

# 3次元ラドン変換を用いた効率的なアセンブリモデルの部分検索手法

平島 栄志<sup>†</sup> 豊田 啓介<sup>†</sup> 片山 薫<sup>††</sup>

<sup>†</sup>首都大学東京 システムデザイン学部 〒191-0065 東京都日野市旭が丘6-6

<sup>††</sup>首都大学東京大学院 システムデザイン研究科 〒191-0065 東京都日野市旭が丘6-6

E-mail: <sup>†</sup>{hirashima-takashi1,toyoda-keisuke}@ed.tmu.ac.jp, <sup>††</sup>kaoru@tmu.ac.jp

あらまし 近年、複数の CAD モデルを組み合わせて構成されるアセンブリモデルの活用は大きな関心を集めている。しかし既存の手法では、アセンブリモデル内の配置構造を考慮するものは少ない。また、あるアセンブリモデルを一部に含むアセンブリモデルを検索する部分検索については処理に時間がかかるという問題がある。そこで本研究では、アセンブリモデルの配置構造を考慮した部分検索の高速化を目的とし、新たな手法を提案する。提案手法では、アセンブリモデルに3次元ラドン変換を行い、配置関係を考慮しながらサブアセンブリ同士の最適な組み合わせを決定し、アセンブリモデル間の部分相違度の最も低いものを検索結果とする。形状や配置の異なる75のアセンブリモデルを利用し、提案手法を実験的に評価した。その結果、提案手法が高速化・省メモリ化・高精度化を実現していると示した。この成果は、既存手法では扱えなかった大規模なアセンブリモデルデータベースにおいても部分検索が可能になるという点で重要である。

キーワード 3次元 CAD アセンブリモデル、部分検索、部品配置、3次元ラドン変換

## 1 はじめに

CAD を用いたモデル設計は製造や建築、コンテンツ産業などのさまざまな分野で普及しており、膨大な量の3次元モデルデータが多く企業に蓄積している。また近年では、Industry4.0 や Society5.0 といったコンセプトを多くの国家や企業が掲げており、その実現には蓄積された3次元モデルデータの活用が必要不可欠である。そのため、3次元モデルの検索・クラスタリング・分類・検出などさまざまな技術の研究開発が広く行われている。これら3次元モデルデータ活用の技術について共通して必要となるものの一つに、モデル間の類似度計算がある。このモデル間の類似度計算の研究として、モデル同士の検索技術の研究が多く行われている。また、あるモデルに類似した一部分を持つような大きなモデルを検索するというモデルデータの部分検索技術は、より柔軟にモデル間類似度を計算することができる。こうした需要に答えるため、3次元モデルの検索技術および部分検索技術に関する研究は盛んに行われているものの、そのほとんどはモデルの形状、つまり外観の情報のみを考慮した検索技術となっている。

しかし、機械設計などで作られる3次元モデルは、複数の部品を組み合わせた3次元モデルであることが多い。特に自動車などの複雑で大型な機械の設計では、複数の階層に渡って部品が組み合わさって構成されており、膨大な数の部品を含んだ巨大なモデルが作られている。このような、複数の部品を組み合わせたモデルを“アセンブリモデル”と呼び、各階層でのまとまりを“サブアセンブリ”という。3次元 CAD モデルの最小単位を部品として、サブアセンブリは部品の集合であり、アセンブリモデルはサブアセンブリの集合である。アセンブリモデルに対しては、その複雑さのために部分検索技術の活用が多く期待され

ているものの、外観による検索技術はアセンブリモデルの内部構造を考慮することができない。現在、モデルの外観以外の情報を用いたアセンブリモデルの検索技術は、構成部品の配置の情報や、力学的な結合構造に着目した検索技術などが提案されているが、検索にはその目的によってさまざまな条件が存在し、また処理速度や精度の問題も多く残っているため、アセンブリモデルの検索技術も未だ十分とはいえない。

本研究では、形状と部品の配置構造のみを利用して、新しいアセンブリモデルの部分検索手法を提案する。同じく形状と部品の配置構造のみを利用して部分検索手法は佐藤ら[1]によって既に提案されているが、検索に4次元特徴量を使っており、処理に時間がかかる。提案手法では新たに3次元ラドン変換を用いることで特徴量を3次元へ削減し、処理の高速化・使用メモリの削減・検索精度の向上を実現した。

本稿の構成を述べる。2章では、本研究および関連研究の対象とするアセンブリモデルの部分検索問題について説明を行う。3章では、関連研究と本研究の位置づけを述べる。4章では提案手法を述べる。5章では提案手法を実験的に評価する。最後に6章で本研究の結論を述べる。

## 2 準 備

本章では、提案手法の対象とするアセンブリモデルの部分検索について説明する。

アセンブリモデルとは、複数の3次元モデルを組み合わせて構成される3次元モデルである。一般にアセンブリ構造はグラフ理論における木構造を成している。木構造の葉にあたる構成の最小単位となるモデルを“部品”と呼び、木構造の節にあたる部品の集合を“サブアセンブリ”と呼ぶ。サブアセンブリを含むサブアセンブリのような、深い構造を持ったアセンブリモデル

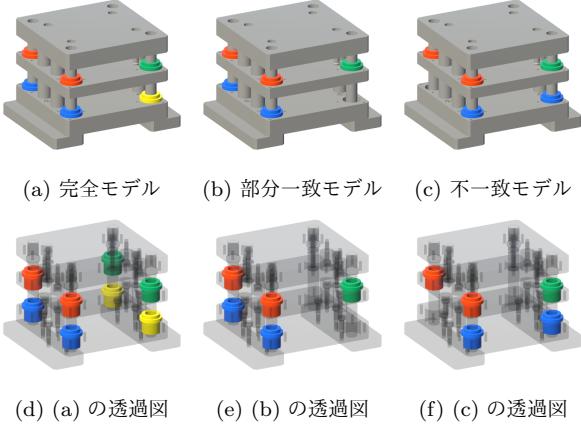


図 1: 部分一致するアセンブリモデルの例. それぞれ色分けされた複数のサブアセンブリから構成されている.

も考えられるが, 本稿ではサブアセンブリが部品のみを含む場合を考える. 以降では, 木構造の根にあたる, すべての部品から構成されるモデルをアセンブリモデルと呼ぶ.

図 1(a)(b)(c) は, 5 章で実験に用いるアセンブリモデルの例であり, それぞれの透過図を (d)(e)(f) に示した. 以降では, あるアセンブリモデルとそれを一部に含むようなアセンブリモデルがあるとき, それらを部分モデルと完全モデルと呼ぶ. (a) に部分一致するようなアセンブリモデルを, (b) および (c) の 2 つの中から部分検索を行うとき, 部分一致していると判断されるのは (b) のみである. なぜなら, (b) のすべてのサブアセンブリは, それぞれ配置と形状が部分一致するサブアセンブリが (a) に存在しており, 対して (c) では, (a) のどのサブアセンブリとも配置の対応しないサブアセンブリが存在するからである.

提案手法の対象とするアセンブリモデルの部分検索とは, このようなサブアセンブリの配置を考慮したものである. 完全モデルと部分モデルを比較した結果, 部分一致していない体積が最も小さいものを検索結果とする.

実際に作成しているアセンブリモデルデータは, そのデータ形式や目的などの違いによって, サブアセンブリの識別子が文字列や数値などさまざまであり, サブアセンブリ同士の対応関係は未知のものとされる. また, モデルデータにより 3 次元座標におけるモデルの位置や向きもさまざまである. よって, これら的情報は検索に用いることができない.

### 3 関連研究

3 次元モデル検索については多くのアプローチが存在する. Bustos ら [2], Iyer ら [3], Tangelder ら [4] および Li ら [5] は, 一般的な 3 次元 CAD モデルを対象とした検索技術の概要を示している. しかし, 一般的な 3 次元 CAD モデルの検索方法のほとんどは, 形状のみによってモデルを識別しており, アセンブリモデルのアセンブリ構造は考慮されていない.

アセンブリモデルの検索に関する研究はあまり多くはないが, Lupinetti ら [6] は既存の手法を識別子や検索の種類などの基準で分類し, 包括的なレビューを行っている. Miura ら [8] は

モデルのアセンブリ構造をグラフとして表現し, 2 つのアセンブリ間の類似度をグラフマッチングアルゴリズムを用いて計算している. Chen ら [9], Deshmukh ら [7] はグラフ間の部分検索を行うことにより, アセンブリモデルの部分検索を行っている. しかし, アセンブリの構造をグラフによって表現するこれらの手法は, サブアセンブリの配置を記述することが難しく, サブアセンブリの配置や形状を考慮したアセンブリモデルの部分検索に対して十分だとはいえない.

佐藤ら [1] はサブアセンブリの配置や形状を考慮したアセンブリモデルの部分検索を提案しているが, 3 次元モデルのさまざまな方向からの 2 次元投影を 2 次元ラドン変換し, 得られる 3 次元特徴量を用いた検索手法であり, 距離計算の処理に時間がかかるという課題を残している.

### 4 提案手法

本章では, 本稿で提案する 3 次元ラドン変換を用いたアセンブリモデルの部分検索手法について述べる. 部分検索は, クエリのモデルに内包されるようなモデルをデータベースから検索する場合と, クエリのモデルを内包するようなモデルをデータベースから検索する場合の両方が考えられるが, 本稿では前者の場合を前提とする. 後者の場合でも, 4.2 節で後述する差の計算の方向を逆にすることで対応できる.

本稿で提案する 3 次元ラドン変換を用いた 3 次元 CAD アセンブリモデルの部分検索のアルゴリズムを Algorithm1 に示す.

Algorithm1 および以降で用いる記号を定義する.  $M_{comp}$  は完全モデルを,  $M_{part}$  は部分モデルを表す. アセンブリモデル  $M_{comp}$  および部分モデル  $M_{part}$  はサブアセンブリ  $sa$  の集合である.  $V_{comp}$  および  $V_{part}$  は, それぞれ 3 次元ラドン変換を行う方向の集合であり, それらの要素  $v$  は空間極座標で方向を表す 2 次ベクトルである.

Algorithm1 は, 以下に示す 3 つのステップを経て完全モデルと部分モデル間の部分一致距離計算を行う.

(1) 球面上の複数の方向から, 完全モデルと部分モデルの各サブアセンブリに対して 3 次元ラドン変換を行い, 特徴量とする.

(2) すべてのサブアセンブリの組および方向の組に対して, 完全モデルと部分モデルの特徴量から, それらの間の部分一致距離を計算する.

(3) 距離の総和が最小となるようなサブアセンブリの組み合わせを選び, その距離の総和を, ある方向の組み合わせにおける完全モデルと部分モデルの部分一致距離とする.

(4) 距離の総和が最小となる方向の組み合わせを選び, その距離の総和を完全モデルと部分モデルの部分一致距離とする.

ステップ (1) については 4.1 節, ステップ (2) については 4.2 節, ステップ (3), ステップ (4) については 4.3 節にて詳細を述べる.

#### 4.1 3次元ラドン変換を用いたアセンブリモデルの特微量抽出

本節では、4章で示した部分検索アルゴリズムのステップ(1)について詳細を示す。完全モデル $M_{comp}$ と部分モデル $M_{part}$ が与えられたとき、2つのアセンブリモデル間のサブアセンブリの対応は一般的に未知である。また、2つのアセンブリモデルの位置や向きの関係も未知である。よって提案手法では、すべてのサブアセンブリ同士の組み合わせ、すべて方向の組み合わせで比較をし、その中でアセンブリモデル間の距離が最短となるような組み合わせを選択する。

3次元ラドン変換のアルゴリズムをAlgorithm2に示す。方向 $v \in V$ に沿って3次元ラドン変換を行うとき、まずは $v$ がx軸正の方向になるようモデルの回転を行う。Algorithm2では2

---

#### Algorithm 1 ComputeAssemblyPartialDissimilarity

---

**Input:** 完全モデル $M_{comp}$ 、部分モデル $M_{part}$

**Output:** モデル間の部分一致距離  $difference$

```

// ステップ (1)
// 完全モデルの特微量抽出
1: for  $sa_{comp} \in M_{comp}$  do
2:    $F_{comp}(sa_{comp}, :) = Radon3D(sa_{comp}, V_{comp})$ 
3: end for
// 部分モデルの特微量抽出
4: for  $sa_{part} \in M_{part}$  do
5:    $F_{part}(sa_{part}, :) = Radon3D(sa_{part}, V_{part})$ 
6: end for
// ステップ (2)
7: for  $(v_{comp}, v_{part}) \in V_{comp} \times V_{part}$  do
8:    $f_{comp,all} = Sum(F_{comp}(:, v_{comp}))$ 
9:    $f_{part,all} = Sum(F_{part}(:, v_{part}))$ 
10:  for  $(sa_{comp}, sa_{part}) \in M_{comp} \times M_{part}$  do
11:     $f_{comp} = (F_{comp}(sa_{comp}, v_{comp}), f_{comp,all})$ 
12:     $f_{part} = (F_{part}(sa_{part}, v_{part}), f_{part,all})$ 
13:     $d_{sa} = ComputePartialDissimilarity(f_{comp}, f_{part})$ 
14:     $DiffTable_{SA}(sa_{comp}, sa_{part}) = d_{sa}$ 
15:  end for
// ステップ (3)
16:   $d_v = SolveMatchingProblem(DiffTable_{SA})$ 
17:   $DiffTable_V(v_{comp}, v_{part}) = d_v$ 
18: end for
// ステップ (4)
19:  $difference = SolveMatchingProblem(DiffTable_V)$ 
20: return  $difference$ 

```

---

#### Algorithm 2 Radon3D

---

**Input:** サブアセンブリモデル $sa$ 、方向の集合 $V$

**Output:** 特微量  $feature$

```

1: for  $v \in V$  do
2:    $sa' = Rotate(sa, v)$ 
3:   for  $x \in \{x \mid \min(sa'.x) \leq x \leq \max(sa'.x), x|T\}$  do
4:      $feature(v, x) = calculateCrossSectionArea(sa', x)$ 
5:   end for
6: end for
7: return  $feature$ 

```

---

行目の関数 $sa' = Rotate(sa, v)$ に回転を行っており、回転前の $sa$ における $v$ の方向は $sa'$ のx軸正の方向と一致している。

次にx軸正の方向に沿って、x軸に垂直な面でのモデルの断面積を計算する。Algorithm2の4行目の関数 $calculateCrossSectionArea(sa', x)$ は、モデル $sa'$ のx軸方向の値が $x$ であるところでの断面積を求める関数である。ここで $x$ は指定されたサンプリング間隔 $T$ ごとに変化し、モデルの存在する区間を走査する。

実際の3次元モデルデータでは、形状の記録形式はさまざまである。その中でも最も単純な形状の記述形式にSTLファイルの形式がある。これは、すべての主要な3次元データ形式から変換可能な形式で、3次元立体の表面を小さな三角形の面の集合で表す、ポリゴン形式である。このような形式において、幾何学的な計算から断面積が計算できる。幾何計算において、断面に頂点が重なった場合などで縮退が発生するが、これは記号摂動法で解決される。こうして得られる特微量 $feature$ は、非負の値を持つ、 $(\text{方向数}) \times (\text{サンプリング数})$ の二次実数配列である。

#### 4.2 各特微量間の部分一致距離の計算

本節では、4章で示した部分検索アルゴリズムのステップ(2)について詳細を示す。Algorithm1のように、比較する特微量 $f_{comp}, f_{part}$ を決めたとき、特微量間の距離を計算するアルゴリズムをAlgorithm3に示す。

4.1節で述べたとおり、アセンブリモデルは一般に、3次元空間内における姿勢や位置がデータによって異なる。そのため、回転や平行移動に対して頑健なモデル間距離とするため、特微量は平行移動させながらその距離を測る。ただし、アセンブリ内におけるサブアセンブリの相対的位置情報は必要なため、着目しているサブアセンブリを平行移動させると同時に、アセンブリモデル全体の特微量を同時に平行移動させ、これについても同様に距離を測る。

完全モデルのサブアセンブリの特微量 $f_{comp,sa}$ および部分モデルのサブアセンブリの特微量 $f_{part,sa}$ には、着目しているサ

---

#### Algorithm 3 ComputePartialDissimilarity

---

**Input:** 完全モデル特微量 $f_{comp}$ 、部分モデル特微量 $f_{part}$

**Output:** 距離  $dissimilarity$

```

1:  $(f_{comp,sa}, f_{comp,all}) = f_{comp}$ 
2:  $(f_{part,sa}, f_{part,all}) = f_{part}$ 
3:  $dissimilarity = \infty$ 
4: for  $shift = -(\text{部分モデル直径})$  to  $(\text{完全モデル直径})$  do
5:    $dis_{sa} = 0, dis_{all} = 0$ 
6:   for  $i = 0$  to  $(\text{部分モデル直径})$  do
7:      $dis_{sa} = dis_{sa} + max(f_{comp,sa}[i + shift] - f_{part,sa}[i], 0)$ 
8:      $dis_{all} = dis_{all} + max(f_{comp,all}[i + shift] - f_{part,all}[i], 0)$ 
9:   end for
10:   $dis_{sa} = dis_{sa} / Sum(f_{part,sa})$ 
11:   $dis_{all} = dis_{all} / Sum(f_{part,all})$ 
12:   $dissimilarity = min(dissimilarity, dis_{sa} + dis_{all})$ 
13: end for
14: return  $dissimilarity$ 

```

---

アセンブリの形状および配置の情報が含まれており、その差の絶対値  $|f_{comp,sa} - f_{part,sa}|$  は、形状や配置の違いを表す値となる。ここで部分モデルが完全モデルの部分に完全に含まれるとき、 $(f_{comp,sa} - f_{part,sa})$  の要素に正の値は含まれず、0 または負の値のみとなる。このとき、Algorithm3 の 7・8 行目のように、差の値と 0 の大きい方の値を採用することにより、完全に部分一致する場合は特微量間の距離は 0 に、そうでない場合は含まれずにはみ出した部分の体積が特微量間の距離となる。

また、完全モデル全体の特微量  $f_{comp,all}$  および部分モデル全体の特微量  $f_{part,all}$  も同時に比較している。これらはそれぞれ、すべてのサブアセンブリの特微量の和をもって計算される。実際の実装では、全体の特微量を前もって計算することにより、全体の特微量同士の距離計算の回数を減らすことができる。Algorithm3 の 11・12 行目において、サブアセンブリの特微量とアセンブリモデル全体の特微量のそれぞれを自身の体積で割ることにより、それぞれの特微量の距離への影響を等しくしている。この操作を行うことで、サブアセンブリの大きさに関わらず、すべてのサブアセンブリの部分一致距離が、等しい重みでアセンブリモデル間の部分一致距離へと影響する。

#### 4.3 アセンブリモデル間の部分一致距離の計算

本節では、4 章で示した部分検索アルゴリズムのステップ (3)について詳細を示す。まず、Algorithm1 の 14 行目にて、これまでの処理で得られた特微量  $F_{comp}, F_{part}$  について、 $sa_{comp} \in M_{comp}$  と  $sa_{part} \in M_{part}$  のすべての組み合わせの中から、特微量間の距離の和が最小となる組み合わせを解き、そのときの距離の和をアセンブリ間の距離とする。この最適な組み合わせを探す割当問題は、二部グラフ最小重み最大マッチング問題に帰着できる。この問題は最小費用流問題を解ける各種アルゴリズムや、ハンガリアン法を用いて最適解を求めることができる。また、扱う組み合わせの数が膨大な場合は、数理計画法を用いて、短い時間で近似解を期待することもできる。

これを繰り返し、すべての方向の組み合わせにおいてアセンブリ間の距離が求まったとき、次に Algorithm1 の 17 行目にて、アセンブリモデル間の距離を計算する。これは先の流れと同様に、 $v_{comp} \in V_{comp}$  と  $v_{part} \in V_{part}$  のすべての組み合わせの中から、距離の和が最小となる組み合わせを解き、そのときの距離の和をアセンブリ間の距離とする。こうして得られた距離をもとに行う検索は、モデルの姿勢や位置に対して頑健であり、サブアセンブリの識別子にも依存せず、モデルの形状と配置とともに考慮した部分検索となっている。

#### 5 評価実験と考察

提案手法を評価するため、佐藤ら [1] の手法と条件を揃え、比較実験を行い、その処理速度について評価を行う。既存手法には、本稿での提案手法に必要なパラメータに加え、2 次元ラドン変換時に用いる角度の数についてのパラメータが存在する。佐藤ら [1] は、角度数を 180 と設定して実験を行い、検索精度 83% を達成したと示している。本稿では、提案手法と条件を揃えて

表 1: 3 つのアセンブリモデルの詳細情報。

Model	ポリゴン数	部品数	サブアセンブリ数	直径	表面積
Die	79552	39	5	400.0	$7.29 \times 10^5$
Clutch	350519	55	5	400.0	$1.75 \times 10^6$
Gear	393663	56	5	400.0	$1.79 \times 10^6$

比較評価を行うため、この角度数パラメータを方向数パラメータの一種とみなしてパラメータを決定した。よって本稿で示す既存手法を用いた実験では、角度数を方向数に合わせて 20 と設定した。佐藤ら [1] の行った実験よりも処理が 9 倍ほど短縮されているため、検索精度は著しく低下していると思われる。しかし、それと同じパラメータに設定した提案手法において、より短い処理時間でより高い検索精度を達成できることを示すことで、提案手法が処理時間・検索精度の両面において既存手法より優れていることを示す。

実験は、プログラムのソースコードを MATLAB・C++・CUDA によって記述し、Intel Core i7 3.7GHz CPU, 32GB RAM, OS に Windows 10 Education を搭載したコンピュータ上で MATLAB R2019b により実行した。実験に用いた 3 次元アセンブリモデルを図 2 に示す。また、各モデルデータの詳細情報を表 1 に示す。

実験では完全モデルに図 2 に示す配置を用いた。また各形状につき、サブアセンブリの配置の異なる 25 個の部分一致モデルデータを作成し、部分モデルデータベースを作成した。モデル Die における部分モデルデータベースを図 3 に示す。25 個のデータベースモデルのうち 5 つが正しく部分一致するモデルであり、その他 20 のモデルは配置が異なり部分一致しない。実験では、これを各形状のデータベースごとに 5 つのグループに分け、各グループに正解のモデルを 1 つ含むようにした。各グループ内ではサブアセンブリ数および部品数がすべて等しい。また、すべてのモデルに対してランダムな回転と平行移動を加えた。

以降に示す実験では、すべてこの 5 グループそれぞれに対し 3 回ずつ実験を行った。図や表に示す実験結果データは、計 15 回の実験結果の平均となっている。

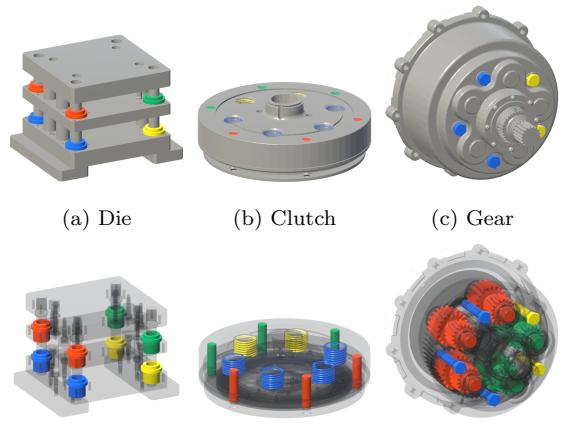


図 2: 実験に用いた 3 つの形状のアセンブリモデル。

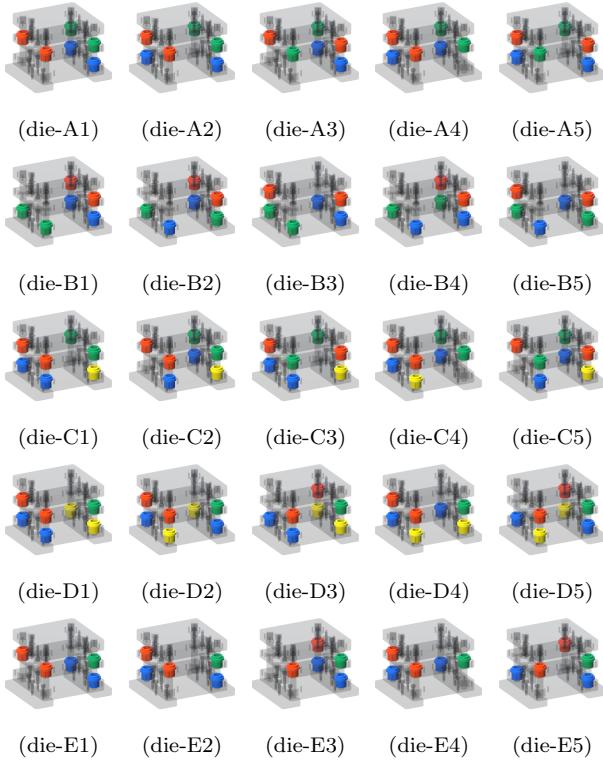


図3: 各5つの配置からなる、5つのDieの部分モデルデータベース。データベースはA,B,C,D,Eの5つがあり、それぞれに1~5の5つの配置の部分モデルが含まれる。

これらのデータを用いて、3種類の実験を行った。まず5.1節では、完全モデルの特微量抽出における方向数を変化させて実験し、処理時間と検索精度について比較する。次に5.2節では、特微量抽出におけるサンプリング数を変化させて実験し、処理時間と検索精度について比較をする。最後に5.3節では、提案手法において、部分モデルと完全モデル両方の方向数を変化させて実験し、処理時間と検索精度の変化を示す。実験結果から、提案手法が処理時間・検索精度・使用メモリのすべてにおいて既存手法よりも優れていることを示す。

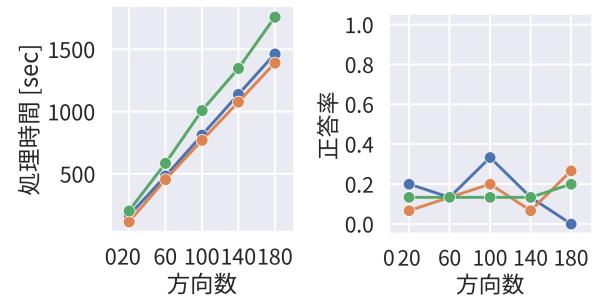
### 5.1 方向数を変化させたときの処理時間と検索精度の比較

完全モデルの特微量抽出における方向数を変化させて実験し、処理時間と検索精度について、既存手法と提案手法を比較する。

実験では、部分モデルの方向数は20で固定とした。また上述の通り、既存手法の2次元ラドン変換における角度数も20とした。既存手法における投影処理と、提案手法における3次元ラドン変換処理のサンプリング間隔はともに20とした。完全モデルの方向数を20, 60, 100, 140, 180と変化させたときの、既存手法での実験結果を図4に、提案手法での実験結果を図5に示す。これらは、完全モデルの特微量抽出・部分モデルの特微量抽出・距離計算・割り当て問題計算の合計時間である。また、処理時間の内訳を、表2と表3に示す。表中の値は、3種類のモデルにおける実験の平均処理時間となっている。

図4(a)および図5(a)の通り、処理時間は方向数に比例する。これは表2および表3から、距離計算時間の削減によるものだ

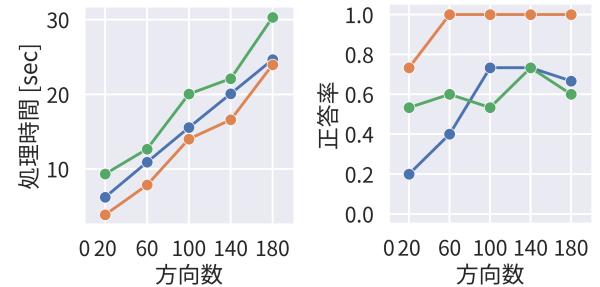
とわかる。特微量の次元の削減により、距離計算の計算量が減少したことによる結果である。提案手法では既存手法に比べ、最大およそ60倍ほど短い処理時間で検索を行えている。



(a) 処理時間の変化.

(b) 検索精度の変化.

図4: 既存手法における、方向数を変化させたときの処理時間および検索精度。



(a) 処理時間の変化.

(b) 検索精度の変化.

図5: 提案手法における、方向数を変化させたときの処理時間および検索精度。

表2: 既存手法における、方向数を変化させたときの処理時間の内訳。Nは完全モデルの方向数。

N	特微量抽出時間 [s]		距離計算時間 [s]	合計時間 [s]
	完全モデル	部分モデル		
20	1.121	5.058	155.582	161.760
60	3.109	5.121	499.555	507.785
100	5.549	5.144	852.163	862.856
140	7.254	5.088	1174.773	1187.115
180	9.604	5.055	1522.618	1537.278

表3: 提案手法における、方向数を変化させたときの処理時間の内訳。Nは完全モデルの方向数。

N	特微量抽出時間 [s]		距離計算時間 [s]	合計時間 [s]
	完全モデル	部分モデル		
20	0.359	3.378	2.746	6.484
60	0.573	3.100	6.810	10.483
100	0.913	3.327	12.299	16.539
140	1.120	3.101	15.373	19.594
180	1.385	3.301	21.620	26.306

さらに図 4(b) および図 5(b) をみると、提案手法は検索精度の向上も実現していることがわかる。これは、既存手法に比べ提案手法がモデルの姿勢に対して頑健な特徴量であるためだと思われる。両手法ともに、方向について離散な特徴量を使用している。ここで、同じモデルに対し少しだけ回転させて特徴量を抽出したとき、既存の手法における投影抽出では、すべての方向の特徴量が回転の影響を受ける。これに対して提案手法における 3 次元ラドン変換では、回転について垂直な方向での特徴量はその影響を受けない。大きな影響を受けるのは、回転と同一平面上に存在する方向での特徴量だが、全球面上でのその割合は小さく、特徴量全体の受ける影響は比較的小さい。よって提案手法は既存手法よりも、回転に対してより頑健であるといえる。

## 5.2 サンプリング間隔を変化させたときの処理時間と検索精度の比較

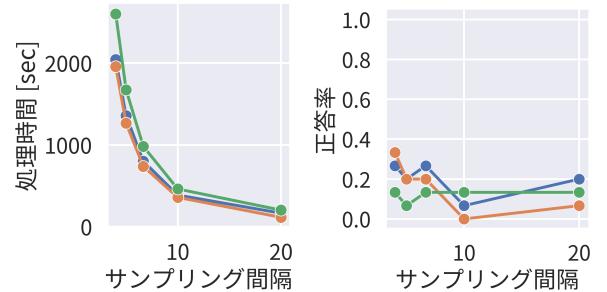
特徴量抽出におけるサンプリング間隔を変化させて実験し、処理時間と検索精度について、既存手法と提案手法を比較する。実験では、完全モデルおよび部分モデルの方向数を 20 で固定した。また上述の通り、既存手法の 2 次元ラドン変換における角度数も 20 とした。既存手法での実験結果を図 6 に、提案手法での実験結果を図 7 に示す。これらは、完全モデルの特徴量抽出・部分モデルの特徴量抽出・距離計算・割り当て問題計算の合計時間である。また、処理時間の内訳を、表 4 と表 5 に示す。表中の値は、3 種類のモデルにおける実験の平均処理時間となっている。

図 6(a) および表 4 より、既存手法はサンプリング数の逆数に比例して処理時間がかかることが分わかる。これは、特徴量抽出および距離計算の計算量が、サンプリング数の逆数に比例するサンプリング数についての線形多項式で表されるからである。また、提案手法についても同様のことがいえるため、処理時間はサンプリング数の逆数に比例すると予想される。しかし、図 7(a) および表 5 では処理時間が十分小さく、線形な変化に近似できる。

結果、提案手法では既存手法に比べ、最大およそ 280 倍ほど短い処理時間で検索を行えている。また、サンプリング間隔をより狭くした場合、速度比はより顕著になると予想される。これは、特徴量の次元の削減により特徴量数が大幅に減り、距離計算の計算量が減少したことによる結果である。

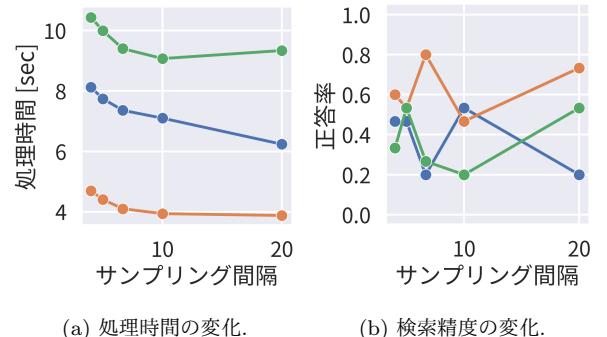
図 4(b) および図 5(b) をみると、最高検索精度は提案手法のほうが高いものの、どちらも結果が安定しておらず、十分な結果を得るには方向数が不十分だと思われる。正答率の上下は、ランダムな回転によって特徴量が変化する影響である。サンプリング間隔と検索精度の関係を見るためには、より多い方向数での実験が必要である。次の 5.3 節にてその結果を示す。

また、表 4 および表 5 には、各種法におけるサンプリング数を示した。既存手法ではサンプリング間隔の逆二乗に、提案手法ではサンプリング間隔の逆数に比例している。提案手法において、特徴量の要素となる Algorithm2 における *feature* は、(方向数) × (サンプリング数) の二次実数配列である。既存手法にてこれに対応する特徴量は、(方向数) × (回転数) × (サンプリング数) の三次実数配列である。本実験では、既存手法の角度数を 20 としているので、提案手法は 20 倍の省メモリとなっている。



(a) 処理時間の変化. (b) 検索精度の変化.

図 6: 既存手法における、サンプリング間隔を変化させたときの処理時間および検索精度.



(a) 処理時間の変化. (b) 検索精度の変化.

図 7: 提案手法における、サンプリング間隔を変化させたときの処理時間および検索精度.

表 4: 既存手法における、サンプリング間隔を変化させたときの処理時間の内訳。T はサンプリング間隔,X はサンプリング数。

T	X	特徴量抽出時間 [s]		距離計算時間 [s]	合計時間 [s]
		完全モデル	部分モデル		
20.00	100	1.121	5.058	155.582	161.760
10.00	200	2.725	12.746	387.559	403.030
6.67	300	5.671	26.108	808.382	840.161
5.00	400	8.839	42.499	1379.959	1431.297
4.00	500	14.177	64.585	2120.238	2199.000

表 5: 提案手法における、サンプリング間隔を変化させたときの処理時間の内訳。T はサンプリング間隔,X はサンプリング数。

T	X	特徴量抽出時間 [s]		距離計算時間 [s]	合計時間 [s]
		完全モデル	部分モデル		
20.00	100	0.359	3.378	2.746	6.484
10.00	200	0.360	3.507	2.835	6.703
6.67	300	0.381	3.586	2.985	6.952
5.00	400	0.406	3.692	3.281	7.379
4.00	500	0.431	3.770	3.548	7.749

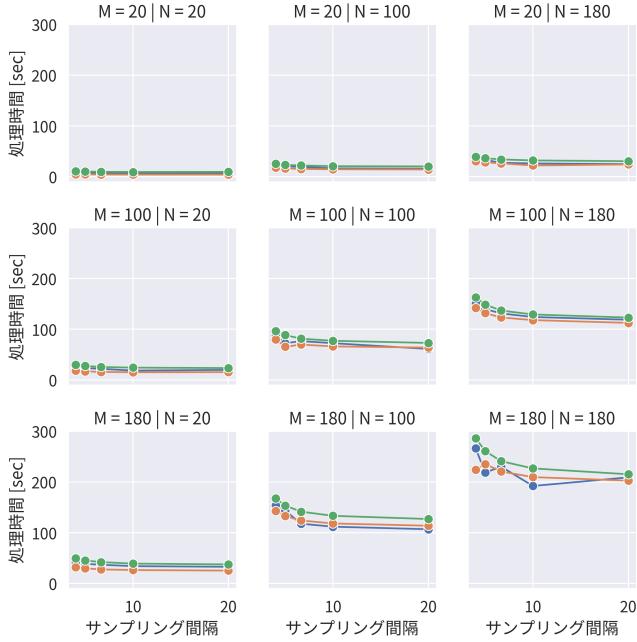


図 8: 提案手法における、複数パラメータを同時に変化させたときの処理時間の変化。 $N$  は完全モデルの方向数、 $M$  は部分モデルの方向数。

### 5.3 提案手法における、複数パラメータを同時に変化させたときの処理時間と検索精度

提案手法の精度および処理時間の評価のため、完全モデルの特徴量抽出における方向数・部分モデルの特徴量抽出における方向数・サンプリング間隔の 3 つのパラメータを同時に動かし、実験を行った。実験では、完全モデルの方向数を 20, 60, 100, 140 と変化させた。部分モデルについては、その方向数が完全モデルの方向数を超えないようにした。また、サンプリング間隔は 5.2 節と同様に変化させた。処理時間の変化を図 8 に、検索精度の変化を図 9 に示した。

図 8 より、処理時間は完全モデルの方向数および部分モデルの方向数の両方に比例するといえる。これは、提案手法が完全モデル・部分モデル問わず同じ方法で特徴量抽出をしているからであり、完全モデルの方向数および部分モデルの方向数については対称な結果が得られる。図 9 より、検索精度は方向数が上がるにつれて向上することが示される。特に完全モデルの方向数が 140、部分モデルの方向数が 100 の場合では、平均正答率が安定して 8 割を超えている。よって提案手法では、適切なパラメータを選択することにより、8 割ほどの検索精度を実現できると思われる。しかし、ランダムな回転による精度のゆらぎも大きく、提案手法では十分に信頼できる検索結果を得ることはできない。また、サンプリング間隔については 5.2 節での結論同様、その変化が検索時間・検索精度に与える影響は比較的小さいと思われる。

また実験の過程にて、本手法の不得意とするサブアセンブリ配置が発見された。図 10 に示すような、鏡像が部分一致するようなアセンブリモデルに対し、誤答することが多かった。今回のモデルでは、サンプリング間隔を狭め、より詳細な特徴量を得

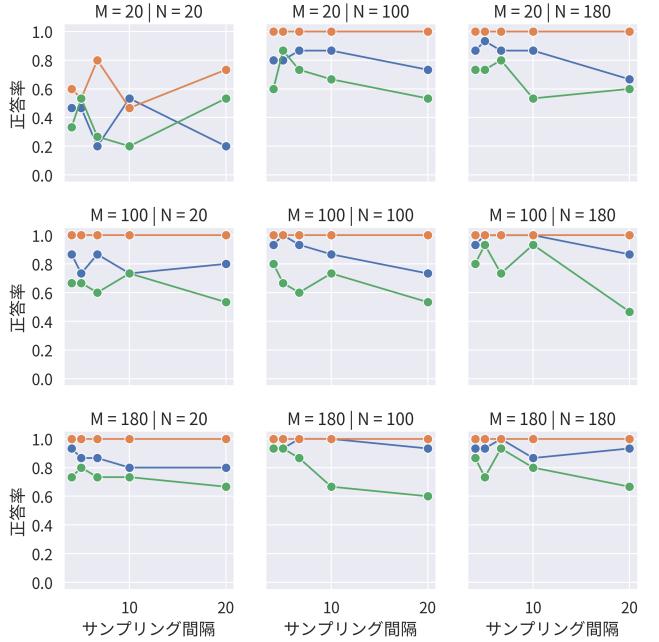
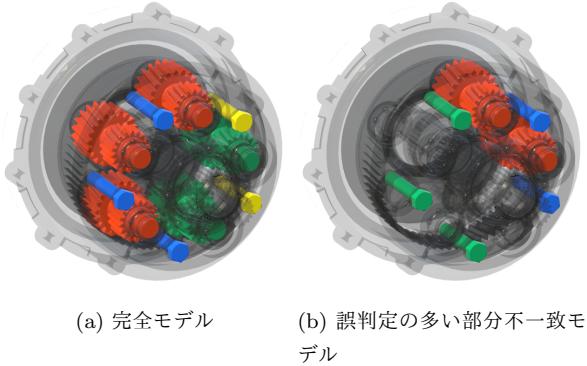


図 9: 提案手法における、複数パラメータを同時に変化させたときの検索精度の変化。 $N$  は完全モデルの方向数、 $M$  は部分モデルの方向数。



(a) 完全モデル  
(b) 誤判定の多い部分不一致モデル

図 10: 完全モデルと、部分一致だと誤判定の多かった、部分不一致なアセンブリモデル。(b) の部分不一致モデルの鏡像は、部分一致モデルとなる。

ることで鏡像を区別することができている。しかしながら、理論的には提案手法では、完全な鏡像である 2 つのモデルを区別することができない。この問題は佐藤ら [1] の手法でも存在しているが、見落とされていた。

以上の結果から、提案手法は既存の手法から高速化・省メモリ化・高精度化できていると示された。これにより、短期間でより多くの実験を行ったり、より大規模なさまざまなモデルについての実験を行うことができる。しかし実際の現場では、今回用いたモデルよりも大きく複雑なモデルが無数に存在するため、さらなる改良が必要である。

## 6 おわりに

本研究では、3 次元ラドン変換を用いて特徴量次元を削減した、新たな部分検索手法を提案した。実験により、提案手法は既存の

手法から高速化・省メモリ化・高精度化できていると示された。これにより、今後はさまざまな検索条件を考慮したより高度な検索や、より大規模かつ複雑なアセンブリモデルデータベースについての検索などを検討することができる。

しかし、ランダムな回転による精度のゆらぎが大きく、また鏡像であるモデルを区別できないという課題も見つかり、実用化に向けてはさらなる改良が求められる。現在我々は、離散的な特徴量を補間することで連続的な特徴量とすることにより、これらの課題を解決し、処理時間をさらに短縮する研究に取り組んでいる。

## 文 献

- [1] 佐藤 渉, & 片山 薫 (2008). 3 次元 CAD モデルのための部品配置を考慮したサイノグラムを用いた部分モデル検索. DEIM Form 2018 I1-4.
- [2] Bustos, B., Keim, D. A., Saupe, D., Schreck, T., & Vranić, D. V. (2005). Feature-based similarity search in 3D object databases. ACM Computing Surveys (CSUR), 37(4), 345-387.
- [3] Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., & Ramani, K. (2005). Three-dimensional shape searching: state-of-the-art review and future trends. Computer-Aided Design, 37(5), 509-530.
- [4] Tangelder, J. W., & Veltkamp, R. C. (2004, June). A survey of content based 3D shape retrieval methods. In Proceedings Shape Modeling Applications, 2004. (pp. 145-156). IEEE.
- [5] Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., ... & Fu, H. (2015). A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. Computer Vision and Image Understanding, 131, 1-27.
- [6] Lupinetti, K., Pernot, J. P., Monti, M., & Giannini, F. (2019). Content-based CAD assembly model retrieval: Survey and future challenges. Computer-Aided Design, 113, 62-81.
- [7] Deshmukh, A. S., Banerjee, A. G., Gupta, S. K., & Sriram, R. D. (2008). Content-based assembly search: A step towards assembly reuse. Computer-aided design, 40(2), 244-261.
- [8] Miura, T., & Kanai, S. (2009). 3d shape retrieval considering assembly structure. In Proceedings of Asian Symposium for Precision Engineering and Nanotechnology.
- [9] Chen, X., Gao, S., Guo, S., & Bai, J. (2012). A flexible assembly retrieval approach for model reuse. Computer-Aided Design, 44(6), 554-574.