

# 3次元ラドン変換を用いた多階層アセンブリモデルの検索手法

平島 栄志<sup>†</sup> 片山 薫<sup>†</sup>

<sup>†</sup> 東京都立大学システムデザイン研究科情報科学域 〒191-0065 東京都日野市旭が丘 6-6

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

**あらまし** 近年の製造業などでは、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が利用されているが、それらのデータの活用技術に大きな関心が集まっている。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く、特に配置構造が多階層となっているアセンブリモデルを扱える手法はさらに少ない。そこで本研究では、配置構造が3階層であるアセンブリモデルについて提案されている既存手法をもとに、配置構造が多階層であるアセンブリモデルを効率的に扱える新たな手法を提案する。提案手法では、はじめにモデルのアセンブリ構造を無視して全体の形状に対して3次元ラドン変換を行い、その結果を元に、モデル間の回転や平行移動量を推定する。その後、階層的な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相違度を計算する。こうすることで、部品ごとに回転や平行移動量を計算する過程を省略でき、既存手法に比べて全体の処理時間が大幅に削減できる。本研究では、形状や配置の異なる複数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、提案手法を用いることで、配置構造が多階層であるアセンブリモデルのデータベースから完全一致検索が可能であることを示した。この成果は、既存手法では扱えなかった複雑なアセンブリモデルのデータベースにおいても検索が可能になるという点で重要である。

**キーワード** 3D モデル, CAD, 多階層アセンブリモデル, 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の企業に蓄積している。近年では多くの産業にてDX(デジタルトランスフォーメーション)が急務となっており、それに伴い3次元モデルデータの活用に大きな注目が集まっている。しかし複数部品からなるアセンブリモデルデータのような、大規模かつ複雑な3次元モデルデータの活用には、実行速度面などに多くの課題が残っている。

本研究では、形状と部品の配置構造のみを利用した、多階層アセンブリモデルの部分検索手法を提案する。提案手法では、過去に我々の提案した3次元ラドン変換を用いた3階層アセンブリモデルの完全一致検索手法[1]および部分検索手法[2]を用いて多階層アセンブリモデルの部分一致検索を行う。これらの手法では処理の高速化および省メモリ化のために3次元ラドン変換が用いられており、本研究ではこれを利用し、より複雑な処理の求められる多階層アセンブリモデルの部分検索を実現した。

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多階層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多階層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た3次元モデルである。例えば1台の自動車は、車体や動力系、内装などの部分から構成されており、動力系はエンジンやブレーキシステム等の複合体であり、さらにエンジンもまた多くの部品から構成されている。このようにアセンブリモデルとは、複数階層からな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ものを“サブアセンブリ”と呼ぶ。そして木構造の根の位置に当たる、全ての部品やサブアセンブリを組み合わせたものをアセンブリモデルと呼ぶ。本研究では、これまでの手法で扱うことの難しかった3階層以上の構造を持ったアセンブリモデルを特に多階層アセンブリモデルと呼ぶ。

アセンブリモデルにおける部分一致検索では、ある大きいモデルに含まれるような小さいモデルを検索する場合と、ある小さいモデルを含むような大きいモデルを検索する場合の2種類の場合が考えられる。以降では、この大きいモデルを完全モデルと呼び、小さいモデルを部分モデルと呼ぶ。提案手法は2つのモデルの距離を計算することを目的としているので、どちらの場合にも対応することができる。

ここで部分モデルが完全モデルに部分一致しているとは、完全モデルからいくつかの部品を取り除くことで部分モデルと完全に同じモデルを作ることができることとする。つまり、完全モ

デルのうち部分モデルに一致する部分は、そのアセンブリ構造や部品の位置・形状も部分モデルと完全に同じ出なければならない。ここで、完全モデルと部分モデルにて同じ位置にある2つの部品について形状が異なり、部分モデルの部品が小さく完全モデルの部品に内包されるような形状であった場合でも、これは部分一致しているとは言わないこととする。また、2つのアセンブリモデルの階層構造を木グラフとして考えた場合、完全モデルの木グラフに対し根が一致しないような部分木グラフを考えることができる。本研究では、このような状態のモデルは完全モデルに部分一致していないものと定義し、提案する手法でも部分一致していないものと判定する。

そして今回提案する手法は、アセンブリモデル全体の位置や姿勢に依存することなく部分一致検索を行うことができる。つまり、アセンブリモデルの3Dデータは3D空間上のある座標軸に従って記述されているが、一般に3Dモデルは座標の基準と点や向きは決まっておらず、それらはデータ製作者が自由に決めることができる。3Dモデル検索手法は、原点や向きの異なる座標系においても等しい検索結果を出すことが求められており、本稿の提案手法もこれを満たしている。

### 3 関連研究

3次元モデル検索については多くの手法が提案されている。Bustos ら[3], Iyer ら[4], Tangelder ら[5] および Li ら[6] は、3DCADモデルを対象とした検索手法およびモデル間の距離計算手法を提案している。しかし、これらの検索手法が対象とする3DCADモデルは、アセンブリ構造を持っていないものであり、モデル全体の形状のみによってモデルを識別している。

アセンブリモデルの検索手法については、Lupinetti ら[7] が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[10] 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、これらの手法はアセンブリ構造の関係については正しく比較できているものの、それを構成するサブアセンブリの配置や部品の形状について正確な比較は行っていない。

佐藤ら[11] 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、特徴量の計算に2次元ラドン変換を用いた検索手法であり、距離計算の処理に膨大な時間がかかってしまう。我々はこの問題の解決として、特徴量の計算に3次元ラドン変換を用いた部分検索手法[2]を提案した。しかしこれは3階層以下のアセンブリモデルしか処理できないという欠点があった。

そこで本稿では、3次元ラドン変換を用いた多階層アセンブリモデルの部分一致検索手法を提案する。

---

#### Algorithm 1 CalcMulti-levelAssemblyPartialDistance

---

**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  $dist$

```

// (1) 両アセンブリモデルに対し 3 次元ラドン変換を行う
1: for  $i = 1$  to  $V_{comp}.Count$  do
2:    $R_{comp}[i] = AssemblyRadon3D(M_{comp}, V_{comp}[i])$ 
3: end for
4: for  $i = 1$  to  $V_{part}.Count$  do
5:    $R_{part}[i] = AssemblyRadon3D(M_{part}, V_{part}[i])$ 
6: end for
// (2) アセンブリ構造を無視した部分一致距離計算を行う
7: for  $i = 1$  to  $V_{comp}.Count$  do
8:   for  $j = 1$  to  $V_{part}.Count$  do
9:      $Root_{comp} = R_{comp}[i].Radon$ 
10:     $Root_{part} = R_{part}[j].Radon$ 
11:     $(d, s) = CalcPartialDist(Root_{comp}, Root_{part})$ 
12:     $DistTable[i, j] = d$ 
13:     $ShiftTable[i, j] = s$ 
14:   end for
15: end for
// (3) 方向の割り当て候補を決定する
16: for  $i = 1$  to  $numCandidates$  do
17:    $idxPairs = HungarianAlgorithm(DistTable)$ 
18:   for  $(i, j) \in idxPairs$  do
19:      $isSelected[i, j] = true$ 
20:      $DistTable[i, j] = 1.0$ 
21:   end for
22: end for
// (4) アセンブリ構造考慮した部分一致距離計算を行う
23: for  $i = 1$  to  $V_{comp}.Count$  do
24:   for  $j = 1$  to  $V_{part}.Count$  do
25:     if  $isSelected[i, j]$  then
26:        $shift = ShiftTable[i, j]$ 
27:        $DistTable[i, j] = CalcMultiLevelPartialDist(R_{comp}[i], R_{part}[j], shift)$ 
28:     else
29:        $DistTable[i, j] = 1.0$ 
30:     end if
31:   end for
32: end for
// (5) 方向の組み合わせを最終決定する
33:  $idxPairs = HungarianAlgorithm(DistTable)$ 
34:  $dist = 0$ 
35: for  $(i, j) \in idxPairs$  do
36:    $dist += DistTable[i, j]$ 
37: end for
38:  $dist /= idxPairs.Count$ 
39: return  $dist$ 

```

---

### 4 提案手法

本章では、本稿で提案する多階層アセンブリモデルの部分検索手法について述べる。提案手法の全体の流れを、Algorithm1 に示す。

Algorithm1 および以降で用いる用語および記号を定義する。

完全モデル  $M_{comp}$  および部分モデル  $M_{part}$  は任意の深さの階層を持つアセンブリモデルであり、各部品の 3D モデルデータの集合と、アセンブリ構造を示す木グラフデータの組みである。1 つのアセンブリモデルデータにおいて各部品の 3D モデルデータはすべて共通の 3 次元座標上で記述されており、個々の形状だけでなく、部品の位置や姿勢の情報も含んでいる。  $V_{comp}$  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、次に示す 5 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全部品・全階層のサブアセンブリに対して 3 次元ラドン変換を行う。

(2) 任意の方向の組み合わせに対して、完全モデル全体の 3 次元ラドン変換結果と部分モデル全体の 3 次元ラドン変換結果を比較し、それらの間の部分一致距離を計算する。これはアセンブリ構造を無視し、アセンブリモデル全体を一つのモデルとして見た場合の部分一致距離である。またこのとき、最適な平行移動量も記録しておく。

(3) ステップ (2) の結果を元に、距離の総和が最小となるような方向の割り当て候補を、指定した数だけ選ぶ。この割り当ての選択にはハンガリアン法を用いる。

(4) 選ばれた方向の組それぞれに対して、階層的なアセンブリ構造を考慮しながら部分一致距離計算を行う。

(5) 計算された方向の割り当て候補の中から、距離の和が最小となるような方向の組み合わせを選ぶ。選ばれた全ての方向の組における距離の総和が、二つのアセンブリモデル間の部分一致距離である。

ここで、多階層アセンブリモデル間の部分一致距離計算を実際に行っているのはステップ (1) とステップ (4) である。ステップ (2) とステップ (3) は、あらかじめ方向の組み合わせや平行移動量に目星をつけることで、ステップ (4) の各階層における距離計算で個々に探索を行うことを防ぎ、全体の処理時間を短縮している。

ステップ (1) については 4.1 節、ステップ (2)、ステップ (3) については 4.2 節、ステップ (4)、ステップ (5) については 4.3 節にて詳細を述べる。

#### 4.1 多階層アセンブリモデルの 3 次元ラドン変換

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。Algorithm1 の 2・5 行目のように、多階層アセンブリモデルと方向を受け取って 3 次元ラドン変換結果を返すアルゴリズムを Algorithm2 に示す。また、Algorithm2 内で用いる、部品の 3D データと方向を受け取って 3 次元ラドン変換結果を返すアルゴリズムを Algorithm3 に示す。

Algorithm2 では、アセンブリモデルの 3 次元ラドン変換を行っている。これは 3 章にて挙げた部分一致検索手法 [2] 内のものと本質的に等しい。具体的には、6 行目や 9 行目にて子要素の 3 次元ラドン変換を行い、15～20 行目でそれらの和を取り、それを自身のラドン変換結果としている。アセンブリモデルが複数階層

#### Algorithm 2 AssemblyRadon3D

**Input:** アセンブリモデル  $M$ , 方向  $V$

**Output:** 数値配列  $radon$ , 先端の位置  $minSide$ , 子のラドン結果  $children$

```

1:  $minSide = \infty$ 
2:  $maxSide = -\infty$ 
3:  $children = \{\}$ 
4: for  $child \in M.Children$  do
5:   if  $child.Children.Count == 0$  then                                ▶  $child$  が部品の場合
6:      $(r, m) = PartsRadon3D(child, V)$ 
7:      $c = \{\}$ 
8:   else                                                                ▶  $child$  がアセンブリモデルの場合
9:      $(r, m, c) = AssemblyRadon3D(child, V)$ 
10:   end if
11:    $minSide = \min(minSide, m)$ 
12:    $maxSide = \max(maxSide, m + r.length + 1)$ 
13:    $children.add(\{Radon = r, minSide = m, Child = c\})$ 
14: end for
15:  $radon = []$ 
16: for  $child \in children$  do
17:   for  $x = 0$  to  $child.Radon.length$  do
18:      $radon[x + child.minSide - minSide] += child.Radon[x]$ 
19:   end for
20: end for
21: return  $radon, minSide, children$ 

```

#### Algorithm 3 PartsRadon3D

**Input:** 部品の 3D モデル  $M$ , 方向  $V$

**Output:** 数値配列  $radon$ , 先端の位置  $minSide$

```

1:  $m = Rotate(M, V)$ 
2:  $minSide = \lceil \min(m.Vertexes.X) \rceil$ 
3:  $maxSide = \lfloor \max(m.Vertexes.X) \rfloor$ 
4:  $radon = []$ 
5: for  $x = minSide$  to  $maxSide$  do
6:    $radon[x - minSide] = calculateCrossSectionArea(m, x)$ 
7: end for
8: return  $radon, minSide$ 

```

を持つとき、この Algorithm2 が再帰的に呼ばれる。

Algorithm3 では、部品 1 つの 3D モデルを 3 次元ラドン変換している。まず 1 行目にて、与えられた方向が  $x$  軸方向と成るよう 3D モデルを回転させる。これは後の計算を簡単にし、数値誤差を減らす効果がある。そして 5～7 行目にて、3D モデルを  $x$  軸に垂直な面でスライスし、その断面積を計算する。スライスを繰り返し、断面積の配列を得ることが 3 次元ラドン変換である。このとき、3D モデルと交わらない面での断面積は 0 であるので、計算不要である。2・3 行目では計算が必要なスライス面の範囲を決めている。

また Algorithm3 においてスライス面の間隔は 1 としているが、この間隔を狭めることでより緻密なラドン変換結果を得ることができる。

#### 4.2 平行移動量の決定および方向の割り当て候補の選出

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 (2),(3)

---

**Algorithm 4** CalcPartialDist

---

**Input:** 完全モデル全体のラドン変換結果  $Root_{comp}$ ,

部分モデル全体のラドン変換結果  $Root_{part}$

**Output:** 部分一致距離  $dist$ , 最適な平行移動量  $bestShift$

```
1:  $dist = \infty$ 
2:  $bestShift = NA$ 
3: for  $shift = -\infty$  to  $\infty$  do
4:    $d = 0$ 
5:   for  $idx_{part} = 0$  to  $Root_{part}.length$  do
6:      $idx_{comp} = idx_{part} + shift$ 
7:      $d += \max(0, Root_{part}[idx_{part}] - Root_{comp}[idx_{comp}])$ 
8:   end for
9:   if  $d < dist$  then
10:     $dist = d$ 
11:     $bestShift = shift$ 
12:   end if
13: end for
14:  $dist /= \sum(Root_{part})$ 
15: return  $dist, bestShift$ 
```

---

について詳細を示す。ステップ (2) では、任意の方向の組み合わせに対して、アセンブリ構造を無視した部分一致距離を計算している。まず 9・10 行目にて、各アセンブリモデル全体のラドン変換結果のみを取り出している。これは全ての部品のラドン変換結果の合計であり、部品や階層の区別なくアセンブリモデル全体の形状のみを表している。

次に、Algorithm2 内で用いている、アセンブリ構造を無視した部分一致距離を返すアルゴリズムを Algorithm4 に示す。Algorithm4 では、2 つの 3 次元ラドン変換結果を元に、それらの部分一致距離を計算している。部分モデルは完全モデルのどこで部分一致しているかわかっておらず、2 つの 3 次元ラドン変換結果の相対的な位置関係は決まっていない。そこで Algorithm4 では、6 行目のように、部分モデルの 3 次元ラドン変換結果の最初の要素を基準に、完全モデルの 3 次元ラドン変換結果の最初の要素が  $shift$  だけずれていると仮定し、4~8 行目で部分一致距離計算を行っている。

ここで部分一致距離は、部分モデルの 3 次元ラドン変換結果のうち完全モデルの 3 次元ラドン変換結果から”はみ出す”部分の割合で定義している。よってこの部分一致距離は、0~1 の値を取る。Algorithm4 の 7 行目では、 $Root_{part}[idx_{part}]$  のうち  $Root_{comp}[idx_{comp}]$  からはみ出している量を計算している。ここで  $Root_{comp}[idx_{comp}]$  が範囲外参照となる場合があるが、このときこれは 0 と評価されるものとする。その後 14 行目にて、この総和を  $Root_{part}$  の総和で割ることにより、はみ出す部分の割合を部分一致距離としている。

また、Algorithm4 の 5 行目の通り、任意の  $shift$  にて距離計算を行い、9~12 行目にて最も距離が小さくなる  $shift$  を採用している。実際のプログラムでは任意の  $shift$  を試す必要はなく、2 つの 3 次元ラドン変換結果が少しでも重なる場合のみ調べれば良い。

ステップ (3) では、距離の総和が小さくなるような、方向の割り当て候補を選ぶ。これは費用を最小化する割り当て問題とし

て解くことができ、一般にハンガリアン法というアルゴリズムが知られている。ここではハンガリアン法を用いて最適な方向の組み合わせを選んでいく。  $V_{comp}$  の要素数と  $V_{part}$  の要素数が異なる場合、要素が少ない方の数だけ方向の組み合わせが選ばれる。選ばれた割り当てを記録した後、その距離を大きい値で上書きして再び割り当て問題を解くことで、複数の割り当て候補を重複することなく選出できる。

このように、あらかじめアセンブリ構造を無視した部分一致距離を用いて最適な方向の組み合わせや平行移動量を絞り込むことで、アセンブリ構造を考慮しながらそれらパラメータを様々な試行することを防いでいる。これはアセンブリ構造が多くの部品・階層を持つ場合において、強力な高速化となる。

### 4.3 階層的なアセンブリ構造を考慮した部分一致距離計算

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 (4) について詳細を示す。ここでは、ステップ (2)(3) で採用した方向の組み合わせ・平行移動量に対して、アセンブリ構造を考慮した部分一致距離を計算している。Algorithm1 の 27 行目で呼び出している、アセンブリ構造を考慮した部分一致距離を返すアルゴリズムを Algorithm5 に示す。また、Algorithm5 の 15 行目で呼び出している、部品同士の完全一致距離を返すアルゴリズムを Algorithm6 に示す。

Algorithm5 は、以下に示す 4 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。

(a) 全く部分一致しない場合は即座に距離 1 を返す。これには、一方のモデルが部品でもう一方がアセンブリモデルの場合や、部分モデルの 3 次元ラドン結果の総和が完全モデルのその倍以上大きい場合が該当する。

(b) 完全モデルと部分モデルが共に部分モデルの場合、Algorithm6 を用いて部品同士の完全一致距離を計算し、モデル間の部分一致距離とする。

(c) 完全モデルと部分モデルが共に部分モデルの場合、子要素の組み合わせをすべて試し、各子要素間の部分一致距離を再帰的に計算する。

(d) 合計距離が最小となるように子要素の組み合わせを決定し、その時の合計距離をモデル間の部分一致距離とする。

Algorithm6 は、2 つの 3 次元ラドン変換結果を元に、部品間の完全一致距離を計算している。これは 3 章にて挙げた完全一致検索手法 [1] 内のものと本質的に等しい。2 つの 3 次元ラドン変換結果の相対的な位置関係与えられた  $shift$  を利用するが、基本的には Algorithm4 と同じ構造をしている。ここで完全一致距離は、2 つの 3 次元ラドン変換結果の”重なっていない”部分の割合で定義している。よってこの完全一致距離は、0~1 の値を取る。Algorithm6 の 4 行目では、 $Root_{part}[idx_{part}]$  と  $Root_{comp}[idx_{comp}]$  の重なっていない量を計算している。ここで  $Root_{comp}[idx_{part}]$  や  $Root_{comp}[idx_{comp}]$  が範囲外参照となる場合があるが、このときこれは 0 と評価されるものとする。その後 6 行目にて、この総和を  $Root_{part}$  の総和  $Root_{comp}$  の総和の和で割ることにより、重なっていない部分の割合を完全一致距離としている。このように、Algorithm5 はその内部にて Algorithm6 を、もしくは再帰的に

**Algorithm 5** CalcMultiLevelPartialDist**Input:** 完全モデルのラドン変換結果  $R_{comp}$ ,部分モデルのラドン変換結果  $R_{part}$ , 平行移動量  $shift$ **Output:** 部分一致距離  $dist$ // (a) 全く部分一致しない場合は  $dist = 1$  を返す

```

1: if  $R_{comp}.Children.Count == 0$  and  $R_{part}.Children.Count != 0$  then
2:   return  $dist = 1$            ▶  $R_{comp}$  が部品,  $R_{part}$  がアセンブリモデル
3: else if  $R_{comp}.Children.Count != 0$  and  $R_{part}.Children.Count == 0$  then
4:   return  $dist = 1$            ▶  $R_{comp}$  がアセンブリモデル,  $R_{part}$  が部品
5: else if  $2 \times \sum(R_{comp}.Radon) < \sum(R_{part}.Radon)$  then
6:   return  $dist = 1$            ▶  $R_{part}$  のほうが倍以上大きい
7: end if

// (b) 共に部品の場合, 完全一致距離を計算する
8: if  $R_{comp}.Children.Count == 0$  and  $R_{part}.Children.Count == 0$  then
9:   return CalcCompleteDist( $R_{comp}.Radon$ ,  $R_{part}.Radon$ ,  $shift$ )
10: end if

```

// (c) 共にアセンブリモデルの場合, 全ての子要素の組み合わせで距離計算を行う

```

11: for  $i = 0$  to  $R_{comp}.Children.Count$  do
12:   for  $j = 0$  to  $R_{part}.Children.Count$  do
13:      $child_{comp} = R_{comp}.Children[i]$ 
14:      $child_{part} = R_{part}.Children[j]$ 
15:      $DistTable[i, j] = CalcMultiLevelPartialDist(child_{comp}, child_{part})$ 
16:   end for
17: end for

// (d) 距離が最小となるような, 最適な子要素の組み合わせを選ぶ
18:  $idxPairs = HungarianAlgorithm(DistTable)$ 
19:  $dist = \max(0, R_{part}.Children.Count - idxPairs.Count)$ 
20: for  $(i, j) \in idxPairs$  do
21:    $dist += DistTable[i, j]$ 
22: end for
23:  $dist /= R_{part}.Children.Count$ 
24: return  $dist$ 

```

**Algorithm 6** CalcCompleteDist**Input:** 完全モデル部品のラドン変換結果  $R_{comp}$ ,部分モデル部品のラドン変換結果  $R_{part}$ , 平行移動量  $shift$ **Output:** 完全一致距離  $dist$ 

```

1:  $dist = 0$ 
2: for  $idx_{part} = -\infty$  to  $\infty$  do
3:    $idx_{comp} = idx_{part} + shift$ 
4:    $dist += |Root_{part}[idx_{part}] - Root_{comp}[idx_{comp}]|$ 
5: end for
6:  $dist /= \sum(Root_{comp}) + \sum(Root_{part})$ 
7: return  $dist$ 

```

Algorithm5 を呼び出している。

Algorithm5 のステップ (3) では, 得られた方向の組み合わせごとの距離に対し, 再度ハンガリアン法を用いて最適な方向の割り当てを一つ選んでいる。

Algorithm5 にて部品間の完全一致距離は 0~1 の値を取り, 23 行目にて子要素ごとの距離の和を子要素の数で割っているため, Algorithm5 も 0~1 の値を返す。よって最終的に, Algorithm1 は各方向の組み合わせごとに部分一致距離を 0~1 の値で得る。

表 1: 実験に用いた 20 個の多階層アセンブリモデルの詳細情報。

	Model	部品数	ノード数	階層の深さ
1~5	完全モデル A~E	39	74~6	5
6~10	部分モデル a~e	36~37	68~71	5
11~15	完全モデル F~J	55	77	5
16~20	部分モデル f~j	49	69	5

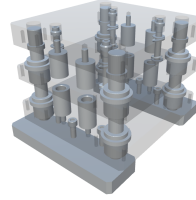
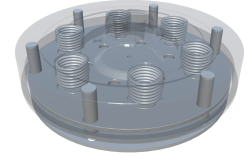
(a) 回転対称性の低いモデル  
(2 回対称)(b) 回転対称性の高いモデル  
(6 回対称)

図 1: 実験に用いた 2 つの形状のアセンブリモデル。

36 行目にてこの総和を方向の組み合わせの数で割っているため, Algorithm1 によって得られるアセンブリモデル間の部分一致距離も 0~1 の値となる。

## 5 評価実験と考察

本章では, 本稿で提案する多階層アセンブリモデルの部分検索手法を実験的に評価する。まず 5.1 節では, 提案手法が多階層アセンブリモデルについて正しく部分一致検索できることを実験的に確認する。次に 5.2 節では, 既存の 3 階層アセンブリモデルの部分検索手法 [2] と同じ条件で 3 階層アセンブリモデルの部分検索を行うことにより, 提案手法を処理速度の観点から比較評価を行う。

提案手法を用いた実験は, プログラムのソースコードを C# によって記述し VisualStudio2018 にてコンパイルしたものを, 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実行した。また, 既存手法 [2] を用いた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, 提案手法の実験と同じコンピュータ上で MATLAB R2019b により実行した。

### 5.1 多階層アセンブリモデル部分検索実験

本節では, 提案手法を用いた多階層アセンブリモデル部分検索実験結果を示し, 提案手法の有用性を評価する。実験のために, 20 個の異なる多階層アセンブリモデルを用意した。各モデルデータの詳細情報を表 1 に示す。なお, 表 1 中のノード数とは, アセンブリ構造を木グラフで示した際のノード数であり, 個々の部品に相当する葉ノードやアセンブリモデル全体に相当する根ノードもその数に含んでいる。階層の深さとは, アセンブリモデル全体に相当する根ノードを深さ 1 としたときの, 最も深い階層にある部品の深さを示している。

表 2: 多階層アセンブリモデル間の部分一致距離計算結果例.

部分モデル	a	b	c	d	e
完全モデル A	<b>0.21763</b>	0.30220	0.48603	0.55706	0.38383
完全モデル B	0.29623	<b>0.21322</b>	0.52955	0.47682	0.42986
完全モデル C	1.00000	1.00000	<b>0.24222</b>	0.30364	0.37506
完全モデル D	1.00000	1.00000	0.30300	<b>0.22126</b>	0.42204
完全モデル E	0.34560	0.39840	0.49998	0.55837	<b>0.27517</b>

提案手法は、実験対象とするアセンブリモデルの回転対称性によって検索精度が大きく左右される。よって図 1 のような回転対称性の低い形状をしたアセンブリモデルと、回転対称性の高い形状をしたアセンブリモデルを用意し、対象アセンブリモデルの回転対称性の違いによる比較を行った。表のアセンブリモデル 1~10 は回転対称性の低いモデルであり、アセンブリモデル 11~20 は回転対称性の高いモデルである。

完全モデル A~E は、全て同じ部品セットから構成されており、非常に似通った構造をしている。完全モデル A と完全モデル B は、あるサブアセンブリを構成する 4 つの部品のうち 2 つが、同階層内の別のサブアセンブリを構成する 4 つの部品のうち 2 つと入れ替わっただけの違いがある。どちらも各ノードにおける子要素の数などだけでは区別不可能であり、さらに入れ替わった部品はすべて同じ形状をしているため、部品の配置を正しく識別できる場合にのみ完全モデル A と完全モデル B は区別される。次に完全モデル A と完全モデル C および完全モデル B と完全モデル D は、それぞれ同じ階層にある 2 つのノードが統合され 1 つのノードとなっている。つまり、アセンブリ構造を正しく考慮できる場合にのみ完全モデル A と完全モデル C は区別される。最後に、完全モデル E は完全モデル A の一部を 1 階層深くしたモデルである。つまり、各サブアセンブリが同じノードに属しているかだけでなく、その深さまで正しく考慮できる場合にのみ完全モデル A と完全モデル E は区別される。

そして表 1 内の部分モデル a~e は、それぞれ 1 つずつ部分一致する完全モデルが存在している。つまり 5 つの完全モデルと 5 つの部分モデル全ての組み合わせで部分一致距離を計算することにより、それぞれの場合で部分一致距離が正しく計算できているのかを評価することができる。完全モデル F~J および部分モデル f~j についても同様の対応がある。

アセンブリモデル 1~10 を用いて総当りで距離計算を行った結果の例を表 2 に示す。まず表 2 を各行ごとに見た場合、これはアセンブリモデルのデータベースに対し完全モデルをクエリとし、それに部分一致するような部分モデルを検索するというケースに相当する。完全モデル A の行では部分モデル a との距離が最も小さくなっており、これは正しく検索できていると言える。またこの他のすべての行においても正しく検索できている。よってこの例では、提案手法は回転対称性の低い多階層アセンブリモデルにおいて正しく部分一致検索を行うことができている。

同様に、全てのモデルに対しパラメータを変化させながら実験を行った平均結果を表 3、表 4 に示す。実験では、部分モデル

表 3: 多階層アセンブリモデル間の部分一致検索結果。(回転対称性の低いモデル)

部分モデルの方向の数		50	100	150	200
1 候補	平均精度	0.95	1.00	1.00	1.00
	平均処理時間 [s]	1.26	1.65	2.01	2.39
5 候補	平均精度	0.50	1.00	1.00	1.00
	平均処理時間 [s]	2.97	3.36	3.74	4.12

表 4: 多階層アセンブリモデル間の部分一致検索結果。(回転対称性の高いモデル)

部分モデルの方向の数		50	100	150	200
1 候補	平均精度	0.60	0.80	0.70	0.70
	平均処理時間 [s]	3.97	5.12	6.30	7.47
5 候補	平均精度	0.40	1.00	0.95	1.00
	平均処理時間 [s]	8.39	9.56	10.77	11.98

の方向数と、方向の割り当て候補の数を変化させた。部分モデルの方向数とは、Algorithm1 のステップ (1) にて用いる  $V_{part}$  の要素数である。また方向の割り当て候補の数とは、Algorithm1 のステップ (3) にて用いる numCordidate である。表の結果は、モデルをランダムに回転させながら各条件それぞれ 20 回の実験を行った平均値となっている。

表 3 では回転対称性の低いモデルの検索では候補 1 の場合でも精度 1.00 に達しており、表 4 では回転対称性の高いモデルの検索でも候補を 5 つに増やすと精度 1.00 を達成できている。このように、提案手法は十分なパラメータを設定すれば、多階層アセンブリモデルの部分検索を正しく行うことができる。ただし表のように、方向の数や候補の数を増やすと処理時間も増えるため、精度と処理時間はトレード・オフの関係にある。

### 5.2 3 階層アセンブリモデル部分検索の比較実験

本節では、既存の 3 次元ラドン変換を用いた 3 階層アセンブリモデルの部分検索手法 [2] でも扱うことのできる、3 階層のアセンブリモデルを用いて部分検索実験を行い、既存手法と提案手法の処理速度を比較する。実験には、52 個の異なる 3 階層アセンブリモデルを用意した。各モデルデータの詳細情報を表 5 に示す。

部分モデル a1~a25 のうち 5 個が完全モデル A に部分一致する部分モデル、部分モデル b1~b25 のうち 5 個が完全モデル B に部分一致する部分モデル、である。個々の実験に用いる部分モ

表 5: 実験に用いた 52 個の 3 階層アセンブリモデルの詳細情報.

	Model	部品数	ノード数	階層の深さ
1	完全モデル A	39	45	3
2~26	部分モデル a1~a25,	36~38	41~44	3
27	完全モデル B	55	61	3
28~52	部分モデル b1~b25	50~54	55~60	3

表 6: 3 階層アセンブリモデル間の部分一致検索結果.(回転対称性の低いモデル)

部分モデルの方向の数		50	100	150	200
提案手法 (1 候補)	平均精度	0.85	1.00	1.00	1.00
	平均処理時間 [s]	4.30	6.11	7.98	9.84
提案手法 (5 候補)	平均精度	0.20	1.00	1.00	1.00
	平均処理時間 [s]	13.18	14.84	16.74	18.54
既存手法	平均精度	1.00	1.00	1.00	1.00
	平均処理時間 [s]	6.68	9.87	12.66	17.79

表 7: 3 階層アセンブリモデル間の部分一致検索結果.(回転対称性の高いモデル)

部分モデルの方向の数		50	100	150	200
提案手法 (1 候補)	平均精度	0.35	0.40	0.70	0.90
	平均処理時間 [s]	14.36	20.50	26.92	33.15
提案手法 (5 候補)	平均精度	0.60	1.00	1.00	1.00
	平均処理時間 [s]	43.25	48.70	55.50	61.31
既存手法	平均精度	1.00	1.00	0.85	1.00
	平均処理時間 [s]	8.28	10.73	14.76	19.96

デルは 5.1 節の実験同様に 5 つとし、5 つのモデルの中から 1 つの正しい部分モデルを検索するという実験を行った。部分モデルの方向数と方向の割り当て候補の数を変化させた場合の処理時間の変化を表 6、表 7 に示す。表の結果は、モデルをランダムに回転させながら各条件それぞれ 20 回の実験を行った平均値となっている。

表 6 の結果について考察を行う。多階層モデルにおける実験と同様、回転対称性の低いモデルについては 1 候補での実験でも精度 1.00 を達成しており、既存手法に並ぶ検索精度を出せている。処理時間を見ると、提案手法は既存手法よりも早い時間で処理が完了しており、また方向の数を増やした場合でもその増加は緩やかである。これは方向の組み合わせおよび平行移動量の決定を先に行うことでその後の計算量を減らしたこと、および各サブアセンブリ同士の比較で体積等を比較し、実際の距離計算を省略したことなど、方向や部品の割り当て探索の枝刈りを行った結果の高速化だと思われる。

次に表 7 の結果について考察を行う。こちらも多階層モデルにおける実験と同様、5 候補での実験で精度 1.00 を達成しており、既存手法に並ぶ検索精度を出せている。しかしながらその処理時間を見ると、既存手法よりも大幅に時間がかかっている。これは、既存手法では個々の部品までは形状の比較を行っていなかったが、これを提案手法では部品単位で形状比較を行うようになったため、部品数に応じて処理時間が増えている。また提案手法の実験では、部品を含めた全ノードにおいて 3 次元ラドン変換結果を保持しており、メモリの不足も問題となっていた。処理の高速化と特徴量の省メモリ化が今後の課題である。

## 6 おわりに

本稿では、3 次元ラドン変換を用いた多階層アセンブリモデルの検索手法を提案した。実験により、提案手法が多階層アセンブリモデルに対して正しく部分一致検索できることが示された。また、回転対称性の低いアセンブリモデルにおいては既存手法よりも速い処理速度で検索を行うことが示された。これにより、これまでの手法では対応できなかったより複雑な構造を持ったアセンブリモデルに対しても部分一致検索を行えるようになった。しかしながら、回転対称性の高いアセンブリモデルの検索においても高い検索精度を出すためには、未だ長い処理時間が必要である。アルゴリズムの高速化と省メモリ化が今後の課題である。

## 文 献

- [1] Kaoru Katayama and Takashi Hirashima. A retrieval method for 3d cad assembly models using 3d radon transform and spherical harmonic transform. *IEICE Transactions on Information and Systems*, Vol. E103.D, No. 5, pp. 992–1001, 2020.
- [2] 平島栄志, 豊田啓介, 片山薫. 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. セッション ID:H8-2 (day2 p.67). DEIM2020, 2020.
- [3] Benjamin Bustos, Daniel A Keim, Dietmar Saupe, Tobias Schreck, and Dejan V Vranić. Feature-based similarity search in 3d object databases. *ACM Computing Surveys (CSUR)*, Vol. 37, No. 4, pp. 345–387, 2005.
- [4] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, Vol. 37, No. 5, pp. 509–530, 2005.
- [5] Johan WH Tangelder and Remco C Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia tools and applications*, Vol. 39, No. 3, pp. 441–471, 2008.
- [6] Bo Li, Yijuan Lu, Chunyuan Li, Afzal Godil, Tobias Schreck, Masaki Aono, Martin Burscher, Qiang Chen, Nihad Karim Chowdhury, Bin Fang, et al. A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries. *Computer Vision and Image Understanding*, Vol. 131, pp. 1–27, 2015.
- [7] Katia Lupinetti, Jean-Philippe Pernot, Marina Monti, and Franca Giannini. Content-based cad assembly model retrieval: Survey and future challenges. *Computer-Aided Design*, Vol. 113, pp. 62–81, 2019.
- [8] Takashi Miura and Satoshi Kanai. 3d shape retrieval considering assembly structure. In *Proceedings of Asian Symposium for Precision Engineering and Nanotechnology*, 2009.
- [9] Xiang Chen, Shuming Gao, Song Guo, and Jing Bai. A flexible assembly retrieval approach for model reuse. *Computer-Aided Design*, Vol. 44, No. 6, pp. 554–574, 2012.
- [10] Abhijit S Deshmukh, Ashis Gopal Banerjee, Satyandra K Gupta, and Ram D Sriram. Content-based assembly search: A step towards assembly reuse. *Computer-aided design*, Vol. 40, No. 2, pp. 244–261, 2008.
- [11] 佐藤渉, 片山薫. 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. セッション ID:I1-4. DEIM2018, 2018.