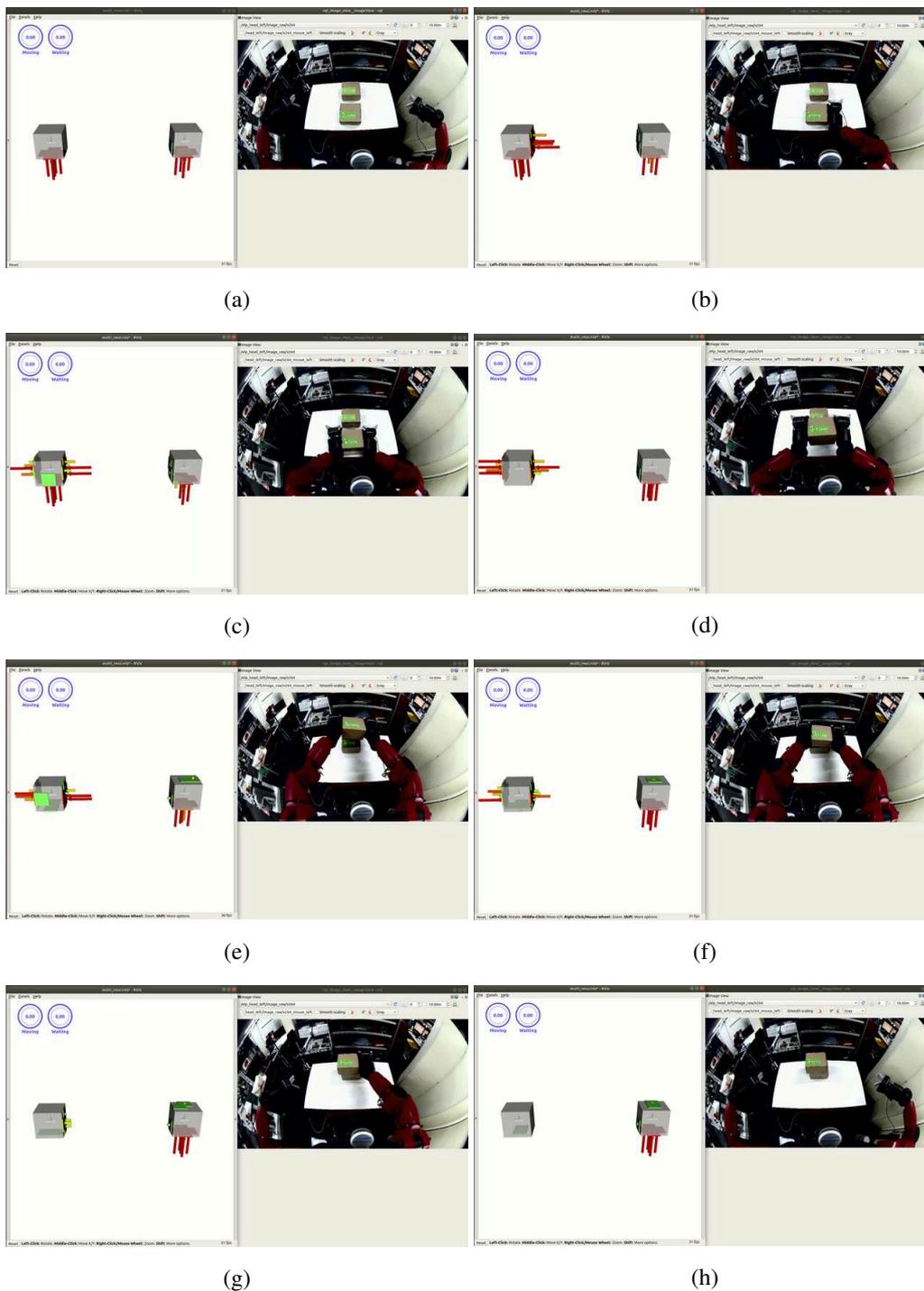


図 4.32: Miniature Tangible Cube による箱積み重ね作業実験 (操縦者側表示画面)

4.9.4 Miniature Tangible Cube による複雑形状物体の操作作業の操縦実行実験

本項では本研究で作成したタンジブルインタフェースである Miniature Tangible Cube を用いた複数対象物に対する物体操作作業の操縦実行実験について説明する。本実験では複数対象物を対象とするため 2 台の Miniature Tangible Cube Rev.2 と双腕ロボット Baxter-JSK-SoftHand を用いてぬいぐるみに対する物体操作作業を行った。本実験では対象物として図 4.33 に示す 2 つのぬいぐるみを用いた。なお本実験では操縦者は遠隔ロボットの頭部カメラの画像に基づいて操縦実行を行った。



図 4.33: 複雑形状物体の操作作業の操縦実行実験で用いるぬいぐるみ

4.9.4.1 Miniature Tangible Cube によるぬいぐるみ詰め寄せ作業実験

本目では Miniature Tangible Cube によるぬいぐるみ詰め寄せ作業実験について説明する。本研究で図 4.33 に示すぬいぐるみを対象物として用いた。また本実験ではロボットの腹部に設置されている RGB-D カメラで撮影された点群を用いて、机の平面除去を行うことで図 4.34 の青色の直方体領域に示すようなぬいぐるみの領域を直方体近似することで認識した。

本研究で行った Miniature Tangible Cube によるぬいぐるみ詰め寄せ作業実験については、ロボット側の様子を図 4.35, 操縦者側の様子を図 B.1, 操縦者側表示画面の様子を図 B.2 に示す。本実験の動作は 4.9.3.1 目と同様に図 4.35 に示すように、まず操縦者が右側に置かれた Miniature Tangible Cube に対して右側面に接触することで、ロボットが右側に置かれたぬいぐるみを右腕で右側面に対して接触した(図 4.35b)。そして操縦者が右側の Miniature Tangible Cube を左に押しいき左側の Miniature Tangible Cube と接触させることで、ロボットが右側に置かれたぬいぐるみを右腕で左に移動させて、左側に置かれたぬいぐるみと接触させた(図 4.35c)。最後に操縦者が右側の Miniature Tangible Cube への接触を離すことで、ロボットが右

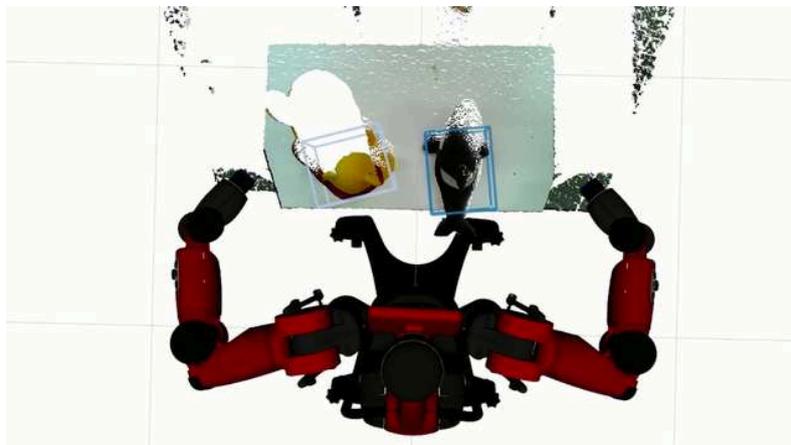


図 4.34: ぬいぐるみ詰め寄せ作業実験における直方体近似を用いたぬいぐるみの領域認識

側に置かれたぬいぐるみへの接触を離れた (図 4.35d). 以上の一連の動作列によってロボットが右側のぬいぐるみを押して左側のぬいぐるみに詰め寄せることに成功した.

4.9.4.2 Miniature Tangible Cube によるぬいぐるみ積み重ね作業実験

本目では Miniature Tangible Cube によるぬいぐるみ積み重ね作業実験について説明する. 本研究で図 4.33 に示すぬいぐるみを対象物として用いた. また本実験では Baxter-JSK-Softhand の腹部に設置されている RGB-D カメラで撮影された点群を用いて, 机の平面除去を行うことで図 4.36 の青色の直方体領域に示すようなぬいぐるみの領域を直方体近似することで認識した.

本研究で行った Miniature Tangible Cube によるぬいぐるみ詰め寄せ作業実験については, ロボット側の様子を図 4.37, 図 4.38, 操縦者側の様子を図 B.3, 図 B.5, 操縦者側表示画面の様子を図 B.4, 図 B.6 に示す. 本実験の動作は 4.9.3.2 目と同様に図 4.37, 図 4.38 に示すように, まず操縦者が手前に置かれた Miniature Tangible Cube に対して左右側面に接触することでロボットが手前に置かれたぬいぐるみを両腕で抱えた (図 4.37b, 図 4.38b). そして操縦者が抱え込んだ手前の Miniature Tangible Cube を持ち上げて奥に置かれたもう一つ Miniature Tangible Cube の上に置くことで, ロボットが手前に置かれたぬいぐるみを左右から抱え込んで持ち上げ, 持ち上げたぬいぐるみをロボットから見て奥にある箱の上に積み重ねた (図 4.37c, 図 4.38c). 最後に操縦者が積み重ねられた Miniature Tangible Cube に対する接触を左右と離すことでロボットが積み重ねられたぬいぐるみに対する接触を離れた (図 4.37d, 図 4.38d). 以上の一連の動作列によってロボットが手前のぬいぐるみを持ち上げて奥の箱に積み重ねることに成功した.

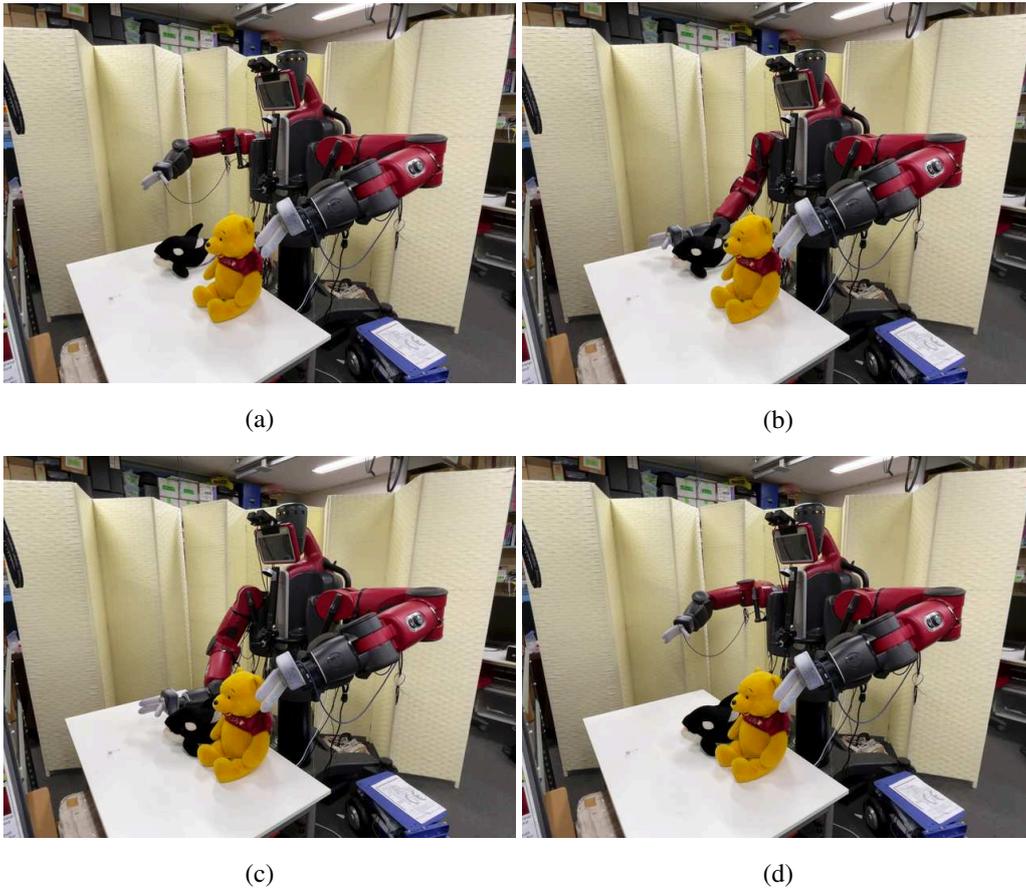
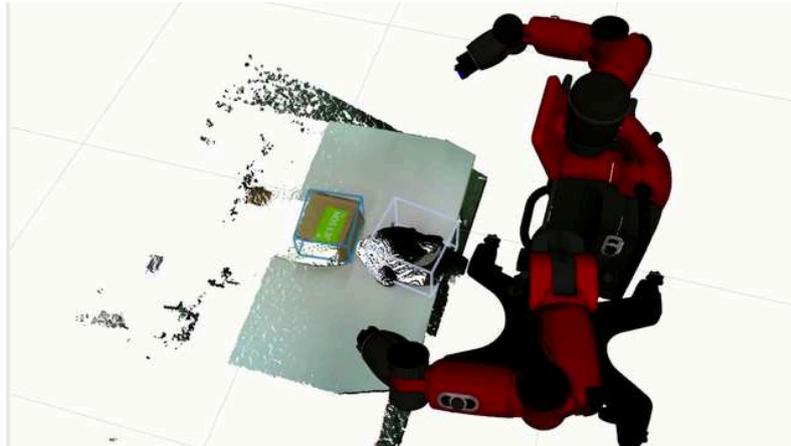


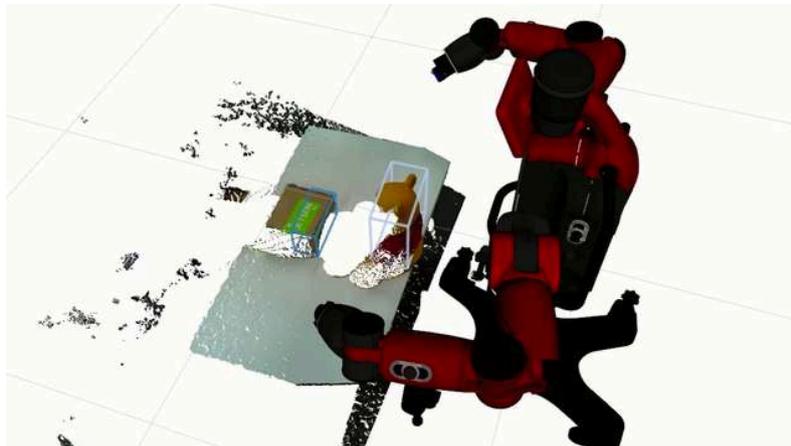
図 4.35: Miniature Tangible Cube によるぬいぐるみ詰め寄せ作業実験 (ロボット側)

4.9.5 Miniature Tangible Cube による把持動作を伴う物体操作作業の操縦実行実験

本項では本研究で作成したタンジブルインタフェースである Miniature Tangible Cube を用いた把持動作を伴う物体操作作業の操縦実行実験について説明する。本実験では複数対象物を対象とするため 1 台の Miniature Tangible Cube Rev.1 と 2 台の Miniature Tangible Cube Rev.2 の計 3 台と、双腕ロボット Baxter-JSK-Softhand を用いてダンボールの箱と把持物体に対する物体操作作業を行った。本実験では対象の把持物体として 図 4.39 に示すスポンジと袋入り靴下を用いた。これら 2 つは Amazon Robotics Challenge [97] に用いられたピッキングタスク対象物品から選択した。なお本実験では操縦者は遠隔ロボットの頭部カメラの画像に基づいて操縦実行を行った。



(a) 直方体近似を用いたシャチのぬいぐるみの領域認識



(b) 直方体近似を用いたくまのぬいぐるみの領域認識

図 4.36: ぬいぐるみ積み重ね作業実験における直方体近似を用いたぬいぐるみの領域認識

4.9.5.1 Miniature Tangible Cube による把持動作を伴う物体ダンボール箱収納作業実験

本目では Miniature Tangible Cube による物体ダンボール箱収納作業実験について説明する。本研究で図 4.39 に示す Amazon Robotics Challenge [97] に用いられたピッキングタスク対象物品を対象物として用いた。また本実験ではロボットの腹部に設置されている RGB-D カメラで撮影された点群を用いて、机の平面除去を行うことで図 4.40 の青色の直方体領域に示すような把持物体の領域を直方体近似することで認識した。

操縦者側の様子を図 4.41, ロボット側の様子を図 4.42, 図 4.44, 操縦者側表示画面の様子を図 4.43 に示す。図 4.41, 図 4.42, 図 4.43, 図 4.44 に示すように、まず操縦者が左側に置かれた Miniature Tangible Cube に対して上側面に接触することで(図 4.41b), ロボットが左側に置かれた把持物体を左腕で把持し(図 4.42b, 図 4.43b, 図 4.44b), 操縦者が左側に置かれた Miniature Tangible Cube を上側面から持ったまま持ち上げることで(図 4.41c), ロボット

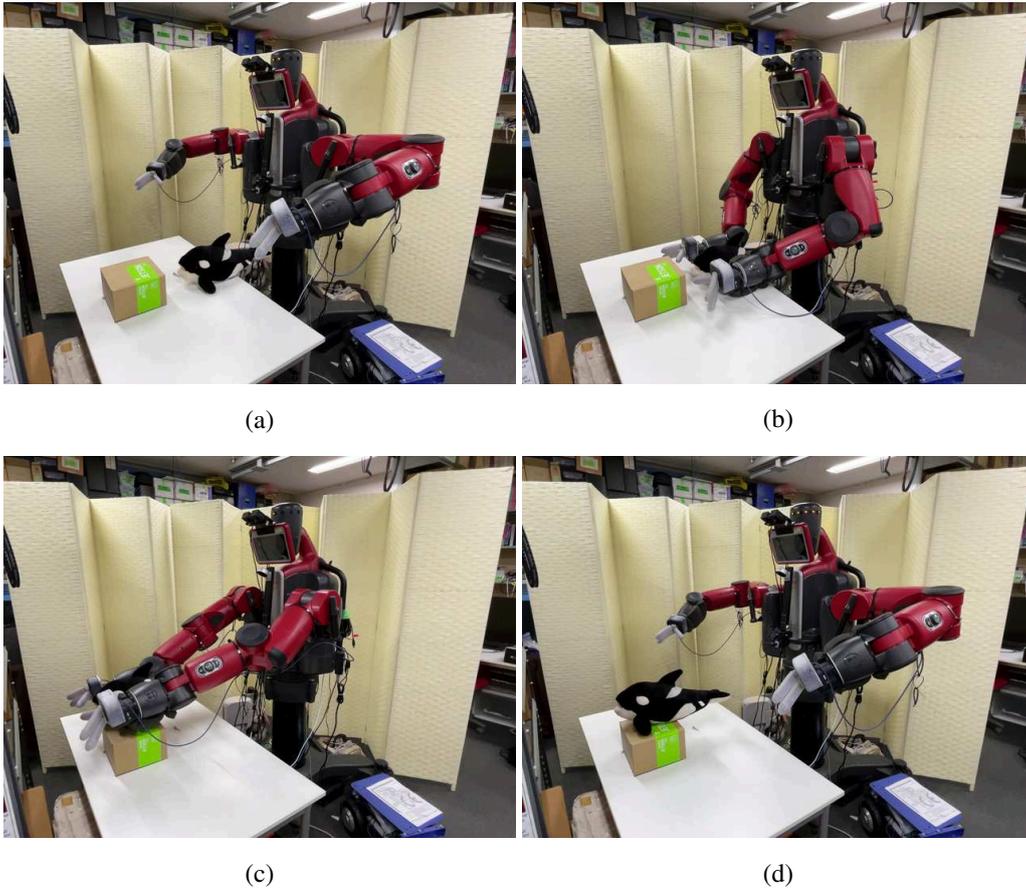


図 4.37: Miniature Tangible Cube によるシャチのぬいぐるみ積み重ね作業実験 (ロボット側)

は把持物体を持ち上げた (図 4.42c, 図 4.43c, 図 4.44c). そして操縦者が持ち上げた Miniature Tangible Cube を中央手前に置かれた Miniature Tangible Cube の上に置くことで (図 4.41d), ロボットは把持物体を中央に置かれたダンボール箱の上まで移動させ (図 4.42d, 図 4.43d, 図 4.44d), 操縦者が上側面から掴んでいた Miniature Tangible Cube を離すことで (図 4.41e), ロボットは把持物体を離して中央に置かれたダンボール箱の中に把持物体を収納した (図 4.42e, 図 4.43e, 図 4.44e).

この後の動作は 4.9.3.2 目と同様に, 操縦者が中央手前に積み重ねられた 2 つの Miniature Tangible Cube の下のタンジブルインタフェースを左右から接触して持ち上げることで (図 4.41f), ロボットは把持物体が収納されたダンボール箱を左右から抱え込んで持ち上げ (図 4.42f, 図 4.43f, 図 4.44f), 操縦者が持ち上げた 2 つの Miniature Tangible Cube を奥に置かれた Miniature Tangible Cube の上に置くことで (図 4.41g), ロボットは把持物体が収納されたダンボール箱を奥にあるダンボール箱の上に積み重ねた (図 4.42g, 図 4.43g, 図 4.44g). 最後に操縦者抱えていた Miniature Tangible Cube に対する接触を左右と離すことで (図 4.41h), ロボットは積み重ねられたダンボール箱に対する接触を離れた (図 4.42h, 図 4.43h, 図 4.44h).

以上の一連の動作列によって, ロボットが左側におかれた物体を把持して持ち上げて, それ

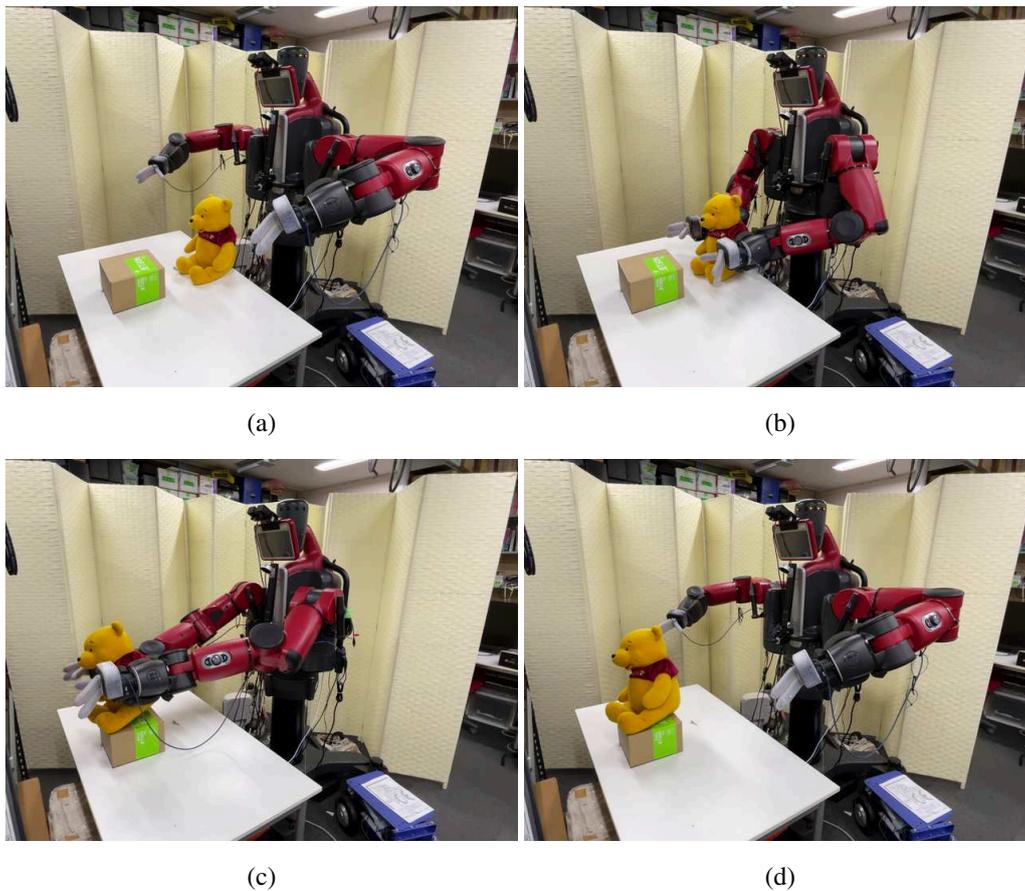
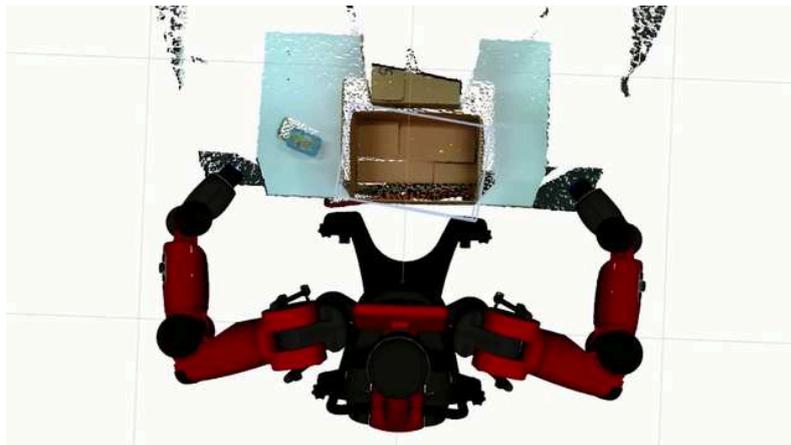


図 4.38: Miniature Tangible Cube によるくまのぬいぐるみ積み重ね作業実験 (ロボット側)

を中央手前に置かれたダンボール箱に収納したあとに、物体を収納したそのダンボール箱を奥にあるダンボール箱の上に積み重ねることに成功した。



図 4.39: 把持動作を伴う物体操作作業の操縦実行実験で用いる把持物体



(a) 直方体近似を用いたスポンジの領域認識



(b) 直方体近似を用いた袋入り靴下の領域認識

図 4.40: 物体ダンボール箱収納作業実験における直方体近似を用いた把持物体の領域認識

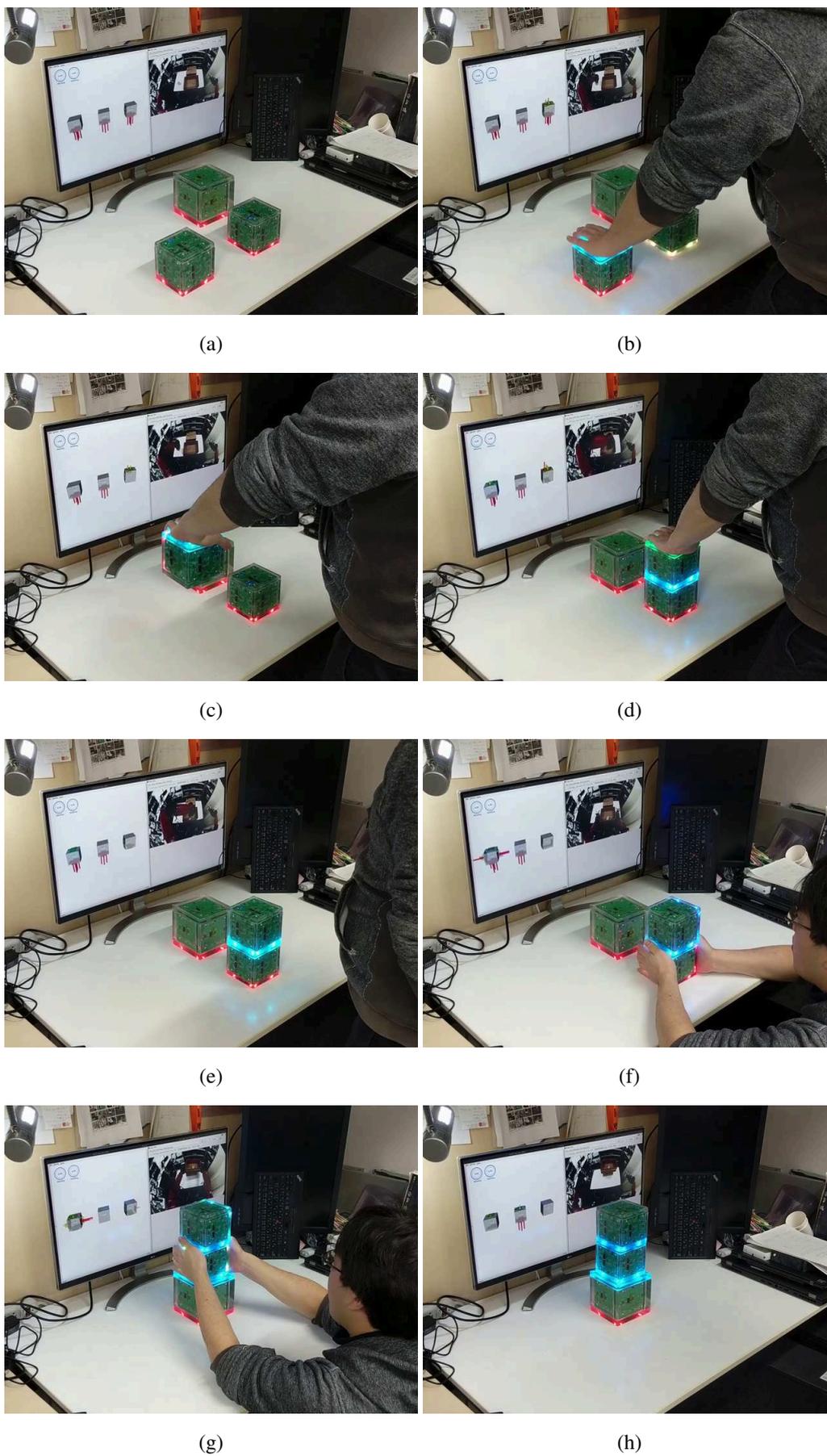


図 4.41: Miniature Tangible Cube によるスポンジのダンボール箱収納作業実験 (操縦者側)

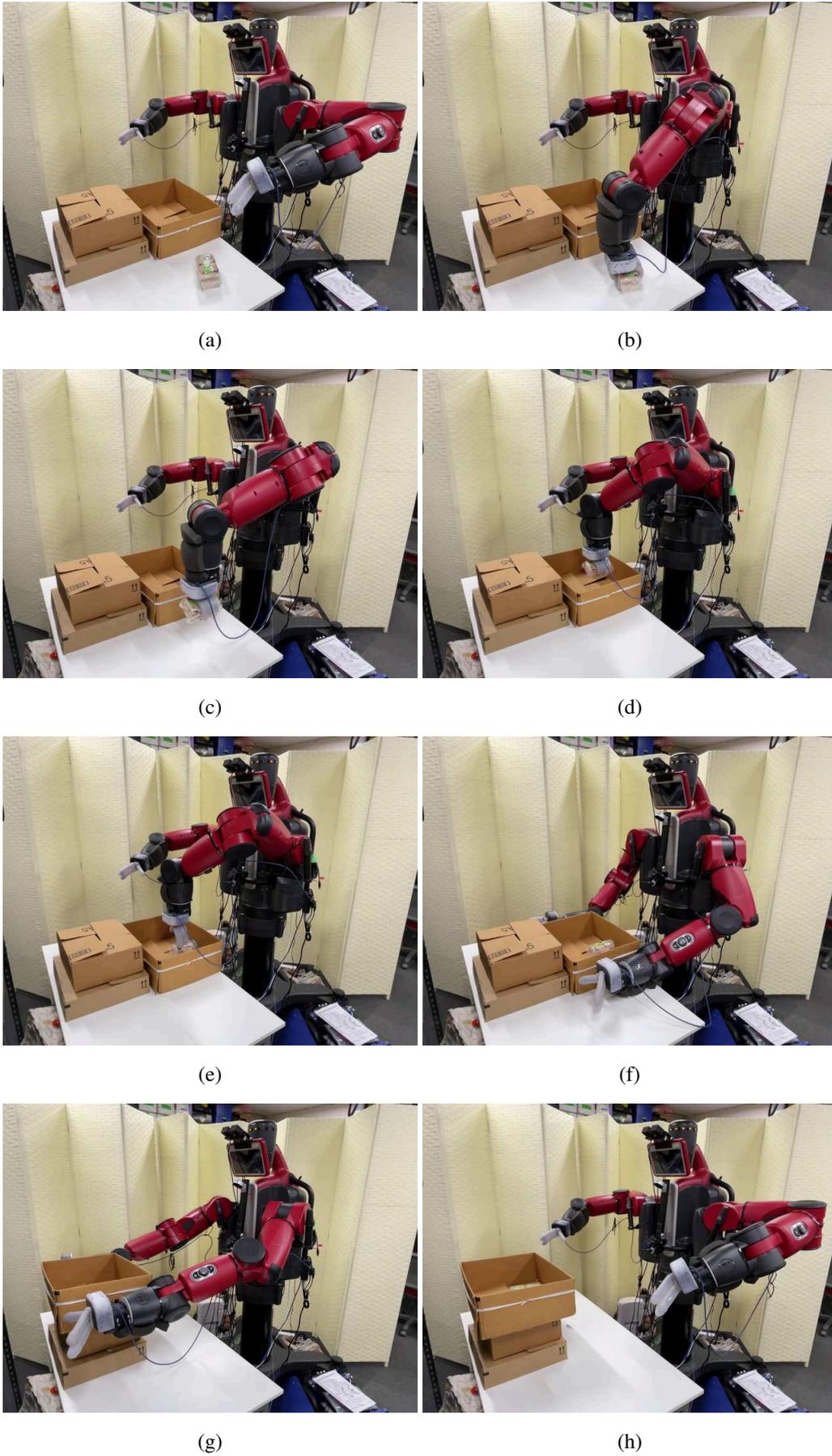


図 4.42: Miniature Tangible Cube によるスポンジのダンボール箱収納作業実験 (ロボット側)



図 4.43: Miniature Tangible Cube によるスポンジのダンボール箱収納作業実験 (操縦者側表示画面)

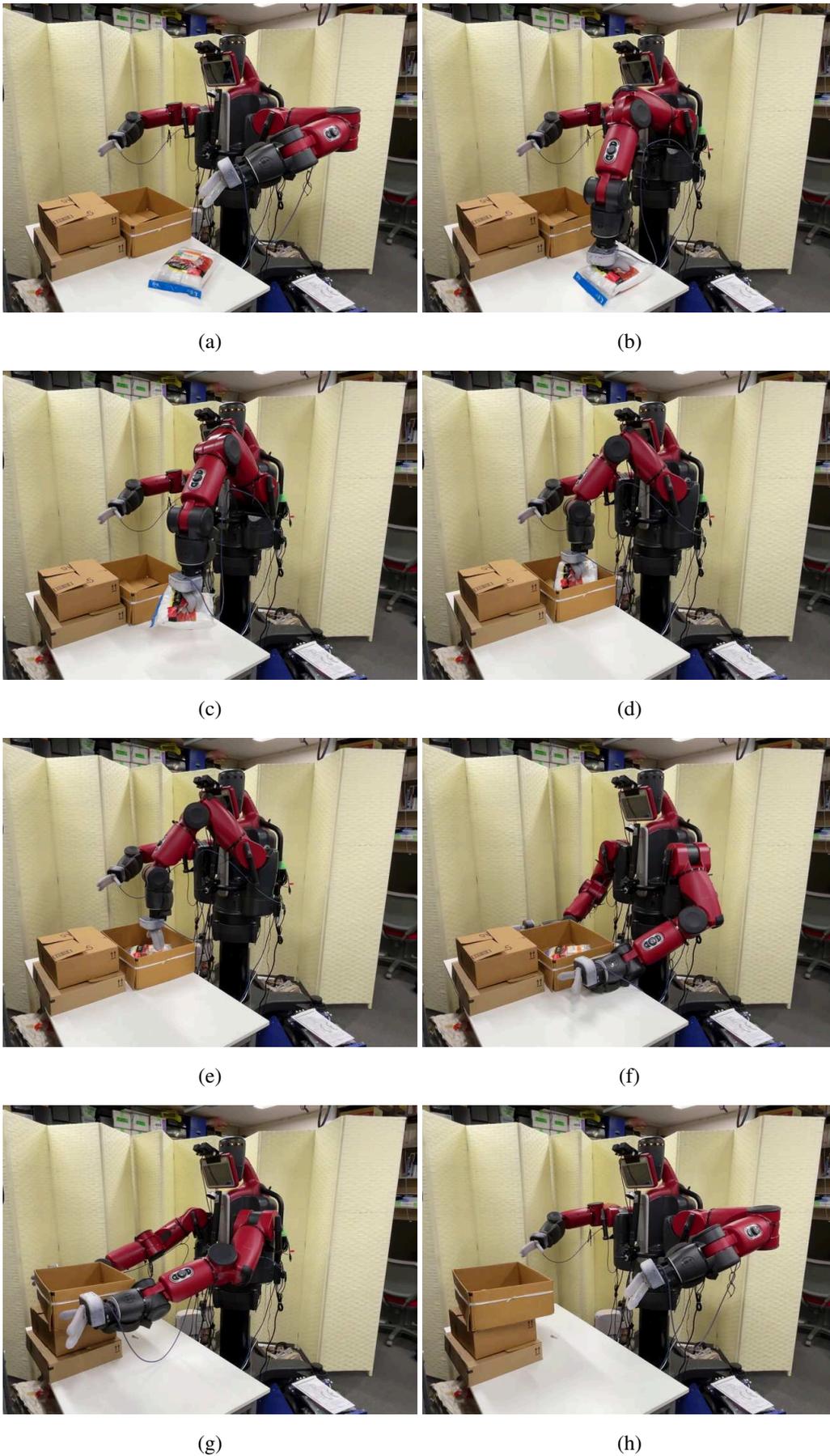


図 4.44: Miniature Tangible Cube による袋入り靴下のダンボール箱収納作業実験 (ロボット側)

4.9.6 Miniature Tangible Cube による教示学習に基づく箱操作作業の自律実行実験

本項では、箱操作作業における Miniature Tangible Cube による動作順序の教示学習に基づく作業の自律実行実験について説明する。本実験には図 4.45 に示す動作順序グラフを用いて自律実行を行う。この動作順序グラフは 4.9.3 項で実行した操縦実行実験で収集されたタンジブルインタフェース状態と動作順序を元に構成されたものであり、本実験ではこの動作順序グラフ上で現在の状態から作業の目標状態に対して経路探索し、その経路上にある動作を有限状態機械として取り出して、経路順序に実行することで自律実行を実現する。

4.9.6.1 箱詰め寄せ作業の教示学習に基づく自律実行実験

本目では箱詰め寄せの教示学習に基づく自律実行実験について説明する。本実験では図 4.26 に示すダンボール製の箱を対象物として用いた。本実験ではまず 4.9.3 項において実行した操縦実行実験のとき実行された動作順序を元に図 4.45 に示す動作順序グラフを構築し、この動作順序グラフに基づいて箱の詰め寄せ作業の自律実行実験を行った。本実験では箱詰め寄せ作業の目標状態は右の箱の左側面と左の箱の右側面が接触している状態として自律実行実験を行い、4.9.3 項において教示学習された動作順序グラフから動作経路として生成された有限状態機械を図 4.46 に示す。このとき右側の箱を物体 0、左側の箱を物体 1 として ID を振ることで区別する。図 4.46 に示すように計画された動作順序は以下のようになった。

1. (rarm-right-push-action): 右側の箱 (物体 0) の右側面に対して右手で接触する
2. (rarm-left-push-next-to-action :cube-index 1): 右側の箱 (物体 0) の左側面と左側の箱 (物体 1) の右側面が接触するように左側に動かす
3. (rarm-push-release-action): 右側の箱 (物体 0) の右手による接触を離す

本実験ではこの動作順序を自律実行する。

本研究で行った箱詰め寄せの自律実行実験については図 4.47 に示す。図 4.47 に示すように、まずロボットが右側に置かれた箱を右腕で右側面に対して接触した (図 4.47b)。そしてロボットが右側に置かれた箱を右腕で左に移動させて、左側に置かれた箱と接触させた (図 4.47c)。最後にロボットが右側に置かれた箱への接触を離した (図 4.47d)。以上の一連の動作順序によってロボットが右側の箱を押して左側の箱に詰め寄せることを操縦実行から自律実行へ切り替えて実行することに成功した。

4.9.6.2 箱積み重ね作業の教示学習に基づく自律実行実験

本目では箱積み重ねの教示学習に基づく自律実行実験について説明する。本実験では図 4.26 に示すダンボール製の箱を対象物として用いた。本実験ではまず 4.9.3 項において実行した操縦実行実験のとき実行された動作順序を元に図 4.45 に示す動作順序グラフを構築し、この動作順序グラフに基づいて箱積み重ね作業の自律実行実験を行った。本実験では箱積み重ね作業

の目標状態は手前の箱の底面と奥の箱の上面が接触している状態として自律実行実験を行い、4.9.3 項において教示学習された動作順序グラフから動作経路として生成された有限状態機械を図 4.48 に示す。このとき手前の箱を物体 0, 奥の箱を物体 1 として ID を振ることで区別する。図 4.48 に示すように計画された動作順序は以下のようになった。

1. (rarm-right-push-action): 手前の箱 (物体 0) の右側面に対して右手で接触する
2. (larm-left-push-action): 手前の箱 (物体 0) の左側面に対して左手で接触することで手前の箱を抱え込む
3. (dualarm-push-hold-up-action): 手前の箱 (物体 0) について両腕抱え込んだまま持ち上げる
4. (dualarm-push-hold-pile-up-action :cube-index 1): 手前の箱 (物体 0) を奥の箱 (物体 1) の上に置く
5. (rarm-push-release-action): 上に置かれた箱 (物体 0) の右手による接触を離す
6. (larm-push-release-action): 上に置かれた箱 (物体 0) の左手による接触を離す

本実験ではこの動作順序を自律実行する。

本研究で行った箱積み重ねの自律実行実験については図 4.49 に示す。図 4.49 に示すように、まずロボットが手前に置かれた箱を右腕で右側面に対して接触し (図 4.49b), ロボットが手前に置かれた箱を左腕で左側面に対して接触することで箱を双腕で抱えた (図 4.49c)。そしてロボットが手前に置かれた箱を左右から抱え込んで持ち上げ (図 4.49d), ロボットが持ち上げた箱をロボットから見て奥にある箱の上に積み重ねた (図 4.49e)。最後にロボットが積み重ねられた箱に対する接触を離した (図 4.49f, 図 4.49g, 図 4.49h)。以上の一連の動作順序によってロボットが手前の箱を持ち上げて奥の箱に積み重ねることを操縦実行から自律実行へ切り替えて実行することに成功した。

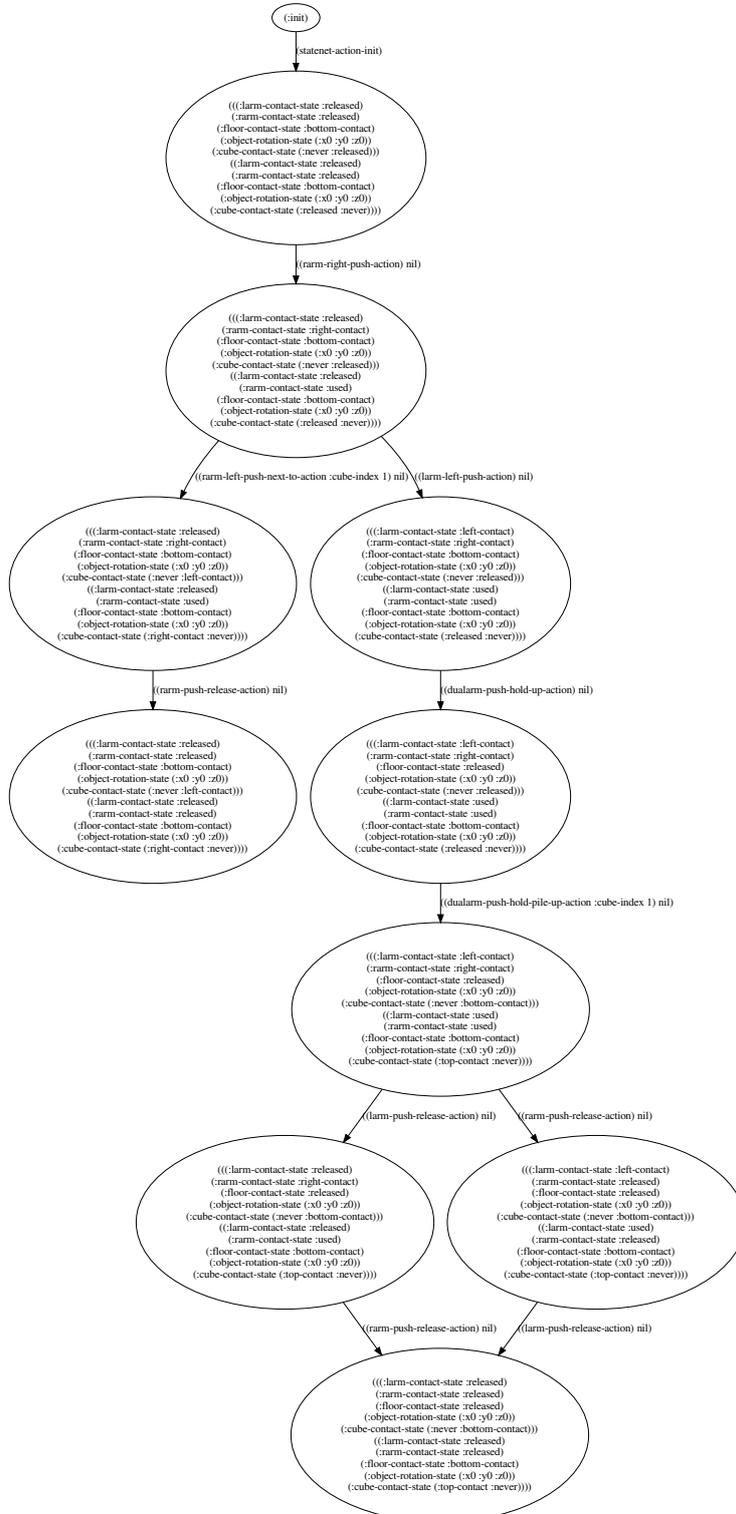


図 4.45: Miniature Tangible Cube による箱操作作業の操縦実行時に教示学習された動作順序グラフ

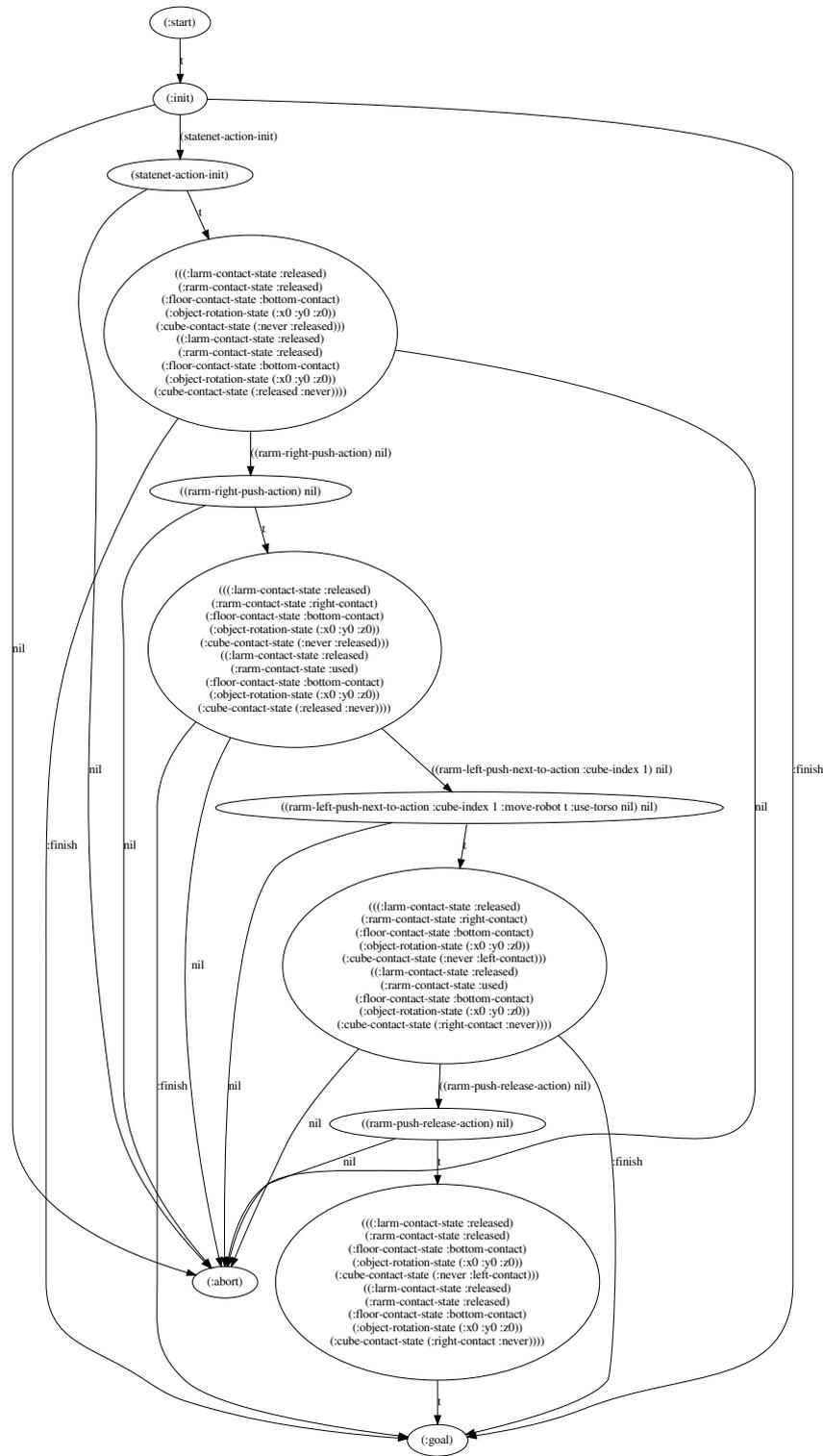


図 4.46: 教示学習された動作順序グラフから構成された箱詰め寄せ作業の自律実行のための有限要素機械



(a)

(b)



(c)

(d)

図 4.47: 箱詰め寄せ作業の教示学習に基づく自律実行

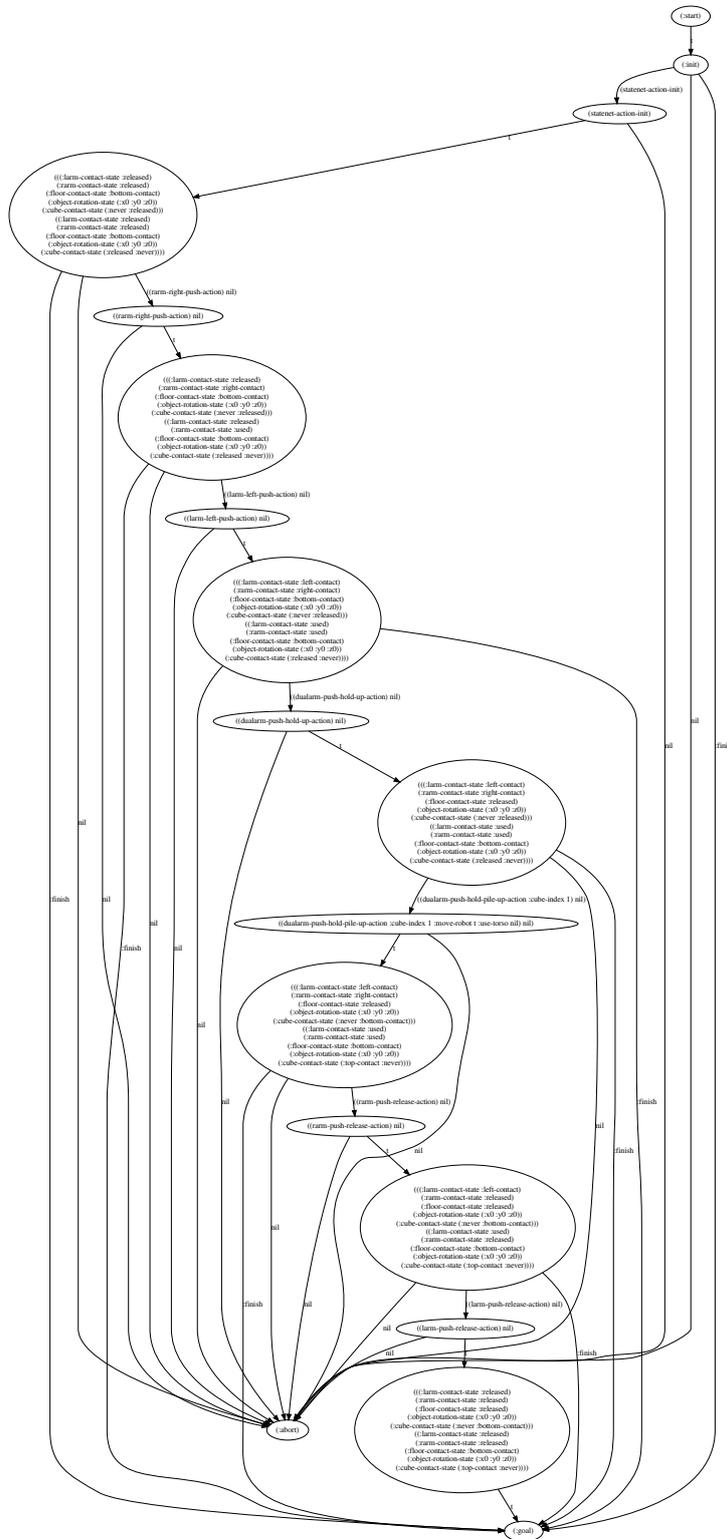


図 4.48: 教示学習された動作順序グラフから構成された箱積み重ね作業の自律実行のための有限要素機械

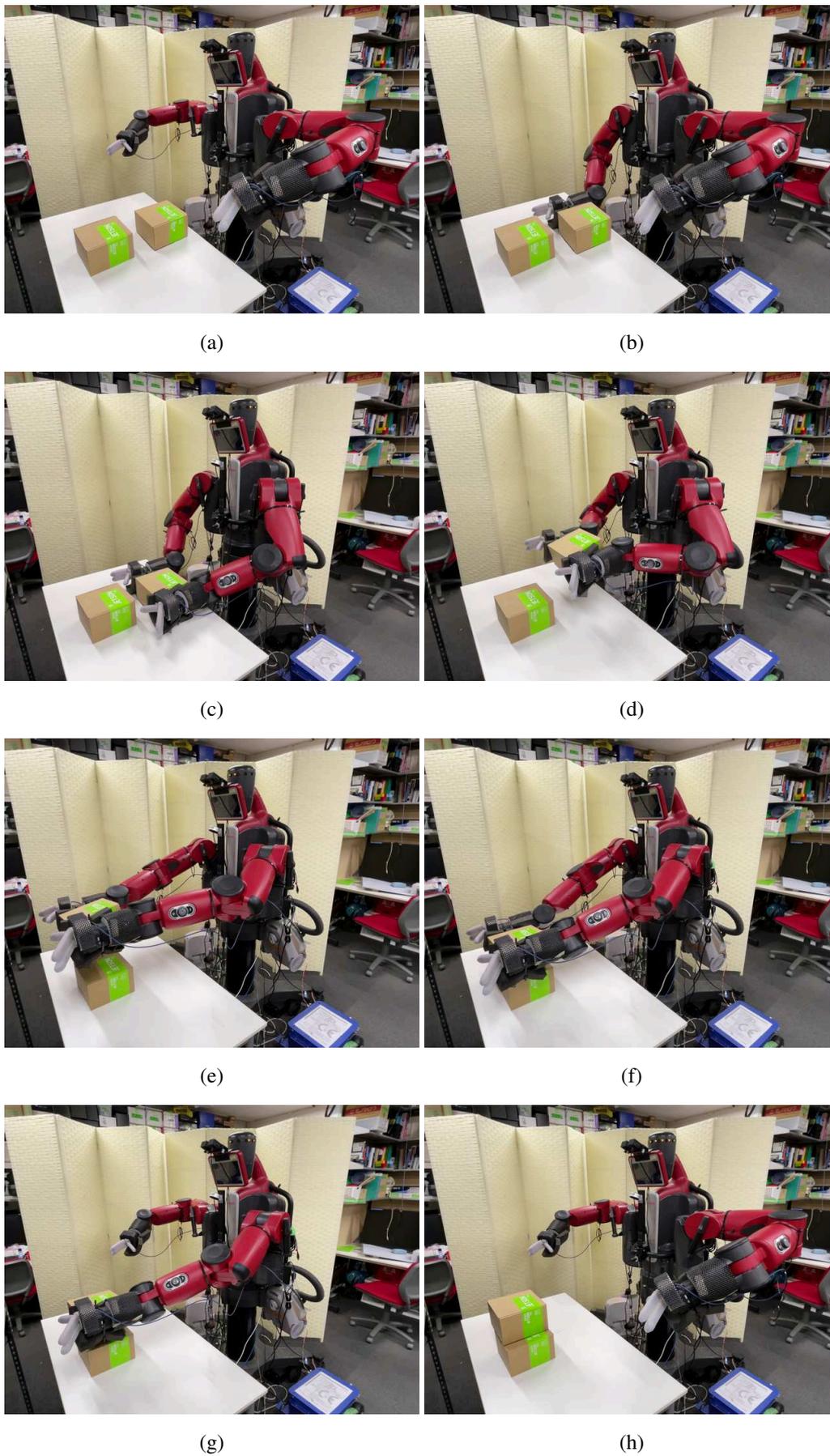


図 4.49: 箱積み重ね作業の教示学習に基づく自律実行

4.10 議論

本節では本研究で提案するタンジブルインタフェースによる動作順序の教示学習手法について議論する。

4.10.1 対象物指向型操縦の全身物体操作動作への拡張

本研究では、タンジブルインタフェースを遠隔物体操作に用いることを提案し、物体の接触状態・回転の有限状態機械に基づいて双腕物体操作の動作プランニングを実装した。この中でも特に 4.9.2.3 目で行った実験にて、Miniature Tangible Cube の操縦から PR2 は箱を抱え込み直して回転させるという抱え直し動作 (Reholding) を実現することができた。これは幾何・物理拘束の観点から見ると単腕による掴み直し動作 (Regrasping) とほぼ同等であり、これの双腕動作への拡張とみなせる。つまり、タンジブルインタフェースは 4.4.2 項でも述べたとおりに、物体の幾何・物理拘束に基づく動作探索プランニング [98] と組み合わせることで全身接触動作に拡張することができるため、4.9.2.3 目の抱え直し動作と同様に、タンジブルインタフェースによる同じ幾何・物理拘束の表現から、ロボット身体・対象物・環境に応じた様々な全身物体操作が実現できると期待する。

4.10.2 より器用な物体操作動作への拡張

本研究ではタンジブルインタフェースの実装として Miniature Tangible Cube を実装し、4.9 節に示すような双腕物体操作作業を行った。しかし、Miniature Tangible Cube は操縦者からの接触と回転のみを認識するため、このデバイスでは双腕間でのすべりを活用した双腕物体操作 [99] など、すべりを活用した双腕物体操作作業は行えない。しかし、小さな力センサ [93, 100] を Miniature Tangible Cube の表面に埋め込み、デバイス表面に加わるせん断応力を計測することで、操縦者からのすべり入力の含まれた指令を正しくロボットに表現できると考える。

4.10.3 形状変化可能なタンジブルインタフェース

本研究は、タンジブルインタフェースの実装として Miniature Tangible Cube を実装したが、4.4.2 項で述べた短所の通り、このデバイスは接触センサを周囲にシリコンゴムモールドしているため、形状を変更することは容易でない。本研究では、箱型の形状を扱うため Miniature Tangible Cube の形状は名前の通り立方体形状に設計したが、想定する対象物に応じて形状が変更できる、もしくは量産が容易いハードウェアであるべきである。しかし、タンジブルインタフェースを小型化かつ組み合わせ可能にすること [86, 87] やアクチュエータをユニット化し形状を変更する先行研究 [101] のように接触センサをユニット化することで様々な対象物に対応できるタンジブルインタフェースへと改良できると考えている。

4.11 おわりに

本章ではタンジブルインタフェースを物体操作作業の操縦実行に用いることを提案し、その長所と短所について議論し、この操縦実行における作業手順の教示学習の有効性について説明した。またタンジブルインタフェースの実装として、対象物の姿勢と接触状態を表現できるタンジブルインタフェース *Miniature Tangible Cube* を作成し、実装した *Miniature Tangible Cube* が対象物の姿勢と接触状態を表現できることを確認した。評価実験として *Miniature Tangible Cube* を用いた対象物指向型操縦システムを用いて双腕移動ロボット *PR2-JSK* と双腕ロボット *Baxter-JSK-SoftHand* による物体操縦実験を行い、単一物体と複数物体を用いた物体操作作業を操縦実行できることと、ぬいぐるみなどの複雑な形状をした物体は領域を直方体近似することで作業を操縦実行できること、把持動作を伴う物体操作作業を操縦実行できることから、タンジブルインタフェースの操縦実行における有効性を確認した。さらに *Miniature Tangible Cube* を用いた操縦実行によって教示された動作順序から物体操作作業を自律実行できることからタンジブルインタフェースによる作業手順の教示学習の有効性を確認した。

第 5 章

実世界適応による動作戦略の自律学習

5.1 はじめに

本章では物体操作作業における動作の学習型動作戦略とその実世界適応による動作戦略の自律学習と、それを用いた物体操作作業の自律実行について説明する。本研究における動作戦略の自律学習の目的は手順があり接触の多い複雑な物体操作作業において自律的に作業の動作の動作戦略を自律学習することであり、図 5.1 に示すように自律学習された動作戦略を用いて作業を操縦実行することで操縦自律間横断学習を実現する。本研究で取り組む学習型動作戦略は CNN 作業失敗予測器と CNN 把持点予測器であり、CNN 把持点予測器の自己教師あり学習において学習データをシミュレーションと実世界での把持試行を用いて自律的にアノテーションを行うことで自律学習を実現する。本研究ではこのシミュレーションによる事前学習と実世界試行による再学習を段階的に行う手法を実世界適応と呼ぶ。まず動作戦略と自律学習についてそれぞれ先行研究と比較しながら議論し、本研究で提案する学習型動作戦略とそのシミュレーションと実世界試行による実世界適応手法について説明する。最後に双腕ロボット Baxter-JSK-APC を用いて実世界適応による動作戦略の自律学習の有効性の評価実験を行う。

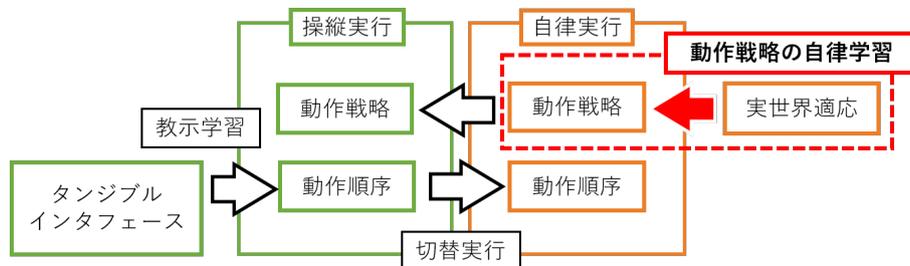


図 5.1: 操縦自律間横断学習における学習型動作戦略の自律学習の位置づけ

5.2 物体操作作業における動作戦略

物体操作作業において、状況に応じて対象物品を安定的に操作できるようロボットは複数動作から最適な動作を選択することで、これまで様々な物体操作作業を行ってきた [7, 8]. これは対象物品への力の作用点という観点から、実世界における物体操作作業が離散的にしか表現できないためである. つまりロボットは物体操作動作の評価関数を用いて対象物品に加える力の作用点の個数を決定する必要がある. この力の作用点数は離散的であり、双腕や全身も考慮した物体操作動作では接触点が複数となる. つまり物体操作作業において、ロボットは複数動作の接触状態などを評価し最適な動作を選択することが求められる. 近年では深層学習技術の発展から物体操作動作の動作戦略を学習的に獲得する手法が多く提案されている [13, 25, 43, 44, 47–49]. しかしこれらの手法は単腕把持など単純な物体操作作業のみにしか対応しておらず、複数の接触を有する物体操作動作については取り扱えていない. そこで本研究では、複雑な物体操作動作に対して学習型動作戦略について、CNN 失敗予測器、CNN 把持点予測器を用いた手法で取り組む.

5.3 動作戦略の実世界試行に基づく実世界適応

物体操作動作の学習型動作戦略について、学習データをどのように効率的に収集し学習するかが課題となり、多くの先行研究ではシミュレーションや操縦を活用したデータ収集手法 [38, 39, 44, 102–104] が行われており、計算機上のシミュレーションで環境を擬似的に作成し自動的にアノテーションを行う手法 [44, 102–104] が提案されてきた. これらの手法は人手のアノテーションと比べて労力が少ないが、接触の多い動作を行う際には実世界とシミュレーションのデータの乖離が問題となっており、データ拡張などを用いてこの乖離を埋める手法が提案されているが [44, 102, 104], 双腕物体操作動作などの接触の多い動作は実現できていない. また別の先行研究ではロボットを実際に動かすことで動作戦略を学習する手法が提案されており、これらはロボット実機が実世界で動作を行い、その動作結果を自らアノテーションする自己教師あり学習手法 (Self-supervised Learning) [13, 43] が提案されているが、これらの手法は実際にロボットが動いているデータを使っているため効果的であるが、接触の多い動作を行う際には実際に成功するデータを収集するための時間やコストが問題となっており、シミュレーションと自己教師あり学習を組み合わせた手法 [44] が提案されているが、双腕物体操作動作などの接触の多い動作は実現できていない. そこで本研究では、シミュレーションの一つである Image Stacking 手法 [102] を用いて学習型動作戦略の事前学習を行い、事前学習された動作戦略を用いて実世界で把持動作を行うことで自己教師あり学習によって学習型動作戦略を再学習するという実世界適応手法を提案し、この手法を用いて双腕物体操作作業の自律学習に取り組む.

5.4 CNN 作業失敗予測器を用いた学習型動作戦略

本節では棚への収納作業のための学習型動作戦略について説明する。本研究では棚への双腕収納作業を安定的に実現するために、作業失敗を少なくかつ効率的な動作戦略を構築する。動作戦略は作業失敗の発生する確率を予測することで、より作業失敗が起こりにくい動作を選択することで実現する。本研究では、CNNを用いた学習型作業失敗予測器を用いることで、作業失敗の経験から学習する動作戦略を構築する。本節で説明する学習型作業失敗予測器を用いた動作戦略の全体像について示す。本節で説明した動作戦略の全体像は図 5.2 に示すように、単腕・双腕収納動作のそれぞれを行った場合に関して評価を行い、よりよい動作を選択するものである。まず、初めに RGB 画像を入力とする CNN を用いて把持物体画像から単腕・双腕収納動作を行ったそれぞれの場合の作業失敗の発生確率を予測する。そして、その予測した作業失敗の発生確率を用いて作業スコアを計算することで作業の安定性を評価し、よりよい作業スコアを示す収納動作をロボットは選択・実行する。つまり、この動作戦略は RGB 画像を用いて把持物体の観測し、その状態から各収納動作についての作業失敗の発生確率を予測することで、各収納動作を行った場合の作業安定性を評価し、より高い作業安定性を示す収納動作を実行する戦略といえる。

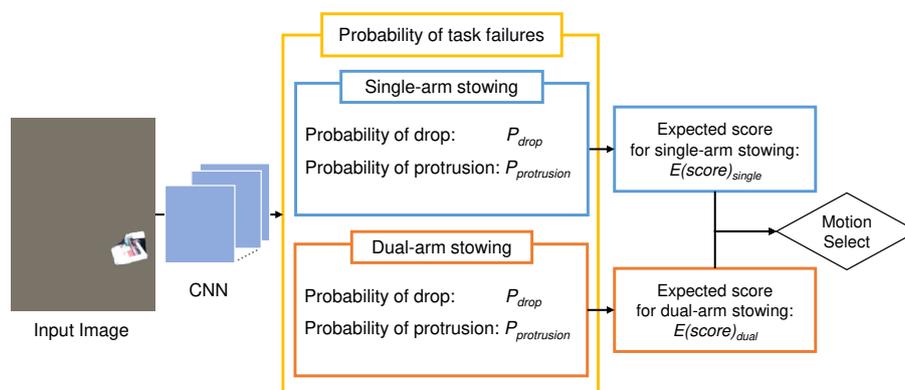


図 5.2: CNN 作業失敗予測器を用いた学習型動作戦略 [105]

5.4.1 倉庫環境における棚への収納作業

まず棚への収納作業について説明する。棚への収納作業とは指定された物品を棚へ収納する作業であり、これは倉庫環境で行われている作業である。まずこの作業における作業失敗とこの作業失敗を抑制するための双腕収納動作について説明する。

5.4.1.1 倉庫環境における棚への収納作業における作業失敗

本目ではまず棚への収納作業とその作業失敗について分類する。棚への収納作業は図 5.3a に示すような倉庫環境において、ロボットの前にあるトートに置かれた物品を棚に収納作業で

あり、把持した物体と棚の入り口の間での衝突が作業失敗の原因となりうる。把持物体と棚入り口の衝突や元々の把持が不安定である場合には、図 5.3b に示すような物体を落とす落下と、図 5.3c に示すような物体の一部が収納されないはみ出しの2つの作業失敗が発生しうる。そしてこの2つの作業失敗が発生しない場合には図 5.3d に示すように物体が収納された状態になり、これを本作業の作業成功とする。



(a) 棚への収納作業が行われる倉庫環境

(b) 棚への収納作業における落下



(c) 棚への収納作業におけるはみ出し

(d) 棚への収納作業における作業成功

図 5.3: 棚への収納作業における作業失敗 [105]

5.4.1.2 作業失敗を防ぐための棚への双腕収納動作

本目では前目で説明した棚への収納作業で起こりうる作業失敗を防ぐ双腕収納動作について説明する。本目の双腕収納動作は、まず対象物を単腕で把持した後にその物体をもう片方の腕で下から支える動作であり、これは人でいうところの手添え動作にあたる(図 5.4)。この双腕

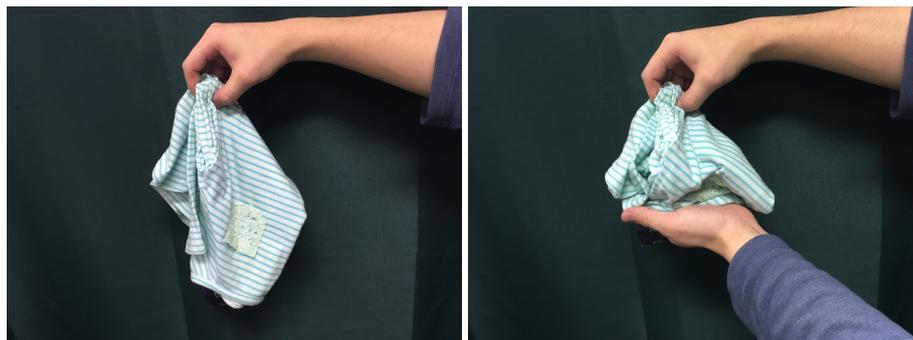
手添え動作は図 5.5 に示すように、物体を下方向から支えることで重力によって発生する垂れ下がりを抑えることで、物体の接触やはみ出しと言った作業失敗を防ぐ働きをする。しかし双腕収納動作は安定性が高い一方で、表 5.1 にもあるように実行時間は 67 秒と単腕収納動作の 52 秒よりも 15 秒も長くかかってしまう。そのためロボットは必要なときのみ双腕収納動作を行うことで、安定性高めつつ実行時間を短くすることを目的にする。このように選択的に単腕・双腕収納動作を行うことを選択的双腕収納動作という。

表 5.1: 双腕収納動作の実行時間

	Single-arm	Dual-arm
Average Execution Time	52 s	67 s



(a) 人による単腕把持動作(袋入りくつした場合) (b) 人による双腕手添え動作(袋入りくつした場合)



(c) 人による単腕把持動作(Tシャツの場合) (d) 人による双腕手添え動作(Tシャツの場合)

図 5.4: 単腕把持動作と双腕手添え動作 [105]

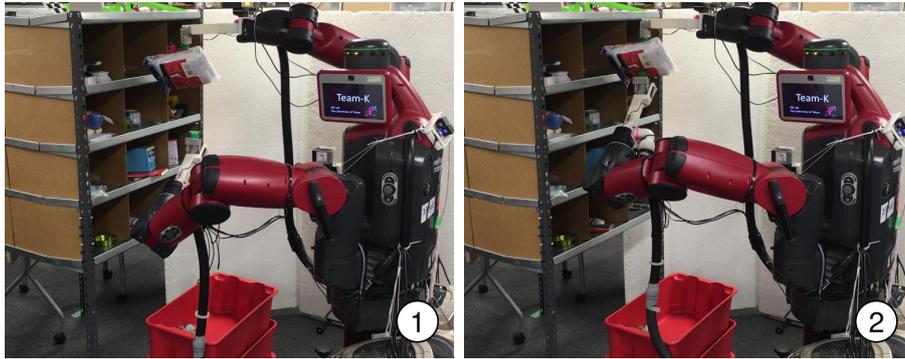


図 5.5: ロボットによる双腕手添え動作 [105]

5.4.2 CNN 作業失敗予測器

本項では CNN を用いた学習型作業失敗予測器について説明する．実際にロボットが作業を行い，その作業に関する経験から作業失敗予測器を学習することで，学習型動作戦略を構築する．具体的には画像を入力とする CNN を用いて，把持した物体の状態から単腕・双腕収納作業の両ケースに関する作業失敗の発生確率を予測し，その発生確率を元に作業安定性を評価し，より作業安定性の高い動作を選択して実行する，という動作戦略を構築する．

5.4.2.1 棚への双腕収納作業における学習型作業失敗予測器

本目では学習型作業失敗予測器について説明する．本研究で用いる学習型作業失敗予測器は AlexNet [106] をベースにした CNN であり，詳しい構造については図 5.6 に記す．この CNN は把持物体の画像を入力として，物体意味クラスと各作業失敗の発生確率を出力するものであり，動作戦略にはこの作業失敗の発生確率のみを用い，作業失敗予測としては単腕・双腕収納動作の各場合の落下とはみ出しの作業失敗の発生確率という 4 つの出力を得る．またこの CNN では学習高速化のために Batch Normalization [107] を全プーリング層の前で行い，過学習の回避のために Dropout [108] を出力層以外の全結合層を行う．この CNN を少ないデータ数のデータセットで学習するために，畳み込み層の初期重みは ImageNet のデータセットで学習された AlexNet [106] の重みを用い，CNN 学習の最適化手法として AdaGrad [109] を用いる，

5.4.3 CNN 作業失敗予測器の学習

本項では CNN 作業失敗予測器の学習について説明する．まずデータセットについて説明した後，このデータセットを用いた教師あり学習について説明する．

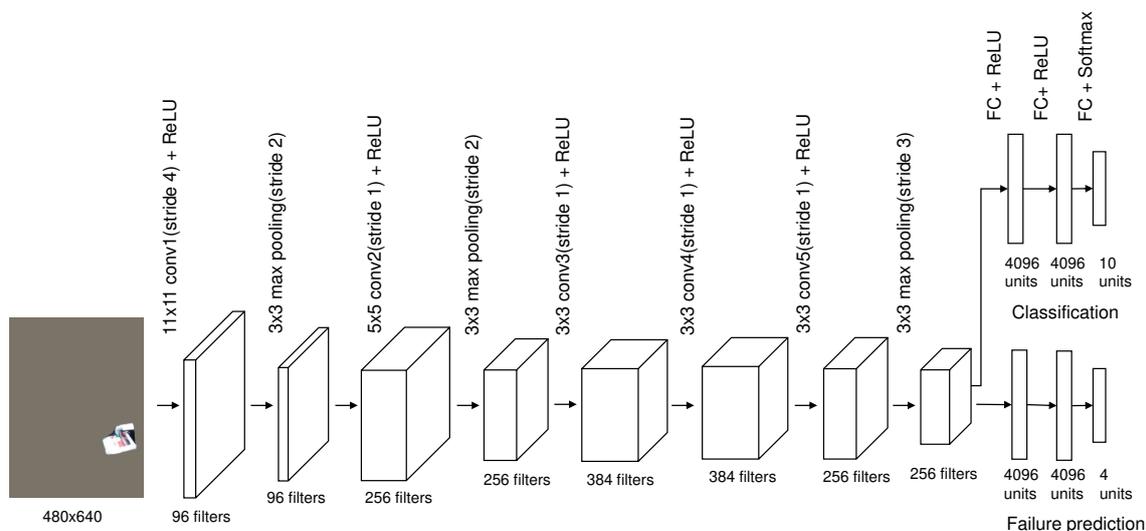


図 5.6: CNN を用いた棚への双腕収納作業の学習型作業失敗予測器 [105]

5.4.3.1 棚への双腕収納作業における作業失敗データセット

本目では、棚への双腕収納作業における学習型失敗予測器のための作業失敗データセットの作成手法について説明する。まず実世界においてロボットと作業環境において、双腕・単腕収納作業を物体 1 つずつに対して行い、その際の RGB 画像と作業失敗ラベルをデータ対として収集する。ロボットが物体を把持している際に図 5.7a に示すような RGB 画像を撮影し、深度画像と手先座標位置から物体領域マスクを自動生成して適応することで、図 5.7b に示すような物体とロボットグリッパのみを抽出した RGB 画像を生成する。このときマスクされた領域には ImageNet Dataset [106] の平均 RGB 値で埋めていき、640x480 の RGB 画像を入力画像として収集する。そして RGB 画像の対として、落下とはみ出しの 2 つの作業失敗について、2 値の作業失敗ラベルを $y_i \in \{0, 1\}$ をアノテーションする。このとき作業失敗が発生した場合には 1 を、発生しなかった場合には 0 をラベルとしてあたえ、単腕・双腕収納動作のうち実行されなかった動作については -1 のラベルをあたえ、合計 4 つのラベルを作業失敗ラベル y として収集する。実行されなかった動作について -1 を設定するのは、実世界のロボットを用いて単腕・双腕のどちらかの作業のみしか実行できないためである。

5.4.3.2 棚への双腕収納作業における学習型作業失敗予測器の学習

作業失敗予測として CNN 作業失敗予測器は以下の 4 つに関する発生確率を予測する。

- 単腕収納動作を行った場合の落下の発生確率
- 単腕収納動作を行った場合はみ出しの発生確率
- 双腕収納動作を行った場合の落下の発生確率
- 双腕収納動作を行った場合はみ出しの発生確率



図 5.7: 棚への双腕収納作業における 作業失敗データセットの RGB 画像 [105]

この時, CNN 学習のための損失関数の計算はソフトマックス交差誤差 (Softmax Cross Entropy, SCE) を用いて以下のように計算する.

$$L_{failure} = \frac{1}{N_{batch}} \sum_n^{N_{batch}} \sum_i^{N_{label}} \delta(y_i) \cdot SCE(y_i, \hat{y}_i) \quad (5.1)$$

この時, N_{batch} はバッチサイズ, N_{label} はラベル数, y_i は i 番目のラベルに関するホットベクタ, \hat{y}_i は i 番目のラベルに関する CNN の出力, $\delta(y_i)$ は以下のような y_i に関するデルタ関数である.

$$\delta(y_i) = \begin{cases} 1 & (y_i \in \{0, 1\}) \\ 0 & (otherwise) \end{cases}$$

なお本研究では, 単腕・双腕収納作業に対する落下・はみ出しの作業失敗に対する発生確率を予測するため, ラベル数は N_{label} は 4 となる. また実行されなかった動作に関する作業失敗ラベルは -1 がアノテーションされているため, 上記のデルタ関数 $\delta(y_i)$ は単腕・双腕収納動作のうちの実行された動作に関するラベルのみについて計算を行う関数となり, 損失 $L_{failure}$ はデータを収集した際に実行された収納動作に関するソフトマックス交差誤差 SCE とみなせる. つまり, バッチ数 N_{batch} が 1 のとき, 入力 RGB 画像が収集されたときに実行された収納動作に関する損失のみが計算・誤差逆伝播されるため, 学習された CNN は単腕・双腕収納作業の両方のケースについて作業失敗の発生確率を予測することができる. また物体意味クラスに関する損失 L_{class} は物体認識タスク同様にソフトマックス交差誤差 SCE で計算され,

$L_{failure}$ と L_{class} の 2 つの損失の合計が誤差逆伝播される．さらに入力 RGB 画像に対する過学習を防ぐために，入力画像に対して RGB 値の変更とフリップをデータセット拡張として行う．

5.4.4 CNN 作業失敗予測器を用いた動作戦略

本項では学習された CNN 作業失敗予測器を用いた動作戦略について説明する．本研究では CNN 作業失敗予測器の失敗予測結果に基づいて単腕・双腕収納動作から最適な動作を選択するアルゴリズムによって動作戦略を構築する．以下では CNN 作業失敗予測器の予測結果からどのように動作選択を行うためのアルゴリズムについて説明する．

5.4.4.1 棚への双腕収納作業における動作選択アルゴリズム

本目では，これまで説明してきた CNN 作業失敗予測器を用いた動作選択アルゴリズムについて説明する．これまで説明してきた CNN 作業失敗予測器は単腕・双腕収納動作の両ケースについて作業失敗の発生確率を予測する．本研究では作業の安定性の評価について作業スコアを設定し，この予測された作業失敗の発生確率を用いてこの作業スコアを計算して，よりよいスコアを示す動作を選択する動作選択アルゴリズムを提案する．各作業失敗に対して作業スコアの減点を設定し，その減点の大きさを用いてどの程度その作業失敗を回避してほしいかを人が決定できる．落下はロボットが物体を再度拾い上げることが出来ない場合があるため，致命的な作業失敗であるときみなすことができるため大きな減点を設定する．一方では見出しは，ロボットが認識することができ，かつあとから修正することができるため，致命的ではないと考え，落下よりも低い減点する．よって，棚への双腕収納作業における作業スコアの期待値 $E(score)$ は以下のように計算する．

$$E(score) = \max(0, 10 - 10 \cdot P_{drop} - 5 \cdot P_{protrude}) \quad (5.2)$$

このとき $P_{failure}$ は作業失敗 $failure$ の発生確率である．

そして，この計算された作業スコアを用いた Algorithm 5 に示す収納動作選択アルゴリズムを設計する．このアルゴリズムは基本的には作業スコアが大きい値を示す収納動作を実行する．しかし，もしも単腕収納動作に関する作業スコアが十分高く，かつ双腕収納動作の作業スコアと差がほぼない場合には，単腕収納動作を行っても作業失敗が起きにくいとみなして，ロボットは単腕収納動作を行う．

5.5 CNN 把持点予測器を用いた学習型動作戦略

本節ではピッキング作業のための CNN 把持点予測器を用いた学習型動作戦略について説明する．本研究ではピッキング作業を安定的に実現するために，把持成功率が高くなるような動作戦略を構築する．この動作戦略は単腕把持動作の把持点と双腕把持動作の把持点の中からより把持成功率が高い把持点と把持動作を選択することで実現する．本研究では CNN を用いた

Algorithm 5 収納動作選択アルゴリズム $E(score)_{single}$: Expected score of single-arm stowing $E(score)_{dual}$: Expected score of dual-arm stowing $threshold$: Threshold for $E(score)$ comparison**if** $E(score)_{dual} - E(score)_{single} > threshold$ **then**

Execute Dual-arm Stowing Motion

else if $E(score)_{single} < 5$ **then**

Execute Dual-arm Stowing Motion

else

Execute Single-arm Stowing Motion

end if

学習型把持点予測器を用いることで、把持動作の経験から自律的に学習する動作戦略を構築する。

5.5.1 倉庫環境におけるピッキング作業

本項では倉庫環境におけるピッキング作業について説明する。まずピッキング作業について分類したのちに、ピッキング作業における双腕把持動作の効果について説明する。

5.5.1.1 倉庫環境におけるピッキング作業の分類

複雑環境におけるピッキングタスクは図 5.8 に示す以下の 2 つに分類される。

- ランダムピッキング：環境内の物品全てを収納するタスク
- ターゲットピッキング：環境内のターゲット物品のみを収納するタスク

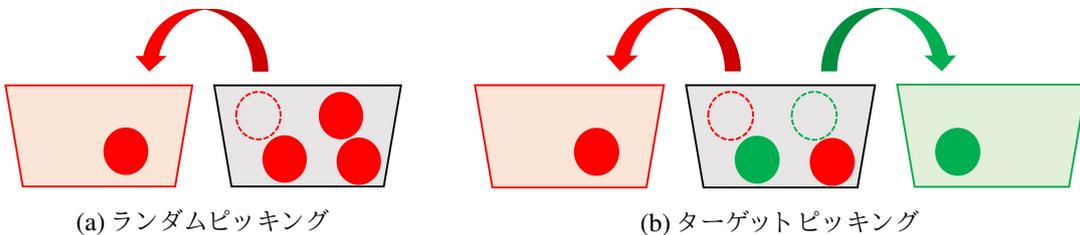


図 5.8: ランダムピッキングとターゲットピッキング

ランダムピッキングとは、図 5.8a に示すように環境に置かれた物品（図 5.8a 内の赤丸）を全て収納するタスクであり、これまで物品の意味ラベルに関係なく把持動作を行う研究が多く行われてきた [13, 25, 43, 47–49, 110]。これらの研究は多種多様な物品を把持するために RGB

画像や三次元情報を入力としたモデルによる学習手法が用いられており、高い把持成功確率を示している。一般的にランダムピッキングでは全ての物品を収納するため、RGB 画像や三次元情報のみを用いて未知の物品を適切な把持動作計画を行うことが目的とされている。しかし実際の倉庫内でのタスクを考えた際に複数の収納場所やタスク途中のタスク停止・再開などの状況に対応する必要があり、ランダムピッキングシステムであっても把持した物品の意味ラベルをロボットが正確に認識しておく必要がある。

ターゲットピッキングとは、図 5.8b に示すように環境内のターゲット物品（図 5.8b 内の赤丸）のみを収納するタスクであり、近年ランダムピッキングの学習手法の発展として研究が行われている [44, 111]。このタスクはターゲット物品をいかに効率的に収納するかが課題となり、ターゲット物品を移動する妨げになる障害物物品（図 5.8b 内の緑丸）を他の場所に移動することができる。またランダムピッキングは、環境内の物品全てがターゲット物品という状況でのターゲットピッキングと言い換えることができるため、ターゲットピッキングシステムを活用することでランダムピッキングを実行することは可能である。ターゲットピッキングにおいて重要なことは、ターゲット物品を正しく認識・把持・収納することであり、複雑環境で物品同士が重なりあった環境の中からターゲット物品を効率的に把持・収納する動作計画が必要になる。Scharz ら [112] や和田ら [111] は、複雑環境での物体の重なりあい（遮蔽領域）を認識しターゲット物品に重なっている物品（障害物物品）を移動することで、ターゲット物品を効率的に把持する動作計画を行う学習的認識手法を提案しており、その有効性は Amazon Robotics Challenge の競技対象である倉庫内複雑環境にて示されている。

以上で説明してきた研究では単腕ロボットのみを対象とされており、複雑環境での双腕ロボットによるピッキングシステムは研究が行われていない。本研究では双腕ロボットによるランダムピッキングとターゲットピッキングを行うロボットシステムを構成する。

5.5.1.2 倉庫環境における双腕把持動作

本項ではピッキングタスクにおける単腕・双腕把持動作の特徴について説明する。単腕・双腕把持動作のそれぞれの把持可能物品について分類すると表 5.2 のようになる。把持の安定性は物品の重心の位置に関係しており、物品の質量分布が不均一・不安定なものは単腕把持が不安定になる傾向がある。例えば本のような関節のある物品は把持動作中に重心が移動・振動するため単腕把持が不安定になる。また重い物品は単腕把持では把持が不可能・不安定になる傾向がある。一方で双腕把持は重い物品・重心が不安定な物品を安定的に把持できるが、双腕の自己干渉の問題から小さな物品を把持することはできない。以上から単腕・双腕把持動作それぞれに得意・不得意な物品が存在し、双腕ピッキングシステムでは物品に応じて単腕・双腕把持動作を使い分けることが求められる。

5.5.2 CNN 把持点予測器

本項では CNN 把持点予測器について説明する。本研究ではランダムピッキング作業とターゲットピッキング作業に対して、意味領域分割を同時に行う把持点予測器と、物体インスタ

表 5.2: 単腕・双腕把持動作の把持可能物品の分類 [35]

	Size	Weight	Object Stability
Single-arm graspable only	Small	Light	Balanced
Dual-arm graspable only	Large	Heavy	Unbalanced
Both graspable	Large	Light	Balanced

ス可視・遮蔽領域分割を同時に行う把持点予測器を用いる。

5.5.2.1 意味領域分割を同時に行う把持点予測器

本目では意味領域分割を同時に行う把持点予測器について説明する。本研究では把持点予測を把持可能領域の領域分割問題として定義し、図 5.9 に示すようなピクセル単位で意味領域と単腕・双腕把持可能領域の領域分割を行うモデルを設計する。この把持点予測器は FCN [113] と呼ばれるピクセル単位の意味領域分割のために考案された CNN をベースに設計しており、意味領域と把持可能領域のそれぞれの予測に対して共通の特徴量抽出器を用いて、意味領域分割と把持可能領域分割を同時に行う。

■5.5.2.1.1 FCN を用いた把持点予測器の設計 意味領域分割を同時に行う把持点予測器の全体構成は図 5.9 に示すように FCN32s [113] をベースに構成されており、特徴量抽出器として VGG16 [114] を用いる。入力である RGB 画像から VGG16 [114] の特徴量抽出器を介して得られる特徴量を、それぞれ意味領域分割ブランチと単腕・双腕把持の把持可能領域分割ブランチに入力し、それぞれのブランチでピクセル単位の意味ラベル c_{class} もしくは把持可能ラベル c_{grasp} の予測を行う。ピクセル単位の意味ラベル c_{class} とは、そのピクセルがどの物品の領域に属しているか、もしくはどれにも属さないかを示す値であり、具体的には背景を含む対象物品クラス集合のことである。ピクセル単位の把持可能ラベル c_{grasp} とは、そのピクセル位置に対して把持動作を行い、ロボットが物体を把持できるかどうかを示すものであり、具体的には以下の 2 つのラベルの集合である。

- 把持可能: *graspable*
- 把持不可能: *ungraspable*

以上の処理をまとめると、設計するモデルは RGB 画像を入力として意味領域、単腕把持の把持可能領域、双腕把持の把持可能領域についての 3 枚のラベル画像を予測する。意味領域分割は FCN [113] で定義されているように、あるピクセルが背景を含むそれぞれ対象物品（意味ラベル）に属する確率を予測する問題であり、本研究では把持可能領域分割も同様にあるピクセルが把持可能であるかの確率を予測する問題と定義し、図 5.9 に示すようにラベルの予測結果としては最も確率の高いラベルを出力する。また CNN の入力として RGB 画像を幅 640 ピ

クセル，高さ 480 ピクセルに拡大・縮小処理をして予測するため出力も同じ画像サイズで得られる．このとき拡大・縮小処理に伴うピクセル値の補間処理は 5.5.3.5 目で説明した処理を RGB 画像と正解データに対して行う．

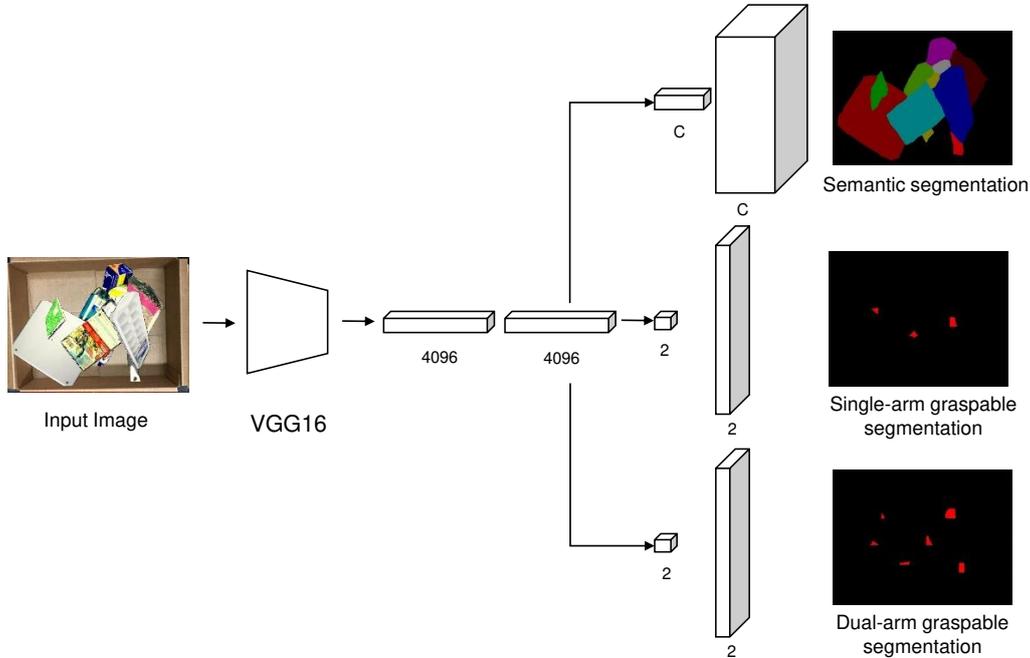


図 5.9: FCN を用いた把持点予測器 [36]

5.5.2.2 物体インスタンス可視・遮蔽領域分割を同時に行う把持点予測器

本節では物体インスタンス可視・遮蔽領域分割を同時に行う把持点予測器について説明する．本研究では把持点予測を把持可能領域の領域分割問題として定義し，図 5.11 に示すような各物体インスタンスに対してピクセル単位で可視・遮蔽領域と単腕・双腕把持可能領域の領域分割を行うモデルを設計する．この把持点予測器は Mask-RCNN [115] と呼ばれる物体インスタンス単位で意味領域分割するために考案された CNN をベースに設計しており，可視・遮蔽領域と把持可能領域のそれぞれの予測に対して共通の特徴量抽出器を用いて，物体インスタンス単位の可視・遮蔽領域分割と把持可能領域分割を同時に行う．まず Mask-RCNN の一部である Region Proposal Network(RPN) [116] について説明したのちに，本研究で提案するモデルの構成と学習について説明する．

■5.5.2.2.1 Region Proposal Network を用いた興味領域予測 まず Mask-RCNN の一部であり，興味領域予測を行う Region Proposal Network(RPN) [116] について説明する．興味領域とは，画像内における物体が存在しうる領域のことであり，背景ではない前景領域のことである．RPN は図 5.10 に示すように画像内の各ピクセルに対して複数の基礎矩形(Anchor) を設定することで，この基礎矩形内に前景が存在するかどうかの確率と，基礎矩形から興味領域

矩形への変形を予測する．基礎矩形 $a \in A$ に対して前景 fg か背景 bg かどうかの予測は，各ピクセルについて前景ラベル $C_{fgbg} = \{fg, bg\}$ のそれぞれのラベル $c_{fgbg} \in C_{fgbg}$ である確率を予測することで行う．次に基礎矩形 $a \in A$ に対する変形の予測については，基礎矩形 a の画像内における中心 x 座標 x_a ，中心 y 座標を y_a ，高さ h_a ，幅 w_a とし，予測結果の興味領域矩形 r の画像内における中心 x 座標 x ，中心 y 座標を y ，高さ h ，幅 w と設定すると，基礎矩形 a から興味領域矩形 r への変形 $o(a)$ は式 (5.3) のように定義して行う．

$$\begin{aligned}
 o_x(a) &= \frac{x - x_a}{w_a} \\
 o_y(a) &= \frac{y - y_a}{h_a} \\
 o_w(a) &= \ln\left(\frac{w}{w_a}\right) \\
 o_h(a) &= \ln\left(\frac{h}{h_a}\right)
 \end{aligned}
 \tag{5.3}$$

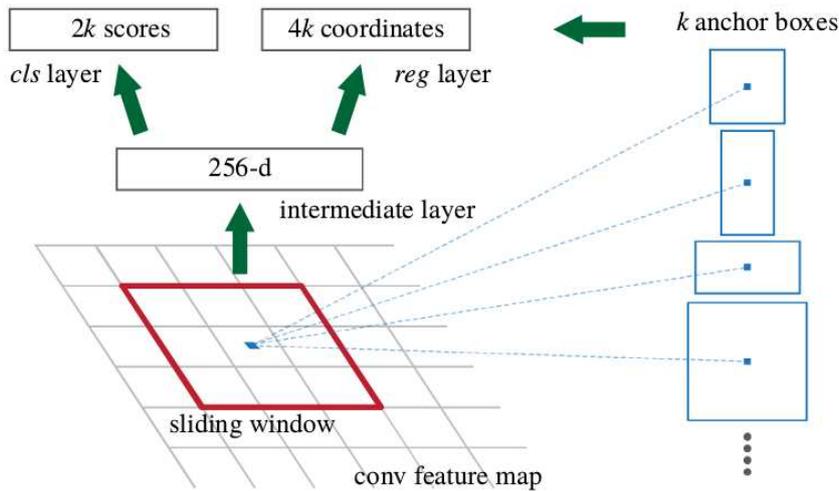


図 5.10: RPN による基礎矩形を用いた興味領域矩形の予測 [116]

■5.5.2.2.2 Mask-RCNN を用いた把持点予測器の設計 次に RPN を用いて物体インスタンス可視・遮蔽領域分割を同時に行う把持点予測器の設計を行う．物体インスタンス可視・遮蔽領域分割を同時に行う把持点予測器の全体構成は図 5.11 に示すように Mask-RCNN をベースに構成されており，RGB 画像を入力として特徴量抽出器 ResNet101 [117] を介して得た特徴量と RPN [116] を用いて，興味領域 (Region of Interest, RoI) を複数同時に予測し，各興味領域 r に対して物体領域矩形 b ，意味ラベル c_{class} ，ピクセル単位のインスタンス意味ラベル c_{inst} ，把持可能ラベル c_{grasp} の予測を行う．以下では提案するネットワークによる予測処理について詳しく説明する．

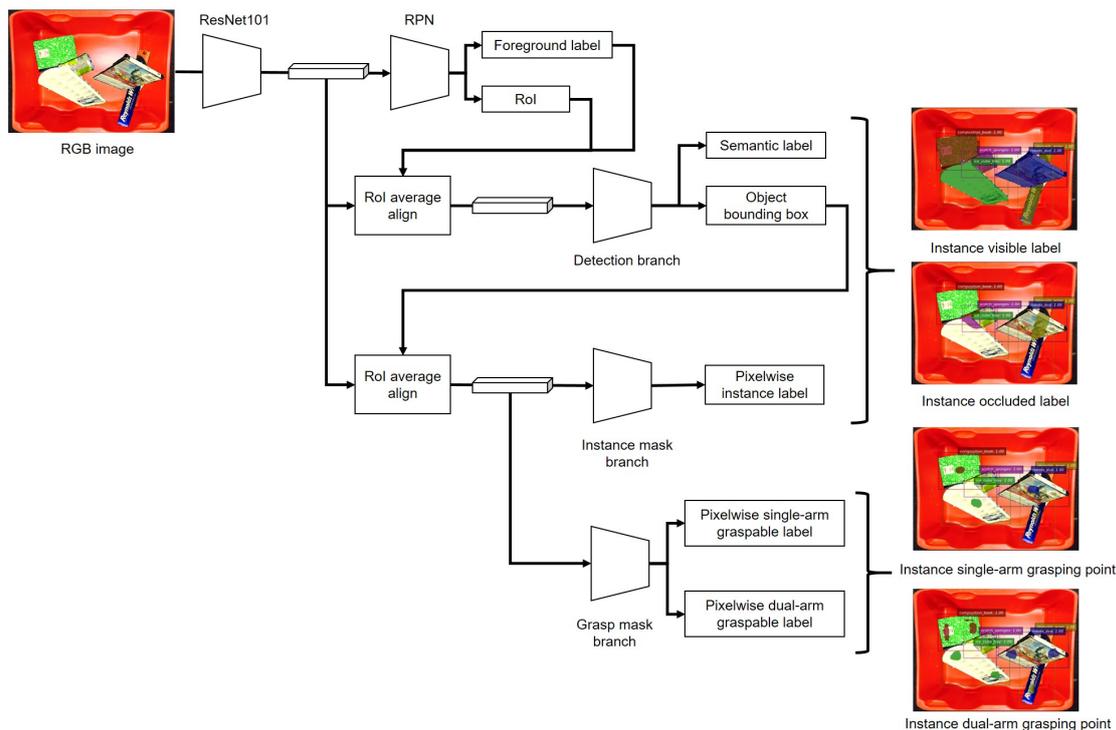


図 5.11: Mask-RCNN を用いた把持点予測器

入力である RGB 画像から ResNet101 [117] の特徴量抽出器を介して得られる特徴量に対して、まず RPN [116] によって興味領域の矩形 r を複数予測する．次に興味領域矩形 r 内に存在する物体インスタンスの意味ラベル c_{class} とその物体領域矩形の予測を行う．この際に用いる意味ラベル $c_{class} \in C_{class}$ は各興味領域内にどの物品が存在するか、もしくは存在しないかを示すものであり、 C_{class} は背景を含む対象物品クラス集合のことである．RGB 画像から得た特徴量に対して、興味領域矩形 r 内の特徴量のみに対して平均プーリング処理 (RoI Average Align [115]) を行い、得られる特徴量を更に畳み込み層に通すことで、興味領域矩形 r 内に存在する物体の意味ラベル c_{class} と物体領域矩形 b を予測する．この処理を各興味領域 r に対して並列に行うことで、画像内に存在する複数物体インスタンスの意味ラベル c_{class} と物体領域矩形 b の予測を同時に行う．

次に予測した物体領域矩形内の各ピクセルのインスタンス意味ラベル c_{inst} の予測を行う．この際に用いるインスタンス意味ラベル $c_{inst} \in C_{inst}$ とは、物体領域矩形 b 内のあるピクセルが物体インスタンスに属しているか、他物体との重なりで見えなくなっているか、もしくは背景であるかを示すものであり、具体的には以下の 3 つのラベルの集合 C_{inst} である．

- 物体インスタンス可視領域: vis
- 物体インスタンス遮蔽領域: occ
- 背景: bg

前段で予測した物体領域矩形 b 内の特徴量を再度平均プーリング処理 (RoI Average Align [115]) を行い、得られる特徴量に対して畳み込み層やデコンボリューション層に通すことで、物体インスタンス可視領域 vis , 物体インスタンス遮蔽領域 occ , 背景 fg の3つのインスタンス意味ラベル c_{inst} のどれに物体領域矩形 b 内の各ピクセルが属するかを予測する。この処理も各物体領域矩形 b に対して並列に行うことで、画像内に存在する複数物体インスタンスのピクセル単位のインスタンス意味ラベル c_{inst} の予測を同時に行う。

最後に予測した物体領域矩形 b 内の各ピクセルに対して、単腕・双腕把持に対する把持可能ラベル $c_{grasp} \in C_{grasp}$ の予測を行う。この際に用いる把持可能ラベルとは、そのピクセル位置に対して把持動作を行い、ロボットが物体を把持できるかどうかを示すものであり、具体的には以下の2つのラベルの集合 C_{grasp} である。

- 把持可能: *graspable*
- 把持不可能: *ungraspable*

前段で予測した物体領域矩形 b 内の特徴量を再度平均プーリング処理 (RoI Average Align [115]) を行い、得られる特徴量に対して畳み込み層やデコンボリューション層に通すことで、物体領域矩形 b 内の各ピクセルについて把持可能ラベル c_{grasp} を予測する。この処理も各物体領域矩形 b に対して並列に行うことで、画像内に存在する複数物体インスタンスのピクセル単位の把持可能ラベル c_{grasp} の予測を同時に行う。

以上の処理をまとめると、設計するモデルは RGB 画像を入力として、複数の物体領域矩形 b とそれらの意味ラベル c_{class} , そして各物体領域矩形内に存在するピクセルのインスタンス意味ラベル c_{inst} と把持可能ラベル c_{grasp} を予測する。ピクセル単位のインスタンス意味ラベル c_{inst} と把持可能ラベル c_{grasp} の予測は FCN [113] 同様に、あるピクセルが各ラベルに属する確率をそれぞれ予測する問題として定義し、ラベルの予測結果としては最も確率の高いラベルを出力する。また CNN の入力画像としては画像の縦・横両辺が最小 600 ピクセル、最大 1000 ピクセルになるようにアフィン変換を行い、この時 5.5.3.5 目で説明したピクセル値の補間処理を RGB 画像と正解データについて行う。

5.5.3 シミュレーションを用いた CNN 把持点予測器の事前学習

本節ではシミュレーションを用いた動作戦略の事前学習について説明する。この事前学習は図 5.12 に示すように、まず各物体が1つずつ撮影された物体インスタンス画像をアノテーションし、この画像を前景抽出したのちに背景画像に貼り合わせることで物体把持データセットを自動生成し、このデータセットを元に CNN 把持予測器を学習するというものである。

自動データセット生成のために、まず物体インスタンス画像に対して物体インスタンス形状の前景抽出を行い、抽出されたインスタンス形状に基づいて自動で把持可能領域のアノテーションを行う。物体インスタンス画像とは図 5.13a に示すように、物体が1つずつに撮影された画像であり、まず各物体に対してそれぞれ4から6枚ほど用意する。そして、把持可能領域の自動アノテーションとして、単腕・双腕把持についての把持可能領域を物体インスタンス

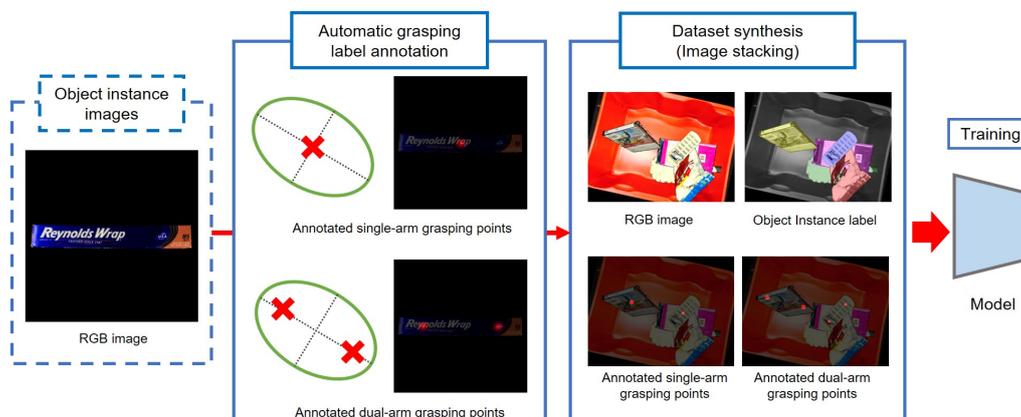


図 5.12: シミュレーションを用いた動作戦略の事前学習 [35]

形状から推定し，この領域を図 5.14 [36] に示すように把持点として把持可能領域をアノテーションしていく．

5.5.3.1 物体把持データセットのためのアノテーションラベル

物体は次データセットのために，本研究では以下の 3 つのピクセルレベルのラベルを設定する．

- 把持ラベル: c_{grasp}
- 物体意味ラベル: c_{class}
- 物体インスタンスラベル: c_{inst}

把持ラベル c_{grasp} はロボットが画像上のピクセルの位置を把持することができるかどうかを示しており，把持ラベル集合 C_{grasp} は以下の 2 つのピクセル領域ラベルで構成されている．

- 把持可能ラベル(把持可能領域) : $graspable$
- 把持不可能ラベル(把持不可能領域) : $ungraspable$

物体意味ラベル c_{class} は画像上のピクセルが属する物体の意味ラベルを示しており，物体意味ラベル集合 C_{class} は背景を含む対象物の意味ラベルの集合で構成されている．物体インスタンスラベル c_{inst} は画像上のピクセルが物体インスタンスの可視な部分，遮蔽された部分に属しているかを示しており，物体インスタンスラベル集合 C_{inst} は以下の 3 つのピクセル領域ラベルで構成されている．

- インスタンス可視ラベル(インスタンス可視領域) : vis
- インスタンス遮蔽ラベル(インスタンス遮蔽領域) : occ
- 背景ラベル(背景領域) : bg

5.5.3.2 多種物品のインスタンス画像における前景抽出

まず対象物品領域を獲得するために対象物品のインスタンス画像の前景抽出を行い，図 5.13a に示すような RGB 画像を入力として図 5.13b のような対象物品領域をインスタンスマスクとして得る．



図 5.13: 対象物品のインスタンス画像と前景抽出による対象物品領域 [35]

対象物品を黒背景で撮影して対象物品のインスタンス画像を得て，各ピクセル値に閾値を設けることで前景抽出を行うことができる．しかしこれでは黒い部分のある物品や透明な物品については対象物品領域をうまく得ることができない．そのため本研究では和田ら [111] によって提案された深層学習を用いた前景抽出モデルで前景抽出を行う．この前景抽出モデルは FCN [113] を基に設計されており，画像上の各ピクセルについて前景と背景の 2 クラス分類を行い，その分類に基づいて前景領域にインスタンス可視領域 vis ，背景領域に背景領域 bg のインスタンスラベル c_{inst} をアノテーションする．またこの前景抽出モデルの学習には教師データとして閾値によって作成した対象物品領域を与えるため，人のアノテーションの必要なく対象物品のインスタンス画像のみで前景抽出モデルを学習することができる．

5.5.3.3 物品の幾何形状に基づく自動把持点アノテーション

自動把持点アノテーションのために，本研究では物体形状に基づく把持点アノテーションアルゴリズムを設計する．本研究で提案する把持点アノテーションアルゴリズムでは，それぞれの対象物品に対して 6 枚以下の RGB 画像を用いて，単腕・双腕把持における把持点を物体の幾何形状から図 5.14 に示すように計算する．用いる対象物品のインスタンス画像はそれぞれ対象物品の上方，下方，左方，右方，そして一部物品では斜め方向から撮影する．本研究で設計する把持点アノテーションアルゴリズムでは，対象物品の重量分布は物品全体で一定であることと，対象物品のどの部分も把持可能であることを仮定するため，把持点アノテーションの際には対象物品の幾何形状のみを考慮するだけでよく，把持点は 2 次元画像内の対象物品領域から計算される．

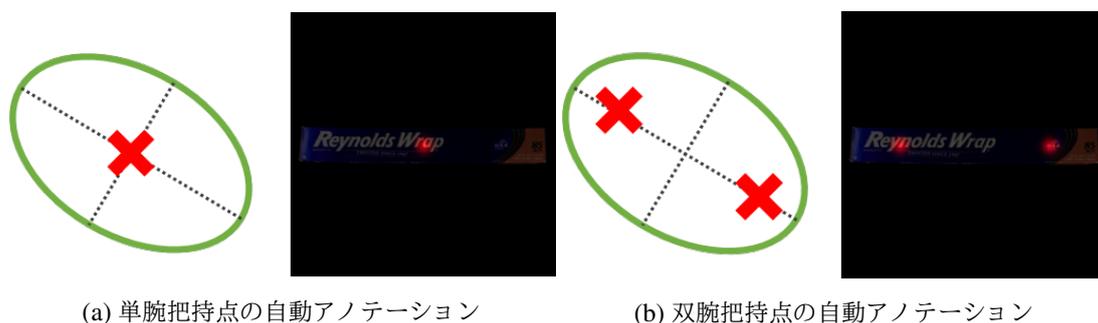


図 5.14: 単腕・双腕把持のための自動把持点アノテーション [35]

単腕・双腕把持それぞれの把持点アノテーションの詳細なアルゴリズムについて説明する。まず単腕把持における把持点は前景抽出によって得られた対象物品領域の中心点に、図 5.14a に示すように把持可能 *graspable* を把持ラベル $c_{graspable}$ としてアノテーションし、その他の領域を把持不可能領域 *ungraspable* とアノテーションすることで把持点画像を生成する。双腕把持における把持点は、得られた対象物品領域について主成分分析を行った際の第一主成分上に整列されているとして、図 5.14b に示すように把持可能領域 *graspable* を把持ラベル $c_{graspable}$ としてアノテーションし、その他の領域を把持不可能領域 *ungraspable* とアノテーションを行う。具体的には、対象物品領域のピクセル位置 (x, y) について主成分分析を行い、その第一主成分上かつ物体領域中心から点対称に、双腕把持点がそれぞれ第一主成分の長さの 80% にあたる距離はなれているようにアノテーションを行う。このようにして単腕・双腕把持の把持点画像は自動データセット生成に用いられる。

5.5.3.4 Image Stacking を用いた把持点予測データセット自動生成

本項では、自動的にアノテーションされた把持点と対象物品のインスタンス画像とその前景抽出を用いて、乱雑環境における意味領域分割と把持点予測のためのデータセットを生成する手法について説明する。本研究では、図 5.15 に示すように把持点データセットの自動生成に画像貼り合わせ手法 (Image Stacking) [102] を応用することで、乱雑環境の背景画像に対して対象物品のインスタンス RGB 画像を貼り合わせ、それに対応する物体意味ラベル c_{class} 、物体インスタンス可視領域 vis 、単腕・双腕把持の把持可能領域 *graspable*、そして他物品との重なりによって生じる物体インスタンス遮蔽領域 occ をアノテーションすることでデータセットを自動生成する。図 5.16 は生成したデータセットの一例であり、縦方向には上から順に物体インスタンス可視領域、物体インスタンス遮蔽領域、単腕把持点、双腕把持点のアノテーションが可視化されている。また横方向に 3 つのシーンが並べられており、左端のシーンから右隣に新たな物品が画像内に 1 つ追加されている。以下では自動データセット生成について詳しく説明する。

インスタンスラベルに着目すると、自動データセットは図 5.16 に示すように複数の対象物品インスタンスの RGB 画像を変形させながら 1 枚の背景画像に貼り合わせることで、複数物

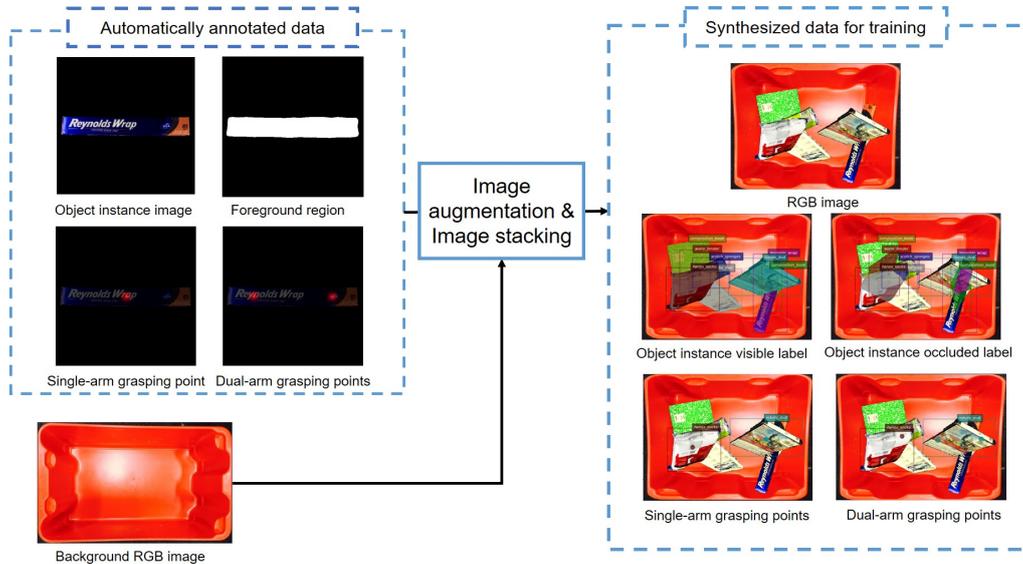


図 5.15: Image Stacking を用いた把持点予測データセット 自動生成 [35]

品が重なりあい遮蔽しあう複雑環境の RGB 画像を自動生成する。また各対象物品の RGB 画像に対応する物体インスタンス可視領域を RGB 画像と同様に變形し貼り合わせることで、画像ピクセル単位の物体インスタンス可視領域 vis とインスタンス遮蔽領域 occ をそれぞれ図 5.16 の上から 1 段目と 2 段目に示すようにアノテーションする。物体インスタンス可視領域 vis については前景抽出の際に得られた前景マスクと対象物品の物品ラベルを用いてアノテーションし、画像貼りあわせによって他物品によって覆いかぶさられた領域をインスタンス遮蔽領域 occ としてアノテーションする。このとき、他物品の重なりによってある物品の可視領域が画像内から見えなくなった場合には、その見えない物品は認識不可能であるとしてアノテーションしない。

同様にして単腕・双腕把持の把持点に関してもそれぞれ図 5.16 の上から 3 段目と 4 段目に示すようにアノテーションを行うが、他物品に対象物品領域の 10% 以上が覆いかぶさっている場合は物理的に対象物品が遮蔽されており、把持不可能 *ungraspable* であるとみなし遮蔽された対象物品の把持点のアノテーションを行わない。つまり本研究では対象物品の 90% 以上の領域が他物品に覆いかぶさられることなく視覚的に観測可能である場合には、その物品は乱雑環境において把持可能 *graspable* であるとみなす。また遮蔽領域についても複数物品のインスタンス領域を RGB 画像と同様に變形し貼り合わせることでアノテーションを行う。本項で自動生成した物体インスタンス可視・遮蔽領域を含む把持点予測データセットを用いて各把持点予測器の学習を行う。

5.5.3.5 データセット自動生成におけるデータ拡張

5.5.3 項で説明したデータセット自動生成における対象物品のインスタンス画像を貼り合わせる際に、各インスタンス画像に対してデータ拡張をすることで複雑環境を再現する。本項で



図 5.16: 生成した単腕・双腕把持の把持点と物体インスタンス可視・遮蔽領域データセットの可視化

はこのデータ拡張について詳しく説明する．RGB 画像とそれに対応する物体インスタンス可視領域 *vis*，インスタンス遮蔽領域 *occ*，単腕・双腕把持の把持可能領域 *graspable* について，50% の確率で左右鏡面処理と 0.5 から 2.0 の割合での拡大・縮小， -180° から 180° の回転処理を行い，画像貼りあわせの際には背景画像からはみ出ない範囲でランダムに配置する．更に RGB 画像に対しては 30% の確率で，RGB 画像を HSV 形式に変換した後に S, V の値を 0.5 から 2.0 の割合で乗算し，RGB 画像の R, G の値に対して 1.0 から 1.5 の割合で乗算する．ま

た拡大・縮小処理には RGB 画像に対しては Bilinear Interpolation を行い、その他のアノテーションデータについては Nearest Interpolation でピクセル値の補間を行う。これらのデータ拡張は実験環境の照明条件やカメラの撮影位置などの変化に対応するためのものである。

5.5.3.6 学習時の把持点予測データセットに対するデータ拡張

5.5.3 項で生成したデータセットを用いて把持点予測器を学習する際に、モデルの予測を環境変化などに対して頑強にするために、入力画像とそれに対応するアノテーションデータについてデータ拡張を行う。5.5.3.5 目で説明したデータ拡張は、データセット生成の際に各対象物品のインスタンス画像について処理を行っているのに対して、本項で説明するデータ拡張は画像貼りあわせ後の複雑環境をシミュレーションした画像について処理を行う。学習時のデータセットに対するデータ拡張として、RGB 画像とそれに対応する物体インスタンス可視領域 vis 、インスタンス遮蔽領域 occ 、単腕・双腕把持の把持可能領域 $graspable$ について、50% の確率で左右鏡面処理と -180° から 180° の回転処理を行う。また RGB 画像の RGB 値については 5.5.3.5 目で説明したものと同一 HSV 値、RGB 値の変更処理を行う。これらのデータ拡張は 5.5.3.5 目同様、実験環境の照明条件やカメラの撮影位置などの変化に対応するためのものである。学習時のデータ拡張を行った把持点予測データセットの可視化は図 5.17 に示す。

5.5.3.7 自動生成した把持点予測データセットと実世界の違い

5.5.3 項に示した手順で把持点予測データセットを自動生成したが、これらは RGB 画像から抽出した前景画像の幾何形状を元にアノテーションを行っている。しかし、把持点の自動アノテーションアルゴリズムは実世界では把持可能であるとは限らず、また逆にアノテーションされていない場所が把持可能であることもある。特に小さい物体はロボットの自己干渉のために双腕把持を実行できない場合、重い物体を単腕把持することはできない場合や質量分布が均等ではない物品を安定的に単腕把持できる把持点が他に存在する場合が発生する。本研究では、自動生成した把持点予測データセットは完璧に実世界を再現できないがある程度再現していると仮定して、自動生成データセットをヒントとして把持点予測器をまず学習した後に実世界把持経験でモデルを再学習するというアプローチをとる。

5.5.3.8 FCN を用いた把持点予測器の学習

本項では、5.5.2.1 目で説明した FCN を用いた把持点予測器の学習について説明する。5.5.2.1 目でも説明したように、この把持点予測器は意味領域分割と把持可能領域分割を同時に行うため、意味領域分割のための損失 L_{seg} 、単腕・双腕把持の把持可能領域分割のための損失 L_{single} , L_{dual} について順に説明する。

はじめにネットワークの出力 h 、正解データの One-hot Vector 表現 l 、ラベル集合 C に関するソフトマックス交差誤差 (Softmax cross entropy, SCE) は式 (5.4) のように定義される。

$$SCE(h, l, C) = - \sum_{c \in C} \ln(\text{Softmax}(h^c)) \cdot l^c \quad (5.4)$$

各ピクセルについてソフトマックス交差誤差 SCE を用いて意味領域分割のための損失 L_{seg}

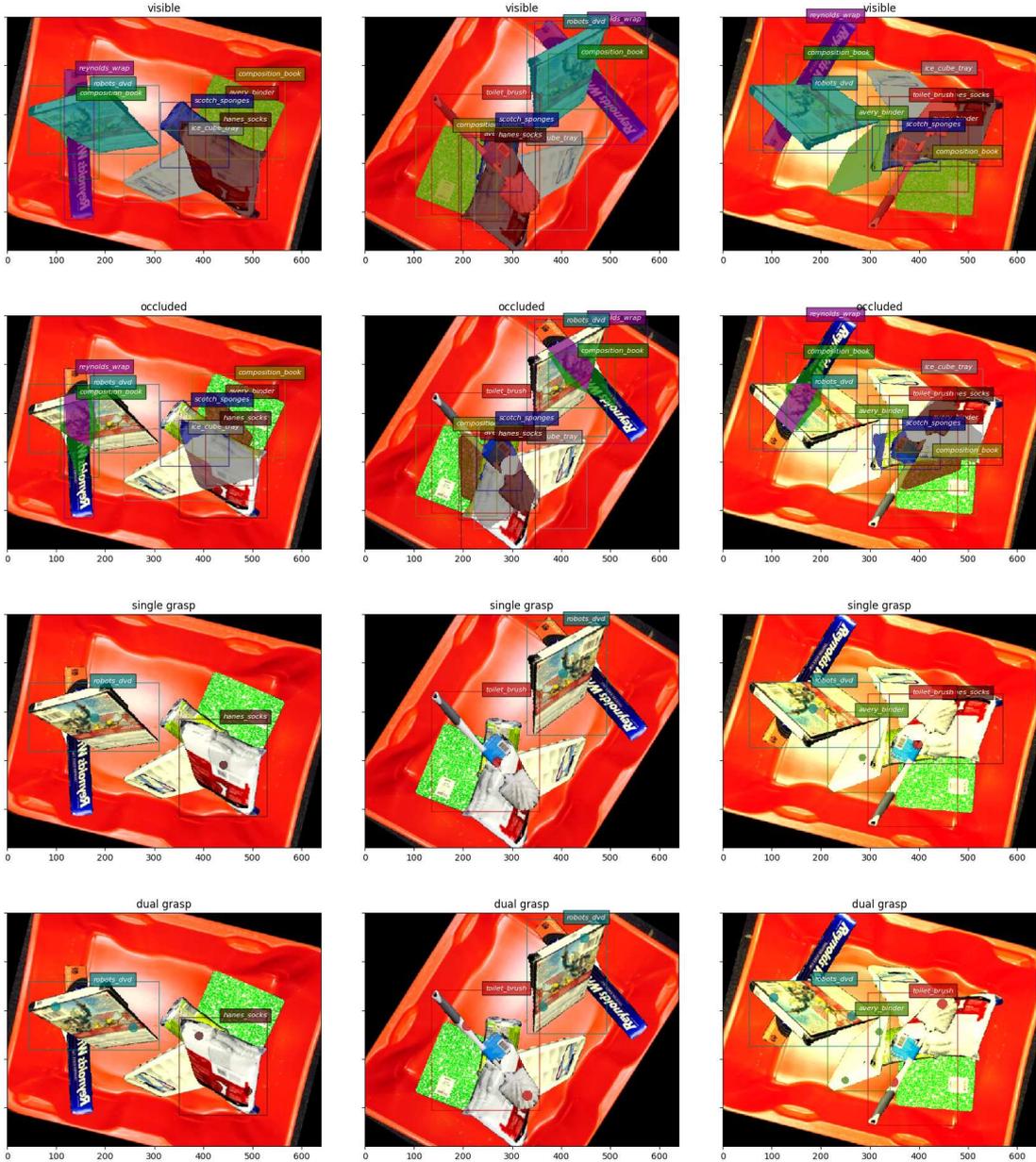


図 5.17: 学習時のデータ拡張後の把持点予測データセットの可視化

は式 (5.5) のように計算する. 式 (5.5) において, C_{seg} は意味ラベルの集合, W は画像の幅, H は画像の高さ, $l_{seg}(x, y)$ はピクセル (x, y) における意味ラベルに関する正解データの One-hot Vector 表現であり, 正解であれば $l_{seg}^{c_{seg}} = 1$, それ以外は $l_{seg}^{c_{seg}} = 0$ の値を示す. $h_{seg}(x, y)$ はピクセル (x, y) における意味ラベルに関するモデルの出力を示す.

$$L_{seg} = \sum_{x=1, y=1}^{W, H} SCE(h_{seg}(x, y), l_{seg}(x, y), C_{seg}) \quad (5.5)$$

この損失 L_{seg} は FCN [113] の意味領域分割でも用いられており、モデルは各ピクセルについて正しい意味ラベルを予測するように学習する。

把持可能領域については、本研究では単腕・双腕把持のどちらについても同じ損失関数を用いて学習する。ここで注意すべきは、図 5.16 に示すように把持可能領域は把持できない領域と比べて小さいため、意味領域分割の損失関数とは異なり、把持可能ラベル $graspable$ と把持不可能ラベル $ungraspable$ について出現頻度に関する重みをつける必要がある。出現頻度の少ないラベルを領域分割する研究として、出現頻度の低いラベルの損失を大きくし、出現頻度の高いラベルの損失を小さくする Frequency balancing [118] を応用し、把持戦略 $s_{grasp} \in S_{grasp}$ と把持ラベル $c_{grasp} \in C_{grasp}$ についての重み $w_{s_{grasp}}^{c_{grasp}}$ を式 (5.6) のように計算する。ここで把持ラベル c_{grasp} は、把持可能 $graspable$ か把持不可能 $ungraspable$ かの 2 種類であるため、把持ラベルの集合 C_{grasp} は $\{graspable, ungraspable\}$ であり、把持戦略は単腕・双腕把持の 2 種類であるため、把持戦略の集合 S_{grasp} は $\{single, dual\}$ である。式 (5.6) において $N_{foreground}$ は教師データの正解ラベル画像における背景ラベルではないピクセル数、 $N_{c_{grasp}}$ は教師データの正解ラベル画像における把持ラベル c_{grasp} であるピクセル数、 $\alpha_{c_{grasp}}$ は把持ラベル c_{grasp} についての定数である。

$$w_{s_{grasp}}^{c_{grasp}} = \begin{cases} \frac{N_{foreground}}{\alpha_{c_{grasp}} \cdot N_{c_{grasp}}} & (N_{c_{grasp}} \neq 0) \\ 0 & (N_{c_{grasp}} = 0) \end{cases} \quad (5.6)$$

本研究では把持ラベルの定数としては $\alpha_{graspable}$ は 20.0、 $\alpha_{ungraspable}$ は 1.0 に設定する。

次にソフトマックス交差 SCE を拡張し、ネットワークの出力 h 、正解データの One-hot Vector 表現 l 、ラベル集合 C 、重み w に関する重み付きソフトマックス交差 (Weighted Softmax Cross Entropy, $WSCE$) を式 (5.7) のように定義する。

$$WSCE(h, l, w, C) = - \sum_{c \in C} \ln(\text{Softmax}(h^c)) \cdot l^c \cdot w^c \quad (5.7)$$

式 (5.6) と式 (5.7) で定義した重み $w_{s_{grasp}}$ と重み付きソフトマックス交差誤差 $WSCE$ を用いて、把持可能領域分割のための損失 $L_{s_{grasp}}$ は $s_{grasp} \in S_{grasp}$ のそれぞれについて式 (5.8) のように計算する。式 (5.8) において、 C_{grasp} は把持ラベルの集合、 W は画像の幅、 H は画像の高さ、 $l_{s_{grasp}}$ はピクセル (x, y) における把持戦略 s_{grasp} に関する正解データの One-hot Vector 表現であり、正解であれば $l_{s_{grasp}}^{c_{grasp}} = 1$ 、それ以外は $l_{s_{grasp}}^{c_{grasp}} = 0$ の値を示す。 $h_{s_{grasp}}(x, y)$ はピクセル (x, y) における把持戦略 s_{grasp} に関するモデルの出力を示す。

$$L_{s_{grasp}} = \sum_{x=1, y=1}^{W, H} WSCE(h_{s_{grasp}}, l_{s_{grasp}}, w_{s_{grasp}}, C_{grasp}) \quad (5.8)$$

式 (5.5) と式 (5.8) で定義した意味領域分割と把持可能領域分割の損失を用いて、提案する把持点予測器の学習を行う。モデル全体の学習のための全体損失 L_{total} は、意味領域分割の損失 L_{seg} と単腕・双腕把持の把持可能領域の損失 L_{single} と L_{dual} の和として式 (5.9) のように計算する。

$$L_{total} = L_{seg} + L_{single} + L_{dual} \quad (5.9)$$

把持点モデルの学習の際には、式 (5.9) で計算された損失を図 5.18 の赤点線に示すように誤差逆伝播することで畳み込み層のパラメータを更新する。このとき更新するのは畳み込み層のみでありデコンボリューション層の更新は行わない。把持点予測器の学習に際して、効率的な学習を行うために特徴量抽出器については学習済みモデルの重みを利用する。特徴量抽出器の VGG16 [114] は ImageNet [119] で学習した重みを初期重みとして学習を行い、その他の畳み込み層のパラメータは 0 を初期重みとして与える。デコンボリューション層の初期重みについては Bilinear Interpolation [113] で計算された値を設定し学習では更新しない。また学習結果を安定化するために 5.5.3.6 目で説明したデータ拡張を行う。学習の最適化には Adam [120] を用いて初期学習率 $1.0e-5$ に設定して学習を行い、自動生成した約 20000 データペアのデータセットに対してバッチサイズ 1 枚かつ 200000 イテレーションで学習を行う。

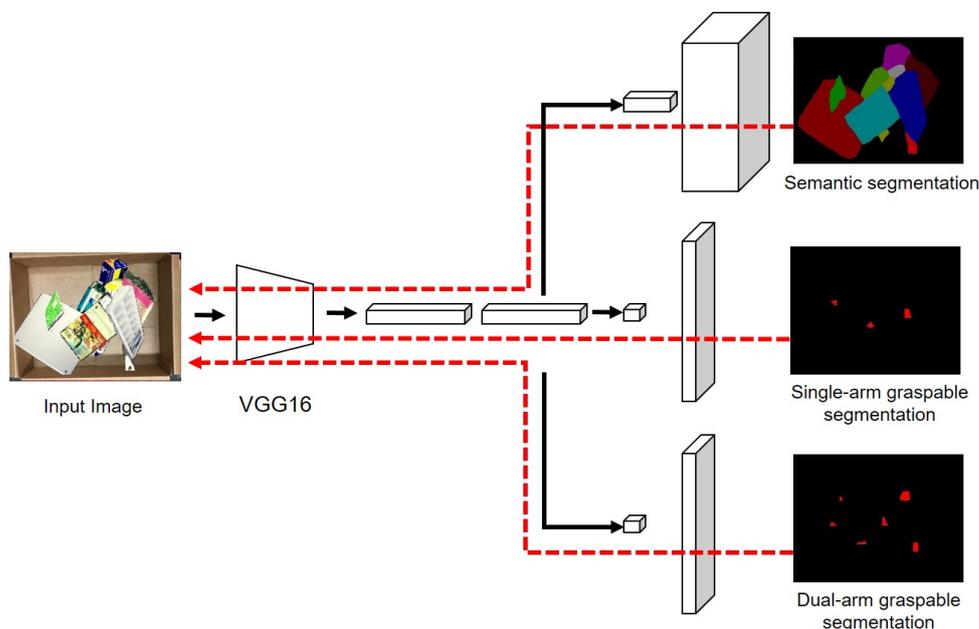


図 5.18: FCN を元にした把持点予測器の最初の学習における誤差逆伝播

5.5.3.9 Mask-RCNN を用いた把持点予測器の学習

本項では 5.5.2.2 目で説明した Mask-RCNN を用いた把持点予測器の学習について説明する。5.5.2.2 目で説明したとおり、この把持点予測器は物体インスタンス可視・遮蔽領域分割と把持可能領域分割を同時に行うため、RPN による興味領域予測に関する損失 L_{rpn} 、物体インスタンス検出に関する損失 L_{det} 、物体インスタンス可視・遮蔽領域分割に関する損失 L_{inst} 、把持可能領域分割に関する損失 L_{single}, L_{dual} について順に説明する。

まず RPN による興味領域予測に関する損失 L_{rpn} について説明する。RPN は基礎矩形 $a \in A$ に対して、基礎矩形 a に関する前景ラベル $c_{fgbg} \in \{fg, bg\}$ の予測出力 $h_{fgbg}(a)$ が正解データ $l_{bgfg}(a)$ を予測し、基礎矩形 a から興味領域矩形への変形 $o(a)$ が正解データ $\hat{o}(a)$ を予測するよう学習する。そのため RPN の損失 L_{rpn} は、前景ラベル c_{fgbg} の予測に関する損

失 L_{fgbg} と基礎矩形の変形に関する損失 L_{loc} によって構成される．このとき前景ラベル集合 $C_{fgbg} = \{fg, bg\}$ について，前景ラベル予測に関する損失 L_{fgbg} はソフトマックス交差誤差 SCE (式(5.4))を用いて，式(5.10)に示すように計算する．式(5.10)において， $l_{fgbg}(a)$ は基礎矩形 a における前景ラベルに関する正解データ One-hot Vector 表現であり， $|A|$ は基礎矩形集合 A に属する基礎矩形数を表す．

$$L_{fgbg} = \frac{1}{|A|} \sum_{a \in A} SCE(h_{fgbg}(a), l_{fgbg}(a), C_{fgbg}) \quad (5.10)$$

次に基礎矩形から興味領域矩形への変形に関する損失 L_{loc} について説明するために，まず式(5.11)に示すようなスムーズ L1 ノルム損失関数(Smooth L1 Loss, $SL1$) [116] と，式(5.12)に示すような前景フィルタ FG を定義する．

$$SL1(x, y) = \begin{cases} 0.5 \cdot (x - y)^2 & (|x - y| < 1) \\ |x - y| - 0.5 & (otherwise) \end{cases} \quad (5.11)$$

$$FG(l) = \begin{cases} 0 & (l^{bg} = 1) \\ 1 & (otherwise) \end{cases} \quad (5.12)$$

スムーズ L1 ノルム損失関数 $SL1$ は絶対値誤差である L1 ノルムを拡張したものであり，絶対値誤差が 0 である場合でも微分可能にした損失関数となっている．前景フィルタ FG を用いることで前景，つまり対象物品に対する損失のみをフィルタすることができる．以上より式(5.11)，式(5.12)を用いて，基礎矩形の変形に関する損失 L_{loc} は式(5.13)に示すように計算する．

$$L_{loc} = \frac{1}{|A|} \sum_{a \in A} \left(FG(l_{fgbg}(a)) \cdot \sum_{j \in \{x, y, w, h\}} SL1(o_j(a), \hat{o}_j(a)) \right) \quad (5.13)$$

これらの結果から，RPNに関する損失 L_{rpn} は式(5.10)と式(5.13)を用いて式(5.14)に示すように計算する．

$$L_{rpn} = L_{fgbg} + L_{loc} \quad (5.14)$$

次に物体インスタンス検出に関する損失 L_{det} について説明する．RPNが予測した興味領域 $r \in R$ に対して，興味領域矩形 r に関する意味ラベル c_{class} の予測出力 $h_{class}(r)$ が正解データ $l_{class}(r)$ を予測し，興味領域矩形 r から物体領域矩形への変形 $o(r)$ が正解データ $\hat{o}(r)$ を予測するよう学習する．そのため物体インスタンス検出に関する損失 L_{det} は，意味ラベル予測に関する損失 L_{class} と興味領域矩形の変形に関する損失 L_{bbox} によって構成される．このとき意味ラベル集合 C_{class} について，意味ラベル予測に関する損失 L_{class} はソフトマックス交差誤差 SCE (式(5.4))を用いて，式(5.15)に示すように計算する．式(5.15)において， $|R|$ は興味領域矩形集合 R に属する興味領域矩形数を表す．

$$L_{class} = \frac{1}{|R|} \sum_{r \in R} SCE(h_{class}(r), l_{class}(r), C_{class}) \quad (5.15)$$

また興味領域矩形の変形に関する損失 L_{bbox} は、式 (5.11), 式 (5.12) を用いて式 (5.16) に示すように計算する.

$$L_{bbox} = \frac{1}{|R|} \sum_{r \in R} \left(FG(l_{class}(r)) \cdot \sum_{j \in \{x,y,w,h\}} SL1(o_j(r), \hat{o}_j(r)) \right) \quad (5.16)$$

以上の結果から、物体インスタンス検出に関する損失 L_{det} は式 (5.15) と式 (5.16) を用いて式 (5.17) に示すように計算する.

$$L_{det} = L_{class} + L_{bbox} \quad (5.17)$$

次に物体インスタンス可視・遮蔽領域分割に関する損失 L_{inst} について説明する. RPN が予測した興味領域 $r \in R$ に対して、興味領域矩形 r 内のピクセル (x, y) におけるインスタンス意味ラベル c_{inst} に関する予測出力 $h_{inst}(r, x, y)$ が正解データ $l_{inst}(r, x, y)$ を予測するように学習する. このときインスタンス意味ラベル集合 $C_{inst} = \{vis, occ, bg\}$ について、興味領域 r におけるインスタンス意味ラベルに関する損失 $L_{inst}(r)$ はソフトマックス交差誤差 SCE (式 (5.4)) を用いて式 (5.18) に示すように計算する.

$$L_{inst}(r) = \frac{1}{W_r \cdot H_r} \sum_{x=1, y=1}^{W_r, H_r} SCE(h_{inst}(r, x, y), l_{inst}(r, x, y), C_{inst}) \quad (5.18)$$

よって物体インスタンス可視・遮蔽領域分割に関する損失 L_{inst} は式 (5.12) と式 (5.18) を用いて式 (5.19) に示すように計算する.

$$L_{inst} = \frac{\sum_{r \in R} FG(l_{class}(r)) \cdot L_{inst}(r)}{\sum_{r \in R} FG(l_{class}(r))} \quad (5.19)$$

このときに注意すべき点は、5.5.2.2 目では把持点予測器が予測した物体領域分割 b を用いて物体インスタンス可視・遮蔽領域分割を行っているのに対して、本項で説明するモデルの学習の際には興味領域矩形 r に対して損失を計算することである.

次に把持可能領域分割に関する損失 L_{single}, L_{dual} について説明する. 把持戦略 $s_{grasp} \in \{single, dual\}$ と RPN が予測した興味領域 $r \in R$ に対して、興味領域矩形 r 内のピクセル (x, y) における把持可能ラベル c_{grasp} に関する予測出力 $h_{s_{grasp}}(r, x, y)$ が正解データ $l_{s_{grasp}}(r, x, y)$ を予測するように学習する. ここで 5.5.3.8 目で説明したように把持可能領域は把持できない領域と比べて小さいため、意味領域分割の損失関数とは異なり、把持可能ラベル $graspable$ と把持不可能ラベル $ungraspable$ について出現頻度に関する重みをつける必要がある. 5.5.3.8 目同様に Frequency balancing [118] を応用し、把持戦略 $s_{grasp} \in S_{grasp}$ と把持ラベル $c_{grasp} \in C_{grasp}$ についての興味領域矩形 r に関する重み $w_{s_{grasp}}^{c_{grasp}}(r)$ を式 (5.20) のように計算する. 式 (5.20) において、 $N_{foreground}(r)$ は教師データの正解ラベル画像における興味領域矩形 r 内の背景ラベルではないピクセル数、 $N_{c_{grasp}}(r)$ は教師データの正解ラベル画像における興味領域矩形 r 内の把持ラベル c_{grasp} であるピクセル数、 $\alpha_{c_{grasp}}$ は把持ラベル c_{grasp}

についての定数である.

$$w_{s_{grasp}}^{C_{grasp}}(r) = \begin{cases} \frac{N_{foreground}(r)}{\alpha_{C_{grasp}} \cdot N_{C_{grasp}}(r)} & (N_{C_{grasp}}(r) \neq 0) \\ 0 & (N_{C_{grasp}}(r) = 0) \end{cases} \quad (5.20)$$

本研究では把持ラベルの定数としては自動データセットに対する学習では $\alpha_{graspable}$ は 1.0, $\alpha_{ungraspable}$ は 1.0 に設定する.

このとき把持可能ラベル集合 $C_{grasp} = \{graspable, ungraspable\}$ について, 把持戦略 s_{grasp} と興味領域 r における把持可能ラベルに関する損失 $L_{s_{grasp}}(r)$ は重み付きソフトマックス交差誤差 $WSCE$ (式 (5.7)) と式 (5.20) を用いて式 (5.21) に示すように計算する.

$$L_{s_{grasp}}(r) = \frac{1}{W_r \cdot H_r} \sum_{x=1, y=1}^{W_r, H_r} WSCE(h_{s_{grasp}}(r, x, y), l_{s_{grasp}}(r, x, y), w_{s_{grasp}}(r), C_{grasp}) \quad (5.21)$$

よって把持戦略 s_{grasp} における把持可能領域分割に関する損失 $L_{s_{grasp}}$ は式 (5.12) と式 (5.21) を用いて式 (5.22) に示すように計算する.

$$L_{s_{grasp}} = \frac{\sum_{r \in R} FG(l_{class}(r)) \cdot L_{s_{grasp}}(r)}{\sum_{r \in R} FG(l_{class}(r))} \quad (5.22)$$

最後に本研究で提案する Mask-RCNN を元にした把持点予測器の全体損失 L_{total} について説明する. 把持点予測器の全体損失 L_{total} は式 (5.14), 式 (5.17), 式 (5.19) と式 (5.22) を用いて RPN による興味領域予測の損失 L_{rpn} , 物体インスタンス検出の損失 L_{det} , 物体インスタンス可視・遮蔽領域分割の損失 L_{inst} , 把持可能領域分割の損失 L_{single}, L_{dual} の和として式 (5.23) に示すように計算する.

$$L_{total} = L_{rpn} + L_{det} + L_{inst} + L_{single} + L_{dual} \quad (5.23)$$

把持点モデルの学習の際には, 式 (5.23) で計算された損失を図 5.19 の赤点線に示すように誤差逆伝播することでモデルのパラメータを更新する. このとき特徴量抽出器の初期重みとして ImageNet [119] の物体認識タスクのために学習した ResNet101 [117] の重みを利用し, ResNet101 の 1 つめと 2 つめの Residual Block については全てのパラメータを固定し更新しない. また ResNet101 における全ての Batch Normalization 層は Affine Channel 層 [115] に変換しパラメータを固定し更新しない. その他の畳み込み層のパラメータに関しては全て 0 で初期化する. また学習結果を安定化するために 5.5.3.6 目で説明したデータ拡張を行う. 自動生成した約 18000 データペアのデータセットに対してバッチサイズ 4 枚かつ 12 エポックで学習を行う. 学習の最適化には Momentum SGD を $momentum = 0.9$ で用い, 学習率は学習開始から 500 イテレーションまでの間に $1.0e-3$ から $1.0e-2$ に線形に変化させていき, その後 8 エポック目までは $1.0e-2$, 10.66 エポック目までは $1.0e-3$, 12 エポック終了まで $1.0e-4$ と段階的に下げることで安定的に学習を行う.

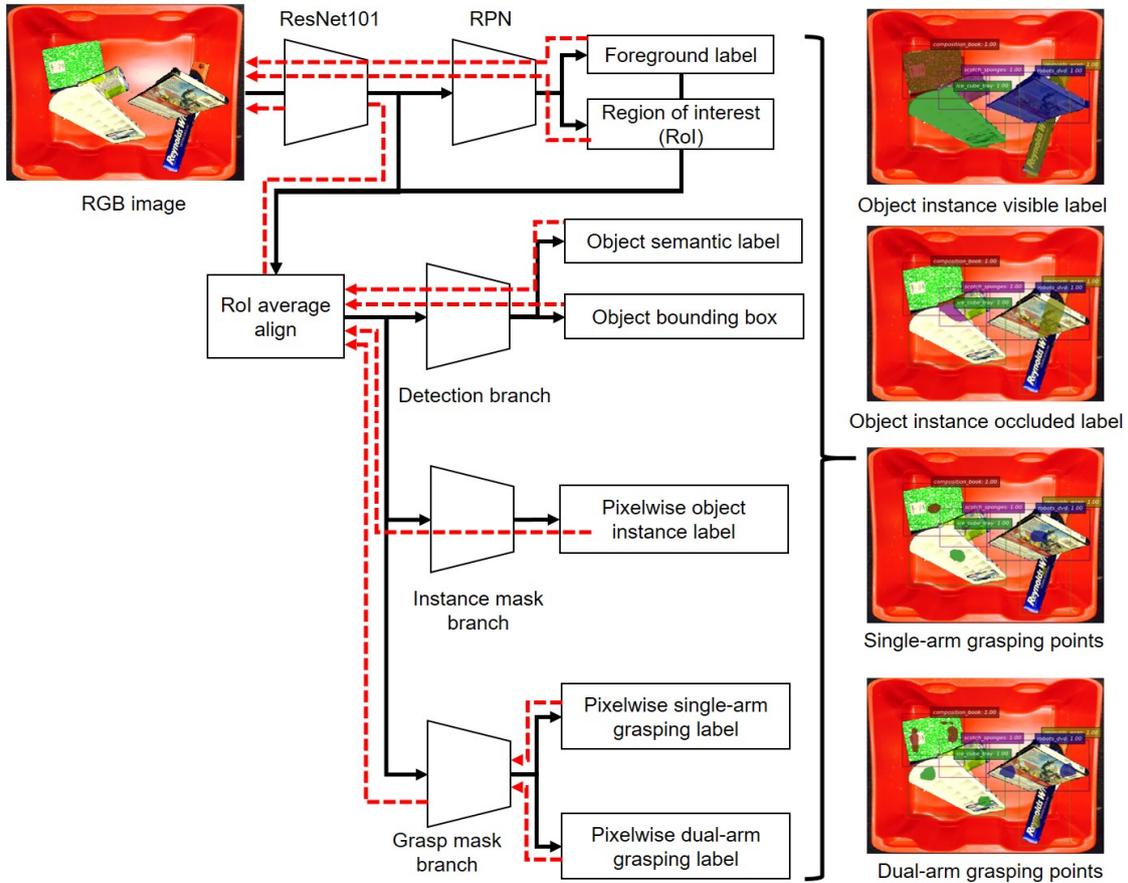


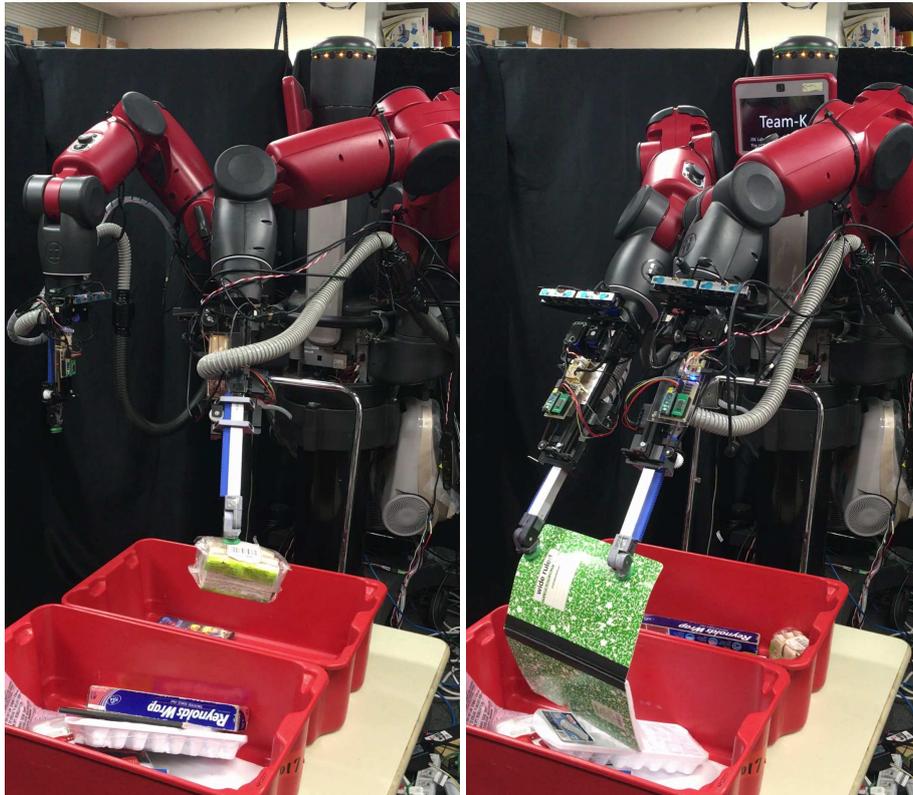
図 5.19: Mask-RCNN を元にした把持点予測器の最初の学習における誤差逆伝播 [35]

5.5.4 CNN 把持点予測器を用いた動作戦略

本項では、CNN 把持点予測器を用いた動作戦略について説明する。本研究では CNN 把持点予測器の出力結果から最も成功確率の高い把持点を選択し実行することで、単腕把持 (図 5.20a) と双腕把持 (図 5.20b) を状況に応じて使い分ける選択的雙腕把持を行う。物体が積み上がり重なりあっている場合にはその重なり合いを動作選択に考慮する必要があるため、物体インスタンスラベル予測結果を用いることでターゲット物体を覆いかぶさっている物体を動かすような動作選択もおこなう。

5.5.4.1 ランダムピッキングのための選択的雙腕把持動作の有限状態機械

本節では選択的雙腕把持動作を行いながらランダムピッキングタスクを実行するための有限状態機械について説明する。ランダムピッキングタスクにおいて、ロボットは置かれた全ての物品について指定された場所に収納する必要があるため、単腕・双腕把持の把持戦略や把持点の中から最も把持しやすい物品を選択し順に把持・収納していく。このとき把持物品の物体領域だ



(a) 単腕把持動作

(b) 双腕把持動作

図 5.20: 選択的雙腕把持動作 [35]

けではなく、最も安定した把持点と把持戦略についても認識し、単腕・双腕把持動作のどちらを行うかの動作選択を行う。

設計する有限状態機械は図 5.21 のようになり、まず 5.6.4 項で再学習を行った把持点予測器を用いて、対象物品の画像から最も把持しやすい物品と把持点、把持戦略を予測し、その予測した把持点と把持戦略について把持動作を実行する。このとき把持動作の失敗を吸引グリップ内の気圧変化や自己干渉チェックなどで認識し、把持動作が失敗した場合には把持動作を中止して把持点認識のステートに戻る。成功した場合は把持した物品を指定された場所に収納し、次の物品を把持するために把持点認識のステートに戻る。以上より把持・収納動作のあとに把持点認識に戻ることで、ロボットが継続的にランダムピッキングタスクを行い続ける有限状態機械を構成する。

5.5.4.2 FCN を元にした把持点予測器を用いたランダムピッキングのための双腕把持動作選択システム

5.5.2.1 目で説明した FCN を元にした把持点予測器を用いて、ロボットは安定的に把持タスクを行うために適切な把持点、把持戦略を実行する。本研究では図 5.22 に示すような、FCN を用いたランダムピッキングのための選択的雙腕把持動作選択システムを提案する。

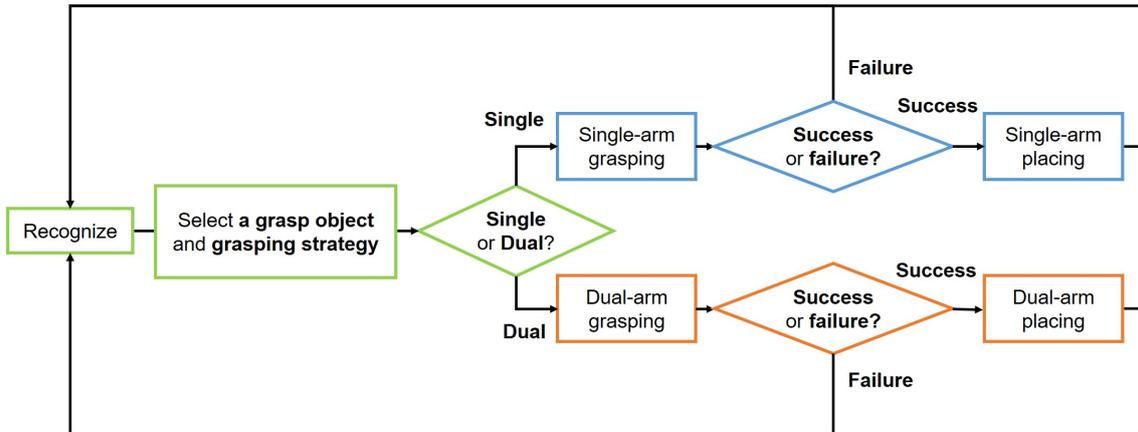


図 5.21: ランダムピッキングのための選択的雙腕把持動作の有限状態機械

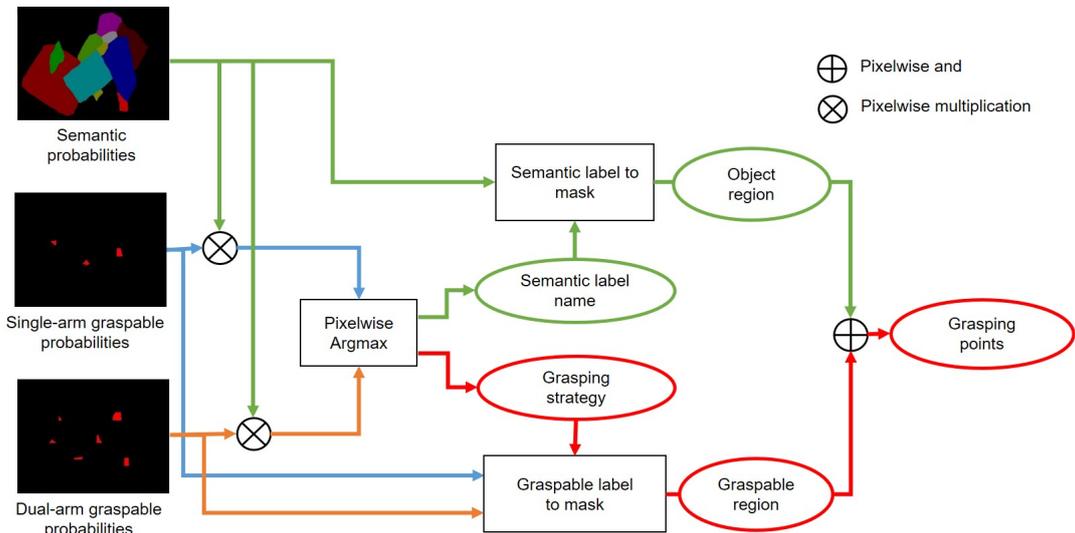


図 5.22: FCN を元にした把持点予測器を用いたランダムピッキングのための雙腕把持動作選択システム

単腕・雙腕動作のそれぞれの動作戦略を評価するために、まず意味ラベルの確率画像と把持可能ラベルの確率画像のピクセル積算 (Pixelwise multiplication) を行うことで積算結果画像を生成し、その積算結果画像のうちで最も高い値を示すピクセルの意味ラベルを把持する物体の意味ラベル、最も高い積算結果の値を示す把持可能ラベルを実行する把持戦略、最も高い積算結果の値を示すピクセルの位置を実行する把持点として *Max* 関数や *Argmax* 関数を用いて計算する。そして、この把持動作選択ののちに、選択した把持物品の意味ラベルとモデル出力の意味ラベルの確率画像から把持物品領域を生成し、同様に選択した把持戦略とモデル出力の把持可能ラベルの確率画像から把持点領域を得る。こうして得られた把持物品領域と把持点領域の重なりあう領域をピクセル論理和 (Pixelwise and) で計算し、最終的に実行する把持点

領域を決定する。つまり本システムは、どの物体を把持するか、単腕・双腕把持のどちらで把持するか、どこを把持するかを、5.6.4項で再学習した把持点予測器の予測結果を用いて選択し、把持点領域、把持戦略を出力するシステムといえる。

5.5.4.3 ターゲットピッキングのための選択的雙腕把持動作の有限状態機械

本節では選択的雙腕把持動作を行いながらターゲットピッキングタスクを実行するための有限状態機械について説明する。ターゲットピッキングタスクにおいて、ロボットは置かれた対象物品の中からターゲット物品のみを指定された場所に収納する必要がある、複雑環境内においてターゲットとなる対象物品を把持・収納する、もしくは障害物となる物品を把持・移動する。このとき把持物品の意味領域だけではなく、最も安定した把持点と把持戦略についても認識し、単腕・双腕把持動作のどちらを行うかの動作選択を行う。さらにターゲットピッキングでは、ターゲット物品が他物品に遮蔽されているかの認識することで、ターゲット物品を把持するか、ターゲット物品を遮蔽している障害物を移動するかの動作選択が重要であるため、把持物品の物体インスタンス可視・遮蔽領域についても認識する必要がある。

設計する有限状態機械は図 5.23 のようになり、まず 5.6.4 項で再学習を行った把持点予測器を用いて、対象物品の画像から把持すべき物品と把持点、把持戦略を選択し、その選択した物品がターゲット物品であるかどうかをチェックする。このとき 5.5.4.1 目で説明したランダムピッキングの有限状態機械と異なり、最も把持しやすい物品ではなく把持すべき物品を選択することに注意する必要がある。これはターゲット物品が遮蔽されている場合には、ターゲット物品を把持・収納するために遮蔽している障害物を順に移動する必要があるためである。そして、把持物品がターゲット物品である場合には、予測した把持点と把持戦略を用いて把持動作ののちに収納動作を実行し、ターゲット物品でない場合には、予測した把持点と把持戦略を用いて把持動作ののちに移動動作を実行する。このとき 5.5.4.1 目で説明したランダムピッキングの有限状態機械と同様に、把持動作の失敗を吸引グリッパ内の気圧変化や自己干渉チェックなどで認識し、把持動作が失敗した場合には把持動作を中止して把持点認識のステートに戻り、成功した場合は把持した物品を指定された場所に収納もしくは移動し、そのあと次の物品を把持するために把持点認識のステートに戻る。以上より把持・収納動作のあとに把持点認識に戻ることによって、ロボットが継続的にターゲットピッキングタスクを行い続ける有限状態機械を構築する。

5.5.4.4 Mask-RCNN を元にした把持点予測器を用いたターゲットピッキングのための雙腕把持動作選択システム

5.5.2.2 目で説明した Mask-RCNN を元にした把持点予測器を用いて、ロボットは安定的に把持タスクを行うために適切な把持点、把持戦略を実行する。本研究では図 5.24 に示すような、Mask-RCNN を用いたターゲットピッキングのための選択的雙腕把持動作選択システムを提案する。

単腕・双腕動作のそれぞれの動作戦略を評価するために、まず各物体インスタンスの可視・遮蔽領域分割を行うことで、各物体インスタンスの可視度 $ratio_{vis}$ を計算する。ある物体イン

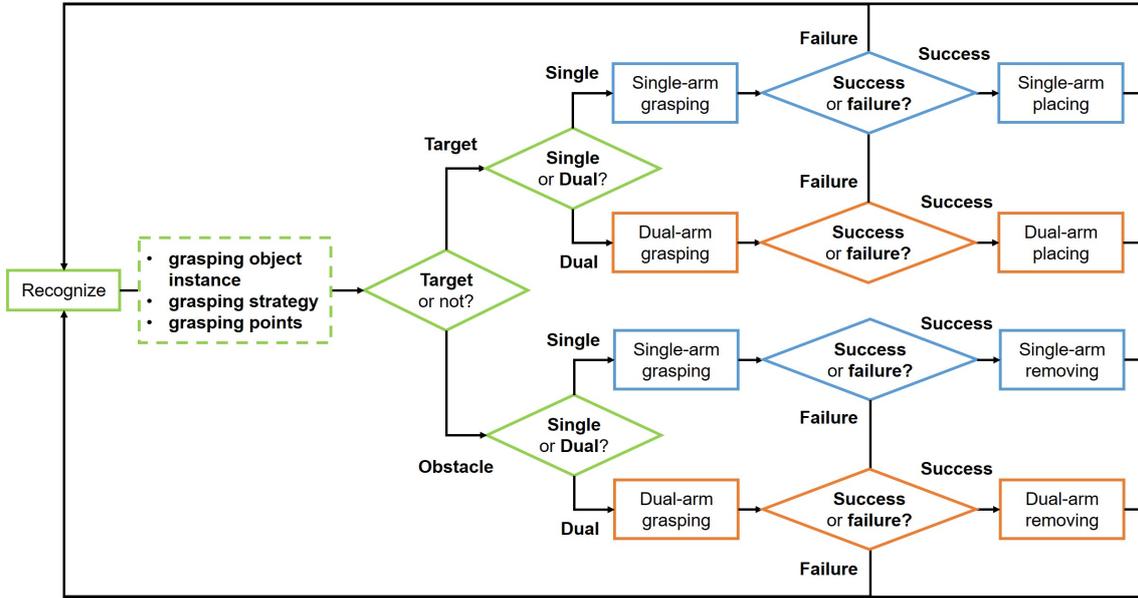


図 5.23: ターゲットピッキングのための選択的 dual 把持動作の有限状態機械 [35]

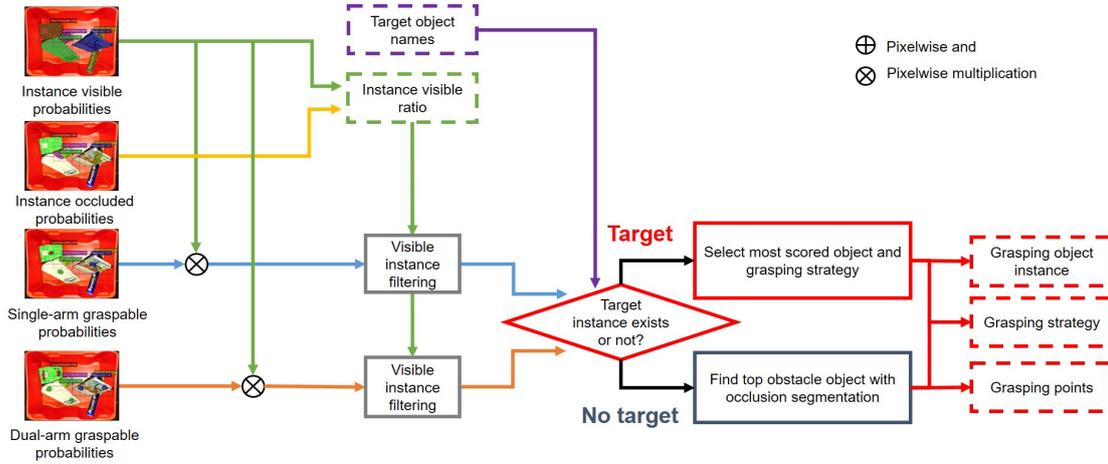


図 5.24: Mask-RCNN を元にした把持点予測器を用いたランダムピッキングのための dual 把持動作選択システム [35]

スタンス i についての可視度 $ratio_{vis}(i)$ は式 (5.24) のように計算する. このとき $N_{vis}(i)$ は物体インスタンス i の可視領域内のピクセル数, $N_{occ}(i)$ は物体インスタンス i の遮蔽領域内のピクセル数を表す.

$$ratio_{vis}(i) = \frac{N_{vis}(i)}{N_{vis}(i) + N_{occ}(i)} \tag{5.24}$$

次に意味ラベルの確率画像と把持可能ラベルの確率画像のピクセル積算 (Pixelwise multiplication) を行うことで積算結果画像を生成し, 式 (5.24) を用いて計算した各物体インスタンスの可視度について, 可視度の閾値 $threshold_{vis}$ を設定することで可視物体インスタンス内の

ピクセルに関する積算結果を抽出(Visible instance filtering)する。この抽出した可視物体インスタンス内の積算結果画像の値について Max 関数, $Argmax$ 関数を用いることで、最も高い積算結果の値を示す把持可能ラベルを実行する把持戦略、最も高い積算結果の値を示すピクセルの位置を実行する把持点として計算する。抽出した可視物体インスタンス内にターゲット物品が存在しない場合には、ターゲット物品を遮蔽している物品について深さ優先探索していき、あるインスタンスの可視度 $ratio_{vis}(i)$ が閾値 $threshold_{vis}$ を超えた物品インスタンスを移動させる障害物インスタンスと設定する。そして選択した障害物インスタンス内の積算結果画像の値について Max 関数, $Argmax$ 関数を用いることで、最も高い積算結果の値を示す把持可能ラベルを実行する把持戦略、最も高い積算結果の値を示すピクセルの位置を実行する把持点として計算する。

把持点、把持戦略選択の後の処理は 5.5.4.2 目で説明した処理と同様に、選択した把持物品の物体インスタンスラベルと、モデル出力の物体インスタンス可視領域の確率画像から把持物品のインスタンス可視領域を生成し、同様に選択した把持戦略とモデル出力である把持可能ラベルの確率画像から把持点領域を得る。こうして得られた把持物品のインスタンス可視領域と把持点領域の重なりあう領域をピクセル論理和(Pixelwise and)で計算し、最終的に実行する把持点領域を決定する。本研究では可視度の閾値 $threshold_{vis}$ を 0.9 と設定して動作選択を行う。

本項の動作選択システムも 5.5.4.2 目で説明したシステム同様に、単腕・双腕把持動作のどちらがより安定的に把持できるかを評価することで実行する把持点、把持戦略を選択する。しかし、5.5.4.2 目で説明したシステムと異なる点は、5.5.4.2 目のシステムでは意味領域を用いて把持物体領域を得ているが、本項のシステムは物体インスタンス単位で把持物体領域を得るため、また環境に置かれた対象物品を物体インスタンス可視・遮蔽領域分割することで、ターゲット物品が他物品に覆いかぶさられていない場合にはそれを把持・収納し、覆いかぶさられている場合にはターゲット物品に重なっている障害物を把持・移動する。ターゲット物品を把持するために障害物を移動することで、少ない障害物を移動して、ターゲット物品を効率的に収納することができるため、効率的にターゲットピッキングタスクを実行することができる。

5.5.4.5 把持点予測器の出力に基づく三次元点群を用いた三次元把持点の決定

本項では 5.5.4.2 目と 5.5.4.4 目で説明した双腕把持動作選択システムで選択された把持点について、選択した把持点領域と三次元点群を用いてロボットが実行する深度も含めた三次元把持点を決定する。5.5.4.2 目と 5.5.4.4 目では RGB 画像を用いた把持点予測であるため、予測した把持点は RGB 画像上の二次元座標であり、ロボットが把持動作を実行するためには三次元座標である三次元把持点に変換する必要がある。本研究では図 5.25 に示すように、選択した把持点領域内にある点群を抽出し、抽出した三次元点群をユークリッド距離でクラスタリングし、生成したクラスタに存在する点群の中心点を三次元把持点とする。ロボットはこの三次元把持点と 5.5.4.2 目と 5.5.4.4 目で選択された把持戦略を元にロボットは把持動作を実行する。

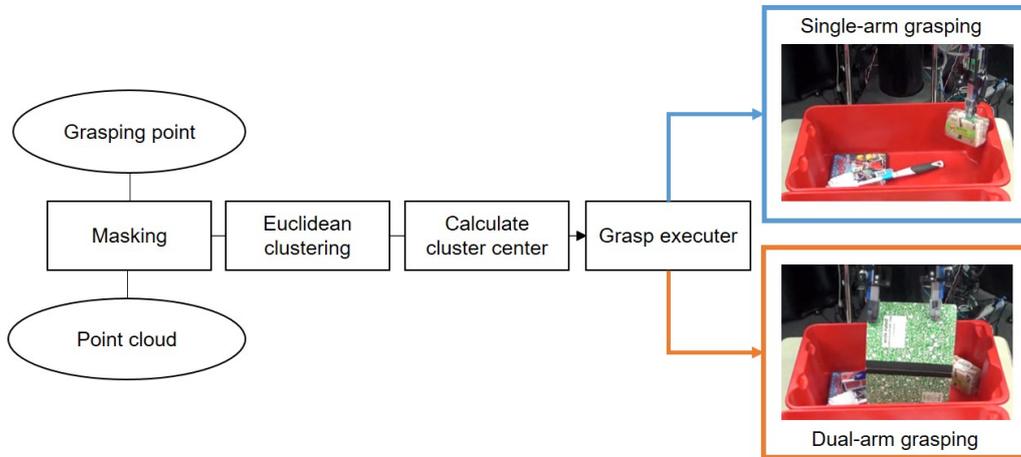


図 5.25: 把持点予測器の出力に基づく 三次元点群を用いた三次元把持点の決定

5.5.4.6 吸引グリッパによる単腕・双腕による把持・収納動作

本目では 5.5.4.5 目で説明した三次元把持点について，ロボットがどのように単腕把持動作と双腕把持動作を実行するかについて説明する．本研究で行う単腕・双腕把持動作は図 5.20 に示す．単腕把持動作と双腕把持動作ともに，得られる三次元把持点について重心鉛直方向の上方向から，吸引グリッパで対象物品を軽く押し付けながら吸引し，吸引グリッパ内の気圧変化から把持成功が認識できるまで 2 度から 3 度の把持再試行を行う．また物品を運搬する際には，把持物品の姿勢が大きく変更しないようにロボット手先軌道を計画する．

5.6 CNN 把持点予測器の実世界適応

本節では双腕把持作業における実世界把持経験収集システムとそれに基づく CNN 把持点予測器の実世界適応について説明する．まず実世界適応の効果について説明した後に，実世界把持経験収集システムと CNN 把持点予測器の再学習による実世界適応について説明する．

5.6.1 実世界把持経験に基づく再学習による実世界適応

本項ではロボットの实世界把持経験による把持点予測器の再学習による実世界適応の効果について説明する．5.5.3 項で説明した把持点予測データセットは，自動把持点アノテーションで生成されているためアノテーションされた把持点の実世界で実現可能であるとは限らない．そのため，実世界にて実現可能である把持点に関するデータを，実世界把持経験としてロボット自身が収集して学習する必要がある．その際には，図 5.26 に示すように実世界把持経験から新たな実現可能な把持点の獲得と，実現不可能な把持点の選定の 2 つの側面で把持点予測器を再学習する．

把持点予測器の予測結果における把持点の獲得と選定は，実世界での把持試行とそれによっ

て収集されたデータセットを用いた把持点予測器の再学習によって実現される。新たな実現可能な把持点の獲得という一面では、把持点予測器の予測結果を用いて把持点を重み付きランダムサンプリングして実世界での把持試行を行い、新たな実現可能な把持点に対する把持成功経験を収集し再学習することで実現する。実現不可能な把持点の選定という一面では、一部の物品は単腕・双腕把持のうちの1つの把持戦略でしか把持実現ができないため、実世界での把持失敗経験を収集することで実現不可能な把持戦略について再学習することで実現する。

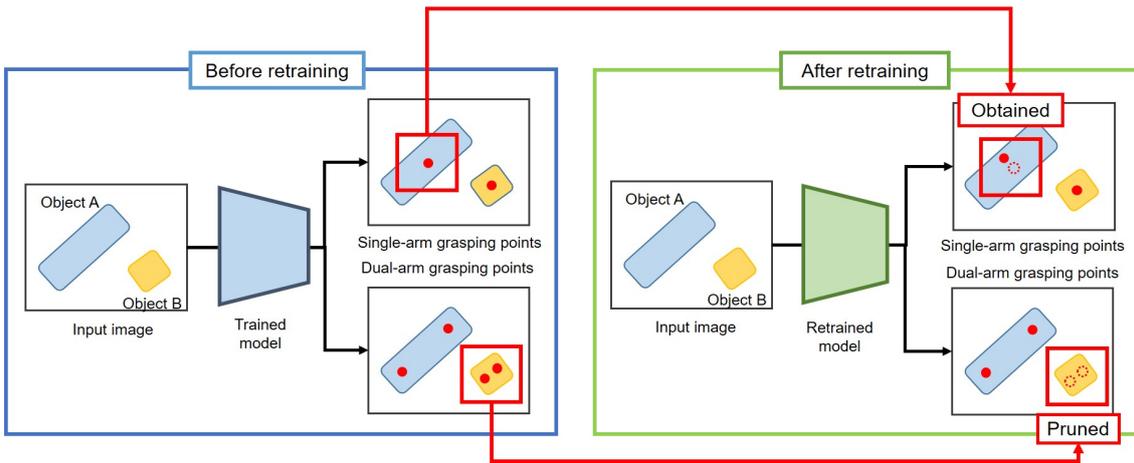


図 5.26: ロボットによる実世界把持経験に基づく把持点の獲得と選定 [35]

5.6.2 双腕把持作業における実世界把持経験収集システム

本研究で構築する双腕把持作業における経験収集システムは図 5.27 に示すように、まず実世界のロボットを用いて対象物に対して把持動作を行い、RGB 画像と把持結果をデータ対として収集していく。そして収集したデータセットについて、5.5.3 項で説明したように背景画像に対して貼り付けていくことでデータセットを生成し、生成されたデータセットを用いて CNN 把持点予測器の再学習を行う。なお把持結果についてはロボット自身がグリッパにある気圧センサから把持成功・失敗を認識し自動でアノテーションする。

5.6.2.1 実世界把持経験収集システムの有限状態機械

本項では実世界把持経験を収集するためのロボット実機を用いたシステムの有限状態機械について説明する。本研究では実世界における把持試行として、把持する対象物品と単腕・双腕把持の中から把持戦略を人が決定し、ロボットの前に各対象物品を1つずつ置き、置かれた物品をロボットが決められた把持戦略で把持できるかを試行する。この把持試行の有限状態機械は図 5.28 に示すように行われ、まず人が与えた対象物品と把持戦略に対して、ロボットが把持点をサンプリングし、その対象物品名と把持戦略、把持点を記録し、そしてロボットは画像を用いた認識に基づいてサンプリングした把持点に対して把持試行を行い、把持成功や失敗などの把持結果について記録する。次項では、画像認識から学習済み把持点予測器を用いてどの

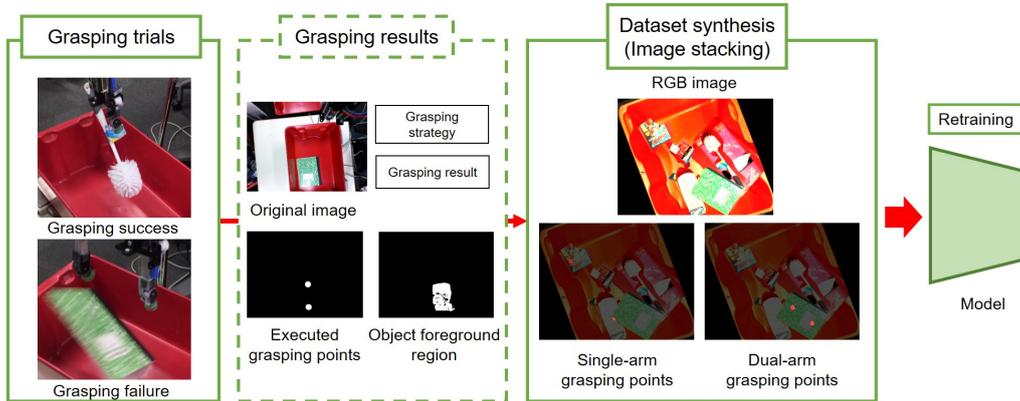


図 5.27: 双腕把持作業における 実世界把持経験収集システムの概観 [35]

ように把持点をサンプリングするかについて説明する。

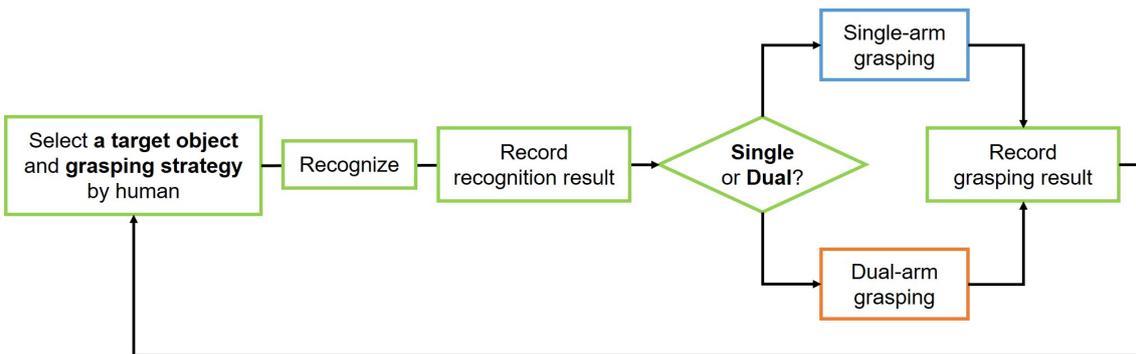


図 5.28: 実世界把持経験収集システムの有限状態機械

5.6.3 CNN 把持点予測器を用いた実世界把持経験収集とデータセット生成

本項では CNN 把持点予測器を用いた実世界把持経験収集と収集された経験に基づく再学習用把持点データセットの生成について説明する。まず実世界把持経験収集について説明した後、再学習用データセットの生成方法について説明する。

5.6.3.1 CNN 把持点予測器を用いた実世界把持経験収集

5.5.2 項で学習した把持点予測器の実世界把持経験への適応のために、で学習したモデルの出力を用いて実世界で把持経験を収集し、収集した経験を元にモデルを再学習する。把持経験の収集工程の概要については図 5.29 に示すように、学習済み把持点予測器の出力を用いた重み付きランダムサンプリングにより把持点を決定し実行することで収集する。

把持経験収集の工程は図 5.29 に示すように行い、ロボットが 1 つ 1 つの物品に対してそれぞれ単腕・双腕把持を実世界で実行し把持経験を収集する。本研究では、把持点予測器の出力

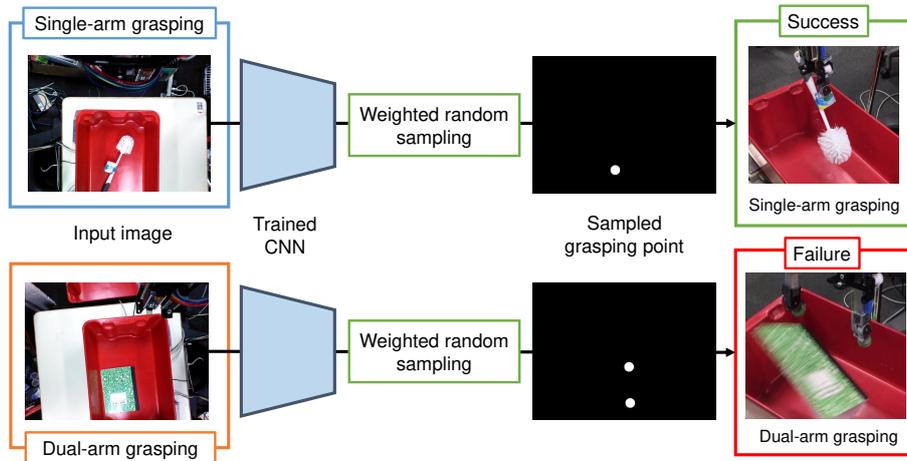


図 5.29: 把持点予測器を用いた実世界把持経験収集 [35]

である対象物品の意味領域やインスタンス可視領域とピクセル単位の把持確率を用いて重み付きランダムサンプリングを行う。このとき把持確率の高いピクセルが把持点としてサンプリングされやすくし、逆に把持確率の低い把持点はサンプリングされにくくなる。しかし重み付きランダムサンプリングは重みであるため、把持確率の低い把持点は低い頻度であるものの把持点としてサンプリングされる。

単腕把持点のサンプリングは図 5.30 と図 5.31 に示すように行う。5.5.2.1 目で説明した FCN を元にしたモデルを用いる場合は、学習した把持点予測器が対象物品の意味領域(図 5.30)、5.5.2.2 目で説明した Mask-RCNN を元にしたモデルを用いる場合は、意味領域かつインスタンス可視領域(図 5.31)と予測するピクセルに対して、把持点予測器出力の把持確率画像を重みとしてピクセルを重み付けランダムサンプリングし、選ばれたピクセルの位置をロボット実機が実行する単腕把持点として決定する。

双腕把持点のサンプリングは図 5.32 と図 5.33 に示すように行う。FCN を元にしたモデルを用いる場合は学習した把持点予測器が対象物品の意味領域(図 5.32)、Mask-RCNN を元にしたモデルを用いる場合は意味領域かつインスタンス可視領域(図 5.33)と予測するピクセルに対して、 $k = 2$ の k-means クラスタリングを行うことで左腕・右腕それぞれの把持領域のクラスタを得る。そして左腕・右腕把持の各クラスタ内のピクセルに対して、把持点予測器出力の把持確率画像を重みとしてピクセルを重み付けランダムサンプリングし、それぞれ選ばれたピクセルの位置のペアをロボット実機が実行する双腕把持点として決定する。

ロボットは把持試行の際には、吸引グリッパではグリッパ内部の空気圧変化から物体が把持できているかを認識し、物体を持ち上げられた場合は把持成功、安定的に持ちあげられなかった場合は把持失敗として把持結果を自動で記録する。吸引グリッパ内部の空気圧は物体を把持した時の値は把持できていない時の値よりも低くなることを利用して、把持を行う前に物体を持たずに吸引している場合の吸引グリッパ内部の空気圧を測定し、その測定した値と現在のグリッパ内部の空気圧の値の差分に対して閾値を設定することで把持結果を認識する。把持失敗

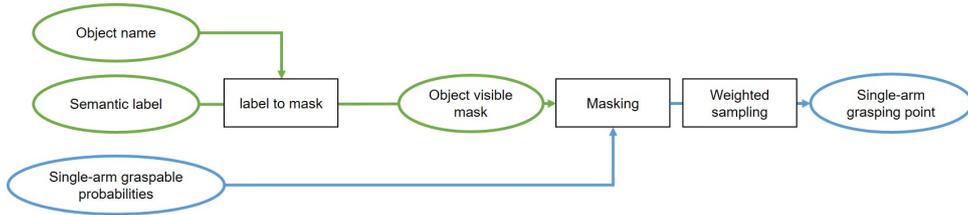


図 5.30: FCN を元にした把持点予測器出力を用いた単腕把持点サンプリング

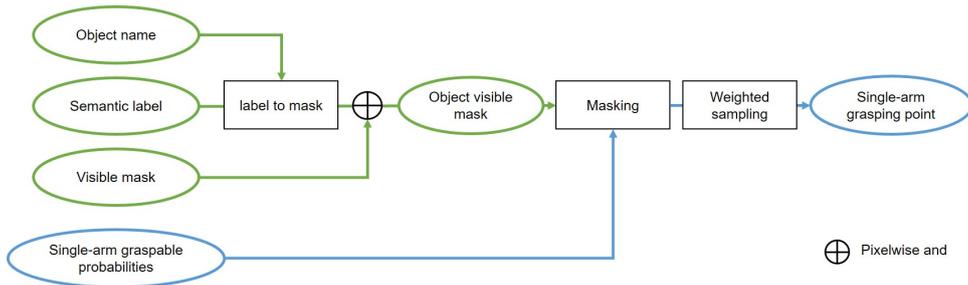


図 5.31: Mask-RCNN を元にした把持点予測器出力を用いた単腕把持点サンプリング

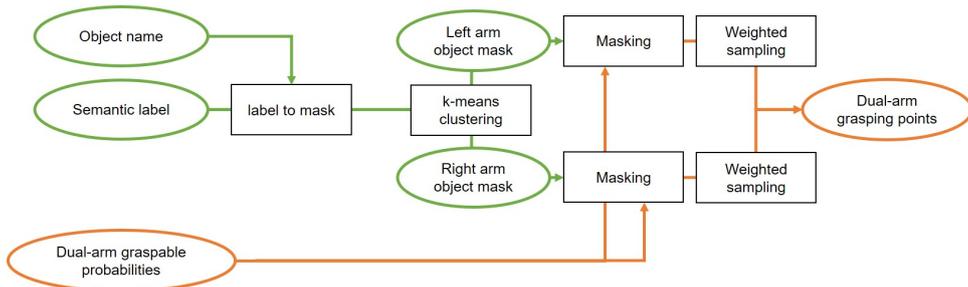


図 5.32: FCN を元にした把持点予測器出力を用いた双腕把持点サンプリング

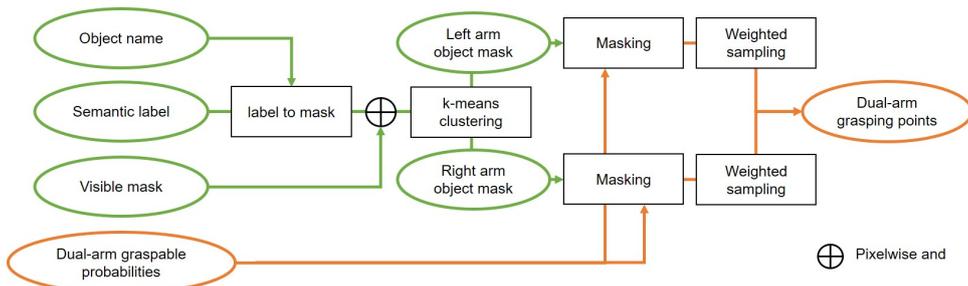


図 5.33: Mask-RCNN を元にした把持点予測器出力を用いた双腕把持点サンプリング

には、把持が成功しない場合と双腕把持のときには双腕が自己干渉によって把持動作を行えない場合がある。把持失敗はサンプリングする把持点が適切でない場合に単腕・双腕把持の両動作でも発生する一方、自己干渉による失敗は双腕把持点が互いに近い場合に発生しやすく双腕把持動作の場合にのみ発生する。また本研究では1つの対象物品に対して単腕・双腕把持の各把持戦略についてそれぞれ最低でも5回の把持試行を行う。

5.6.3.2 実世界把持経験を用いた再学習用把持点データセットの自動生成

実世界での把持試行と収集した把持経験データから把持点予測器の再学習することで、学習済み把持点予測器の実世界把持経験への適応を行う。本項では収集した把持経験データから再学習用把持点データセットを自動生成する手法について説明する。この自動生成の工程の概要については図 5.34 に示す。

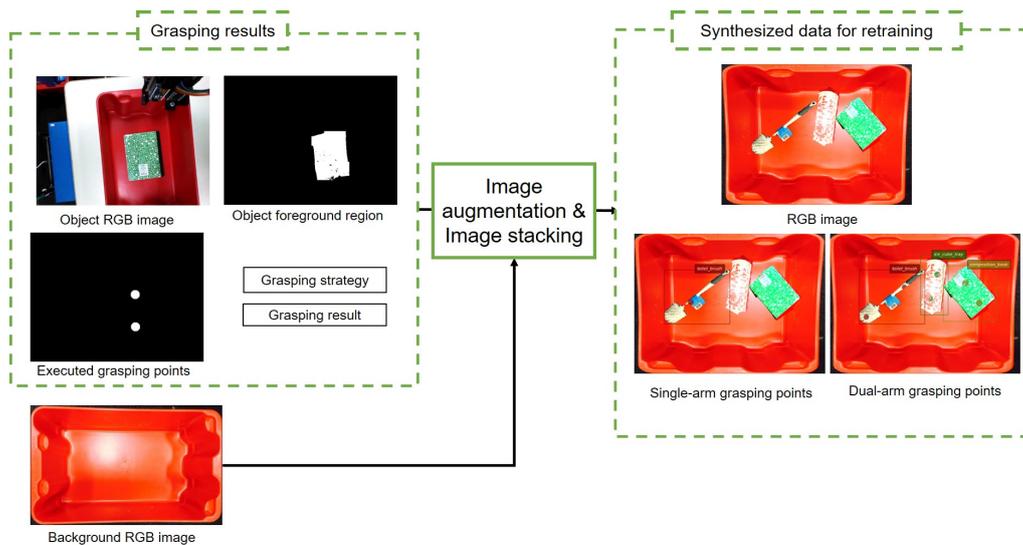


図 5.34: 画像貼りあわせを用いた再学習用把持点データセットの自動生成 [35]

把持試行の際に、把持点予測器が予測した対象物品領域から対象物品のマスク画像を作ることができ、また把持点とそれに対応する把持試行結果から把持点をアノテーションすることができるため、図 5.34 にしめす工程で再学習用把持点データセットの自動生成を行うことができる。生成したデータセットの一例については図 5.35 に示す。再学習用把持点データセットの生成の具体的な手法については以下に示す。まず把持試行の際に把持点予測器が予測した対象物品の意味領域から対象物品のマスク画像を生成する。そして把持試行を行った把持点に対して、把持成功した場合には把持可能ラベルを、把持失敗した場合には把持不可能ラベルをアノテーションする。また把持点以外の領域については把持不可能ラベルをアノテーションするため、把持成功した時に収集したデータは把持可能であり、把持失敗した時に収集したデータは把持不可能であるとアノテーションされる。そして対象物品の RGB 画像、マスク画像、アノテーションされた把持ラベル画像を用いて 5.5.3 項で説明した手法で単腕・双腕把持点画像のみを自動生成する。以上の工程から単腕・双腕把持の把持点についてアノテーションした結果がそれぞれ図 5.35 の上段と下段である。

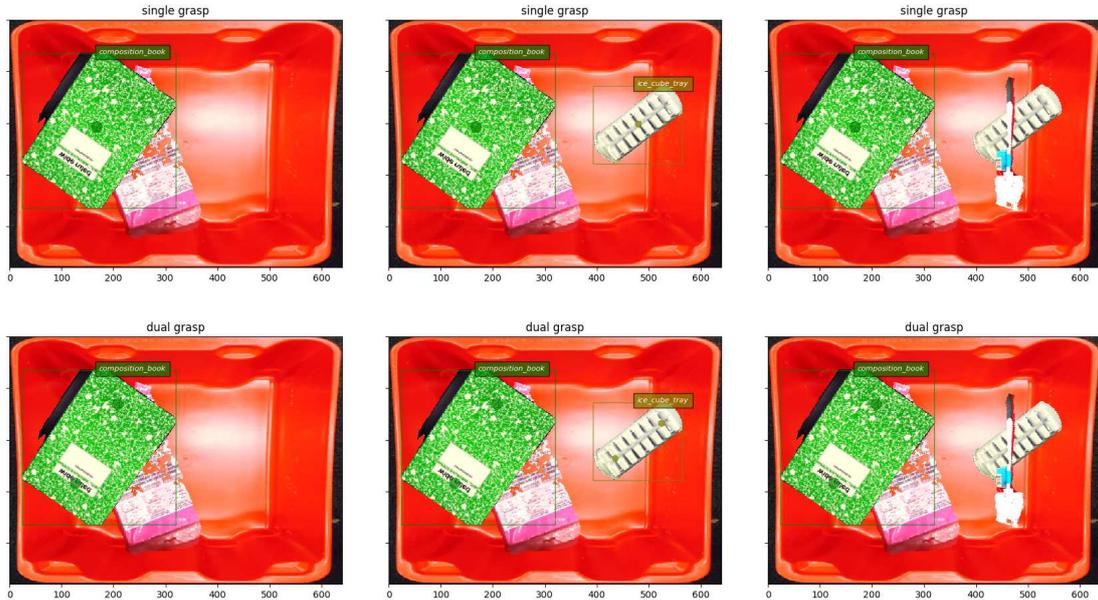


図 5.35: 生成した単腕・双腕把持の再学習用把持点データセットの可視化

5.6.4 実世界把持経験に基づく CNN 把持点予測器の再学習による実世界適応

本項では CNN 把持点予測器の再学習による実世界適応について FCN を用いた把持点予測器と Mask-RCNN を用いた把持点予測器の 2 つについてそれぞれ順に説明する。

5.6.4.1 実世界把持経験に基づく FCN を用いた把持点予測器の再学習

まず FCN を元にした把持点予測器の再学習について説明する。FCN を元にした把持点予測器の再学習において図 5.36 の赤点線に示すように、把持点に関する損失のみを誤差逆伝播し把持点予測に関するパラメータのみを更新する。自動生成データセットを用いた最初の学習では 5.5.3.8 目で説明した通り、意味領域分割に関する損失 L_{seg} と単腕・双腕把持の把持点に関する損失 L_{single} , L_{dual} の合計を全体の損失 L_{total} として式 (5.9) に示すように計算し、この合計損失 L_{total} をネットワーク全体に逆伝播することで把持点予測器のパラメータが更新した。しかし本項の再学習の際には、把持点予測器は意味ラベルに関しては学習済みであるとみなし、式 (5.8) で定義した単腕・双腕把持の把持点に関する損失 L_{single} , L_{dual} の合計のみをネットワーク全体の再学習の損失 L_{total}^{adapt} として式 (5.25) のように計算する。

$$L_{total}^{adapt} = L_{single} + L_{dual} \quad (5.25)$$

この合計損失 L_{total}^{adapt} を逆伝播するが、把持点予測に関する畳み込み層のパラメータのみを更新することで、特徴量抽出器と意味ラベル予測ブランチの予測性能を変えることなく、把持点予

測ブランチのみの実世界把持経験への適応を行う。

また再学習の際に用いる把持点予測器の初期重みとして、再学習の直前に把持試行による実世界把持経験収集のために学習したネットワークの重みを用いる。再学習の初期学習率は $1.0e-6$ 、バッチサイズ 1 枚かつ 12000 イテレーションとし、他のハイパーパラメータや最適化手法に関して 5.5.3.8 目と同じ設定で学習を行い、5.5.3.8 目と同様に学習結果を安定化するために 5.5.3.6 目で説明したデータ拡張を行う。

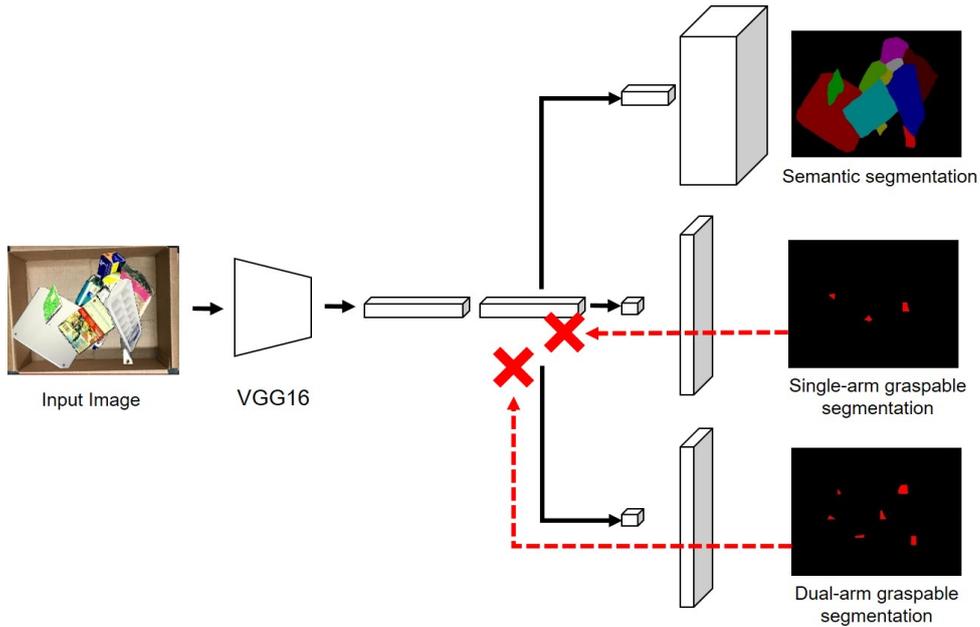


図 5.36: FCN を元にした把持点予測器の再学習における誤差逆伝播

5.6.4.2 実世界把持経験に基づく Mask-RCNN を用いた把持点予測器の再学習

次に Mask-RCNN を元にした把持点予測器の再学習について説明する。Mask-RCNN を元にした把持点予測器の再学習において図 5.37 の赤点線に示すように、把持点に関する損失のみを誤差逆伝播し把持点予測に関するパラメータのみを更新する。自動生成データセットを用いた最初の学習では 5.5.3.9 目で説明した通り、RNP に関する損失 L_{rpn} と物体インスタンス検出に関する損失 L_{det} 、物体インスタンス可視・遮蔽領域分割に関する損失 L_{inst} 、単腕・双腕把持の把持点に関する損失 L_{single} 、 L_{dual} の合計を全体の損失 L_{total} として式 (5.23) に示すように計算し、この合計損失 L_{total} をネットワーク全体に逆伝播することで把持点予測器のパラメータが更新した。しかし本項の再学習の際には、把持点予測器は RPN、物体インスタンス検出および物体インスタンス可視・遮蔽領域分割に関しては学習済みであるとみなし、式 (5.22) で定義した単腕・双腕把持の把持点に関する損失 L_{single} 、 L_{dual} の合計をネットワーク全体の再学習の損失 L_{total}^{adapt} として式 (5.26) のように計算する。

$$L_{total}^{adapt} = L_{single} + L_{dual} \quad (5.26)$$

この合計損失 L_{total}^{adapt} を逆伝播するが、把持点予測に関する畳込み層のパラメータのみを更新することで、特徴量抽出器や RPN、物体インスタンス検出や物体インスタンスの可視・遮蔽領域分割の予測性能を変えずに、把持点予測ブランチのみの実世界把持経験への適応を行う。また再学習の際に用いる把持点予測器の初期重みとして、再学習の直前に把持試行による実世界把持経験収集のために学習したネットワークの重みを用いる。

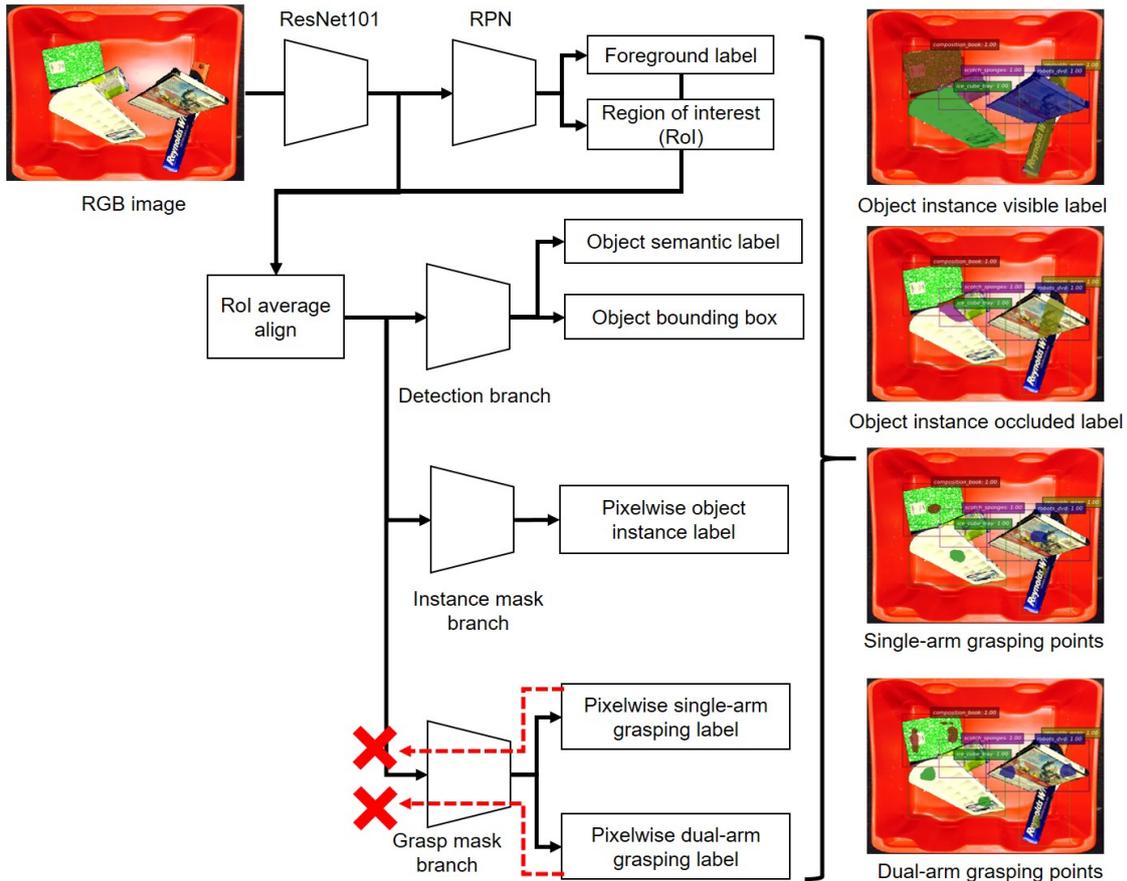


図 5.37: Mask-RCNN を元にした把持点予測器の再学習における誤差逆伝播 [35]

Mask-RCNN を元にした把持点予測器に関しては、実世界把持経験収集のための再学習と把持タスク実行のための再学習について異なる設定で学習を行った。まず実世界把持経験収集のための再学習の場合には、自動生成した約 18000 データペアのデータセットに対してバッチサイズ 4 枚かつ 3 エポックで学習を行う。また学習率は学習開始から 500 イテレーションまでの間に $1.0e-4$ から $1.0e-3$ に線形に変化させていき、その後 2 エポック目までは $1.0e-3$ 、2.66 エポック目までは $1.0e-4$ 、3 エポック終了まで $1.0e-5$ と段階的に下げることで安定的に学習を行う。一方、把持タスク実行のための再学習の場合には、自動生成した約 18000 データペアのデータセットに対してバッチサイズ 4 枚かつ 1 エポックで学習を行う。また学習率は学習開始から 500 イテレーションまでの間に $1.0e-4$ から $1.0e-3$ に線形に変化させていき、その後 1 エポック終了まで $1.0e-3$ のまま学習を行った。さらに把持タスク実行のための再学習

の場合にのみ，式 (5.20) で説明した把持ラベルに関する定数 $\alpha_{graspable}$ を 0.05, $\alpha_{ungraspable}$ を 1.0 に設定して学習を行う．他のハイパーパラメータや最適化手法に関して 5.5.3.9 目と両場合とも同じ設定で学習を行い，5.5.3.9 目と同様に学習結果を安定化するために 5.5.3.6 目で説明したデータ拡張を行う．

5.6.4.3 実世界把持経験収集のための再学習と把持タスク実行のための再学習の違い

実世界把持経験を用いた再学習手法には図 5.38 に示すように 2 種類あり，1 つは実世界把持経験収集のための再学習であり，もう 1 つは把持タスク実行のための再学習である．実世界把持経験収集のための再学習では，対象物品のインスタンス画像と把持試行の際に収集したデータの両方を使い，5.5.3 項で説明した手法で自動生成した把持データセットを用いて再学習する．その際に対象物品のインスタンス画像と自動把持点アノテーションによって作ったデータを 25%，把持試行の際に収集したデータを 75% の割合で生成した把持データセットを用いるため，把持点予測器は実世界把持経験を主にして学習するが，自動でアノテーションされた把持点を忘れないようにもしている．一方，把持タスク実行のための再学習では，把持試行の際に収集したデータのみを用いて把持データセットを自動生成し，生成されたデータセットを用いて把持点予測器を実世界把持経験に適応させる．この再学習は把持タスク実行のためであるので，5.5.3.3 目でアノテーションされた把持点を忘却してもよく，把持試行の際に収集したデータのみを用いて再学習することで，把持点予測器を実世界把持経験に適応させる．以上の 2 つの再学習手法を行うことで，実世界把持経験収集のための再学習では様々な把持試行をサンプリングするように再学習され，一方で最終的な把持タスク実行のための再学習では，把持点予測器は自動でアノテーションされた把持点を忘却し，実世界把持経験のみに適応される．

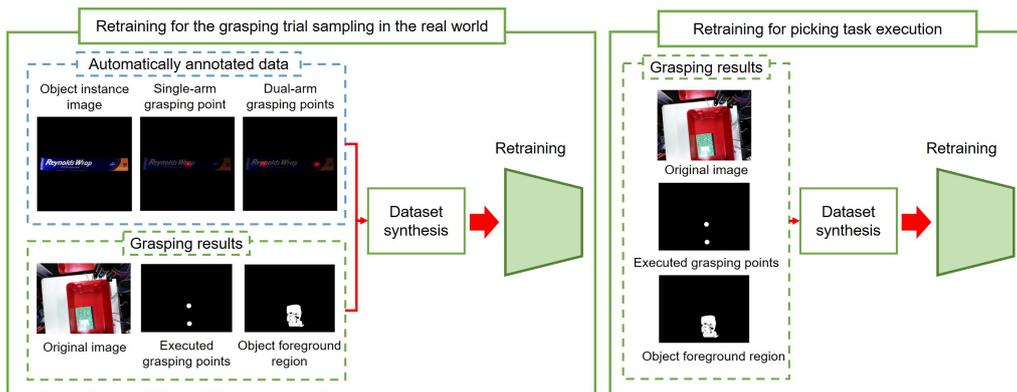


図 5.38: 実世界把持経験収集における把持試行のための再学習と把持タスク実行のための再学習の違い

5.7 評価実験

本節では実世界適応による動作戦略の自律学習の評価実験について説明する．まず CNN 作業失敗予測器を用いた学習型動作戦略について棚への双腕収納作業実験を行うことで評価した．そして CNN 把持点予測器を用いた学習型動作戦略とその実世界適応を用いた自律学習について倉庫環境での双腕ピッキング作業実験を行うことで評価した．本実験では双腕ロボット Baxter-JSK-APC を用いて実世界適応による動作戦略の自律学習の有効性の評価を行った．

5.7.1 作業失敗予測に基づく棚への双腕収納作業実験

本項では CNN 作業失敗予測器を用いた学習型動作戦略の評価として行った棚への双腕収納作業について説明する．まず実験設定について説明した後実験結果について評価を行う．

5.7.1.1 棚への収納作業の作業失敗データセット

本実験で用いる作業失敗データセットは図 5.39 に示す 10 個の物体に対してデータの収集を行った．この 10 個の物体は Amazon Picking Challenge 2015-2017 [121, 122] で対象物として設定されたものから選んでおり，T シャツや本，くつしたなど柔軟物体やバランスの悪い物体を含んでいる．これらの 10 個の物体に対して 200 回の収納作業を行い，1327 対のデータをデータセットとして収集した．データの収集の際にロボットは単腕・双腕収納動作の両方を行った，収集したデータセットの詳細については表 5.3 に示す．



図 5.39: 棚への双腕収納作業の対象 10 物品 [105]

5.7.1.2 棚への収納作業における作業失敗予測学習

前目で収集した作業失敗データセットを用いて CNN 作業失敗予測器の学習を行った．データセットのデータ数が多くないため，データセットの 10% をテストデータセット，その他 90% を学習データセットとしてクロスバリデーションを行った．CNN 学習の際にはバッチ数 N_{batch} は 10 として設定して，イテレーション数 5000 回で学習を行った．学習した CNN の評価として 0.5 を閾値として，CNN の出力結果がこの閾値より高い場合にはその作業失敗が

表 5.3: 収納作業における作業失敗データセット [105]

Stowing Motion	Success Data	Drop Data	Protrusion Data	Total
Single-arm	362	243	48	653
Dual-arm	398	130	146	674
Total	760	373	194	1327

起こると予測したものとしてデータセットとの正解ラベルと比較し、1枚の画像に対して全ての作業失敗予測が正解している場合のみを作業失敗予測の成功としてCNNの作業失敗予測性能の評価を行った。比較ベースラインとして、データセットの作業失敗の平均値を作業失敗の発生確率として返す手法を準備した。評価の結果、CNN作業失敗予測器は68.9%の精度で作業失敗予測を行っており、これはベースラインの57.9%よりも高い結果となった。CNN作業失敗予測器の予測結果の一部を図5.40に示す。以上の結果から、このCNN作業失敗予測器は作業失敗の発生確率の予測を正しく学習したといえる。



図 5.40: CNN 作業失敗予測器の予測結果 (緑: 予測成功例, 赤: 予測失敗例) [105]

5.7.1.3 作業失敗予測に基づく棚への双腕収納作業

学習型動作戦略の評価実験として、双腕ロボット Baxter-JSK-APC を用いて棚への双腕収納作業を行った。作業に用いる対象物は作業失敗データセットの作成に用いた 10 個の物品を用意し、それぞれの物品に対して 10 回ずつ棚への双腕収納作業を行った。このとき Algorithm 5 に用いる閾値 $threshold$ は 0.5 とし、CNN 作業失敗予測器は全目で学習した CNN を用いた。比較実験として、全て単腕収納動作を行う以外は全て同じ条件で同様の実験を行った。

実験結果は表 5.4 と表 5.5 に示す。表 5.4 に示すように、ベースライン手法の成功率 49.0% に対して本章で提案した学習型動作戦略は 58.0% と高い成功率を示した。また表 5.5 に示すように、単腕収納動作を行った場合には 56.8% と高い成功率を示しており、これは表 5.4 に示す全て単腕収納動作を行うベースライン手法の成功率より高いものとなった。以上の結果が

ら，学習型動作戦略によって作業の成功率と安定性を向上することができたといえる。

表 5.4: 選択的両腕収納動作を用いた収納作業結果 [105]

Stowing System	Success Times (Success Rate)	Drop Times (Drop Rate)	Protrusion Times (Protrusion Rate)	Ave. Score
Non-selective Single-arm	49 (49.0%)	41 (41.0%)	10 (10.0%)	5.40
Our Method	58 (58.0%)	20 (20.0%)	22 (22.0%)	6.90

表 5.5: 選択的両腕収納動作を用いた収納作業において選択された動作の結果 [105]

Executed Stowing Motion	Total Times	Success Times (Success Rate)	Drop Times (Drop Rate)	Protrusion Times (Protrusion Rate)	Ave. Score
Single-arm	81	46 (56.8%)	15 (18.5%)	20 (24.7%)	6.91
Dual-arm	19	12 (63.2%)	5 (26.3%)	2 (10.5%)	6.84

表 5.6: 選択的両腕収納動作を用いた収納作業結果の詳細

	Single-arm Stowing			Dual-arm stowing		
	Success	Drop	Protrude	Success	Drop	Protrude
Book	3	5	1	1		
Curtain	6	2	2			
DentalTreats	9		1			
ExpoEraser	10					
IceTray	8		2			
KidsBook	4		5	1		
PaperTowels	2	3	1	2	2	
Socks		3	1	5	1	
Stems	4	1	4	1		
T-shirt		1	3	3	1	2
Total	46	15	20	12	5	2

5.7.2 把持点予測に基づく倉庫環境での両腕ピッキング作業実験

本項では CNN 把持点予測器を用いた学習型動作戦略とその実世界適応を用いた自律学習の評価として行った倉庫環境での両腕ピッキング作業実験について説明する。まず実験設定につ

いて説明した後に実験結果について評価を行う。

5.7.2.1 実験設定

本章の実験で用いる対象9物品について説明する。対象9物品は Amazon Robotics Challenge [97] においてピッキングタスク対象物品として選ばれた40物品の中から選んだ。この9物品は図5.41に示す品目であり、左上からそれぞれバインダー(White Binder)、ノートブック(Green notebook)、袋入りくつした(White socks)、製氷皿(Ice cube tray)、アルミニウムホイル(Aluminum foil)、DVD、スポンジ(Brown sponges)、テーブルクロス(Pink table cloth)、トイレブラシ(Toilet brush)である。この選んだ9物品は図5.41に示すように、双腕把持しづらいが単腕把持では把持できる、単腕把持で把持しづらいが双腕把持では把持できる、どちらの単腕・双腕把持でも把持できるの3つのカテゴリに分類でき、各カテゴリには表5.7に示すようにそれぞれ2から4物品が属する。

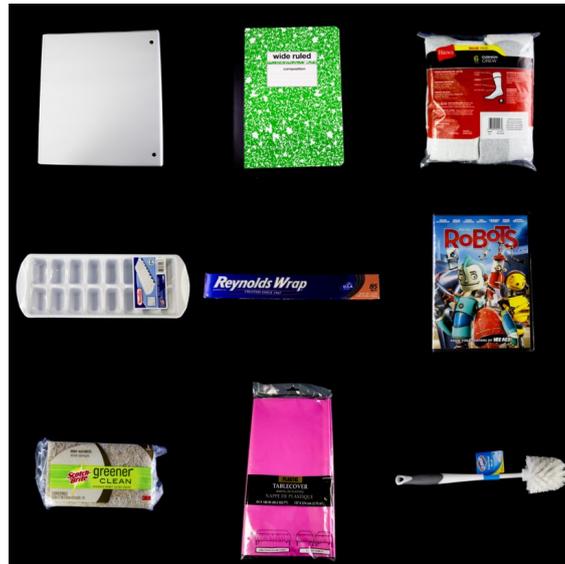


図 5.41: 自律双腕ピッキング作業実験の実験対象9物品 [123]

表 5.7: 実験対象物の把持可能性に基づく分類 [35]

Category	Object names
Single-arm grasping preferred	DVD, Toilet brush, Brown sponges
Dual-arm grasping preferred	White binder, Green notebook
Graspable by both strategy	Ice cube tray, White socks, Aluminum foil, Pink table cloth

5.7.2.2 FCN を元にした把持点予測器を用いた再学習のための把持経験の収集

本実験では再学習による実世界把持経験への適応のために、自動データセット生成の後に2回の把持試行と再学習を行った。2回の把持試行は表 5.8 に示すような結果となっており、1回目の把持試行では 94 ペア、2回目では 121 ペアの把持経験のデータを収集した。また2回目の把持試行の前には 5.6.3.2 目に示すように、自動生成したデータセットと1回目の把持試行データを元に生成したデータセットの両方を用いて FCN を元にした把持点予測器を再学習し、その再学習された把持点予測器を用いて2回目の把持試行を行った。そして全ての把持試行の後には 5.6.3.2 目に示すように、実世界で収集した把持経験データを元に生成したデータセットのみを用いて FCN を元にしたモデルの再学習を行った。

表 5.8: FCN を元にした把持点予測器を用いた収集した把持経験 [36]

	1st sampling	2nd sampling	Total
Single-arm	44	52	96
Dual-arm	50	69	119
Total	94	121	215

5.7.2.3 再学習のための Mask-RCNN を元にした把持点予測器を用いた把持経験の収集

本実験では再学習による実世界把持経験への適応のために、自動データセット生成の後に2回の把持試行と再学習を行った。2回の把持試行は表 5.9 に示すような結果となっており、1回目、2回目の把持試行ともに 90 ペアの把持経験のデータを収集した。また2回目の把持試行の前には 5.6.3.2 目に示すように、自動生成したデータセットと1回目の把持試行データを元に生成したデータセットの両方を用いて把持点予測器を再学習し、その再学習された把持点予測器を用いて2回目の把持試行を行った。そして全ての把持試行の後には 5.6.3.2 目に示すように、実世界で収集した把持経験データを元に生成したデータセットのみを用いて Mask-RCNN を元にしたモデルの再学習を行った。

表 5.9: Mask-RCNN を元にした把持点予測器を用いた収集した把持経験 [35]

	1st grasping trials	2nd grasping trials	Total
Single-arm grasping	45	45	90
Dual-arm grasping	45	45	90
Total	90	90	180

5.7.2.4 意味領域分割の評価指標

FCNを元にした把持点予測器による意味領域分割について先行研究 [113, 124] に定義されている以下の4つの評価指標を導入して評価を行う。

- ピクセル正解度 (Pixel accuracy): PA
- クラス別平均ピクセル正解率 (Mean pixel accuracy): mPA
- クラス別平均重なり度 (Mean Intersection of Union): $mIoU$
- 頻度重み付きクラス別平均重なり度 (Frequency Weighted Intersection of Union): $fwIoU$

それぞれの評価指標は順に式 (5.27), 式 (5.28), 式 (5.29), 式 (5.30) に示すように計算される。このとき k は背景を除いた意味ラベルのクラス数であり $k+1$ は背景を含む意味ラベルのクラス数となる。また $k=0$ は背景の意味ラベルを示す意味ラベルの指数であり, 背景以外の意味ラベルには指数として $k > 0$ の異なる整数がそれぞれ割り当てられている。 N_{ij} は正解値では意味ラベルのクラス i に属して、かつクラス j とモデルに予測されたピクセルの総数である。

$$PA = \frac{\sum_{i=0}^k N_{ii}}{\sum_{i=0}^k \sum_{j=0}^k N_{ij}} \quad (5.27)$$

$$mPA = \frac{1}{k+1} \sum_{i=0}^k \frac{N_{ii}}{\sum_{j=0}^k N_{ij}} \quad (5.28)$$

$$mIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{N_{ii}}{\sum_{j=0}^k N_{ij} + \sum_{j=0}^k N_{ji} - N_{ii}} \quad (5.29)$$

$$fwIoU = \frac{1}{\sum_{i=0}^k \sum_{j=0}^k N_{ij}} \sum_{i=0}^k \frac{\sum_{j=0}^k N_{ij} \cdot N_{ii}}{\sum_{j=0}^k N_{ij} + \sum_{j=0}^k N_{ji} - N_{ii}} \quad (5.30)$$

ここで各評価指標が何を示しているかについて説明する。まずピクセル正解率 PA は, 全てのピクセルにおいて予測結果が正しいピクセルの割合を示す。クラス別平均ピクセル正解率 mAP は, 意味ラベルの各クラス別にピクセル正解率を計算し, それを平均した値になる。これはクラス別の平均となるためクラス別の出現頻度によるピクセル正解率が緩和される。クラ

ス別平均重なり度 $mIoU$ は、正解の領域と予測結果の領域の重なりがどれくらい重なっているかを示しており、正解の領域と予測結果の領域の共通領域に存在するピクセル数をそれらの和集合に存在するピクセル数で割った値を、各クラス別に計算し平均したものである。最後に頻度重み付きクラス別重なり度 $fwIoU$ は、クラス別平均重なり度について各クラスの出現頻度に応じて重み付けをした値となる。

5.7.2.5 物体インスタンス可視・遮蔽領域分割の評価指標

Mask-RCNN を元にした把持点予測器による物体インスタンス可視・遮蔽領域分割について先行研究 [125, 126] で定義され、物体インスタンス可視・遮蔽領域分割の研究 [111] で拡張されている以下の 6 つの評価指標を導入して評価を行う。

- クラス別平均平均適合率 (Mean Average Precision): mAP
- クラス別平均領域分割精度 (Mean Segmentation Quality): mSQ
- 可視領域に関するクラス別平均領域分割精度 (Mean Segmentation Quality for Visible Region): mSQ^{vis}
- 遮蔽領域に関するクラス別平均領域分割精度 (Mean Segmentation Quality for Occluded Region): mSQ^{occ}
- クラス別平均インスタンス検出精度 (Mean Recognition Quality): mRQ
- クラス別平均インスタンス領域分割精度 (Mean Panoptic Quality): mPQ

はじめにクラス別平均平均適合率 mAP について説明する。この指標は正であると予測したデータの内に実際に正であるものの割合を示すものであり、これを意味ラベルのクラス別で計算して平均を取ることでクラス毎における出現頻度を考慮した指標となっている。この指標の計算式を求めるためにまず 2 つの物体のインスタンス領域 p, g の重なり度 $IoU_{inst}(p, g)$ を定義する。まず 2 つの物体のインスタンス領域 p, g の重なり度 $IoU_{inst}(p, g)$ を、 p, g の共通集合に属するピクセル数を $|p \cap g|$ 、 p, g の和集合に属するピクセル数を $|p \cup g|$ を用いて式 (5.31) のように定義する。

$$IoU_{inst}(p, g) = \frac{|p \cap g|}{|p \cup g|} \quad (5.31)$$

次に意味ラベルのクラス i とインスタンス意味ラベル $c_{inst} \in C_{inst}$ に関する予測結果陽性のインスタンス領域の集合 $P_i^{c_{inst}}$ と正解のインスタンス領域の集合 $G_i^{c_{inst}}$ を、それぞれ正解かつ陽性のインスタンス領域の集合 $TP_i^{c_{inst}}$ 、不正解かつ陽性のインスタンス領域の集合 $FP_i^{c_{inst}}$ 、正解かつ陰性のインスタンス領域の集合 $FN_i^{c_{inst}}$ の 3 つの領域の集合に分類する。このときインスタンス意味ラベル集合 C_{inst} は可視領域ラベル vis と遮蔽領域ラベル occ の集合であり、 $C_{inst} = \{vis, occ\}$ と表す。まず意味ラベルのクラス i に関する予測結果陽性のインスタンス領域の集合 $P_i^{c_{inst}}$ と正解のインスタンス領域の集合 $G_i^{c_{inst}}$ について、以下の式 (5.32) と式 (5.34)、式 (5.33)、式 (5.35) が成立する。

$$TP_i^{c_{inst}} \cup FP_i^{c_{inst}} = P_i^{c_{inst}} \quad (5.32)$$

$$TP_i^{c_{inst}} \cup FN_i^{c_{inst}} = G_i^{c_{inst}} \quad (5.33)$$

$$TP_i^{Cinst} \cap FP_i^{Cinst} = \emptyset \quad (5.34)$$

$$TP_i^{Cinst} \cap FN_i^{Cinst} = \emptyset \quad (5.35)$$

式 (5.31) と式 (5.32), 式 (5.34), 式 (5.33), 式 (5.35) より, 正解かつ陽性のインスタンス領域の集合 TP_i^{Cinst} , 不正解かつ陽性のインスタンス領域の集合 FP_i^{Cinst} , 正解かつ陰性のインスタンス領域の集合 FN_i^{Cinst} の3つの領域の集合はそれぞれ式 (5.36), 式 (5.37), 式 (5.38) のように表される.

$$TP_i^{Cinst} = \{(p_i^{Cinst}, g_i^{Cinst}) | p_i^{Cinst} \in P_i^{Cinst}, g_i^{Cinst} \in G_i^{Cinst}, IoU_{inst}(p_i^{Cinst}, g_i^{Cinst}) \geq 0.5\} \quad (5.36)$$

$$FP_i^{Cinst} = \{p_i^{Cinst} | p_i^{Cinst} \in P_i^{Cinst}, (p_i^{Cinst}, \forall g_i^{Cinst}) \notin TP_i^{Cinst}\} \quad (5.37)$$

$$FN_i^{Cinst} = \{g_i^{Cinst} | g_i^{Cinst} \in G_i^{Cinst}, (\forall p_i^{Cinst}, g_i^{Cinst}) \notin TP_i^{Cinst}\} \quad (5.38)$$

また各集合に属する領域の個数をそれぞれ $|TP_i^{Cinst}|$, $|FP_i^{Cinst}|$, $|FN_i^{Cinst}|$ と表す. このとき意味ラベルのクラス i に関する物体インスタンス可視領域の再現率 r_i^{vis} と適合率 p_i^{vis} は式 (5.39), 式 (5.40) のように定義される. またこの時に再現率 r_i^{vis} に関する適合率 p_i^{vis} の関数を式 (5.41) のように表す.

$$r_i^{vis} = \frac{|TP_i^{vis}|}{|TP_i^{vis}| + |FN_i^{vis}|} \quad (5.39)$$

$$p_i^{vis} = \frac{|TP_i^{vis}|}{|TP_i^{vis}| + |FP_i^{vis}|} \quad (5.40)$$

$$p_i^{vis} = Prec(r_i^{vis}) \quad (5.41)$$

そしてある再現率 r_i^{vis} における適合率を得るために補間した適合率 $Prec_{interp}(r_i^{vis})$ を定義する. この補間した適合率 $Prec_{interp}(r_i^{vis})$ は, 再現率 \tilde{r}_i の際に測定される適合率 $Prec(r_i^{\tilde{vis}})$ を用いて式 (5.42) のように定義する.

$$Prec_{interp}(r_i^{vis}) = \max_{r_i^{\tilde{vis}}: r_i^{\tilde{vis}} \geq r_i^{vis}} Prec(r_i^{\tilde{vis}}) \quad (5.42)$$

この時, クラス別平均平均適合率 mAP は, 0 から 1 の間 11 等分割した再現率 $r_i^{vis} \in \{0.0, 0.1, \dots, 1.0\}$ についての補間した適合率 $Prec_{interp}(r_i^{vis})$ の平均を, 更に背景を除く意味ラベルのクラス別平均として式 (5.43) のように求められる. このとき k は背景を除いた意味ラベルのクラス数である.

$$mAP = \frac{1}{k} \sum_{i=1}^k \left(\frac{1}{11} \sum_{r_k \in \{0.0, 0.1, \dots, 1.0\}} Prec_{interp}(r_i^{vis}) \right) \quad (5.43)$$

次にクラス別平均領域分割精度 mSQ について説明する. 本研究では可視領域と遮蔽領域の2種類のインスタンス領域について評価するため, 可視領域のインスタンス意味ラベル vis と遮蔽領域のインスタンス意味ラベル occ についてのクラス別平均重なり度 mSQ^{vis} と mSQ^{occ} も評価指標として用いる. 意味ラベルのクラス i に関する可視領域・遮蔽領域の領域

分割精度 SQ_i^{vis} と SQ_i^{occ} は式 (5.44) と式 (5.45) に示すように得られる。 TP_i^{vis} と TP_i^{occ} を用いてそれぞれ式 (5.46) と式 (5.47) のように計算する。

$$SQ_i^{vis} = \frac{\sum_{(p_i^{vis}, g_i^{vis}) \in TP_i^{vis}} IoU_{inst}(p_i^{vis}, g_i^{vis})}{|TP_i^{vis}|} \quad (5.44)$$

$$SQ_i^{occ} = \frac{\sum_{(p_i^{occ}, g_i^{occ}) \in TP_i^{occ}} IoU_{inst}(p_i^{occ}, g_i^{occ})}{|TP_i^{occ}|} \quad (5.45)$$

この結果を用いて可視領域・遮蔽領域のクラス別平均領域分割精度 mSQ^{vis} と mSQ^{occ} は式 (5.46) と式 (5.47) に表すように計算する。

$$mSQ^{vis} = \frac{1}{k} \sum_{i=1}^k SQ_i^{vis} \quad (5.46)$$

$$mSQ^{occ} = \frac{1}{k} \sum_{i=1}^k SQ_i^{occ} \quad (5.47)$$

以上より式 (5.46) と式 (5.47) を用いて、クラス別平均領域分割精度 mSQ は式 (5.48) のように定義する。このとき $|C_{inst}|$ はインスタンス意味ラベルのクラス数を表す。

$$mSQ = \frac{1}{|C_{inst}|} \sum_{c_{inst} \in C_{inst}} mSQ^{c_{inst}} \quad (5.48)$$

このとき SQ_i は意味ラベルのクラス i に関する平均領域分割精度である。

次にクラス別平均インスタンス検出精度 mRQ について説明する。意味ラベルのクラス i に関するインスタンス検出精度 RQ_i は $|TP_i^{vis}|$, $|FP_i^{vis}|$, $|FN_i^{vis}|$ を用いて式 (5.49) のように求める。

$$RQ_i = \frac{|TP_i^{vis}|}{|TP_i^{vis}| + \frac{1}{2}|FP_i^{vis}| + \frac{1}{2}|FN_i^{vis}|} \quad (5.49)$$

以上より式 (5.49) を用いてクラス別平均インスタンス検出精度 mRQ は式 (5.50) のように定義する。

$$mRQ = \frac{1}{k} \sum_{i=1}^k RQ_i \quad (5.50)$$

このとき RQ_i は意味ラベルのクラス i に関する平均インスタンス検出精度である。

最後にクラス別インスタンス領域分割精度 mPQ について説明する。クラス別インスタンス領域分割精度 mPQ は、意味ラベルのクラス i とインスタンス意味ラベル c_{inst} に関する領域分割精度 $SQ_i^{c_{inst}}$ と意味ラベルのクラス i に関するインスタンス検出精度 RQ_i を用いて式 (5.51) のように定義する。

$$mPQ = \frac{1}{|C_{inst}|} \sum_{c_{inst} \in C_{inst}} \left(\frac{1}{k} \sum_{i=1}^k SQ_i^{c_{inst}} \cdot RQ_i \right) \quad (5.51)$$

ここで各評価指標が何を示しているかについて説明する．まずクラス別平均平均適合率 mAP は，予測した物体インスタンスの適合率の各クラス別平均値である．クラス別平均領域分割精度 mSQ は，正解かつ陽性である物体インスタンスについて，物体インスタンス領域の予測結果と正解がどのくらい重なっているかを表した値を，各クラス別に平均した値となっている．このとき mSQ_{vis} と mSQ_{occ} は各物体インスタンスの可視・遮蔽領域に関して計算しており， mSQ はこの2つの値の平均の値を示す．クラス別平均インスタンス検出精度 mRQ は，どのくらい物体インスタンスを検出したかのクラス別の平均を示した値であり，クラス別平均インスタンス領域分割精度 mPQ は mSQ と mRQ を掛けあわせた値となっており，物体検出かつ物体インスタンス内の領域分割の性能について示した値となっている．

5.7.2.6 意味領域分割と物体インスタンス可視・遮蔽領域分割の評価データセットの作成

本項では意味領域分割と物体インスタンス可視・遮蔽領域分割の評価データセットの作成方法について説明する．図 5.42 に示すように籠の中に対象 9 物品をランダムに積み重ねて配置し，上から 1 物品ずつ順番に他物品を動かさないように取り除き，1 物品取り除くたびに撮影することで RGB 画像を収集する．このとき各 RGB 画像において最も上に存在する物品の領域をアノテーションすることで，それ以前の同一シーンの画像におけるその物品の遮蔽領域をアノテーションすることができる．図 5.42 は左から順に 1 物品ずつ取り除いたシーンのアノテーション結果であり，上段が可視領域，下段が遮蔽領域のアノテーション結果を示している．

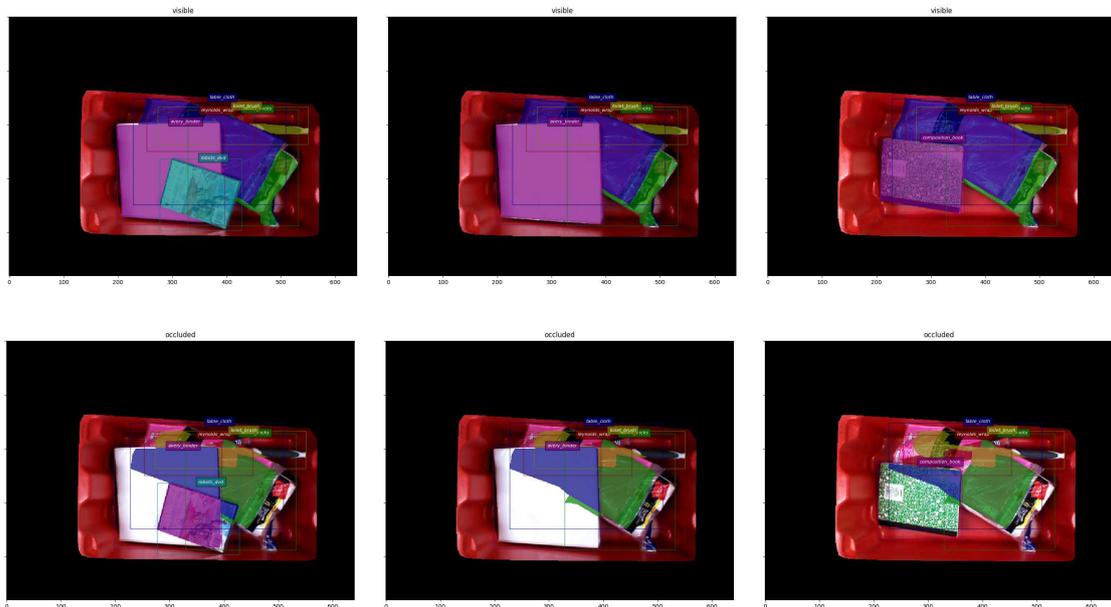


図 5.42: 意味領域分割と物体インスタンス可視・遮蔽領域分割の評価データセット

図 5.42 の左下の画像では DVD がバインダーに重なっているため DVD の領域がバインダーの遮蔽領域としてアノテーションされているが，中下の画像では DVD が取り除かれたためバ

インダーの遮蔽領域はなくなっている．このようにランダムな順番で9物品を重ねた10シーンについて撮影とアノテーションを行い，計90データペアのデータセットを作成した．また比較実験のための比較用学習データセットとして6シーンの計54データペアのデータセットも同様にして作成した．さらにこれらデータセットは物体インスタンス単位でアノテーションされているため，意味領域分割と物体インスタンス可視・遮蔽領域分割の双方の検証実験に用いることができる．作成した評価データセットと比較用学習データセットを用いて意味領域分割と物体インスタンス可視・遮蔽領域分割の検証実験を行う．

5.7.2.7 FCN を元にした把持点予測器による意味領域分割の評価

FCN を元にした把持点予測器による意味領域分割の評価として，5.7.2.4 目で説明したピクセル正解率 PA ，クラス別平均ピクセル正解率 mPA ，クラス別平均重なり度 $mIoU$ ，クラス別頻度重み付き重なり度 $fwIoU$ の4指標を用いて評価を行う．5.7.2.6 目で作成した評価データセットに対して評価を行ったところ，表 5.10 上段に示す結果が得られ，自動生成データセットで学習したモデルでもクラス別平均重なり度 $mIoU$ は 0.595 と高い値を示していることがわかる．また表 5.10 下段は 5.7.2.6 目で作成した比較用学習データセットを用いて学習したモデルの評価結果であり，自動生成データセットで学習したモデルと大差ない結果をしており，クラス別平均重なり度 $mIoU$ も 0.018 だけ低い値を示すという結果が得られた．以上の結果から，本研究で提案する自動データセット生成と FCN を元にした把持点予測器は意味領域分割で高い性能を示しており，このモデルは人手のアノテーションによるデータセットで学習したモデルと大差ない意味領域分割性能を有していると言える．

表 5.10: FCN を元にした把持点予測器による意味領域分割結果

Model	PA	mPA	$mIoU$	$fwIoU$
Model trained with synthesized dataset	0.949	0.716	0.595	0.916
Model trained with human annotated dataset	0.885	0.775	0.613	0.848

5.7.2.8 Mask-RCNN を元にした把持点予測器による物体インスタンス可視・遮蔽領域分割結果

Mask-RCNN を元にした把持点予測器による意味領域分割の評価として，5.7.2.5 目で説明したクラス別平均平均適合率 mAP ，クラス別平均領域分割精度 mSQ ，可視領域に関するクラス別平均領域分割精度 mSQ^{vis} ，遮蔽領域に関するクラス別平均領域分割精度 mSQ^{occ} ，クラス別平均インスタンス検出精度 mRQ ，クラス別平均インスタンス領域分割精度 mPQ の6指標を用いて評価を行う．5.7.2.6 目で作成した評価データセットに対して評価を行ったところ，表 5.11 上段に示す結果が得られ，自動生成データセットで学習したモデルでもクラス

別平均領域分割精度 mSQ は 0.503 と高い値を示していることがわかる。また表 5.11 下段は 5.7.2.6 目で作成した比較用学習データセットを用いて学習したモデルの評価結果であり、自動生成データセットで学習したモデルと大差ない結果をしておりクラス別平均領域分割精度 mSQ も 0.004 だけ高い値を示すという結果が得られた。以上の結果から、本研究で提案する自動データセット生成と Mask-RCNN を元にした把持点予測器は物体インスタンス可視・遮蔽領域分割で高い性能を示しており、このモデルは人手のアノテーションによるデータセットで学習したモデルと大差ない物体インスタンス可視・遮蔽領域分割性能を有していると言える。

表 5.11: Mask-RCNN を元にした把持点予測器による物体インスタンス可視・遮蔽領域分割結果 [35]

Model	mAP	mSQ	mSQ^{vis}	mSQ^{occ}	mRQ	mPQ
Model trained with synthesized dataset	0.491	0.503	0.669	0.337	0.449	0.240
Model trained with human annotated dataset	0.606	0.499	0.621	0.377	0.329	0.169

5.7.2.9 対象 1 物品に対する FCN を元にした把持点予測器を用いた選択的両腕把持動作の検証

選択的両腕把持動作について、本研究では対象 9 物品を 1 つずつ把持することで評価実験を行った。まずロボットの前に 1 つずつ対象物品を置いていき、ロボットは各物品に対してそれぞれ 10 回ずつ計 90 回ほど選択的両腕把持を行い、把持動作の成功回数・成功率について評価を行う。また比較実験として、収集した把持経験による再学習前の把持点予測器と再学習後のモデルでそれぞれ実験を行った。実験は図 5.43 に示すような状況で行われ、実験結果については表 5.12 と表 5.13、表 5.14 に示す。図 5.43 に示すようにロボットは小さな物品については単腕把持を行うようになり、大きく、重く、重心が不安定な物品については両腕把持を行い、物品に応じた選択的両腕把持の実行を行った。

表 5.12 に示すように再学習前と後では把持成功率が 56.7% と 75.6% と大きく異なっており、この結果から把持経験収集による再学習は把持動作の成功率が向上させる効果があると言える。また単腕把持の場合には 62.8% から 78.8%、両腕把持の場合には 16.6% から 50.0% に把持成功率が再学習を経て上昇しており、この結果から把持経験収集による再学習は単腕・両腕把持の両場合において効果があると言える。以上の結果から把持経験収集による再学習を行うことで、ロボットは高い成功率で把持を実行することができ、把持経験データによる再学習によって単腕・両腕把持のそれぞれにおける把持成功確率の向上が確認できた。

本研究ではタスク失敗について把持失敗と自己干渉を設定しているが、自己干渉は把持動作を行う前に検知することができるため回避可能であるが、把持失敗は事前回避不可能であり、把持物品をロボットの作業スペース外に落としてしまい、把持動作再開不能になってしまう場

合がある。こういった可能性から把持失敗は自己干渉よりも回避すべき把持失敗であるが、表 5.13 と表 5.14 に示す通り再学習前では把持失敗発生確率が 33.3% であるのに対して再学習後では 17.8% と大きく減少している。このことから再学習によって把持点予測器が実世界把持経験に適応していると言える。

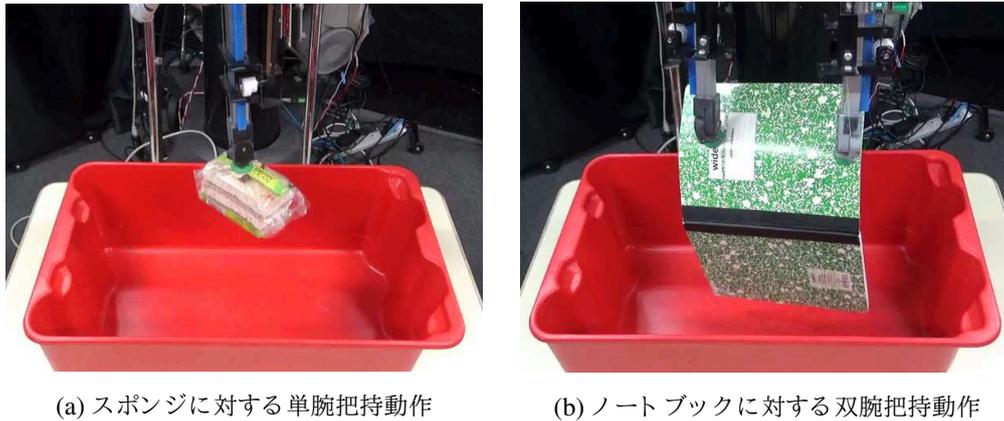


図 5.43: ロボットによる選択的雙腕把持動作 [36]

表 5.12: FCN を元にした把持点予測器を用いた選択的雙腕把持動作結果

	Single-arm success	Dual-arm success	Total success
Before retraining	49 (62.8%)	2 (16.7%)	51 (56.7%)
After retraining	63 (78.8%)	5 (50.0%)	68 (75.6%)

表 5.13: 再学習前の FCN を元にしたモデルを用いた選択的雙腕把持動作結果

	Success	Grasp failure	Self-collision	Total
Single-arm	49 (62.8%)	29 (37.2%)	0 (0.0%)	78
Dual-arm	2 (16.7%)	1 (8.3%)	9 (75.0%)	12
Total	51 (56.7%)	30 (33.3%)	9 (10.0%)	90

5.7.2.10 複雑環境における FCN を元にした把持点予測器を用いたランダムピッキング

FCN を元にした把持点予測器による最終実験として、再学習した把持点予測器とその出力に基づく選択的雙腕把持動作を用いて、複雑環境における選択的雙腕把持動作を行うランダム

表 5.14: 再学習後の FCN を元にしたモデルを用いた選択的雙腕把持動作結果

	Success	Grasp failure	Self-collision	Total
Single-arm	63 (78.8%)	17 (21.2%)	0 (0.0%)	80
Dual-arm	5 (50.0%)	0 (0.0%)	5 (50.0%)	10
Total	68 (75.6%)	17 (18.9%)	5 (5.6%)	90

ピックアップを行った。図 5.44 に示すように対象 9 物品を全て 1 つの籠に入れることで倉庫内の複雑環境を再現した。この際にロボットは複雑環境において重なり合う物体の意味ラベルを正しく認識し、かつ把持成功しうる把持点、把持戦略を選択し実行する必要がある。

本実験では図 5.44a と図 5.44b に示す 2 つの複雑環境でランダムピックアップを行い、図 5.44a のシーン 1 の場合には 9 物品中 6 物品を、図 5.44b に示すシーン 2 の場合には 9 物品中 8 物品を把持し、他の籠へ運搬、収納することができた。ロボットは本実験において図 5.45a に示すような単腕把持動作を大半の場合に実行したが、図 5.44a と図 5.44b の両状態において 1 度ずつ図 5.45b に示すような雙腕把持動作を行った。またアルミホイルは重い物体であるため単腕で把持するには重心を正しく把持する必要があるが、図 5.45a に示すように単腕で重心を把持するように実世界把持経験に適応していることが確認できた。



(a) 複雑環境シーン 1



(b) 複雑環境シーン 2

図 5.44: ランダムピックアップを行った複雑環境 [36]

5.7.2.11 作業失敗予測に基づく 1 物体雙腕ピックアップ作業

選択的雙腕把持動作について、本研究では対象 9 物品を 1 つずつ把持することで評価実験を行った。まずロボットの前に 1 つずつ対象物品を置いていき、ロボットは各物品に対してそれぞれ 10 回ずつ計 90 回ほど選択的雙腕把持を行い、把持動作の成功回数・成功率について評価を行う。また比較実験として、収集した把持経験データによる再学習前の把持点予測器と再学習後のモデルでそれぞれ実験を行った。実験は 5.7.2.9 目同様に図 5.43 に示すような状況で行われ、実験結果については表 5.15 と表 5.16、表 5.17 に示す。

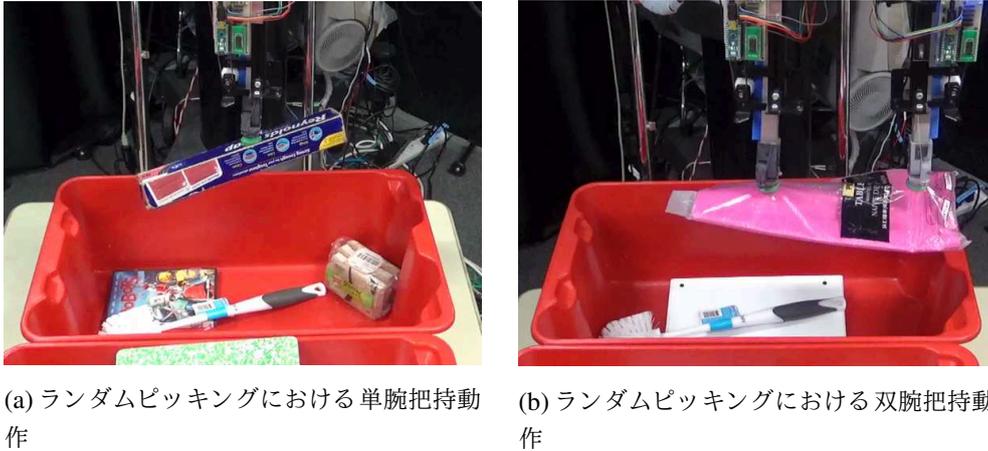


図 5.45: 複雑環境でのランダムピッキングにおける単腕・双腕把持動作 [36]

表 5.15 に示すように再学習前と後では把持成功率が 52.2% と 84.4% と大きく異なっており，この結果から把持経験収集による再学習は把持動作の成功率が向上させる効果があると言える．また単腕把持の場合には 68.1% から 91.5%，双腕把持の場合には 34.9% から 57.9% に把持成功率が再学習を経て上昇しており，この結果から，把持経験収集による再学習は単腕・双腕把持の両場合において効果があると言える．以上の結果から，把持経験収集による再学習を行うことで，ロボットは高い成功率で把持動作を実行することができ，把持経験データによる再学習によって単腕・双腕把持のそれぞれにおける把持成功確率の向上が確認できた．

5.7.2.9 目での評価と同様にタスク失敗の中でも把持失敗について評価する．表 5.16 と表 5.17 に示す通り，再学習前では把持失敗の発生確率が 24.4% であるのに対して，再学習後では 6.7% と大きく減少している．このことから再学習によって把持点予測器が実世界把持経験に適応していると言える．

表 5.15: Mask-RCNN を元にした把持点予測器を用いた選択的雙腕把持動作結果 [35]

	Single-arm success	Dual-arm success	Total success
Before retraining	32 (68.1%)	15 (34.9%)	47 (52.2%)
After retraining	65 (91.5%)	11 (57.9%)	76 (84.4%)

5.7.2.12 作業失敗予測に基づく 3 物体双腕ターゲットピッキング作業

本項ではターゲットピッキングに関する検証実験について説明する．まずターゲットピッキングにおけるタスク成功と失敗について分類したのちに，本研究の対象 9 物品から 3 物品選び

表 5.16: 再学習前の Mask-RCNN を元にしたモデルを用いた選択的雙腕把持動作結果

	Success	Grasp failure	Self-collision	Total
Single-arm	32 (68.1%)	15 (31.9%)	0 (0.0%)	47
Dual-arm	15 (34.9%)	7 (16.3%)	21 (48.8%)	43
Total	47 (52.2%)	22 (24.4%)	21 (23.3%)	90

表 5.17: 再学習後の Mask-RCNN を元にしたモデルを用いた選択的雙腕把持動作結果

	Success	Grasp failure	Self-collision	Total
Single-arm	65 (91.5%)	6 (8.5%)	0 (0.0%)	71
Dual-arm	11 (57.9%)	0 (0.0%)	8 (42.1%)	19
Total	76 (84.4%)	6 (6.7%)	8 (8.9%)	90

重ねることで作成した複雑環境でのターゲットピッキングの実験結果について評価する。

まずターゲットピッキングで実行されるロボットの動作について説明する。ロボットは指定されたターゲット物品を収納位置に把持・収納することが求められており、その際に障害物となる物品は元の位置のまま、もしくは移動位置に退避することができ、ターゲット物品の収納動作は図 5.46 に、障害物物品の移動動作は図 5.47 に示す動作をロボットが実行する。このときターゲット物品は画像手前側、ロボットからみて籠の奥の収納位置にある段ボール箱に置かれ、障害物物品は籠の中もしくは画像上方、ロボットからみて籠の手前側の移動位置にある段ボール箱に置かれる。

次にターゲットピッキングにおけるタスク結果について分類する。ターゲットピッキングのタスク結果は以下の 5 つに分類することができる。

- タスク成功 (Success)
- 物品落下 (Item drop)
- 誤認識 (Mis-recognition)
- 誤把持 (Mis-grasp)
- 障害物移動失敗 (Obstacle removal failure)

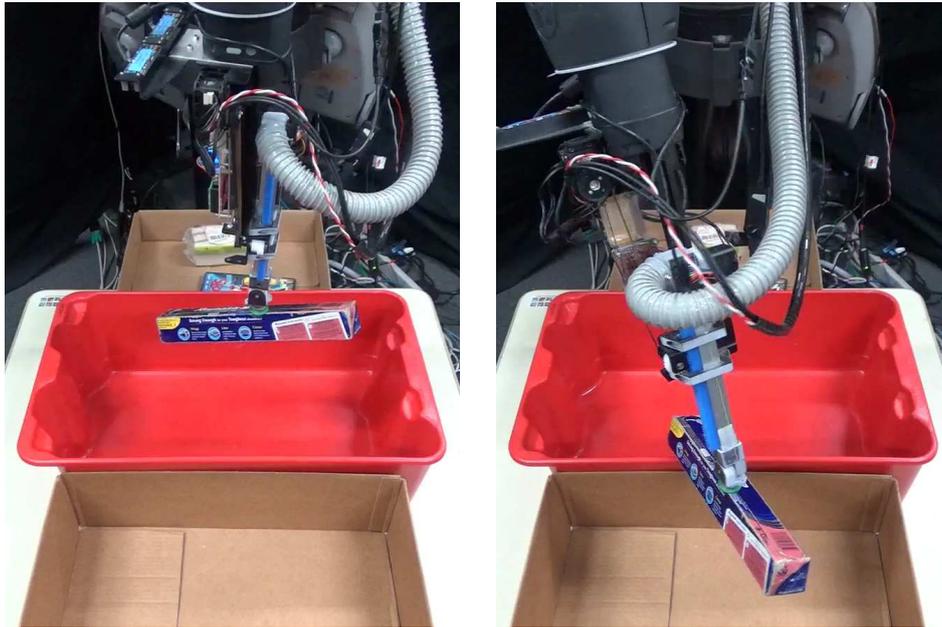
まずタスク成功 (Success) とは、ターゲット物品が指定された収納位置にあり障害物物品が籠もしくは移動位置に置かれており、各物品の最終的な位置をロボットが正しく把握している最終状態である。物品落下 (Item drop) とは、物品を把持・収納動作の途中で落下させてロボットのワークスペース外に出てしまい、作業復帰不能に陥ることである。誤認識 (Mis-recognition) とは、物体認識において物品の意味ラベルを誤って認識してしまい、各物品の最終的な位置をロボットが誤って認識することである。誤把持 (Mis-grasp) とは、物体認識および把持点

表 5.18: 再学習後の Mask-RCNN を元にしたモデルを用いた選択的両腕把持動作の詳細結果 [35]

		Single-arm grasping		Dual-arm grasping	
		Success	Failure	Success	Failure
White binder	Before retraining	1	3	2	4
	After retraining	0	1	4	5
Green notebook	Before retraining	0	9	0	1
	After retraining	0	0	7	3
White socks	Before retraining	3	2	0	5
	After retraining	8	2	0	0
Ice cube tray	Before retraining	2	0	7	1
	After retraining	10	0	0	0
Aluminum foil	Before retraining	3	0	4	3
	After retraining	8	2	0	0
DVD	Before retraining	9	0	0	1
	After retraining	10	0	0	0
Brown sponges	Before retraining	4	0	0	6
	After retraining	10	0	0	0
Pink table cloth	Before retraining	7	0	2	1
	After retraining	10	0	0	0
Toilet brush	Before retraining	3	1	0	6
	After retraining	9	1	0	0
Total	Before retraining	32 (68.1 %)	15 (31.9 %)	15 (34.9 %)	28 (65.1 %)
	After retraining	65 (91.5 %)	6 (8.5 %)	11 (57.9 %)	8 (42.1 %)

認識は正しく行えているものの、把持動作の際に誤って近くにある別物品を把持してしまい、結果として各物品の最終的な位置をロボットが誤って認識することである。障害物移動失敗 (Obstacle removal failure) とは、障害物物品の移動に複数回失敗し続け、そもそもターゲット物品の把持動作を行えないことである。本研究では連続で4回の同じ障害物に対する把持動作を失敗した場合には、障害物移動失敗としてターゲットピッキングを終了した。本検証実験では上記の5つのタスク結果を設定することで評価を行う。

最後に対象3物品に対するターゲットピッキングの検証実験について説明する。各物品1つずつについてターゲットピッキングを実行する際に、対象9物品を図5.41に撮影されている順番に並べ、この物品リスト上でターゲット物品の直後にリストアップされている2つの物品を障害物として、リストの順にターゲット物品の上に重ねることで複雑環境を作成した。



(a) アルミホイル(ターゲット物品)の単腕把持動作

(b) アルミホイル(ターゲット物品)の単腕収納動作

図 5.46: ターゲットピッキングにおけるターゲット物品の単腕把持・収納動作

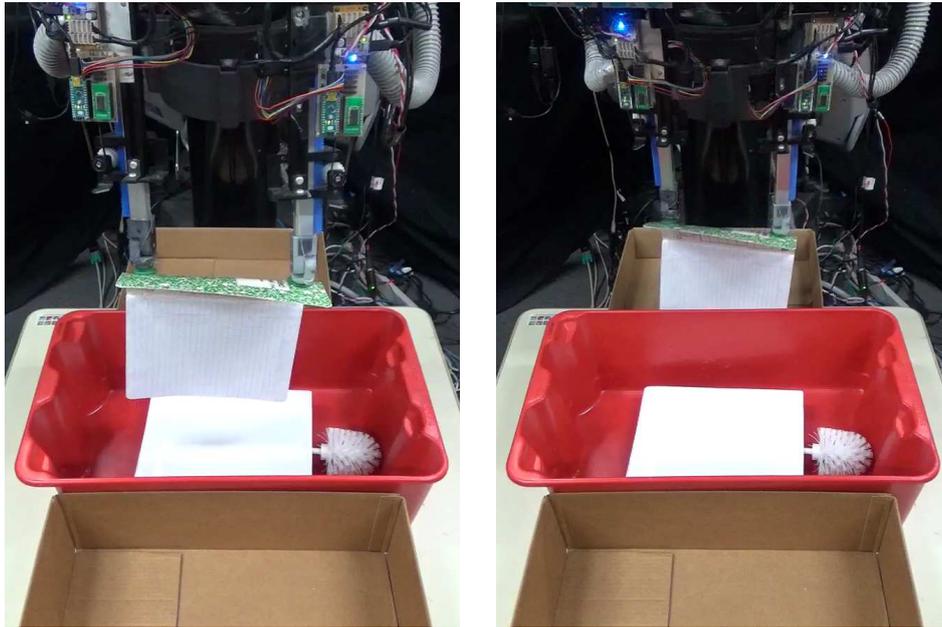
作成した複雑環境は籠の中にターゲット物品と障害物物品が重ねられており、この複雑環境を表 5.21 に示すように計 9 シーン作成して実験を行い、各シーン 2 回のタスク実行、計 18 回のターゲットピッキングを行った。ターゲットピッキング結果は表 5.19 と表 5.20、表 5.21 に示す。以下では各結果について評価していく。

表 5.19 に示すように、本研究で提案するターゲットピッキングシステムは 18 回中 11 回成功の 61.1% と高いタスク成功率を示している。ターゲットピッキングにおけるタスク成功は、前述の 4 つのタスク失敗をどれも発生させずにターゲット物品を収納することで達成されるため、各タスク失敗を高い確率で回避できなければタスク成功は達成できないため、61.1% は十分に高いタスク成功率であると評価できる。

表 5.19: Mask-RCNN を元にしたモデルを用いたターゲットピッキング結果 [35]

Success	Item dropping	Misrecognition	Misgrasping	Object removal failure	Total
11 (61.1%)	1 (5.6%)	3 (16.7%)	1 (5.6%)	2 (11.1%)	18

次に表 5.20 について評価する。この表 5.20 は、表 5.19 において把持成功と物体落下の 12 回のターゲットピッキングにおいて、ロボットがターゲット物品に対してどの把持戦略で把持動作を行ったかの結果を示したものである。このとき単腕・双腕把持動作はそれぞれ 8 回と 4 回実行されており、単腕動作は全動作、双腕動作は 1 回を除いて把持動作を成功してターゲッ



(a) ノートブック (障害物物品) の双腕把持動作

(b) ノートブック (障害物物品) の双腕移動動作

図 5.47: ターゲットピッキングにおける障害物物品の双腕把持・移動動作

ト物品を収納している。この結果から、このターゲットピッキングシステムにおいてもロボットは物品に適した把持戦略を選択的に実行していることがわかる。

表 5.20: Mask-RCNN を元にしたモデルを用いたターゲットピッキングにおけるターゲット物品の把持結果

	Success	Grasp failure	Self-collision	Total
Single-arm	8	0	0	8
Dual-arm	3	1	0	4
Total	11	1	0	12

しかし表 5.21 に示すように、バインダーの把持の不安定さが物品落下と障害物移動失敗、袋入りくつしたを製氷皿と認識してしまうことが誤認識のタスク失敗を引き起こしている。これは本システムにおいて、ターゲットピッキングが得意な物品とそうでない物品が存在することが表しており改善すべき点である。

5.7.2.13 作業失敗予測に基づく狭隘環境における双腕ターゲットピッキング作業

Mask-RCNN を元にした把持点予測器による最終実験として、再学習した把持点予測器とその出力に基づく選択的雙腕把持動作を用いて、対象 9 物品が重なりあう複雑環境における選択

表 5.21: Mask-RCNN を元にしたモデルを用いたターゲットピッキングの各シーンの結果 [35]

Target item	Non-target items	1st trial	2nd trial
White binder	Green notebook & White socks	Success	Item dropping (The robot dropped White Binder)
Green notebook	White socks & Ice cube tray	Success	Success
White socks	Ice cube tray & Aluminum foil	Misrecognition (The robot recognized Ice cube tray as White socks)	Misrecognition (The robot recognized Ice cube tray as White socks)
Ice cube tray	Aluminum foil & DVD	Success	Success
Aluminum foil	DVD & Brown sponges	Success	Success
DVD	Brown sponges & Pink table cloth	Success	Success
Brown sponges	Pink table cloth & Toilet brush	Misrecognition (Brown sponges not found)	Success
Pink table cloth	Toilet brush & White binder	Misgrasping (The robot grasped Pink table cloth as Toilet brush)	Object removal failure (The robot could not remove White binder)
Toilet brush	White binder & Green notebook	Object removal failure (The robot could not remove White binder)	Success

的両腕把持動作を行うターゲットピッキングを行った。このときバインダー、ノートブック、袋入りくつつた、アルミホイル、トイレブラシの5物品をターゲット物品として、その他の物品を障害物物品として設定し、図 5.48 に示すように全ての物品を籠のなかにランダムに重ねて配置してターゲットピッキングを行った。またこの実験では物品の認識や把持が困難でタスク実行が終了できない場合が発生したため、その際には人が籠の中にある物品を重ねなおしてタスクを再開した。

本実験では図 5.48a と図 5.48b に示す 2 つの複雑環境でターゲットピッキングを行った。両シーンともにテーブルクロスをターゲット物品と誤認識して収納しているが、それ以外は全てのターゲット物品を正しく収納している。また両シーンでともにバインダーとノートブックの把持の際に両腕把持動作を選択して行い、その他の物品に対しては単腕把持動作を行った。また図 5.49 に示すように、複雑環境においてロボットはターゲット物品であるバインダーを正しくターゲット物品であると認識し、かつ安定的に把持できる両腕把持動作とその把持点を適切に選択することで、把持・収納動作を問題なく実行した。さらに図 5.50 では、ロボットは DVD をターゲット物品を把持する際の障害物であると認識し、かつ安定的に把持できる単腕把持動作とその把持点を適切に選択することで、把持・移動動作を問題なく実行した。



(a) 複雑環境シーン 1

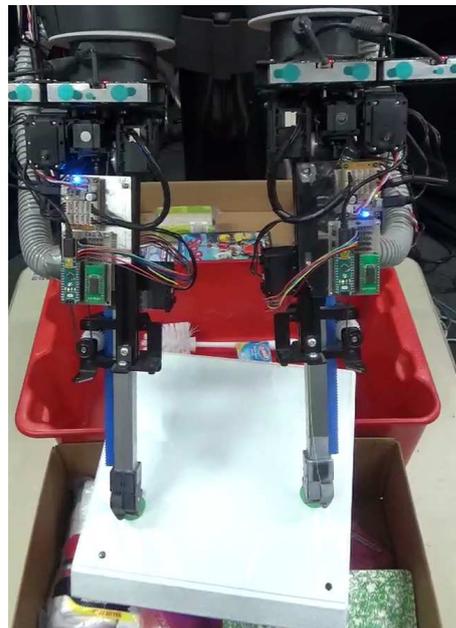


(b) 複雑環境シーン 2

図 5.48: ターゲットピッキングを行った複雑環境



(a) バインダー(ターゲット物品)の双腕把持動作



(b) バインダー(ターゲット物品)の双腕収納動作

図 5.49: 複雑環境でのターゲットピッキングにおけるターゲット物品の双腕把持・収納動作 [35]



(a) DVD(障害物物品)の単腕把持動作



(b) DVD(障害物物品)の単腕移動動作

図 5.50: 複雑環境でのターゲットピッキングにおける障害物物品の単腕把持・移動動作 [35]

5.8 議論

本節では本研究で提案する実世界適応による動作戦略の自律学習について議論する。

5.8.1 再学習による新たな把持点の獲得

本項では 5.6.4 項で説明した再学習の効果を表した図 5.51 に示す 2 つの例について説明する。

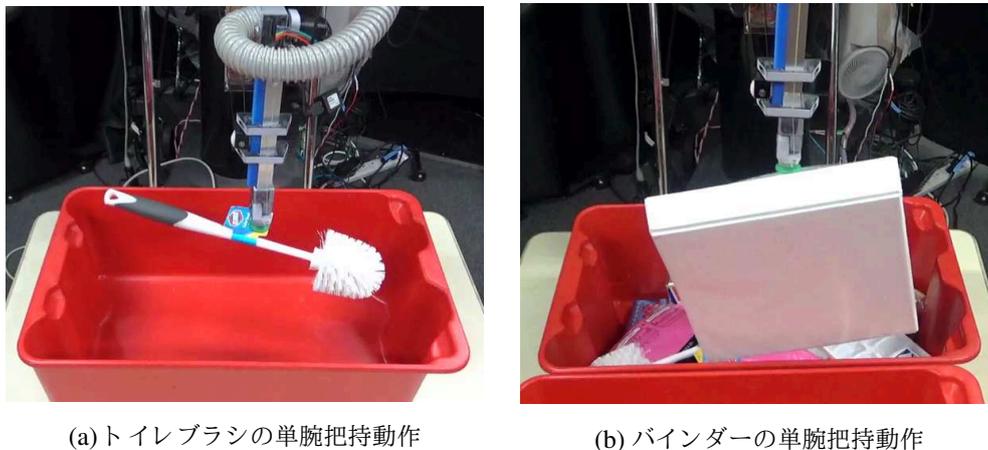


図 5.51: 再学習により新たに獲得した把持点への把持動作

まず図 5.51a について説明する。図 5.51a は再学習した FCN を元にした把持点予測器を用いてトイレブラシを把持する様子を撮影したものであり、このときロボットは 5.5.3.3 目で説明した自動把持点アノテーションにおいてアノテーションされた把持点ではなく、トイレブラシの商品タグを把持することで安定的にタスクを実行した。これはトイレブラシが細い物体であり自動アノテーションされた把持点(物体領域の中心)では安定的に把持できず、また把持試行によって得られた商品タグを新たな把持点として再学習し獲得したことを表している。

次に同様のケースを示した図 5.51b について説明する。ロボットは単腕把持の際に物体領域の中心ではなくバインダーの背の近くの点を把持し安定的にタスクを実行した。バインダーは本と同様に背に回転関節を持っており、ロボットは実世界における把持経験からバインダーの背の近くを把持することが安定的であると学習したと考えられる。本やバインダーなどの背に回転関節を有した物体は重心や回転トルクの関係から、背の近くを把持することで安定的にタスク実行が行える。

以上の 2 つの例では 5.6.4 項で説明した再学習により把持点予測器が把持経験に適応し、適した把持点を新たに獲得したと考えられる。

5.8.2 実世界適応による動作戦略の改善

本節では 5.6.4 項で説明した再学習による把持に適さない把持点の選定について表 5.22 と表 5.23 を例にして説明する。

まずトイブラシの場合について表 5.22 を用いて説明する。表 5.22 は 5.7.2.9 目と 5.7.2.11 目で行った対象 1 物品への選択的雙腕把持実験のトイブラシに関する結果である。表 5.22 が示すように、ロボットは再学習前の FCN を元にしたモデルを用いた場合にはトイブラシに対して雙腕把持を 4 回実行し、再学習前の Mask-RCNN を元にしたモデルを用いた場合にはトイブラシに対して雙腕把持を 6 回実行したが全てにおいて把持失敗するという結果になった。しかし再学習後の場合には、雙腕把持でトイブラシは把持困難であることを把持試行で収集した把持経験から学習したため、ロボットはどちらのモデルの場合にも雙腕把持を一度も行わずにすべて單腕把持を実行し、それぞれモデルの場合で 10 回中 6 回と 9 回の把持成功を記録した。

一方、ノートブックの場合については正反対な選定が行われた。表 5.23 は 5.7.2.9 目と 5.7.2.11 目で行った対象 1 物品への選択的雙腕把持実験のノートブックに関する結果である。表 5.23 に示すように、ロボットは再学習前の FCN を元にしたモデルを用いた場合も再学習前の Mask-RCNN を元にしたモデルを用いた場合にも、ノートブックに対して單腕把持を 9 回実行し全てで把持失敗を記録した。しかし再学習後の把持点予測器を用いた場合には、ロボットは單腕把持でノートブックが把持困難であることを把持試行で収集した把持経験から学習したため、ロボットは單腕把持を一度も行わずにすべて雙腕把持を実行し、それぞれのモデルの場合で 10 回中 5 回と 7 回の把持成功を記録した。

以上の結果から再学習によって把持点予測器は各対象物品に対して、把持失敗しやすい把持点を予測結果から選定するように再学習が行われたと考えられる。

表 5.22: トイブラシに対する選択的雙腕把持動作結果

Model	Adaptation	Single-arm		Dual-arm	
		Success	Failure	Success	Failure
FCN based Model	Before retraining	3	3	0	4
	After retraining	6	4	0	0
Mask-RCNN based Model	Before retraining	3	1	0	6
	After retraining	9	1	0	0

5.8.3 実世界適応における学習サンプルの偏りとその影響

本研究では提案する把持試行を用いた把持点予測器の再学習手法の有効性について議論してきたが、この再学習手法は把持試行で収集される把持経験データの分布に大きく依存する。本

表 5.23: ノートブックに対する選択的両腕把持動作結果

Model	Adaptation	Single-arm		Dual-arm	
		Success	Failure	Success	Failure
FCN based Model	Before retraining	0	9	0	1
	After retraining	0	0	5	5
Mask-RCNN based Model	Before retraining	0	9	0	1
	After retraining	0	0	7	3

節では把持経験データの分布が及ぼす再学習への影響について、表 5.24 に示す 5.7.2.2 目の際に収集された把持経験データの統計と、表 5.25 に示す 5.7.2.3 目の際に収集された把持経験データの統計を用いて議論する。表 5.14 が示すように、5.7.2.9 目で行った評価実験においてロボットは再学習後の把持点予測器を用いて両腕把持を 10 回のみしか実行していない。これは表 5.24 が示すように、把持試行における把持成功率が単腕把持の場合には 51.0% であるのに対して、両腕把持の場合には 17.6% と低いことが原因であると考えられる。また同様に表 5.17 が示すように、5.7.2.11 目で行った評価実験においてロボットは再学習後の把持点予測器を用いて両腕把持を 19 回のみしか実行していない。これは表 5.25 が示すように、把持試行における把持成功率が単腕把持の場合には 57.8% であるのに対して、両腕把持の場合には 21.1% と低いことが原因であると考えられる。

以上の結果から 5.7.2.9 目と 5.7.2.11 目で行った実験では、把持試行で収集した把持経験データを全て用いて把持点予測器の再学習を行ったため、再学習後の把持点予測器は収集した把持経験データの分布も含めて学習し、結果として両腕把持があまり実行されなかったと考えられる。より効率的かつ安定的な再学習のためには、収集した把持経験データについて単腕・両腕把持の分布を正規化する必要がある。把持経験データの分布を正規化するためには、再学習に用いるデータを選定することで把持成功と失敗の回数を同じにし、かつ選定された実世界把持経験によって適切に把持点予測器を再学習する手法を導入する必要がある。

表 5.24: 学習済みの FCN を元にしたモデルを用いて収集された把持経験データの統計 [36]

	Success	Failure	Total
Single-arm	49 (51.0%)	47 (49.0%)	96
Dual-arm	21 (17.6%)	98 (82.4%)	119
Total	70 (32.6%)	145 (67.4%)	215

表 5.25: 学習済みの Mask-RCNN を元にしたモデルを用いて収集された把持経験データの統計 [35]

	Success	Failure	Total
Single-arm	52 (57.8%)	38 (42.2%)	90
Dual-arm	19 (21.1%)	71 (78.9%)	90
Total	71 (39.4%)	109 (60.6%)	180

5.9 おわりに

本章では物体操作作業の自律実行において学習型動作戦略を用いることを提案し、その事前学習にシミュレーションを用いることを提案し、CNN 作業失敗予測器、CNN 把持点予測器を用いた学習型動作戦略を構築した。さらに実世界試行に基づく動作戦略の実世界適応について説明し、提案手法である CNN 把持点予測器の実世界適応について説明した。評価実験として CNN 作業失敗予測器を用いた学習型動作戦略を用いて双腕ロボット Baxter-JSK-APC による棚への双腕収納作業を行い、学習型動作戦略の有効性を確認した。さらに CNN 把持点予測器を用いた学習型動作戦略を用いて、双腕ロボット Baxter-JSK-APC による倉庫環境での自律双腕ピッキング作業実験を行い、学習型動作戦略の実世界適応の有効性を確認した。

第 6 章

物体操作作業の操縦自律間切替実行システム

6.1 はじめに

本章では物体操作作業の操縦自律間切替実行システムについて説明する．本研究における操縦自律間切替実行の目的は手順があり接触の多い複雑な物体操作作業において作業を継続しながら操縦と自律を切り替えることであり，図 6.1 に示すように操縦と自律を切り替える際に操縦実行において教示学習された動作順序を自律実行に，自律実行において自律学習された動作戦略を操縦実行に共有し，それらを用いて作業を継続的に実行することで操縦自律間横断学習を実現する．まず操縦自律間切替実行システムについて議論した後に，操縦自律間切替実行システムの構成要素である物体操作作業の作業実行管理システム，視聴覚統合遠隔操縦システム，作業実行管理システム，実世界経験収集システム，そしてそれらを支える人・環境・物体認識について説明する．最後に操縦自律間切替実行システムの評価実験として，双腕移動ロボット PR2-JSK，単腕移動ロボット Fetch-JSK を用いて様々な物体操作作業実験を行う．

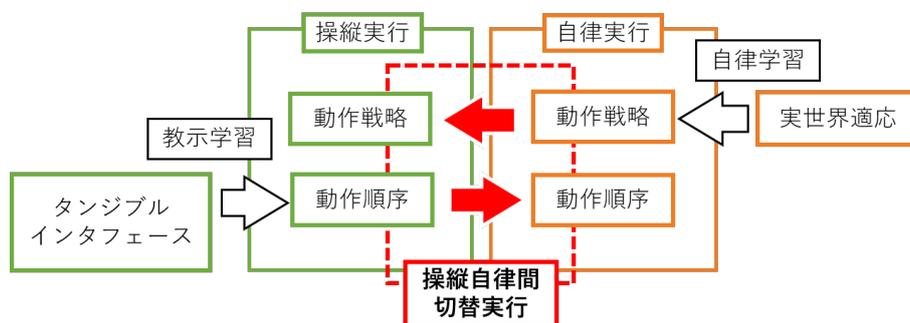


図 6.1: 操縦自律間横断学習における操縦自律間切替実行の位置づけ

6.2 物体操作作業の実行システム

本研究では，物体操作作業の実行の中でも以下の2つの実行形態について取り組む．

- 操縦実行
- 自律実行

操縦実行とは，人が認識や計画に介入しながら物体操作作業を行う実行形態であり，遠隔操縦やテレエグジスタンス分野では盛んに研究が行われている [11, 12]．自律実行とは，ロボットが一人で認識や計画を行いながら物体操作作業を行う実行形態であり，工場や倉庫の自動化分野で研究が行われている [9, 10]．操縦実行と自律実行はこれまで別々に研究が行われてきたが，近年では操縦実行と自律実行の間での作業に関する知識の共有が着目されており，教示学習という側面から操縦実行に基づく自律実行の研究が行われている [31, 38, 39]．しかし，これらの操縦実行で収集した操縦データに対して教示学習を行い，その教示学習結果を元に自律実行をおこなうという研究であり，自律実行から操縦実行に切り替わることはなされておらず，操縦実行からの自律実行の切替も作業実行中に行えるようなシステムにはなっていない．そこで本研究では，タンジブルインタフェースを用いた操縦実行と自律実行を切り替えながら作業を実行することができるような操縦自律実行システムの構築を目指す．

6.3 物体操作作業の操縦自律間切替実行システム

本節では操縦自律間切替実行システムについて説明する．本研究で構築する操縦自律間切替実行システムの全体像は図 6.2 のようになっており，ユーザは音声や Web GUI を通じて操縦実行と自律実行を切り替えながら，タンジブルインタフェースと手先指向型インタフェースを用いて操縦実行を行い，音声や Web GUI を用いてロボットに指示することで作業を自律実行することができるシステムとなっている．操縦自律間切替実行システムは以下の要素で構成されている．

- 操縦自律間切替実行
- 作業実行管理システム
- 視聴覚統合遠隔操縦システム
- 実世界経験収集システム
- 人・物体・環境認識

操縦自律間切替実行とは，その名の通り操縦実行と自律実行を切り替えて実行するためのシステムであり，本研究ではユーザの音声入力に対応して切り替えられるようなシステムを構築する．作業実行管理システムとは，作業の操縦実行と自律実行の両方を管理し，ユーザの音声入力や Web GUI 入力から作業の実行・停止し，作業の状態や結果をユーザに提示するシステムであり，これは作業の音声や Web GUI を通じた実行，もしくはスケジュール実行の際に用

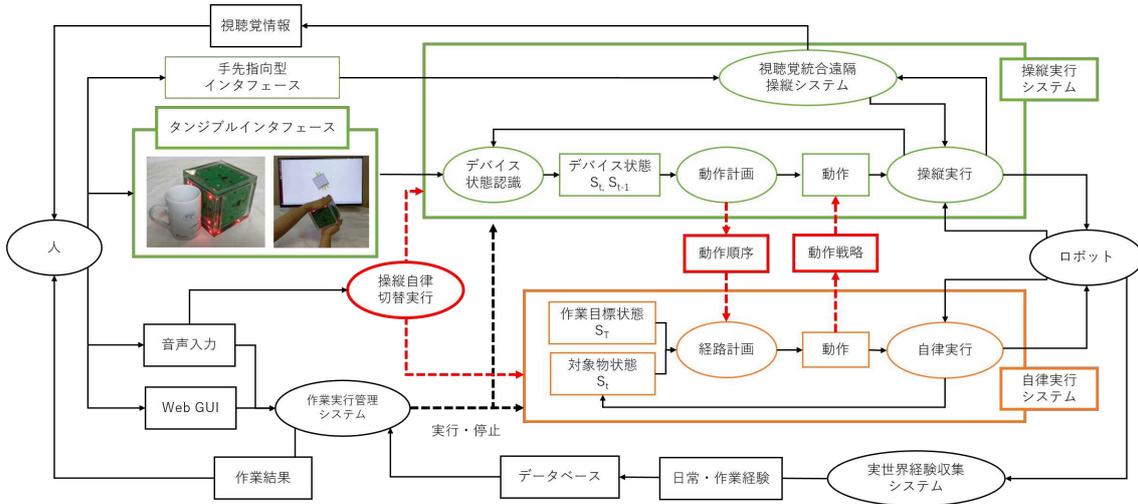


図 6.2: 物体操作作業の操縦自律間切替実行システム

いられる。視聴覚統合遠隔操縦システムとは、遠隔地から操縦実行を行うためのシステムであり、本研究で取り組むタンジブルインタフェースのみでなく VR デバイスなどの手先指向型インタフェースを用いて遠隔地のロボットを操縦し、遠隔地のロボットのセンサから収集された視聴覚情報を操縦者に提示するシステムであり、これは遠隔操縦の際に用いられる。実世界経験収集システムとは、作業の実行やロボットの日常的な動作についてのデータを自動的に収集するシステムであり、これはロボットのメンテナンスや作業実行の結果の収集のために用いられる。人・物体・環境認識とは、物体操作作業のために必要な認識モジュールであり、本研究では GPU サーバ、エッジデバイス、クラウドサービスを状況に応じて使い分ける。以下では各要素について順に説明する。

6.4 ユーザの入力に応じた操縦自律間切替実行

本節ではタンジブルインタフェースの操縦実行で教示学習された動作順序と、実世界適応によって自律学習された動作戦略を共有しながらどのようにして操縦実行と自律実行の切替を行うかについて説明する。操縦実行と自律実行の切替は図 6.2 の一部分であり、図 6.3 に示すように、本研究では操縦実行と自律実行をユーザの入力に基づいて切り替えることで操縦自律間切替実行を実現する。以下では操縦実行から自律実行への切替、自律実行から操縦実行への切替、ユーザの入力に応じた操縦自律間切替実行のそれぞれについて詳しく説明する。

6.4.1 操縦実行から自律実行への切替

本項では操縦実行からの自律実行への切替について説明する。本研究では操縦実行からの自律実行への切替の際には操縦実行で教示学習された動作順序を自律実行に共有することで、操縦実行と自律実行の 2 つの実行形態での横断的な作業学習ができるようなシステムを構築す

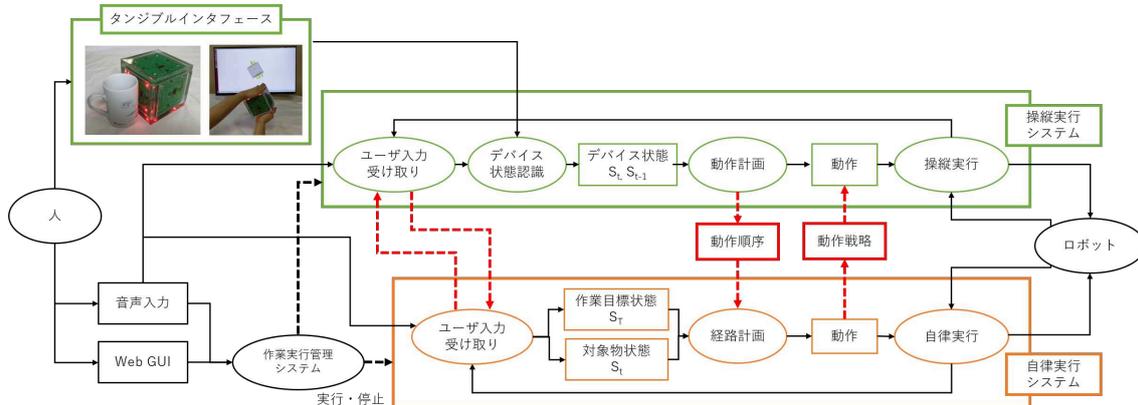


図 6.3: ユーザの入力に応じた操縦自律間切替実行

る．具体的には第 4 章で説明したように，本研究では動作順序を対象物の接触・回転状態をノード，ロボットの動作をエッジとする有向グラフである動作順序グラフとして表現し，タンジブルインタフェースを用いることで操縦実行における動作順序の教示学習を実現する．教示学習された動作順序に基づく自律実行の際には図 6.3 に示すように，操縦実行で学習された動作順序グラフ上で目標状態への経路を計画し，その経路を実行することで作業の自律実行を実現する．このとき操縦実行で学習された動作順序グラフとして表現することで，操縦実行時には実行された作業に含まれる動作順序を実行可能なものとして動作順序グラフに追加でき，かつ自律実行時にはその追加学習された動作順序グラフ上での経路を計画して作業を実行できるため，操縦実行と自律実行を切り替えながら横断的に学習することに適している．本研究では実装としては図 6.4 に示すように，学習された動作順序を用いて経路計画された経路に基づいて自律実行のための有限状態機械を自動生成して実行することで，作業の操縦実行からの自律実行の切替を実現する．なお図 6.4 に示すような学習された動作順序に基づいて生成される有限状態機械は，図 4.14b に示すように各対象物状態を表すノードの間に動作ノードが配置されており，この動作ノードに記述された動作を実行することで作業の実行することができる．

6.4.2 自律実行から操縦実行への切替

本項では自律実行からの操縦実行への切替について説明する．本研究では自律実行からの操縦実行への切替の際には自律実行で自律学習された動作戦略を操縦実行に共有することで，操縦実行と自律実行の 2 つの実行形態での横断的な作業学習ができるようなシステムを構築する．具体的には第 5 章で説明したように，本研究では深層学習を用いて学習型動作戦略を表現し，実世界適応学習を用いることで自律実行における動作戦略の自律学習を実現する．自律学習された動作戦略に基づく操縦実行の際には図 6.3 に示すように，自律実行で自律学習された動作戦略をタンジブルインタフェースの状態変化に基づいて指示される 1 動作として使い，動作を実行する際に学習された動作戦略を用いることで作業の操縦実行を実現する．このとき自律実行で学習された単一動作の動作戦略は自律実行時にはロボットが単一動作の試行を行いな

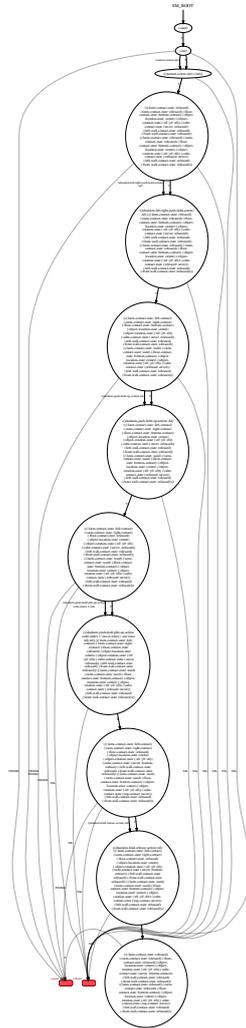


図 6.4: 学習された動作順序を用いた経路計画に基づく自律実行のための有限状態機械(箱積み重ね作業の例)

がら実世界適応することができ、かつ操縦実行時にはその学習された動作戦略を用いて動作を実行することができるため、操縦実行と自律実行を切り替えながら横断的に学習することに適している。本研究では実装としては、学習型動作戦略によって認識された把持点を用いて操縦実行にてタンジブルインタフェースで指示された把持動作を行うことで、作業の自律実行からの操縦実行への切替を実現する。

6.4.3 ユーザの音声入力に応じた操縦自律間切替実行

本項ではユーザの入力に応じた操縦自律間切替実行について説明する。本研究では操縦実行と自律実行を切り替える際にはユーザからの指示を音声や Web GUI からの入力として受け付けることで操縦実行と自律実行を切り替えながら実行できるようなシステムを構築する。本研

究で構築する操縦自律間切替実行は図 6.3 に示すように以下の 2 つの実装で実現する。

- 作業を停止する場合: 作業実行管理システムによる実装
- 作業を継続する場合: 自律実行と操縦実行中にユーザ入力を受け取る実装

この 2 つの実装は使用する状況が異なり、まず作業を停止して切り替える場合には、6.5 節にて後述する作業実行管理システムを用いて作業の操縦実行・自律実行を 1 つずつアプリケーションとして音声入力と Web GUI 入力で作業を停止して新たな作業を実行することで実現する。一方で作業を停止することなく継続して切り替える場合には、作業の実行中でも操縦実行と自律実行の両実行において 1 動作を実行するごとにユーザからの入力を受け取ることによって、音声入力に基づいて作業中に切り替えることを実現する。

作業を停止していい場合とは具体的には作業実行管理システムから操縦実行を行っている状態から自律実行に切り替える場合であり、ユーザは一旦音声や Web GUI を用いて操縦実行を停止し、新たな自律実行を再度音声や Web GUI から実行することができる。一方で作業を停止しない場合とは具体的には作業実行管理システムから自律実行を行っている状態から操縦実行に切り替える場合であり、ユーザは自律実行を音声入力によって中断し、操縦実行によって新たな動作を自律実行中に教示し、さらに音声入力によって操縦実行から再度自律実行に戻って作業を継続することができる。

6.4.3.1 作業を停止する場合の操縦自律間切替実行

まず 1 つめの 6.5 節の作業実行管理システムを用いた実装について説明する。ユーザは作業を操縦実行もしくは自律実行を行う際には、6.5 節の作業実行管理システムを起動し、音声入力や Web GUI 入力から作業実行をアプリケーションとして選択することで作業実行を指示する。またこの作業実行管理システムは作業を停止することも可能であり、作業停止する場合も同様に音声入力や Web GUI 入力を通じて作業実行を停止する。このように作業を停止していい場合であれば、6.5 節の作業実行管理システムを用いることで操縦自律間切替実行を実現する。

6.4.3.2 作業を継続する場合の操縦自律間切替実行

次に 2 つめの自律実行と操縦実行中にユーザ入力を受け取る実装について説明する。前述のとおりユーザは作業を操縦実行もしくは自律実行を行う際には、6.5 節の作業実行管理システムを起動し、音声入力や Web GUI 入力から作業実行をアプリケーションとして選択することで作業実行を指示する。このとき作業を停止せずに自律実行と操縦実行を切り替えたい場合に対応するため、本研究では自律実行と操縦実行において 1 動作を実行するごとにユーザからの音声入力を受け取ることで実現する。具体的な実装としては図 6.5 に示すように、6.4.1 項で説明した自律実行のための動作順序グラフから有限状態機械を自動生成する際に、各動作の実行後にユーザ音声受け取りノード (図 6.6) を追加し、各動作の実行後にユーザの音声入力による自律実行から操縦実行への切替を受け付ける。そして切り替わった教示学習ノード (図 6.7) にて操縦実行することができ、再度自律実行に切り替えたい場合には操縦実行中にユーザは音

声入力を通じて、操縦実行から自律実行へ切り替えることができる実装となっている。以下ではユーザ音声受け取りノードと教示学習ノードについて説明する。

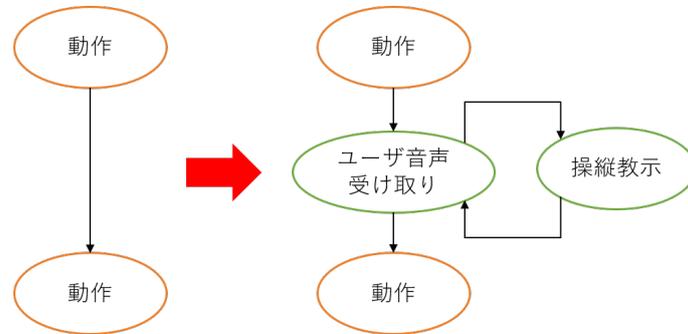


図 6.5: 自律実行時の有限状態機械へのユーザ音声受け取りノードと教示学習ノードの追加のコンセプト

まずユーザ音声受け取りノードについて説明する。ユーザ音声受け取りノードは図 6.6 に示すような構成になっており、主に以下の子ノードによって構成されている。

- (interrupt-wait): ユーザの音声入力を一定時間待つ。
- (ask-user-state-decision): ユーザの音声入力から操縦実行に切り替えるか自律実行を続けるかを選択する。

ロボットが動作実行を終えたらユーザ音声受け取りノードは、まず (interrupt-wait) ノードに遷移しロボットはユーザの音声入力を一定時間待つ。この待ち時間の間にユーザの音声入力(例えば「Wait」など)がある場合には、操縦実行への切替を行う可能性があるともみなして (ask-user-state-decision) に遷移し、ユーザの音声入力がない場合には自律実行を継続するものとみなす。本研究では 5 秒間ユーザの音声入力がない場合には自律実行を継続するものとみなして次の動作を実行する。そして (ask-user-state-decision) ノードにて、再度操縦実行に切り替えることを確認し、ユーザが操縦実行に切り替えることを望んでいる場合には教示学習ノードへ遷移する。もしも (interrupt-wait) ノードでのユーザの音声入力が誤認識だった場合には、自律実行を継続することをユーザは音声で選択することができる。

次に教示学習ノードについて説明する。教示学習ノードは図 6.7 に示すような構成になっており、主に以下の子ノードによって構成されている。

- (ask-user-ready-device): ユーザにタンジブルインタフェースの準備が出来ているかを確認し、準備が整うまで待つ。
- (wait-device-user-input): ユーザのタンジブルインタフェースへの入力を待つ。
- (execute-device-user-input): ユーザのタンジブルインタフェースへの入力に基づいてロボットが動作を行う。

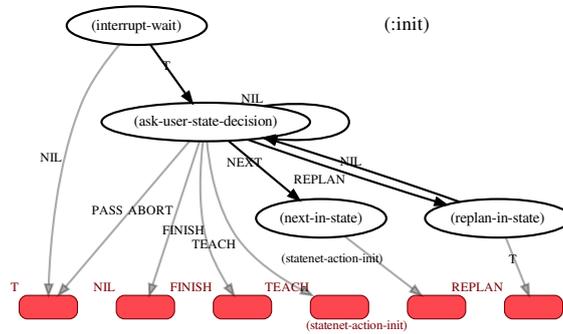


図 6.6: 音声入力に応じるための自律実行から操縦実行への切替するユーザ音声受け取りノード

- (update-from-teach-execution): ユーザのタンジブルインタフェースの操縦実行から動作順序グラフを更新し学習する。
- (ask-user-teach-decision): ユーザに教示学習を継続するかを尋ねる。
- (replan-in-teach): 教示学習ノードから自律実行に再度切り替えるときに、操縦実行で変更された対象物状態から再度自律実行のための有限状態機械を再生成する。

ユーザ音声受け取りノードから遷移してきたら教示学習ノードは、まず (ask-user-ready-device) ノードに遷移しユーザが作業手順の教示学習するためのタンジブルインタフェースが準備できているかを確認する。そしてユーザはタンジブルインタフェースの準備ができたなら、(wait-device-user-input) ノードに遷移しユーザのタンジブルインタフェースに対する入力を待ち、デバイスからの入力を認識できた時点で (execute-device-user-input) に遷移し入力された動作をロボットが実行し、(update-from-teach-execution) に遷移し入力された動作を教示動作順序として動作順序グラフを更新する。そして入力された動作が完了した後は (ask-user-teach-decision) に遷移し操縦実行を継続するか、自律実行に切り替えるか、作業実行を終了するかをユーザは音声入力で決めることができる。操縦実行を継続する場合には再度 (wait-device-user-input) ノードに遷移し同様の作業手順の教示学習を繰り返し、作業実行を終了する場合には FINISH に遷移することで作業実行を終了する。自律実行に切り替える場合には (replan-in-teach) に遷移し、(replan-in-teach) にて操縦実行で変更された対象物状態と作業の目標状態を元に再度追加学習された動作順序グラフ上で経路計画を行うことで図 6.4 に示すような自律実行のための有限状態機械を再生成し、作業の自律実行を継続することができる。なお操縦実行において教示された動作順序は (update-from-teach-execution) にて動作順序グラフを更新する形で保存されているため、(replan-in-teach) では教示された動作順序も用いて経路計画が行われる。

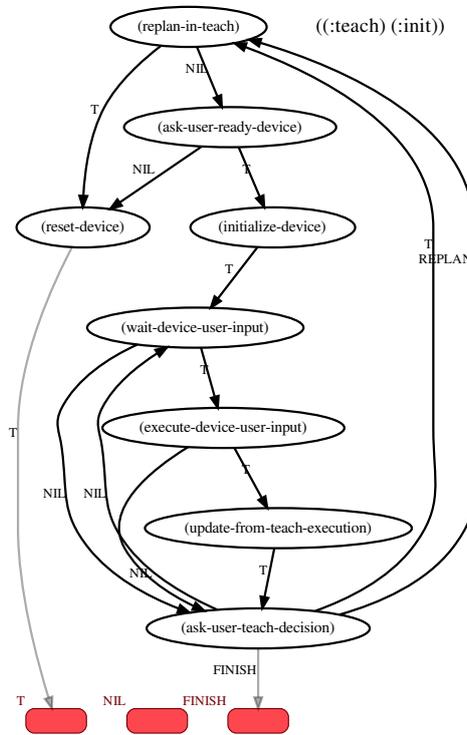


図 6.7: 音声入力に応じて遷移された操縦実行にて教示学習を行う教示学習ノード

以上の2つのユーザ音声受け取りノードと教示学習ノードを図 6.4 に示すような自律実行のための有限状態機械に追加することで、作業実行を継続しながら操縦実行と自律実行を切り替える操縦自律間切替実行に対応することができる。図 6.4 に示す箱積み重ね作業の自律実行のための有限状態機械にユーザ音声受け取りノードと教示学習ノードを追加し、音声入力による操縦自律間切替実行に対応した有限状態機械を図 6.8 に示す。図 6.4 で示されていた対象物状態ノードが、図 6.8 ではユーザ音声受け取りノードに変換されており、各ユーザ音声受け取りノードは教示学習ノードを繋がっており、このノード間を遷移することで操縦実行と自律実行を切り替えることができる。本研究では図 6.8 に示すような、操縦自律間切替実行することができる有限状態機械を自律実行に使うことで操縦自律間切替を実現する。

6.5 物体操作作業の作業実行管理システム

本節では物体操作作業のための作業実行管理システムの実装について説明する。本研究では、各物体操作作業をアプリケーション (app, アプリ) としてアプリケーション実行管理システム `app_manager` [127] に登録・実行することで実現し、`app_manager` のアプリケーションを CUI のみではなく音声、や Web ブラウザから実行できるようにすることでユーザの様々な入

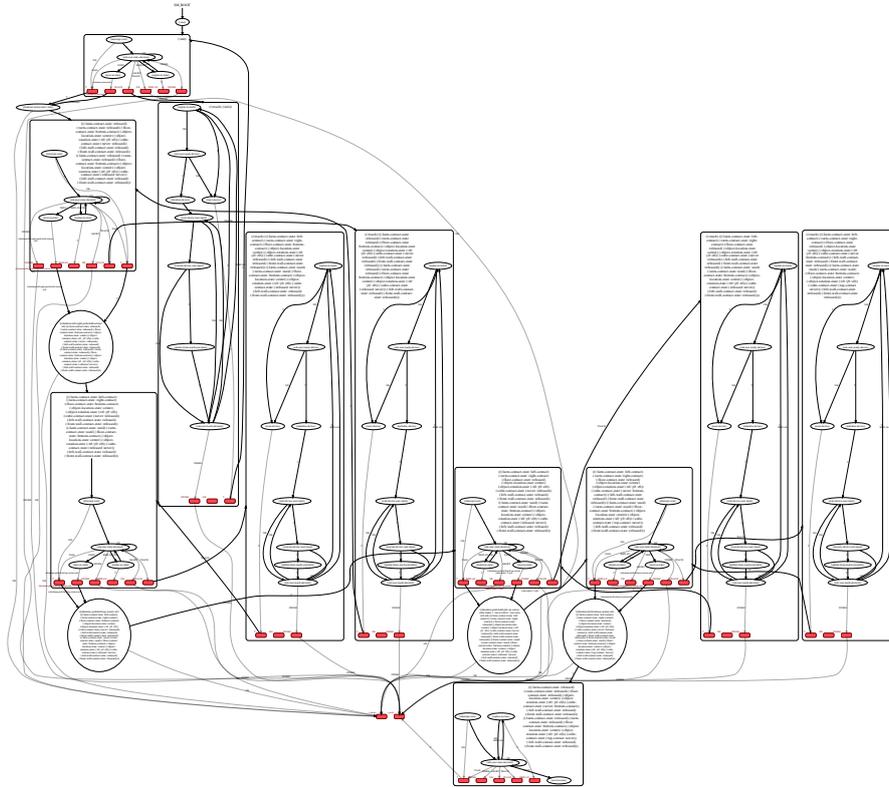


図 6.8: 音声入力による操縦自律間切替実行に対応した自律実行のための有限状態機械(箱積み重ね作業の例)

力に対応し、またアプリケーションを定期的に行うことでスケジュール実行を実現する。本研究で実装する作業実行管理システムの概観については図 6.9 に示す。

6.5.1 物体操作作業の作業管理システム

6.5.1.1 アプリケーション作業管理システム app_manager

本目ではアプリケーション作業管理システム `app_manager` について説明する。`app_manager` は各物体操作作業をアプリケーションとして登録・実行・管理するシステムであり、ユーザはアプリケーションを ROS のサービスコールとして呼び出すことで、アプリケーションをサブプロセスとして実行することができる。またアプリケーションがすでに実行されている場合には、物体操作作業がすでに実行されていると判断して、アプリケーションの実行を拒否することもできる。つまり音声認識プログラムや Web ブラウザは `app_manager` に対してアプリケーション実行を呼び出すことで、物体操作作業を遠隔手続き呼び出し型で実行することができる。またユーザ指示からのアプリケーション停止機能もあり、後述する機能と組み合わせることで音声や Web ブラウザからアプリケーションを停止することもできる。さらにタイムアウト機能もあるため、最大作業時間を設定することで作業にスタックした場合でも自動的に作業

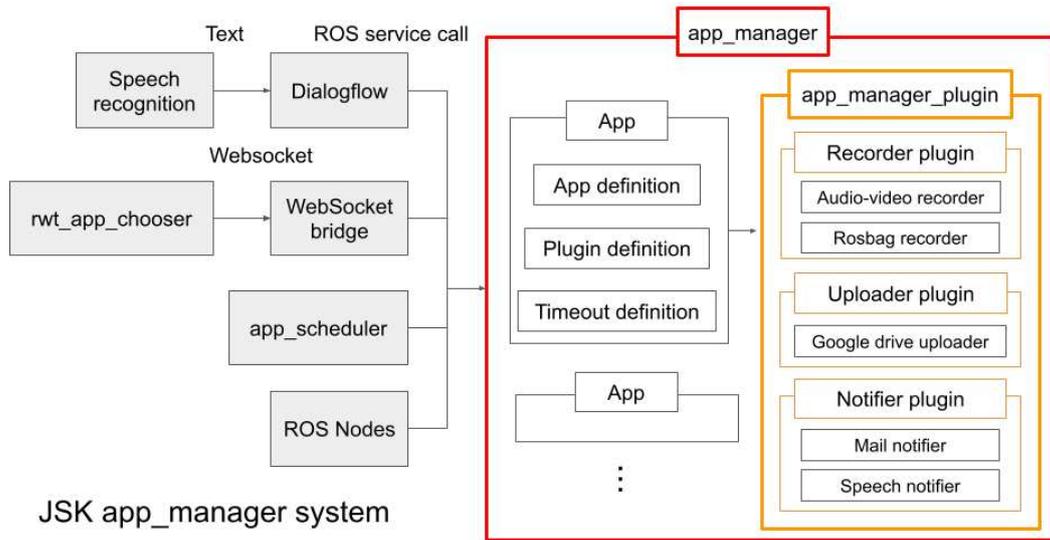


図 6.9: app_manager を中心とした物体操作作業の作業実行管理システム

を終了することができる。

6.5.1.2 アプリケーション作業管理システム拡張プラグイン app_manager_plugin

本目ではアプリケーション実行管理システム app_manager の拡張プラグイン app_manager_plugin について説明する。app_manager_plugin は作業実行管理システム app_manager の各アプリケーションに対して、アプリケーション開始時、実行中、終了時の3つの状態に対してプラグインを追加することができる。実装されているプラグイン機能としては主に以下の3つの機能がある。

- センサデータや作業結果の保存
- データのクラウドストレージサービスへのアップロード
- 音声やメールによるユーザへの通知

これらプラグイン機能を用いることでアプリケーション開始時と終了時に作業結果をメールで送信し音声を発することができ、作業実行中のみのセンサデータを収集し、これを作業終了時にクラウドデータストレージサービスにアップロードすることができる。

6.5.2 物体操作作業の作業実行システム

6.5.2.1 音声ベース作業実行システム dialogflow_task_executive

本目では音声ベース作業実行システム dialogflow_task_executive について説明する。これは6.5.1.1 目で説明した app_manager を自然言語対話から実行するためのシステムである。まずユーザの音声データをクラウドサービスなどを用いて自然言語の文字列に変換した後に、ク

クラウド 意図認識サービスの Dialogflow にて文字列が意図する `app_manager` のアプリケーション呼び出しに変換し，物体操作作業をアプリケーションとして実行するというものである。Dialogflow の意図認識の登録は図 6.10 に示すように，あるアプリケーションに対して自然言語列を複数登録することで実現できる。例えば図 6.10 の例では `PickObject` という冷蔵庫物品取り出し作業のアプリケーションに対して，「ジョージアとってきて」「ボスを持ってきて」といった文字列を登録することで，「ジョージアをもってきて」という文字列に対して `PickObject` アプリケーションを呼び出すことができるシステムである。また図 6.10 に示すように「ジョージア」や「ボス」といったアプリケーションの引数を登録することができ，この引数を使ってアプリケーション実行時に呼び出すことで，物体操作作業を音声からの引数を含んだ遠隔手続き呼び出し型で実行することができる。またアプリケーション停止機能も同様のホットワードを設定することで，「とまって」と呼びかけることでアプリケーションの実行を停止することができる。

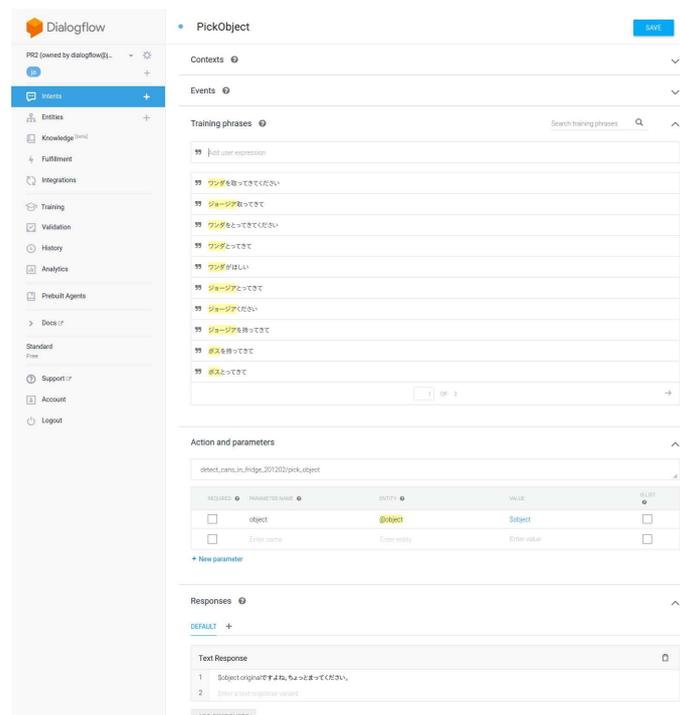


図 6.10: クラウドサービス Dialogflow によるユーザの作業指示文字列の登録

6.5.2.2 Web ブラウザベース作業実行システム `rwt_app_chooser`

本目では Web ブラウザベース作業実行システム `rwt_app_chooser` について説明する。これは 6.5.1.1 目で説明した `app_manager` を Web ブラウザから実行するためのシステムである。まず Web ブラウザから図 6.11 に示すような `rwt_app_chooser` のページをロードする。このページはロボットと WebSocket を介して通信をしており，`rwt_app_chooser` のページにはロボットにおいて実行可能なアプリケーションのアイコンが表示され，ユーザはこのアイコンをクリック

クすることでアプリケーションを実行することができる。またアイコンを実行する際にポップアップウィンドウを用いてパラメータを入力することで、物体操作作業を Web ブラウザからの引数を含んだ遠隔手続き呼び出し型で実行することができる。またアプリケーション停止機能も実行中のアプリケーションのアイコンをクリックすることでアプリケーションの実行を停止することができる。

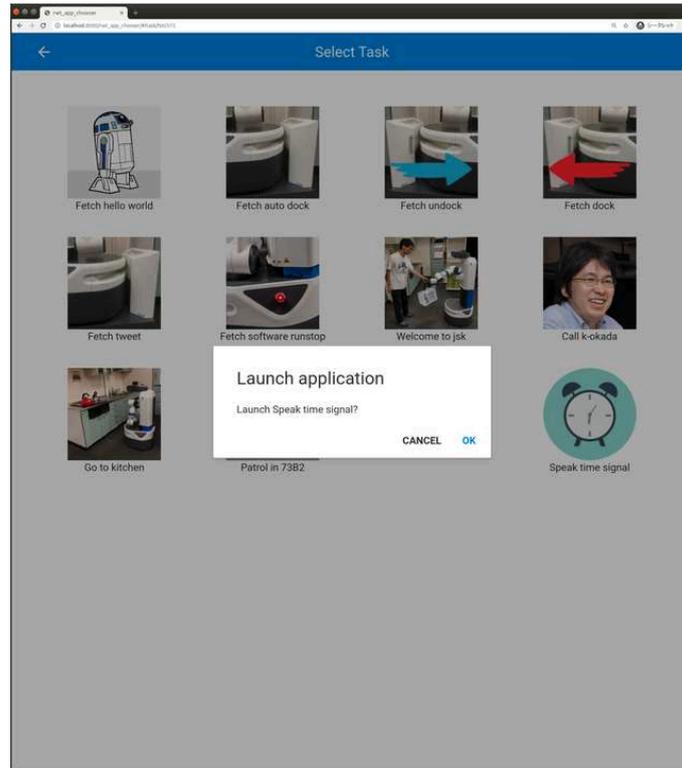


図 6.11: rwt_app_chooser による app_manager のアプリケーションの実行

6.5.2.3 スケジュールベース作業実行システム app_scheduler

本目ではスケジュールベース作業実行システム app_scheduler について説明する。これは 6.5.1.1 目で説明した app_manager をスケジュールに応じて定期的に行うためのシステムである。定期的に行うアプリケーションとそのスケジュールについて YAML 形式で記述することで作業を定期的に行うことができる。

6.6 物体操作作業のための視聴覚統合遠隔操縦システム

本節では物体操作作業のための視聴覚統合遠隔操縦システムの実装について説明する。本研究では第 4 章で説明したタンジブルインタフェースと手先指向型インタフェースのどちらも使える遠隔操縦システムである eus_vive [128] を構築する。手先指向型インタフェースとしては VR デバイスを用いる。このときこれらのユーザインタフェースを用いて遠隔操縦するための

システムを本研究では構築し，これを用いて遠隔操縦実験を行う．まず VR デバイスを用いた手先指向型遠隔操縦システムについて説明し，そのあとにユーザインタフェースにかかわらず使用することができる視聴覚統合遠隔操縦システムについて説明する．

6.6.1 VR デバイスを用いた手先指向型遠隔操縦システム

本研究では視聴覚統合型の手先指向型遠隔操縦システムを構築する．構築する視聴覚統合型遠隔操縦システムは図 6.12 に示すように，VR デバイスのヘッドマウントディスプレイ (HMD)，コントローラの位置姿勢とコントローラのボタン操作を入力として，遠隔地の双腕ロボット Baxter-JSK-Softhand の両腕と頭とソフトハンドをネットワーク越しに動かし，視聴覚情報として，ロボットの頭部カメラの映像，腹部マイクの音声，さらに衝突や手先位置やソフトハンドの開閉状態などを視聴覚情報や振動として，スピーカやヘッドマウントディスプレイ，コントローラを通じて操縦者にフィードバックするものである (図 6.13)．

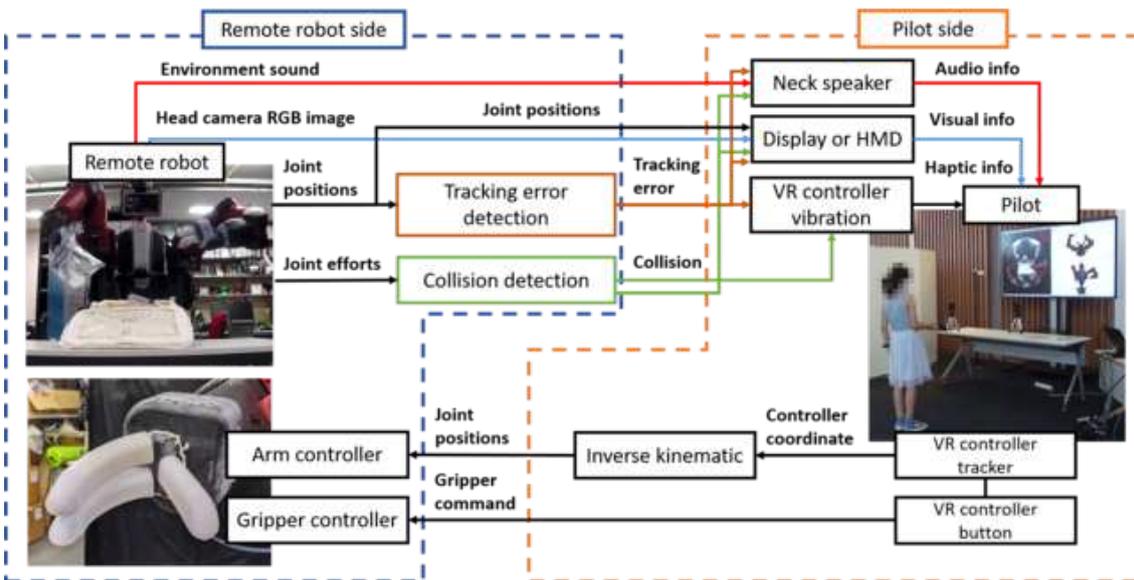


図 6.12: 視聴覚統合型遠隔操縦システムの全体図

6.6.1.1 VR デバイスを用いた遠隔操縦のための手先指向型インタフェース

本研究で用いる視聴覚統合型遠隔操縦システムでは，ロボットの腕と頭，ソフトハンドの操縦インタフェースとして VR デバイスを用いる．まず操縦者の入力として HMD と両手コントローラの位置姿勢を計測し，各位置姿勢を目標座標として逆運動学を制限時間付き収束計算で解くことで関節角度を得る．この関節角度を 10Hz の周期で計算と遠隔地への送信を行い，遠隔地ロボット制御 PC にて Cubic Spline で軌道補間してロボット実機に反映することで，遠隔ロボットの両腕と頭部を操縦する．さらに VR デバイスのボタン入力を押すことでソフトハンドの開閉を指示できるようにすることで，両手コントローラを動かしてボタンひとつを押す

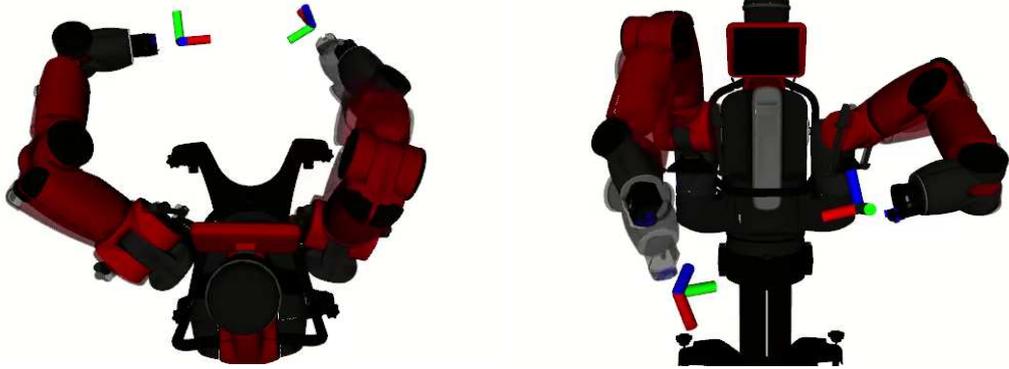


図 6.13: VR デバイスを用いた遠隔操縦の様子

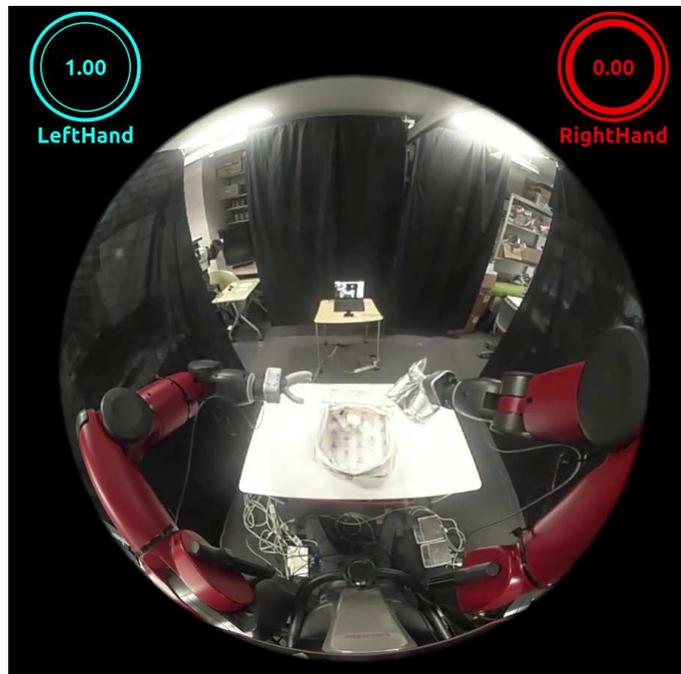
ことで物体を把持できるインタフェースを構築する．また HMD なしコントローラのみでロボットを操縦できるようにすることで，HMD 装着が好ましくない状況やユーザにも対応可能となる．

6.6.2 ロボット状態・周囲環境情報の視聴覚・振動情報提示システム

本研究で用いる視聴覚統合型遠隔操縦システムでは，図 6.14 に示すように遠隔ロボット状態と周囲環境情報の提示のために視聴覚・振動情報を用いる．まず頭部カメラの映像は H.264 などの形式に圧縮された画像データとして，腹部マイクで集音された音は Wav データとして遠隔地へと送信され，それぞれ HMD もしくはディスプレイとネックスピーカに再生することで，遠隔地のロボット状態・周囲環境情報を視聴覚情報としてユーザに提示する（図 6.14c）．さらにソフトハンドの開閉状態についてはロボット自身や環境とのオクルージョンから視認できない場合があるため，開閉に関するアイコンを画面上に表示し（図 6.14c），このアイコンは赤色がソフトハンドが閉じている状態，青色がソフトハンドが開いている状態を示し，同時に効果音でソフトハンドの開閉動作が実行されていることをユーザに提示する．またロボットと環境や対象物が衝突した場合や，ユーザの VR コントローラの操作が速すぎるか到達不可能でロボットが追従できない場合には，効果音と VR コントローラの振動でユーザに警告し，視覚情報として遠隔ロボットの実際の状態とユーザ操作の目標姿勢を提示する（図 6.14a，図 6.14b）．



(a) Baxter-JSK-SoftHand ロボット 身体情報の視覚情報提示システム (正面) (b) Baxter-JSK-SoftHand ロボット 身体情報の視覚情報提示システム (頭上)



(c) Baxter-JSK-SoftHand 頭部カメラ及びソフトハンド情報の視覚情報提示システム

図 6.14: 遠隔操縦のための遠隔ロボット状態・周囲環境情報の視聴覚・振動情報提示システム

6.7 物体操作作業における実世界経験収集システム

本節では物体操作作業においてどのように実世界経験を収集するかについて説明する。実世界経験とは以下の2つが存在する。

- 作業経験

- 日常経験

作業経験とは、何かしらの作業を行っている期間の経験であり、作業経験データにはどのような作業を行っていたかや、作業が成功したかなどの作業情報・作業結果と関連付けて保存することができるため、作業に関する知識の学習やモニタリングに用いることができる。日常経験とは作業経験に対して作業を行っていない期間を含めたロボットが起動している間の全ての経験を表し、日常経験データは作業に関係しない一般的な物体認識などの学習やモニタリング、メンテナンスに用いることができる。以下では作業経験と日常経験それぞれの収集システムについて説明する。

6.7.1 作業経験収集システム

本項では作業経験の収集システムについて説明する。作業実行システムは 6.5.1.1 目で説明した `app_manager` [127] を用いる。このとき作業経験収集は作業実行中のセンサデータや作業結果を収集し作業終了時に自動で保存することで実現し、実装としては 6.5.1.2 目で説明した `app_manager` のプラグインとクラウドデータストレージサービスを用いる。具体的には `app_manager` からアプリケーションとして実行された作業に対して、以下の 4 つの `app_manager` のプラグインを実装することで、`app_manager` から作業をアプリケーションとして実行するだけで自動的に作業経験をクラウドに保存し、そのデータへのアクセス方法をユーザに教えてくれるというシステムを構築する。

- 作業実行中のセンサデータなどを保存するプラグイン
- 作業実行後に作業結果を保存するプラグイン
- 作業実行後に保存したセンサデータと作業結果をクラウドデータストレージにアップロードするプラグイン
- アップロードしたクラウドデータストレージについての情報をユーザに伝えるプラグイン

6.7.1.1 クラウドデータストレージサービスを用いた作業経験収集システム

本目ではクラウドデータストレージサービスを用いた作業経験収集システムについて具体的に説明する。本研究で収集する作業経験データとして、画像・音声を含むセンサデータを動画ファイル、音声ファイル、`ROS Bag` ファイルに保存し、有限状態機械のステート遷移については `JSON` ファイルに保存し、作業結果については `YAML` 形式で以下の 3 種類のうちの一つを保存する。

- 作業成功
- 作業失敗
- 作業停止

作業成功と作業失敗については、各作業アプリケーションの返り値が0の場合には作業成功として、返り値が1の場合には作業失敗として作業結果を保存する。またユーザによる停止命令やタイムアウトによる `app_manager` からのアプリケーション停止が発生した場合に、作業停止として作業結果を保存する。そして最後に、上記のように保存されたファイルを作業終了時に Google Drive というクラウドデータストレージサービスにアップロードし、データへのアクセス URL をユーザにメールで伝えることで作業経験を統一的に管理する。

6.7.2 日常経験収集システム

本項では日常経験をどのように収集するかについて説明する。日常経験は作業経験とは異なり、作業実行時ではなく常時データを収集する必要があるため、データ容量や保存形式が課題となる。本研究では日常経験データの特性に合わせて以下の3つの保存形式にそれぞれ保存する。

- 時系列データ (関節角度・座標)
- 視聴覚データ (画像・音声)
- 関連データ (物体認識結果)

時系列データとは関節角度・速度・トルクや地図上での座標位置などを表し、これらのデータは構造化されておらず容量が少ないものの 100Hz などの高周期で送信されるため、書き込み速度が早く時系列データを圧縮保存できる時系列データベース (Time-series Database) の一つである InfluxDB [129] に保存する。視聴覚データとは画像と音声データであり、これらは 30Hz といった中周期でありながら構造化されたデータであるため、データ容量が大きくても対応できるクラウドデータストレージサービスである Google Drive に保存する。視聴覚データとして画像と音声を同時に収集する理由は、これらのデータの同期をとるためには同じファイルに映像データ (AVI 形式など) として保存することが効果的であるためである。最後に関連データは低周期かつ記号的なためデータ用呂も小さく書き込み速度も遅くて良いため、関連データベース (Relational Database) の一つである MongoDB [130] に保存する。以下ではこれら3つの収集システムについて説明する。

6.7.2.1 InfluxDB を用いた日常時系列経験収集システム

本目では InfluxDB [129] を用いた時系列データの日常経験収集システムについて説明する。本研究では、ロボットから 100Hz 程度の高周期で送信される各関節角度・速度・トルクなどの時系列データをバッファに保存していき、数秒おきに InfluxDB にバッファ内部のデータを書き込むことで実現する [131]。PR2-JSK について保存された InfluxDB を可視化ツール Grafana [132] で可視化した結果を図 6.15 に示す。図 6.15 に示すようにロボットの各関節情報や手先座標情報・ベース座標情報・バッテリー情報・ネットワーク通信速度を日常時系列経験として保存し、可視化ツール Grafana を用いて監視することでロボットの常時モニタリングを実現している。



図 6.15: PR2-JSK の InfluxDB に蓄積された日常時系列経験データの Grafana による可視化

6.7.2.2 クラウドデータストレージサービスを用いた日常視聴覚経験収集システム

本目ではクラウドデータストレージサービスを用いた視聴覚データの日常経験収集システムについて説明する。本研究では、ロボットから 30Hz 程度の中周期で送信される画像と音声データに対して同期を取りながら圧縮しながら動画ファイルとして保存していき、数分おきに動画ファイルを分割してクラウドデータストレージサービスである Google Drive にアップロードすることで実現する。画像と音声データを動画ファイルとして保存することで動画の圧縮技術を用いることができ、常時録画した画像をクラウドデータストレージサービスにアップロードすることでロボット体内 PC のデータ容量に関する制限を解決している。また Google Drive には動画ファイルに対して Web ブラウザ上でのプレビュー機能があるため、全データをダウンロードすることなく Web ブラウザ上で閲覧・検索することができる。

6.7.2.3 MongoDB を用いた日常関係経験収集システム

本目では MongoDB [130] を用いた関連データの日常経験収集システムについて説明する。本研究で用いる関連データの収集手法は図 6.16 に示すように MongoDB [130] のレプリケーション機能を用いた手法 [133] であり、ロボット体内で MongoDB に保存した日常経験データをレプリケーションとして周期的に外部サーバに転送し、最終的にはクラウドデータストレージサービスである Google Drive に保存するというものである。

6.8 物体操作作業のための深層学習モデルによる認識

本節では物体操作作業に必要な認識について以下の 2 つの認識について重点的に説明する。

- 物体操作作業のための物体・環境認識
- 物体操作作業のための人認識

物体操作作業において操作する物体と周りの環境の種類や位置を認識するための物体・環境認

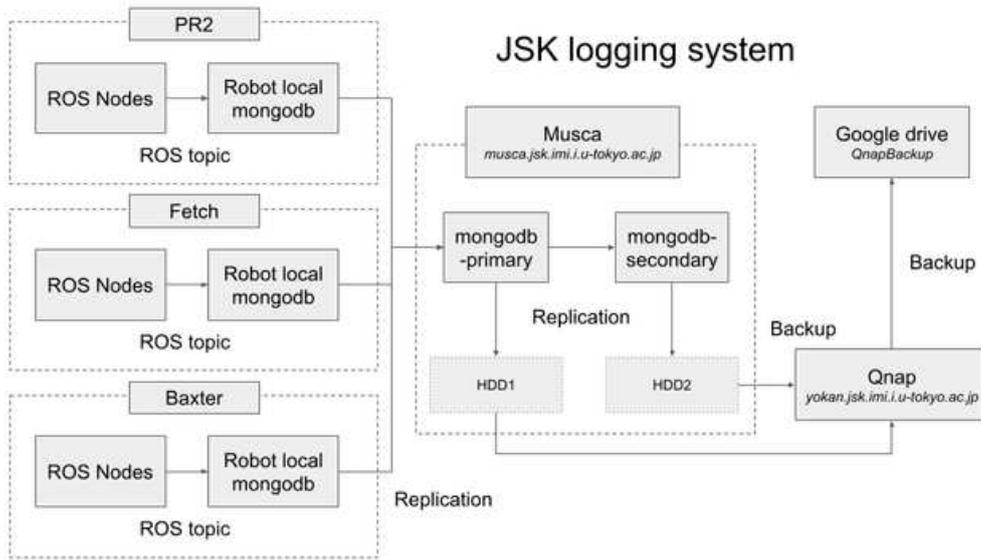


図 6.16: MongoDB を用いた日常関係経験収集システム

識と操縦する人の姿勢や音声を認識する人認識が必要であり，本研究ではこれらの2つの認識について深層学習モデルを用いる．深層学習モデルは人・物体・環境認識において高い精度が示しているが，GPUなどの電力消費の大きい特殊な並列計算器が必要となる．そこで本研究では深層学習モデルの実行について以下の3つの実行手法を採用する．

- ローカルサーバを用いた認識
- エッジデバイスを用いた認識
- クラウドサービスを用いた認識

ローカルサーバとはロボットの外に置かれているが同一ネットワークに置かれたサーバであり，ロボットはこのサーバに対してセンサデータを送信し，サーバ内で深層学習モデルを実行しロボットに認識結果を返信する手法である．この手法は拡張性が高いものの，ロボットの通信帯域を必要とし，中周期で認識をする手法となる．一方エッジデバイスを用いる手法では，ロボットに深層学習実行用のエッジデバイスを搭載することで，ロボット体内で高速かつ低電力で深層学習モデルを実行する手法である．この手法は高周期かつロボットと外部との通信を必要としないが，拡張性が低く深層学習モデルを逐次的に学習することは出来ず，また実行できる深層学習モデルの容量や構成にもハードウェア的な制限がある．最後にクラウドサービスを用いる手法では，ロボットはインターネットを介してクラウドサービスにセンサデータを送信し，クラウドサービスが認識結果を返信するという手法である．この手法は，拡張性はあるものの，ロボットの通信帯域とインターネット環境が必要となり，3つの手法の中でも一番低周期な認識手法となる．よってこの3つはそれぞれ利点欠点があるため，本研究では以下のように使い分けることで解決する．

- 人姿勢認識・物体検出: エッジデバイスでの実行
- 音声認識: クラウドサービスでの実行
- 特殊なモデル・高性能なモデル: ローカルサーバでの実行

以下ではこれらの認識について順に説明する。

6.8.1 物体操作作業のための物体・環境認識

本項では深層学習モデルを用いた物体操作作業のための物体・環境認識について説明する。本研究ではローカルサーバを用いた実行とエッジデバイスを用いた実行の2つを使い分ける。

6.8.1.1 ローカルサーバでの深層学習モデルによる物体・環境認識

本目ではローカルサーバでの GPU を用いた深層学習モデルによる物体・環境認識について説明する。ローカルサーバでの GPU を用いた深層学習モデルの実行は拡張性があるため 5.5.2 項で説明した FCN [113] や Mask-RCNN [115] を改良した CNN 把持点予測器などの特殊なモデルが実行できる。しかしロボットセンサデータを一度ローカルサーバに送信するため、ロボットの通信帯域がボトルネックとなるため中周期の認識となる。

6.8.1.2 エッジデバイスでの深層学習モデルによる物体・環境認識

本目ではエッジデバイスを用いた深層学習モデルによる物体・環境認識について説明する。本研究では深層学習用のエッジデバイスとして Google 社が開発した TPU [134] を搭載した Coral TPU USB デバイス [135] を用いる。このエッジデバイスは USB 接続かつ USB バスの電力のみで実行できるためロボットへの搭載が容易であるが、実行できる深層学習モデルの構成や容量に制限があり、また逐次学習などエッジデバイス上での学習が行えないため拡張性が低い。そのため本研究では先に学習した物体・環境認識モデル (図 6.17) をエッジデバイス用のモデルに変換し、図 6.18 に示すようにカメラ画像に対してロボット内で物体・環境認識を作業実行に関わらず高周期で常時実行するためのソフトウェアを構築する [136]。

6.8.2 物体操作作業のための人認識

本節では深層学習モデルを用いた物体操作作業のための人認識について説明する。本研究ではローカルサーバを用いた実行、エッジデバイスを用いた実行、クラウドサービスを用いた実行の3つを使い分ける。

6.8.2.1 ローカルサーバでの深層学習モデルによる人インスタンス領域・姿勢認識

本目ではローカルサーバでの GPU を用いた深層学習モデルによる人インスタンス領域・姿勢認識について説明する。ローカルサーバでの GPU を用いた深層学習モデルの実行は拡張性があるため高精度なモデルや特殊なモデルが実行でき、図 6.19 に示すような OpenPose [137] や Mask-RCNN [115] を用いた高精度な人のインスタンス領域認識や姿勢認識を実行すること

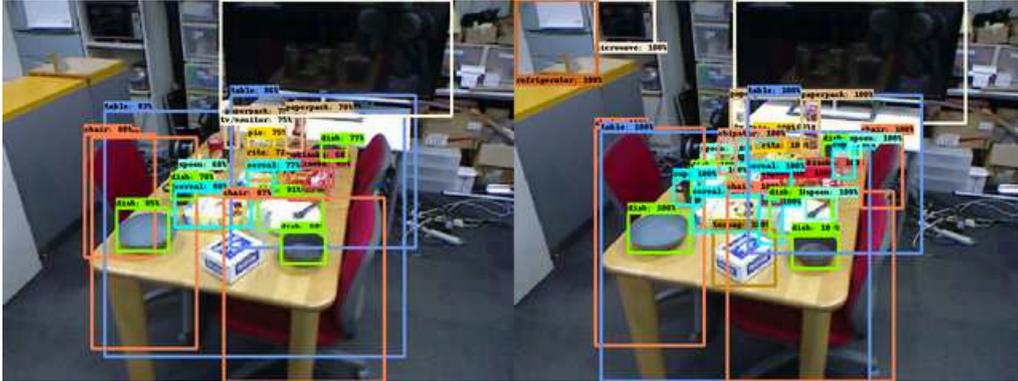


図 6.17: Coral TPU USB エッジデバイス上で実行するための SSD モデルの学習結果 (左: 予測結果, 右: 正解データ)

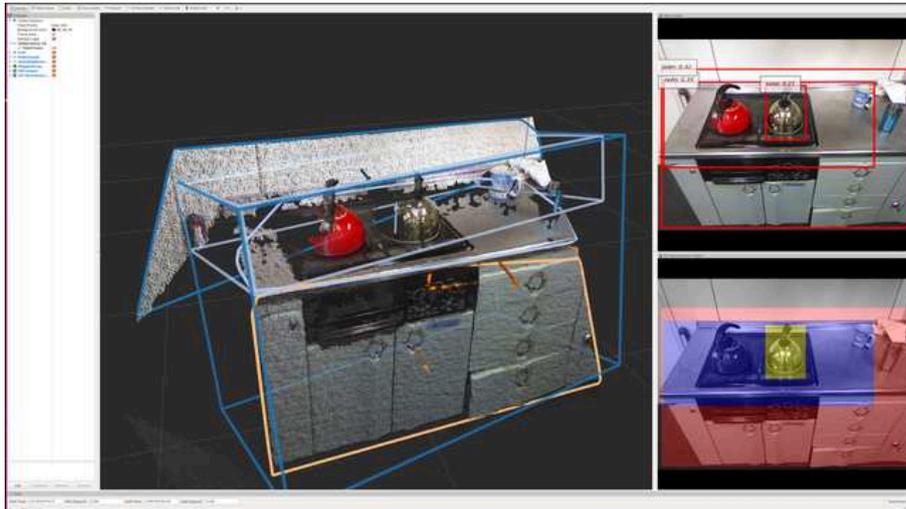


図 6.18: Coral TPU USB エッジデバイス上での SSD によるキッチン環境の認識

ができる。しかしロボットセンサデータを圧縮した後にローカルサーバに送信するため、ロボットの通信帯域がボトルネックとなるため中周期の認識となる。

6.8.2.2 エッジデバイスでの深層学習モデルによる人姿勢認識

本目ではエッジデバイスを用いた深層学習モデルによる人姿勢認識について説明する。本研究では深層学習用のエッジデバイスとして Google 社が開発した TPU [134] を搭載した Coral TPU USB デバイス [135] を用いる。このエッジデバイスは USB 接続かつ USB バスの電力のみで実行できるためロボットへの搭載が容易であるが、実行できる深層学習モデルの構成や容量に制限があり、また逐次学習などエッジデバイス上での学習が行えないため拡張性が低い。そのため本研究では先に学習した人姿勢認識モデルをエッジデバイス用モデルに変換し、図 6.20 に示すようにカメラ画像に対してロボット内で人姿勢認識を作業実行にかかわらず高周

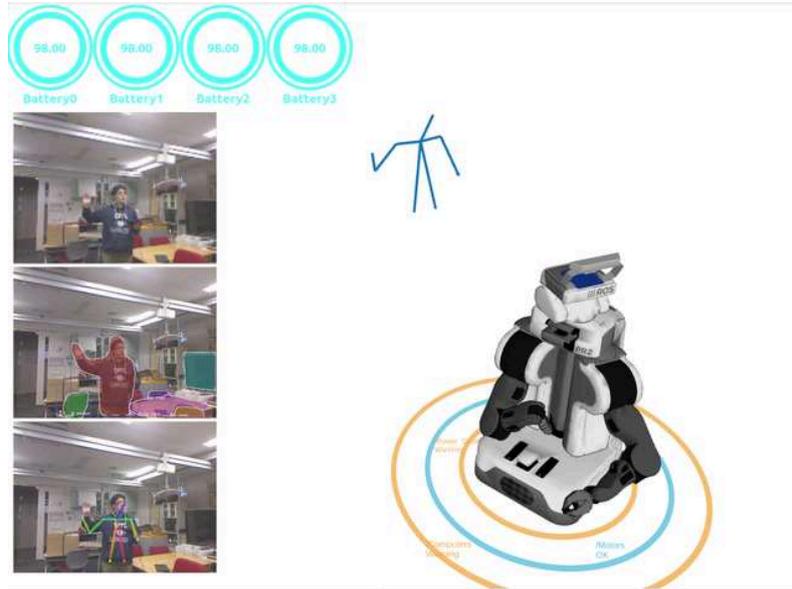


図 6.19: GPU 上での Mask-RCNN と OpenPose による人のインスタンス領域・姿勢認識

期で常時実行する。



図 6.20: Coral TPU USB エッジデバイス上での人の姿勢認識

6.8.2.3 クラウドサービスによる音声認識

本目ではクラウドサービスを用いた深層学習モデルによる音声認識について説明する。本研究ではクラウドサービスとして Google 社のクラウド音声認識サービスを利用し、ロボットのマイクから取得された音声データをインターネットを通じてクラウドサービスに送信し、認識結果を自然言語の文字列として利用する。このクラウドサービスを用いる手法はインターネッ

トの通信が必要であるためインターネット環境のない環境では実行できないが、ロボットはインターネット接続を常にモニタしながら、インターネット接続がない場合にはロボット体内で音声認識ソフトウェア Julius [138] を用いるようにすることで、常に音声認識ができるようなシステムを構築する。この手法はロボットセンサデータをクラウドサービスに送信するため、ロボットのインターネット通信帯域がボトルネックとなるため低周期の認識となるが、ネットワークの状況によるが数秒程度の応答性であり、発話や作業実行の実行時間のほうが長いいため致命的な問題にはならない。

6.9 評価実験

6.9.1 冷蔵庫物品取り出し作業操縦実行実験

本実験では 6.6.1 項で説明した視聴覚統合型遠隔操縦システムの有効性について評価を行うために、6.6.1 項で説明した視聴覚統合型遠隔操縦システムと双腕移動ロボット PR2-JSK を用いて冷蔵庫の中にある缶コーヒーを取り出す作業を遠隔操縦で行った。操縦者はロボットから離れた場所から VR デバイスを用いて、ヘッドマウントディスプレイに表示されるロボットの頭部カメラ画像データを元に操縦を行った。実験の結果は図 6.21 に示すように、本研究で用いた遠隔操縦システムを用いて PR2-JSK を操縦して冷蔵庫の扉をあけて、冷蔵庫内にある缶コーヒーをつかみ、冷蔵庫の外にある机の置くことが出来た。

6.9.2 冷蔵庫物品取り出し作業自律実行実験

本実験では 6.5 節で説明した作業実行管理システムの有効性について評価するために、6.5 節で説明した作業実行管理システムと双腕移動ロボット PR2-JSK を用いて冷蔵庫の中にある缶コーヒーを取り出す作業を音声ベース自律実行で行った。本実験は、人がまずロボットに対して「ねえねえ」とホットワードを話しかけ、それに対してロボットが「なんでしょう?」などと作業依頼受け入れ状態になったときに、人が「ジョージアとってきて」という風に缶コーヒーの銘柄を指定して、冷蔵庫から物品を取り出すよう依頼することで作業実行を行った。実験の様子は図 6.22 に示す。実験の結果は図 6.22 に示すように、ロボットは冷蔵庫を開けた後に冷蔵庫内にあるジョージアの缶コーヒーを色ヒストグラムを用いて認識し、その指定された缶コーヒーを取り出し、扉を閉めて作業を依頼した人に受け渡すことが出来た。また冷蔵庫内に置かれている複数の物品の中から青色の缶コーヒー銘柄であるジョージアを正しく認識し把持し冷蔵庫から取り出すことが出来た。

6.9.3 毎日キッチン巡回作業自律実行実験

本実験では 6.5 節で説明した作業実行管理システムの有効性について示すために、6.5 節で説明した作業実行管理システムと単腕移動ロボット Fetch-JSK を用いてキッチンの巡回をする実験をスケジュールベースの自律実行で行った。本実験は毎朝 10 時にスケジュールされた

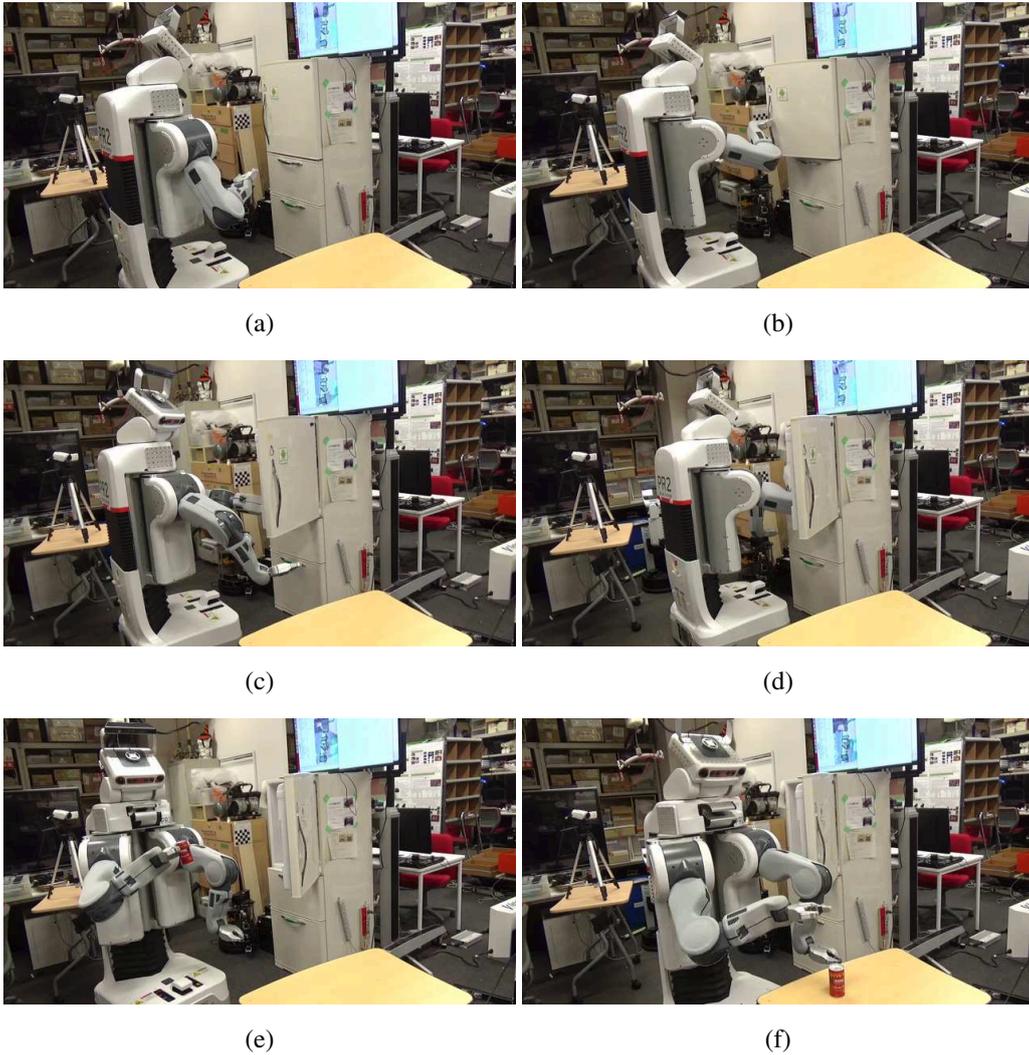


図 6.21: PR2-JSK による冷蔵庫物品取り出し作業操縦実行実験

キッチン巡回作業を実行するものである。実験の様子は図 6.23 に示す。実験の結果は図 6.23 に示すように、ロボットが部屋の明かりをインターネット越しに IoT デバイスを用いて点灯したのちに、充電ベースから自ら発進し、キッチンまで障害物を避けながら向かい、キッチンにてコンロやシンクの様子の写真や映像を撮影し、充電ベースに戻ってくることができた。またこの巡回作業の際の頭部カメラから撮影した映像や物体認識結果のデータは、作業終了後に自動的にクラウドサービスにアップロードされた後に、作業結果をメールでユーザに知らせることが出来た。またこの実験は 2019 年 11 月 1 日から毎日行っており、失敗することもあるが 2021 年 10 月現在まで約 2 年間毎日実験を続けている。

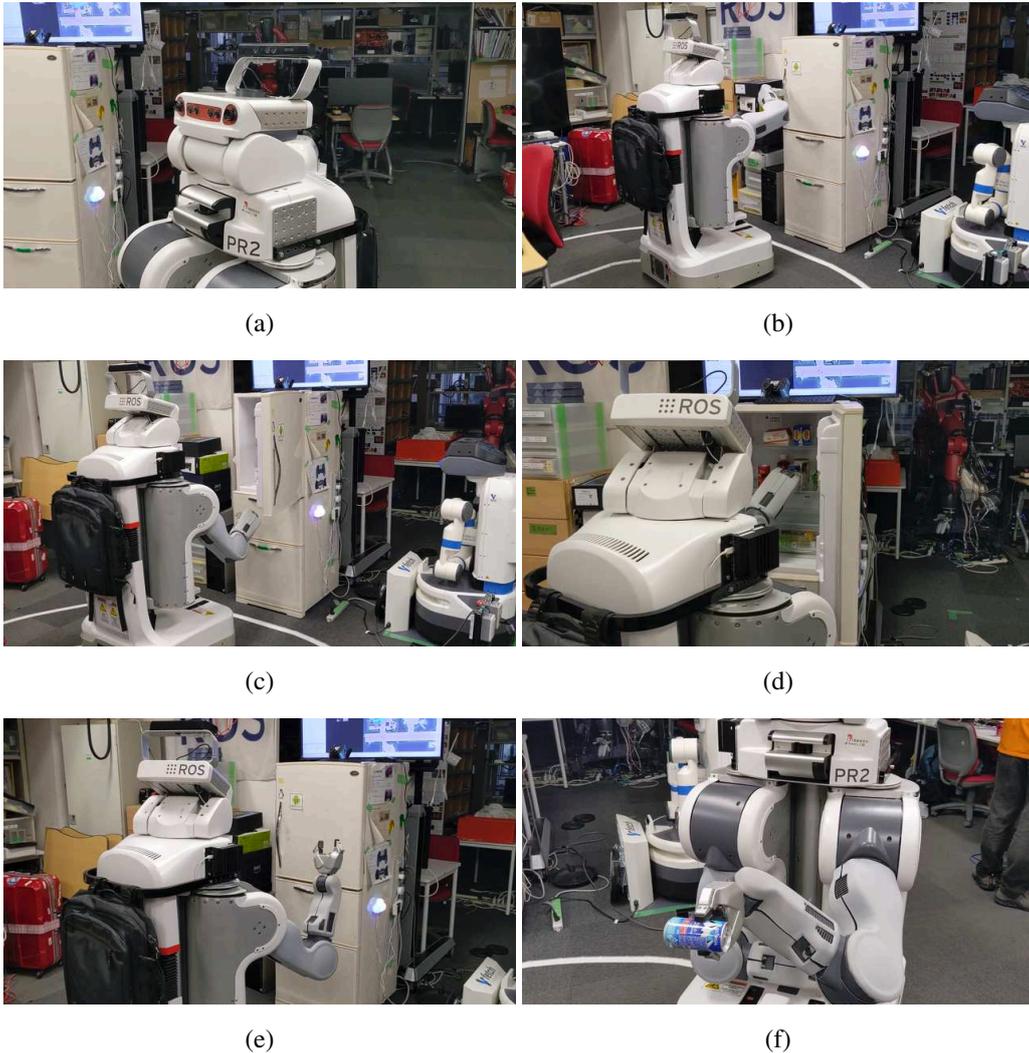


図 6.22: PR2-JSK による冷蔵庫物品取り出し作業自律実行実験

6.10 おわりに

本章では対話的教示が可能な物体操作作業の操縦自律間切替実行システムについて説明し、このシステムを、ユーザの入力に応じた操縦自律間切替実行、作業実行管理システム、視聴覚統合遠隔操縦システム、実世界経験収集システム、人・物体・環境認識の5つの要素で構成した。評価実験として双腕移動ロボット PR2-JSK、単腕移動ロボット Fetch-JSK を用いて物体操作作業の自律実行と操縦実行の実験を行い、操縦自律間切替実行以外の各システムの有効性について確認し、本実行システムの有効性を示した。操縦自律間切替実行については第7章にて評価実験を行う。

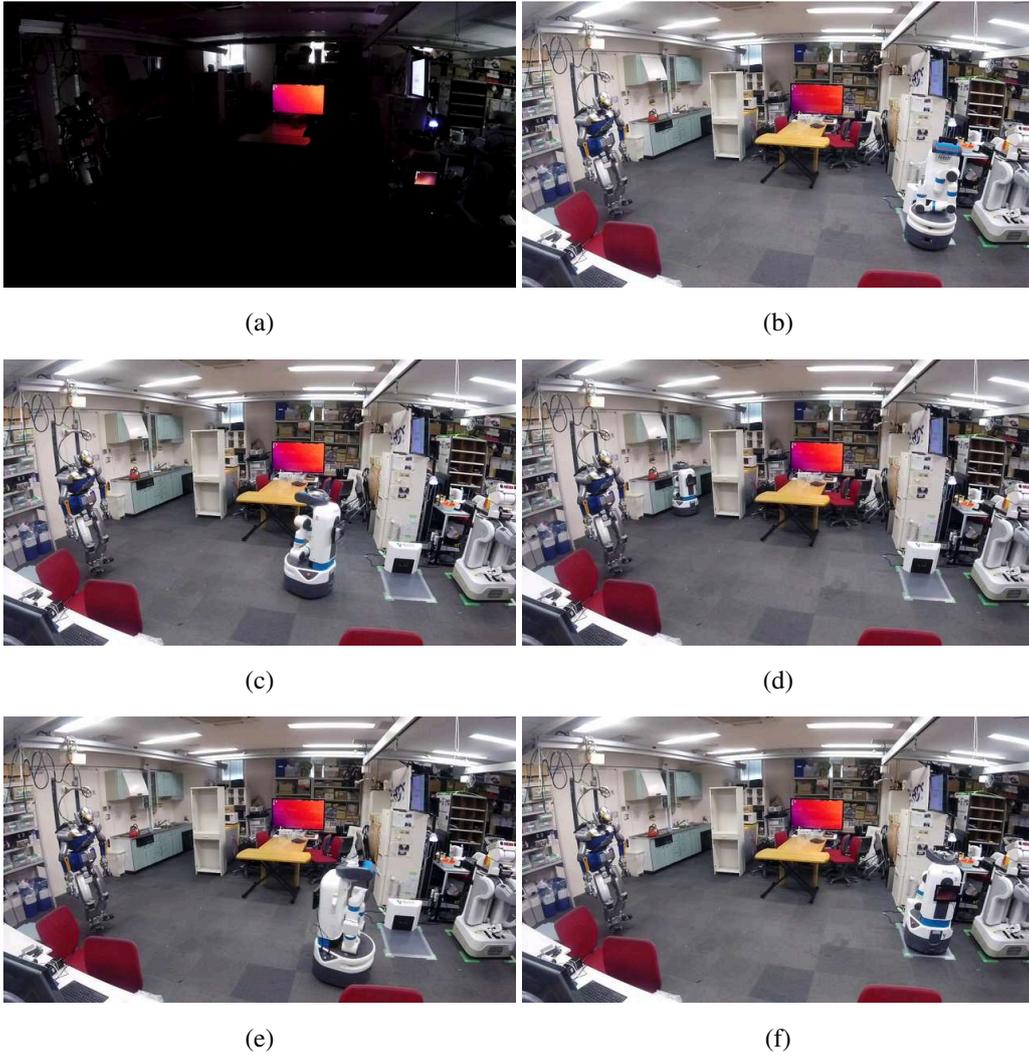


図 6.23: Fetch による毎日キッチン巡回作業自律実行実験

第 7 章

物体操作作業の操縦自律間横断学習 ロボットシステムの評価

7.1 はじめに

本章では、本研究で構成する操縦実行と自律実行を切り替えながら行うことができる物体操作作業の操縦自律間横断学習ロボットシステムについて評価し、評価実験として双腕ロボット Baxter-JSK-SoftHand を用いて物体操作作業の操縦自律間切替実験について説明する。本研究では実験として、箱の回転積み重ね作業の操縦自律間切替実行における作業手順の教示学習実験と自律学習型把持動作を伴う物体ダンボール箱収納作業の操縦自律間切替実行における作業手順の教示学習実験を行った。

1 つめの実験である箱の回転積み重ね作業の操縦自律間切替実行における作業手順の教示学習実験では本研究の提案システムの操縦自律間横断学習機能について評価し、第 4 章で説明したタンジブルインタフェースを用いた操縦実行・教示学習と、第 6 章で説明した操縦自律間切替実行システムを用いた物体操作作業の操縦自律間横断学習を実現し、その有用性を確認した。本実験で取り組んだ箱の回転積み重ね作業は箱を抱え上げて他の箱の上に積み重ねることや箱を Z 軸 90° もしくは Z 軸 180° 回転させることが作業を遂行する際に求められる。本実験で用いる双腕ロボット Baxter-JSK-SoftHand がこの作業を行う際には、箱を双腕で抱え込んで持ち上げる動作列や双腕を用いて箱を両側から挟み込んで 45° 回転させ、一度離して箱を挟み込み直してから更に 45° 回転させる動作列を実行する必要がある。これは本研究で取り組む手順があり接触の多い複雑な物体操作作業であるといえ、失敗や新たな作業に対応するための作業手順を変更する必要があると考え、提案システムの評価実験で行う物体操作作業に適している。本実験では新規の実行という状況に対して作業手順の変更を行うことで提案システムの評価を行った。

2 つめの実験である自律学習型把持動作を伴う物体ダンボール箱収納作業の操縦自律間切替実行における作業手順の教示学習実験では、本研究の提案システムの操縦自律間横断学習機能について評価し、第 4 章で説明したタンジブルインタフェースを用いた操縦実行・教示学習と、第 5 章で説明した学習型動作戦略を用いた自律実行・自律学習、第 6 章で説明した操縦自

律間切替実行システムを用いた本研究で提案した操縦自律間横断学習ロボットシステムと、このシステムによる操縦自律間横断学習を実現し、その有用性を確認した。本実験で取り組んだ自律学習型把持動作を伴う物体ダンボール箱収納作業は物体を把持することや物体をダンボール箱に収納すること、ダンボール箱を抱え上げて他のダンボール箱に積み重ねることが作業を遂行する際に求められる。本実験で用いる双腕ロボット Baxter-JSK-SoftHand がこの作業を行う際には、箱を把持する動作列や双腕を用いて箱を抱え込んで持ち上げる動作列を実行する必要があり、これは本研究で取り組む手順があり接触の多い複雑な物体操作作業であるとして、失敗や新たな作業に対応するための作業手順を変更する必要があると考え、提案システムの評価実験で行う物体操作作業に適している。本実験では新たな作業の実行という状況に対して作業手順の変更を行うことで提案システムの評価を行った。

7.2 箱の回転積み重ね作業の操縦自律間切替実行における作業手順の教示学習実験

本研究で提案する操縦自律間横断学習ロボットシステムの操縦自律間横断学習機能の評価実験として、箱を対象とした回転積み重ね作業の操縦自律間切替実行における作業手順の教示学習実験を行う。本実験では第4章で説明したタンジブルインタフェースを用いた操縦実行・教示学習と、第6章で説明した操縦自律間切替実行システムを用いることで物体操作作業の操縦自律間切替実行を行いながら作業手順の教示学習を行い、本研究で提案する操縦実行自律横断学習を実現する。本実験では第4章で説明したタンジブルインタフェースによる操縦実行と教示学習と第6章で構築した操縦自律間切替実行を用いて操縦自律間横断学習による作業手順の操縦自律間切替実行における作業手順の教示学習を検証することを目的とし、作業手順の教示学習を作業の序盤と途中で行うことで操縦自律間横断学習の有用性について評価した。

本実験については以下の6つの実験を以下に示す順番に行う。

- 7.2.2 項「箱の積み重ね作業の作業手順の教示学習実験」
- 7.2.3 項「箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業の作業手順の教示学習実験」
- 7.2.4 項「作業手順の教示学習結果に基づく箱の積み重ね作業の自律実行実験」
- 7.2.5 項「作業手順の教示学習結果に基づく箱の Z 軸 90° 回転積み重ね作業の自律実行実験」
- 7.2.6 項「箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験」
- 7.2.7 項「作業手順の教示学習結果に基づく箱の Z 軸 180° 回転積み重ね作業の自律実行実験」

本実験はロボットは作業手順を何も学習していない状態から一連の実験を行った。まず Miniature Tangible Cube を用いて箱の積み重ね作業を操縦実行し、ロボットに箱の積み重ね作

業の作業手順を教示学習した(7.2.2項). 次にロボットに箱の積み重ね作業を自律実行させた際に, 作業の序盤にて音声で操縦実行に切り替えて箱をZ軸に90°回転する動作(Z軸90°回転動作)を操縦実行し, この動作を箱の積み重ね動作の前にこの動作を行うZ軸90°回転積み重ね作業への作業手順の変更をロボットに教示学習した(7.2.3項). そして作業手順の変更が正しく教示学習されているかを確かめるために, 教示学習された箱の積み重ね作業とZ軸90°積み重ね作業の自律実行を行った(7.2.4項, 7.2.5項). さらに, ロボットに箱のZ軸90°積み重ね作業の自律実行させた際に, 箱をZ軸90°回転させたあとに音声で操縦実行に切り替えてさらにZ軸90°回転動作を操縦実行し, この動作を箱の積み重ね動作の前にこの動作を行うZ軸180°回転積み重ね作業への作業手順の変更をロボットに教示学習した(7.2.6項). 最後に作業手順の変更が正しく教示学習されているかを確かめるために, 教示学習された箱のZ軸180°回転積み重ね作業を自律実行した(7.2.7項). 以上の一連の6つの実験から, 物体操作作業の操縦自律間横断学習を評価した. なお本実験では操縦者は遠隔ロボットの頭部カメラの画像に基づいて操縦実行を行い, タンジブルインタフェースとして2台のMiniature Tangible Cube Rev.2を用いた. また本実験では対象とする箱として図4.26に示す幅220mm, 奥行き210mm, 高さ130mmのダンボール製の箱を用いた.

7.2.1 箱の回転積み重ね作業で検証するユーザ入力に応じた操縦自律間切替実行

具体的な個々の実験を説明する前に, まず本実験で想定するユーザ入力に応じた操縦自律間切替実行について説明する. 本実験では以下の2つのユーザ入力に応じた操縦自律間切替実行を評価した.

- 作業動作開始前の自律実行から操縦による教示学習への切替
- 作業動作途中の自律実行から操縦による教示学習への切替

まず1つめの作業動作開始前の切替とは, 作業を自律実行する際に作業動作を開始する直前で操縦による教示学習に切り替えることで事前に学習した作業手順とは最初から異なる作業手順を教示学習するものであり, 本実験では7.2.3項「箱の積み重ね作業の自律実行時におけるZ軸90°回転積み重ね作業の作業手順の教示学習実験」にて評価を行った. 想定する状況としては, 操縦者はある作業の自律実行を指示したがロボットの前にある対象物の状態が想定と異なっていた, もしくは作業の目標状態を変更したため新しい作業のための作業手順の変更を教示学習するというものであり, 本実験では箱を積み重ねる作業の自律実行の際に, 作業動作の開始前にユーザが音声で割り込むことで自律実行から操縦実行へ切り替え, 箱を先にZ軸90°回転させたのちに積み重ねるという作業を教示学習する実験を行った.

2つめの作業動作途中の切替とは, 作業を自律実行している途中で操縦による教示学習に切り替えることで事前に学習した作業手順に対して途中で分岐する形で作業手順を教示学習するものであり, 本実験では7.2.6項「箱のZ軸90°回転積み重ね作業の自律実行時におけるZ軸180°回転積み重ね作業の作業手順の教示学習実験」にて評価を行った. 想定する状況として

は、操縦者はある作業を自律実行しているが途中で作業のロボットの前にある対象物の状態が想定と異なっていた、もしくは作業の目標状態を変更したため作業の途中で新たな作業のための作業手順の変更を教示学習するというものであり、本実験では箱を Z 軸 90° 回転させたのちに積み重ねる作業の自律実行の際に、箱を Z 軸 90° 回転させたところでユーザが音声で割り込むことで自律実行から操縦実行へ切り替え、箱を更に Z 軸 90° 回転させることで箱を合計 Z 軸 180° 回転させたのちに積み重ねるといった作業を教示学習する実験を行った。

7.2.2 箱の積み重ね作業の作業手順の教示学習実験

本項では箱の積み重ね作業の教示学習実験について説明する。本実験ではロボットは作業手順について何も学習していない状態から実験を開始し、Miniature Tangible Cube を用いて箱の積み重ね作業の操縦実行を行い、その作業手順を教示学習した。本実験の教示学習前の作業手順を表す動作順序グラフは図 7.1 に示すように、本実験の教示学習前はロボットは作業手順を何も学習しておらず、実験前の対象物の状態のみが与えられていた。また箱の積み重ね作業の操縦実行のための有限状態機械は図 7.2 に示すように、作業開始すると有限状態機械は作業の初期化ノードである (:init) と (statenet-action-init) に遷移したのちに、教示学習ノードである (:teach) に遷移することで操縦実行および教示学習を実現した。

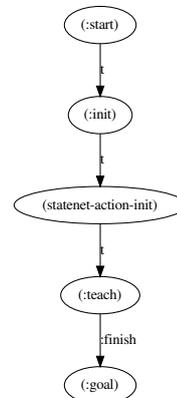
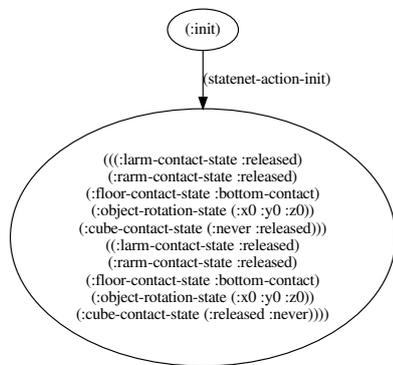


図 7.1: 箱の積み重ね作業の教示学習実験における学習前の動作順序グラフ

図 7.2: 箱の積み重ね作業の教示学習のための有限状態機械

本研究で行った箱の積み重ね作業の教示学習実験については、ロボット側の様子を図 7.3、操縦者側の様子を図 7.4、操縦者側表示画面を図 7.5 に示す。図 7.3、図 7.4、図 7.5 に示すように、まず操縦者が Miniature Tangible Cube を用いた操縦実行することを 6.5.2.2 目で説明した Web GUI を通じて指示した。そして操縦者が手前に置かれた Miniature Tangible Cube に対して左側面と右側面に接触することで (図 7.4b)、ロボットは手前に置かれた箱を両腕で左側面と右側面に対して接触することで箱を抱え (図 7.3b、図 7.5b)、操縦者が抱え込んだ手前の

Miniature Tangible Cube を持ち上げて奥に置かれた Miniature Tangible Cube の上に置くことで (図 7.4c), ロボットは手前に置かれた箱を左右から抱え込んで持ち上げて奥にある箱の上に積み重ね (図 7.3c, 図 7.5c), 最後に操縦者が積み重ねられた Miniature Tangible Cube に対する接触から両腕を離すことで (図 7.4d), ロボットは積み重ねられた箱に対する接触を離れた (図 7.3d, 図 7.5d).

以上の一連の動作順序の操縦実行によって, ロボットが手前の箱を持ち上げて奥の箱に積み重ねる箱の積み重ね作業の作業手順を教示学習することができた.

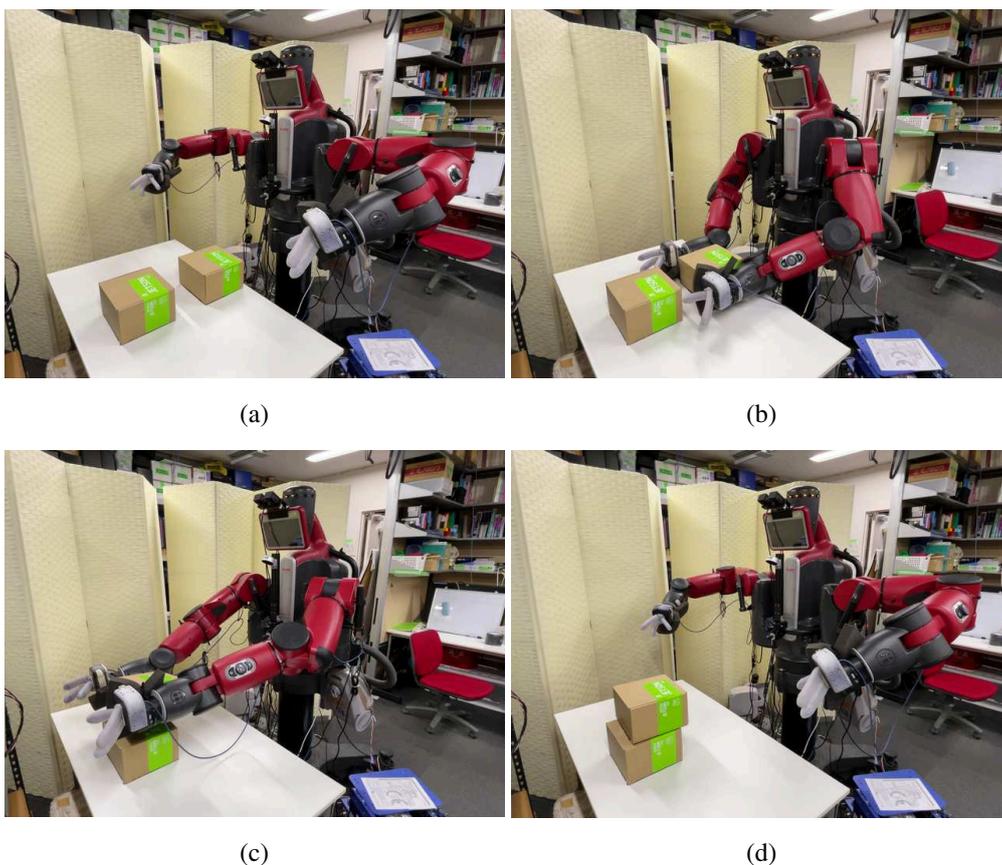


図 7.3: 箱の積み重ね作業の教示学習実験 (ロボット側)

本実験により教示学習された作業手順の動作順序グラフを図 7.6 に示す. 本実験で教示学習された作業手順は図 7.6 の動作順序グラフのエッジである以下の 4 つの動作の順序となる.

1. (dualarm-left-right-push-hold-action): 手前の箱 (対象物 0) の左側面に対して左手で, 右側面に対して右手で接触する
2. (dualarm-push-hold-up-action): 手前の箱 (対象物 0) について両腕抱え込んだまま持ち上げる
3. (dualarm-push-hold-put-up-action :cube-index 1): 手前の箱 (対象物 0) を奥の箱 (対象物 1) の上に置く

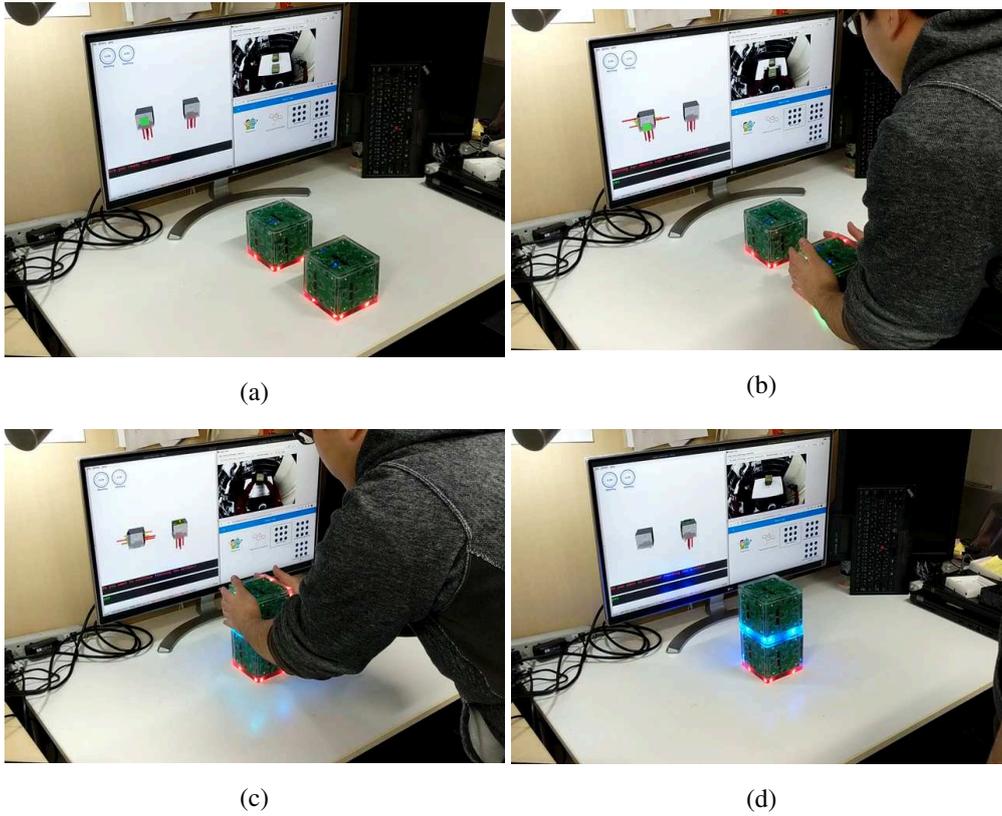


図 7.4: 箱の積み重ね作業の教示学習実験 (操縦者側)

4. (dualarm-hold-release-action): 上に置かれた箱 (対象物 0) の左手と右手による接触を離す

以上 4 つの動作順序より本実験の操縦実行で行ったように、まず箱を両腕で左右側面を挟むことで抱え上げ、これを奥の箱の上に置き、そして両腕の接触を離すという、箱の積み重ね作業の一連の作業手順を教示学習することができた。

7.2.3 箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業の作業手順の教示学習実験

本項では箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業の作業手順の教示学習実験について説明する。本実験は 7.2.2 項の箱の積み重ね作業の教示学習実験を行ったあとに実験を行う。操縦者は箱の積み重ね作業の自律実行を指示するが、ロボットが箱への操作動作を行う前に操縦者が音声を通じて操縦実行への切替を指示する。そしてロボットは操縦実行に切り替えたのちに、操縦者は Miniature Tangible Cube を用いて箱の Z 軸 90° 回転積み重ね作業の操縦実行を行い、その作業手順を教示学習する。本実験の教示学習前の作業手順を表す動作順序グラフは 7.2.2 項で学習された動作順序グラフであり図 7.6 に示す。本実験の

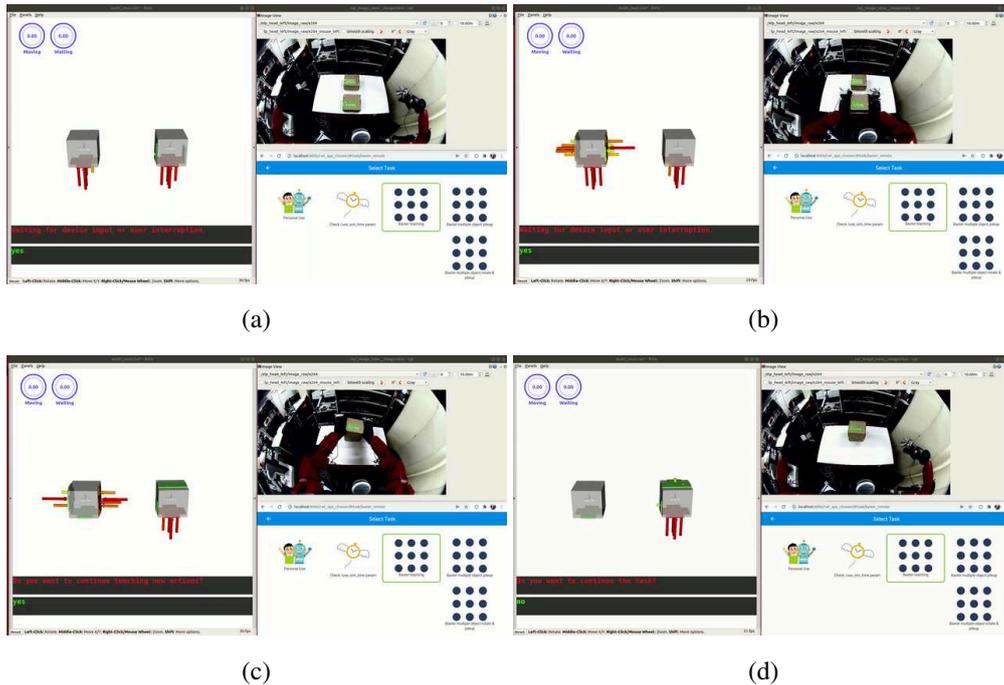


図 7.5: 箱の積み重ね作業の教示学習実験 (操縦者側表示画面)

教示学習前の作業手順を表す動作順序グラフは図 7.6 に示すように箱の積み重ね作業の作業手順のみを学習しており、箱を Z 軸 90° 回転させるための動作順序は学習していなかった。また本実験で用いる箱の積み重ね作業の自律実行のための有限状態機械は図 7.7 に示すように、対象物の状態を表すノード（対象物状態ノード）は動作ノードを 1 つ介して遷移することができ、作業の初期の対象物状態ノードから作業の目標の対象物状態ノードへの経路を実行することで作業の自律実行を実現した。またこの有限状態機械は操縦者による操縦実行への切替に対応するために、6.4 節で説明したように全ての対象物状態ノードは教示学習ノードである (:teach) に遷移することができ、操縦者の音声入力に基づいて教示学習ノードに遷移することで、自律実行から操縦による教示学習への切替を実現した。

本研究で行った箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業の作業手順の教示学習実験については、ロボット側の様子を図 7.8, 操縦者側の様子を図 7.9, 操縦者側表示画面を図 7.10 に示す。図 7.8, 図 7.9, 図 7.10 に示すように、まず操縦者が 7.2.2 項で教示学習された箱の積み重ね作業を自律実行することを 6.5.2.2 目で説明した Web GUI を通じて指示した。そこでロボットは 7.2.2 項で教示学習された箱の積み重ね作業を自律実行しようとするが、ロボットは作業動作を実行する前に操縦者は音声を通じて操縦実行に切り替えるように指示することで、Miniature Tangible Cube を用いた操縦実行に移行した。そして操縦者が手前に置かれた Miniature Tangible Cube に対して両側面上方に接触することで (図 7.9b), ロボットは手前に置かれた箱を両腕で左側面上方と右側面上方に対して接触し (図 7.8b, 図 7.10b), 操縦者が手前に置かれた Miniature Tangible Cube を左側面上方と右側面上方を持ちな

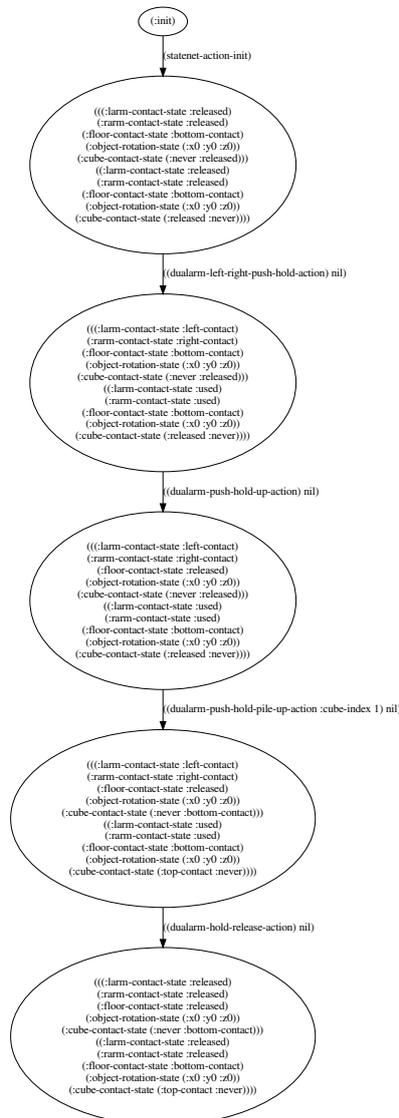


図 7.6: 箱の積み重ね作業の教示学習実験で学習された動作順序グラフ

から Z 軸 45° 回転させることで (図 7.9c), ロボットは手前に置かれた箱を両腕で抱えながら Z 軸 45° 回転させた (図 7.8c, 図 7.10c). さらに操縦者が手前に置かれた Miniature Tangible Cube への両腕の接触を一度離してから, 同じ Miniature Tangible Cube に対して接触していた側面とは異なる側面である左前側面上方と右後側面上方に両腕で接触することで (図 7.9d), ロボットは手前に置かれた箱を一度両腕の接触を離れた後に左前側面上方と右後側面上方に両腕で抱え直し (図 7.8d, 図 7.10d), 操縦者が手前に置かれた Miniature Tangible Cube を左前側面上方と右後側面上方を持ちながら Z 軸 45° 回転させることで (図 7.9e), ロボットは手前に置かれた箱を両腕で抱えながらさらに Z 軸 45° 回転させた (図 7.8e, 図 7.10e).

最後は 7.2.2 項で説明した操縦実行と同様に, 操縦者が手前に置かれた Miniature Tangible

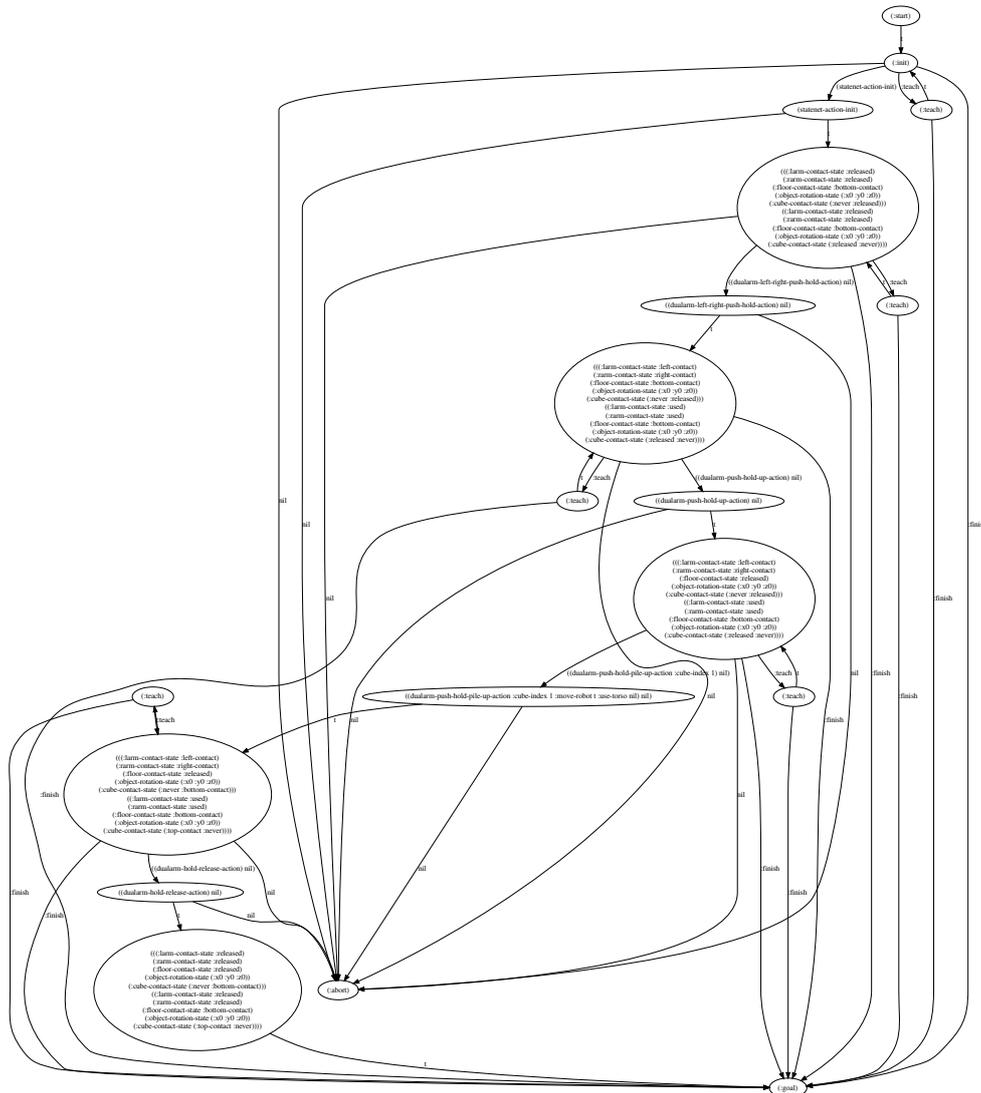


図 7.7: 箱の積み重ね作業の自律実行のための有限状態機械

Cube への両腕の接触を一度離して、同じ Miniature Tangible Cube に対して左側面と右側面に接触することで (図 7.9f), ロボットは手前に置かれた箱を両腕で左側面と右側面に対して接触することで箱を抱え (図 7.8f, 図 7.10f), 操縦者が抱え込んだ手前の Miniature Tangible Cube を持ち上げて奥に置かれた Miniature Tangible Cube の上に置くことで (図 7.9g), ロボットは手前に置かれた箱を左右から抱え込んで持ち上げて奥にある箱の上に積み重ね (図 7.8g, 図 7.10g), 操縦者が積み重ねられた Miniature Tangible Cube に対する接触から両腕を離すことで (図 7.9h), ロボットは積み重ねられた箱に対する接触を離れた (図 7.8h, 図 7.10h).

以上の一連の動作順序の操縦実行によって、ロボットが手前の箱を持ち替えながら Z 軸 90° 回転した後に持ち上げて奥の箱に積み重ねる箱の Z 軸 90° 回転積み重ね作業の作業手順の教示学習を自律実行から操縦実行に作業の動作前に切り替えて行うことができた。

本実験により教示学習された作業手順を図 7.11 に示す。本実験で教示学習された作業手順は図 7.11 の動作順序グラフの青色に示す左側経路上にあるエッジである以下の 10 個の動作順序となる。

1. (dualarm-left-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左側面上方に対して左手で、右側面上方に対して右手で接触する
2. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
3. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
4. (dualarm-front-left-back-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左前側面上方に対して左手で、右後側面に対して右手で接触する
5. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
6. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
7. (dualarm-left-right-push-hold-action): 手前の箱 (対象物 0) の右側面に対して右手で接触、左側面に対して左手で接触する
8. (dualarm-push-hold-up-action): 手前の箱 (対象物 0) について両腕抱え込んだまま持ち上げる
9. (dualarm-push-hold-put-up-action :cube-index 1): 手前の箱 (対象物 0) を奥の箱 (対象物 1) の上に置く
10. (dualarm-hold-release-action): 上に置かれた箱 (対象物 0) の左手と右手による接触を離す

以上の 10 個の動作順序より本実験の操縦実行で行ったように、まず箱を両腕で左右側面上方をはさみながら Z 軸 45° 回転し、持ち替えて更に Z 軸 45° 回転させたのちに、箱を両腕で左右側面を挟むことで抱え上げ、これを奥の箱の上に置き、そして両腕の接触を離すという、箱の Z 軸 90° 回転積み重ね作業の一連の作業手順を教示学習することができた。なお教示学習された動作順序グラフは 7.2.2 項で教示学習された動作順序グラフ (図 7.6) に対して、に対して、作業の初期対象物状態から分岐する形で 1 つの動作順序グラフとして教示学習することができた。

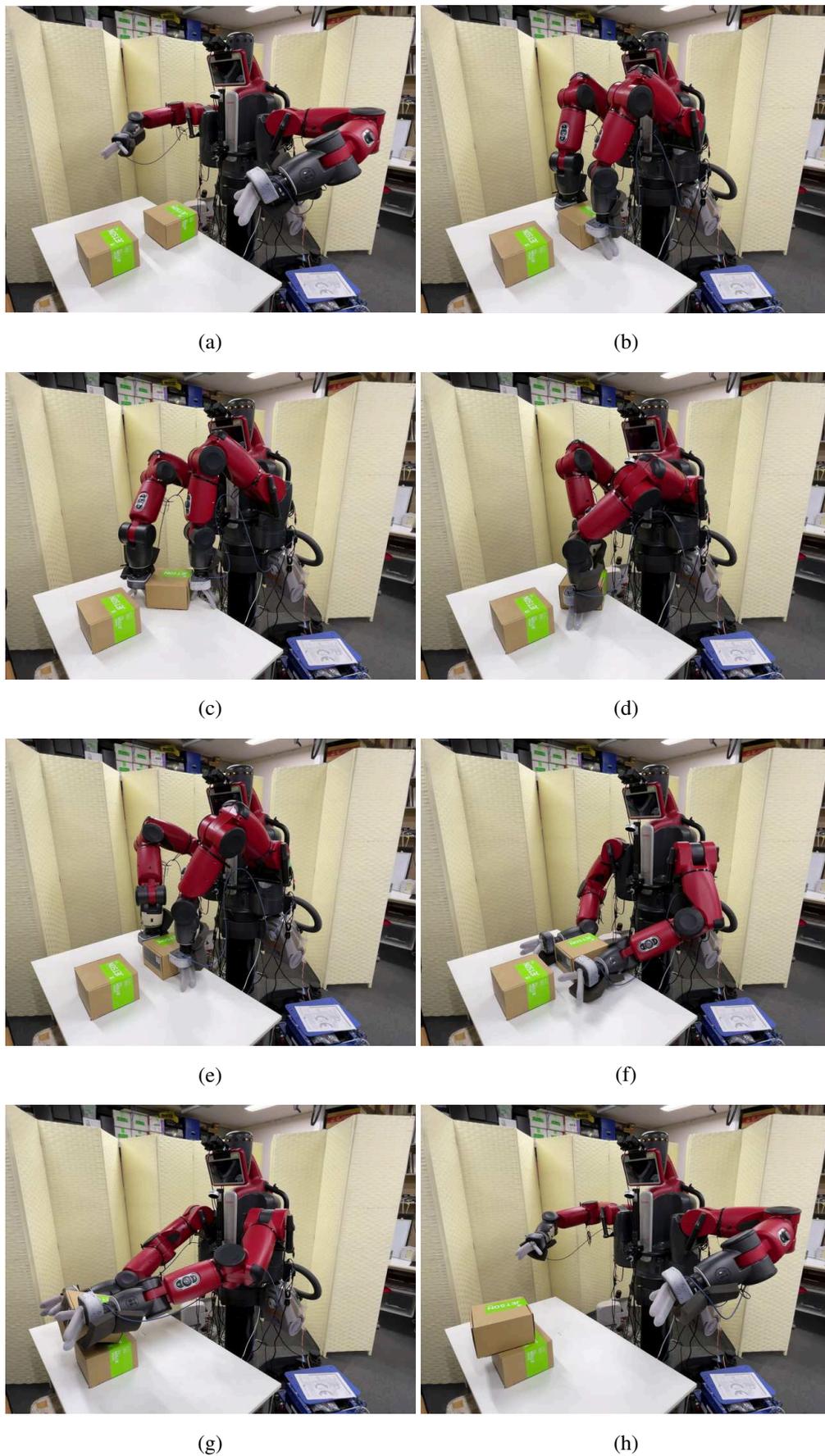


図 7.8: 箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業の作業手順の教示学習実験 (ロボット側)

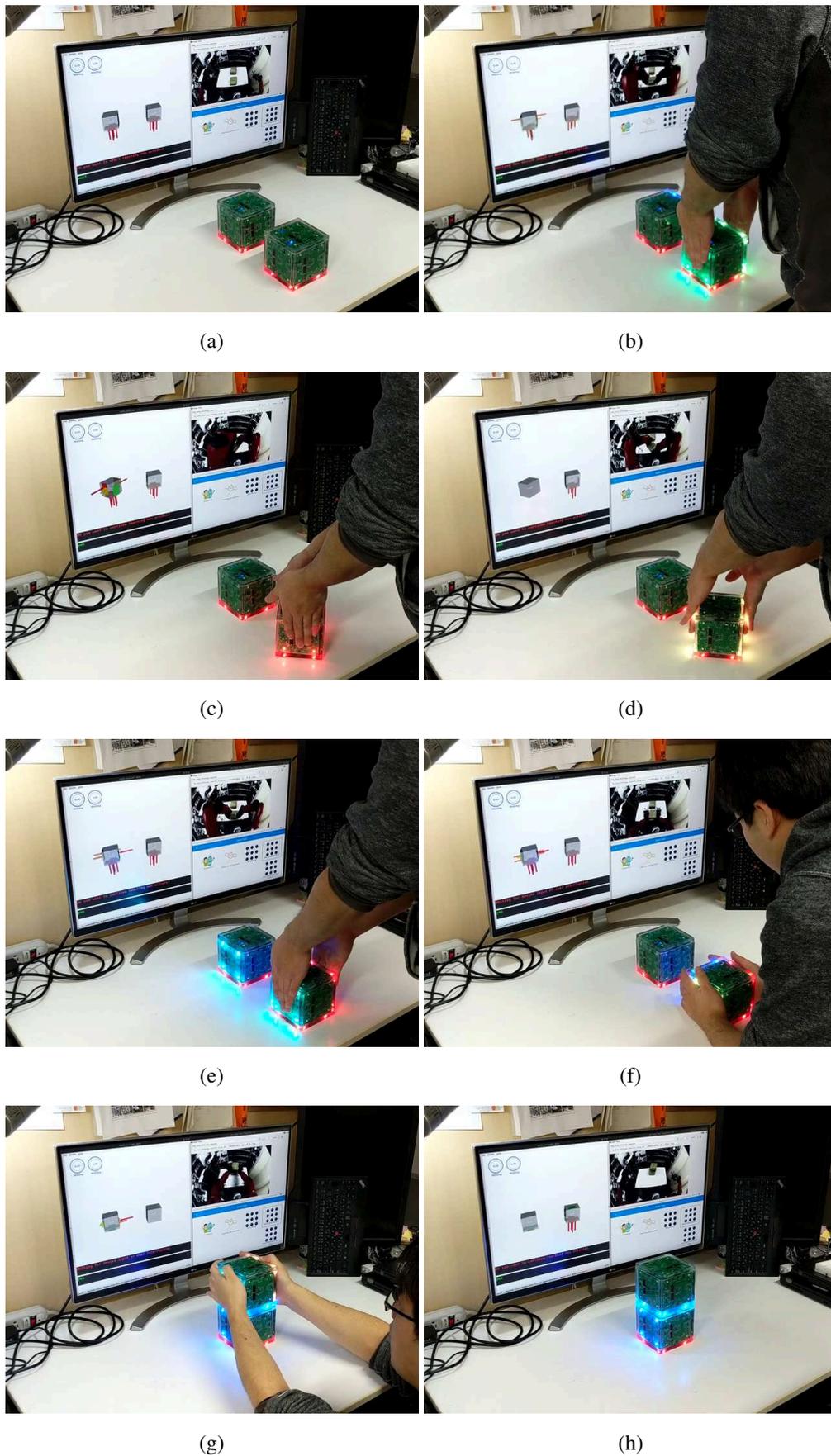


図 7.9: 箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業の作業手順の教示学習実験 (操縦者側)



図 7.10: 箱の積み重ね作業の自律実行時における Z 軸 90° 回転積み重ね作業の作業手順の教示学習実験 (操縦者側表示画面)

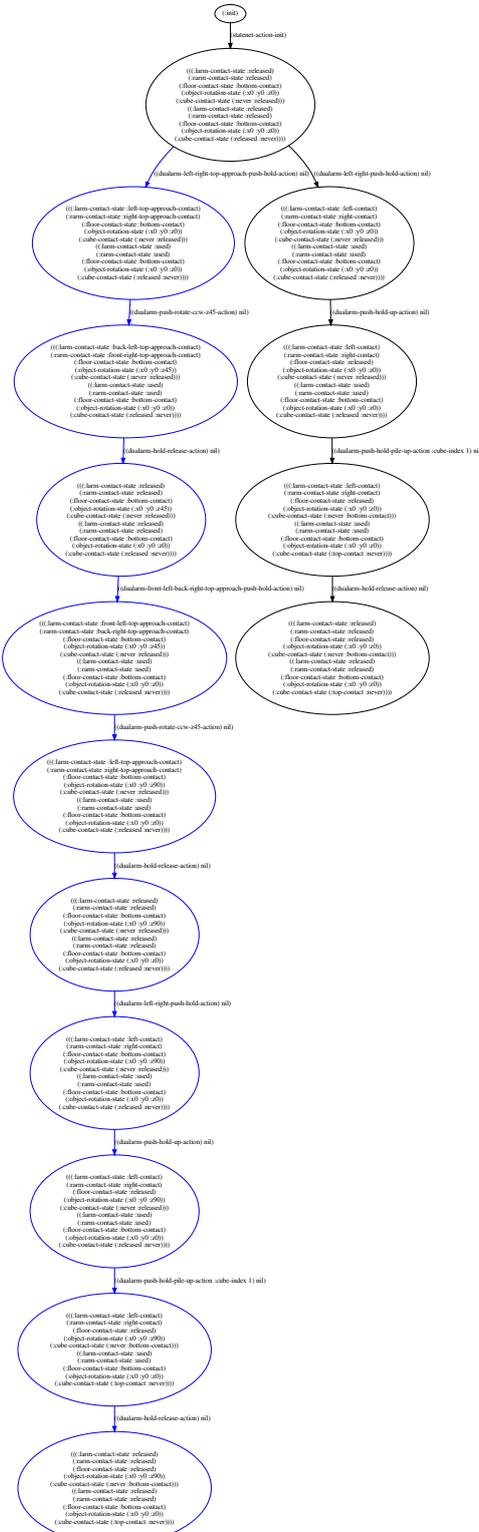


図 7.11: 箱の積み重ね作業の自律実行時における Z 軸 90° 回転作業の作業手順の教示学習実験で学習された動作順序グラフ: 本実験によって図 7.6 に示す動作順序グラフに対して Z 軸 90° 回転作業の作業手順として青色の動作順序を新たな動作順序として教示学習した。

7.2.4 作業手順の教示学習結果に基づく箱の積み重ね作業の自律実行実験

本項では作業手順の教示学習結果に基づく箱の積み重ね作業の自律実行実験について説明する。本実験は7.2.3項の箱の積み重ね作業の自律実行時におけるZ軸90°回転積み重ね作業の作業手順の教示学習実験を行ったあとに実験を行い、操縦者は箱の積み重ね作業を指示しロボットは箱の積み重ね作業の自律実行を行った。本実験で用いる作業手順を表す動作順序グラフは7.2.3項で学習された動作順序グラフであり図7.11に示す。本実験で用いる動作順序グラフは図7.11に示すように、箱の積み重ね作業と箱のZ軸90°回転積み重ね作業の作業手順を学習しており、この学習された動作順序グラフ上で初期の対象物状態から目標の対象物状態への経路を探索することで、箱の積み重ね作業の自律実行を実現した。また本実験で用いる箱の積み重ね作業の自律実行のための有限状態機械は7.2.3項で用いた有限状態機械と同じであり、図7.7に示す。この有限状態機械は操縦者の入力に応じた操縦自律間の切替に対応するために、6.4節で説明したように全ての対象物状態ノードは教示学習ノードである (:teach) に遷移することができるようになっている。

本研究で行った教示学習結果に基づく箱の積み重ね作業の自律実行実験については図7.12に示す。図7.12に示すように、まず操縦者が7.2.2項で教示学習された箱の積み重ね作業を自律実行することを6.5.2.2目で説明したWeb GUIを通じて指示し、ロボットが手前に置かれた箱を両腕で左側面と右側面に対して接触することで箱を抱え(図7.12b)、そしてロボットが手前に置かれた箱を左右から抱え込んで持ち上げて奥にある箱の上に積み重ね(図7.12c)、最後にロボットが積み重ねられた箱に対する接触を離れた(図7.12d)。以上よりロボットが手前の箱を持ち上げて奥の箱に積み重ねる箱の積み重ね作業を自律実行することができた。

本実験で自律実行された動作は以下の4つの動作順序であり、これは7.2.2項で教示学習された動作順序と同じものであり、図7.11の動作順序グラフの黒色に示す経路上にあるエッジである。

1. (dualarm-left-right-push-hold-action): 手前の箱(対象物0)の左側面に対して左手で、右側面に対して右手で接触する
2. (dualarm-push-hold-up-action): 手前の箱(対象物0)について両腕抱え込んだまま持ち上げる
3. (dualarm-push-hold-pile-up-action :cube-index 1): 手前の箱(対象物0)を奥の箱(対象物1)の上に置く
4. (dualarm-hold-release-action): 上に置かれた箱(対象物0)の左手と右手による接触を離す

以上の4つの動作順序より、本実験では7.2.2項で教示学習されたように、まず箱を両腕で左右側面を挟むことで抱え上げ、これを奥の箱の上に置き、そして両腕の接触を離すという、7.2.2項にて教示学習された箱の積み重ね作業の一連の作業手順を自律実行することができた。

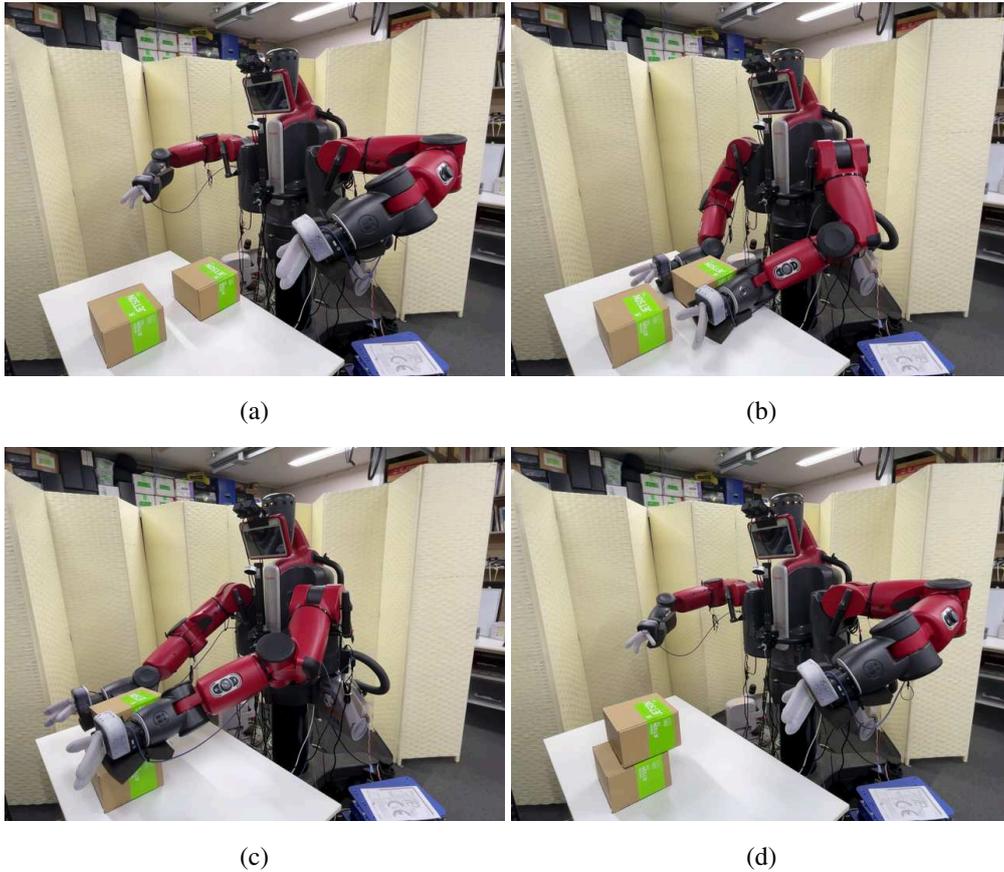


図 7.12: 教示学習結果に基づく箱の積み重ね作業の自律実行実験

7.2.5 作業手順の教示学習結果に基づく箱の Z 軸 90° 回転積み重ね作業の自律実行実験

本項では作業手順の教示学習結果に基づく箱の Z 軸 90° 回転積み重ね作業の自律実行実験について説明する。本実験は 7.2.4 項の教示学習結果に基づく箱の積み重ね作業の自律実行実験を行ったあとに実験を行う。操縦者は箱の Z 軸 90° 回転積み重ね作業を指示し、ロボットは Z 軸 90° 回転箱の積み重ね作業の自律実行を行った。本実験で用いる作業手順を表す動作順序グラフは 7.2.3 項で学習された動作順序グラフであり図 7.11 に示す。本実験で用いる動作順序グラフは図 7.11 に示すように、箱の積み重ね作業と箱の Z 軸 90° 回転積み重ね作業の作業手順を学習しており、この学習された動作順序グラフ上で初期の対象物状態から目標の対象物状態への経路を探索することで、箱の Z 軸 90° 回転積み重ね作業の自律実行を実現した。また本実験で用いる箱の Z 軸 90° 回転積み重ね作業の自律実行のための有限状態機械は図 7.13 に示す。図 7.13 に示す有限状態機械のノードとエッジの色は図 7.11 と対応しており、この有限状態機械は図 7.11 に示す青色の左側経路の動作順序を自律実行でき、かつユーザ入力に応じ

て操縦に切り替えて作業手順の教示学習を行うことができる有限状態機械となっている。またこの有限状態機械は操縦者の入力に応じた操縦自律間の切替に対応するために、6.4節で説明したように全ての対象物状態ノードは教示学習ノードである (:teach) に遷移することができるようになっている。

本研究で行った教示学習結果に基づく箱の Z 軸 90° 回転積み重ね作業の自律実行実験については図 7.14 に示す。図 7.14 に示すように、まず操縦者が 7.2.3 項で教示学習された箱の Z 軸 90° 回転積み重ね作業を自律実行することを 6.5.2.2 目で説明した Web GUI を通じて指示し、ロボットが手前に置かれた箱を両腕で左側面上方と右側面上方に対して接触し、ロボットが手前に置かれた箱を両腕で抱えながら Z 軸 45° 回転させた (図 7.14b)。さらにロボットが手前に置かれた箱を一度両腕の接触を離れた後に左前側面上方と右後側面上方に両腕で抱え直し、手前に置かれた箱を両腕で抱えながらさらに Z 軸 45° 回転させた。最後はロボットが手前に置かれた箱を両腕で左側面と右側面に対して接触することで箱を抱え (図 7.14c)、手前に置かれた箱を左右から抱え込んで持ち上げて奥にある箱の上に積み重ね、積み重ねられた箱に対する接触を離れた (図 7.14d)。以上よりロボットが手前の箱を持ち替えながら Z 軸 90° 回転した後に持ち上げて奥の箱に積み重ねる箱の Z 軸 90° 回転積み重ね作業を自律実行することができた。

本実験で自律実行された動作は以下の 10 個の動作順序であり、これは 7.2.3 項で教示学習された動作順序と同じものであり、図 7.11 の動作順序グラフの青色に示す経路上にあるエッジである。

1. (dualarm-left-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左側面上方に対して左手で、右側面上方に対して右手で接触する
2. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
3. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
4. (dualarm-front-left-back-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左前側面上方に対して左手で、右後側面に対して右手で接触する
5. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
6. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
7. (dualarm-left-right-push-hold-action): 手前の箱 (対象物 0) の右側面に対して右手で接触、左側面に対して左手で接触する
8. (dualarm-push-hold-up-action): 手前の箱 (対象物 0) について両腕抱え込んだまま持ち上げる
9. (dualarm-push-hold-pile-up-action :cube-index 1): 手前の箱 (対象物 0) を奥の箱 (対象物 1) の上に置く

10. (dualarm-hold-release-action): 上に置かれた箱 (対象物 0) の左手と右手による接触を離す

以上の 10 個の動作順序より，本実験では 7.2.3 項で教示学習されたように，まず箱を両腕で左右側面上方をはさみながら Z 軸 45° 回転し，持ち替えて更に Z 軸 45° 回転させたのちに，箱を両腕で左右側面を挟むことで抱え上げ，これを奥の箱の上に置き，そして両腕の接触を離すという，7.2.3 項にて教示学習された箱の Z 軸 90° 回転積み重ね作業の作業手順を自律実行することができた。

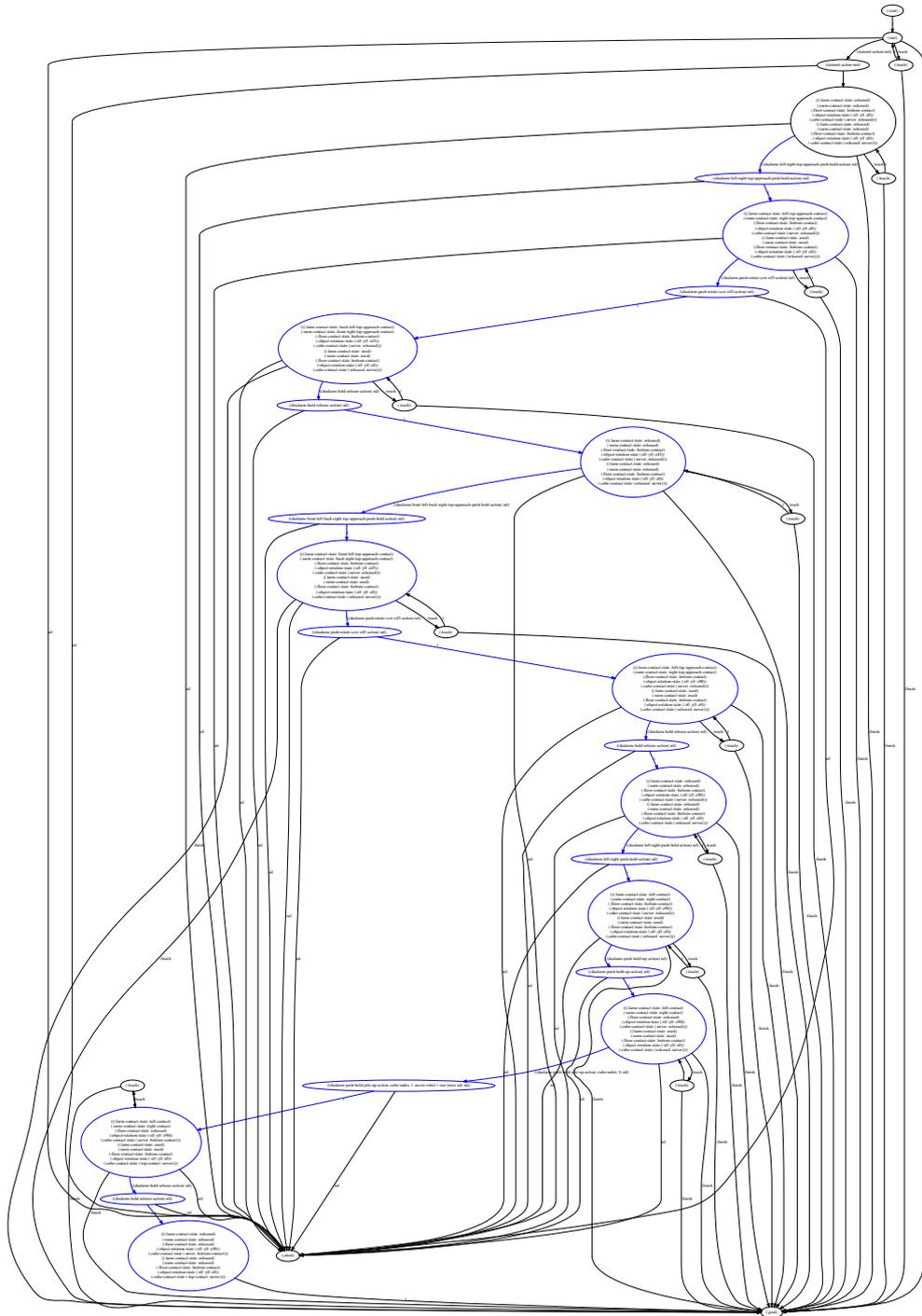


図 7.13: 箱の Z 軸 90° 回転積み重ね作業の自律実行のための有限状態機械: ノードとエッジの色は図 7.11 と対応しており, この有限状態機械は図 7.11 に示す青色の左側経路の動作順序を自律実行でき, かつユーザ入力に応じて操縦に切り替えて作業手順の教示学習を行うことができる有限状態機械となっている.

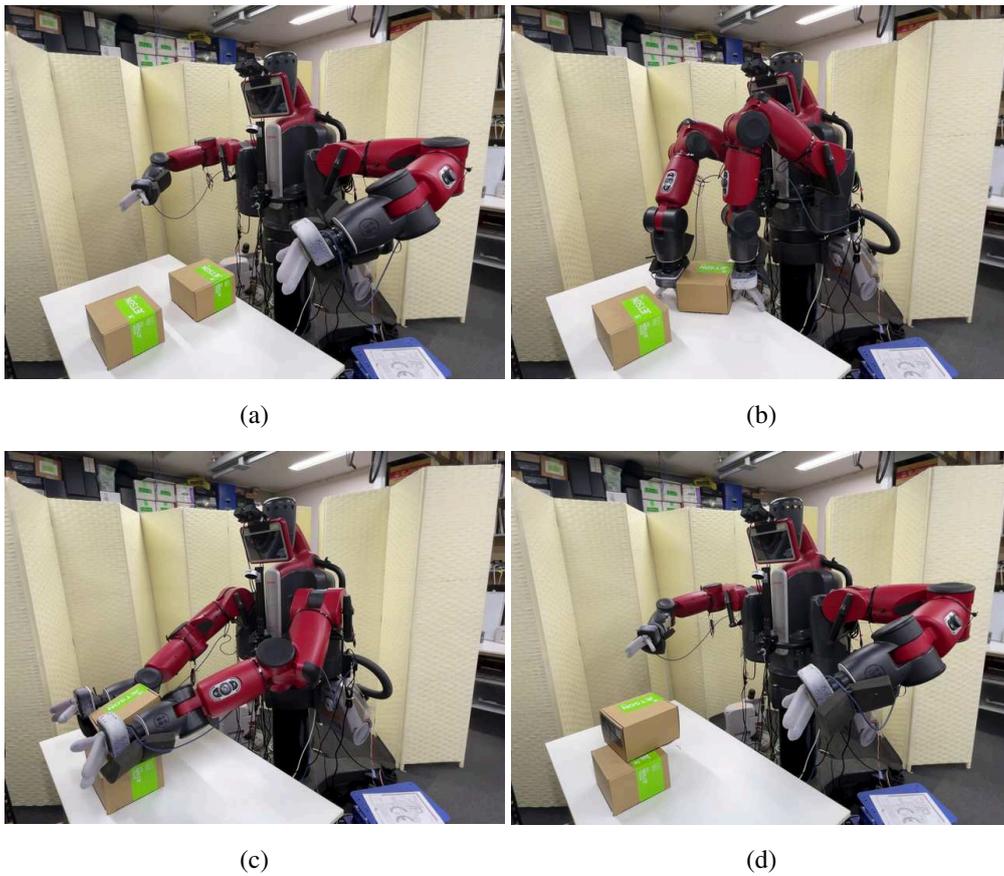


図 7.14: 教示学習結果に基づく箱の Z 軸 90° 回転積み重ね作業の自律実行実験

7.2.6 箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験

本項では箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験について説明する。本実験は 7.2.5 項の教示学習結果に基づく箱の Z 軸 90° 回転積み重ね作業の自律実行実験を行ったあとに実験を行う。操縦者は箱の Z 軸 90° 回転積み重ね作業の自律実行を指示するが、ロボットが箱への Z 軸 90° 回転動作を行ったあとに操縦者が音声を通じて操縦実行への切替を指示する。そしてロボットは操縦実行に切り替えたのちに、操縦者は *Miniature Tangible Cube* を用いて箱の Z 軸 90° 回転積み重ね作業の操縦実行を行い、その作業の作業手順を教示学習する。本実験の教示学習前の作業手順を表す動作順序グラフは 7.2.3 項で学習された動作順序グラフであり図 7.11 に示す。本実験の教示学習前の動作順序グラフは図 7.11 に示すように、箱の積み重ね作業と箱の Z 軸 90° 回転積み重ね作業の作業手順を学習しており、箱を Z 軸 180° 回転させるための動作順序は学習していなかった。また本実験で用いる箱の積み重ね作業の自律実行のための有限状態機械は 7.2.5 項で用いた有限状態機械と同じであり、図 7.13 に示す。この有限状態機械は操縦者による操縦実行への切替に対応するために、6.4 節で説明したように全ての対象物状態ノードは教示学習ノードである (:teach) に遷移することができ、操縦者の音声入力に基づいて教示学習ノードに遷移することで、自律実行から操縦による教示学習への切替を実現した。

本研究で行った箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験については、ロボット側の様子を図 7.15 と図 7.15、操縦者側の様子を図 7.16、操縦者側表示画面を図 7.17 に示す。図 7.15、図 7.16、図 7.17 に示すように、まず操縦者が 7.2.3 項で教示学習された箱の 90° 回転積み重ね作業を自律実行することを 6.5.2.2 目で説明した Web GUI を通じて指示し、ロボットは 7.2.2 項で教示学習された箱の 90° 積み重ね作業を自律実行し、7.2.3 項で教示学習したとおりに手前に置かれた箱を両腕で左側面上方と右側面上方に対して接触し (図 7.15b)、その箱を両腕で抱えながら Z 軸 45° 回転させ (図 7.15c)、一度両腕の接触を離れた後に左前側面上方と右後側面上方に両腕で抱え直し (図 7.15d)、両腕で抱えながらさらに Z 軸 45° 回転させ (図 7.15e)、ロボットは箱を積み重ねるために一度左側面と右側面の接触を離れた (図 7.15f)。

そして、本実験では手前の箱が Z 軸 90° 回転まで自律実行された状態 (図 7.15f) で、操縦者は音声を通じて操縦実行に切り替えるように指示することで、*Miniature Tangible Cube* を用いた操縦実行に移行して、更に Z 軸 90° 回転動作を行うことを教示した。この追加の Z 軸 90° 回転動作のための動作順序の操縦実行は 7.2.6 項で説明した操縦実行と同様に、操縦者が手前に置かれた *Miniature Tangible Cube* に対して両側面上方に接触することで (図 7.16b)、ロボットは手前に置かれた箱を両腕で左側面上方と右側面上方に対して接触し (図 7.15g, 図 7.17b)、操縦者が手前に置かれた *Miniature Tangible Cube* を左側面上方と右側面上方を持ちながら Z 軸 45° 回転させることで (図 7.16c)、ロボットは手前に置かれた箱を両腕で抱えながら Z 軸 45° 回転させた (図 7.15h, 図 7.17c)。さらに操縦者が手前に置かれた *Miniature Tangible Cube* へ

の両腕の接触を一度離してから，同じ **Miniature Tangible Cube** に対して接触していた側面とは異なる側面である左前側面上方と右後側面上方に両腕で接触することで (図 7.16d)，ロボットは手前に置かれた箱を一度両腕の接触を離れた後に左前側面上方と右後側面上方に両腕で抱え直し (図 7.15i, 図 7.17d)，操縦者が手前に置かれた **Miniature Tangible Cube** を左前側面上方と右後側面上方を持ちながら Z 軸 45° 回転させることで (図 7.16e)，ロボットは手前に置かれた箱を両腕で抱えながらさらに Z 軸 45° 回転させた (図 7.15j, 図 7.17e)。

最後は 7.2.2 項で説明した操縦実行と同様に，操縦者が手前に置かれた **Miniature Tangible Cube** への両腕の接触を一度離して，同じ **Miniature Tangible Cube** に対して左側面と右側面に接触することで (図 7.16f)，ロボットは手前に置かれた箱を両腕で左側面と右側面に対して接触することで箱を抱え (図 7.15k, 図 7.17f)。操縦者が抱え込んだ手前の **Miniature Tangible Cube** を持ち上げて奥に置かれた **Miniature Tangible Cube** の上に置くことで (図 7.16g)，ロボットは手前に置かれた箱を左右から抱え込んで持ち上げて奥にある箱の上に積み重ね (図 7.15l, 図 7.15m, 図 7.17g)。操縦者が積み重ねられた **Miniature Tangible Cube** に対する接触から両腕を離すことで (図 7.16h)，ロボットは積み重ねられた箱に対する接触を離れた (図 7.15n, 図 7.17h)。

以上の一連の動作順序の操縦実行によって，ロボットが手前の箱を持ち替えながら Z 軸 180° 回転した後に持ち上げて奥の箱に積み重ねる箱の Z 軸 180° 回転積み重ね作業の作業手順の教示学習を自律実行から操縦実行に作業の動作途中で切り替えて行うことができた。

本実験により教示学習された動作順序グラフを図 7.18 に示す。本実験で教示学習された作業手順は図 7.18 に示す動作順序グラフの赤色に示す左側経路上にあるエッジである以下の 9 個の動作順序となる。

1. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
2. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
3. (dualarm-front-left-back-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左前側面上方に対して左手で，右後側面に対して右手で接触する
4. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
5. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
6. (dualarm-left-right-push-hold-action): 手前の箱 (対象物 0) の右側面に対して右手で接触，左側面に対して左手で接触する
7. (dualarm-push-hold-up-action): 手前の箱 (対象物 0) について両腕抱え込んだまま持ち上げる
8. (dualarm-push-hold-pile-up-action :cube-index 1): 手前の箱 (対象物 0) を奥の箱 (対象物 1) の上に置く

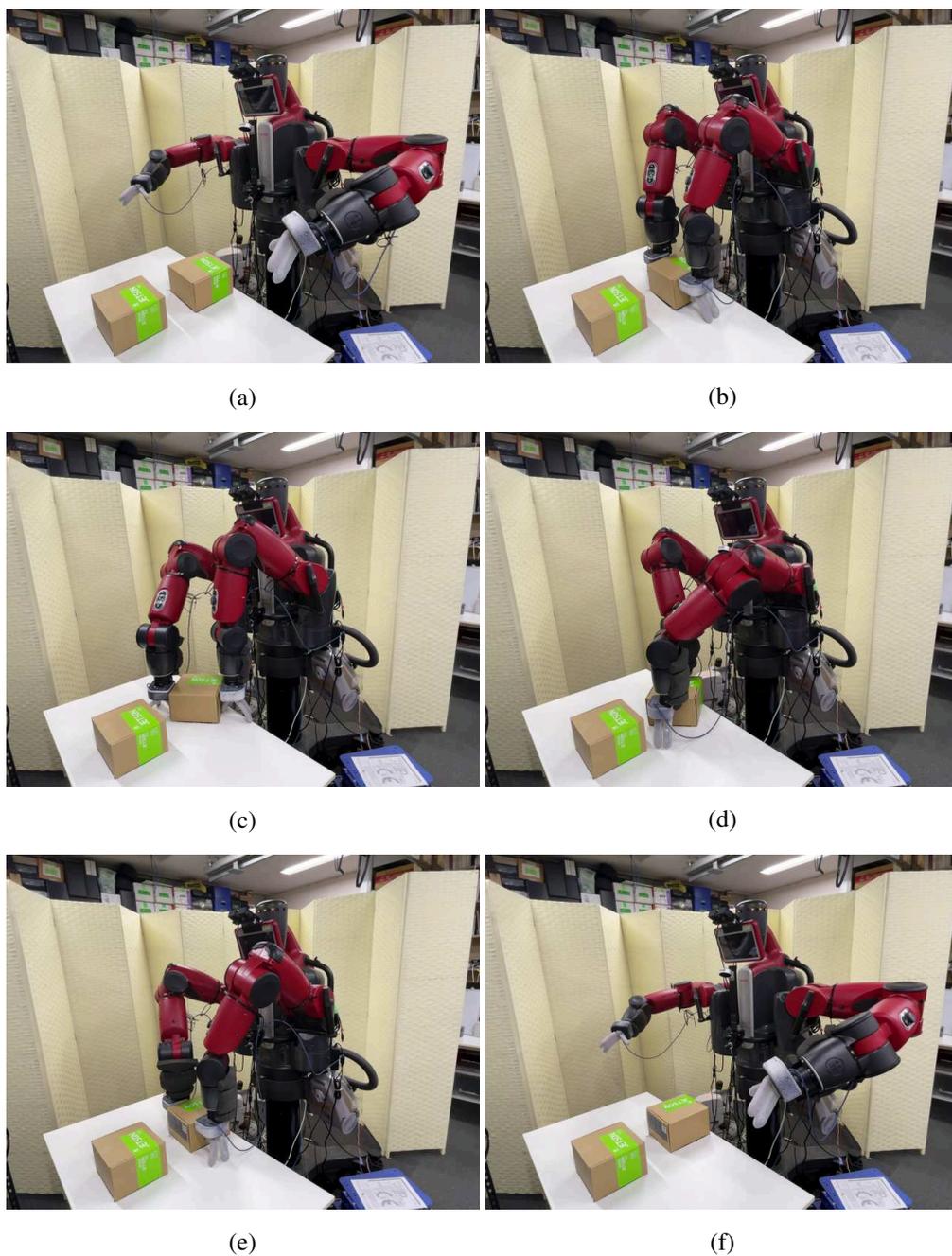


図 7.15: 箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験 (ロボット側) (自律実行部分)

9. (dualarm-hold-release-action): 上に置かれた箱 (対象物 0) の左手と右手による接触を離す

なお本実験では箱を Z 軸 90° 回転させた状態で一度ロボットが接触を離れた状態で自律から操縦に切り替えているため、手前の箱 (対象物 0) の左側面上方に対して左手で、右側面上方に対し

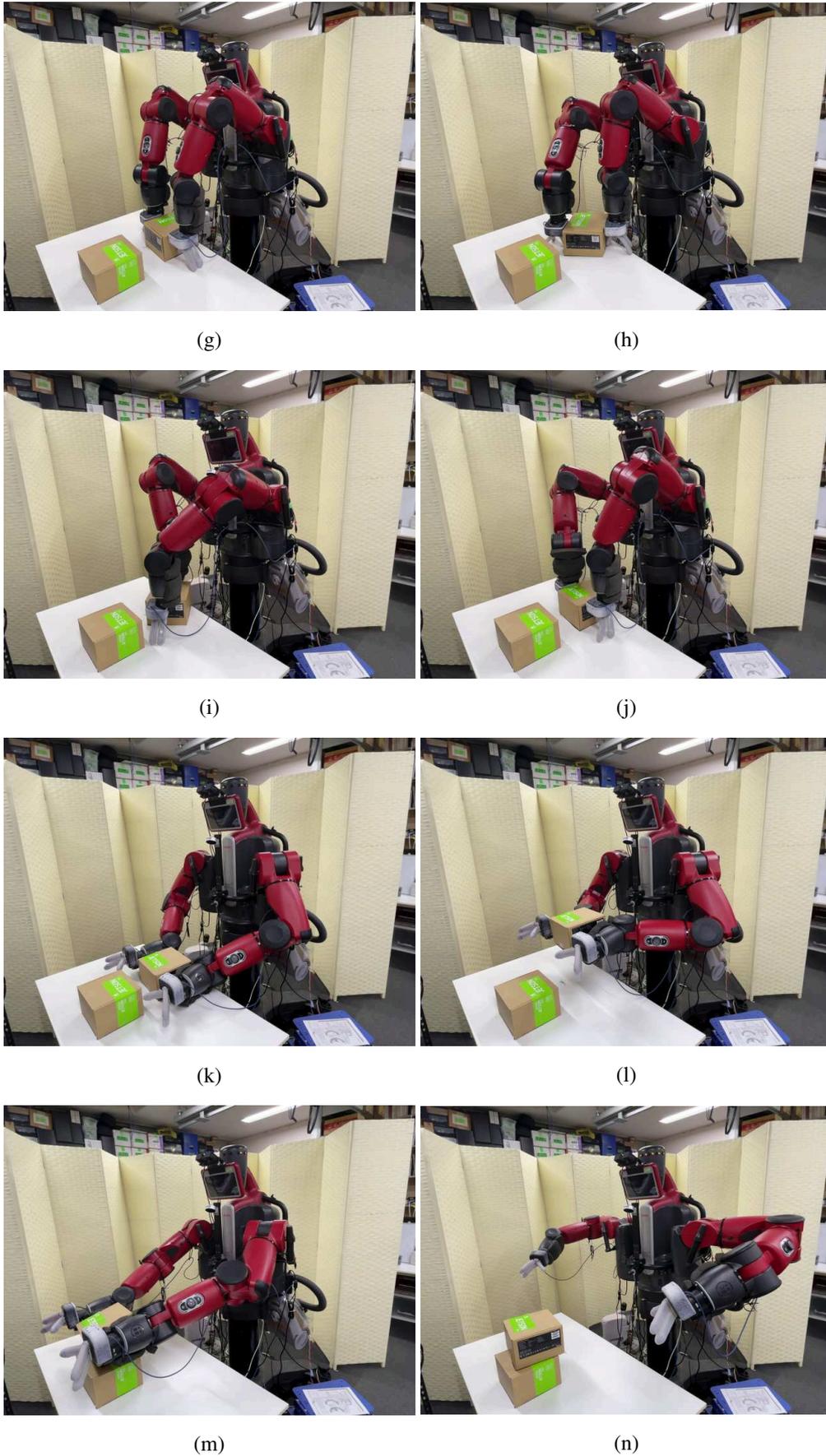


図 7.15: 箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験 (ロボット側) (操縦実行部分)

て右手で接触する動作((dualarm-left-right-top-approach-push-hold-action))を行うことで再度箱に対して左右側面上方から接触する状態に戻る動作の操縦実行を行っているが、これはZ軸180°回転積み重ね作業に必要な動作順序ではないため、教示学習された作業手順とはみなさない。以上の9個の動作順序より、本実験の操縦実行で行ったように、まず箱を両腕で左右側面上方をはさみながらZ軸45°回転し、持ち替えてZ軸45°回転させ、これをもう一通り繰り返して更に箱をZ軸90°回転させたのちに、箱を両腕で左右側面を挟むことで抱え上げ、これを奥の箱の上に置き、そして両腕の接触を離すという、箱のZ軸180°回転積み重ね作業の一連の作業手順を教示学習することができた。

教示学習された動作順序グラフは7.2.3項で教示学習された動作順序グラフ(図7.11)に対して、手前の箱がZ軸90°回転した状態から分岐する形で1つの動作順序グラフとして教示学習することができた。つまり本実験では初期状態から手前の箱がZ軸90°回転した状態に移行するまでは自律実行しているため、この部分の動作順序は7.2.3項で学習された動作順序を再利用しており、それ以降の上記の9個の動作順序を教示学習することで、新たな作業である箱のZ軸180°回転積み重ね作業の作業手順の教示学習を実現した。これは作業手順の再利用による教示学習の効率化が実現しており、図7.18に示すように箱のZ軸180°回転積み重ね作業の作業手順である14個の動作順序(最上部のノードから左下のノードへの経路)のうち、青色の経路部分である5個の動作順序を他の作業の教示学習から再利用し、かつ、上記の9個の動作順序を新たに学習することで新たな作業に対応するための作業手順の教示学習を実現し、切替までの自律実行結果と過去の学習結果を再利用した効率的な作業手順の教示学習の有用性を実証した。

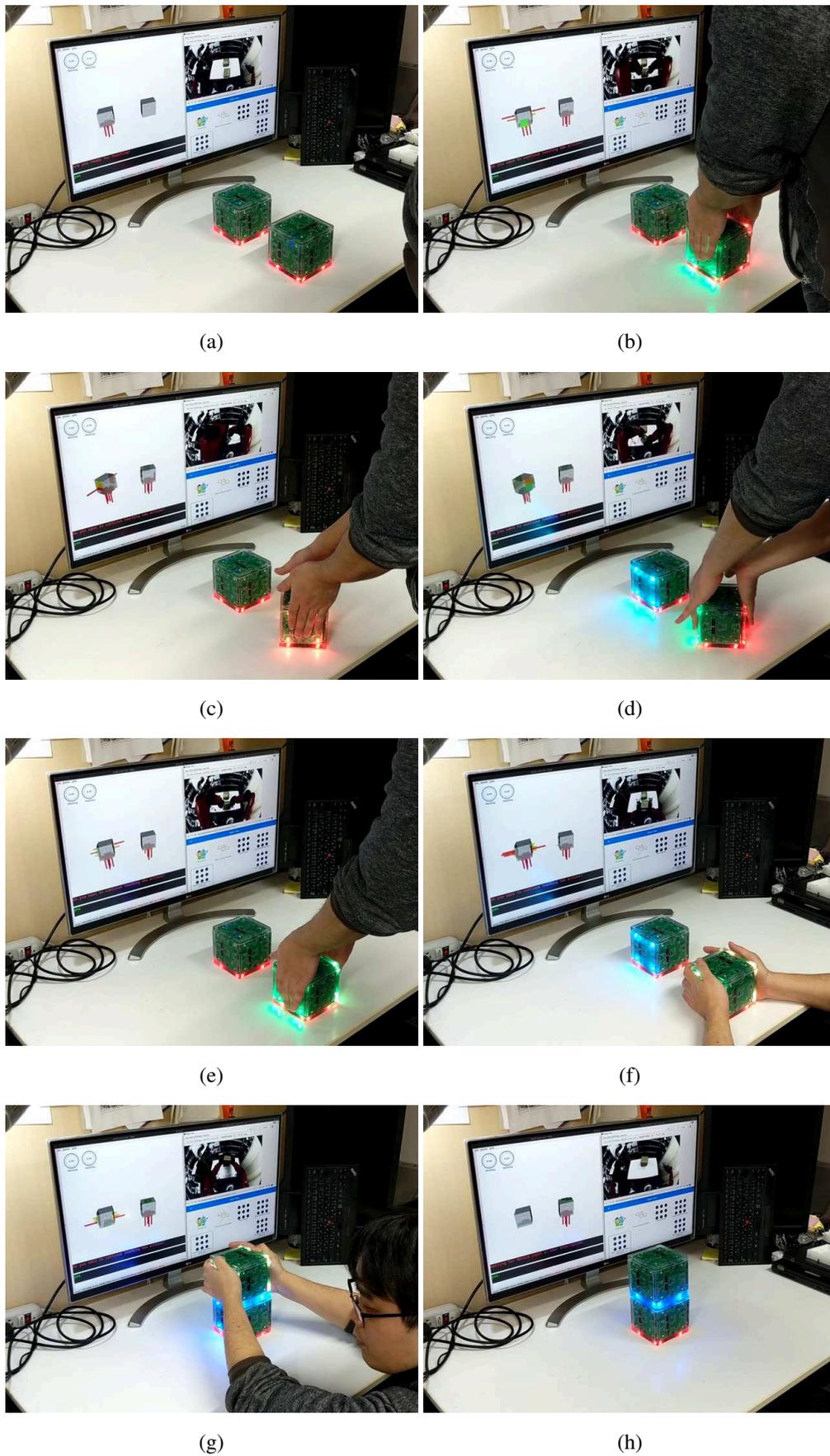


図 7.16: 箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験 (操縦者側)

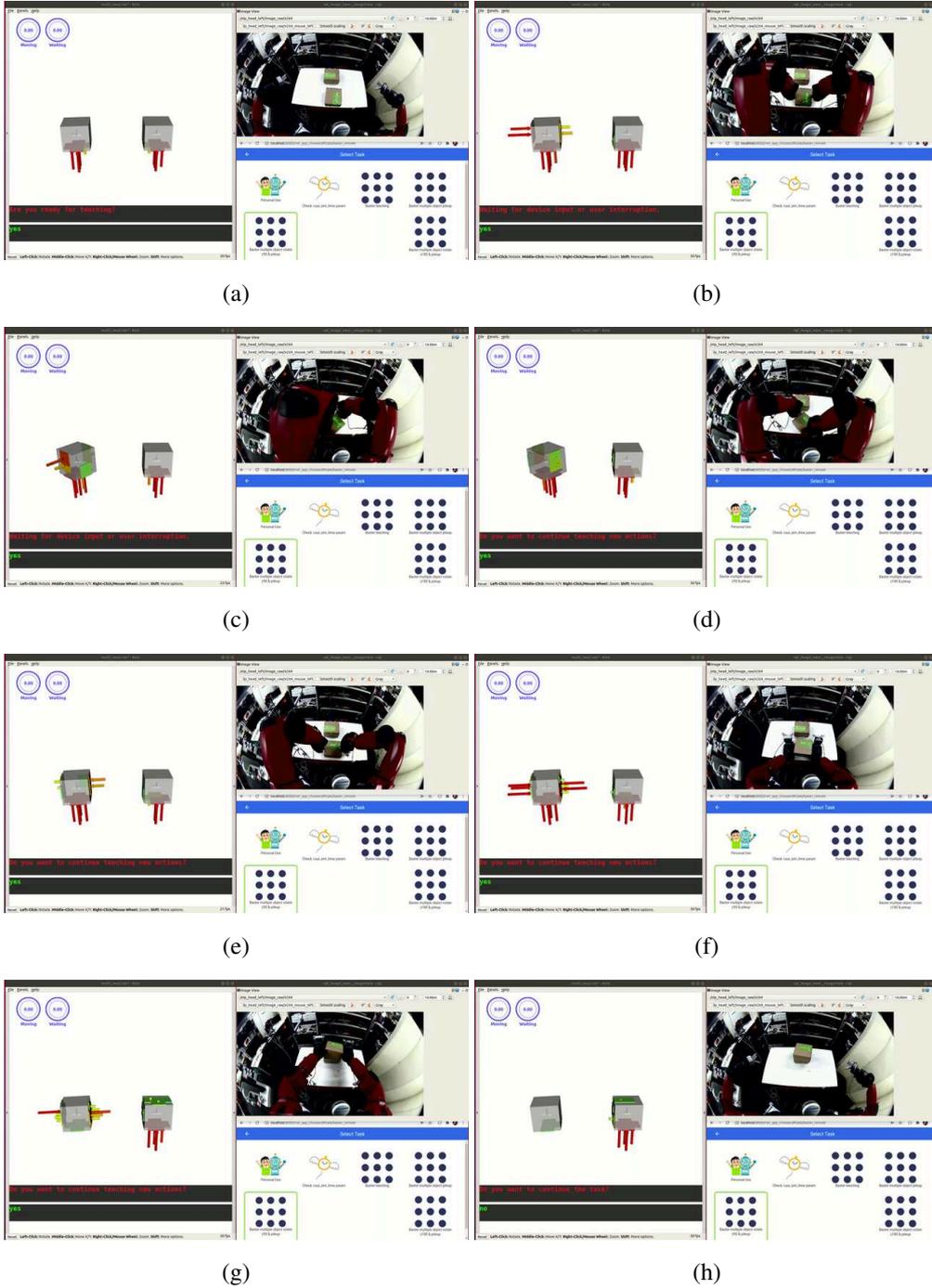


図 7.17: 箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験 (操縦者側表示画面)

7.2.7 作業手順の教示学習結果に基づく箱の Z 軸 180° 回転積み重ね作業の自律実行実験

本項では作業手順の教示学習結果に基づく箱の Z 軸 180° 回転積み重ね作業の自律実行実験について説明する。本実験は 7.2.6 項の箱の Z 軸 90° 回転積み重ね作業の自律実行時における Z 軸 180° 回転積み重ね作業の作業手順の教示学習実験を行ったあとに実験を行う。操縦者は箱の Z 軸 180° 回転積み重ね作業を指示し、ロボットは Z 軸 180° 回転箱の積み重ね作業の自律実行を行った。本実験で用いる作業手順を表す動作順序グラフは 7.2.6 項で学習された動作順序グラフであり図 7.18 に示す。本実験で用いる動作順序グラフは図 7.18 に示すように、箱の積み重ね作業と箱の Z 軸 90° 回転積み重ね作業、箱の Z 軸 180° 回転積み重ね作業の作業手順を学習しており、この学習された動作順序グラフ上で初期の対象物状態から目標の対象物状態への経路を探索することで、箱の Z 軸 180° 回転積み重ね作業の自律実行を実現した。また本実験で用いる箱の Z 軸 180° 回転積み重ね作業の自律実行のための有限状態機械は図 7.19 に示す。図 7.19 に示す有限状態機械のノードとエッジの色は図 7.18 と対応しており、この有限状態機械は図 7.18 に示す上部が青色、下部が赤色の左側経路の動作順序を自律実行でき、かつユーザ入力に応じて操縦に切り替えて作業手順の教示学習を行うことができる有限状態機械となっている。またこの有限状態機械は操縦者の入力に応じた操縦自律間の切替に対応するために、6.4 節で説明したように全ての対象物状態ノードは教示学習ノードである (:teach) に遷移することができるようになっている。

本研究で行った教示学習結果に基づく箱の Z 軸 180° 回転積み重ね作業の自律実行実験については図 7.20 に示す。図 7.20 に示すように、まず操縦者が 7.2.6 項で教示学習された箱の Z 軸 180° 回転積み重ね作業を自律実行することを 6.5.2.2 目で説明した Web GUI を通じて指示し、ロボットは手前に置かれた箱を両腕で左側面上方と右側面上方に対して接触し、その箱を両腕で抱えながら Z 軸 45° 回転させ (図 7.20b)、一度両腕の接触を離れた後に左前側面上方と右後側面上方に両腕で抱え直し、両腕で抱えながらさらに Z 軸 45° 回転させ、続けて持ち替えずに箱を Z 軸 45° 回転させた (図 7.20c)。そして、ロボットが手前に置かれた箱を一度両腕の接触を離れた後に左前側面上方と右後側面上方に両腕で抱え直し、手前に置かれた箱を両腕で抱えながらさらに Z 軸 45° 回転させた。最後はロボットが手前に置かれた箱を両腕で左側面と右側面に対して接触することで箱を抱え (図 7.20d)、手前に置かれた箱を左右から抱え込んで持ち上げて奥にある箱の上に積み重ね (図 7.20e)。積み重ねられた箱に対する接触を離れた (図 7.20f)。以上よりロボットが手前の箱を持ち替えながら Z 軸 180° 回転した後に持ち上げて奥の箱に積み重ねる箱の Z 軸 180° 回転積み重ね作業を自律実行することができた。

本実験で自律実行された動作は以下の 14 個の動作順序であり、これは 7.2.6 項で教示学習された動作順序から 6 つめの動作である (dualarm-hold-release-action) と 7 つめの動作である (dualarm-left-right-top-approach-push-hold-action) を除いたものとなっており、図 7.18 の動作順序グラフの上部は青色、下部を赤色に示す左側経路上にあるエッジである。

1. (dualarm-left-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左側面上方に対して左手で, 右側面上方に対して右手で接触する
2. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
3. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
4. (dualarm-front-left-back-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左前側面上方に対して左手で, 右後側面に対して右手で接触する
5. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
6. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
7. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
8. (dualarm-front-left-back-right-top-approach-push-hold-action): 手前の箱 (対象物 0) の左前側面上方に対して左手で, 右後側面に対して右手で接触する
9. (dualarm-push-rotate-ccw-z45-action): 手前の箱 (対象物 0) を Z 軸に 45° 回転する
10. (dualarm-hold-release-action): 手前の箱 (対象物 0) の左手と右手による接触を離す
11. (dualarm-left-right-push-hold-action): 手前の箱 (対象物 0) の右側面に対して右手で接触, 左側面に対して左手で接触する
12. (dualarm-push-hold-up-action): 手前の箱 (対象物 0) について両腕抱え込んだまま持ち上げる
13. (dualarm-push-hold-pile-up-action :cube-index 1): 手前の箱 (対象物 0) を奥の箱 (対象物 1) の上に置く
14. (dualarm-hold-release-action): 上に置かれた箱 (対象物 0) の左手と右手による接触を離す

以上の 14 個の動作順序より, 本実験の操縦実行で行ったように, まず箱を両腕で左右側面上方をはさみながら Z 軸 45° 回転し, 持ち替えて Z 軸 45° 回転させ, これをもう一通り繰り返して更に箱を Z 軸 90° 回転させたのちに, 箱を両腕で左右側面を挟むことで抱え上げ, これを奥の箱の上に置き, そして両腕の接触を離すという, 7.2.6 項にて教示学習された箱の Z 軸 180° 回転積み重ね作業の作業手順を自律実行することができた。

本実験における箱の Z 軸 180° 回転積み重ね作業は上記の 14 個の動作順序を自律実行することで実現したが, この動作順序は 7.2.3 項にて学習された図 7.18 において青色で示す 5 個の動作順序と, 7.2.6 項にて学習された図 7.18 において赤色で示す 9 個の動作順序が繋がったものであり, これは 7.2.6 項において箱の Z 軸 180° 回転積み重ね作業の作業手順の教示学習す

る際に 7.2.3 項にて学習された動作順序の一部を再利用して効率的に教示学習したことを示しており，よって本実験ではこのような過去の実行・学習結果を再利用した効率的な操縦手順の教示学習の有用性と，その効率的な教示学習によって学習された操作手順を用いた作業の自律実行が可能であることを実証した。

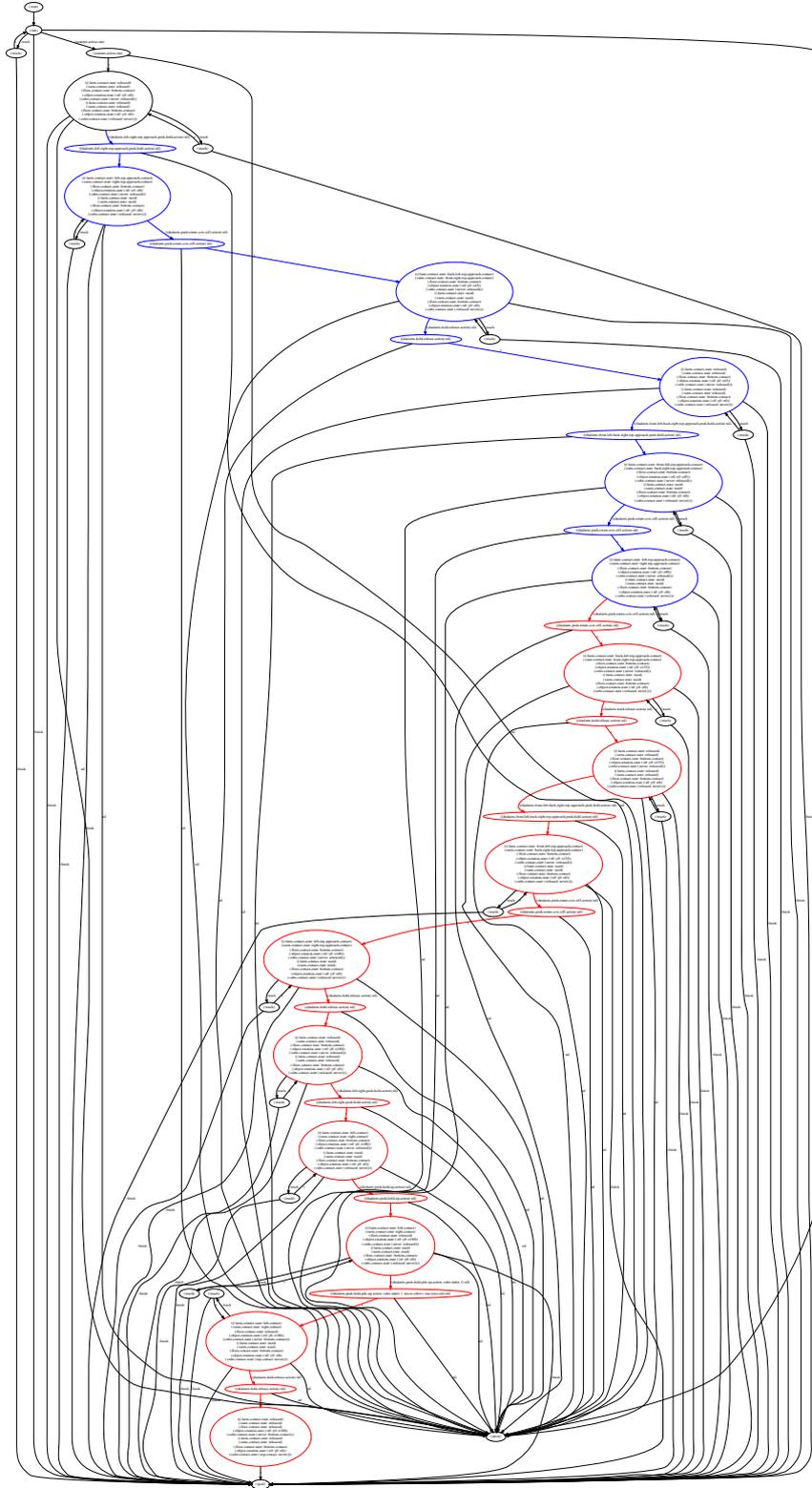


図 7.19: 箱の Z 軸 180° 回転積み重ね作業の自律実行のための有限状態機械: ノードとエッジの色は図 7.18 と対応しており, 図 7.18 に示す上部が青色, 下部が赤色の左側経路の動作順序を自律実行することができ, かつユーザ入力に応じて操縦に切り替えて作業手順の教示学習を行うことができる有限状態機械となっている.

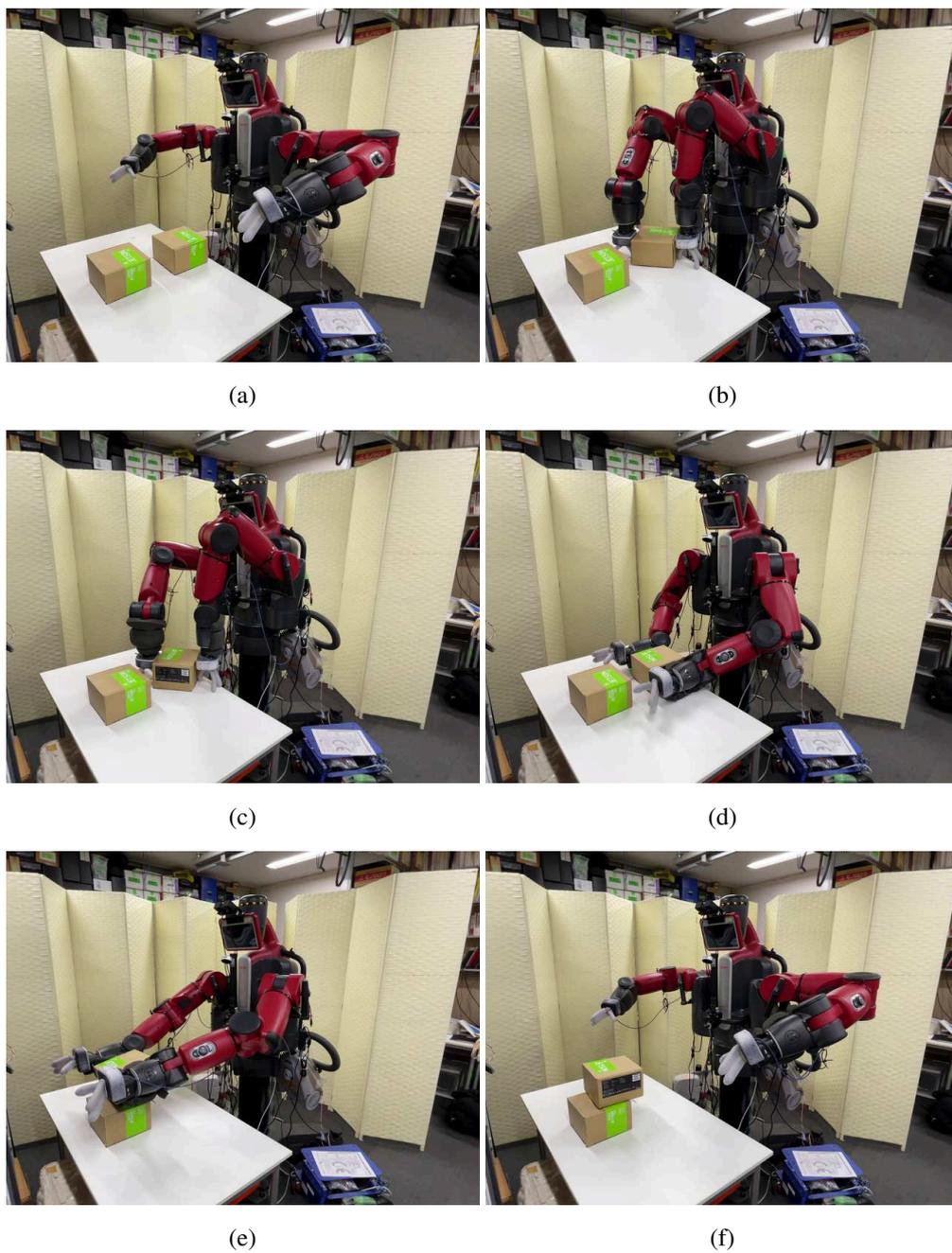


図 7.20: 教示学習結果に基づく箱の Z 軸 180° 回転積み重ね作業の自律実行実験

7.3 自律学習型把持動作を伴う物体ダンボール箱収納作業の操縦自律間切替実行における作業手順の教示学習実験

本研究で提案する操縦自律間横断学習ロボットシステムの操縦自律間横断学習機能と自律学習との統合可能性の評価実験として、自律学習型把持動作を伴う物体ダンボール箱収納作業の操縦自律間切替実行における作業手順の教示学習実験を行う。本実験では第4章で説明したタンジブルインタフェースを用いた操縦実行・教示学習と、第5章で説明した学習型動作戦略を用いた自律実行・自律学習、第6章で説明した操縦自律間切替実行システムを用いることで、本研究で提案する操縦自律間横断学習ロボットシステムによる作業実行形態横断的な学習を実現し、この操縦自律間横断学習ロボットシステムが構築可能であることを実証した。

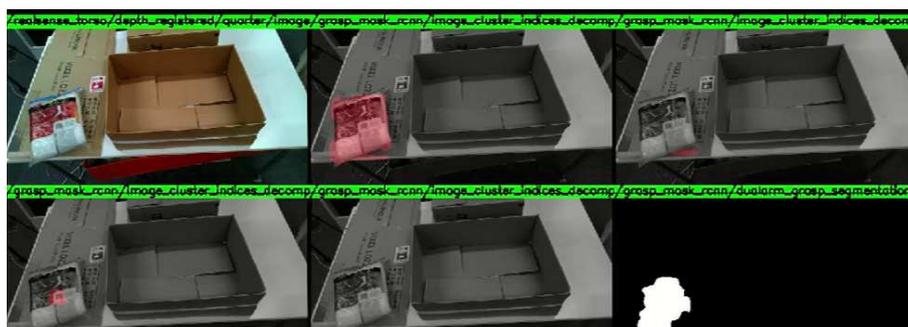
本実験については以下の4つの実験を以下に示す順番に行う。

- 7.3.2 項「ダンボール積み重ね作業の作業手順の教示学習実験」
- 7.3.3 項「作業手順の教示学習結果に基づくダンボール積み重ね作業の自律実行実験」
- 7.3.4 項「ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験」
- 7.3.5 項「作業手順の教示学習結果に基づく自律学習型把持動作を伴う物体ダンボール箱収納作業の自律実行実験」

本実験はロボットは作業手順を何も学習していない状態、CNN 把持点予測器については既に自律学習された状態から一連の実験を順に行った。なお本実験で用いた CNN 把持点予測器は 5.7.2 項で自律学習を行った Mask-RCNN を元にした把持点予測器を用い、把持動作の際にはこの把持点予測器で予測された把持点を用いて把持動作を行った。まず Miniature Tangible Cube を用いてダンボール積み重ね作業を操縦実行し、ロボットにダンボール積み重ね作業の作業手順を教示学習した (7.3.2 項)。そして、この操縦実行によって正しく作業手順が教示学習されているかを確認するためにダンボール積み重ね作業の自律実行を行った (7.3.3 項)。さらに、次にロボットに箱の積み重ね作業を自律実行させた際に、作業の序盤にて音声で操縦実行に切り替えて物体を把持してダンボール箱に収納する動作(把持物体収納動作)を操縦実行し、この動作を箱の積み重ね動作の前にこの動作を行う物体ダンボール箱収納作業をロボットに教示学習した (7.3.4 項)。最後に作業手順が正しく教示学習されているかを確認するために、教示学習された物体ダンボール箱収納作業を自律実行した (7.3.5 項)。なお本実験では操縦者は遠隔ロボットの頭部カメラの画像に基づいて操縦実行を行い、タンジブルインタフェースとして1台の Miniature Tangible Cube Rev.1 と2台の Miniature Tangible Cube Rev.2 を用いた。また本実験では把持物体として Amazon Robotics Challenge [97] に用いられたピッキングタスク対象物品から図 4.39 に示す袋入り靴下を用いた。

本実験で用いる Mask-RCNN を元にした把持点予測器による単腕把持可能領域の予測結果の一例を図 7.21 に示す。本実験ではこの把持点予測器の画像入力には遠隔ロボットの腹部

RGB-D カメラの RGB 画像を用い、単腕把持の把持可能領域 (図 7.21a の右下の白色の領域) を予測し、予測された領域の深度画像情報から把持可能領域を直方体近似 (図 7.21b の緑色の直方体) することで認識し、把持動作はこの直方体領域の中央に対して把持動作を上方から行った。なお図 7.21a の左下は単腕把持可能ラベルの可視化であるが、5.5.4.4 目で説明したように把持可能ラベルの確率画像と意味ラベルの確率画像のピクセル積算された結果に対して閾値を設定することで図 7.21a の右下の単腕把持可能領域を計算した。本実験では以上のように予測された把持可能領域に対して把持動作を行った。



(a) Mask-RCNN を元にした把持点予測器による予測結果の可視化 (赤もしくは白色が予測領域) (左上: 入力 RGB 画像, 中央上: 物体意味領域, 右上: インスタンス遮蔽領域, 左下: 単腕把持可能ラベル, 中央下: 双腕把持可能ラベル, 右下: 単腕把持可能領域)



(b) Mask-RCNN を元にした把持点予測器による単腕把持可能領域の直方体近似

図 7.21: Mask-RCNN を元にした把持点予測器による単腕把持可能領域の予測結果

7.3.1 物体ダンボール箱収納作業で検証するユーザ入力に応じた操縦自律間切替実行

具体的な個々の実験を説明する前に、まず本実験で想定するユーザ入力に応じた操縦自律間切替実行について説明する。本実験ではユーザ入力に応じた操縦自律間切替実行として、作業動作開始前の自律実行から操縦による教示学習への切替について評価を行った。作業動作開始前の切替とは、7.2.1 項で説明したように作業を自律実行する際に作業動作を開始する直前で操縦による教示学習に切り替えることで事前に学習した作業の作業手順とは最初から異なる作業手順を教示学習するものであり、本実験では7.3.4 項「ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験」にて評価を行った。想定する状況としては、操縦者はある作業の自律実行を指示したがロボットの前にある対象物の状態が想定と異なっていた、もしくは作業の目標状態を変更したため新しい作業のための作業手順の変更を教示学習するというものであり、本実験では箱を積み重ねる作業の自律実行の際に、作業動作の開始前にユーザが音声で割り込むことで自律実行から操縦実行へ切り替え、先に物体を把持して箱に収納したのちに積み重ねるといった作業を教示学習する実験を行った。

7.3.2 ダンボール積み重ね作業の作業手順の教示学習実験

本項ではダンボール積み重ね作業の作業手順の教示学習実験について説明する。本実験ではロボットは作業手順について何も学習していない状態から実験を開始し、Miniature Tangible Cubeを用いてダンボール積み重ね作業の操縦実行を行い、その作業の作業手順を教示学習した。なお本実験は使用する Miniature Tangible Cube の台数や対象とする箱の種類は異なるが、7.2.2 項と同じ作業手順を教示学習するものである。

本実験の教示学習前の作業手順を表す動作順序グラフは図 7.22 に示すように、本実験の教示学習前はロボットは作業手順については学習しておらず、実験前の対象物の状態のみが与えられていた。また箱の積み重ね作業の操縦実行のための有限状態機械は図 7.2 に示すように、作業開始すると有限状態機械は作業の初期化ノードである (:init) と (statenet-action-init) に遷移したのちに、教示学習ノードである (:teach) に遷移することで操縦実行および教示学習を実現した。

本研究で行ったダンボール積み重ね作業の教示学習実験については、ロボット側の様子を図 7.24、操縦者側の様子を図 7.25、操縦者側表示画面を図 7.26 に示す。図 7.24、図 7.25、図 7.26 に示すように、まず操縦者が Miniature Tangible Cube を用いた操縦実行することを 6.5.2.1 目で説明した音声ベース作業実行システムを用いて音声入力で指示した。そして操縦者が手前に置かれた Miniature Tangible Cube に対して左側面と右側面に接触することで (図 7.25b)、ロボットは手前に置かれたダンボール箱を両腕で左側面と右側面に対して接触することでダンボール箱を抱え (図 7.24b、図 7.26b)、操縦者が抱え込んだ手前の Miniature Tangible Cube を

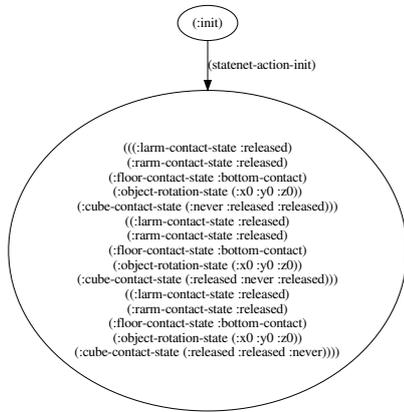


図 7.22: ダンボール積み重ね作業の教示学習実験における学習前の動作順序グラフ

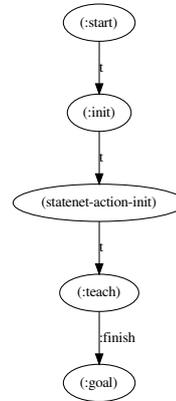


図 7.23: ダンボール積み重ね作業の教示学習のための有限状態機械

持ち上げて奥に置かれた Miniature Tangible Cube の上に置くことで (図 7.25c), ロボットは手前に置かれたダンボール箱を左右から抱え込んで持ち上げて奥にあるダンボール箱の上に積み重ね (図 7.24c, 図 7.26c), 最後に操縦者が積み重ねられた Miniature Tangible Cube に対する接触から両腕を離すことで (図 7.25d), ロボットは積み重ねられたダンボール箱に対する接触を離れた (図 7.24d, 図 7.26d). 以上の一連の動作順序の操縦実行によって, ロボットが手前のダンボール箱を持ち上げて奥のダンボール箱に積み重ねるダンボール箱の積み重ね作業の作業手順を教示学習することができた.

本実験により教示学習された動作順序グラフを図 7.27 に示す. 教示学習された作業手順を表す動作順序グラフは 7.2.2 項で教示された動作順序グラフと対象物状態ノードが異なるものの動作順序は同じであり, 図 7.27 の動作順序グラフのエッジである以下の 4 つの動作の順序となる.

1. (dualarm-left-right-push-hold-action): 手前のダンボール箱 (対象物 0) の左側面に対して左手で, 右側面に対して右手で接触する
2. (dualarm-push-hold-up-action): 手前のダンボール箱 (対象物 0) について両腕抱え込んだまま持ち上げる
3. (dualarm-push-hold-pile-up-action :cube-index 1): 手前のダンボール箱 (対象物 0) を奥のダンボール箱 (対象物 1) の上に置く
4. (dualarm-hold-release-action): 上に置かれたダンボール箱 (対象物 0) の左手と右手による接触を離す

以上 4 つの動作順序より本実験の操縦実行で行ったように, まずダンボール箱を両腕で左右側面を挟むことで抱え上げ, これを奥のダンボール箱の上に置き, そして両腕の接触を離すとい

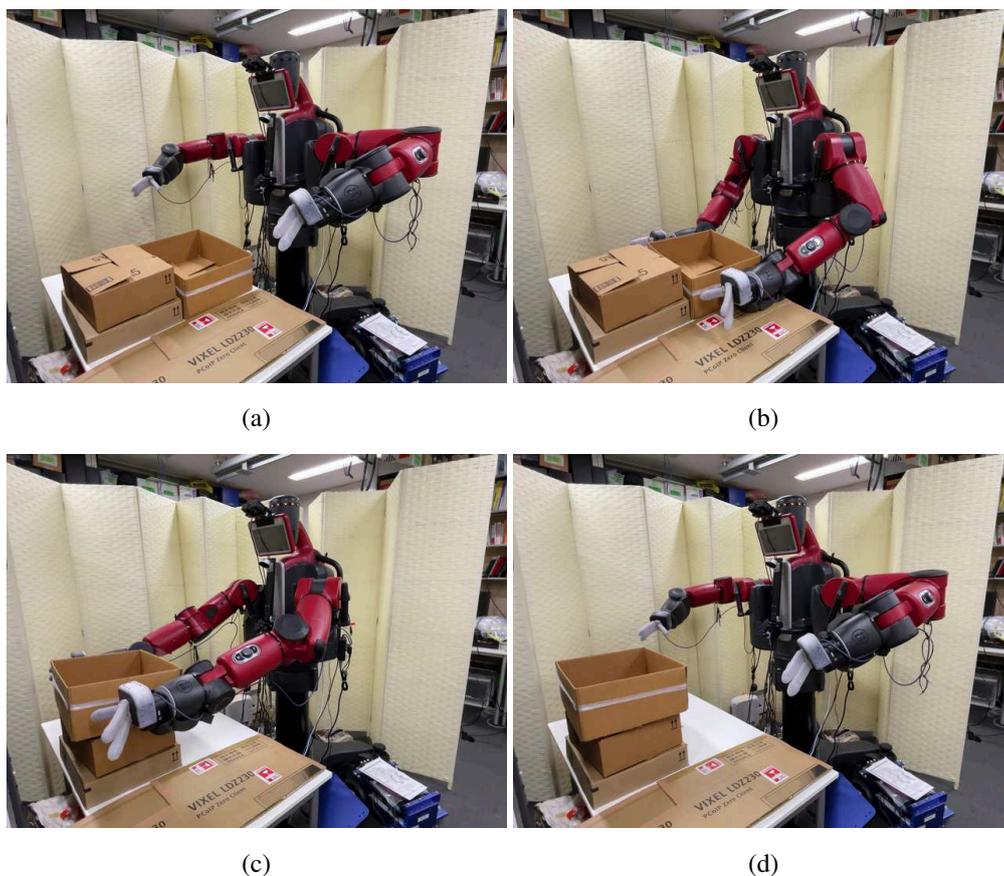


図 7.24: ダンボール積み重ね作業の教示学習実験 (ロボット側)

う，ダンボール積み重ね作業の一連の作業手順を教示学習することができた。

7.3.3 作業手順の教示学習結果に基づくダンボール積み重ね作業の自律実行実験

本項では教示学習結果に基づくダンボール積み重ね作業の自律実行実験について説明する。本実験は 7.3.2 項のダンボール積み重ね作業の教示学習実験を行ったあとに実験を行い，操縦者はダンボール積み重ね作業を指示しロボットはダンボール積み重ね作業の自律実行を行った。なお本実験は使用する Miniature Tangible Cube の台数や対象とする箱の種類は異なるが，7.2.4 項と同じ操作手順を自律実行するものである。本実験で用いる作業手順を表す動作順序グラフは 7.3.2 項で学習された動作順序グラフであり，図 7.27 に示す。本実験で用いる動作順序グラフは図 7.27 に示すように，ダンボール積み重ね作業の作業手順のみを学習しており，この学習された動作順序グラフ上で初期の対象物状態から目標の対象物状態への経路を探索することで，ダンボール積み重ね作業の自律実行を実現した。また本実験で用いるダンボール積み重ね作業の自律実行のための有限状態機械は 7.2.4 項で用いた有限状態機械と対象物状態は

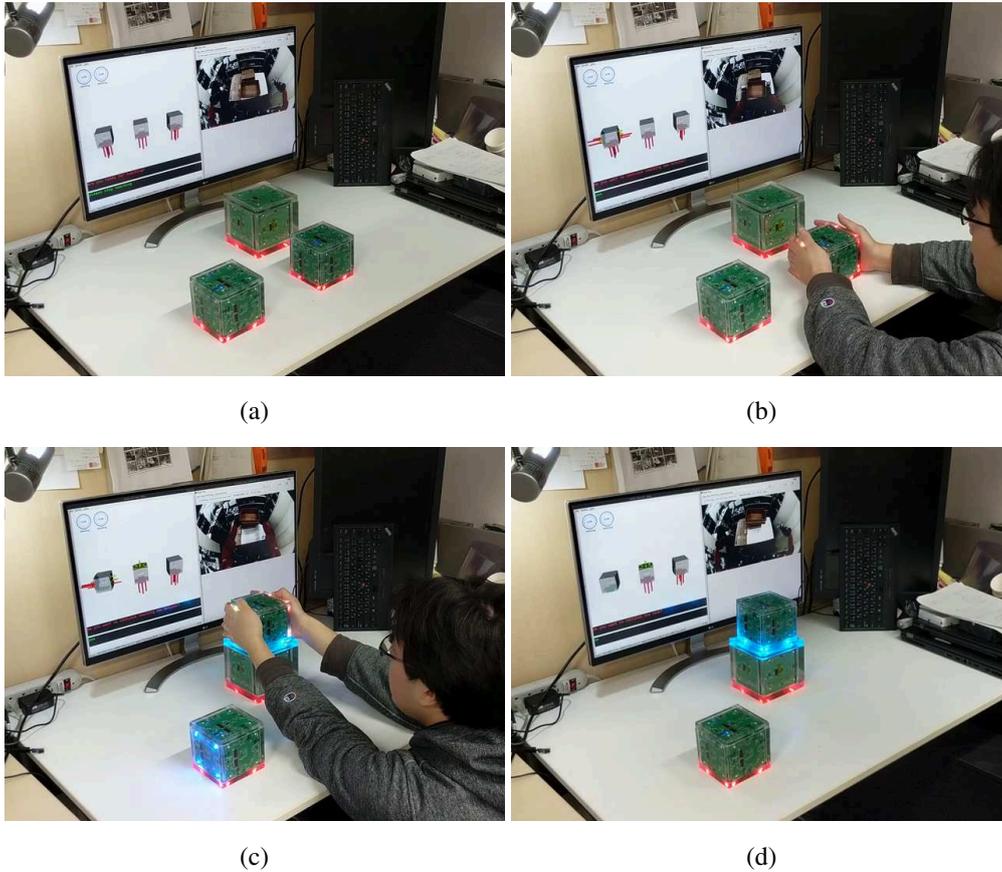


図 7.25: ダンボール積み重ね作業の教示学習実験 (操縦者側)

異なるものの実行される動作は同じであり，図 7.7 に示す．この有限状態機械は操縦者の入力に応じた操縦自律間の切替に対応するために，6.4 節で説明したように全ての対象物状態ノードは教示学習ノードである (`:teach`) に遷移することができるようになっている．

本研究で行った教示学習結果に基づくダンボール積み重ね作業の自律実行実験については図 7.28 に示す．図 7.28 に示すように，まず操縦者が 7.3.2 項で教示学習された箱の積み重ね作業を自律実行することを 6.5.2.1 目で説明した音声ベース作業実行システムを用いて音声入力で指示し，ロボットが手前に置かれたダンボール箱を両腕で左側面と右側面に対して接触することでダンボール箱を抱え (図 7.28b)，そしてロボットが手前に置かれたダンボール箱を左右から抱え込んで持ち上げて奥にあるダンボール箱の上に積み重ね (図 7.28c)，最後にロボットが積み重ねられたダンボール箱に対する接触を離れた (図 7.28d)．以上よりロボットが手前のダンボール箱を持ち上げて奥のダンボール箱に積み重ねるダンボール箱の積み重ね作業を自律実行することができた．

本実験で自律実行された動作は以下の 4 つの動作順序であり，これは 7.3.2 項で教示学習された動作順序と同じものであり，図 7.27 の動作順序グラフにあるエッジである．

1. (`dualarm-left-right-push-hold-action`): 手前のダンボール箱 (対象物 0)

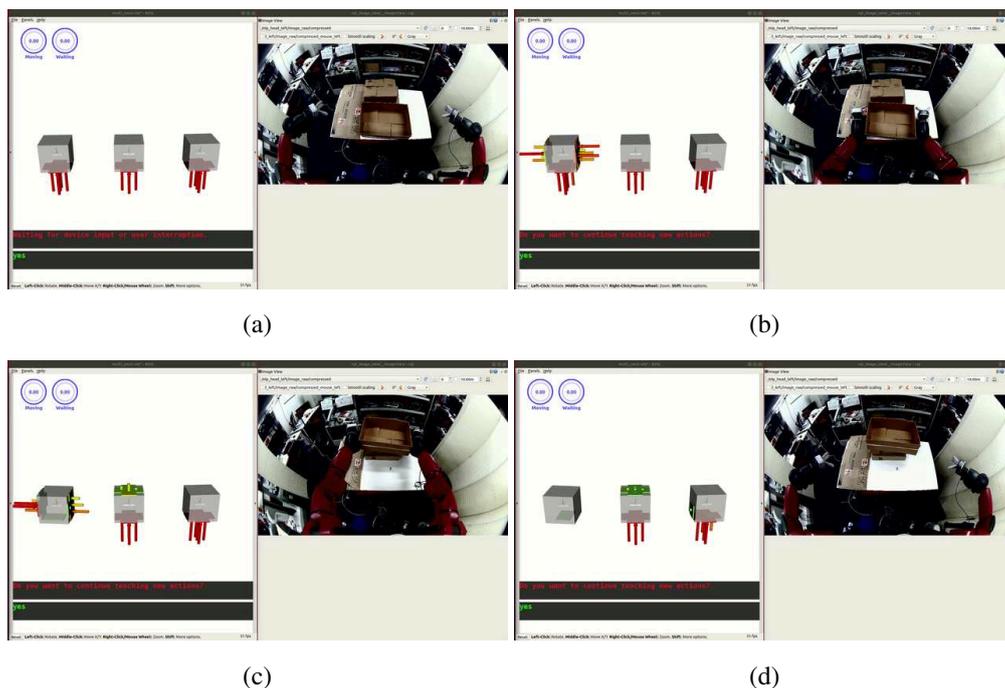


図 7.26: ダンボール積み重ね作業の教示学習実験 (操縦者側表示画面)

- の左側面に対して左手で、右側面に対して右手で接触する
2. (dualarm-push-hold-up-action): 手前のダンボール箱 (対象物 0) について両腕抱え込んだまま持ち上げる
 3. (dualarm-push-hold-raise-action :cube-index 1): 手前のダンボール箱 (対象物 0) を奥のダンボール箱 (対象物 1) の上に置く
 4. (dualarm-hold-release-action): 上に置かれたダンボール箱 (対象物 0) の左手と右手による接触を離す

以上の 4 つの動作順序より、本実験では 7.3.2 項で教示学習されたように、まずダンボール箱を両腕で左右側面を挟むことで抱え上げ、これを奥のダンボール箱の上に置き、そして両腕の接触を離すという、7.3.3 項にて教示学習されたダンボール積み重ね作業の作業手順を自律実行することができた。

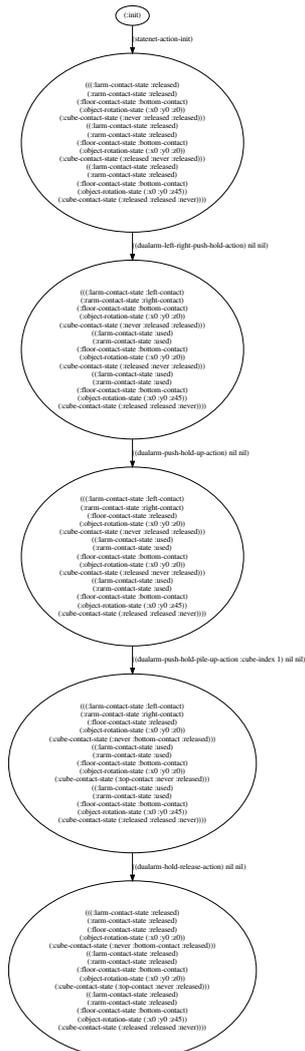


図 7.27: ダンボール積み重ね作業の教示学習実験で学習された動作順序グラフ

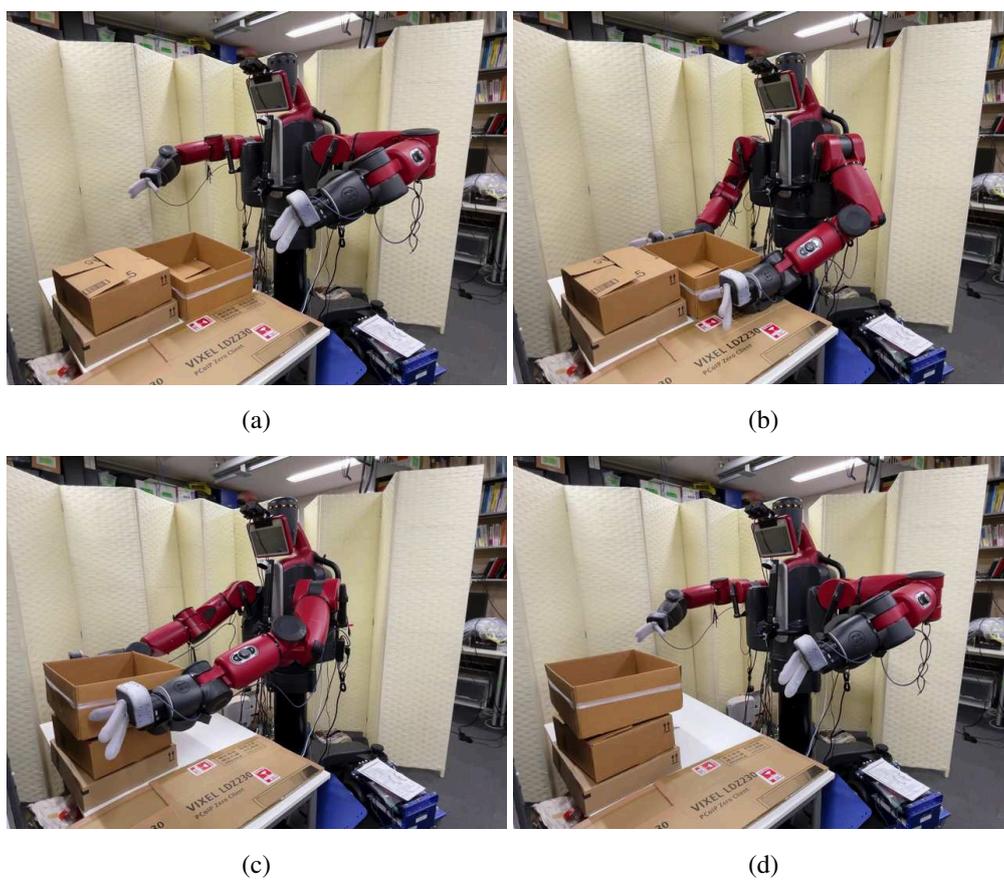


図 7.28: 教示学習結果に基づくダンボール積み重ね作業の自律実行実験

7.3.4 ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験

本項ではダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験について説明する。本実験は7.3.3項のダンボール積み重ね作業の自律実行実験を行った後に実験を行う。操縦者はダンボール積み重ね作業の自律実行を指示するが、ロボットがダンボール箱への操作動作を行う前に操縦者が音声を通じて操縦実行への切替を指示する。そしてロボットは操縦実行に切り替えたのちに、操縦者は **Miniature Tangible Cube** を用いて自律学習型把持動作を伴う物体ダンボール箱収納作業の操縦実行を行い、その作業の作業手順を教示学習する。本実験の教示学習前の作業手順を表す動作順序グラフは7.3.2項で学習された動作順序グラフであり図7.27に示す。本実験の教示学習前の動作順序グラフは図7.27に示すようにダンボール積み重ね作業の作業手順のみを学習しており、物体を把持し収納させるための動作順序は学習していなかった。また本実験で用いる箱の積み重ね作業の自律実行のための有限状態機械は7.2.4項で用いた有限状態機械と対象物状態は異なるものの実行される動作は同じであり、図7.7に示す。この有限状態機械は操縦者による操縦実行への切替に対応するために、6.4節で説明したように全ての対象物状態ノードは教示学習ノードである (:teach) に遷移することができ、操縦者の音声入力に基づいて教示学習ノードに遷移することで、自律実行から操縦による教示学習への切替を実現した。

本研究で行ったダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験については、ロボット側の様子を図7.29、操縦者側の様子を図7.30、操縦者側表示画面を図7.31に示す。図7.29、図7.30、図7.31に示すように、まず操縦者が7.3.2項で教示学習されたダンボール箱の積み重ね作業を自律実行することを6.5.2.1目で説明した音声ベース作業実行システムを用いて音声入力で指示した。そこでロボットは7.3.2項で教示学習されたダンボール箱の積み重ね作業を自律実行しようとするが、ロボットが作業動作を実行する前に操縦者は音声を通じて操縦実行に切り替えるように指示することで、**Miniature Tangible Cube** を用いた操縦実行に移行した。そして4.9.5.1目と同様に、操縦者が左側に置かれた **Miniature Tangible Cube** に対して上側面から掴んで持ち上げることで(図7.30b)、ロボットは左側に置かれた把持物体を左腕で把持し持ち上げ(図7.29b, 図7.31b)、操縦者が持ち上げた **Miniature Tangible Cube** を中央手前に置かれた **Miniature Tangible Cube** の上に置き、上側面から掴んでいた **Miniature Tangible Cube** を離すことで(図7.30c)、ロボットは把持物体を中央に置かれたダンボール箱の上まで移動させ、把持物体を離して中央に置かれたダンボール箱の中に把持物体を収納した(図7.29c, 図7.31c)。

この後の動作は7.3.2項と同様に、操縦者が中央手前に積み重ねられた2つの **Miniature Tangible Cube** の下のデバイスを左右から接触して持ち上げることで(図7.30d)、ロボットは把持物体が収納されたダンボール箱を左右から抱え込んで持ち上げ(図7.29d, 図7.31d)、操縦者が持ち上げた2つの **Miniature Tangible Cube** を奥に置かれた **Miniature Tangible Cube** の上に置くことで(図7.30e)、ロボットは把持物体が収納されたダンボール箱を奥にあるダンボー

ル箱の上に積み重ねた (図 7.29e, 図 7.31e). 最後に操縦者抱えていた Miniature Tangible Cube に対する接触を左右と離すことで (図 7.30f), ロボットは積み重ねられたダンボール箱に対する接触を離れた (図 7.29f, 図 7.31f).

以上の一連の動作順序の操縦実行によって, ロボットが左側におかれた物体を把持して持ち上げて, それを中央手前に置かれたダンボール箱に収納したあとに, 物体を収納したそのダンボール箱を奥にあるダンボール箱の上に積み重ねるという, 物体ダンボール箱収納作業の作業手順の教示学習を自律実行から操縦実行に作業の動作前に切り替えて行うことができた.

本実験により教示学習された動作順序グラフを図 7.32 に示す. 本実験において教示学習された作業手順は図 7.32 の動作順序グラフの赤色に示す左側経路上にあるエッジである以下の 8 個の動作順序となる.

1. (larm-top-grasp-action): 左腕で把持物体 (対象物 2) を上から把持する
2. (larm-grasp-up-action): 左腕で把持した把持物体 (対象物 2) を持ち上げる
3. (larm-grasp-pile-up-action :cube-index 0): 左腕で把持した把持物体 (対象物 2) を手前のダンボール箱 (対象物 0) の上まで移動する
4. (larm-grasp-release-action): 左腕で把持した把持物体 (対象物 2) を離す
5. (dualarm-left-right-push-hold-action): 手前のダンボール箱 (対象物 0) の右側面に対して右手で接触, 左側面に対して左手で接触する
6. (dualarm-push-hold-up-action): 手前のダンボール箱 (対象物 0) について両腕抱え込んだまま持ち上げる
7. (dualarm-push-hold-pile-up-action :cube-index 1): 手前のダンボール箱 (対象物 0) を奥のダンボール箱 (対象物 1) の上に置く
8. (dualarm-hold-release-action): 上に置かれたダンボール箱 (対象物 0) の左手と右手による接触を離す

なお教示学習された動作順序グラフは 7.3.2 項で教示学習された動作順序グラフ (図 7.27) に対して, に対して, 作業の初期対象物状態から分岐する形で 1 つの動作順序グラフとして教示学習することができた. 以上の 8 個の動作順序より本実験の操縦実行で行ったように, まず把持物体を左腕で把持し持ち上げ手前のダンボール箱の上まで移動させて把持を離すことで物体を収納したのちに, ダンボール箱を両腕で左右側面を挟むことで抱え上げ, これを奥のダンボール箱の上に置き, そして両腕の接触を離すという, 物体ダンボール箱収納作業の作業手順の教示学習することができた.

7.3.5 作業手順の教示学習結果に基づく自律学習型把持動作を伴う物体ダンボール箱収納作業の自律実行実験

本項では教示学習結果に基づく自律学習型把持動作を伴う物体ダンボール箱収納作業の自律実行実験について説明する. 本実験は 7.3.4 項のダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験を行ったあ



図 7.29: ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験 (ロボット側)

とに実験を行い、操縦者は物体ダンボール箱収納作業を指示しロボットは物体ダンボール箱収納作業の自律実行を行った。本実験で用いる動作順序グラフは7.3.4項で学習された動作順序グラフであり図7.32に示す。本実験で用いる動作順序グラフは図7.32に示すように、ダンボール積み重ね作業と物体ダンボール箱収納作業の作業手順を学習しており、この学習された

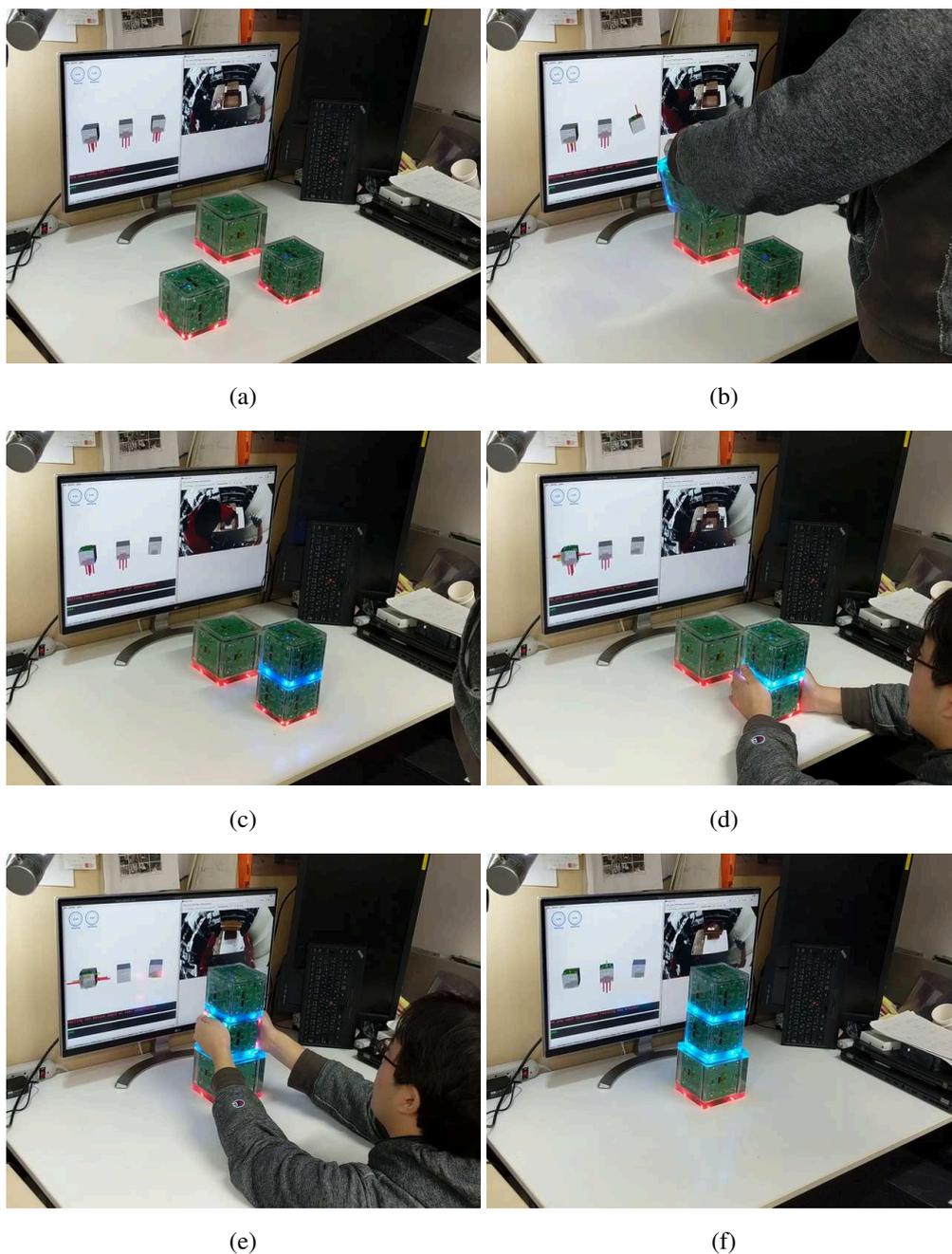


図 7.30: ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験 (操縦者側)

動作順序グラフ上で初期の対象物状態から目標の対象物状態への経路を探索することで、物体ダンボール箱収納作業の自律実行を実現した。また本実験で用いる物体ダンボール箱収納作業の自律実行のための有限状態機械は図 7.33 に示す。有限状態機械のノードとエッジの色は図 7.32 と対応しており、この有限状態機械は図 7.32 に示す赤色の左側経路の動作順序を自律実



図 7.31: ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験 (操縦者側表示画面)

行でき、かつユーザ入力に応じて操縦に切り替えて作業手順の教示学習を行うことができる有限状態機械となっている。またこの有限状態機械は操縦者による操縦実行への切替に対応するために、6.4 節で説明したように全ての対象物状態ノードは教示学習ノードである (`:teach`) に遷移することができ、操縦者の音声入力に基づいて教示学習ノードに遷移することで、自律実行から操縦による教示学習への切替を実現した。

本研究で行った教示学習結果に基づく物体ダンボール箱収納作業については図 7.34 に示す。図 7.34 に示すように、まず操縦者が 7.3.4 項で教示学習された物体ダンボール箱収納作業を自律実行することを 6.5.2.1 目で説明した音声ベース作業実行システムを用いて音声入力で指示し、ロボットが左側に置かれた把持物体を左腕で把持し持ち上げ (図 7.34b)、把持物体を中央に置かれたダンボール箱の上まで移動させ、その把持物体を離して中央に置かれたダンボール

箱の中に把持物体を収納した (図 7.34c). そしてロボットは把持物体が収納されたダンボール箱を左右から抱え込んで持ち上げ (図 7.34d, 把持物体が収納されたダンボール箱を奥にあるダンボール箱の上に積み重ね, (図 7.34e). 最後にロボットは積み重ねられたダンボール箱に対する接触を離れた (図 7.34f). 以上よりロボットが左側におかれた物体を把持して持ち上げて, それを中央手前に置かれたダンボール箱に収納したあとに, 物体を収納したそのダンボール箱を奥にあるダンボール箱の上に積み重ねる物体ダンボール箱収納作業を自律実行することができた.

本実験で自律実行された動作は以下の 8 つの動作順序であり, これは 7.3.4 項で教示学習された動作順序と同じものであり, 図 7.32 の動作順序グラフの赤色に示す左側経路上にあるエッジである.

1. (larm-top-grasp-action): 左腕で把持物体 (対象物 2) を上から把持する
2. (larm-grasp-up-action): 左腕で把持した把持物体 (対象物 2) を持ち上げる
3. (larm-grasp-pile-up-action :cube-index 0): 左腕で把持した把持物体 (対象物 2) を手前のダンボール箱 (対象物 0) の上まで移動する
4. (larm-grasp-release-action): 左腕で把持した把持物体 (対象物 2) を離す
5. (dualarm-left-right-push-hold-action): 手前のダンボール箱 (対象物 0) の右側面に対して右手で接触, 左側面に対して左手で接触する
6. (dualarm-push-hold-up-action): 手前のダンボール箱 (対象物 0) について両腕抱え込んだまま持ち上げる
7. (dualarm-push-hold-pile-up-action :cube-index 1): 手前のダンボール箱 (対象物 0) を奥のダンボール箱 (対象物 1) の上に置く
8. (dualarm-hold-release-action): 上に置かれたダンボール箱 (対象物 0) の左手と右手による接触を離す

以上の 8 つの動作順序より, 本実験では 7.3.4 項で教示学習されたように, まず把持物体を左腕で把持し持ち上げ手前のダンボール箱の上まで移動させて把持を離すことで物体を収納したのちに, ダンボール箱を両腕で左右側面を挟むことで抱え上げ, これを奥のダンボール箱の上に置き, そして両腕の接触を離すという, 7.3.4 項にて教示学習された物体ダンボール箱収納作業の作業手順を自律実行することができた.

7.4 おわりに

本章では物体操作の操縦自律間横断学習ロボットシステムの評価のために箱の回転積み重ね作業における操縦自律間横断学習実験と自律学習型把持動作を伴う物体ダンボール箱収納作業における操縦自律間横断学習実験の 2 種類の実験を双腕ロボット Baxter-JSK-SoftHand を用いて行った. 箱の回転積み重ね作業における操縦自律間横断学習実験では, 第 6 章で構築した操縦自律間切替実行システムを第 4 章で説明したタンジブルインタフェースによる操縦実行と教示学習を用いて検証し, 自律実行と操縦実行を切り替えながら実行でき, かつ作業手順を教示

学習できることを確認し、操縦自律間横断学習の有用性を確認した。また作業の自律実行の途中で操縦に切り替えて作業手順の教示学習を行うことで、切替までの自律実行結果と過去の学習結果を再利用した効率的な作業手順の教示学習を実現し、この有用性を教示学習に必要な動作順序数から評価した。そして自律学習型把持動作を伴う物体ダンボール箱収納作業における操縦自律間横断学習実験では、統合実験として第4章で説明したタンジブルインタフェースによる操縦実行と教示学習と、第5章で説明した学習型動作戦略を用いた自律学習、第6章で構築した操縦自律間切替実行システムを統合することで、本研究で提案する操縦自律間横断学習ロボットシステムが構築可能であることを示し、このシステムによる実行形態横断的な物体操作作業の学習が実現できることを確認し、操縦自律間横断学習ロボットシステムの有用性を示した。

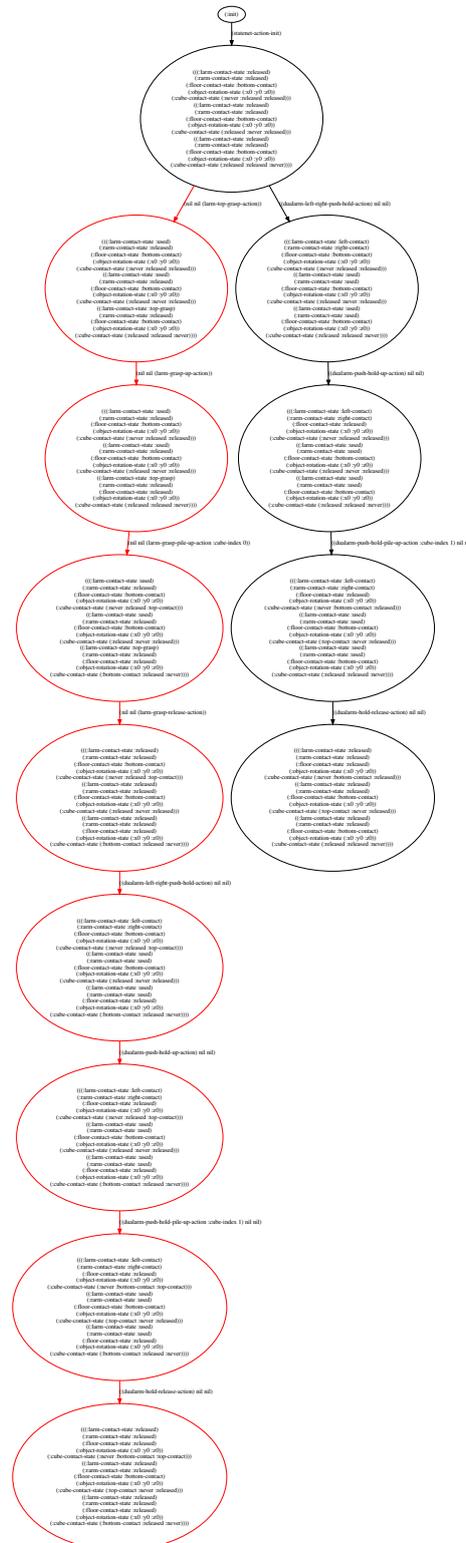


図 7.32: ダンボール積み重ね作業の自律実行時における自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順の教示学習実験で学習された動作順序グラフ: 本実験によって図 7.27 に示す動作順序グラフに対して自律学習型把持動作を伴う物体ダンボール箱収納作業の作業手順として赤色の動作順序を新たな動作順序として教示学習した。

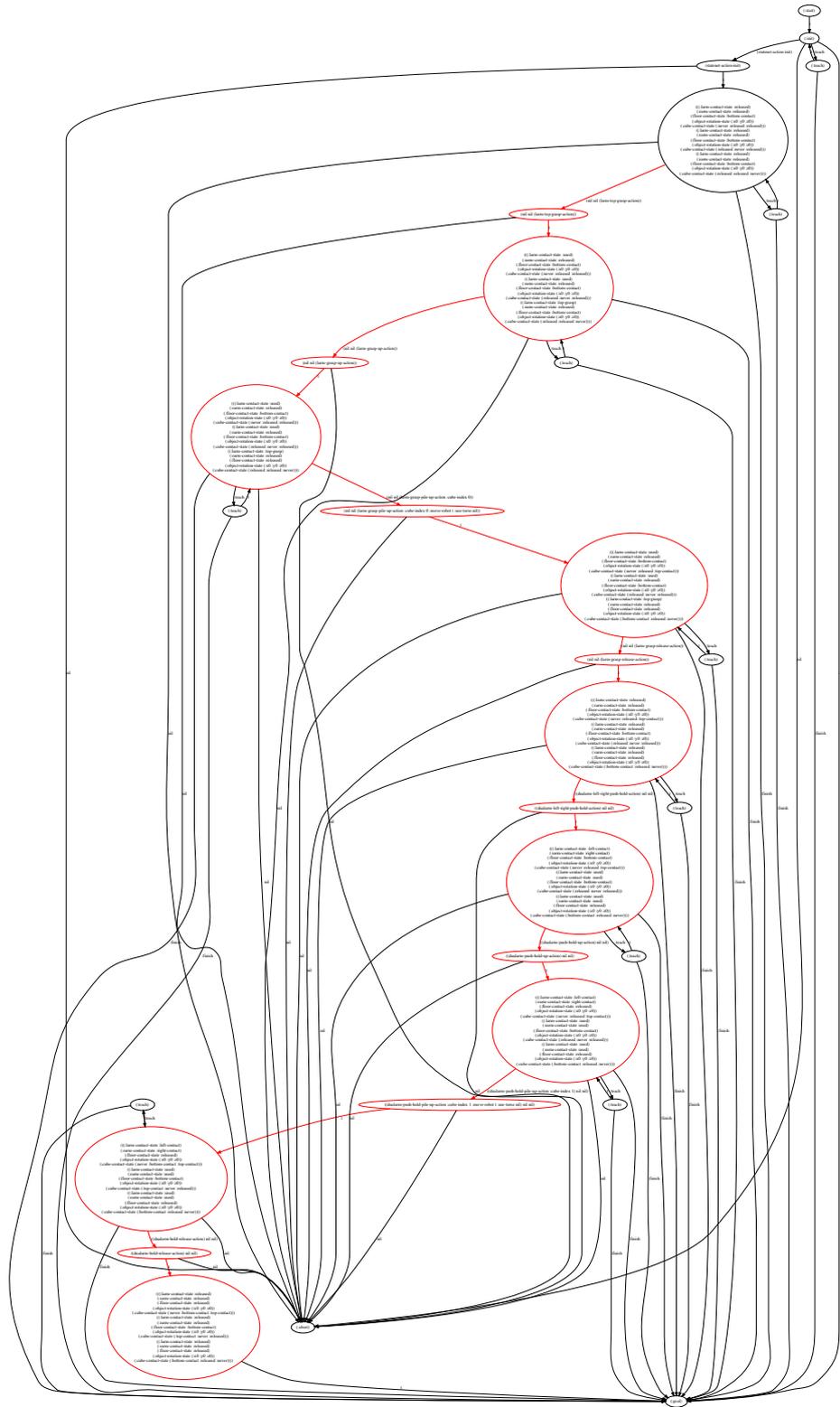


図 7.33: 自律学習型把持動作を伴う物体ダンボール箱収納作業の自律実行のための有限状態機械: ノードとエッジの色は図 7.32 と対応しており、この有限状態機械は図 7.32 に示す赤色の左側経路の動作順序を自律実行でき、かつユーザ入力に応じて操縦に切り替えて作業手順の教示学習を行うことができる有限状態機械となっている。

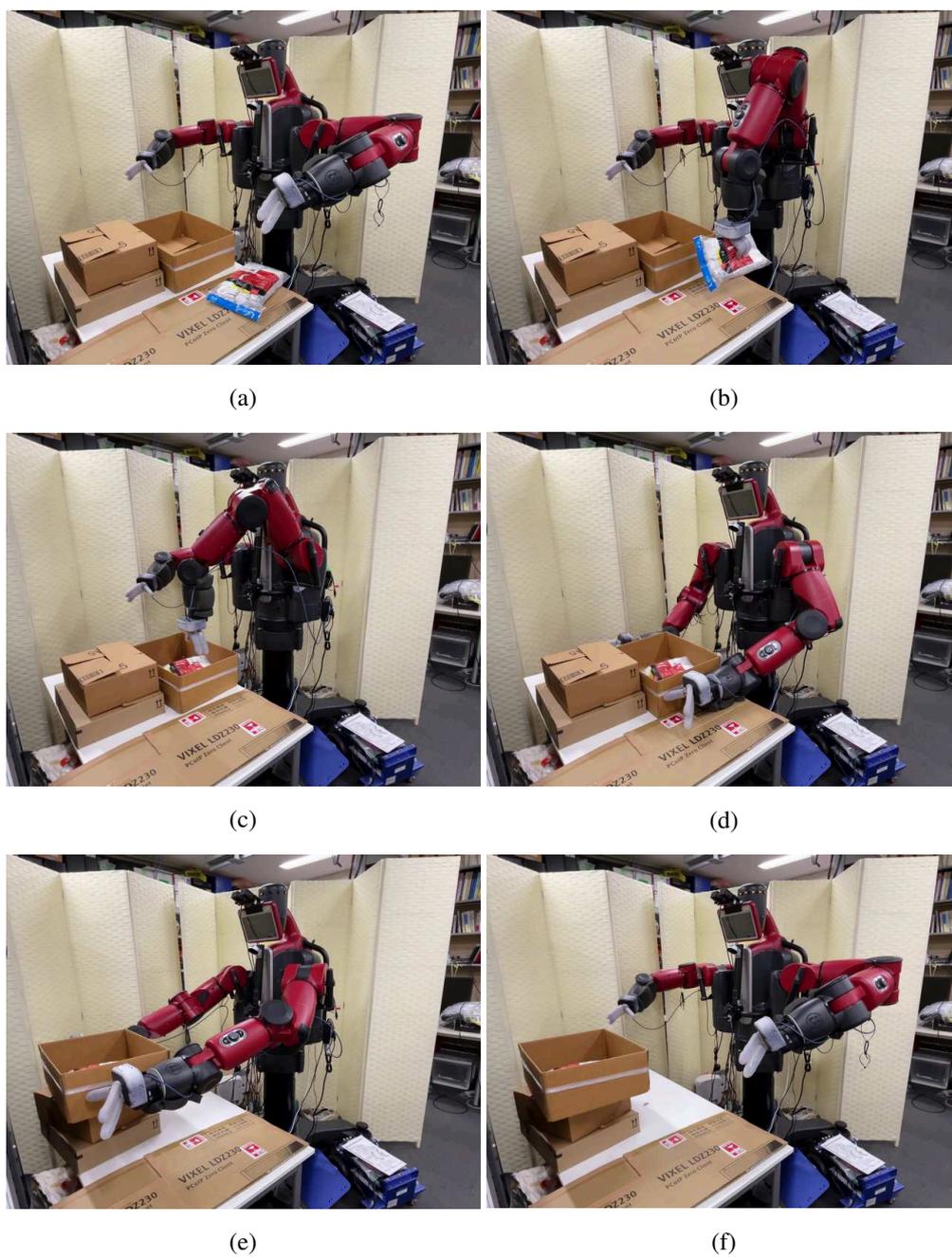


図 7.34: 教示学習結果に基づく 自律学習型把持動作を伴う 物体ダンボール箱収納作業の自律実行実験

第 8 章

結論

8.1 本研究の総括

本研究では物体操作作業のためのロボットシステムを構築する際に操縦・自律の2つの作業実行形態を切り替えながら実行でき、かつ作業に関する知識を共有することができる操縦自律間横断学習機能と操縦自律間切替実行機能をロボットシステムに持たせるためのシステム構成法を主題に取り組み、この2つの機能を有することで手順があり接触の多い複雑な物体操作作業を自律と操縦を切り替えながら横断的に学習する操縦自律間横断学習ロボットシステムを実現した。

第1章「序論」では、本研究の背景と目的を示し、本論文の構成を述べた。

第2章「物体操作作業の操縦自律間横断学習ロボットシステム」では、物体操作作業とその実行システムについて作業実行形態の観点から分類し、操縦・自律の2つの作業実行形態を切り替えながら実行する操縦自律間横断学習ロボットシステムを提案し、このシステムの全体構成と構成要素について議論した。

第3章「操縦自律間横断学習ロボットシステムのハードウェア・ソフトウェア構成」では、本研究で用いるロボットシステムのハードウェア・ソフトウェア構成について説明した。

第4章「タンジブルインタフェースによる作業手順の教示学習」では、手順があり接触の多い複雑な物体操作作業において対象物の接触・回転状態を表現するタンジブルインタフェースを用いて操縦実行し、かつこのインタフェースを用いて作業手順を表す動作順序を教示学習することを提案し、長所・短所についてこれまでのユーザインタフェースと比較することで議論した。さらに、操縦のためのタンジブルインタフェースとして *Miniature Tangible Cube* を実装し、これを用いた操縦実行手法と教示手法有効性を複数対象物が接触しあう物体操作作業を行うことで示した。

第5章「実世界適応による動作戦略の自律学習」では、接触の多い複雑な動作の動作を制御する動作戦略を自律学習するための学習型動作戦略の構成と、その実世界適応による自律学習手法について議論し、CNN 失敗予測器、CNN 把持点予測器を用いた学習型動作戦略手法とシミュレーションと実世界での動作試行を用いた動作戦略の実世界適応手法を提案し、これらの手法の有効性について棚への双腕収納作業と倉庫環境での双腕ピッキング作業を行うことで示

した。

第6章「物体操作作業の操縦自律間切替実行システム」では、操縦自律間横断学習ロボットシステムを実現するための操縦自律間切替実行システムとこれを構成する各システムについて議論し、これらの各システムの有効性について物体操作作業の操縦実行と自律実行を行うことで示した。

第7章「物体操作作業の操縦自律間横断学習ロボットシステムの評価」では、本研究で提案する操縦自律間横断学習ロボットシステムの有用性について、操縦自律間における切替実行と横断学習の観点から評価を行い、本ロボットシステムを用いることで作業手順を表す動作順序と動作を制御する動作戦略を共有しながら操縦と自律を切り替えて複雑な物体操作作業を実行できることを実証し、自律実行の操縦に切り替えることでその場で作業手順を教示学習できるという即応性と、かつ切替までの自律実行結果と過去の学習結果を再利用して学習するという効率性を兼ね備えた作業手順の教示学習を実現・評価し、本研究の提案ロボットシステムの有用性を示した。

8.2 本研究の成果

本研究では対象物の接触・回転状態に基づいて作業を動作順序と動作に分割することで、タンジブルインタフェースを用いた作業手順を表す動作順序の教示学習手法と実世界適応を用いた動作を制御する動作戦略の自律学習手法を提案し、操縦自律実行システムを用いて上記の2つの手法を統合することで操縦自律実行機能と操縦自律学習機能を有する操縦自律間横断学習ロボットシステムの構成法を示した。以下では本研究の成果を、タンジブルインタフェースによる作業手順の教示学習と実世界適応による動作戦略の自律学習、操縦自律間切替実行システムのそれぞれについて説明する。

タンジブルインタフェースによる作業手順の教示学習

本研究のタンジブルインタフェースによる作業手順の教示学習における成果を以下に示す。

- タンジブルインタフェースを用いた対象物指向型操縦手法
- 対象物の接触・回転状態を表現するタンジブルインタフェースの実装手法
- タンジブルインタフェースを用いた作業手順を表す動作順序の教示学習手法
- タンジブルインタフェースを用いた物体操作作業の操縦実行・教示の実現

本研究では物体操作作業の操縦実行と教示学習において、対象物を中心として対象物・ロボット・環境の間の接触と回転を対象物状態として表現できるタンジブルインタフェースを用いることを提案し、これを用いた対象物指向型操縦手法を構築した。さらにこのタンジブルインタフェースを近接センサとIMUを用いて実装する手法を構築し、実装したタンジブルデバイスである Miniature Tangible Cube を用いて単一、複数、複雑形状物体を用いた物体操作作業の操縦実行実験を行い、タンジブルインタフェースを用いた操縦実行手法の有効性を示した。ま

たこのタンジブルインタフェースを用いた操縦実行において作業手順を表す動作順序を教示学習することを提案し、複数対象物が接触しあう物体操作作業の教示学習実験を行うことで、このタンジブルインタフェースによる作業手順の教示学習手法の有効性を示した。

実世界適応による動作戦略の自律学習

本研究の実世界適応による動作戦略の自律学習における成果を以下に示す。

- 学習型動作戦略のシミュレーションと実世界試行を用いた自律学習手法
- 学習型動作戦略を用いた物体操作作業の自律実行の実現

本研究では物体操作作業の自律実行と自律学習において、シミュレーションによる事前学習と実世界試行による再学習を組み合わせた動作戦略の実世界適応による自律学習手法を提案し、動作戦略を学習型動作戦略である CNN 把持点予測器として実装し、この CNN 把持点予測器を用いた倉庫環境における双腕ピッキング作業実験を行うことで、この実世界適応による動作戦略の自律学習手法の有効性を示した。

操縦自律間切替実行システム

本研究の操縦自律間切替実行システムにおける成果を以下に示す。

- ユーザ入力に応じて操縦と自律を切り替えながら実行する操縦自律間切替実行手法
- 操縦と自律の間で作業の動作順序と動作戦略を共有する操縦自律間横断学習の実現

本研究では操縦実行と自律実行を切り替えながら実行する操縦自律間切替実行において、音声や GUI などを通じたユーザ入力に応じて操縦と自律を切り替えながら、作業手順を表す動作順序と動作を制御する動作戦略を共有することができる操縦自律間切替実行システムを構築した。この操縦自律間切替実行システムとタンジブルインタフェース、学習型動作戦略を用いて、タンジブルインタフェースによってその場で作業手順を教示学習でき、かつ自律実行に共有して実行できることを箱の回転積み重ね作業切替実行実験で示し、実世界適応によって自律学習された動作戦略が自律実行でき、かつ操縦実行に共有して実行できることを物体ダンボール箱収納作業実験において示した。さらに作業の自律実行の途中で操縦に切り替えて作業手順の教示学習を行うことで、過去の実行・学習結果を再利用した即応的かつ効率的な作業手順の教示学習を実現し本研究で提案する操縦自律間横断学習ロボットシステムの有用性を示した。

8.3 本研究の結論

本研究では、操縦・自律の2つの異なる作業実行形態を切り替えながら、作業手順と動作を横断的に学習する物体操作作業のロボットシステムの構築を目的とし、対象物の接触・回転状態に着目して作業手順を表す動作順序と動作に作業を分割することで、操縦と自律の間を切り替えて実行しながら作業の知識を横断的に学習する操縦自律間横断学習ロボットシステムの構

成法を示した．そしてこの操縦自律間横断学習ロボットシステムの構成要素としてタンジブルインタフェースによる作業手順を表す動作順序の教示学習手法，実世界適応による動作を制御する動作戦略の自律学習手法を提案し，上記の2つの手法を統合する操縦自律間切替実行システムを構築することで，ユーザ入力に応じて操縦自律間切替実行しながら，操縦と自律の間で作業手順と動作戦略を横断的に学習し，かつ自律から操縦への切り替えることによって即応的かつ効率的に作業手順を教示学習することを実現し，実世界での手順があり接触の多い物体操作作業へ適用することで，提案システムの有用性を評価した．以上のことから本研究により，タンジブルインタフェースによる物体操作作業の操縦実行手法と教示学習手法，実世界適応による動作戦略の自律学習手法が確立され，操縦と自律の2つの作業実行形態を切り替えながら操縦自律間で作業手順と動作を横断的に学習できる物体操作ロボットシステムの構成法が明らかにされたと結論付けられる．

8.4 今後の展開

本研究では，操縦・自律の2つの異なる作業実行形態を切り替えながら実行し，作業手順を表す動作順序と動作を制御する動作戦略を横断的に学習できる操縦自律間横断学習ロボットシステムの構成法を明らかにし，このシステムが実現可能であり，かつこのシステムを用いて操縦自律間横断学習しながら手順があり接触の多い複雑な物体操作作業が実行可能性，作業の自律実行の途中で操縦に切り替えることによる作業手順の教示学習の即応性と効率性について実世界での物体操作作業実験において実証した．本研究は操縦自律間横断学習ロボットシステムの実現可能性，手順があり接触の多い複雑な物体作業の操縦自律間横断学習の実現可能性，操縦自律間横断学習による作業手順の変更の即応性と効率性を評価したものであり，統合実験において評価を行った作業手順の変更のシチュエーションは新たな作業への対応のみであり，このシステムによって失敗などのどの程度の状況において作業手順の変更が可能となっているかについては十分な定量評価が行えておらず，より多様な状況における作業手順の変更やその定量評価が今後の課題としてあげられる．今後はより作業手順の変更が必要となる複雑な作業に対して，失敗や作業の変更，人の介入などに対して操縦によってどの程度頑強に対応できるか，そしてその対応を操縦・自律間を切り替えながら行うことでどの程度速く簡単に行えるか，といったシステムの対応性・適応性について評価をしていく予定である．

本研究では作業手順の操縦による教示学習においてタンジブルインタフェースを用いた教示学習手法を提案し，この手法を用いてタンジブルインタフェースで操縦実行された動作順序を作業手順として，操縦経験に基づいて教示学習することを実現した．この手法では動作順序を対象物の接触・回転状態をノード，ロボットの動作をエッジとする有向グラフで表現し，自律実行時には目標状態への経路を探索することで作業実行を行ったが，同じ作業でも操縦者が異なる動作順序で行う可能性が考えられる．例えばこれまで先に物体を把持していたが先に箱を回転させるなど，作業の目標は同じであるが動作順序の変更を操縦者が求める可能性が考えられるため，確率的な表現が可能な有向グラフを用いて動作順序を表現することが今後の展開の1つとして考えられる．また確率的な表現が可能になることで，他の作業で学習された動作順

序に基づいて新たな作業を自律実行の際に最も操縦者が求めているであろう経路を探索することも可能であると考え、よって今後は動作順序を確率的な表現ができる有向グラフで表現することで、新たな作業の自律実行や同じ作業における動作順序の切り替えに対して最も適している動作順序を選択・実行することに取り組む予定である。

本研究では動作戦略の自律学習において学習型動作戦略の実世界適応手法を提案し、この手法を用いて CNN 把持点予測器を実世界での把持試行経験に基づいて自律学習することを実現した。この実世界適応手法はシミュレーションと実世界試行を組み合わせることで少ないデータ数で学習を行える手法であるが、本研究では把持動作についてしか自律学習が行われておらず、また把持試行については作業とはオフラインに自律学習を行っているため、作業を行いながらオンラインで学習することは実現できていない。また今回は把持動作の自律学習を行ったが箱の蓋をあけるなど更に複雑な動作が必要となる作業では自己教師あり学習では先に初期動作を作り込まなければ動作の探索範囲が小さくなってしまうため自律学習が困難となってしまう。しかし本システムは作業を動作順序と動作に分割しているため動作は1つのモジュールとして取り替え可能であり、把持動作以外の動作について人の操縦データに基づく模倣学習や、強化学習を用いた学習型動作戦略に取り替えることは容易であると考え。また実世界適応手法で用いたシミュレーションと実世界試行の組み合わせは把持動作以外の動作やオンライン学習でも用いることができ、模倣学習モデルや強化学習モデルに対してシミュレーションを用いて事前学習を行い、作業中に作業を構成する様々な動作をそれぞれオンライン学習することは可能であると考え。よって今後は本研究で取り組んだ実世界適応手法を把持動作以外の動作とオンライン学習に拡張することで、各動作の動作戦略が作業を行っている最中でも学習され、複雑な物体操作作業において接触の多い動作を自律的にオンラインで学習することに取り組む予定である。

発表文献と研究活動

筆頭論文 *1

査読付国際学術雑誌論文

1. **Shingo Kitagawa**, Kentaro Wada, Shun Hasegawa, Kei Okada, Masayuki Inaba: Few-experiential learning system of robotic picking task with selective dual-arm grasping, *Advanced Robotics*, Vol.34, No.18, pp.1171-1189, 2020.
2. **Shingo Kitagawa**, Shun Hasegawa, Naoya Yamaguchi, Kei Okada, Masayuki Inaba: Miniature Tangible Cube: Concept and Design of Target-Object-Oriented User Interface for Dual-Arm Telemanipulation, *IEEE Robotics and Automation Letters*, Vol.6, No.4, pp.6977-6984, 2021.

査読付国際学術会議論文

3. **Shingo Kitagawa**, Kentaro Wada, Shun Hasegawa, Kei Okada, Masayuki Inaba: Multi-stage Learning of Selective Dual-arm Grasping Based on Obtaining and Pruning Grasping Points Through the Robot Experience in the Real World, in *Proceedings of The 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.7123-7130, 2018.
4. **Shingo Kitagawa**, Kentaro Wada, Kei Okada, Masayuki Inaba: Learning-Based Task Failure Prediction for Selective Dual-Arm Manipulation in Warehouse Stowing, in Marcus Strand and Rudiger Dillmann and Emanuele Menegatti and Stefano Ghidoni (Eds.): *Intelligent Autonomous Systems 15*, pp.428-439, Springer International Publishing, 2019.

国内学術会議論文

5. 北川 晋吾, 和田 健太郎, 岡田 慧, 稲葉 雅幸: 作業失敗の予測学習に基づく棚への収納作業における双腕手添え動作の選択的実行, in 第 31 回人工知能学会全国大会論文集,

*1 太文字: 筆頭著者・筆頭貢献者, 下線: 本論文筆者

- 2N2-4, 2017.
6. 北川 晋吾, 和田 健太郎, 岡田 慧, 稲葉 雅幸: Learning-based Selective Dual-arm Grasping for Warehouse Picking, in 第 32 回人工知能学会全国大会論文集, 1G3-03, 2018.
 7. 北川 晋吾, 岡田 慧, 稲葉 雅幸: 物体インスタンスの重なりを考慮した双腕ロボットによる自律学習型ターゲットピッキングシステム, in 第 33 回人工知能学会全国大会論文集, 1L3-J-11-05, 2019.
 8. 北川 晋吾, 岡田 慧, 稲葉 雅幸: 双腕組み立て作業の遠隔操縦のためのタンジブルユーザインターフェースデバイスの設計と開発, in 第 38 回日本ロボット学会学術講演会講演論文集, 2C3-04, 2020.
 9. 北川 晋吾, 廣瀬 俊典, 新城 光樹, 岡田 慧, 稲葉 雅幸: ソフトハンドを有するロボットの視聴覚統合型遠隔マニピュレーションシステムの開発, in 第 21 回計測自動制御学会システムインテグレーション部門講演会講演概要集, 3G2-07, 2020.

共著論文 *¹

査読付国際学術会議論文

10. **Kentaro Wada**, Shingo Kitagawa, Kei Okada, Masayuki Inaba: Instance Segmentation of Visible and Occluded Regions for Finding and Picking Target from a Pile of Objects, in Proceedings of The 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2048-2055, 2018.
11. **Shun Hasegawa***, **Kentaro Wada***, Shingo Kitagawa, Yuto Uchimi, Kei Okada, Masayuki Inaba (* equal contributions): GraspFusion: Realizing Complex Motion by Learning and Fusing Grasp Modalities with Instance Segmentation, in Proceedings of The 2019 International Conference on Robotics and Automation, pp.7235-7241, 2019.
12. **Ayaka Fujii**, Kanae Kochigami, Shingo Kitagawa, Kei Okada, Masayuki Inaba: Development and Evaluation of Mixed Reality Co-eating System: Sharing Behavior of Eating Food with a Robot Could Improve Our Dining Experience, in Proceedings of The 29th IEEE International Conference on Robot & Human Interactive Communication, pp.357-362, 2020.

国内学術会議論文

13. 二井谷 勇佑, 和田 健太郎, 長谷川 峻, 北川 晋吾, 板東 正祐, 岡田 慧, 稲葉 雅幸: 物品棚のピッキング作業のための深層学習でのセマンティック画像分割と三次元物品輪郭抽出, in 第 34 回日本ロボット学会学術講演会講演論文集, 2G2-03, 2016.
14. 和田 健太郎, 北川 晋吾, 岡田 慧, 稲葉 雅幸: Instance Segmentation of Visible and Occluded Regions for Finding and Picking Target from a Pile of Objects, in 第 32 回人工知能学会全国大会論文集, 4A2-02, 2018.

15. 内海 佑斗, 北川 晋吾, 矢野倉 伊織, 岡田 慧, 稲葉 雅幸: CNN による深度予測に基づく鏡を活用した死角の物体知覚, in 第 33 回人工知能学会全国大会論文集, 1L2-J-11-02, 2019.
16. 東風上 奏絵, 北川 晋吾, 岡田 慧, 稲葉 雅幸: 手繋ぎ誘導を通したロボットへの場所教示・移動経験記憶と移動時の周囲の人とのインタラクション評価, in 第 20 回計測自動制御学会システムインテグレーション部門講演会講演概要集, 1E1-05, 2019.
17. 藤井 綺香, 東風上 奏絵, 北川 晋吾, 岡田 慧, 稲葉 雅幸: ロボットとの共食による食体験の向上を目指した複合現実システムの開発とその評価, in 日本機械学会ロボティクス・メカトロニクス講演会 2020 講演論文集, 1P2-E08, 2020.
18. 金沢 直晃, 北川 晋吾, 室岡 貴之, 岡田 慧, 稲葉 雅幸: 調理道具を扱う双腕ロボットによる野菜皮剥き切断操作の認識行動実現システム, in 第 21 回計測自動制御学会システムインテグレーション部門講演会講演概要集, 3E2-08, 2020.
19. 若林 隼平, 北川 晋吾, 河原塚 健人, 室岡 貴之, 岡田 慧, 稲葉 雅幸: 視覚情報に基づく食器類の把持の冗長性を考慮した自己教師あり把持学習, in 日本機械学会ロボティクス・メカトロニクス講演会 2021 講演論文集, 1A1-F09, 2021.
20. 大日方 慶樹, 山口 直也, 北川 晋吾, 岡田 慧, 稲葉 雅幸: スマート IoT 機器を用いたビル設備操作とサービスロボットおつかい行動の協調システム, in 日本機械学会ロボティクス・メカトロニクス講演会 2021 講演論文集, 2A1-H02, 2021.
21. 金沢 直晃, 北川 晋吾, 室岡 貴之, 岡田 慧, 稲葉 雅幸: テンプレートマッチング認識とプリミティブ動作により状態を考慮した家電操作を行うロボットシステム, in 第 22 回計測自動制御学会システムインテグレーション部門講演会講演概要集, 1G2-02, 2021.
22. 新城 光樹, 北川 晋吾, 山口 直也, 大日方 慶樹, 岡田 慧, 稲葉 雅幸: 自らタスクを御用聞きする四脚ロボットの配送システムの実現, in 第 22 回計測自動制御学会システムインテグレーション部門講演会講演概要集, 3E4-12, 2021.

受賞

23. **Kentaro Wada**, Shingo Kitagawa, Kei Okada, Masayuki Inaba: Instance Segmentation of Visible and Occluded Regions for Finding and Picking Target from a Pile of Objects, IEEE Robotics and Automation Society (RAS) Japan Joint Chapter Young Award (in IROS2018), 2018.
24. **Shun Hasegawa***, **Kentaro Wada***, Shingo Kitagawa, Yuto Uchimi, Kei Okada, Masayuki Inaba (* equal contributions): GraspFusion: Realizing Complex Motion by Learning and Fusing Grasp Modalities with Instance Segmentation, IEEE Robotics and Automation Society (RAS) Japan Joint Chapter Young Award (in ICRA2019), 2019.
25. 内海 佑斗, 北川 晋吾, 矢野倉 伊織, 岡田 慧, 稲葉 雅幸: CNN による深度予測に基づく鏡を活用した死角の物体知覚, 人工知能学会全国大会学生奨励賞, 2019.
26. 東風上 奏絵, 北川 晋吾, 岡田 慧, 稲葉 雅幸: 手繋ぎ誘導を通したロボットへの場所教示・

移動経験記憶と移動時の周囲の人とのインタラクション評価, 第 20 回計測自動制御学会システムインテグレーション部門講演会(SI2019) 優秀講演賞, 2019.

27. 金沢 直晃, 北川 晋吾, 室岡 貴之, 岡田 慧, 稲葉 雅幸: テンプレートマッチング認識とプリミティブ動作により状態を考慮した家電操作を行うロボットシステム, 第 22 回計測自動制御学会システムインテグレーション部門講演会(SI2021) 優秀講演賞, 2021.
28. 新城 光樹, 北川 晋吾, 山口 直也, 大日方 慶樹, 岡田 慧, 稲葉 雅幸: 自らタスクを御用聞きする四脚ロボットの配送システムの実現, 第 22 回計測自動制御学会システムインテグレーション部門講演会(SI2021) 優秀講演賞, 2021.

国際競技大会 *1

29. **Shingo Kitagawa***, **Shun Hasegawa***, **Yusuke Niitani***, **Kentaro Wada***, Masahiro Bando, Kei Okada, Masayuki Inaba (* equal contributions): Amazon Picking Challenge 2016, in Robocup 2016, 5th place in 16 teams in pick task and 8th place in 16 teams in stow task, 2016.
30. **Shingo Kitagawa***, **Shun Hasegawa***, **Kentaro Wada***, Yuto Uchimi, Yusuke Niitani, Naoya Yamaguchi, Kei Okada, Masayuki Inaba (* equal contributions): Amazon Robotics Challenge 2017, in Robocup 2017, 12th place in 16 teams in pick task and 13th place in 16 teams in stow task, 2017.

参考文献

- [1] K. Takeyasu T. Goto and T. Inoyama. Control algorithm for precision insert operation robots. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 10, No. 1, pp. 19–25, 1980.
- [2] H. Ueno K. Hayashi and H. Murakami. 精密仕上げロボットシステムの開発. 計測自動制御学会論文集, Vol. 51, No. 1, pp. 32–40, 2015.
- [3] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, Vol. 15, No. 1, pp. 172–188, Jan 2018.
- [4] K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato, and M. Inaba. Vision based behavior verification system of humanoid robot for daily environment tasks. In *Proceedings of The 2006 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 7–12, 2006.
- [5] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth. Robotic roommates making pancakes. In *Proceedings of The 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 529–536, 2011.
- [6] L. Kunze, M. Beetz, M. Saito, H. Azuma, K. Okada, and M. Inaba. Searching objects in large-scale indoor environments: A decision-theoretic approach. In *Proceedings of The 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4385–4390, 2012.
- [7] S. Nozawa, M. Murooka, S. Noda, K. Okada, and M. Inaba. Description and execution of humanoid’s object manipulation based on object-environment-robot contact states. In *Proceedings of The 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2608–2615. IEEE/RSJ, Nov 2013.
- [8] M. Murooka, S. Noda, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Manipulation strategy decision and execution based on strategy proving operation for carrying large and heavy objects. In *Proceedings of The 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3425–3432. IEEE, May 2014.
- [9] K. Harada, T. Foissotte, T. Tsuji, K. Nagata, N. Yamanobe, A. Nakamura, and Y. Kawai.

- Pick and place planning for dual-arm manipulators. In *Proceedings of The 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2281–2286. IEEE, 2012.
- [10] I. Rodriguez, K. Nottensteiner, D. Leidner, M. Kassecker, F. Stulp, and A. Albu-Schaffer. Iteratively refined feasibility checks in robotic assembly sequence planning. *IEEE Robotics and Automation Letters*, Vol. 4, No. 2, pp. 1416–1423, 2019.
- [11] P. Schmaus, D. Leidner, T. Kruger, R. Bayer, B. Pleintinger, A. Schiele, and N. Y. Lii. Knowledge driven orbit-to-ground teleoperation of a robot coworker. *IEEE Robotics and Automation Letters*, Vol. 5, No. 1, pp. 143–150, 2020.
- [12] S. Hayati, T.S. Lee, K.S. Tso, P.G. Backes, and J. Lloyd. A unified teleoperated-autonomous dual-arm robotic system. *IEEE Control Systems Magazine*, Vol. 11, No. 2, pp. 3–8, 1991.
- [13] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, Vol. 37, No. 4-5, pp. 421–436, 2018.
- [14] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase. Development of human support robot as the research platform of a domestic mobile manipulator. *ROBOMECH journal*, Vol. 6, No. 1, pp. 1–15, 2019.
- [15] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich. Fetch and freight: Standard platforms for service robot applications. In *Workshop on autonomous mobile service robots*, 2016.
- [16] J. Pages, L. Marchionni, and F. Ferro. Tiago: the modular robot that adapts to different research needs. In *International workshop on robot modularity, IROS*, 2016.
- [17] Rethink Robotics. Baxter Collaborative Robots for Industrial Automation. <http://www.rethinkrobotics.com/baxter>, 2019.
- [18] S. Cremer, L. Mastromoro, and D. O. Popa. On the performance of the baxter research robot. In *Proceedings of 2016 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, pp. 106–111, 2016.
- [19] J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Msenlechner, W. Meeussen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the pr2. In *Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5568–5575, 2011.
- [20] ABB. ABB’s Dual-Arm Collaborative Robot - Industrial Robots From ABB Robotics. <https://new.abb.com/products/robotics/industrial-robots/yumi>, 2021.
- [21] Kawasaki Heavy Industries. Dual Arm SCARA Robot - Kawasaki Robotics. <https://robotics.kawasaki.com/en1/products/robots/dual-arm-scara/>, 2021.
- [22] K. Kaneko, M. Morisawa, S. Kajita, S. Nakaoka, T. Sakaguchi, R. Cisneros, and F. Kane-

- hiro. Humanoid robot hrp-2kai—improvement of hrp-2 towards disaster response tasks. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 132–139. IEEE, 2015.
- [23] K. Kojima, T. Karasawa, T. Kozuki, E. Kuroiwa, S. Yukizaki, S. Iwaishi, T. Ishikawa, R. Koyama, S. Noda, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Development of life-sized high-power humanoid robot jaxon for real-world use. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 838–843. IEEE, 2015.
- [24] K. Kaneko, H. Kaminaga, T. Sakaguchi, S. Kajita, M. Morisawa, I. Kumagai, and F. Kanehiro. Humanoid robot hrp-5p: An electrically actuated humanoid robot with high-power and wide-range joints. *IEEE Robotics and Automation Letters*, Vol. 4, No. 2, pp. 1431–1438, 2019.
- [25] D. Fischinger, M. Vincze, and Y. Jiang. Learning grasps for unknown objects in cluttered scenes. In *Proceedings of The 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 609–616. IEEE, May 2013.
- [26] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger, M. Morariu, J. Ju, X. Gerrmann, R. Ensing, J. Van Frankenhuyzen, and M. Wisse. Team delft’s robot winner of the amazon picking challenge 2016. In *RoboCup 2016: Robot World Cup XX*, pp. 613–624, Cham, 2017. Springer International Publishing.
- [27] A. Edsinger and C. C. Kemp. Two arms are better than one: A behavior based control system for assistive bimanual manipulation. In *Recent progress in robotics: Viable robotic service to human*, pp. 345–355. Springer, 2007.
- [28] K. Kosuge, H. Kakuya, and Y. Hirata. Control algorithm of dual arms mobile robot for cooperative works with human. In *Proceedings of 2001 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5, pp. 3223–3228. IEEE, 2001.
- [29] W. R. Ferrell and T. B. Sheridan. Supervisory control of remote manipulation. *IEEE Spectrum*, Vol. 4, No. 10, pp. 81–88, 1967.
- [30] S. Hirai. Shared autonomy の理論. *日本ロボット学会誌*, Vol. 11, No. 6, pp. 788–793, 1993.
- [31] M. Zeestraten, I. Havoutis, and S. Calinon. Programming by demonstration for shared control with an application in teleoperation. *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, pp. 1848–1855, 2018.
- [32] S. Tachi, K. Komoriya, K. Sawada, T. Nishiyama, T. Itoko, M. Kobayashi, and K. Inoue. Telexistence cockpit for humanoid robot control. *Advanced Robotics*, Vol. 17, pp. 199–217, 2003.
- [33] J. Kofman, W. Xianghai, T. J. Luu, and S. Verma. Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Transactions on Industrial Electronics*,

- Vol. 52, No. 5, pp. 1206–1219, 2005.
- [34] H. Ishii. Tangible bits: Beyond pixels. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*, pp. 15–25. ACM, 2008.
- [35] S. Kitagawa, K. Wada, S. Hasegawa, K. Okada, and M. Inaba. Few-experiential learning system of robotic picking task with selective dual-arm grasping. *Advanced Robotics*, Vol. 34, No. 18, pp. 1171–1189, 2020.
- [36] S. Kitagawa, K. Wada, S. Hasegawa, K. Okada, and M. Inaba. Multi-stage learning of selective dual-arm grasping based on obtaining and pruning grasping points through the robot experience in the real world. In *Proceedings of The 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7123–7130. IEEE/RSJ, october 2018.
- [37] S. Kitagawa, S. Hasegawa, N. Yamaguchi, K. Okada, and M. Inaba. Miniature tangible cube: Concept and design of target-object-oriented user interface for dual-arm telemanipulation. *IEEE Robotics and Automation Letters*, Vol. 6, No. 4, pp. 6977–6984, 2021.
- [38] P-C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata. Repeatable folding task by humanoid robot worker using deep learning. *IEEE Robotics and Automation Letters*, Vol. 2, No. 2, pp. 397–403, 2017.
- [39] H. Kim, Y. Ohmura, and Y. Kuniyoshi. Gaze-based dual resolution deep imitation learning for high-precision dexterous robot manipulation. *IEEE Robotics and Automation Letters*, Vol. 6, No. 2, pp. 1630–1637, 2021.
- [40] C. C. Kemp, C. D. Anderson, H. Nguyen, A. J. Trevor, and Z. Xu. A point-and-click interface for the real world: Laser designation of objects for mobile manipulation. In *Proceedings of 2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 241–248, 2008.
- [41] S. Hashimoto, A. Ostanin, M. Inami, and T. Igarashi. Photograph-based interaction for teaching object delivery tasks to robots. In *Proceedings of 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 153–154, 2010.
- [42] K. Ishii, Y. Takeoka, M. Inami, and T. Igarashi. Drag-and-drop interface for registration-free object delivery. In *Proceedings of 19th International Symposium in Robot and Human Interactive Communication*, pp. 228–233, 2010.
- [43] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *Proceedings of The 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3406–3413. IEEE, May 2016.
- [44] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan. Multi-task domain adaptation for deep learning of instance grasping from simulation. In *Proceedings of The 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3516–3523, May 2018.
- [45] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser. Tossingbot: Learning to throw

- arbitrary objects with residual physics. *IEEE Transactions on Robotics*, Vol. 36, No. 4, pp. 1307–1319, 2020.
- [46] S. Ha, J. Kim, and K. Yamane. Automated deep reinforcement learning environment for hardware of a modular legged robot. In *Proceedings of 2018 15th International Conference on Ubiquitous Robots (UR)*, pp. 348–354, 2018.
- [47] 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.
- [48] A. Saxena, L. S. Wong, and A. Y. Ng. Learning grasp strategies with partial shape information. In *Proceedings of Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 3, pp. 1491–1494, 2008.
- [49] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, Vol. 36, No. 13-14, pp. 1455–1473, 2017.
- [50] Open Source Robotics Foundation. ROS. <http://wiki.ros.org>, 2021.
- [51] S. Hasegawa, K. Wada, Y. Niitani, K. Okada, and M. Inaba. A three-fingered hand with a suction gripping system for picking various objects in cluttered narrow space. In *Proceedings of The 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1164–1171. IEEE/RSJ, September 2017.
- [52] T. Hirose, Y. Kakiuchi, K. Okada, and M. Inaba. Design of Soft Flexible Wire-driven Finger Mechanism for Contact Pressure Distribution. *IEEE International Conference on Intelligent Robots and Systems*, pp. 4699–4705, 2019.
- [53] Rethink Robotics. Hardware Specifications. http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications, 2021.
- [54] Michael Ferguson. roserial. <http://wiki.ros.org/roserial>, 2021.
- [55] Orbbec 3D. Astra-Orbbec. <https://orbbec3d.com/product-astra/>, 2021.
- [56] Intel Corporation. LiDAR Camera L515 Intel RealSense Depth and Tracking Cameras. <https://www.intelrealsense.com/lidar-camera-l515/>, 2021.
- [57] JK Imaging Ltd. Kodak Digital Cameras — VR Camera SP360 4K. <https://kodakpixpro.com/AsiaOceania/jp/cameras/vrcamera/sp3604k/>, 2021.
- [58] Elpcctv.com. 1/2.5 SONY IMX317 High resolution 4K USB camera wide angle with fisheye lens ELP-USB4KHDR01-BL170 [ELP-USB4KHDR01-BL170]. <http://www.elpcctv.com/125-sony-imx317-high-resolution-4k-usb-camera-wide-angle-with-fisheye-lens-elpusb4khdr01bl170-p-177.html>, 2021.
- [59] Seeed Technology Co.,Ltd. ReSpeaker Mic Array v2.0 - Seeed Wiki. https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/, 2021.
- [60] Willow Garage. PR2 User Manual. <https://www.clearpathrobotics.com/>

- assets/downloads/pr2/pr2_manual_r321.pdf, 2012.
- [61] Y. Furuta. 長期経験に基づくタスク具体化機能を備えた日常生活支援ロボットシステムの構成法. PhD thesis, The University of Tokyo, 2019.
- [62] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, Vol. 19, No. 2, pp. 4–10, 2012.
- [63] 北陽電機株式会社. UTM-30LX. <https://www.hokuyo-aut.co.jp/search/single.php?serial=21>, 2021.
- [64] Fetch Robotics. Robot Hardware Overview. https://docs.fetchrobotics.com/robot_hardware.html, 2019.
- [65] SICK AG. TiM571-2050101. <https://www.sick.com/jp/en/detection-and-ranging-solutions/2d-lidar-sensors/tim5xx/tim571-2050101/p/p412444>, 2021.
- [66] ARTCREATION 3D TECHNOLOGY LIMITED. PRIMESENSE CARMINE 1.09. <http://xtionprolive.com/primesense-carmine-1.09>, 2021.
- [67] T. Matsui and M. Inaba. EusLisp: an object-based Implementation of Lisp. *Journal of Information Processing*, Vol. 13, No. 3, pp. 327–338, 1990.
- [68] D. Coleman, I. A. Sukan, S. Chitta, and N. Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *CoRR*, Vol. abs/1404.3785, , 2014.
- [69] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *Robotics & Automation Magazine*, Vol. 19, No. 4, pp. 72–82, December 2012. <http://ompl.kavrakilab.org>.
- [70] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings of The 2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–4. IEEE, May 2011.
- [71] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [72] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a Next-Generation Open Source Framework for Deep Learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The 29th Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.
- [73] Preferred Networks. Preferred Networks. <https://www.preferred-networks.jp>, 2021.
- [74] R. Nishino, Y. Okuta, D. Unno, S. Hido, and C. Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations.
- [75] T. Akiba, K. Fukuda, and S. Suzuki. Chainermn: Scalable distributed deep learning framework. *CoRR*, Vol. abs/1710.11351, , 2017.
- [76] Y. Niitani, T. Ogawa, S. Saito, and M. Saito. Chainercv: a library for deep learning in computer vision. In *Proceedings of The 2017 ACM International Conference on Multimedia*, pp. 1217–1220. ACM, 2017.

- [77] J. Bohren. SMACH. <http://wiki.ros.org/smach>, 2021.
- [78] S. Tachi, H. Arai, and T. Maeda. Tele-existence master-slave system for remote manipulation. ii. In *Proceedings of 29th IEEE Conference on Decision and Control*, Vol. 1, pp. 85–90. IEEE, 1990.
- [79] G. Burdea, J. Zhuang, E. Roskos, D. Silver, and N. Langrana. A portable dextrous master with force feedback. *Presence: Teleoper. Virtual Environ.*, Vol. 1, No. 1, p. 1828, 1992.
- [80] K. Hashtrudi-Zaad and S. E. Salcudean. Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation. *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 1, pp. 108–114, 2002.
- [81] K. Minamizawa, D. Prattichizzo, and S. Tachi. Simplified design of haptic display by extending one-point kinesthetic feedback to multipoint tactile feedback. In *Proceedings of 2010 IEEE Haptics Symposium*, pp. 257–260. IEEE, 2010.
- [82] T. Murakami, T. Person, C. Fernando, and K. Minamizawa. Altered touch: Miniature haptic display with force, thermal and tactile feedback for augmented haptics. In *Proceedings of ACM SIGGRAPH 2017 Emerging Technologies*, pp. 1–2. ACM, 2017.
- [83] J. Funda and R. P. Paul. Efficient control of a robotic system for time-delayed environments. In *Proceedings 5th International Conference on Advanced Robotics 'Robots in Unstructured Environments*, Vol. 1, pp. 219–224. IEEE, 1991.
- [84] E-J. Rolley-Parnell, L. Dimitrios D. Kanoulas, A. Laurenzi, B. Delhaisse, L. Rozo, D. G. Caldwell, and N. G. Tsagarakis. Bi-manual articulated robot teleoperation using an external rgb-d range sensor. In *Proceedings of 2018 15th International Conference on Control, Automation, Robotics and Vision*, pp. 298–304. IEEE, 2018.
- [85] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. High speed whole body dynamic motion experiment with real time master-slave humanoid robot system. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5835–5841. IEEE, 2018.
- [86] Y. Kitamura, Y. Itoh, T. Masaki, and F. Kishino. Activecube: a bi-directional user interface using cubes. In *Proceedings of Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES)*, pp. 99–102, 2000.
- [87] D. Anderson, J. L. Frankel, J. Marks, A. Agarwala, P. Beardsley, J. Hodgins, D. Leigh, K. Ryall, E. Sullivan, and J. S. Yedidia. Tangible interaction + graphical interpretation: A new approach to 3d modeling. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques ()*, p. 393402, 2000.
- [88] W. Wan, K. Harada, and F. Kanehiro. Preparatory manipulation planning using automatically determined single and dual arm. *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 1, pp. 442–453, 2020.
- [89] B. Cohen, S. Chitta, and M. Likhachev. Search-based planning for dual-arm manipulation with upright orientation constraints. In *Proceedings of 2012 IEEE International Conference*

- on Robotics and Automation (ICRA)*, pp. 3784–3790. IEEE, 2012.
- [90] N. Yamaguchi, S. Hasegawa, M. Murooka, K. Okada, and M. Inaba. Selective grasp in occluded space by all-around proximity perceptible finger. *Robotics and Autonomous Systems*, Vol. 127, p. 103464, 2020.
- [91] E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158. IEEE, 2000.
- [92] NXP Semiconductors. MPL115A2. <https://www.nxp.com/docs/en/data-sheet/MPL115A2.pdf>, 2021.
- [93] Y. Tenzer, L. P. Jentoft, and R. D. Howe. The feel of mems barometers: Inexpensive and easily customized tactile array sensors. *IEEE Robotics Automation Magazine*, Vol. 21, No. 3, pp. 89–95, 2014.
- [94] F. Kanehiro, M. Inaba, H. Inoue, and S. Hirai. Developmental realization of whole-body humanoid behaviors based on statenet architecture containing error recovery functions. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2000.
- [95] T. Ogura, K. Okada, M. Inaba, and H. Inoue. Behavior network acquisition in multisensor space for whole-body humanoid. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI2003.*, pp. 317–322, 2003.
- [96] Y. Kakiuchi, M. Kamon, N. Shimomura, S. Yukizaki, N. Takasugi, S. Nozawa, K. Okada, and M. Inaba. Development of life-sized humanoid robot platform with robustness for falling down, long time working and error occurrence. In *Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 689–696, 2017.
- [97] D. Morrison, A. W. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-McCue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, K. Vijay, Z. Zhuang, C. F. Lehnert, I. D. Reid, P. Corke, and J. Leitner. Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge. *CoRR*, Vol. abs/1709.06283, , 2017.
- [98] M. Murooka, R. Ueda, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Global planning of whole-body manipulation by humanoid robot based on transition graph of object motion and contact switching. *Advanced Robotics*, Vol. 31, No. 6, pp. 322–340, 2017.
- [99] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8863–8869. IEEE, 2020.
- [100] T. P. Tomo, S. Somlor, A. Schmitz, L. Jamone, W. Huang, H. Kristanto, and S. Sugano. Design and characterization of a three-axis hall effect-based soft skin sensor. *Sensors*, Vol. 16, No. 4, 2016.
- [101] K. Nakagaki, J. Leong, J. L. Tappa, J. Wilbert, and H. Ishii. Hermits: Dynamically recon-

- figuring the interactivity of self-propelled tuis with mechanical shell add-ons. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, p. 882896. ACM, 2020.
- [102] D. Dwibedi, I. Misra, and M. Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1301–1310. IEEE, October 2017.
- [103] F. Bálint-Benczédi and M. Beetz. Variations on a theme: “ it’s a poor sort of memory that only works backwards” . In *Proceedings of The 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8390–8396, Oct 2018.
- [104] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake. Label fusion: A pipeline for generating ground truth labels for real rgb-d data of cluttered scenes. In *Proceedings of The 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, May 2018.
- [105] S. Kitagawa, K. Wada, K. Okada, and M. Inaba. Learning-based task failure prediction for selective dual-arm manipulation in warehouse stowing. In M. Strand, R. Dillmann, E. Menegatti, and S. Ghidoni, editors, *Intelligent Autonomous Systems 15*, pp. 428–439, Cham, 2019. Springer International Publishing.
- [106] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105. 2012.
- [107] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*, Vol. abs/1502.03167, , 2015.
- [108] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.
- [109] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, Vol. 12, pp. 2121–2159, July 2011.
- [110] H. Kusano, A. Kume, E. Matsumoto, and J. Tan. Fcn-based 6d robotic grasping for arbitrary placed objects. In *Proceedings of Warehouse Picking Automation Workshop 2017: Solutions, Experience, Learnings and Outlook of the Amazon Picking Challenge in The 2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, June 2017.
- [111] K. Wada, S. Kitagawa, K. Okada, and M. Inaba. Instance segmentation of visible and occluded regions for finding and picking target from a pile of objects. In *Proceedings of The 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2048–2055. IEEE/RSJ, october 2018.
- [112] M. Schwarz, C. Lenz, G. M. García, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke. Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing. In *Proceedings of The 2018 IEEE International Conference on Robotics*

- and Automation (ICRA)*, pp. 3347–3354. IEEE, 2018.
- [113] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of The 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440. IEEE, June 2015.
- [114] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, Vol. abs/1409.1556, , 2014.
- [115] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of The 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. IEEE, Oct 2017.
- [116] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.
- [117] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of The 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, June 2016.
- [118] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 12, pp. 2481–2495, Dec 2017.
- [119] J. Deng, W. Dong, R. Socher, L-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of The 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE, June 2009.
- [120] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, Vol. abs/1412.6980, , 2014.
- [121] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman. Lessons from the Amazon Picking Challenge. *CoRR*, Vol. abs/1601.05484, , 2016.
- [122] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger, M. Morariu, J. Ju, X. Germann, R. Ensing, J. van Frankenhuyzen, and M. Wisse. Team Delft’s Robot Winner of the Amazon Picking Challenge 2016. *CoRR*, Vol. abs/1610.05514, , 2016.
- [123] D. Morrison, A. W. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-McCue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, K. Vijay, Z. Zhuang, C. Lehnert, I. Reid, P. Corke, and J. Leitner. Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge. In *Proceedings of 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7757–7764. IEEE, May 2018.
- [124] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, Vol.

- abs/1704.06857, , 2017.
- [125] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, Vol. 88, No. 2, pp. 303–338, Jun 2010.
 - [126] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. *CoRR*, Vol. abs/1801.00868, , 2018.
 - [127] Y. Furuta J. Leibs, K. Conley. PR2/app_manager: app_manager dependency for the pr2, required for pr2_app_manager. https://github.com/PR2/app_manager, 2021.
 - [128] S. Kitagawa. knorth55/eus_vive: ROS package for robot tele-manipulation with HTC Vive/Oculus. https://github.com/knorth55/eus_vive, 2021.
 - [129] InfluxData Inc. InfluxDB: Purpose-Built Open Source Time Series Database — InfluxData. <https://www.influxdata.com/>, 2021.
 - [130] K. Chodorow and M. Dirolf. *MongoDB: The Definitive Guide*. O’Reilly Media, Inc., 1st edition, 2010.
 - [131] S. Kitagawa. knorth55/influxdb_store: InfluxDB logger for ROS. https://github.com/knorth55/influxdb_store, 2021.
 - [132] Grafana Labs. Grafana: The open observability platform. <https://grafana.com/>, 2021.
 - [133] Y. Furuta, K. Okada, Y. Kakiuchi, and M. Inaba. An everyday robotic system that maintains local rules using semantic map based on long-term episodic memory. In *Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–7, 2018.
 - [134] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghamsi, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon. In-Datacenter Performance Analysis of a Tensor Processing Unit. *ACM SIGARCH Computer Architecture News*, Vol. 45, No. 2, pp. 1–12, 2017.
 - [135] Google LLC. USB Accelerator. <https://coral.ai/products/accelerator/>, 2021.
 - [136] S. Kitagawa. knorth55/coral_usb_ros: ROS package for Coral Edge TPU USB Accelerator. https://github.com/knorth55/coral_usb_ros, 2021.

- [137] Z. Caoe, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 1, pp. 172–186, 2021.
- [138] A. Lee, T. Kawahara, and K. Shikano. Julius - an open source real-time large vocabulary recognition engine. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 1691–1694, 2001.
- [139] G. Adamides, G. Christou, C. Katsanos, M. Xenos, and T. Hadzilacos. Usability guidelines for the design of robot teleoperation: A taxonomy. *IEEE Transactions on Human-Machine Systems*, Vol. 45, No. 2, pp. 256–262, 2015.

謝辞

本論文は筆者が東京大学情報理工学系研究科創造情報学専攻博士課程に在学中、情報システム工学研究室 (JSK) において、岡田慧教授の御指導のもとで執筆したものです。

岡田慧教授には学部4年生から6年間の長きに渡りご指導いただきました。思い返すと筆者は岡田先生にとって生意気な学生であったかもしれず、大変申し訳ありませんでした。ただ筆者にとって岡田先生はエンジニア、研究者、教育者としての師であり、心の底から尊敬しています。岡田先生のソフトウェアに関する技術と研究に関する広大な展望や意見に憧れ、何とか追いつきたいと頑張ってきました。岡田先生のレベルへ到達することは出来ませんでしたが、エンジニアとして、研究者としての矜持をたくさん教えていただき、少しは憧れに近づくことができたら嬉しいです。また岡田先生は教育者としても素晴らしく、冗談を言いながらもきちんと道を示してくれる頼もしさと、困ったときには相談に真剣に向き合ってくださいる真摯さに博士課程では大変助けていただきました。問題が整理されていないときや精神的に追い込まれているときには、一緒になって現状を分割して解決策を模索していただき、何度も窮地に陥った時に助けていただきました。改めて感謝を申し上げます。ありがとうございました。今後も岡田先生のようなエンジニア、研究者、教育者になれるように邁進していきます。

稲葉雅幸教授には学部4年生から6年間の長きに渡りご指導いただきました。学部4年生の研究室配属から6年間に渡って、研究テーマや研究の進め方などの研究指導はもちろん、研究者としての心構えや立ち振る舞いなど多くのことを教えていただきました。稲葉先生の研究に関する展望や意見は未熟な筆者には最初は理解できず、失敗した後になって身に沁みて理解するというのが何度もありました。稲葉先生の鋭い洞察力と高い行動力に憧れ、尊敬しております。また筆者のような言うことを聞かない学生に対しても根気強くご指導・サポートしていただきました。大変心配していただきご迷惑をおかけいたしました。筆者の研究を受け止めて理解し助言をしていただくなど、寛大に接していただきました。改めて感謝を申し上げます。ありがとうございました。今後も稲葉先生のような洞察力、行動力、包容力を身につけられるよう邁進していきます。

五十嵐健夫教授、國吉康夫教授、猿渡洋教授、千葉滋教授、中山英樹准教授には本論文をまとめるにあたり非常に有意義な多くのご意見を頂きましたこと感謝致します。ロボット、人工知能、深層学習、情報システム、ユーザインタフェースなど様々な研究分野の視点から本論文をまとめる上で必要不可欠なご指摘をいただきました。ありがとうございました。

本研究の一部は、文部科学省博士課程教育リーディングプログラム「東京大学ソーシャル

ICT グローバル・クリエイティブリーダー育成プログラム」の支援を受けています。苗村健教授、水越伸教授にはワークショップ設計について非常に有意義な多くのご意見をいただきましたことを感謝いたします。GCL 関係者の方々にも大変お世話になり、心より感謝致しております。

技術専門職員の山口さんには事務処理・会計処理について様々なサポートをしていただきました。創造情報事務の荒木さんと GCL 事務局の大野さんには書類提出・論文提出の際に大変ご迷惑をお掛けし、サポートしていただきました。垣内洋平准教授には PR2 を用いた生活支援実験について多くの助言をいただき、産総研での実験を計画していただきました。浅野悠紀講師、趙漠居講師、小島邦生助教、小椎尾侑多助教には筆者が疎いヒューマノイドや制御、ハードウェア、回路について教えていただき、ロボットハードウェアの故障をサポートしていただきました。改めて職員・先生方に感謝を申し上げさせていただきます。ありがとうございました。

野沢峻一先輩、矢口裕明先輩、長濱虎太郎先輩には研究の進め方について多くの助言をいただきました。菅井文仁先輩には回路について教えていただき、PR2 の修理をサポートしていただきました。室岡雅樹先輩にはヒューマノイドや全身マニピュレーションに関する知識を教えてください、研究について多くのことを教えていただきました。違うロボットを使っていたにもかかわらず、些細な相談にもものっていただき、こまめに助言をしていただきました。また Euslisp の更なる可能性を室岡さんには見せていただき、ロボットプログラミングやマニピュレーションの面白さを発見することが出来ました。室岡さんの研究に対するストイックな態度とプログラマとしての手の早さを尊敬しています。野田晋太郎先輩、木村航平先輩には筆者が疎いヒューマノイドやその制御について教えていただきました。中島慎介先輩には部屋や研究テーマも全く異なるのにも関わらず、些細な相談にもものっていただき、こまめに助言をしていただきました。石黒康裕先輩には遠隔操縦について様々な情報や知識を教えてください、PR2 の修理を手伝っていただきました。古田悠貴先輩には 73B2 の部屋のロボットやそのソフトウェアの開発方法について教えてください、ロボットプログラミングの楽しさを教えてくださいました。古田さんが卒業されてから PR2 や他のロボットの保守を引き継ぎましたが、そこで古田さんが研究室のロボットソフトウェアを支えておられたことを痛感しました。古田さんのプログラマとしての問題解決能力と洞察力を尊敬しています。廣瀬俊典先輩には Baxter のハンドの作り方を教えてください、Baxter のカメラなどのハードウェアサポートをしていただきました。東風上奏絵先輩には未来館で行う実験について助言をいただき、未来館との交渉を手伝っていただきました。和田健太郎先輩、同期であり先輩であるという不思議な関係のなか、研究室配属からロボットや深層学習のプログラミングについて教えてください、論理的な研究の展開方法を教えてくださいました。また意見が合わないときには夜まで議論するなど、筆者の研究に対して大変多くの助言をしていただきました。矢野倉伊織先輩には同期であり先輩であるという不思議な関係のなか、ロボット研究や深層学習の今後の発展について夜遅くまで議論をさせていただきました。また新たな技術の情報を教えてください、ロボットの学習分野の新たな側面を学ぶことができました。同期の河原塚健人くん、長谷川峻くん、板東正祐くん、安齋智紀くん、藤井綺香さん、石田寛和くん、西尾卓純くんとは、博士課程の苦楽を共に

する仲間として時にぶつかり合い愚痴をこぼしながらも、励まし合って切磋琢磨することができました。特に長谷川くんとは多くの相談や議論をしてもらい、博士課程の励みとなっていました。後輩の内海祐斗くん、山口直也くん、伊藤秀朗くん、新城光樹くん、平岡直樹くん、室岡貴之くんには PR2 や Baxter, Fetch のメンテナンスの手伝いや、ロボット研究者としてのあり方と今後のロボット研究についての長時間の議論をさせてもらい、実験や論文執筆を行う上でお世話になりました。同じ部屋の後輩である河村洋一郎くん、若林隼平くん、種本雅さん、金沢直晃くん、大日方慶樹くん、塚本直人くん、海士部宏紀くんには実験や論文執筆を行う上で部屋の一部分を専有するなどご迷惑をおかけいたしました。改めて先輩・後輩方に感謝を申し上げさせていただきます。ありがとうございました。

ドイツ航空宇宙センターの Daniel Leidner さんには 1 ヶ月という短い間でしたが、Justin とその操縦システムである Hug を実際に使って研究をさせていただき、センター内で筆者の研究を発表する機会をいただきました。また同センターの Adrian Bauer, Anne Köopken, 阪上遼さんには現地での生活から Justin を用いた実験のやり方までサポートしていただきました。改めてドイツ航空宇宙センターの方々には感謝を申し上げます。ありがとうございました。

その他、ご卒業された先輩・後輩を含め、多くの方々とともに研究室でともに過ごし、夜遅くまで議論し夢を語り合い、ロボットがうまく動かない現実には逃げることなく向きあい、大変有意義で学びの多い研究室生活を送ることができました。このロボット三昧の研究生生活を共に送り、楽しく学びあるものにしていただいた全ての方に感謝致しております。

筆者が研究室配属されてからの 6 年間使い続けている Baxter, 博士課程の 3 年間使い続けた PR1012, PR1040 や Fetch15, Fetch1075 は本論文を構成する上で不可欠な実験を行うために 1 年を超えた長期間実験や名古屋までの長距離輸送、電源や通信の多くの故障を乗り越えてくれました。ありがとうございました。

最後に筆者の希望を尊重し、10 年間にも及ぶ長い大学生活を理解し、筆者のことを支えてくれた両親と家族、真由に心から感謝致します。

2021 年 12 月 3 日 北川 晋吾

付録 A

Miniature Tangible Cube の実装

本付録では 4.5.4 項で説明した Miniature Tangible Cube の回路構成の具体的な基板実装について説明する。

A.1 Miniature Tangible Cube の基板実装

Miniature Tangible Cube Rev.2 の詳細な回路図と基板実装について記載する。Miniature Tangible Cube には上下面と左右前後面の 2 種類の基板があり、これらを組み合わせることで立方体形状を形成する。Miniature Tangible Cube Rev.1 は各辺約 120 mm の立方体であるのに対して Miniature Tangible Cube Rev.2 は各辺約 100 mm の立方体であり、Rev.1 と Rev.2 の違いとしては大きさと電源スイッチの有無が挙げられるが近接センサや計算機などの機能については大きな違いはない。また Miniature Tangible Cube Rev.2 には、拡張性として NXP 社の気圧センサ MPL115A2 [92, 93] を各面 5 つずつ、とそのため I2C マルチプレクサを配置できるように設計されているが本研究では使用していない。Miniature Tangible Cube Rev.1 の回路図については図 A.1 と図 A.3、その回路実装については図 A.2 と図 A.4、Miniature Tangible Cube Rev.2 の回路図については図 A.5 と図 A.7、その回路実装については図 A.6 と図 A.8 にそれぞれ上下面と左右前後面の 2 種類について示す。

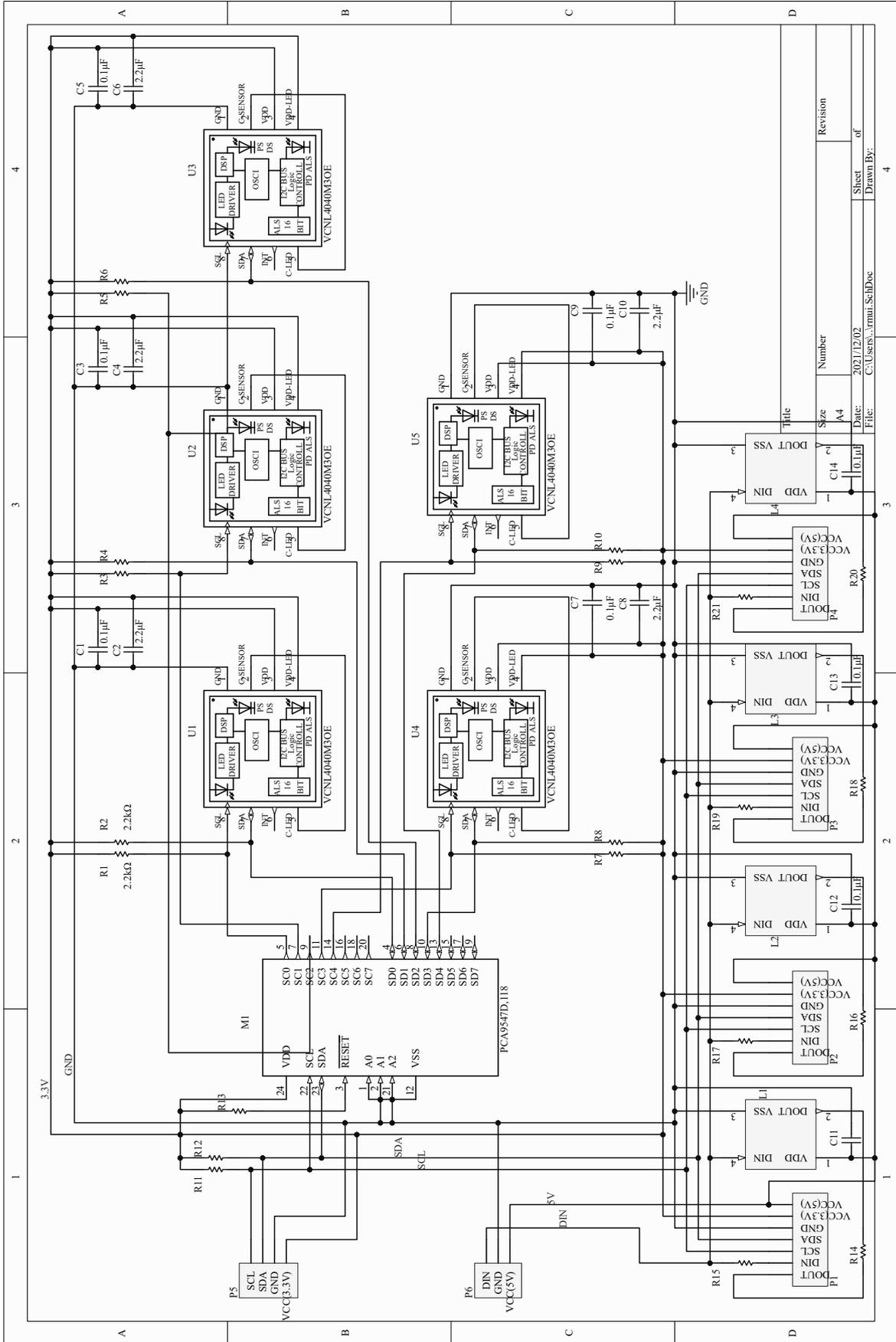


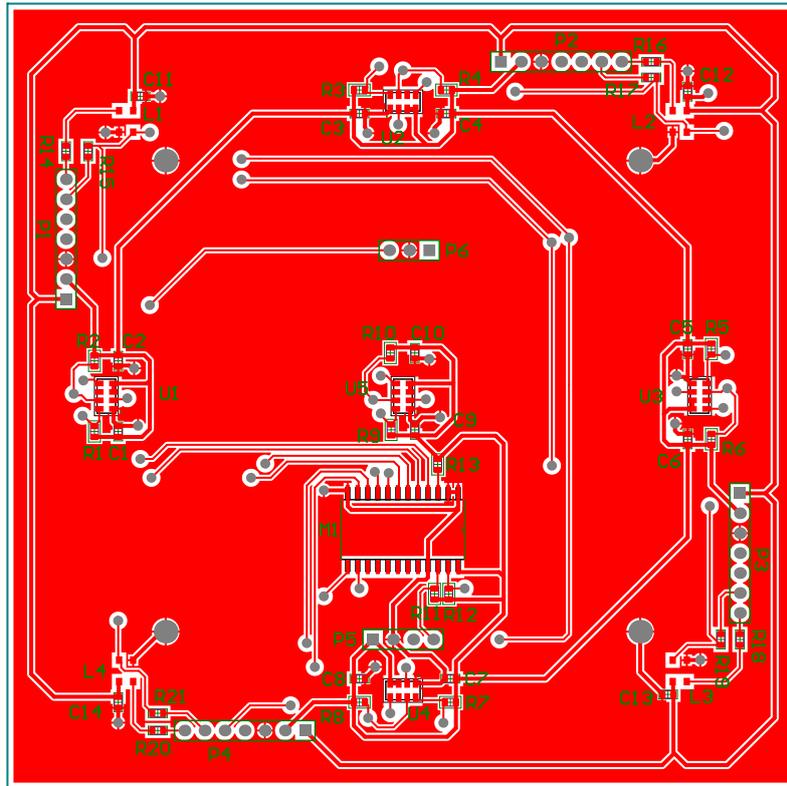
図 A.1: Miniature Tangible Cube Rev.1 の上下センサ基板の回路図

Title	Size	Number	Revision
	A4		

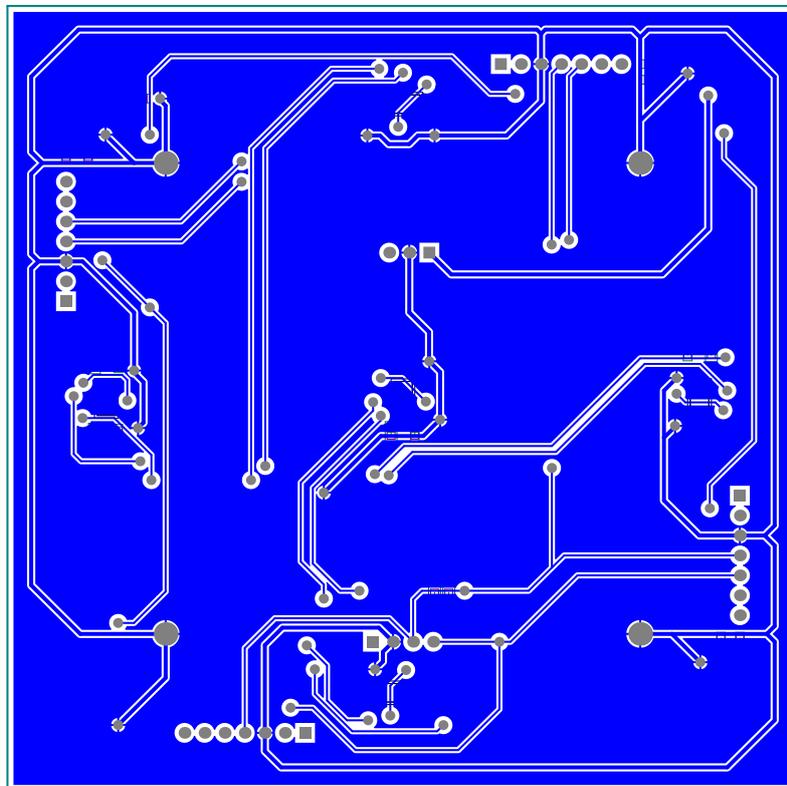
Sheet	of
2	4

Date:	Drawn By:
2021/12/02	

File:
C:\Users\...rmai\ScdDoc



(a) Miniature Tangible Cube Rev.1 の上下センサ基板の表面



(b) Miniature Tangible Cube Rev.1 の上下センサ基板の裏面

図 A.2: Miniature Tangible Cube Rev.1 の上下センサ基板の回路図と実装

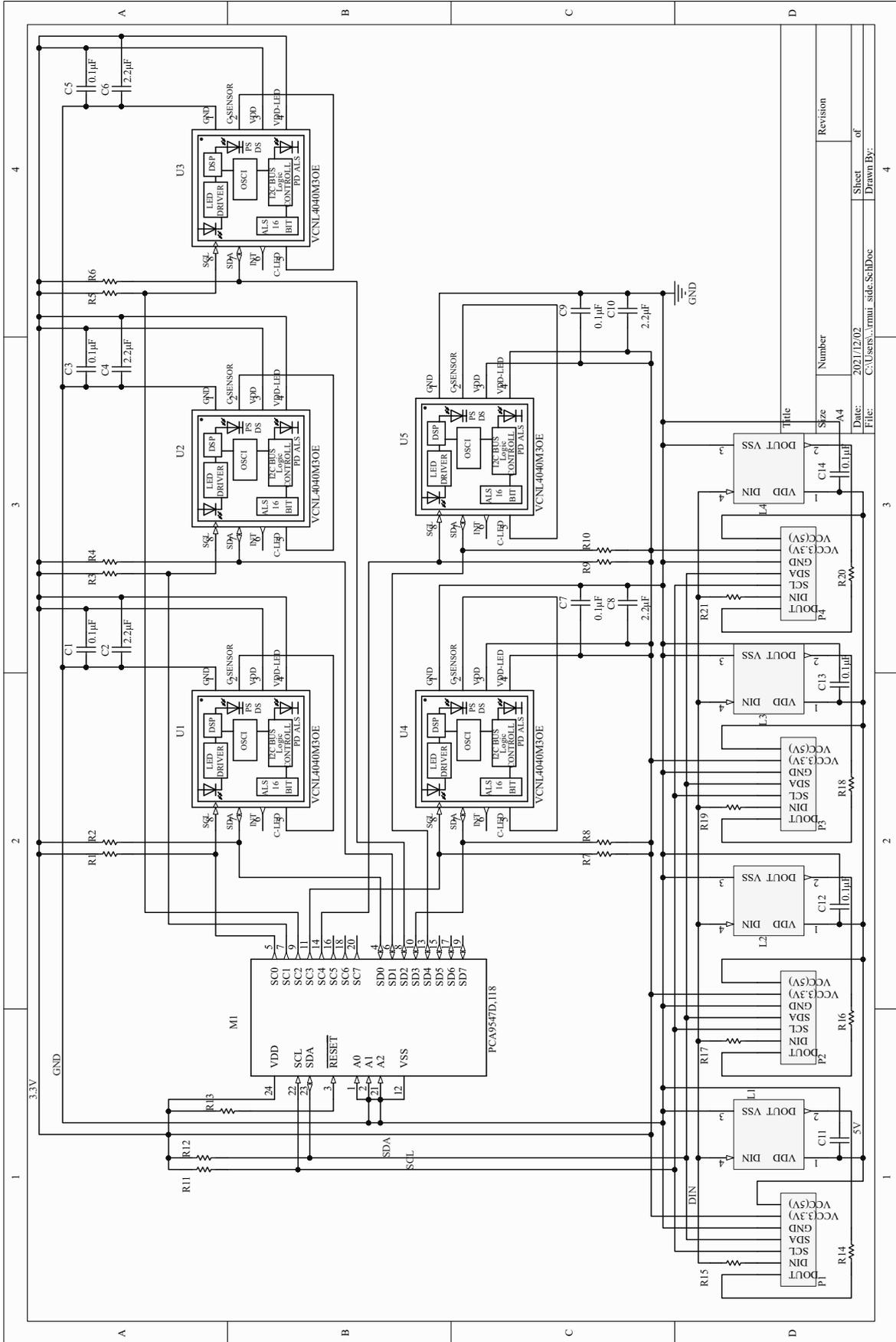
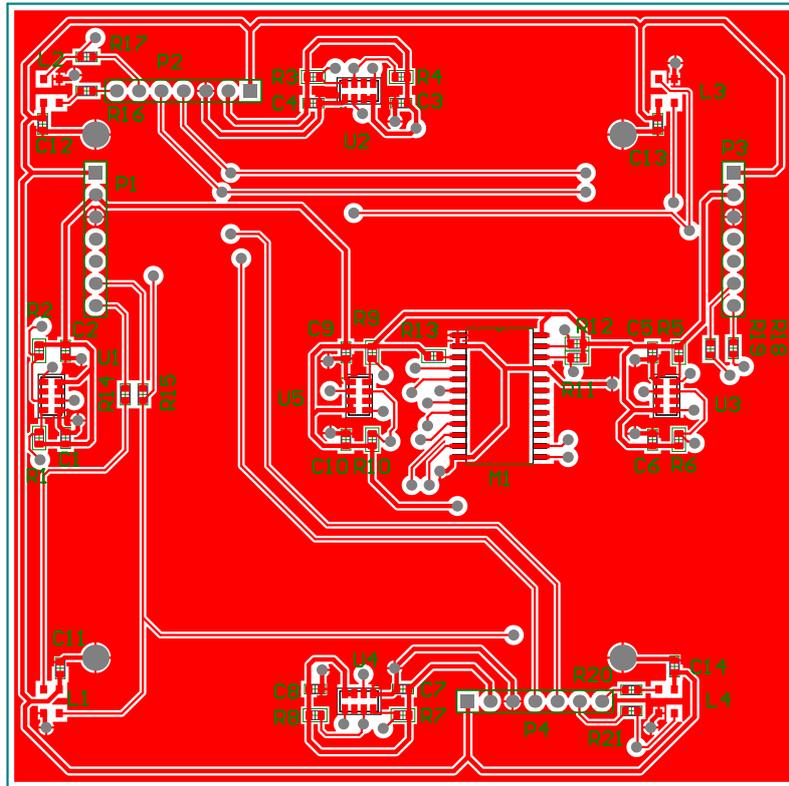
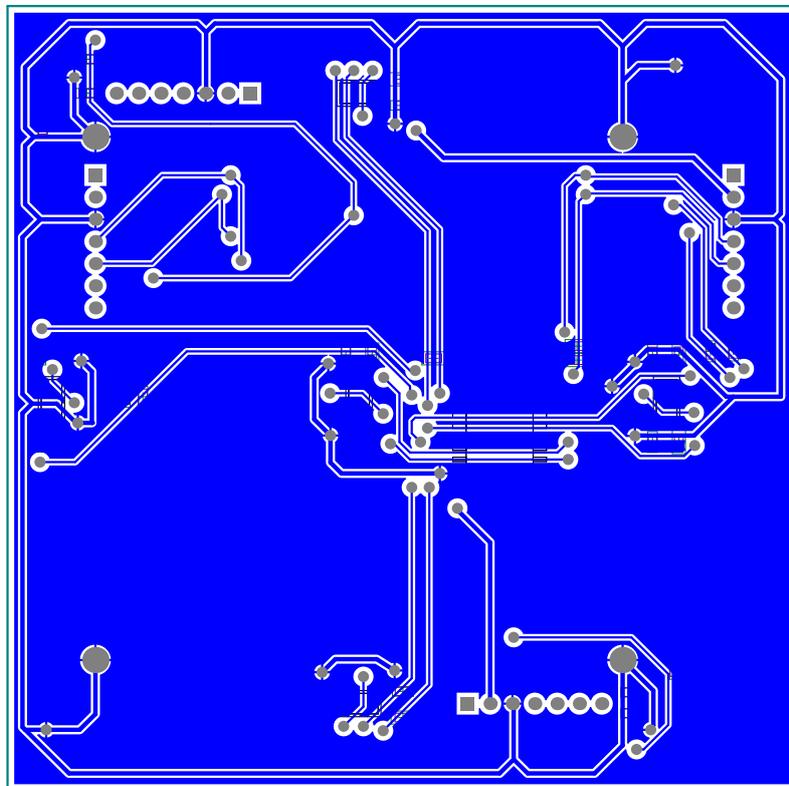


図 A.3: Miniature Tangible Cube Rev.1 の前後左右センサ基板の回路図



(a) Miniature Tangible Cube Rev.1 の前後左右センサ基板の表面



(b) Miniature Tangible Cube Rev.1 の前後左右センサ基板の裏面

図 A.4: Miniature Tangible Cube Rev.1 の前後左右センサ基板の回路図と実装

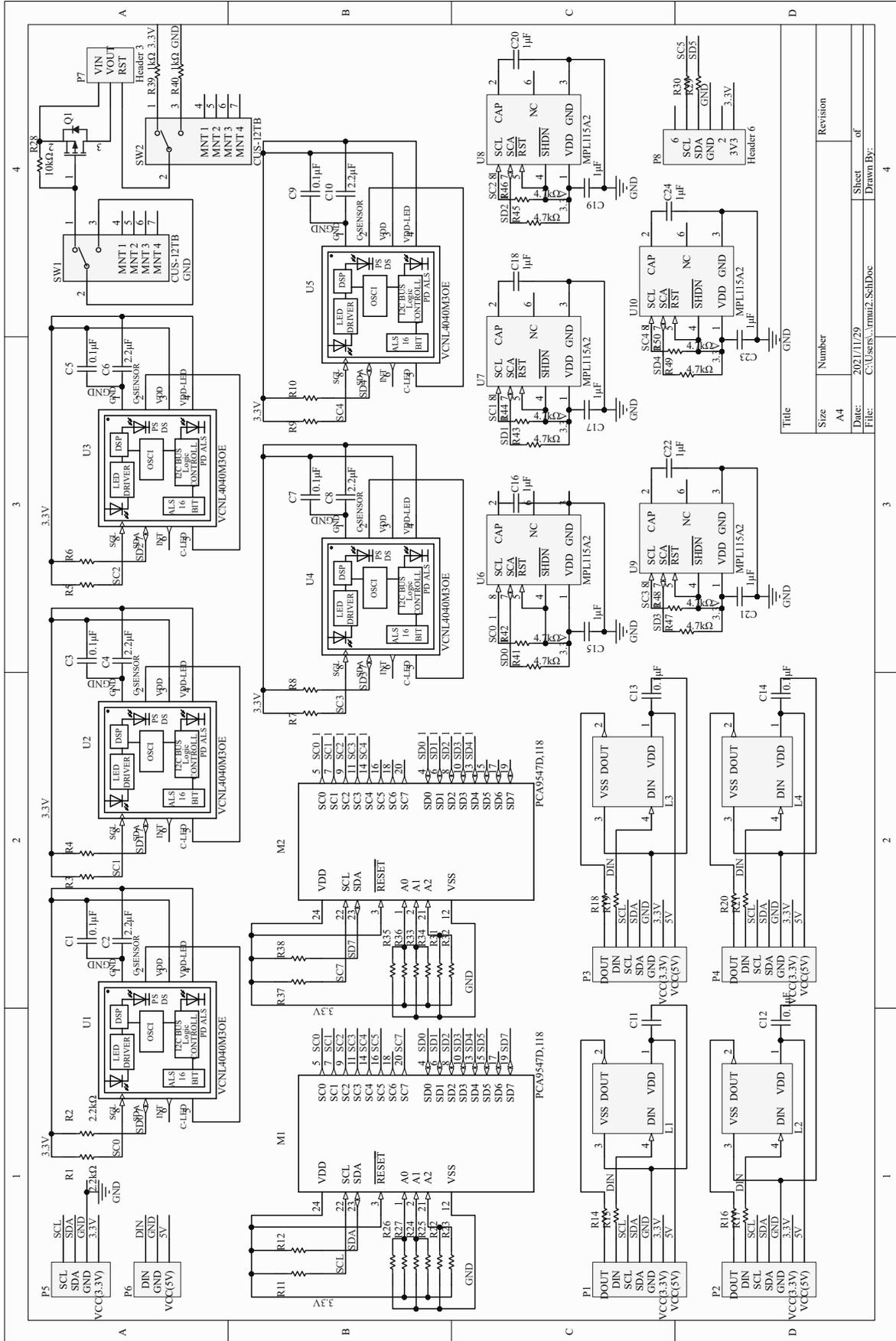


図 A.5: Miniature Tangible Cube Rev.2 の上下センサ基板の回路図 (図 4.10 の拡大)

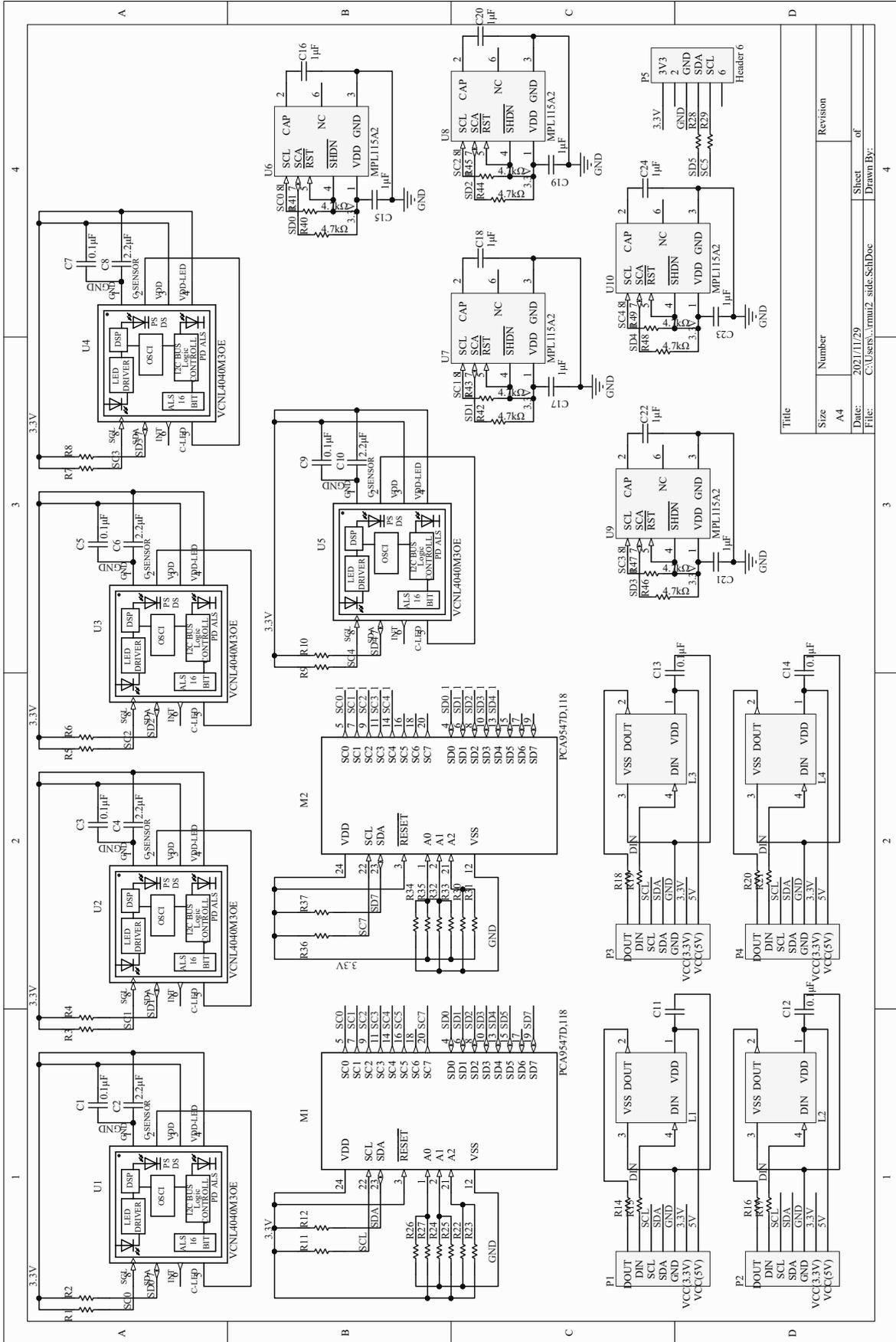
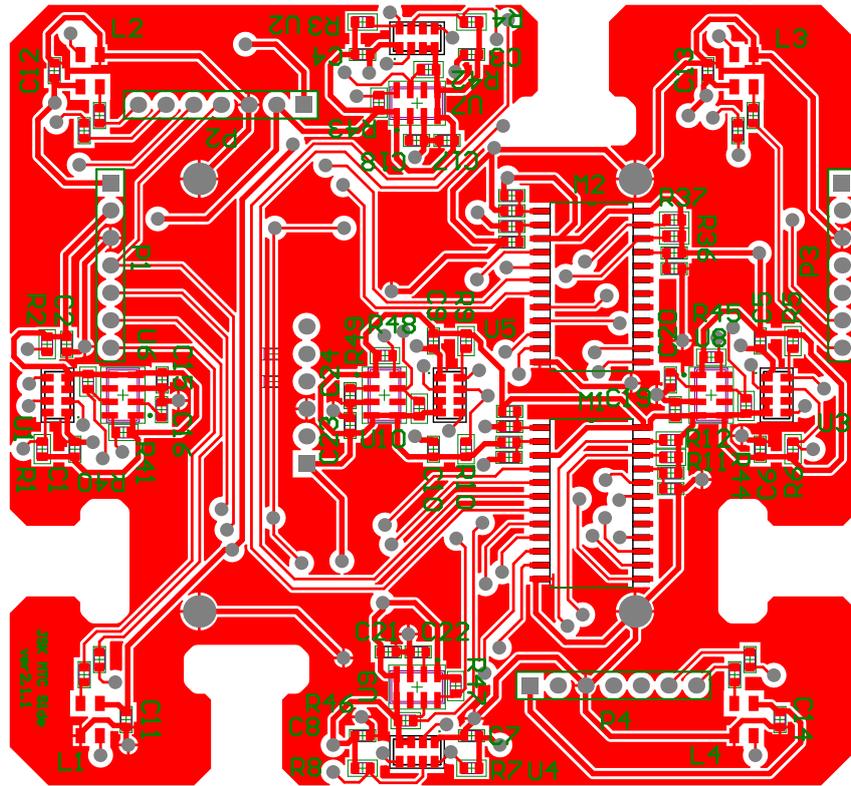
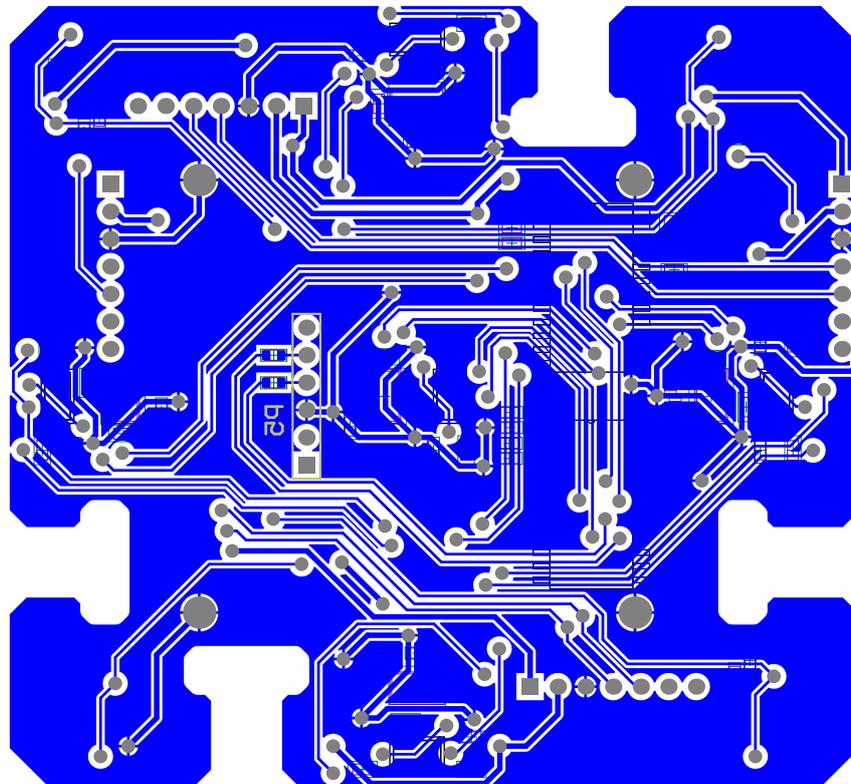


図 A.7: Miniature Tangible Cube Rev.2 の前後左右センサー基板の回路図 (図 4.11 の拡大)



(a) Miniature Tangible Cube Rev.2 の前後左右センサ基板の表面



(b) Miniature Tangible Cube Rev.2 の前後左右センサ基板の裏面

図 A.8: Miniature Tangible Cube Rev.2 の前後左右センサ基板の回路図と実装

付録 B

Miniature Tangible Cube による物体操作作業の操縦実行実験

本付録では 4.9 節で説明した Miniature Tangible Cube による物体操作作業の操縦実行実験について示す。

B.1 Miniature Tangible Cube によるぬいぐるみ詰め寄せ実験

本章では 4.9.4.1 目で説明した Miniature Tangible Cube によるぬいぐるみ詰め寄せ実験の操縦者側の様子について示す。図 B.1, 図 B.2 に示すように、まず操縦者が右側に置かれた Miniature Tangible Cube に対して右側面に接触することで、(図 B.1b), ロボットが右側に置かれたぬいぐるみを右腕で右側面に対して接触した(図 4.35b, 図 B.2b)。そして操縦者が右側の Miniature Tangible Cube を左に押し進め左側の Miniature Tangible Cube と接触させることで(図 B.1c), ロボットが右側に置かれたぬいぐるみを右腕で左に移動させて、左側に置かれたぬいぐるみと接触させた(図 4.35c, 図 B.2c)。最後に操縦者が右側の Miniature Tangible Cube への接触を離すことで(図 B.1d), ロボットが右側に置かれたぬいぐるみへの接触を離した(図 4.35d, 図 B.2d)。以上の一連の動作列によってロボットが右側のぬいぐるみを押し進め左側のぬいぐるみに詰め寄せることに成功した。

B.2 Miniature Tangible Cube によるぬいぐるみ積み重ね実験

本章では 4.9.4.1 目で説明した Miniature Tangible Cube によるぬいぐるみ積み重ね実験の操縦者側の様子について示す。図 B.3, 図 B.4, 図 B.5, 図 B.6 に示すように、まず操縦者が手前に置かれた Miniature Tangible Cube に対して左右側面に接触することで(図 B.3b, 図 B.5b), ロボットが手前に置かれたぬいぐるみを双腕で抱えた(図 4.37b, 図 B.4b, 図 4.38b, 図 B.6b)。そして操縦者が抱え込んだ手前の Miniature Tangible Cube を持ち上げて、奥に置かれたもう一つ Miniature Tangible Cube の上に置くことで、(図 B.3c, 図 B.5c), ロボットが手前に置かれたぬいぐるみを左右から抱え込んで持ち上げ、持ち上げたぬいぐるみをロボットから見て奥

にある箱の上に積み重ねた (図 4.37c, 図 B.4c, 図 4.38c, 図 B.6c). 最後に操縦者が積み重ねられた Miniature Tangible Cube に対する接触を左右と離すことで (図 B.3d, 図 B.5d), ロボットが積み重ねられたぬいぐるみに対する接触を離した. (図 4.37d, 図 B.4d, 図 4.38d, 図 B.6d). 以上の一連の動作列によってロボットが手前のぬいぐるみを持ち上げて奥の箱に積み重ねることに成功した.

B.3 Miniature Tangible Cube による袋入り靴下の箱収納作業 実験

本章では 4.9.5.1 目で説明した Miniature Tangible Cube による袋入り靴下の箱収納作業実験の操縦者側の様子について示す. 図 4.44, 図 B.8, 図 B.7 に示すように, まず操縦者が左側に置かれた Miniature Tangible Cube に対して上側面に接触することで (図 B.7b), ロボットが左側に置かれた把持物体を左腕で把持し (図 4.44b, 図 B.8b), 操縦者が左側に置かれた Miniature Tangible Cube を上側面から持ったまま持ち上げることで (図 B.7c), ロボットは把持物体を持ち上げた (図 4.44c, 図 B.8c). そして操縦者が持ち上げた Miniature Tangible Cube を中央手前に置かれた Miniature Tangible Cube の上に置くことで (図 B.7d), ロボットは把持物体を中央に置かれたダンボールの箱の上まで移動させ (図 4.44d, 図 B.8d), 操縦者が上側面から掴んでいた Miniature Tangible Cube を離すことで (図 B.7e), ロボットは把持物体を離して中央に置かれたダンボールの箱の中に把持物体を収納した (図 4.44e, 図 B.8e). この後の動作は 4.9.3.2 目と同様に, 操縦者が中央手前に積み重ねられた 2 つの Miniature Tangible Cube の下のデバイスを左右から接触して持ち上げることで (図 B.7f), ロボットは把持物体が収納されたダンボールの箱を左右から抱え込んで持ち上げ (図 4.44f, 図 B.8f), 操縦者が持ち上げた 2 つの Miniature Tangible Cube を奥に置かれた Miniature Tangible Cube の上に置くことで (図 B.7g), ロボットは把持物体が収納されたダンボールの箱を置くにあるダンボールの箱の上に積み重ねた (図 4.44g, 図 B.8g). 最後に操縦者が抱えていた Miniature Tangible Cube に対する接触を左右と離すことで (図 B.7h), ロボットは積み重ねられたダンボールの箱に対する接触を離した (図 4.44h, 図 B.8h). 以上の一連の動作列によって, ロボットが左側におかれた物体を把持して持ち上げて, それを中央手前に置かれたダンボールの箱に収納したあとに, 物体を収納したその箱を置くにあるダンボールの箱の上に積み重ねることに成功した.

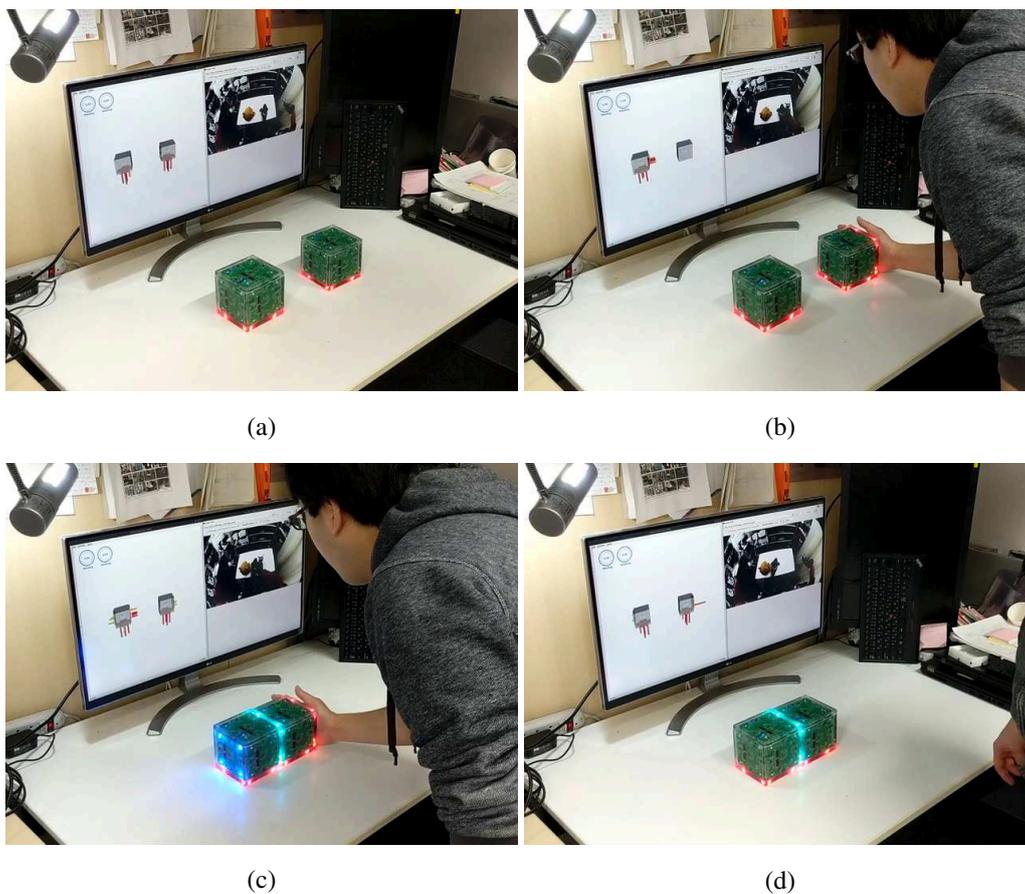


図 B.1: Miniature Tangible Cube によるぬいぐるみ詰め寄せ実験 (操縦者側)

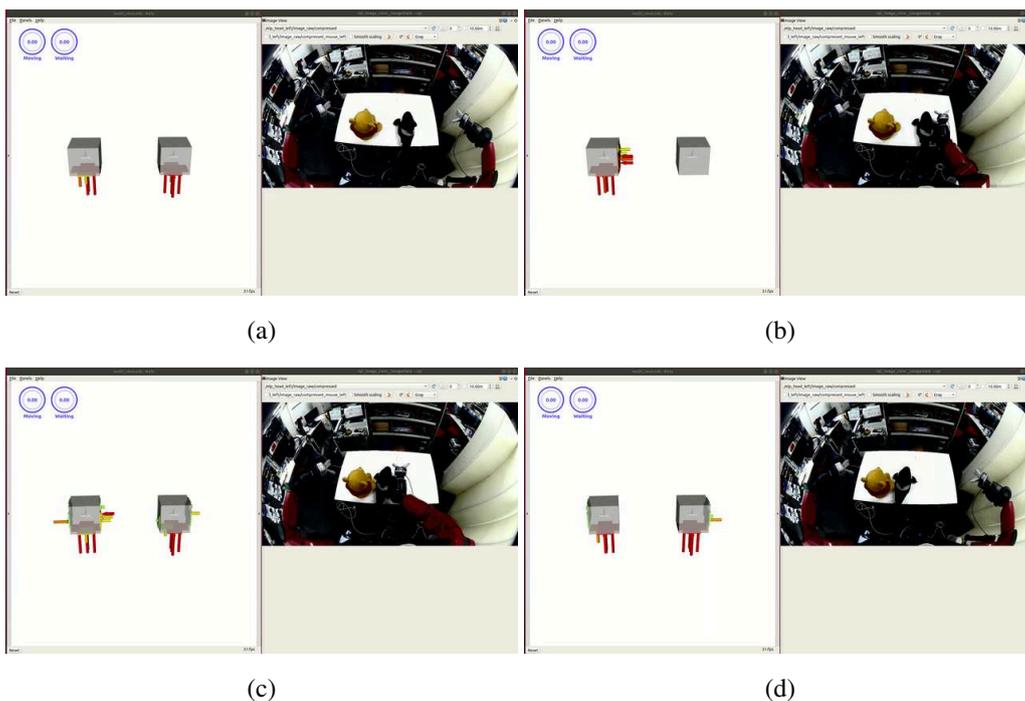


図 B.2: Miniature Tangible Cube によるぬいぐるみ詰め寄せ実験 (操縦者側表示画面)

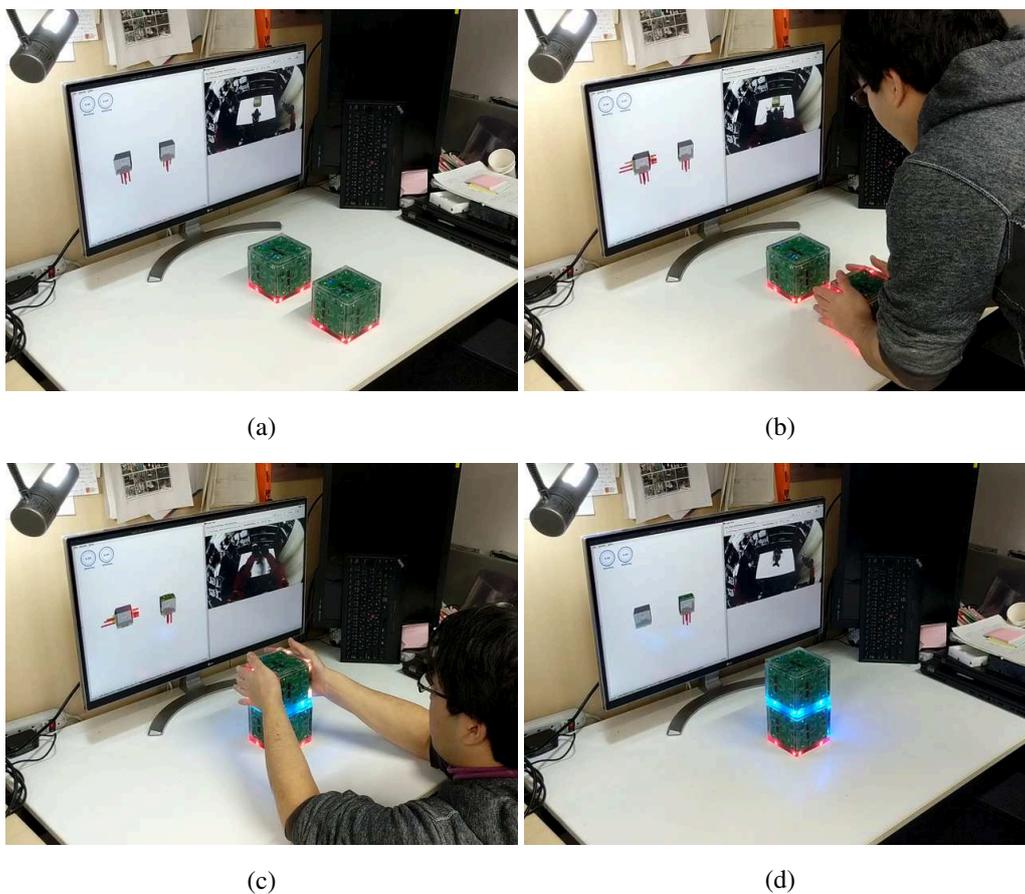


図 B.3: Miniature Tangible Cube によるシャチのぬいぐるみ積み重ね実験 (操縦者側)

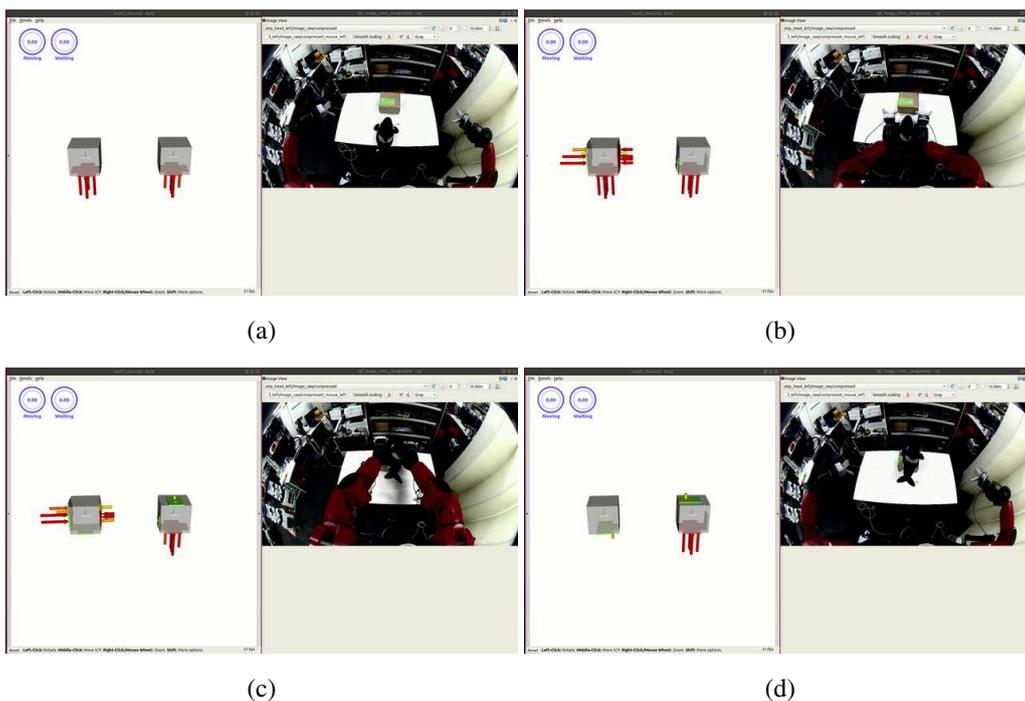


図 B.4: Miniature Tangible Cube によるシャチのぬいぐるみ積み重ね実験 (操縦者側表示画面)

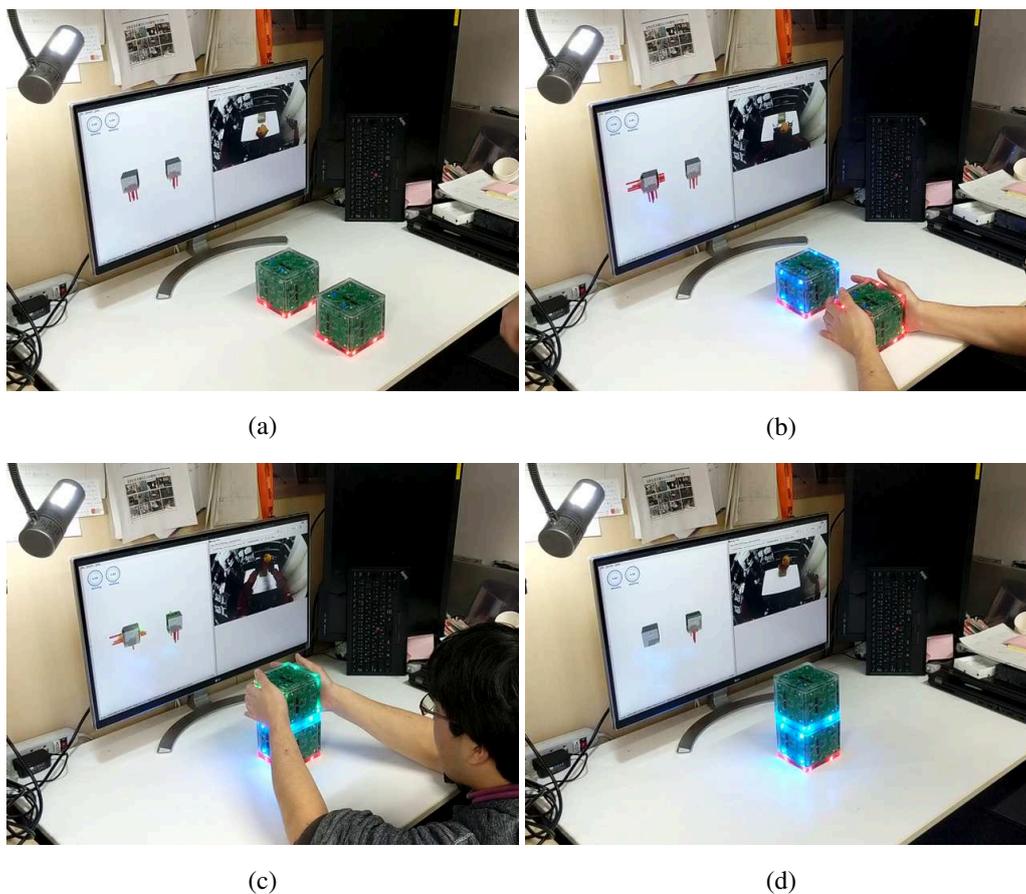


図 B.5: Miniature Tangible Cube によるくまのぬいぐるみ積み重ね実験 (操縦者側)

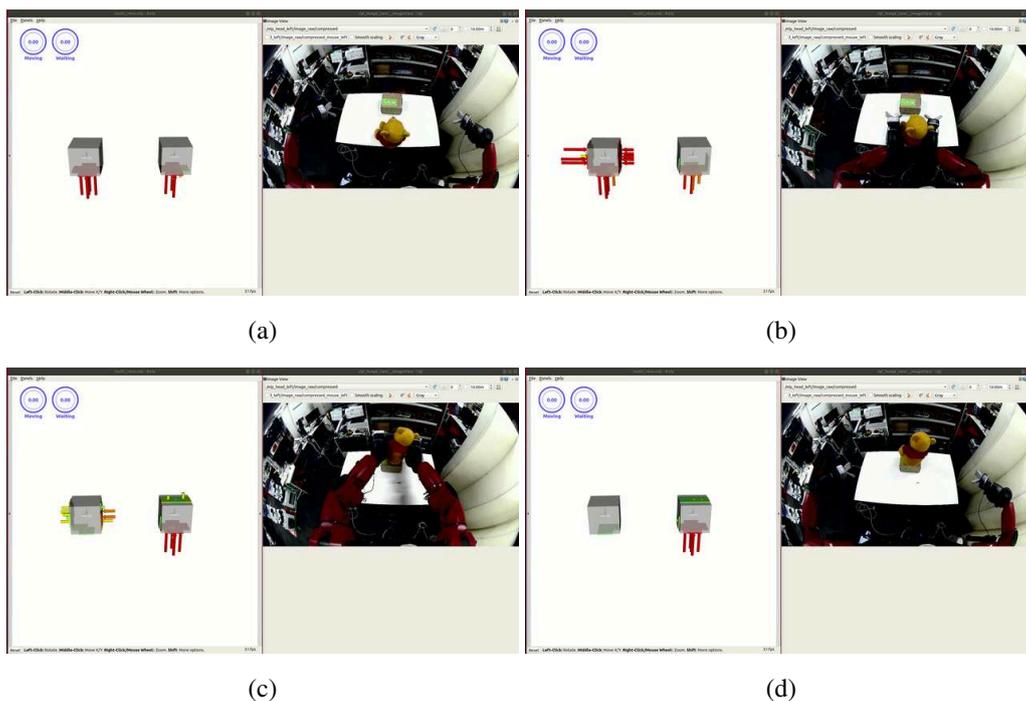


図 B.6: Miniature Tangible Cube によるくまのぬいぐるみ積み重ね実験 (操縦者側表示画面)

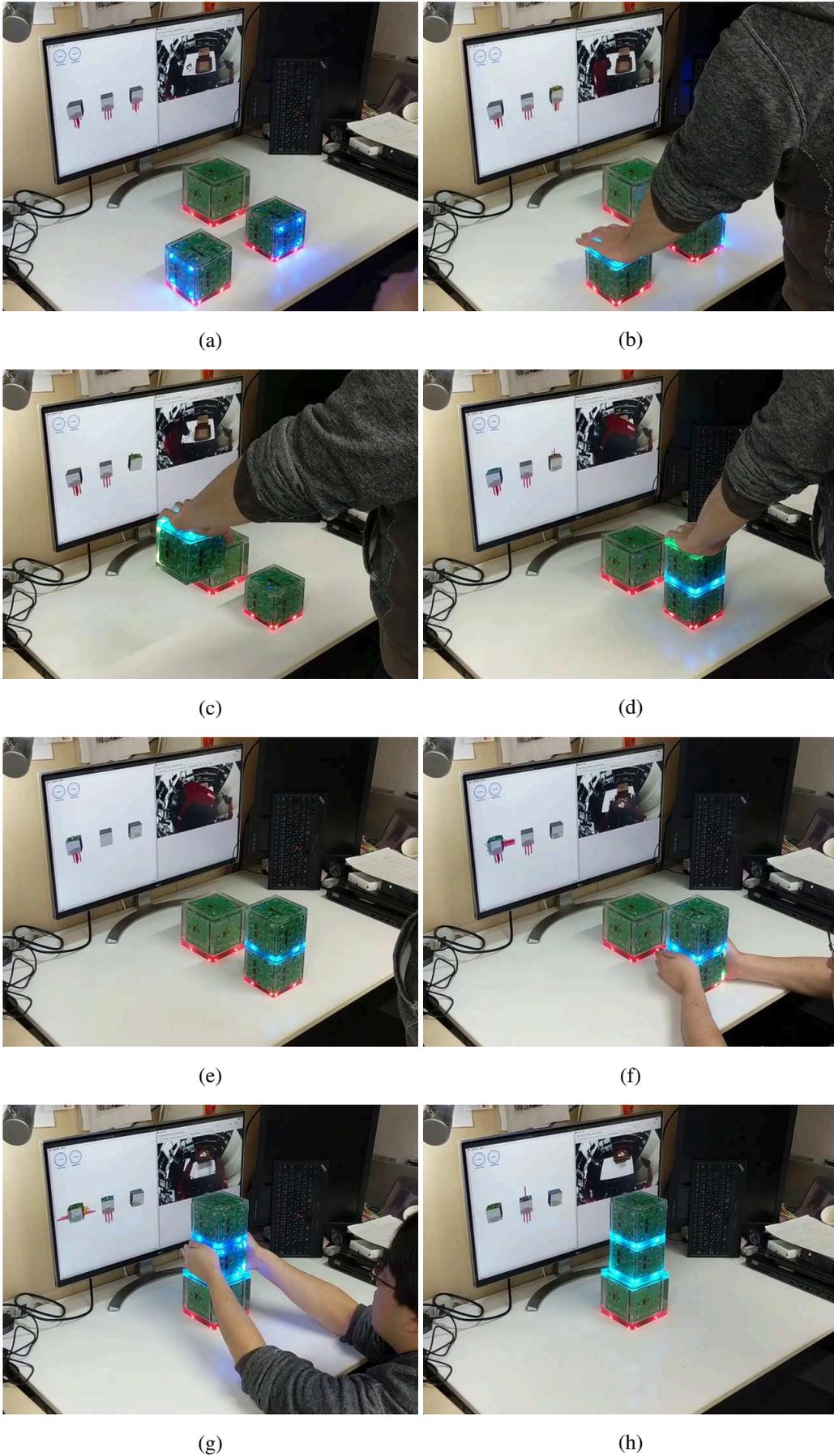


図 B.7: Miniature Tangible Cube による袋入り靴下の箱収納作業実験 (操縦者側)



図 B.8: Miniature Tangible Cube による袋入り靴下の箱収納作業実験 (操縦者側表示画面)

付録 C

コンビニ物品操作操縦実験

本実験では 6.6.1 項で説明した視聴覚統合型遠隔操縦システムの有効性について評価を行うために、6.6.1 項で説明した視聴覚統合型遠隔操縦システムと 双臂移動ロボット PR2-JSK を用いて産業総合研究所内のコンビニ環境で実験を行った。実験としては、コンビニ物品を対象として以下の 4 種類の実験を遠隔操縦で行った。

- 品出し
- 陳列
- ペットボトル取り出し
- ゴミ分別

操縦者はロボットから離れた場所から VR デバイスを用いて、ヘッドマウントディスプレイに表示されるロボットの頭部カメラ画像データと、イヤホンから提示されるロボットの頭部マイクからの音声データを元に操縦を行った。

C.1 コンビニ物品品出し操縦実験

本実験では、操縦者は双臂移動ロボット PR2-JSK を遠隔操縦して棚に積まれたインスタントヌードルを台車に移動して運び出すという品出し作業を遠隔操縦で行った。結果は図 C.1 に示すように操縦者はロボットの双臂を使って棚に積まれたインスタントヌードルを抱え上げて台車の上に移動し、その台車を両腕で固定した後に移動することで、品出し作業を行うことが出来た。この実験の際に図 C.1d に示すように積み上がったインスタントヌードルを台車に置く際に崩れてしまったが、図 C.1e に示すように並べ直すことで、物品を落とすことなく台車で移動することが出来た。

C.2 コンビニ物品陳列操縦実験

本実験では、操縦者は双臂移動ロボット PR2-JSK を遠隔操縦して台車に積まれたインスタントヌードルを棚に並べるといった陳列作業を遠隔操縦で行った。結果は図 C.2 に示すように

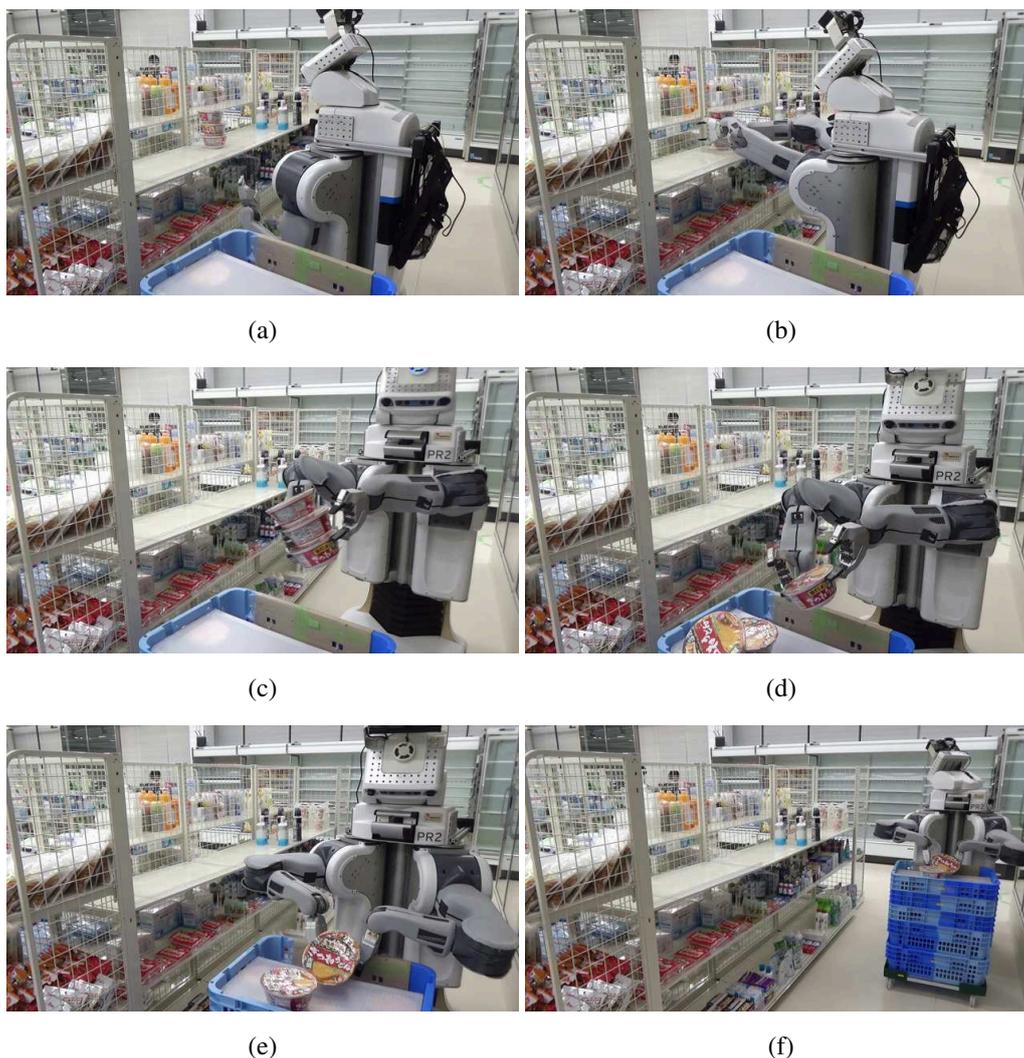


図 C.1: PR2-JSK によるコンビニ物品品出し 操縦実験

操縦者はロボットの双腕を使って台車の上に積まれたインスタントヌードルを抱え上げ、棚の上に配置することで、陳列作業を行うことが出来た。この実験の際に、図 C.2e に示すように、一度インスタントヌードルを配置したあとに、右手を用いてインスタントヌードルを押すことで配置位置の調整を行った。

C.3 コンビニペットボトル取り出し 操縦実験

本実験では、操縦者は双腕移動ロボット PR2-JSK を遠隔操縦して飲料棚に陳列されているペットボトルを取り出すというペットボトル取り出し作業を遠隔操縦で行った。結果は図 C.3 に示すように操縦者はロボットの双腕を用いてペットボトル棚の扉を開けた後に、右腕で棚に陳列されたペットボトルを上引き抜きながら取り出し、棚の扉を右腕で閉めることで、ペッ



図 C.2: PR2-JSK によるコンビニ物品陳列操縦実験

トボトル取り出し作業を行うことが出来た。この実験の際に図 C.3a, 図 C.3b に示すように扉のハンドルをまず右腕でつかみ開けた後に、左手で扉を抑えながら右腕で掴んでいた扉をハンドルを離すことで、扉が自重で閉まることを抑えながら扉を開けることが出来た。

C.4 コンビニゴミ分別操縦実験

本実験では、操縦者は双腕移動ロボット PR2-JSK を遠隔操縦してゴミ箱の上に置かれている以下の3種類のゴミをそれぞれ分別して捨てるというゴミ分別作業を遠隔操縦で行った。

- 紙コップ (燃えるゴミ)
- ペットボトル (ペットボトル)
- プラスチック袋 (燃えるゴミ)

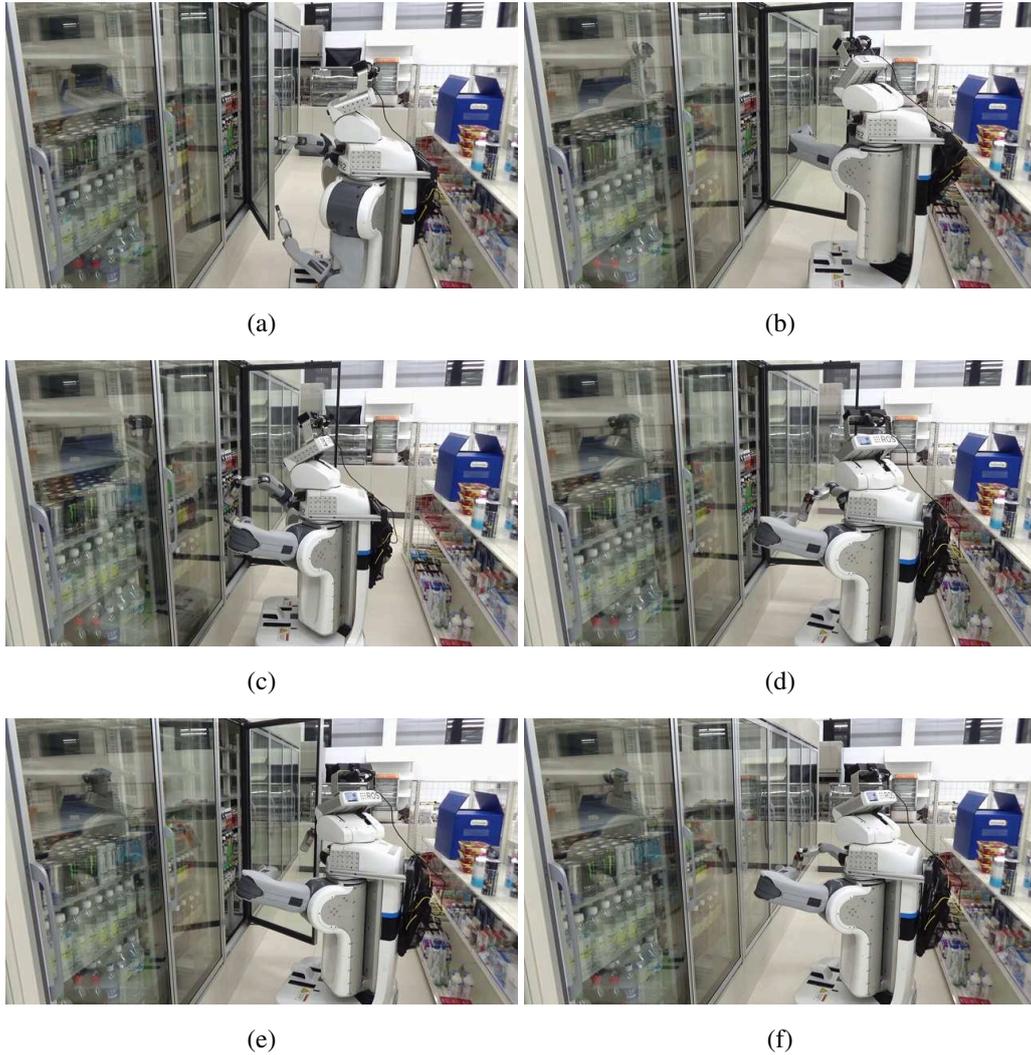


図 C.3: PR2-JSK によるコンビニペットボトル取り出し操縦実験

結果は図 C.4 に示すようにまず紙コップを掴み上げて燃えるゴミのゴミ箱に廃棄し、次にペットボトル上部を持ち上げてペットボトルのゴミ箱に廃棄し、最後にプラスチック袋を燃えるゴミに分別した。この実験の際に図 C.4c, 図 C.4d に示すようにペットボトルを把持したまま一度ペットボトルのゴミ箱に入れたが、ゴミ箱の入り口で詰まってしまったため、一度ペットボトルの把持をやめてからペットボトルを右手で押し込むことでペットボトルの廃棄が出来た。

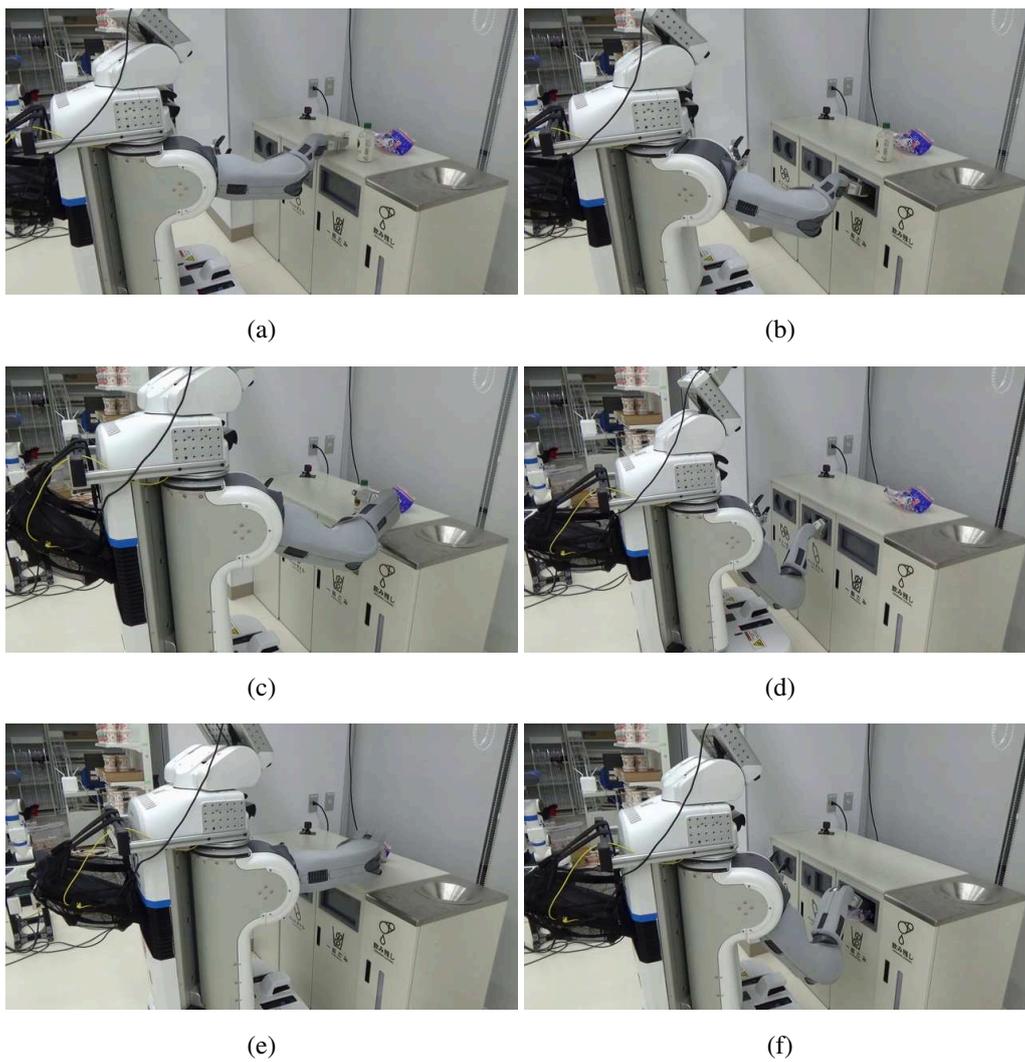


図 C.4: PR2-JSK によるコンビニゴミ捨て操縦実験

付録 D

日本科学未来館・東京大学間遠隔物体ピックアップ操縦実験

本実験では 6.6.1 項で説明した視聴覚統合型遠隔操縦システムの有効性について評価を行うために、6.6.1 項で説明した視聴覚統合型遠隔操縦システムと双腕ロボット Baxter-JSK-SoftHand を用いて図 D.1 に示すような、ロボット経験の少ない一般の方を対象とした日本科学未来館・東京大学間遠隔物体取り出し実験を行った。本実験では遠隔物体取り出し作業を、遠隔地側環境にあるロボットの前に置かれた袋の中に図 D.1b のように置かれた対象物品を把持し、袋の外の机の上に取り出す作業と定義し、操縦者が図 D.3 の対象物品を取り出すことができたかと作業にかかった時間を計測した。本実験で用いた袋はハーフサイズ物資輸送用バッグ(CTB)は宇宙ステーションで実際に使われているもの、対象物品は図 D.3 に示すような宇宙食が入った袋を用意し、参加者には宇宙にいるロボットを遠隔操縦するという説明のもと実験を行った。

図 D.1a と図 D.2 は、それぞれ東京大学の遠隔地側環境と日本科学未来館の操縦者側環境を映したものであり、操縦者側環境では操縦者は正面にあるディスプレイとネックスピーカから視聴覚情報を認識しながら、VR コントローラを用いて双腕ロボット Baxter-JSK-SoftHand の片腕を操縦して、遠隔物体取り出し作業を行った。本実験では、遠隔地側の Baxter-JSK-SoftHand とロボット制御用 PC、操縦者側の制御用 PC と情報提示用 PC の計 4 台を VPN で接続し、VR コントローラの計測と逆運動学計算は操縦者側制御用 PC で行われたのちに、関節角度指令が遠隔地側ロボット制御用 PC に送信され、遠隔地側ロボット制御用 PC から送信された各関節角度情報やカメラ画像のセンサ情報は、操縦者側情報提示用 PC でディスプレイとネックスピーカから操縦者に提示された。

D.1 実験参加者・実験手順

本実験は、日本科学未来館において希望する来館者を対象として、ロボットに関わる研究・仕事を行っていない 21 人(男性 9 人女性 12 人、年齢 9-59 歳)の一般来館者に対して 2020 年 9 月 11 日から 13 日の 3 日間行った。なお本実験の倫理審査は東京大学の倫理審査委員会



(a) 遠隔物体取り出し実験の遠隔地側環境 (東京大学) (b) 遠隔物体取り出し実験に使用したハーフサイズ物資輸送用バッグ (CTB)

図 D.1: 遠隔物体取り出し実験の遠隔地側環境



図 D.2: 遠隔物体取り出し実験の操作者側環境

で承認を受けている。実験手順については以下に記す、

1. インフォームド・コンセントと基本情報の記入
2. 遠隔操縦システムの説明・注意
3. 遠隔操縦システムの操作練習
4. 遠隔物体取り出し作業実験
5. 遠隔物体取り出し作業の再体験(希望者のみ)
6. ユーザビリティ評価アンケートの記入
7. 遠隔物体取り出し実験についての解説



図 D.3: 遠隔物体取り出し実験の対象物品

(5) に示すように，実験内で作業を失敗した参加者の中で希望者のみに作業を再度体験してもらったが実験の作業時間と作業成功計測については (4) の 1 度目の作業実験の計測のみを用いた．なお本実験では実験監督者が常に遠隔物体取り出し作業を監督し，遠隔地のロボットや環境が破壊される場合やかばんを大きく動かすなど作業続行できない場合には停止した．また実験参加者の危険な操縦動作を避けるために操縦者に対して口頭のみで助言はするものとした．

D.2 遠隔物体取り出し実験結果

本実験における作業成功率と作業所要時間については，それぞれ表 D.1，表 D.2 に示す．作業成功率については，1 物品目は実験参加者全員 21 人が取り出すことに成功し，2 物品目は参加者の 81% にあたる 17 人が成功した．失敗した 4 ケースは，1 物品目を取り出した際に袋や他の物品を移動させてしまい，2 物品目を認識できなくなるケースとロボットの腕が到達できなくなるケースであった．また 2 物品目まで取り出し出来たケースにおける平均作業所要時間の 128.6 秒であった．

D.3 視聴覚統合型遠隔操縦システムのユーザビリティ評価

D.3.1 遠隔操縦物体操作システムのユーザビリティ評価アンケート

本研究では遠隔操縦物体操作システムをユーザビリティの観点から評価するための，ユーザビリティ評価アンケートの設計手法を提案する．遠隔操縦物体操作システムをユーザビリティの観点から評価するために，先行研究 [139] で提案された表 D.3 に示す遠隔操縦に関するタクソノミを参考に，遠隔操縦システムのユーザビリティに関する評価軸を設計し，その評価軸に

表 D.1: 作業成功率

	1 物品のみ作業成功	2 物品の作業成功
作業成功人数	21 人	17 人
作業成功率	100.0 %	81.0 %

表 D.2: 作業成功時の作業所要時間

	1 物品目までの所要時間	2 物品目までの所要時間
平均	87.2 秒	128.6 秒
最低	32 秒	51 秒 (2 物品同時把持)
最高	189 秒	231 秒

基づいてユーザビリティ評価アンケートを作成する。

表 D.3: 先行研究 [139] における遠隔操縦のタクソノミ

分類名	
1	Platform architecture and scalability
2	Error prevention and recovery
3	Visual design
4	Information presentation
5	Robot state awareness
6	Interaction effectiveness and efficiency
7	Robot environment / Surroundings awareness
8	Cognitive factors

D.3.2 遠隔操縦物体操作システムのユーザビリティ評価軸の設計

本研究で設計した遠隔操縦物体操作システムのユーザビリティ評価軸は表 D.4 に示す。評価軸を設計手法としては、まず先行研究 [139] の遠隔操縦に関するタクソノミ（表 D.3）から、ユーザビリティの観点から Platform architecture and scalability を除き、残りの評価軸に Impression, Operability, Reliability, Responsiveness, Complexity, Audio design, Haptic design の新たな 7 つの評価軸を追加する。それぞれの評価軸の追加理由については以下に示す。まず Impression, Operability, Reliability については、先行研究 [139] では遠隔操縦システムに対するユーザの主観的意見を評価するために追加する。同様に Responsiveness についても、先行研究 [139] では遠隔操縦のネットワーク遅延を含むシステムの応答性について考慮されてい

かったため追加する。そして先行研究 [139] では視覚インタフェースのみを対象としていたのに対して、本研究では視聴覚・振動情報を扱うため、Visual design に更に Audio design, Haptic design を追加し、視聴覚統合インタフェースの総合的な複雑さを評価するために Complexity を追加する。

表 D.4: 遠隔操縦物体操作システムにおけるユーザビリティ評価軸

評価軸名	
1	Impression: 印象
2	Operability: 操作性
3	Interaction effectiveness and efficiency: 有効性・効率
4	Reliability: 信頼性
5	Responsiveness 応答性
6	Complexity 複雑さ
7	Cognitive factors: 作業・認識の補助
8	Error prevention and recovery: 失敗回避と修正
9	Information presentation: 情報の表示
10	Visual design: 視覚デザイン
11	Audio design: 聴覚デザイン
12	Haptic design: 触覚デザイン
13	Robot state awareness: ロボットの状況の把握
14	Robot environment / Surroundings awareness: ロボットの周りの環境の把握

D.3.3 遠隔操縦物体操作システムのユーザビリティ評価アンケートの設計

前節で設計した遠隔操縦物体操作システムのユーザビリティ評価軸を元にユーザビリティ評価アンケートを設計する。設計したアンケートは表 D.5 に示す。

表 D.5: 遠隔操縦物体操作システムのユーザビリティ評価アンケートの質問内容

質問番号	質問内容
印象	
1	操縦システムは面白かった?
2	操縦システムは魅力的だった?
3	操縦システムは賢いと思った?
操作性	
4	操縦システムは使いやすかった?
5	操縦システムをうまく使いこなせた?
有効性・効率	
6	指示通り作業できた?
7	手早く作業できた?
信頼性	
8	操縦システムは信頼できた?
9	指示したとおりに動いた?
応答性	
10	操縦システムの反応はよかった?
11	操縦システムの時間の遅れに対応できた?
複雑さ	
12	この操縦システムは使うのは簡単だった?
13	他の人にとっても簡単に使える?
14	使い続ければ、使いこなせる?
作業・認識の補助	
15	操縦システムは、ロボットを動かすための手助けをしてくれた?
16	操縦システムは、状況認識の手助けをしてくれた?
失敗回避と修正	
17	操縦を間違えても簡単にやり直せた?
18	失敗せず物体をつかむことができた?
19	つかみ損ねても簡単にやり直せた?
情報の表示	
20	情報は作業するために十分だった?
21	情報が出るタイミングは正しかった?
視覚デザイン	
22	ロボットの腕の映像(腕の様子など)はわかりやすかった?
23	物体や環境の映像(物体・袋の形の変化など)はわかりやすかった?
24	画面の表示はわかりやすかった?

質問番号	質問内容
聴覚デザイン	
25	ロボットの腕の音(ロボットが動く音)はわかりやすかった?
26	物体や環境の音(ロボットが物体・袋を触る音など)はわかりやすかった?
27	警告音などの音はわかりやすかった?
触覚デザイン	
28	コントローラの振動はわかりやすかった?
ロボットの状況の把握	
29	指がどのような形になっているか, わかりやすかった?
30	腕がどのような形になっているか, わかりやすかった?
31	腕をどれくらい伸ばしているか, わかりやすかった?
ロボットの周りの環境の把握	
32	物体がどこにあるか, わかりやすかった?
33	物体がどのように置かれているか, わかりやすかった?
34	ロボットと物体の距離はわかりやすかった?

なお本ユーザビリティ評価アンケートは各質問に対して7段階評価を行う形式であり, 本研究の検証実験である遠隔物体取り出し実験のために, 遠隔操縦の中でも, 特に遠隔ロボットによる物体の把持に着目した質問を設計したものである。また表 D.5 の質問以外にも, 基本情報として性別, 年齢, ロボットに関する経験を実験参加者に記述してもらうものとする。

D.3.4 ユーザビリティ評価アンケート結果

本実験参加者によるユーザビリティ評価アンケート結果を図 D.4 に示す。図 D.4 は x 印が平均値, 横線が中央値, o 印が外れ値を示した箱ひげ図である。

図 D.4 を詳しく見ていくと, 6 番「指示通り作業できた?」に対して 7 番「手早く作業できた?」の評価平均値が低いことから, 本システムは遠隔物体取り出し作業を行えるが, 作業所要時間については改善が求められていることがわかる。また 12 番「この操縦システムを使うのは簡単だったか?」や 13 番「他の人にとっても簡単に使える?」に対して, 14 番「使い続けられれば, 使いこなせる?」の評価平均値が高いことから, 本システムには慣れが必要であると実験参加者が感じていることが見受けられる。さらに 29-31 番に対して 32-34 番の評価平均値が低いことから, ロボットの状況の把握よりもロボットの周りの環境の把握について改善すべきであることがわかる。以上の結果から本研究で提案したアンケートについて, 遠隔操縦システムのユーザビリティに関する評価と課題抽出についてある程度の効果があると考えられる。

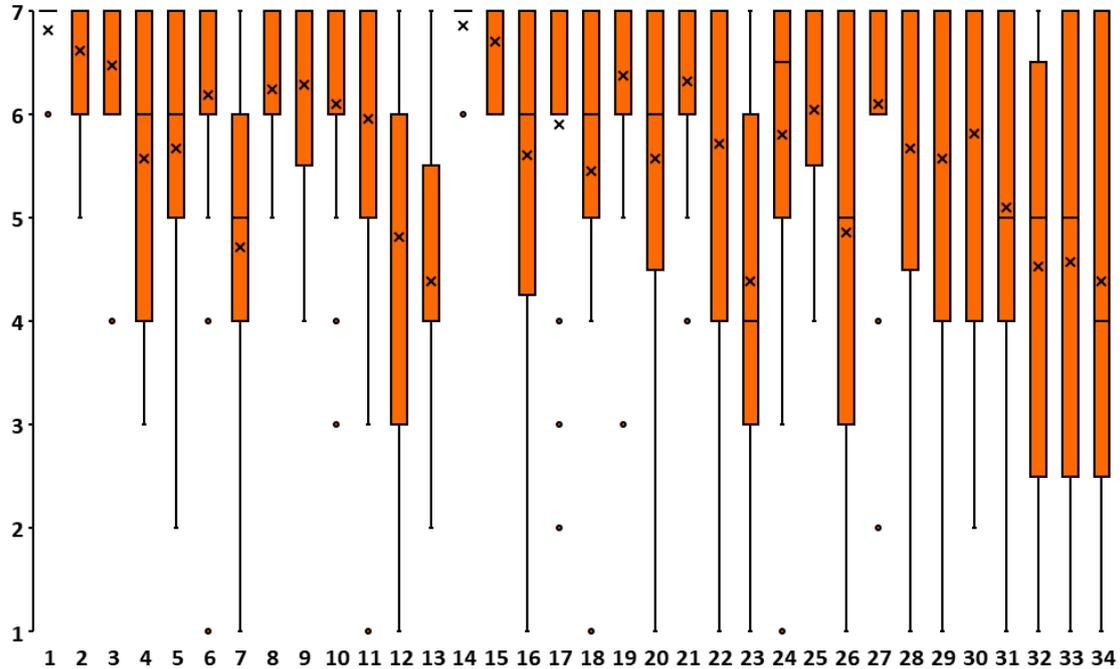


図 D.4: ユーザビリティ評価アンケート結果(質問番号は表 D.5 を参照)

D.3.5 ユーザビリティ評価アンケートの議論

本研究で提案したユーザビリティ評価アンケートでは、ユーザビリティに関する評価と課題抽出についてある程度の効果があったが、一部の質問で分散の大きな解答が得られる質問となった。特に「ロボットの周りの環境の把握」に関する質問である 23 番、26 番、32-34 番では評価平均値が低く分散が大きい結果となった。この結果の原因としては、人とロボットの身体の形状に起因するオクルージョンが大きな問題ではないかと考える。1 つ目のオクルージョンは、ロボットの腕やグリッパによるオクルージョンであり、特に物体を把持しようとする時に大きな問題になっていると考えられる。2 つ目のオクルージョン、物体と物体、物体と環境のオクルージョンであり、ロボットの視点位置姿勢を人のように大きく変更できないことに起因すると考えられる。今後の研究の課題として、ロボットの周りの環境の把握について、オクルージョンの観点からより明確に課題を抽出できるようにアンケートを設計していきたい。