

修士論文

物理的な特性と物体配置を考慮した ロボットによる容器への積み込み

安田 晃也

2019 年 3 月 15 日

奈良先端科学技術大学院大学
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

安田 晃也

審査委員：

小笠原 司 教授	(主指導教員)
杉本 謙二 教授	(副指導教員)
高松 淳 准教授	(副指導教員)
丁 明 助教	(副指導教員)
GARCIA RICARDEZ Gustavo Alfonso 助教	(副指導教員)

物理的な特性と物体配置を考慮した ロボットによる容器への積み込み*

安田 晃也

内容梗概

物流における配送容器への物体積込は自動化の求められる作業の一つである。これまでの物体積み込み計画の研究は箱詰めされた対象物の積み込みが目的であり、対象物が柔らかかったり、壊れやすかったりといった物理的な制約や配置における制約は考慮されてこなかった。また現状の物体操作の研究では衝突を避ける動作と物体を決められた姿勢に配置する動作の両立は考慮していない。本論文では、ロボットによる容器への詰め込みを実現するため、1. 多種類物体に適用可能な積み込み計画手法と 2. 詰め込み計画を実行可能なロボット動作に変換する手法の 2 つを提案する。1. では物体の配置を最適化問題として設定し、物体の種類と配置関係の制約を最適化関数に組み込み積込位置を決定する。2. では、吸引グリッパを持つロボットが最終状態で容器に接触しない条件を考慮して把持方法を決定し、直線軌道により物体を積み込む動作を生成する。提案した物体積込計画の手法を評価するため従来手法と比較する実験を行った。実験では生成された結果が設定した制約を満たした配置を行っていることを確認した。また、積込動作の手法を評価するためにロボットアームによる容器への物体積込実験を行った。11 個すべての物体に対して容器の衝突を考慮した動作生成ができ動作実行の結果環境への衝突が起きずに 10 個の積込に成功した。

キーワード

ビンパッキング, マニピュレーション, 箱への積込

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, 2019 年 3 月 15 日.

Robotic Container Loading Considering the Physical Characteristics and Relative Placement of Objects*

Akiya Yasuda

Abstract

Object packing work is required for automation. Therefore, research is necessary to plan how to load objects into a container, including the best arrangement of the objects inside the container. However, in previous research, the constraints due to the type and placement relation of the object are not considered. Also, the grasping operation is not considered when loading objects in the container. In this research, We proposed method to execute packing into containers by robot. We set the placement of the object as an optimization problem, and extend the constraints on the type and arrangement relationship of items. We proposed an algorithm to search for the grasping angle that does not collide with the container when placing an object in the container. In the loading experiment in simulation, we set three constraints and we confirmed that the optimization result is consistent with the constraints. In the loading experiment using a robot arm, we loaded 11 items into the container. We succeeded in motion generation considering collision with the container for all 11 objects and succeeded in the actual loading of 10 objects.

Keywords:

Bin packing, manipulation, box loading

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, March 15, 2019.

目 次

1. はじめに	1
1.1 背景	1
1.2 関連研究	2
1.2.1 物体の操作	2
1.2.2 積込計画	3
1.3 研究目的	3
1.4 本論文の構成	4
2. 物体積込計画	5
2.1 概要	5
2.2 従来手法で設定された最適化関数	8
2.2.1 物体重複の最適化	8
2.2.2 重心の最適化	9
2.3 物体の位置関係の取得	10
2.4 底面接触面積の最適化	13
2.5 物体の種類と配置関係の最適化	14
2.6 物体位置の最適化	15
3. 積込動作の生成手法	16
3.1 概要	16
3.2 動作生成のための角度探索	17
4. 積込物体と計測	19
4.1 積込対象物体	19
4.2 物体の位置・姿勢計測	25
5. 物体積込計画実験	27
5.1 設定した配置関係の制約	27
5.1.1 制約の設定	27

5.1.2	Rigid	28
5.1.3	Deformable	28
5.1.4	Breakable	29
5.1.5	最適化関数の設定	29
5.2	実験内容	30
5.3	実験結果	31
5.4	考察	36
6.	物体積込実験	38
6.1	実験環境	38
6.1.1	ハードウェア	38
6.1.2	ロボットアーム	38
6.1.3	エンドエフェクタ	40
6.1.4	ソフトウェア構成	43
6.1.5	運動学的到達性	45
6.2	実験内容	46
6.3	実験結果	46
6.4	考察	52
7.	おわりに	53
7.1	まとめ	53
7.2	今後の課題と展望	53
	謝辞	55
	参考文献	56
	付録	59
A.	積込動作結果	59

目 次

1	Example of item placement	6
2	Resize according to orientation of placement	7
3	Example of overlapping items	8
4	Overlapping relation	9
5	Example of items placement relation	10
6	Flowchart to know the positional relationship of items on X-coordinate	12
7	Example of non-contact area	13
8	Local coordinate system of suction gripper	16
9	Flowchart to generate loading motion	18
10	Rigid items	20
11	Deformable items	21
12	Breakable items	21
13	Items used in Amazon Robotics Challenge 2017	22
14	Example using AR markers	25
15	Constraint of rigid items	27
16	Constraint of deformable items	28
17	Constraint of breakable items	29
18	Optimized placement result by previous method	34
19	Optimized placement result by proposed method	35
20	Experimental setup	39
21	End effector for item manipulation	41
22	Software configuration	44
23	Collecting kinematics reachability map	45
24	Result of loading Speed Stick	50
25	Result of loading Tissue Box	51
26	Result of loading Baby Wipes	59
27	Result of loading Powder pouch	60
28	Result of loading Canned food	61

29	Result of loading Ball Container	62
30	Result of loading Liquid detergent	63
31	Result of loading Coffee bottlet	64
32	Result of loading Expo Eraser	65
33	Result of loading Snack A	66
34	Result of loading Snack B	67

表 目 次

1	Items size and weight	23
2	Items type	24
3	Computer specifications	30
4	Constant condition	31
5	Weights of optimized function	31
6	Optimization result by previous method	32
7	Optimization result by proposed method	32
8	Items color	33
9	Specification of the robot arm	39
10	Specification of joint motor	42
11	Specification of vacuum pump	42
12	Result of experiment loading motion generation	48
13	Result of experiment loading motion generation	49

1. はじめに

1.1 背景

物流における倉庫内作業には，入荷のための商品の棚入れや出荷のためのピッキング・梱包など数多くの軽作業が存在する．しかし近年，電子商取引の市場増加に伴い，取り扱う商品数や入荷数・出荷数は増大し続けている．さらに，倉庫内作業者の人手不足が深刻化している [1]．そこで，これらの作業の支援，自動化が求められている．

作業を支援するロボットとして実用化に至っている事例に，Amazon.com 社が開発したピッキング作業を支援するために商品を棚ごと運搬するロボットである Amazon Robotics [2] など，棚入れやピッキングの際に必要な商品棚までの移動を代わりに担うことで作業者の負担を軽減するロボットが挙げられる．また，自動化の事例として MUJIN と ASKUL が共同で開発したピッキング工程を教示なしで自動で行うロボット [3] などが挙げられる．

また，商品のピッキングを自動化するためのロボットの研究は数多く行われている．Amazon.com 社は物流倉庫の自動化を競うロボット競技大会 Amazon Robotics Challenge 2017(以下 ARC2017) を開催した．棚から商品を取り出して運搬用ボックスへ移動する pick タスクと，運搬用ボックスから商品を取り出し棚に収納する stow タスクを取り上げ，さらに物流倉庫では日々新商品が追加されることを考慮し一部の商品が競技直前に提供された．この大会を通して，多種物体が無造作に置かれた状態で検出・識別し把持を行う研究や新規物体の検出のための高速学習を行うための研究に取り組まれた [4]．

これらの研究では，様々な物体の検出や姿勢の認識，および物体の安定した把持に焦点があたっている．しかし，多種物体を顧客に郵送するための容器に積込・梱包を行う作業は未だ実用化されていない．倉庫作業の全自動化のためにはピッキングされた商品をロボットにより容器に積込・梱包する必要がある．

1.2 関連研究

1.2.1 物体の操作

ロボットアームによる作業の自動化を目的とした物体操作を行うための研究がいくつか行われている。ロボットによる物体を把持する研究として Jeffrey らは並行グリップでのロバストな把持実現のため、テスト用の把持物体の 3D モデルとそこからシミュレーション上で合成された把持データセットから学習した Grasp Quality Convolutional Neural Network を用いて物体の深度画像から把持に最適なグリップ位置を予測することにより、データセットにない新規の物体の把持に成功している [5]。また、この手法は把持のための接触点が 1 つのみである吸引グリップにも適用されている [6]。

ロボットによって物体を配置する研究として、部品の組み立てを目的とした研究が行われている。Guan らは組立作業において把持物体と環境との接触が許された状況下で、物体の環境接触を考慮した把持物体と同じ幅を持つ空間への挿入を実現している [7]。一方で不規則形状の物体配置についても研究が行われている。Fadri らはロボットハンドで不規則な形状の石を垂直に積み上げるために、石の 3D モデルを用いて事前に石同士の接触をシミュレーション上で探索し、石同士の接触面積や重心距離から最も安定した石の積上姿勢を予測することで、天然の石灰石を積み重ねることに成功している [8]。

これらの研究では多くの種類の物体の把持や配置を行っており、RGB-D カメラによる認識や物体の 3D モデルによる事前シミュレーションによって物体をロバストに把持・配置している。しかし、積込に於ける容器との衝突を避ける動作が必要だが、これらの研究による把持・配置の一連の動作は環境への衝突を考慮されていない。これらの研究に使用されているグリップは把持した状態で物体の姿勢を変えることはできないため、把持した時点で環境と衝突せずに容器内に配置できる物体の姿勢が限られてしまう。このため、物体の配置姿勢を考慮せずに物体の把持を行った場合容器への衝突を避けつつ指定された姿勢で物体を配置することが困難であると考えられる。

1.2.2 積込計画

物体積込作業のロボットによる自動化には，容器に積込む物体の姿勢や位置をあらかじめ決定することが望ましい．箱やコンテナなど限られた領域への物体の積込計画は3次元ビンパッキング問題として研究の対象となっている．ビンパッキング問題では前提条件として取り扱う物体のサイズが既知であり，物体および容器を直方体として近似し配置計算を行う．さらに，積込容器が単一のみか少種・多種類から最適なものを選択するか，また積込物体が少種・多種類であるかによって，6通りの副問題に分類される [9]．倉庫内作業における物体積込は，数種類の段ボール箱に対して大量に扱われている商品を積み込むため複数ビンサイズビンパッキング問題として扱える．解法の一つとして，岩澤らは積み込む直方体名を並べた三つの順列内での直方体の並び順に対して直方体間に上下，左右，前後の位置関係を割り当てる Sequence-Triple という表現法を用いて局所探索法により配置物体の体積最小化に取り組んでいる [10]．

しかしこれらは物体の大きさ，重さおよび直方体での積載強度による制約を考慮しているが，直方体以外の積載強度や，物理的特性による物体の配置の制約や，物体の特性による配置関係の制約は考慮していない．積み込むべき物体には固形物や柔軟物などの多様な物体が存在する．この際，大きさや重さのみならず物体の種類による制約が発生する．例えば，缶やビンなどといった固い物体とスナック菓子のような壊れやすい物体とを隣合わせに置いた場合，運搬の際破損する恐れがある．多様な特性を持った物体らを積み込む際はこういった制約を満たした配置を行うべきである．

1.3 研究目的

本研究ではロボットによって，物体の容器への積込を行うことを目的とする．本研究では物体積込計画とロボットによる容器への積込動作生成に分割して取り組む．積込対象となる物体の一覧から，物体積込計画によって容器内での各物体の姿勢および3次元位置を決定し，それをもとにロボットによって容器との衝突を回避しながら物体を積み込む．

物体積込計画では物体を直方体に近似されたものとして扱い，物体の配置を最適化問題として設定し物体の種類や特性と配置関係の制約を最適化関数に組み込むことで制約を満たした物体の配置を検討する．ロボットによる積込動作生成では，吸引グリップを持つロボットが容器内に物体を配置した際に環境と衝突しないロボットの姿勢を探索しその結果から物体の把持角度を求めることで，容器と衝突せずに指定された姿勢で物体を配置する動作を生成する．容器に積み込む時の環境との衝突回避を考慮した動作生成にはロボットアームの逆運動学とモーションプランナーを使用する．物体の積込時に容器は既知の位置に固定されているものとする．

本研究では ARC2017 にて取り上げられた物体や通販サイトにて販売されている日用品に対して箱への積込計画を行い，実際にロボットによって容器への積込を行うことで提案手法の有効性を検証する．

1.4 本論文の構成

本論文の構成は以下のようになっている．2章では物体積込計画の手法について述べる．3章ではロボットアームが環境に衝突せずに容器に物体を積み込むための把持角度探索の方法について述べる．4章では本研究で使用した物体とその姿勢認識の方法について述べる．5章では提案した積込計画の比較実験とその結果および考察を述べる．6章ではロボットによる物体積込実験と結果および考察について述べる．最後に7章にて本研究についてまとめるとともに，今後の課題と展望について述べる．

2. 物体積込計画

2.1 概要

物体積込計画では積込対象となる物体の一覧から，物体積込計画によって容器内での各物体の姿勢および3次元位置を決定する．その際に，物体の物理的な特性による配置の制約を満たす積込計画を行うために物体の物理的特性の分類を，物体の種類として定義し種類ごとの配置関係の制約の設定を行う．3次元ビンパッキング問題の解法として，物体位置を最適化問題として扱う手法 [11] を使用する．各物体 ($i = 1, 2, 3, \dots, N$) の3次元中心位置 $P_i = (x_i, y_i, z_i)$ に対して評価する最適化関数を設定し，最適化問題として各物体の位置を求める．最適化関数は次式で与えられる．

$$f(P_1, P_2, \dots, P_N) = F \quad (1)$$

物体の配置に対する評価項目は複数存在するため，上式による最適化関数は複数の関数を総和し導出される．従来手法では物体同士の重複体積の最適化関数と全物体の重心の最適化関数が設定された．

しかし，この二つのみでは物体の安定した積載を考慮するための制約や物体の特性による配置関係の制約は考慮することができない．本研究ではそれに加え，物体を積み重ねた際に下の物体と接触していない底面の面積を最小化するための最適化関数と，物体の種類と位置関係との制約を満たすための最適化関数を拡張する．様々な種類の物体の混合により発生する積込の制約を，物体の種類をパラメータとして与え配置関係による制約を最適化関数に組み込み評価値の一つとして考慮することで制約を満たした物体の配置を検討する．

本研究での関数の最適化には粒子群最適化法 [12] を用いた．関数の最適化手法には関数の勾配を利用し最大値，最小値を求める手法が多い．しかし，本研究で設定した最適化関数は物体の位置関係により非連続的に変化するため，勾配を用いた最適化を行うことができない．それに対し，粒子群最適化法は評価値の優劣が定量的であれば連続型の最適化問題でなくとも適用できる [13] ため採用した．

物体配置のイメージ図を Fig. 1 に示す．各物体 i はサイズ (W_i, D_i, H_i)，重量 (m_i) および配置方向のパラメータを持つ．配置方向は Fig. 2 のような回転を持た

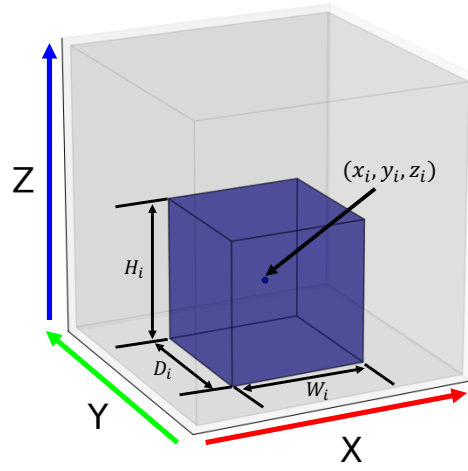


Fig. 1: Example of item placement

ない6通りのうちのいずれかが選択され、容器内における物体の3方のサイズが変更される．さらに物体の位置関係と種類による制約を考慮するため、各物体に種類のパラメータを与える．以上を容器内の最適な物体の中心位置 (x_i, y_i, z_i) を求める際の固定パラメータとして、最適化関数の評価が最も高い時の各物体位置を求める．

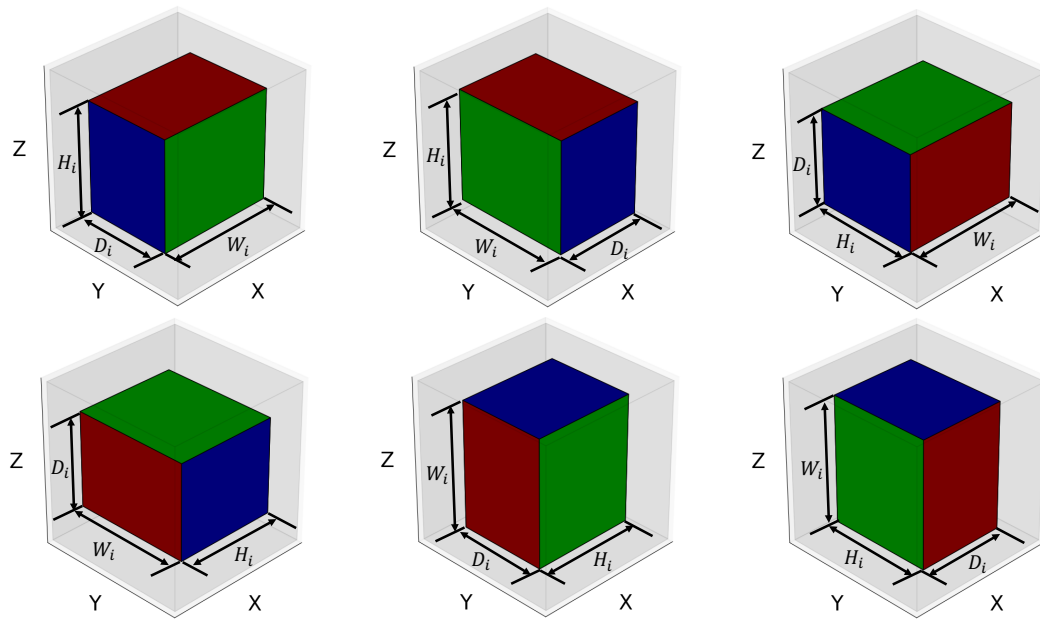


Fig. 2: Resize according to orientation of placement

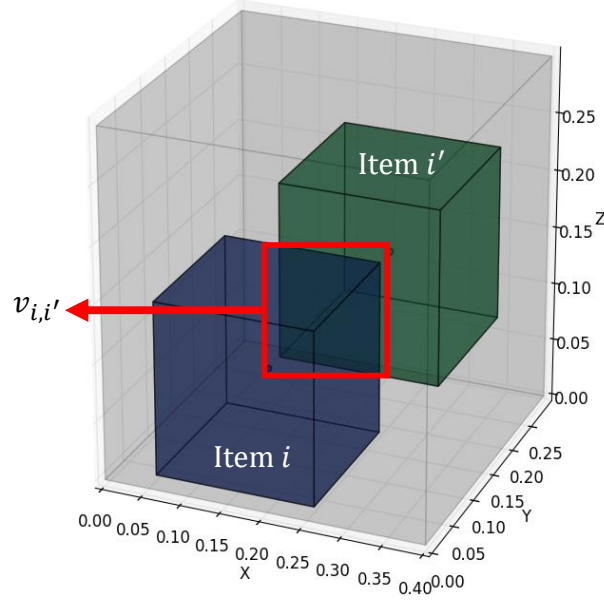


Fig. 3: Example of overlapping items

2.2 従来手法で設定された最適化関数

2.2.1 物体重複の最適化

物体積込計画において，Fig. 3のような物体同士の体積重複が起きないように配置することは最も重要である．各物体の体積重複が無いとき最大化される最適化関数を設定する．2つの物体 i および i' について，各次元での重複長さを計算する．X 軸を例に取った時，一次元での重複はFig. 4のようになり，各物体の位置とサイズによって重複長さ $ol_{i,i'}^x$ は以下のように表せる．

$$ol_{i,i'}^x = \begin{cases} 0 & (d_{i,i'}^x \geq (W_i + W_{i'})/2) \\ \min\{W_i, W_{i'}\} & (d_{i,i'}^x < |1/2(W_i - W_{i'})|) \\ 1/2(W_i + W_{i'}) - d_{i,i'}^x & (\text{otherwise}) \end{cases} \quad (2)$$

各次元の重複長さの積が2つの物体の重複体積 $v_{i,i'}$ となる．

$$v_{i,i'} = ol_{i,i'}^x \times ol_{i,i'}^y \times ol_{i,i'}^z \quad (3)$$

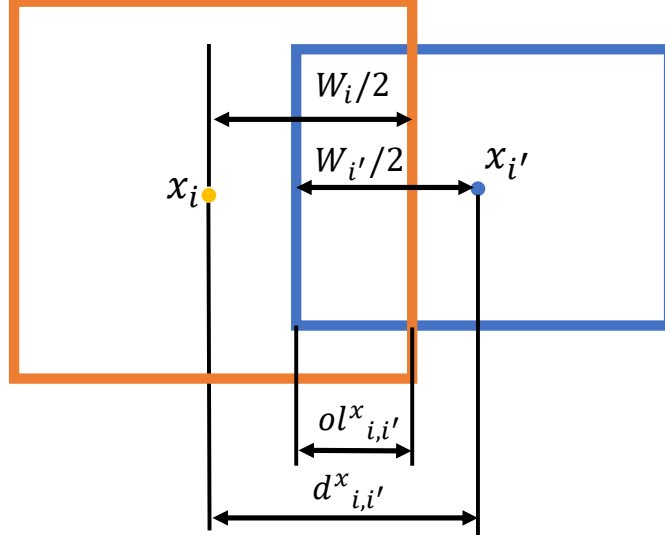


Fig. 4: Overlapping relation

ここで、3次元すべて重複した場合のみ重複体積が発生する．これをすべての2つの物体の組合せについて計算し体積を総和する．

$$V_{\text{overlap}} = \sum_{i=1}^{N-1} \sum_{i'=i+1}^N v_{i,i'} \quad (4)$$

よって重複最適化関数は以下に設定できる．ここで $\epsilon_{\text{overlap}}$ はゼロ除算を避けるための微小数である．

$$F_{\text{overlap}} = \frac{1}{V_{\text{overlap}} + \epsilon_{\text{overlap}}} \quad (5)$$

2.2.2 重心の最適化

複数の物体を同一領域内に配置した際、全物体の重心が底面の中心 $C = ((x, y, z) = (0, 0, 0))$ に近いほど安定した配置である．全物体の重心 G は

$$G = (\bar{x}, \bar{y}, \bar{z}) = \left(\frac{\sum_i^N x_i m_i}{\sum_i^N m_i}, \frac{\sum_i^N y_i m_i}{\sum_i^N m_i}, \frac{\sum_i^N z_i m_i}{\sum_i^N m_i} \right) \quad (6)$$

であり，底面の中心とのユークリッド距離

$$d_{\text{cog}} = \|G - C\| \quad (7)$$

が小さいほど高くなる最適化関数を設定する．ここで ϵ_{cog} はゼロ除算を避けるための微小数である．

$$F_{\text{cog}} = \frac{1}{d_{\text{cog}} + \epsilon_{\text{cog}}} \quad (8)$$

2.3 物体の位置関係の取得

後の2.4節および2.5節で述べる最適化関数には，各物体同士の隣接する位置関係が必要である．そこで，各物体 i について他の物体が Fig. 5 に示す6通りの隣接する位置関係いずれかに当てはまるかを確認する．物体 i と物体 i' について，X軸方向の配置関係を確認する際のフローチャートを Fig. 6 に示す．まず，2.2.1節で求めた重複長さにおいて，X以外の座標での重複長さを持つか確認する．Y,Z座標両方の重複長さを持つ場合，2つの物体はX軸方向に重なっているため右方

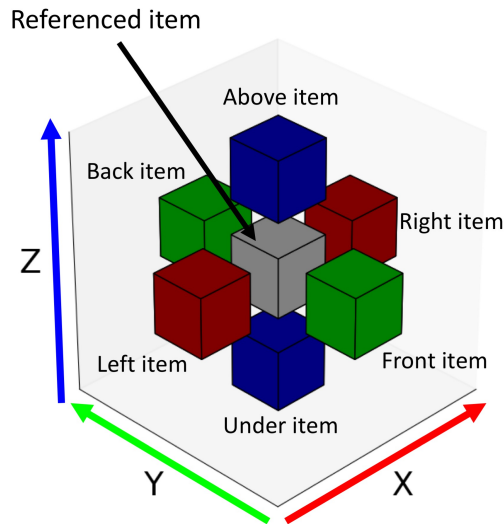


Fig. 5: Example of items placement relation

向および左方向に隣接している可能性が有る．次に 2 つの物体が隣接しているかを確認する．X 座標において 2 つの物体の間に他の物体が配置されているか確認する．存在しない場合は，X 座標の大小関係から物体同士が右方向もしくは左方向に隣接するかを確認する．存在する場合は，その物体が物体 i および物体 i' に対して Y,Z 座標両方の重複長さを持つかを確認する．確認したすべてに重複長さを持つ場合，その物体は X 軸方向に対して，物体 i および物体 i' の間に配置されているため，2 つの物体は隣接していないことになる．いずれかに重複長さを持たなかった場合は，X 座標の大小関係から物体同士が右方向もしくは左方向に隣接するかを確認する．この確認を Y 軸，Z 軸方向でも確認し，各物体の隣接する配置関係を取得する．

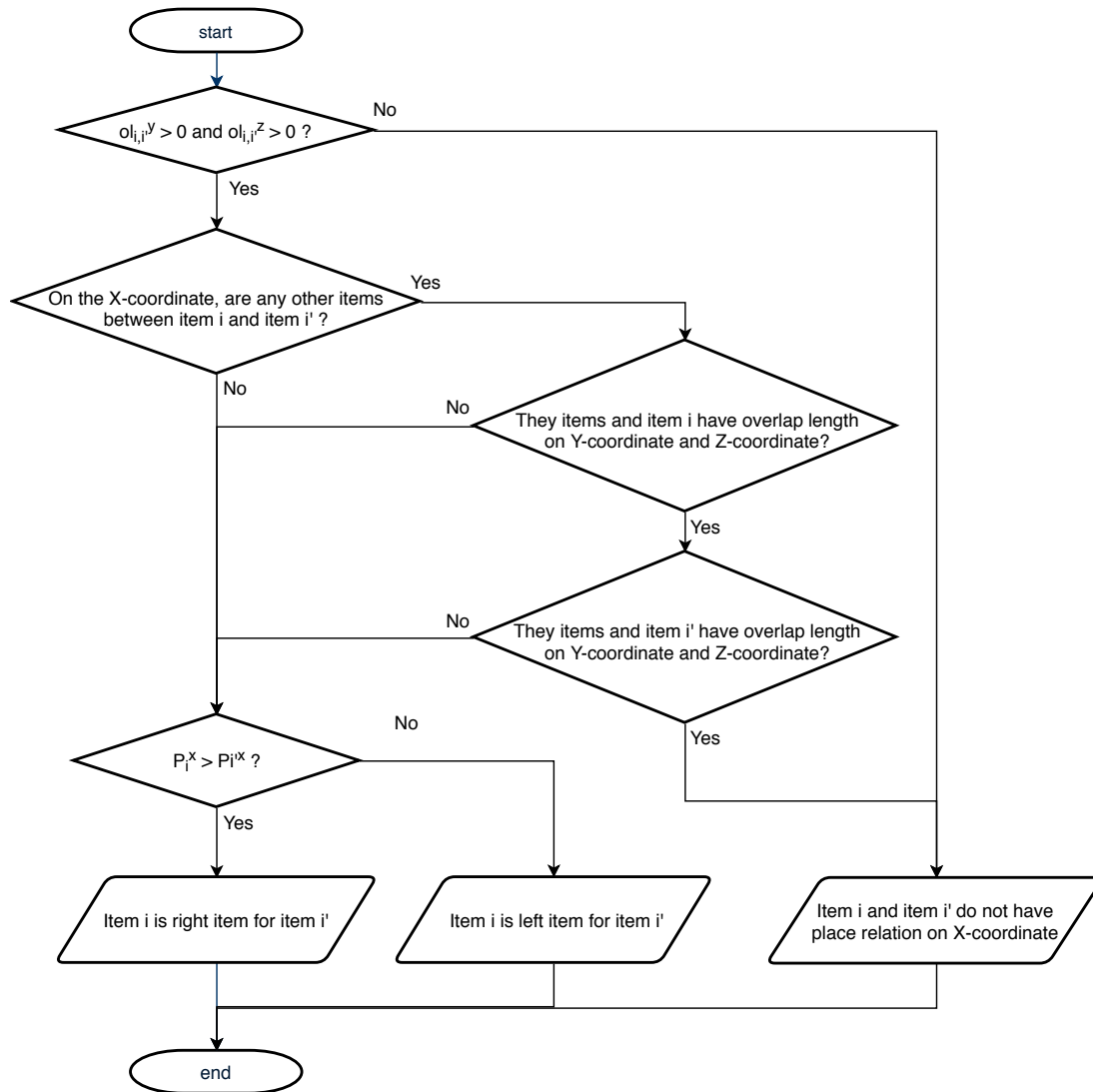


Fig. 6: Flowchart to know the positional relationship of items on X-coordinate

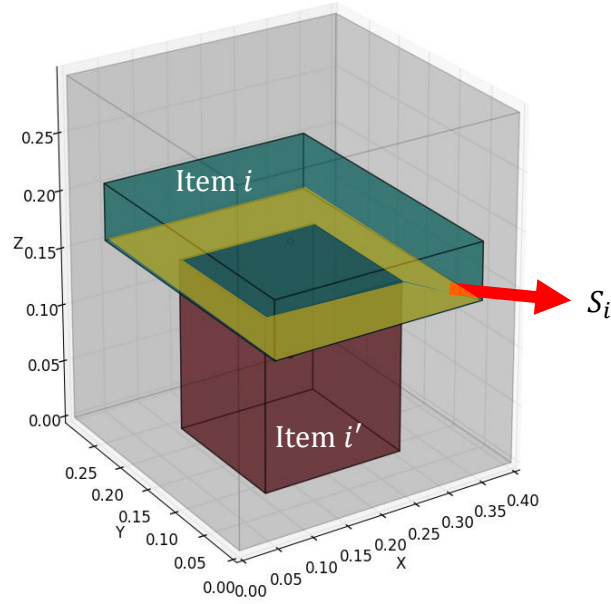


Fig. 7: Example of non-contact area

2.4 底面接触面積の最適化

物体積込計画において，Fig. 7のように物体を積載する配置を取る際，下部との接触していない面積（以下非接触面積） S_i は小さいほど物体の安定した積載が可能になる．各物体の非接触面積の合計が小さいほど最大化する最適化関数を設定する．各物体 i について，下方に他の物体が存在しない場合，非接触面積 s_i は

$$s_i = \begin{cases} 0 & (z_i = H_i/2) \\ W_i \times D_i & (\text{otherwise}) \end{cases} \quad (9)$$

となる．一方で，下方に物体 i' が存在する場合，非接触面積はX軸とY軸上での重複長さから以下で求めることができる．

$$s_i = \begin{cases} (W_i \times D_i) - (ol_{i,i'}^x \times ol_{i,i'}^y) & (z_i - H_i/2 \leq z_{i'} + H_{i'}/2) \\ W_i \times D_i & (\text{otherwise}) \end{cases} \quad (10)$$

よって各物体の非接触面積の合計は

$$S_{\text{contact}} = \sum_i^N s_i \quad (11)$$

であり、最適化関数は以下に設定できる．ここで $\epsilon_{\text{contact}}$ はゼロ除算を避けるための微小数である．

$$F_{\text{contact}} = \frac{1}{S_{\text{contact}} + \epsilon_{\text{contact}}} \quad (12)$$

2.5 物体の種類と配置関係の最適化

複数の種類の物体が混合して配置する場合に発生する配置関係の制約 $k (= 1, 2, \dots, K)$ を定義し、その制約を守った場合に最大化される最適化関数を設定する．最適化問題において、制約条件は最適化関数と別で設定する必要がある．また、この関数は物体の位置関係により非連続的に変化するため、最適化問題として解く難易度が非常に高くなる．しかし本研究で取り扱う制約は、物体同士の位置関係の確認と種類の確認といった2段階での確認が必要ゆえに複雑であるため最適化関数に組み込むことで対応した．

物体の種類と配置関係の制約について、以下を例に挙げるができる．

- 物体 i が種類 A の場合：物体の右側に他の物体の配置を禁止
- 物体 i が種類 B の場合：物体の下に種類 C の物体の配置を禁止

制約 k に反した場合に a_k だけ増大する係数 c_{ki}

$$c_{ki} = \begin{cases} a_k & (\text{if constraint violation}) \\ 0 & (\text{otherwise}) \end{cases} \quad (13)$$

を定義する．各制約 k の確認を各物体に対して行い、総和する．

$$C = \sum_k^K \sum_i^N c_{ki} \quad (14)$$

よって、最適化関数は以下に設定できる．ここで ϵ_{type} はゼロ除算を避けるための微小数である．

$$F_{\text{type}} = \frac{1}{C + \epsilon_{\text{type}}} \quad (15)$$

2.6 物体位置の最適化

以上の各最適化関数と重み $\omega_l (l = 1, 2, 3, 4)$ の積をすべて足したものが最終的な最適化関数となる.

$$F = \omega_1 F_{\text{overlap}} + \omega_2 F_{\text{cog}} + \omega_3 F_{\text{contact}} + \omega_4 F_{\text{type}} \quad (16)$$

ここで, 最適化の結果物体同士の重複が発生した場合, 積込の実行が不可能になるため ω_1 は他の最適化関数と比べ大きく設定する必要がある.

3. 積込動作の生成手法

3.1 概要

積込動作は対象物体を把持しつつ容器内の目標位置に配置する動作である．この動作は容器の上から目標位置への物体の運搬中，物体は姿勢を変更せず直線的な軌道を描いて配置されることが望ましい．そのため物体を把持するロボットアームも同様の軌道が必要である．容器の中央や上部への積込のための直線動作はロボットが十分な自由度を持っていれば可能である．しかし，容器の下部や隅に配置する場合，容器への衝突を避けながらの直線動作が求められるため動作生成が困難である．

本研究では吸引グリッパでの積込動作について扱う．吸引グリッパのローカル座標系を Fig. 8 に示す． E_X 軸， E_Y 軸， E_Z 軸の角度をそれぞれ θ_x ， θ_y ， θ_z とする．吸引グリッパが物体を把持する場合，把持点の表面に対して法線方向に吸着する必要がある．そのため，物体の把持点に対して， θ_x および θ_y はほぼ決定される．しかし θ_z は把持点の表面に対して法線方向であるため，ロボットの運動学に問題がない範囲で θ_z は自由に決定できる．

本研究では物体の把持点に対し，容器への衝突を考慮した把持角度 θ_z を探索す

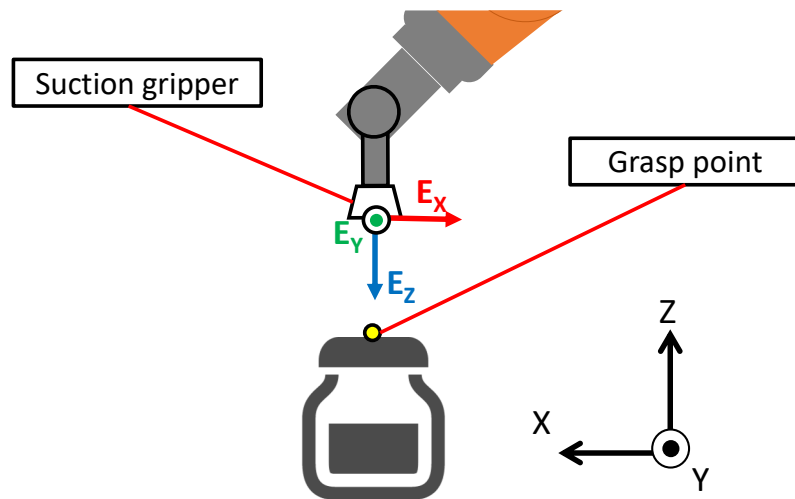


Fig. 8: Local coordinate system of suction gripper

ることで把持・容器への積込動作を生成する．目標点に対する環境との衝突回避を考慮した動作生成にはロボットアームの逆運動学とモーションプランナーを使用する．本論文では物体の姿勢を変更するための物体操作は取り扱っていない．

3.2 動作生成のための角度探索

フローチャート図を Fig. 9 に示す．容器の座標系と衝突回避用のモデルがモーションプランナーに追加された状態で，積込対象の物体を検出する．そして把持が可能な点を検出し， $\theta_z = 0$ で把持した場合の容器への物体積込が完了したときの姿勢を，目標点として生成し衝突を考慮した逆運動学を計算する．

逆運動学が解けた場合，次に積込の動作計画を行う．生成した目標点を元に，容器の上から目標点への直線動作を計画する．動作計画に成功した場合，その θ_z で物体を把持した場合，容器への衝突なしで物体を積み込むことができる．最後に，把持した後容器の上まで運搬する動作計画を行う．この時運搬中に物体の姿勢が崩れないように経路制約をかける．いずれかの動作計画が失敗した場合， θ_z をステップ角度分加算し再度物体配置時の逆運動学の計算から行う．以上を繰り返し，すべての動作計画が成功した時の θ_z で物体を把持し，直線動作で容器に積込む． θ_z を全周($\theta_z \geq \pi$)した場合，その把持点での容器への積込動作はできないため，新しい把持点を探索する．以上を繰り返し，対象物体に対して容器への積込が可能な θ_z を探索する．

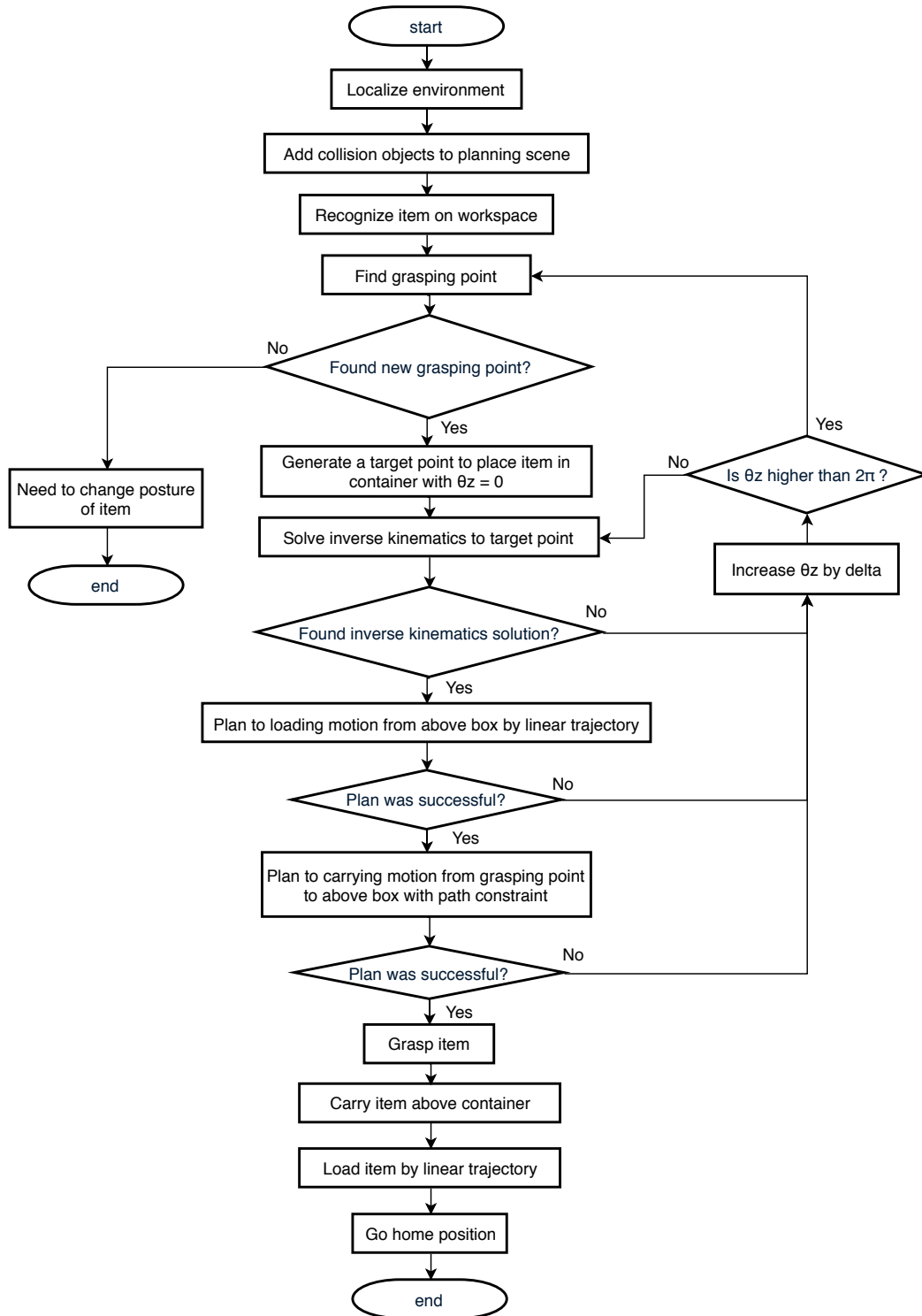


Fig. 9: Flowchart to generate loading motion

4. 積込物体と計測

本研究では容器に積み込む物体をあらかじめ指定する必要がある。また、第2章で述べた物体積込計画には各物体のサイズ、重量、種類の情報が必要であり、第3章の積込動作生成には物体の検出と姿勢の認識および把持点の情報が必要である。本章では、後述される実験で使用した物体について説明する。物体の検出と姿勢の認識には AR マーカーと RGB-D センサを使用する。

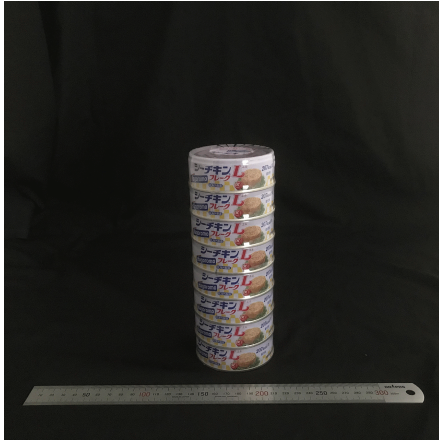
4.1 積込対象物体

今回パラメータとして与えた物体の種類を以下に示す。

- Rigid : 固い物体
- Deformable : 表面に大きな変形を伴う物体
- Breakable : 壊れやすい物体
- Other : 上記以外の物体

本研究で取り扱う物体として、物体の種類 Rigid, Deformable, Breakable のいずれかに当てはまる物体と、当てはまらない物体を分けて選定した。種類を Rigid として使用した物体を Fig. 10 に示す。Rigid として扱う物体には缶詰 (Fig. 10a) や瓶 (Fig. 10b) といった固い物体を選定した。種類を Deformable として使用した物体を Fig. 11 に示す。Deformable として扱う物体にはパウチ製品を選定し、中身が液状 (Fig. 11a) や粉状 (Fig. 11b) であり、姿勢や積載によって大きな変形を伴う物体を選定した。種類を Breakable として使用した物体を Fig. 12 に示す。Breakable として扱う物体には中身が壊れやすいスナック菓子を選定した。また、「SnackB (Fig. 12b)」のみ、配置姿勢や物体を積み重ねた際の表面の状態が大きく変わるため、Deformable としても扱った。上記のどの種類にも当てはまらない、種類が Other である物体として ARC2017 にてピッキング対象物体として事前に提供された物体を使用した。外観図を Fig. 13 に示す。

本研究では通販における郵送用の容器への積込を対象とするため、対象物体は Amazon ジャパンのウェブサイト [14] にて購入可能なものを選定した。本研究で



(a) Canned food



(b) Coffee bottle

Fig. 10: Rigid items

使用する物体は後述するロボットのグリッパによって把持が可能であることを確認している．物体積込計画時に必要な各物体のサイズおよび重量は，事前に実測し物体の種類とともにデータセットとして登録し計画時に物体名から各情報を引用する．Table 1 に本研究で取り扱った物体のサイズ，重量および種類の一覧を示す．



(a) Liquid detergent



(b) Powder pouch

Fig. 11: Deformable items



(a) Snack A



(b) Snack B

Fig. 12: Breakable items



(a) Baby Wipes



(b) Expo Eraser



(c) Ball Container



(d) Speed Stick



(e) Tissue Box

Fig. 13: Items used in Amazon Robotics Challenge 2017

Table 1: Items size and weight

Item name	Width[mm]	Depth[mm]	Height[mm]	Mass[g]
Canned food (Fig. 10a)	77	77	196	681
Coffee bottle (Fig. 10b)	90	90	175	570
Liquid detergent (Fig. 11a)	155	80	235	784
Powder pouch (Fig. 11b)	140	70	200	485
Snack A (Fig. 12a)	150	240	42	181
Snack B (Fig. 12b)	170	52	250	138
Baby Wipes (Fig. 13a)	230	115	65	373
Expo Eraser (Fig. 13b)	135	55	35	20
Ball Container (Fig. 13c)	80	80	220	220
Speed Stick (Fig. 13d)	65	60	144	272
Tissue Box (Fig. 13e)	110	110	130	168

Table 2: Items type

Item name	Item type
Canned food (Fig. 10a)	Rigid
Coffee bottle (Fig. 10b)	Rigid
Liquid detergent (Fig. 11a)	Deformable
Powder pouch (Fig. 11b)	Deformable
Snack A (Fig. 12a)	Breakable
Snack B (Fig. 12b)	Breakable and Deformable
Baby Wipes (Fig. 13a)	Other
Expo Eraser (Fig. 13b)	Other
Ball Container (Fig. 13c)	Other
Speed Stick (Fig. 13d)	Other
Tissue Box (Fig. 13e)	Other

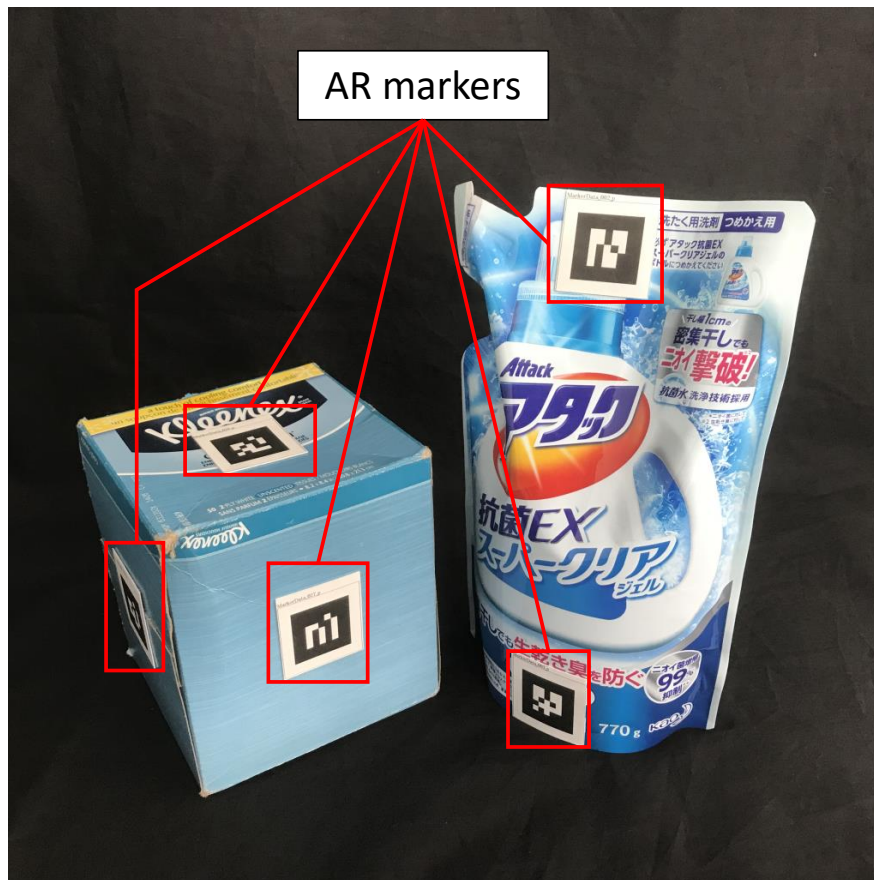


Fig. 14: Example using AR markers

4.2 物体の位置・姿勢計測

Fig. 14 のように物体にマーカーを張り付け、それを RGB-D センサで検出することで姿勢を認識する．物体の中心にローカル座標系を設定し，その座標系におけるマーカーの 3 次元位置 $P_o = (x_o, y_o, z_o)$ と 3 次元回転姿勢 $R_o = (\theta_{ox}, \theta_{oy}, \theta_{oz})$ から，物体の座標系におけるマーカーの変換行列 T_{om} を求める．

$$T_{om} = f(P_o, R_o) \quad (17)$$

RGB-D センサがマーカーを検出し，ロボット座標系におけるマーカーの 3 次元位置 $P_r = (x_r, y_r, z_r)$ と 3 次元回転姿勢 $R_r = (\theta_{rx}, \theta_{ry}, \theta_{rz})$ から，ロボット座標系

におけるマーカーの変換行列 T_r を下記の通り求め,

$$T_r = f(P_r, R_r) \quad (18)$$

物体座標系におけるマーカーの変換行列の逆行列をかけることでロボット座標系における物体の変換行列 T_{ro} を求める.

$$T_{ro} = T_{om}^{-1} T_r \quad (19)$$

求まった T_{ro} から, ロボット座標系における物体の3次元位置 $P = (x, y, z)$ と3次元回転姿勢 $R = (\theta_x, \theta_y, \theta_z)$ を導出することでロボットに対する物体の姿勢を求める.

5. 物体積込計画実験

従来手法と比べ、提案手法が2.4節および2.5節にて拡張した制約を考慮した物体配置を行っているか比較を行う。本研究で採用した物体の種類ごとの制約を定め、設定した最適化関数を用いて物体位置の最適化を行う。

5.1 設定した配置関係の制約

5.1.1 制約の設定

本論文では4.1節で設定した物体の種類について、以下の制約条件を設定した。

- Rigid : 物体の下側に他の物体の配置を禁止
- Deformable : 物体の上側に他の物体の配置を禁止
- Breakable : 前後左右および下に Rigid 物体を配置を禁止

以下に詳細を記載する。



Fig. 15: Constraint of rigid items



Fig. 16: Constraint of deformable items

5.1.2 Rigid

Fig. 15 のように，重量にかかわらず固い物体を他の物体に積載した場合，運搬中に配置が不安定になり他の物体を破損させる恐れがある．物体 i の種類が Rigid の場合，他の物体が下方に存在しないか確認する．結果によって係数

$$c1_i = \begin{cases} 3 & (\text{There are other items under item } i) \\ 0 & (\text{otherwise}) \end{cases} \quad (20)$$

を定義する．

5.1.3 Deformable

Fig. 16 のように，変形可能な物体の上に別の物体を積載した場合，ロボットによる積込時に配置が不安定になり他の物体の積込を妨げる恐れがある．物体 i の種類が Deformable の場合，他の物体が上方に存在しないか確認する．結果によって係数

$$c2_i = \begin{cases} 1 & (\text{There are other items above item } i) \\ 0 & (\text{otherwise}) \end{cases} \quad (21)$$

を定義する．

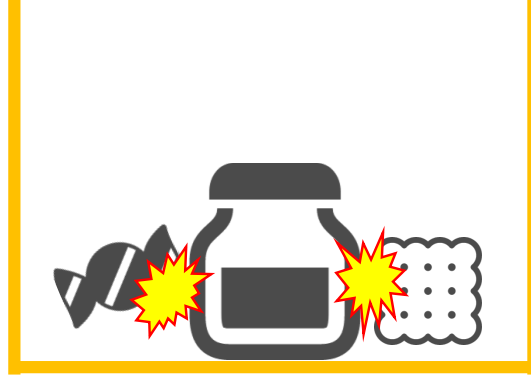


Fig. 17: Constraint of breakable items

5.1.4 Breakable

Fig. 17 のように，固い物体の前後左右に脆い物体を配置した場合，運搬中に固い物体が脆い物体を破損させる恐れがある．物体 i の種類が Breakable の場合，Rigid 物体が前後左右および下に存在しないか確認する．結果によって係数

$$c3_i = \begin{cases} 3 & \text{(There are Rigid objects in front, back, left, right or bottom)} \\ 0 & \text{(otherwise)} \end{cases} \quad (22)$$

を定義する．

5.1.5 最適化関数の設定

以上の制約確認を各物体に対して行い，総和する．

$$C = \sum_i^N (c1_i + c2_i + c3_i) \quad (23)$$

以上を物体の種類と配置関係の最適化関数（式 (15)）に使用した．

5.2 実験内容

実験手順を示す．

1. 各物体名からサイズ，重量，種類のパラメータを取得する
2. 積込対象の物体の配置方向を決定する
3. 各物体位置の最適化を行う．
4. 各物体の重複体積 V_{overlap} が 0 の場合，実行可能な積込計画としてリストに追加する
5. 2 から 4 を全配置方向の組合せについて繰り返す
6. リスト内を最適化関数の評価値順にソートする

以上を従来手法 [11] と提案手法両方で行い，最適化関数の評価値上位 10 個の物体配置結果を比較する．容器サイズは $320 \times 270 \times 360[\text{mm}]$ とする．物体の配置方向の組み合わせは最大で 6 の物体の個数乗存在する．例えば 6 方向に配置可能な物体が 2 つ積み込む場合，配置方向の組合せは 36 通り存在する．今回は計算時間省略のため各物体に対してあらかじめ安定した配置が可能な方向のみを指定した．さらに容器内の 3 次元サイズが変わらない配置方向については省略した．それにより，6144 通りの配置方向の組み合わせについて計算を行った．使用した計算機の概要を 3

各物体の配置方向の組合せは従来手法と提案手法の実験において同じ組合せで行う．各最適化関数は評価値が 0 から 1 になるように正規化した．本実験では各最適化関数の微小数を Table 4 に，最適化関数の重みを Table 5 のように設定した．これらは実験的に求めた．

Table 3: Computer specifications

PC	ThinkPad T460s (Lenovo)
OS	Ubuntu 16.04
CPU	Intel(R) Core i7-6600U
Programming language	Python 2.7

Table 4: Constant condition

Parameter	$\epsilon_{\text{overlap}}$	ϵ_{cog}	$\epsilon_{\text{contact}}$	ϵ_{type}
Value	0.000001	0.001	0.001	1

Table 5: Weights of optimized function

Parameter	ω_1	ω_2	ω_3	ω_4
Value	10000000	800000	500000	300000

5.3 実験結果

従来手法の結果を Table 6 に，提案手法の結果を Table 7 に示す．また，従来手法にて最も評価値の高かった配置結果を Fig. 18 に，提案手法にて最も評価値の高かった配置結果を Fig. 19 に示す．各物体の配色を Table 8 に示す．提案手法での最適化結果について 2.4 節で設定した底面接触面積の最適化において評価値の改善，すなわち非接触面積の減少が確認できた．また，配置が最適化された上位 10 個の結果すべてにおいて，5.1 節にて設定した配置関係の制約をすべて満たした配置を行っていることを確認した．しかし，従来手法で設定されていた全物体の重心については最適化後の評価値が減少した．図での配置結果を確認すると，従来手法での配置結果は 2 段目に積まれている物体（Baby Wipes）が下の物体からはみ出ているため接触していない部分があるのに対し，提案手法による最適化結果では 2 段目に配置されている物体は底面のすべての部分が下の物体に接していることがわかる．しかし，上部に配置された物体は下の物体のどこにも接していない状態で配置された．この配置は両手法の結果で発生した．

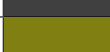
Table 6: Optimization result by previous method

	F_{overlap}	F_{cog}	F_{surface}	F_{type}
0	1	0.0086325105	0.0152472388	0.3333333333
1	1	0.0085828213	0.0109011238	0.3333333333
2	1	0.0088752748	0.0119985494	0.1428571429
3	1	0.0085486624	0.0103938056	0.1428571429
4	1	0.0085239053	0.0111312027	0.1666666667
5	1	0.0087415125	0.0132285777	0.1111111111
6	1	0.0092066716	0.014520145	0.25
7	1	0.0086111901	0.0139908218	0.0714285714
8	1	0.0088031352	0.0119016498	0.25
9	1	0.008671709	0.0128194754	0.0909090909
Avg.	1	0.0087197393	0.012613259	0.1892496392

Table 7: Optimization result by proposed method

	F_{overlap}	F_{cog}	F_{surface}	F_{type}
0	1	0.0080191061	0.0223230024	1
1	1	0.0074777215	0.0193685157	1
2	1	0.0077819554	0.018799816	1
3	1	0.0075582463	0.0173010379	1
4	1	0.0063066631	0.0185184107	1
5	1	0.0067495735	0.0179456595	1
6	1	0.0071867578	0.0167433808	1
7	1	0.0072313943	0.0167334917	1
8	1	0.0068500563	0.017985611	1
9	1	0.0066952111	0.0176951137	1
Avg.	1	0.0071856685	0.018341404	1

Table 8: Items color

Item name	Color
Canned food (Fig. 10a)	
Coffee bottle (Fig. 10b)	
Liquid detergent (Fig. 11a)	
Powder pouch (Fig. 11b)	
Snack A (Fig. 12a)	
Snack B (Fig. 12b)	
Baby Wipes (Fig. 13a)	
Expo Eraser (Fig. 13b)	
Ball Container (Fig. 13c)	
Speed Stick (Fig. 13d)	
Tissue Box (Fig. 13e)	

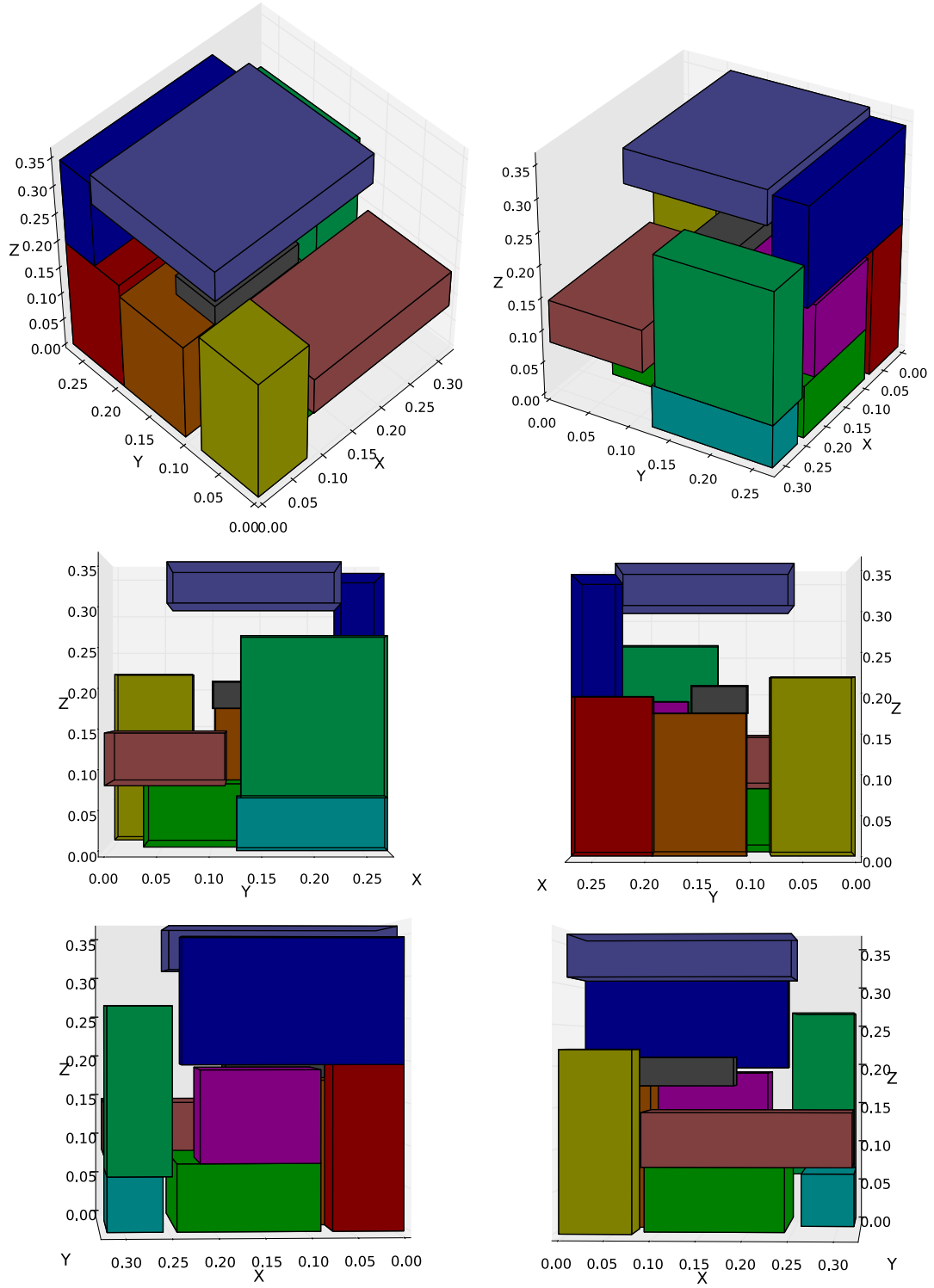


Fig. 18: Optimized placement result by previous method

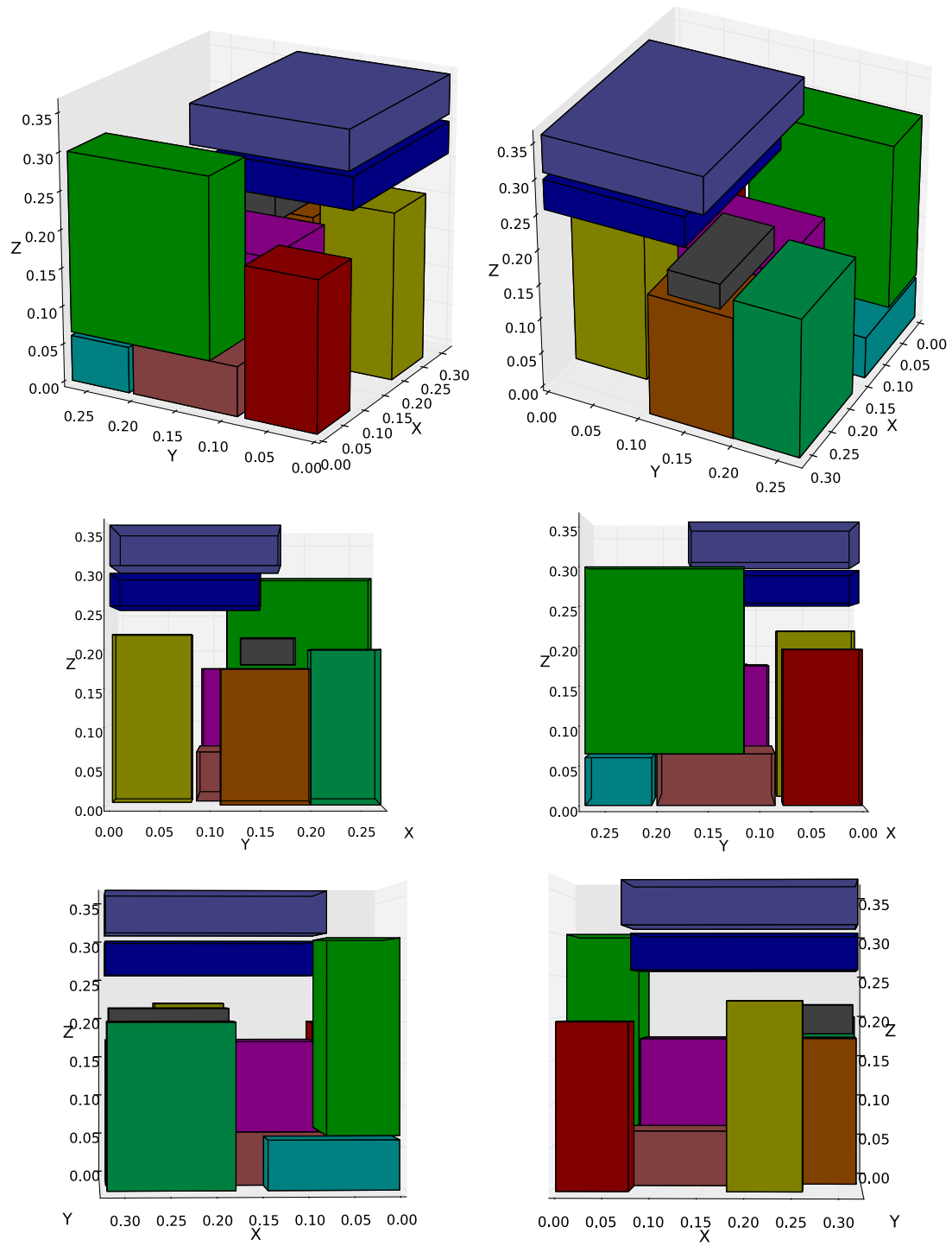


Fig. 19: Optimized placement result by proposed method

5.4 考察

本実験において、評価値の高い最適化結果が物体の種類と配置関係の制約を満たしていることから、本手法で設定した評価関数が有効であることが示唆された。また最適化結果において積載されている物体の宙に浮いた面積が減少していることから、本手法で設定した底面接触面積の評価関数が有効であることを示した。物体の配置方向の組み合わせによっては、物体をどの位置においても物体の種類と配置関係の制約を満たせない場合がある。今回設定した種類と配置関係の制約を評価関数に組み込むことで、最適化後の評価値が高い結果を抽出したとき、この制約を満たせる配置方向の組合せを見つけることができる。

全物体の重心の評価値の減少については、最適化を行う評価関数を増やしたことが原因だと考えられる。物体位置の最適化は各評価関数を合計した評価値が対象である。そのため、評価関数が増加すると最適化中に物体の位置が変更された時の評価値の変化が大きい物が優先される。本実験では評価値の重みは重心の評価値のほうが高く設定したが、重心の評価値よりも物体の種類と配置関係による評価値のほうが評価値の影響が大きかったためであると考える。最適化後の重心位置を比較すると、従来手法による最適結果の重心位置の平均は

$$d_{\text{cog}} = \frac{\epsilon_{\text{cog}}}{F_{\text{cog}}} - \epsilon_{\text{cog}} = \frac{0.001}{0.0087197393} - 0.001 \approx 0.1137[\text{m}] \quad (24)$$

また、提案手法による最適結果の重心位置の平均は

$$d_{\text{cog}} = \frac{\epsilon_{\text{cog}}}{F_{\text{cog}}} - \epsilon_{\text{cog}} = \frac{0.001}{0.0071856685} - 0.001 \approx 0.1382[\text{m}] \quad (25)$$

今回採用した容器の高さは0.360[m]であり底面から高さの半分未満に位置するため、提案手法での最適結果の重心でも配置上問題ないを考える。

最適化結果にて宙に浮いた物体が存在したことについて、物体をより下方に配置した場合一部でも他の物体へ重複するためであると考える。実際の配置のためには重心の最適化よりも物体同士の重複を避けたほうが望ましいため、この結果になったと考える。そして、宙に浮いた物体が他の物体と比べて軽いことも原因の一つと考えられる。全物体の重量の合計は3892[g]であり、平均は353[g]であり、宙に浮いた物体は130–180[g]であるため、重心の評価関数に対する影響が少

なかったため浮いた状態で最適化が完了したと判断したと考える。最適化結果にて物体が宙に浮いた状態を実際に配置した場合自由落下による位置や姿勢変更が発生する。しかし、物体配置は配置方向が固定された状態であり、回転を考慮していない。また、従来手法で設定された重心の最適化や本手法で設定した底面接触面積は重力による配置を考慮したものではないため、宙に浮いた状態で最適化されたと考える。

6. 物体積込実験

本手法で生成した積込動作を評価するため，ロボットアームによる積込実験を行った．積込動作の性能評価として，動作生成の有効性と動作の正確性の要素が考えられる．そこで，本実験では積込中の動作生成の成功率と積込位置の正確性を計測することで評価した．積込物体に対して，容器へ干渉せずに積み込む動作生成に失敗した回数を評価することで積込動作の有効性を検証した．さらに積込後の物体の位置と姿勢を計測することで積込動作の正確性を評価した．

6.1 実験環境

6.1.1 ハードウェア

本システムのハードウェア構成について述べる．実験で使用したシステムの概要図を Fig.20 に示す．ハードウェアシステムは作業台に設置されたロボットアームとその先端に取り付けたエンドエフェクタによって構成されている．作業台は床から 0.85[m] の高さであり，内部に外部電源，ロボットコントローラ，真空ポンプおよびエンドエフェクタ制御回路が搭載されている．また，作業台に置かれた積込物体を配送用容器に積み込むためにロボットは作業台の上面から 0.155[m] 下部に床に垂直な向きで設置した．ロボットアームと PC は Ethernet ケーブル，PC と RGB-D センサおよびエンドエフェクタ制御回路とは USB3.0 でそれぞれ接続されている．

6.1.2 ロボットアーム

ロボットアームには KUKA 社製のロボットアームである KUKA LBR iiwa14 を使用する．仕様を Table 9 に示す．人間の上肢と同等の自由度を持つため，人間の動作の模倣に適している．また 7 軸全てにトルクセンサを有しており，環境と接触した際はすぐに停止し再稼動が可能である．

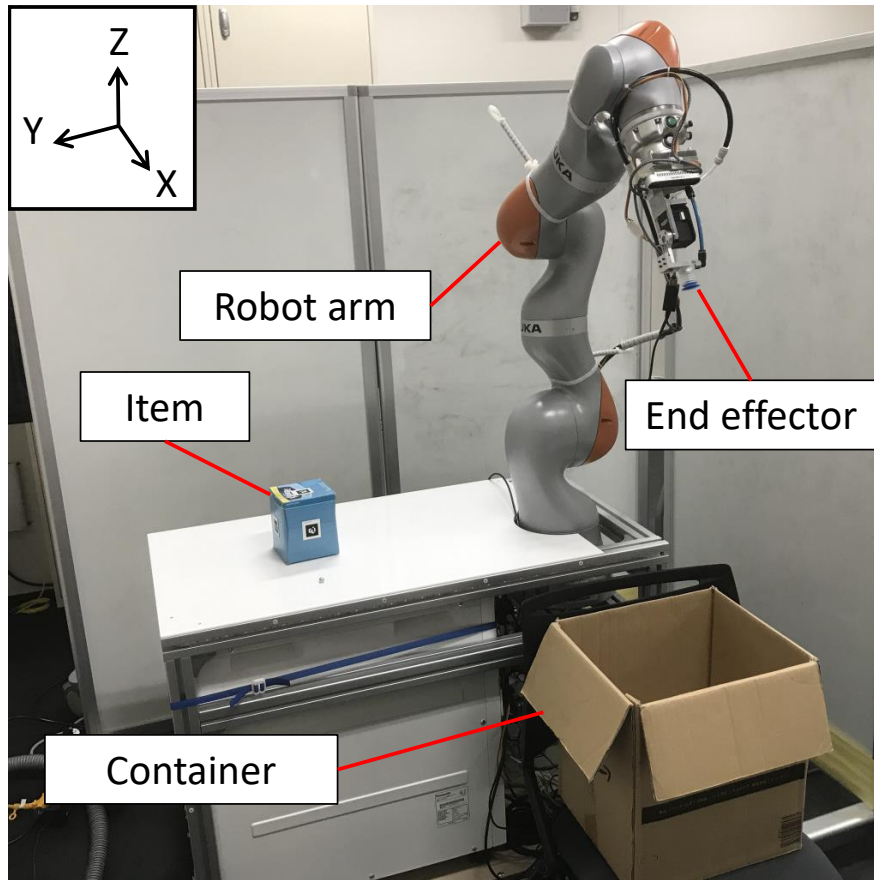


Fig. 20: Experimental setup

Table 9: Specification of the robot arm

Specification	Value / Type
Model	LBR iiwa 14 R820
Number of controlled axes	7
Volume of working envelope	1.8 [m ³]
Pose repeatability	± 0.15 [mm]
Weight	approx. 29.9 [kg]
Related payload	± 14 [kg]
Maximum reach	820 [mm]

6.1.3 エンドエフェクタ

ロボットアームに取り付けたエンドエフェクタを Fig.21 に示す。エンドエフェクタは AR マーカーを認識するための RGB-D カメラ，物体を把持するための吸引カップ，追加の関節用モータおよび直線受動関節によって構成されている。RGB-D センサとして Intel 社の RealSense D415 を使用した。RGB 画像の解像度は 1920×1080 [px] であり，深度情報の計測範囲は $0.3\text{--}10$ [m]（較正，背景，照度状況による）かつ解像度は 1280×720 [px] である。また，吸引カップには PISCO 社の包装袋用真空パッド ベローズ 5 段タイプ（VPE30PBS-5M5-FF）を使用した。吸引パッドの外形は 30 [mm] で吸込防止アダプタが付いている。吸引カップは，真空ポンプが稼働した状態で吸引カップを物体の表面に対して垂直に押し付けることで物体を把持することができる。使用した真空ポンプの仕様を Table 11 に示す。追加の関節モータは箱や棚など狭い空間での物体操作を実現するため，ロボットアームにさらに自由度を追加している。関節モータの仕様を Table 10 に示す。直線受動関節は物体を把持する際に物体に過度な力が加わらないようにする役割をもつ。エンドエフェクタの構成部品の接合部は，物体操作中に負荷が大きくかかる部品には金属加工によって，残りは ABS 樹脂を材料に 3D プリンタで製作した。このエンドエフェクタはパナソニック株式会社と共同で開発した。

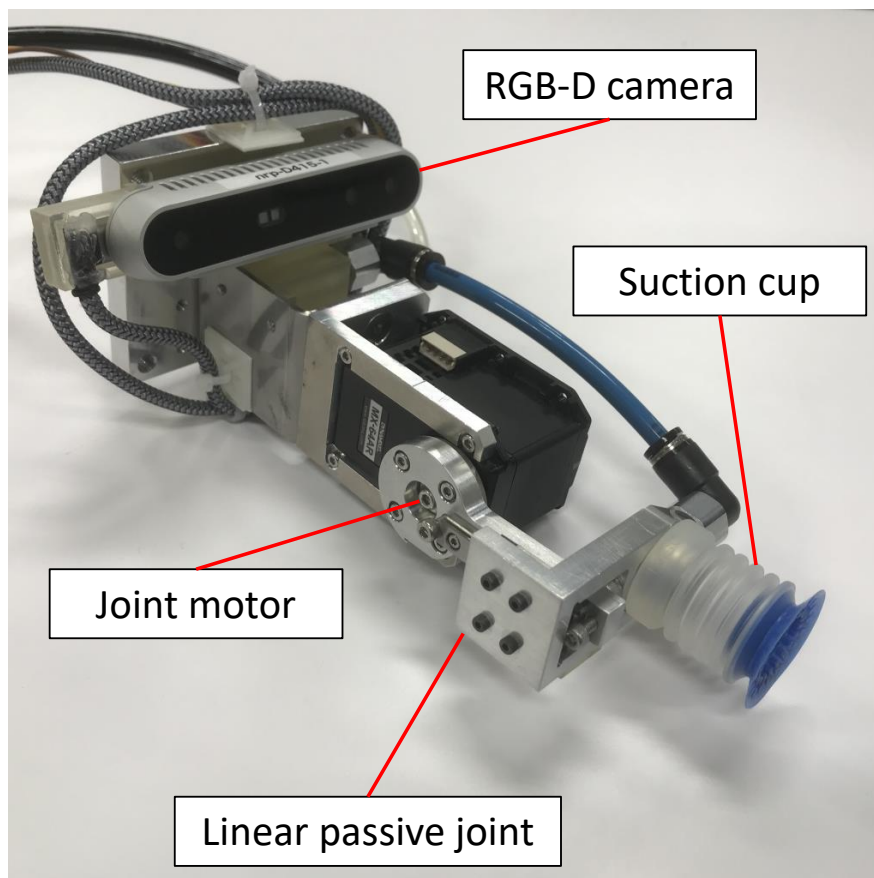


Fig. 21: End effector for item manipulation

Table 10: Specification of joint motor

Specification	Value / Type
Product Name	MX-64AR
Weight	135 [g]
Dimension	$40.2 \times 61.1 \times 41$ [mm]
Gear Ratio	200:1
Minimum Control Angle	0.088 [deg] $\times 4,096$
Operating Voltage	12 [V]
Stall Torque	6 [$N \cdot m$]
Stall Current	4.1 [A]
Baud Rate	8000 [bps] - 4.5 [Mbps]

Table 11: Specification of vacuum pump

Specification	Value / Type
Product Name	DA-120S
Pumping speed	120 (50[Hz]) or 144 (60[Hz])
Ultimate pressue	13.3×10^3 [Pa]
Port diameter	O.D. $\phi 14 \times$ I.D. $\phi 9$ [mm]

6.1.4 ソフトウェア構成

物体積込動作のソフトウェア構成を Fig.22 に示す．本研究に使用するシステムのソフトウェア構成は全て Robot Operation System (ROS) によって構成されている [15]．ROS はロボットソフトウェアプラットフォームの一種であり，異なるハードウェアでも共通して使用できるロボットオペレーションシステムである．ROS はノードと呼ばれるプロセス間で，トピックを介して各情報を交換する．使用する PC の OS は Ubuntu16.04 であり，ROS のバージョンは ROS Kinetic を選択した．RealSense からの RGB 画像および深度情報は ROS パッケージの librealsense を通して ROS のトピックを受信する．また，マーカーの認識には ROS パッケージの ar_track_alvar を通して ROS のトピックを受信する．ロボットアームの逆運動学と動作計画にはモーションプランニングのフレームワークである MoveIt![16] を利用する．ロボット動作の軌道生成には RRT-Connect[17] を採用した．また，ロボットアームである KUKA LBR iiwa14 への軌道入力には ROS パッケージの iiwa_stack を使用してコントローラに軌道を入力する．

軌道生成は以下の流れで実行される．RGB-D センサから取得したマーカーの位置情報を PC が取得し，物体の 3 次元姿勢を算出する．算出結果を元に把持動作，積込動作の軌道をモーションプランナーである MoveIt! が計算して関節角度を送信する．ロボットアームのコントローラとエンドエフェクタ制御回路に指令値を入力することで動作を実行する．

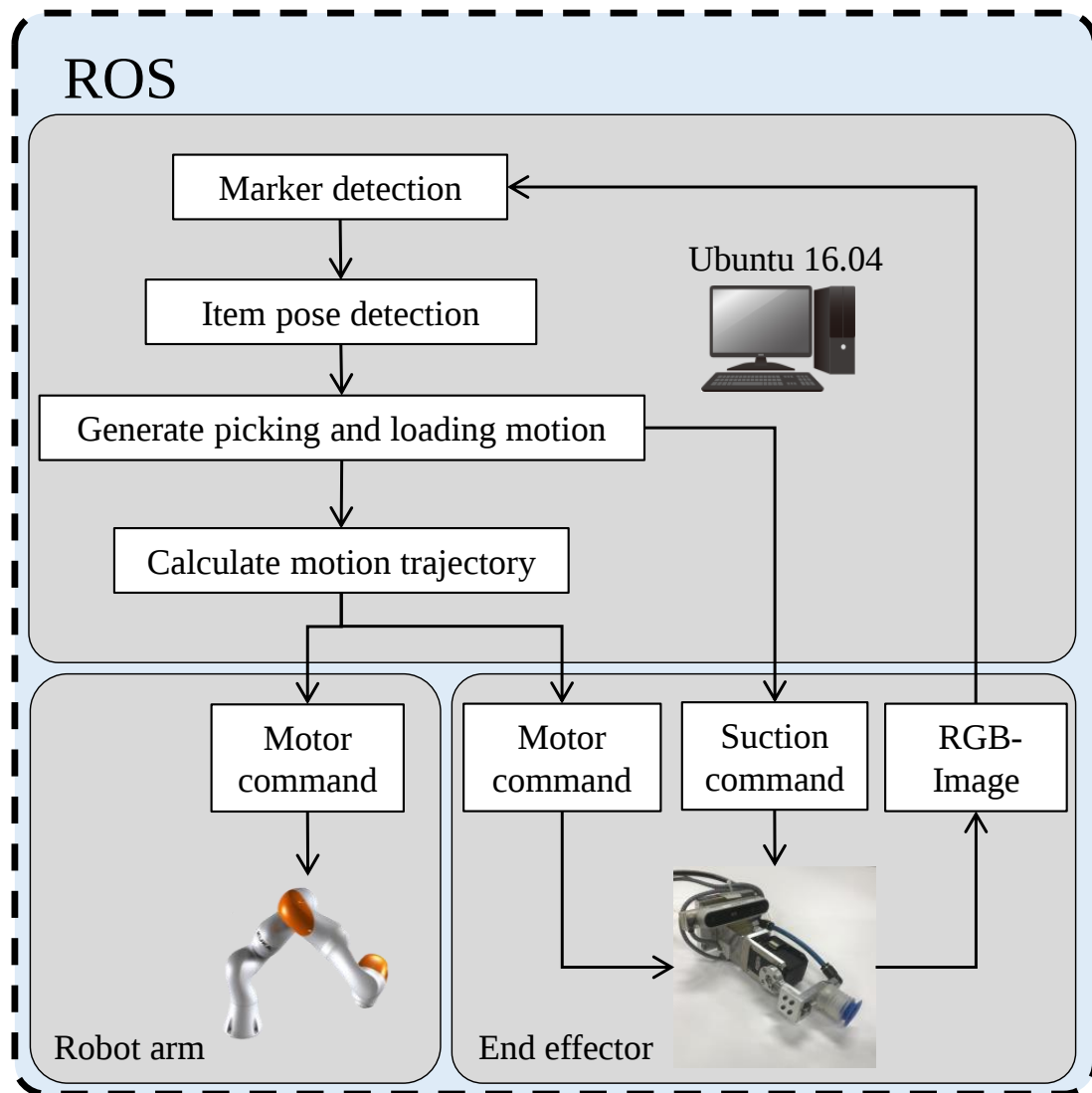


Fig. 22: Software configuration

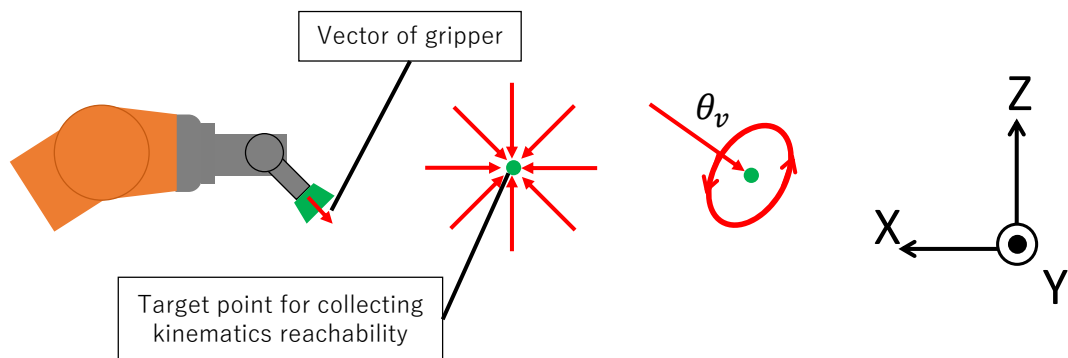


Fig. 23: Collecting kinematics reachability map

6.1.5 運動学的到達性

ロボットアームの動作生成失敗の原因には、周囲の環境への衝突が避けられないために失敗する場合と、ロボットの運動学的に到達が不可能であることの2つが主に挙げられる．前者については、作業中に代わりの目標座標を探索する必要がある．作業のための動作生成を確実に成功させるためには、後者による失敗を未然に防ぐことが望ましい．そこで、事前にロボットアームに対する運動学的到達性マップ[18]を作成する．概要図を Fig. 23 に示す．ロボットの作業空間内に一定距離ごとに運動学的到達性を確認する目標点を設ける．目標点に対してエンドエフェクタが垂直な方向を囲うように設定し、さらに方向ごとにベクトル方向の回転 θ_v を考慮したロボットの目標姿勢を設定する．以上の目標姿勢に対してロボットの逆運動学を解き、どれだけの割合で解くことができたかで到達性を確認することで運動学的到達性の分布を作成した．今回は 5[cm] ごとに目標点を設け、収集点に対して 26 方向から運動学的到達性を確認し、90[%] 以上の方向から到達できた領域内に容器を設置した．

6.2 実験内容

ロボットアームを用いて容器への物体積込の実験を行った。第5章にて、提案手法にて最適化され最も評価値の高かった積込計画の結果に従って積込作業を行った。実験では操作者によって配置した全 11 個の積込対象物体に対して、吸引グリッパを持ったロボットにより同じ容器に積み込むタスクを行った。

実験手順を示す。

1. 積込容器を設置する。
2. 積込対象の物体を作業台に配置する。
3. 物体に対して積込動作の生成を実行する。
4. 物体の積込に失敗し他の物体の配置が不可能になった場合、操作者によって物体を回収し再度配置を行う。それ以外は放置する。
5. 2 から 4 を繰り返す。
6. すべての物体を積み込んだ時点で動作を終了する。

本実験では容器に積み込む物体の順番は操作者によって決定した。また、本研究では物体の姿勢変更についてのマニピュレーションは取り扱っていないため、作業台にのせる物体の姿勢は積込時の姿勢に対して、Z 軸方向にのみ任意に回転した状態で配置した。

6.3 実験結果

積込した物体の順番と積みこみ動作の生成結果および配置後の物体位置を示す。Table 12 に物体の積み込んだ順番と積込動作生成の成否の結果と実際に積み込んだ際に発生した事象を示す。Table 13 に積込計画による物体の積込位置とロボットによる配置後の位置およびその誤差を示す。

積込を行った全 11 個の物体について、積みこみ動作の生成に成功した。ロボットでの実機での積込結果として、Canned food や Coffee bottle では積込直後に横方向に移動し、Tissue Box ではすでに積み込まれた物体（Coffee bottle）と衝突が発生した。Snack A と Snack B は積込後に下の物体と接触するまで落下した。

積込後の物体の位置について、積込時に衝突や落下が起きていない物体で、Ball Container や Liquid detergent で 20 [mm] 以上の誤差が見られた。

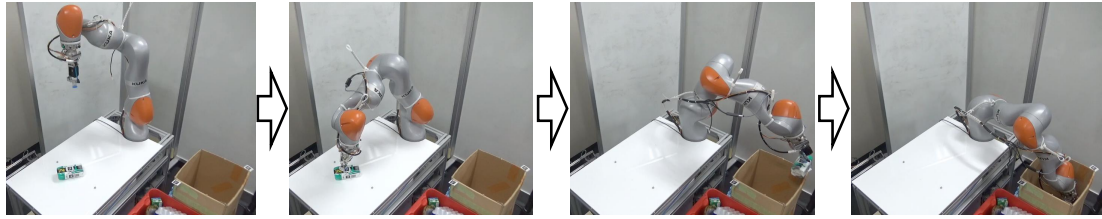
各物体の積込中の様子とモーションプランナーの動作結果を図に示す。積込動作の成功例として Speed Stick の積込時の動作結果を Fig. 24 に、他の物体と衝突が発生した Tissue Box の積込時の動作結果を Fig. 25 に、それ以外の物体の積込の様子を付録 A の Fig. 26 から Fig. 34 に示す。各 Fig. の (a) は全体の様子を (b) は容器上部からの視点での様子を、(c) は実験時に生成された積込動作のシミュレーション結果である。

Table 12: Result of experiment loading motion generation

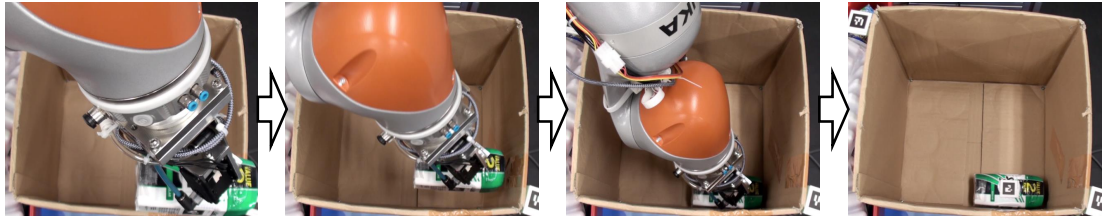
Loading order	Item name	Planning result	Remarks
1	Speed Stick (Fig. 13d)	Success	
2	Baby Wipes (Fig. 13a)	Success	
3	Powder pouch (Fig. 11b)	Success	
4	Canned food (Fig. 10a)	Success	Change item position after placement.
5	Ball Container (Fig. 13c)	Success	
6	Liquid detergent (Fig. 11a)	Success	Soft collision between robot and container.
7	Coffee bottle (Fig. 10b)	Success	
8	Tissue Box (Fig. 13e)	Success	Hard collision between loading item and loaded item.
9	Expo Eraser (Fig. 13b)	Success	
10	Snack A (Fig. 12a)	Success	Item dropped softly.
11	Snack B (Fig. 12b)	Success	Item dropped softly.

Table 13: Result of experiment loading motion generation

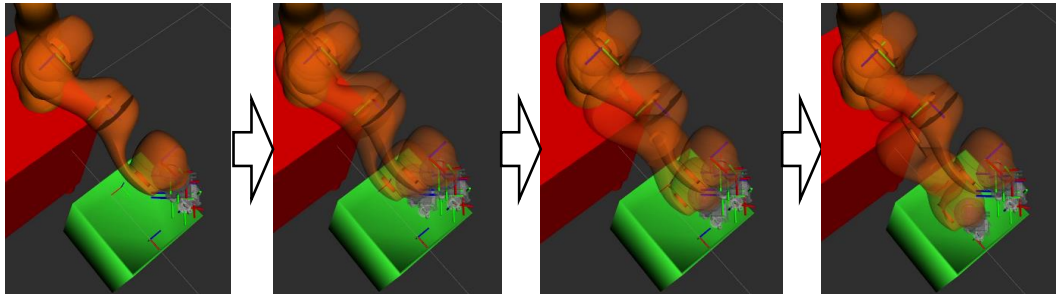
Loading order	Item name	Planned position x, y, z [mm]	Placed position x, y, z [mm]	Error x, y, z [mm]
1	Speed Stick (Fig. 13d)	72, 237, 30	72, 234, 33	0, 3, 3
2	Baby Wipes (Fig. 13a)	115, 142, 32	111, 140, 29	4, 2, 3
3	Powder pouch (Fig. 11b)	249, 235, 100	237, 244, 103	12, 9, 3
4	Canned food (Fig. 10a)	38, 38, 98	78, 17, 102	40, 21, 4
5	Ball Container (Fig. 13c)	221, 40, 110	230, 12, 100	9, 28, 10
6	Liquid detergent (Fig. 11a)	40, 192, 182	13, 167, 159	27, 25, 23
7	Coffee bottle (Fig. 10b)	275, 155, 87	264, 123, 79	11, 32, 8
8	Tissue Box (Fig. 13e)	148, 145, 120	164, 138, 111	16, 7, 9
9	Expo Eraser (Fig. 13b)	249, 157, 197	247, 157, 186	2, 0, 11
10	Snack A (Fig. 12a)	200, 75, 277	180, 82, 229	20, 7, 48
11	Snack B (Fig. 12b)	195, 85, 334	178, 111, 283	17, 26, 51



(a) Over view

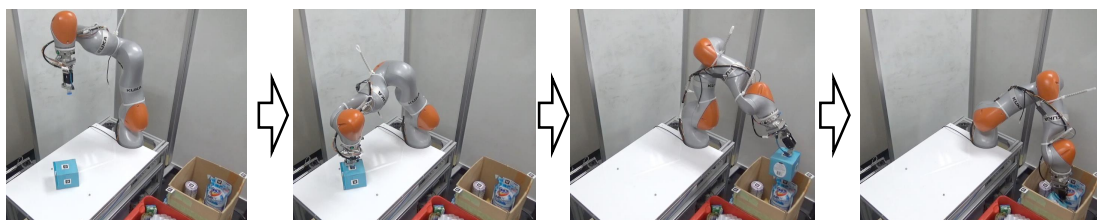


(b) View from above container



(c) Motion planning result of loading

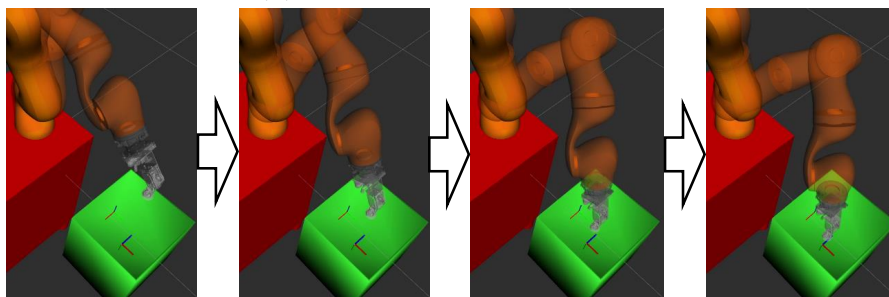
Fig. 24: Result of loading Speed Stick



(a) Over view



(b) View from above container



(c) Motion planning result of loading

Fig. 25: Result of loading Tissue Box

6.4 考察

ロボットアームによる積込実験において、本論文で使用したすべての物体について積込動作の生成に成功していること、また Tissue Box 以外の物体を大きな衝突なく積み込めたことから、本手法で生成した積込動作が有効であることが示唆された。

実機での積込時において、積込直後に Canned food や Coffee bottle の移動が発生した原因について、把持した物体が傾くことで配置時の底面の接触部分が計画時と異なるため、グリッパが物体の把持状態を解除した際に物体の移動が起きたと考える。本手法では把持中の物体の姿勢を考慮していないため、このような事象が発生したと考える。すでに積み込まれた物体との衝突が起きた Tissue Box について、実機のロボット動作が生成された動作と異なることが原因と考える。Fig.25 より、生成された動作は直線軌道による積込動作を生成しているのに対し、実機での動作軌道は湾曲を描いている。本実験で使用したモーションプランナーである MoveIt! は軌道生成の結果として動作中の各中継位置でのロボットの各関節角度をロボットのコントローラに出力する。この時に、動作軌道での中継位置の解像度が低いと、実機の動作が想定された軌道に追従できないため、動作計画と異なる軌道で動作したと考える。積込後に落下が起きた Snack A と Snack B について、物体積込計画時にはこの2つの物体が宙に浮いた状態で配置されていたため実際積み込み時に落下が発生したと考える。

7. おわりに

7.1 まとめ

本研究ではロボットによって、多種類物体の種類と配置関係における制約を満たした箱への積込を行うために、物体積込計画とロボットによる容器への積込動作生成を行った。物体積込計画では物体位置を最適化問題として扱う手法を用いて、物体の積込の安定化を図るため物体を積載した時に生ずる宙に浮いて面積を最小化する評価を追加した。さらに物体の種類とは位置関係による制約を評価値に組み込むことで、制約を満たした物体の積込配置を決定するための手法を提案した。積込動作生成では、吸引グリッパを持つロボットを対象に逆運動学とモーションプランナーを用いて把持角度を探索することで、容器と衝突を考慮した積込動作を生成した。

物体積込計画実験では使用した物体に3つの種類を設定し、それぞれに制約を与え設定した最適化関数を用いて、最適化後に評価値の最も高い結果が物体の種類と配置関係の制約を満たしていることを確認した。最後にロボットアームによる物体積込実験を行った。物体の姿勢認識はARマーカーを用いてロボットに搭載したRGB-Dセンサから取得した。積込計画実験で最適化された積込計画の結果に従って容器への物体積込を行った。本研究で使用した11種類の物体に対し積みこみ動作の生成に成功し、10個の物体に対し積みこみ動作を完了させ本手法で生成した動作の有効性を示した。

7.2 今後の課題と展望

物体積込計画について、本研究では物体の種類と制約は3つずつ設定したが実際の物体の種類はそれより多く存在し、対象が多くなるほど発生する制約も多くなる。本研究の手法は、物体の配置関係における制約を最適化関数として設定するため、物体の種類のコマボンドによる制約が増えても最適化関数を拡張することで複数の制約を同時に考慮して物体を配置することができる。積込計画には事前に物体のパラメータとして各物体のサイズや重量をあらかじめ入力する必要がある。

これらの問題には、積込計画を行う前に各物体を RGB-D センサと重量センサで計測することで対処することが可能であると考えられる。物体の種類の取得については、通販サイトなどには商品にカテゴリーが割り振られているため、それをデータベース化し参照することで対処することが可能であるとする。物体を容器に積み込んだ後は、余剰空間に緩衝材を積み込まなければならない。この問題には、物体を配置した後に緩衝材を追加で配置させることで対処することが可能である。

積込動作の生成について、動作計画と実際の動作が異なることによる物体の衝突について、生成する動作の軌道の解像度を上げ、より精密な動作を実現することで対処できると考える。また、物体を把持した際の姿勢が変わることによる積みこみ位置の誤差の発生については、物体を把持し持ち上げた後にグリッパに対する物体の姿勢を再度計測することで積みこみ動作の微調整が可能であるとする。また、より多くの物体を取り扱った際、ロボット本体およびエンドエフェクタに対して非常に小さい物体を容器の隅に積み込む場合、たとえ十分な把持が可能であっても容器の衝突が避けられない場合がある。この問題には、隅から少し離れた位置に一度配置し、容器の隅へ押す動作を生成することで対処できると考える。また、今回生成した動作は吸引グリッパを対象に取り組んだが、多くの物体の安定して積込するためには並行グリッパなど多指グリッパによる積込にも対応する必要がある。この問題には、多指グリッパによる把持点を検出した際に、3次元角度のうち比較的自由に設定できる軸を検証し、その軸に対して積込可能な角度を探索することで対処可能であるとする。最後に、今回は積込む順番は操作者によって決定したが、これらは自動で決定されることが望ましい。この問題は、積み込むロボットが事前に決まっていれば、配置後の物体のモデルをモーションプランナーに読み込ませることと、積込計画の段階で物体の配置関係が既知であることから、容器の隅に配置される物体を始めとした破綻のない積込順の自動生成が可能であるとする。

謝辞

本研究の遂行にあたり，様々なご指導とご助言を賜りました小笠原 司 教授に深く感謝致します。

論文執筆にあたり，丁寧なご検討，ご教示を頂きました副指導教員である本学先端科学技術研究科情報科学領域の杉本 謙二教授に御礼申し上げます。

本研究の遂行，論文執筆や，グループミーティング，研究会での多くのご助言，ご指導を頂きました本研究科の高松 淳 准教授に深く感謝致します。

グループミーティング，研究会での多くのご指導，ご助言を頂きました本研究科の丁 明 助教に深く感謝致します。

本研究についての様々な議論や論文執筆，英文添削など多くの点においてご指導，ご助言を頂きました本研究室博士研究員の Gustavo Garcia 助教に深く感謝致します。

事務処理の面で様々なサポートをして頂きました本研究室秘書の大脇 美千代 氏に深く感謝致します。

論文執筆の指導，研究を進める上での多大なご助力ならびに研究会やミーティング等で多くのご助言を頂きました本研究科ロボティクス研究室博士後期課程学生の山崎 亘 氏，袴田 有哉 氏，趙 崇貴 氏，白倉 尚貴 氏，Pedro Uriguen 氏に深く感謝致します。

本研究室の先輩および同輩の皆様，後輩達には研究を進めるうえで多くのご協力を頂きました。ここに深く御礼申し上げます。

最後に，学生生活を支えていただいた両親，家族，および友人達に心から感謝致します。

参考文献

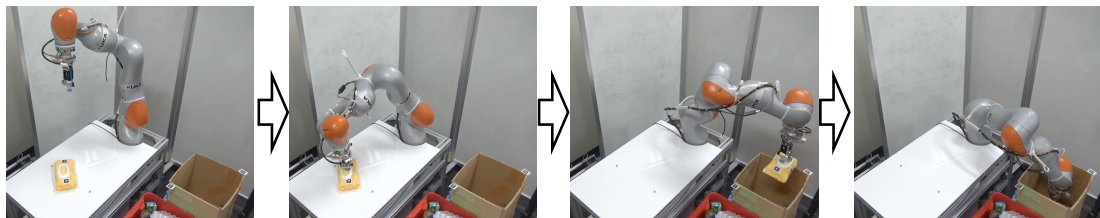
- [1] “下請等中小企業における取引条件の改善状況調査結果”, <http://www.chusho.meti.go.jp/keiei/torihiki/2018/180605Shitaukechousa.htm>, Accessed 7 Jan. 2018.
- [2] “Amazon Robotics,” <https://www.amazonrobotics.com>, Accessed 7 Jan. 2018.
- [3] “MUJIN 様×アスクル「ピッキングロボット」開発プロジェクト”, <https://www.askul.co.jp/kaisya/saiyo/new/projectstory/robot.html>, Accessed 7 Jan. 2018.
- [4] 藤吉弘亘, 岡田慧, 江原浩二, G. Garcia: “[特別講演] Amazon Robotics Challenge 2017 の参加レポート”, 信学技報, vol. 117, no. 442, PRMU2017-170, pp. 145–154, 2018.
- [5] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio 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.
- [6] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy and K. Goldberg: “Dex-Net 3.0: Computing Robust Robot Suction Grasp Targets using a New Analytic Model and Deep Learning,” IEEE International Conference on Robotics and Automation (ICRA), pp. 5620–5627, 2018.
- [7] C. Guan, W. Vega-Brown and N. Roy: “Efficient planning for near-optimal compliant manipulation leveraging environmental contact,” IEEE International Conference on Robotics and Automation (ICRA), pp. 215–222, 2018.
- [8] F. Furrer, M. Wermelinger, H. Yoshida, F. Gramazio, M. Kohler, R. Siegwart and M. Hutter: “Autonomous Robotic Stone Stacking with Online next Best

- Object Target Pose Planning,” IEEE International Conference on Robotics and Automation (ICRA), pp. 2350–2356, 2017.
- [9] X. Zhao, J. A. Bennell, T. Bektaş, and K. Dowsland: “A comparative review of 3D container loading algorithms,” *International Transactions in Operational Research*, vol. 23, no. 1–2, pp. 287–320, 2016.
- [10] 岩澤宏紀, 今堀慎治: “Sequence-Triple を用いた 3 次元配置問題に対する局所探索法 (最適化アルゴリズムの進展 : 理論・応用・実装)”, 数理解析研究所講究録, vol. 1931, pp. 129–147, 2015.
- [11] T. S. Li, C. Liu, P. Kuo, N. Fang, C. Li, C. Cheng, C. Hsieh, L. Wu, J. Liang, C. Chen: “A three-dimensional adaptive PSO-based packing algorithm for an IoT-based automated e-fulfillment packaging system,” *IEEE Access*, vol. 5, pp. 9188–9205, 2017.
- [12] J. Kennedy and R. Eberhart: “Particle swarm optimization,” *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.
- [13] 井出東, 安田恵一郎: “適応型 Particle Swarm Optimization に関する基礎的検討”, 電気学会論文誌 C (電子・情報・システム部門誌), vol. 124, no. 2, pp. 550–557, 2004.
- [14] “Amazon Japan Home Page,” <https://www.amazon.co.jp/>, Accessed 14 Jan. 2019.
- [15] 岡田慧: “ROS(ロボット・オペレーティング・システム)”, 日本ロボット学会誌, vol. 30, no. 9, pp. 830–835, 2012.
- [16] “MoveIt!,” <https://moveit.ros.org/>, Accessed 11 Jan. 2019.
- [17] J. Kuffner and S.M. LaValle: “RRT-connect: An efficient approach to single-query path planning,” in *Proc. 2000 IEEE Intl. Conf. on Robotics and Automation*, pp. 995–1001, Apr. 2000.

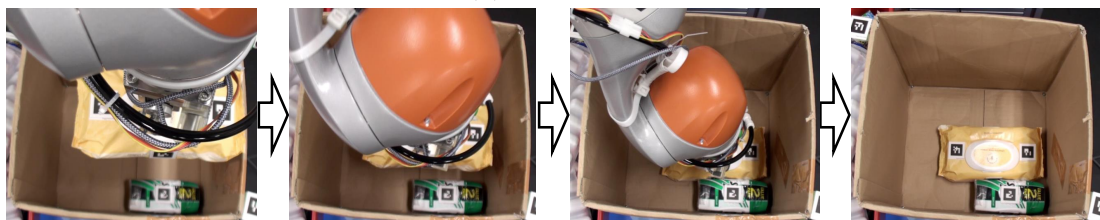
- [18] A. Makhal, A. K. Goins: “Reuleaux: Robot Base Placement by Reachability Analysis,” IEEE International Conference on Robotic Computing (IRC), pp. 137–142, 2018.

付録

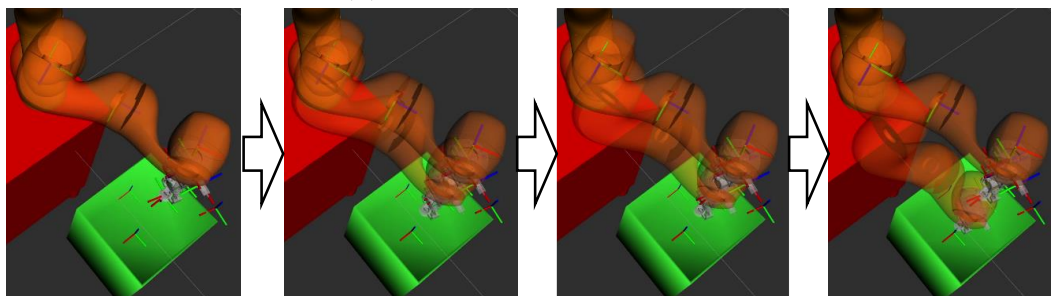
A. 積込動作結果



(a) Over view

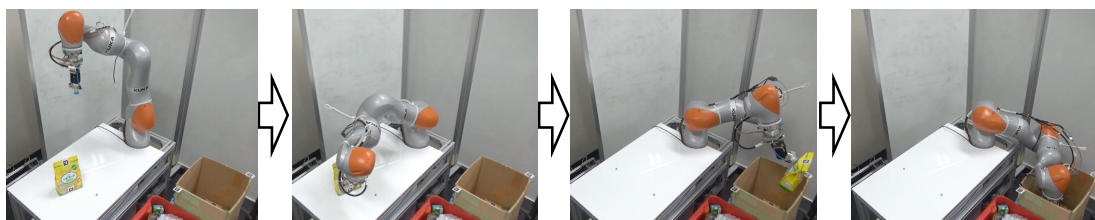


(b) View from above container

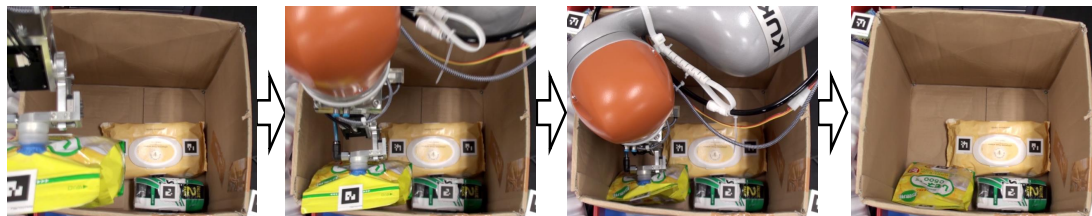


(c) Motion planning result of loading

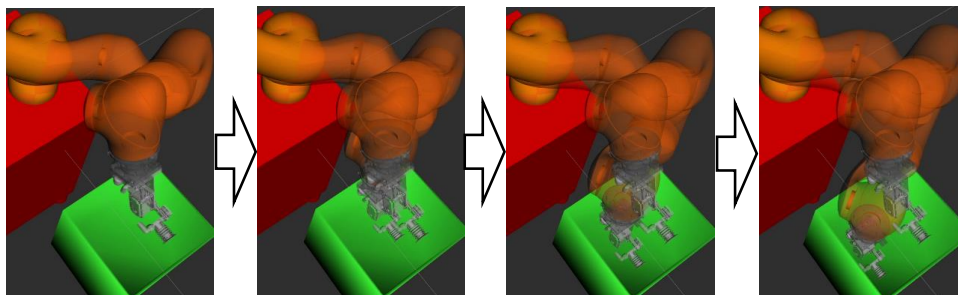
Fig. 26: Result of loading Baby Wipes



(a) Over view

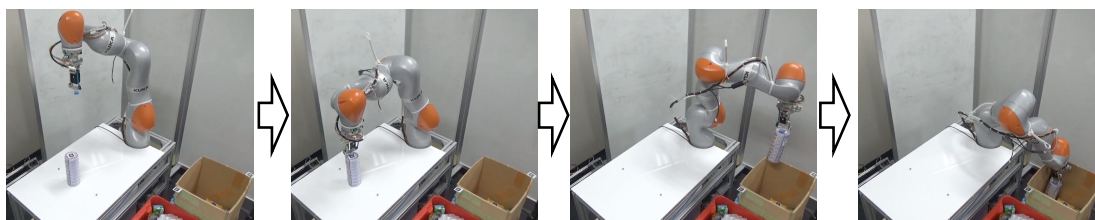


(b) View from above container



(c) Motion planning result of loading

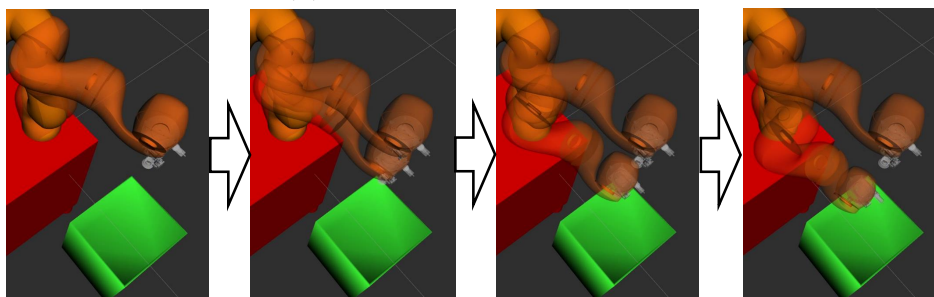
Fig. 27: Result of loading Powder pouch



(a) Over view

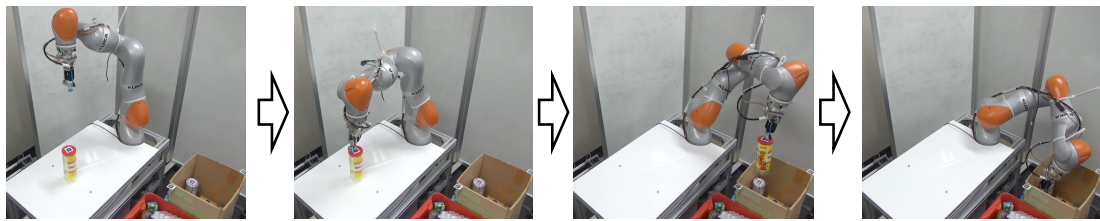


(b) View from above container



(c) Motion planning result of loading

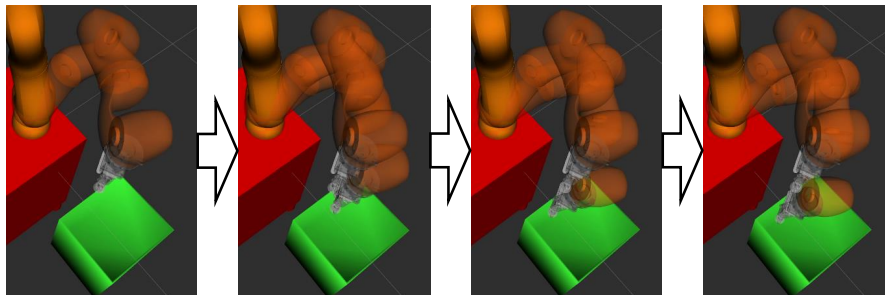
Fig. 28: Result of loading Canned food



(a) Over view

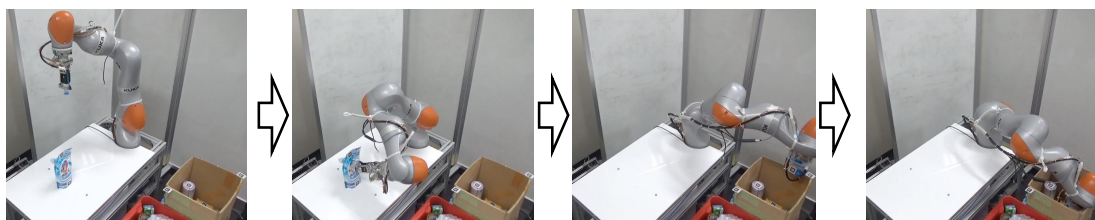


(b) View from above container

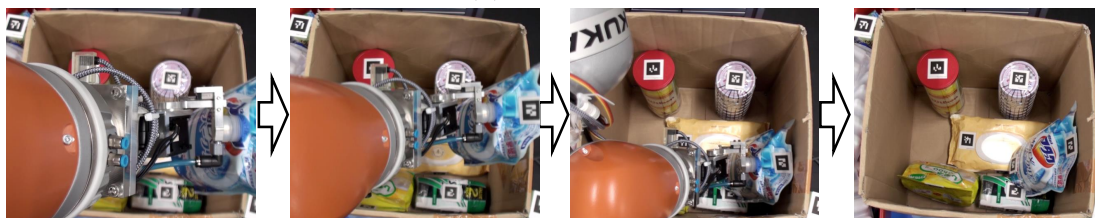


(c) Motion planning result of loading

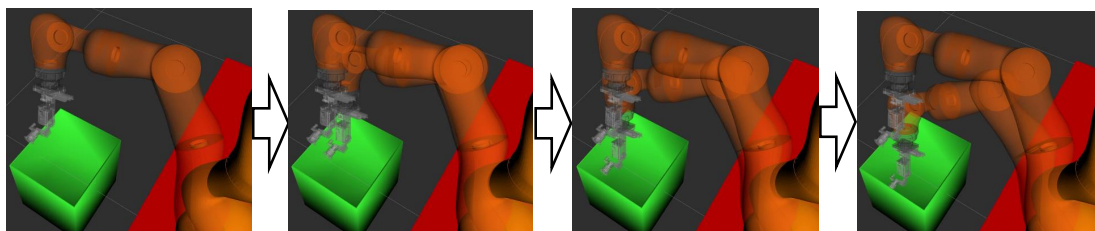
Fig. 29: Result of loading Ball Container



(a) Over view



(b) View from above container



(c) Motion planning result of loading

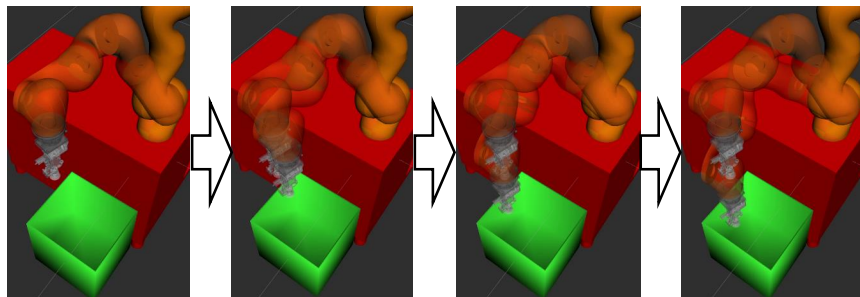
Fig. 30: Result of loading Liquid detergent



(a) Over view



(b) View from above container



(c) Motion planning result of loading

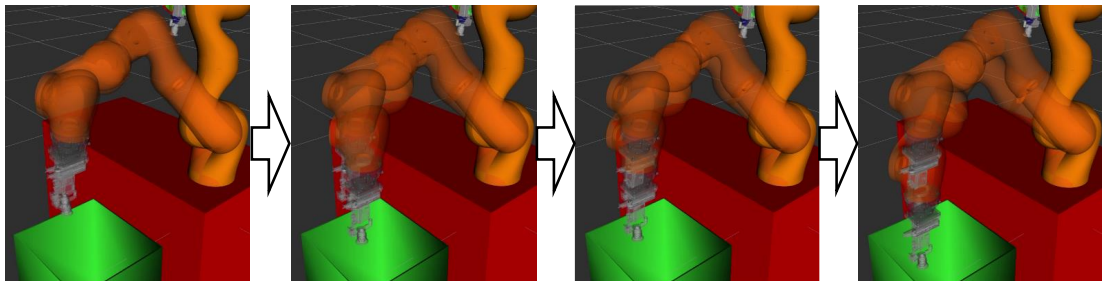
Fig. 31: Result of loading Coffee bottlet



(a) Over view



(b) View from above container



(c) Motion planning result of loading

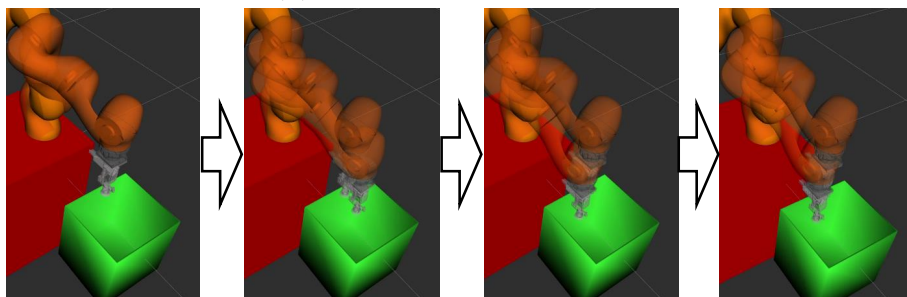
Fig. 32: Result of loading Expo Eraser



(a) Over view

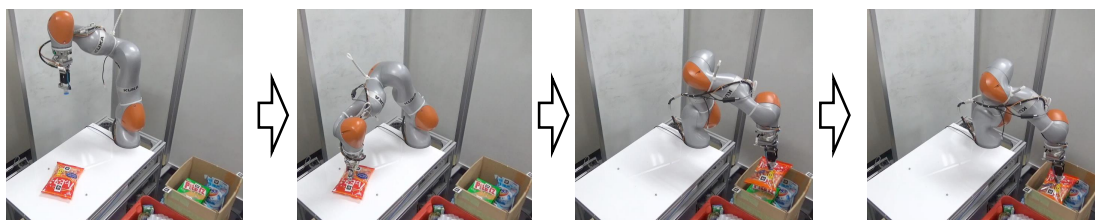


(b) View from above container



(c) Motion planning result of loading

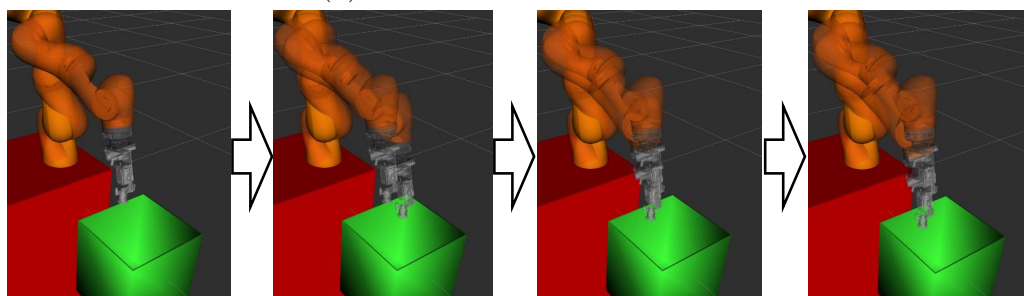
Fig. 33: Result of loading Snack A



(a) Over view



(b) View from above container



(c) Motion planning result of loading

Fig. 34: Result of loading Snack B