

3D 仮想物体に対する 分解手法と自動グループ化の提案

大下 勉, 大槻 麻衣, 木村 朝子, 柴田 史久, 田村 秀行

本研究では, 3D 空間において, 多数のパーツからなる複雑な仮想物体をジェスチャによって操作することを考える. 既存のモデリングソフト等では, 複雑な仮想物体に対する効率的な操作のために, グループ化や階層化が用いられている. 我々は, ユーザの意図した箇所を分解する手法を提案し, その詳細とユーザテストについて述べる. その上で我々は, これを各パーツ間の情報に基づいて自動的にグループ化を行う手法を提案する.

A Realization of a Method for Ungrouping and Automatic Grouping for 3D Virtual Objects

Tsutomu Oshita, Mai Otsuki, Asako Kimura,
Fumihisa Shibata, and Hideyuki Tamura

In this research, we consider a gesture-based manipulation method for 3D virtual objects consisting of many parts in mixed reality space. In general modeling software, parts of such a complex 3D object are grouped and manipulated together, but not individually, and hierarchized for efficient operation. Firstly, we propose an ungrouping technique. Using our technique, users can select and detach the parts that they need, from the hierarchical object. Secondly, we also propose the method for grouping parts automatically which is based on the information between each part.

1. はじめに

近年, コンピュータの性能向上に伴い, 3D 仮想物体を実時間で操作することが日常的に行われている. 具体的には, 構造の把握, 学習などの作業において, 車やバイクのような多数のパーツで構成される工業製品や多数の臓器を内包する人体モデルなど, 多数のパーツが複雑に入り組んだ仮想物体が扱われるようになっている.

こうした仮想物体を扱う際, 効率的な作業のために, 既存のモデリングソフトでは, 個々のパーツを別々に扱うのではなく, 複数のパーツをグループ化し, まとめて扱うことが頻繁に見られる. また, 多段階でグループ化し, その階層構造に沿って操作することもある.

我々はこれまでに, 現実と仮想を実時間で融合する複合現実 (Mixed Reality; MR) 空間において, このような多段階でグループ化された仮想物体を分解・観察する手法について検討してきた [1]. 提案システムでは「パーツを直接手でつかみ出したい」という要求を満たし, より実世界に近い感覚での操作を行うために, ジェスチャ操作によって仮想物体の分解, 移動・回転などを可能とした.

これを運用する中で, 以下のような問題が明らかになった.

- (1) オブジェクト全体から, 特定の階層のグループのみを取り出し, 操作したい場合でも, 階層順に分解する必要がある
- (2) 多数のパーツを設計者がグループ化, 階層化するのはわずらわしい

これらを踏まえ, 本研究ではユーザが意図した階層で分解可能な入力手法を提案し, 各パーツの情報に基づいた自動的なグループ化・階層化の実現方法を提案することにより, 上記 2 点の問題の解決をめざす.

2. 関連研究

仮想現実 (Virtual Reality; VR) 空間の構成において, 階層構造を持つツリーで記述する (シーングラフを用いる) ことはしばしば行われる. このシーングラフに基づいて仮想物体を選択, 分解する手法としては別途シーングラフ用のウインドウ上で変更することが一般的である. しかし我々は MR でそのようなウインドウを別途出すことは好ましくないと考え, 直接オブジェクトに触れて操作することを考える.

階層構造で管理された樹木モデルの部分形状を両手操作によって動的に選択可能な手法 [2] が提案されている. これは樹木のある一部分だけを選択, 分解し, 枝ぶりを編集することが可能である. しかし対象は樹木モデルに限定されており, 本研究で目的とするような複雑なオブジェクトは想定されていない.

デスクトップ上で多数のオブジェクトに対しグループ化を行い, 配置支援を行っている研究がいくつか見られる. Wigdor らの Rock & Rails [3] では, オブジェクト同士を手動でつなぎ, グループ化を行う. Watanabe らの Bubble cluster [4] では, 近づけたオブジェクトを自動的にグループ化する手法を提案している.

VR においても、複数のオブジェクトを自動的にグループ化して配置作業を支援する研究がいくつかある。Stuerzlinger らは、3D 空間において近傍にあるオブジェクト群を自動的にグループ化する手法 [5] を提案している。また、Oh らの手法 [6] では重力に基づいた関係 (gravitational relationship) による階層化を行っている。

本研究ではこれらを発展させ、近傍や重力関係ではなく、より一般的な「接続関係」によるグループ化、階層化を行う。具体的には、複数のパーツからなるオブジェクトにおいて、各パーツの接続関係をグラフに見立て、グラフクラスタリングによってグループを生成する手法を考える。

一般的にグラフクラスタリングは、つながりをグラフで記述し、複雑につながった大量の情報を分類・整理し、有益な知識を発見する手がかりとして用いられている。例えば、P2P などのファイル共有アプリケーションの性能向上のため、ネットワーク上で接続されたコンピュータをクラスタリングする研究 [7]、文書集合をグラフ、文書同士の関連をエッジとし、文書を話題ごとに分類し、話題間の関係や話題の中心となっている文書を特定する研究 [8]、画像認識の分野で、複数の車両の軌跡群をクラスタリングし、車両の追跡を実現する研究 [9] などがある。

グラフクラスタリングを用いた研究の多くは、最終的なクラスタリング結果のみに着目しているが、本研究ではその途中経過にも着目し、これに基づいて自動的に多段階のグループ化、ツリー (階層構造) を生成する。階層構造を生成するのに適しており、クラスタリング手法としては高速な Newman 法 [10] が知られている。本研究ではこれをもとにパーツを自動的にグループ化・階層化を行う手法を提案する。

3. 基本システム

3.1 ツリー構造による多段階のグループの管理

本研究では図 1 に示すような多段階のグループノードとその終端に接続された 1 段の実体ノード (1 つの実体ノードが 1 つのパーツに対応する) で構成されるツリー (n 分木) によって各パーツを管理する。パーツの選択、分解などはすべてこのツリーに基づいて行われる。

3.2 パーツに対する操作

【選択】

右手、左手いずれを用いた場合でも、パーツに触れると、そのパーツが含まれるグループに属している全てのパーツが選択される。ツリー上では、触れたパーツの実体ノードと同一のツリーに属する実体ノードを全て「選択」状態にする (図 2)。

【観察 (移動・回転)】

- (1) 左手の親指と人差し指でパーツをつまむと、選択状態になっているパーツを全て「観察」状態にする。

- (2) 観察状態になったパーツは、左手をひねる、移動させることで、その位置姿勢にあわせた移動・回転を行える。

【分解】

- (1) 右手でパーツをつまむと、そのパーツの実体ノードからルートノードに向かって探索を行い、最もルートノードに近いグループノード以下の実体ノードを全て「分解」状態にする (図 3a)。この際、最もルートノードに近いノードが実体ノードの場合は、その実体ノードのみを「分解」状態にする。
- (2) 分解状態のパーツは元の位置から一定距離以上引き離すことで「分解完了」となる。システムでは (1) で探索されたグループノードとルートノード間の接続を切断し、ツリーを分割する (図 3b)。

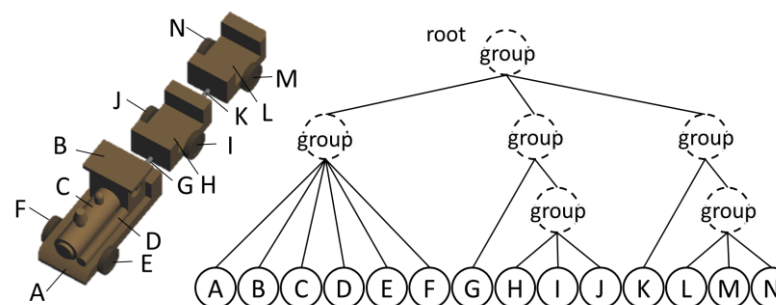


図 1 モデルとツリー

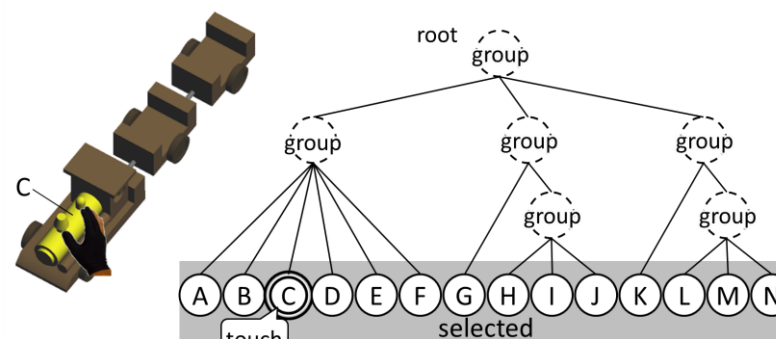


図 2 選択操作

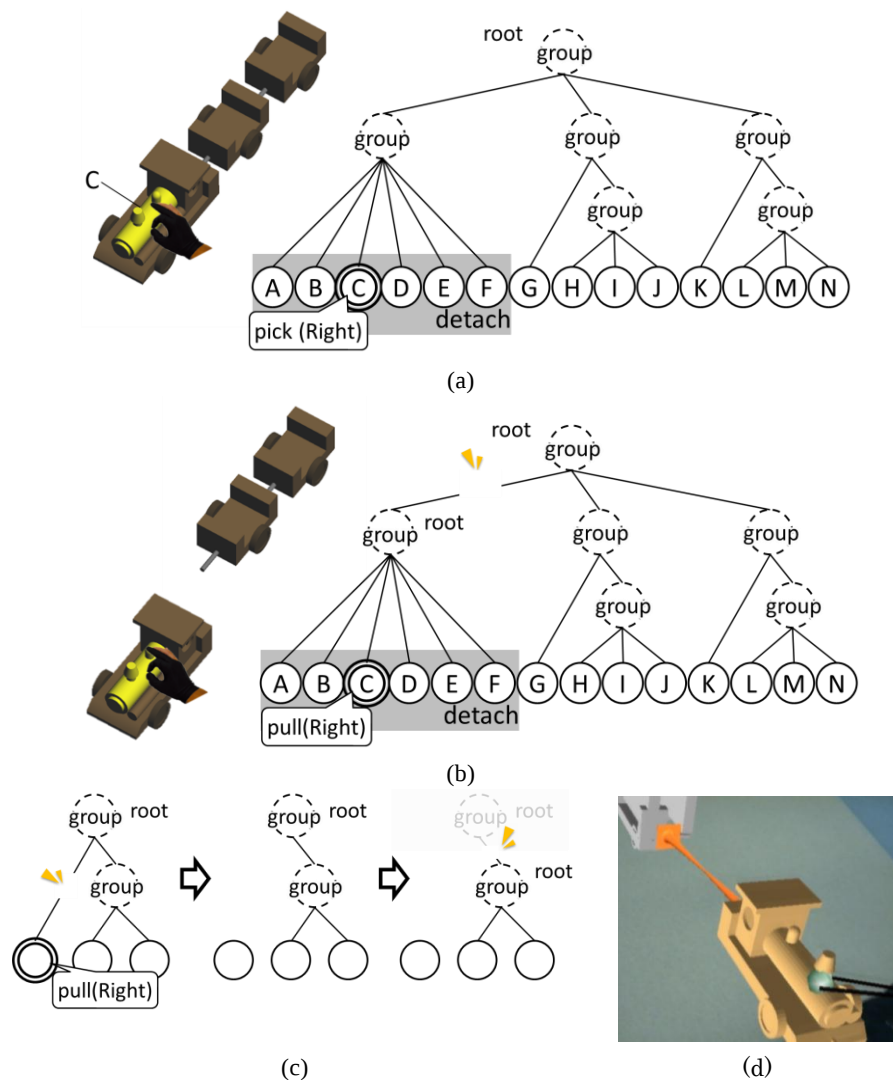


図 3 分解操作

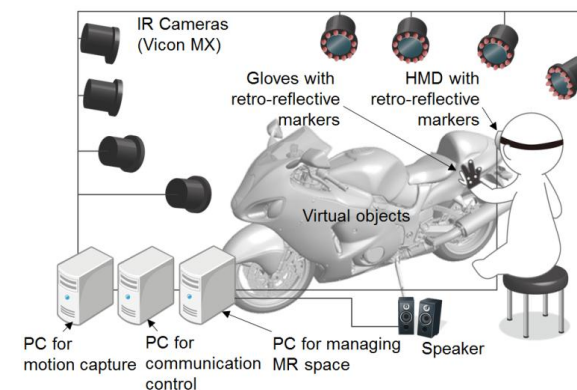


図 4 システム構成

(3) 分解が完了する度に各ツリーのルートノードを更新する．また，ルートノードから実体ノードに向かって探索を行い，グループノード，もしくは実体ノードが 1 つしかない階層があれば，自動的にその接続を切断し，新しいツリーを作成する (図 3c)．

なお，上記 (2) においては，誤操作による分解を避けるため，前研究 [1] のゴム状接着剤メタファに基づいた視聴覚フィードバックを提示する (図 3d)．

3.3 システム構成

設計した手法を実現するために，両眼立体視可能なビデオシースルー方式の MR システムとして実現した (図 4)．MR 空間の管理・構築には Windows XP OS, Intel Core2 Duo E4300 CPU を搭載する PC を用いた．MR 空間の映像提示にはカメラ内蔵型の HMD (Head Mounted Display) である Canon VH-2002 を用いる．HMD の位置姿勢取得および各種ジェスチャの認識にはモーションキャプチャシステム (ViconPeaks 社製 MX カメラシステム) を使用する．これは，カメラ 8 台とカメラ制御用 PC，通信制御用 PC 各 1 台から構成される．利用者は，親指，人差し指，手の甲に再帰性反射マーカを貼付した手袋状デバイスを装着し，また，HMD にも再帰性反射マーカを貼付することで 3 次元位置姿勢を検出可能である．モーションキャプチャシステムにより得られたこれらの情報は通信制御用 PC を介して MR 空間管理用 PC に送られる．

MR 空間の生成・提示には，HMD に内蔵された左目用，右目用の各カメラからビデオキャプチャカード (ViewCast Osprey 440) を介して実風景を PC へ取り込み，HMD の位置姿勢に合わせた CG を重畳描画した後，グラフィックカード (NVIDIA Quadro FX 1700) からそれぞれ HMD の左目，右目用の各ディスプレイへ出力することでユーザに提示する．

開発言語には C++ を、グラフィックス API には OpenGL および GLUT (OpenGL Utility Toolkit) を使用している。

4. ユーザの入力に基づく分解手法

4.1 提案手法

前章のシステムにおいて、ユーザが入力パラメータを変化させることで、ユーザが任意の階層で分解を行うことが可能な手法を提案する。システムではつまんでいるパーツの実体ノードからルートノードに向かって探索を行い、その実体ノード、あるいは任意の階層のグループノードのいずれかに着目することで、図 5 (a) ~ (c) のように、分解される階層を切り替える。これを行う手法として以下の 3 種類を提案する。

(a) クリック回数を用いた分解

「クリック」を「同じパーツに対して連続してつまむジェスチャを行うこと (図 6)」とし、その回数に応じて分解される階層を順に切り替える。分解される階層が 1 つの実体ノードになった状態 (図 5c) で、さらにクリックすると初期状態 (図 5a) に戻る。また、分解操作を行っているパーツから右手を外した場合にも初期状態に戻る。

(b) ユーザからパーツまでの距離を用いた分解

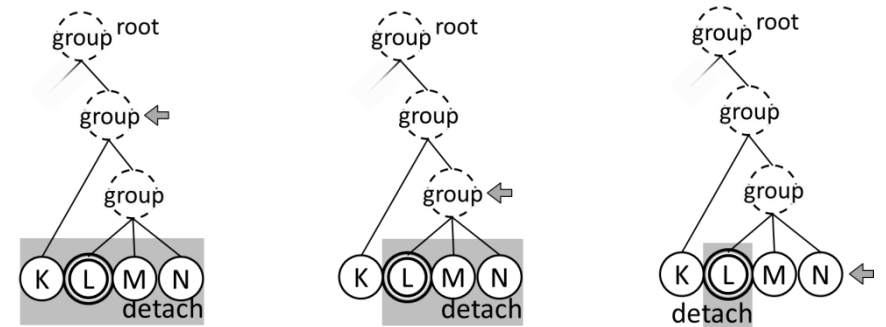
対象の詳細を見たい場合は近づき、広い範囲を見渡したい場合は離れる、という実世界の動作を基にする。具体的には、図 7 に示すように、ユーザの頭部位置から選択しているパーツまでの距離に応じて、分解される階層を順に切り替える。

ここで、現在つまんでいるパーツの実体ノードの階層を 0、ルートノードの階層を m_{level} とする。ユーザの頭部位置から選択しているパーツまでの距離を d とし、 m_{level} 階層目までを切り替え可能な距離の範囲として最大値 d_{max} と最小値 d_{min} を設定する。分解する階層を 1 段階切り替えるのに必要な移動距離 d_{move} は以下の式で表される。

$$d_{move} = \frac{d_{max} - d_{min}}{n_{level}} \quad (1)$$

実際に分解する実体ノードからの階層数 n_{detach} の決定には以下の式を用いる。 n_{detach} も自然数であり、つまんでいるパーツの実体ノードを 0 とし、ルートノードに近づくほど数が大きくなる。

$$\begin{aligned} & \begin{aligned} & \text{if } (d \geq d_{max}) \\ & \text{else if } (d_{min} \geq d) \end{aligned} & \begin{aligned} & n_{detach} = m_{level} - 1 \\ & n_{detach} = 0 \end{aligned} \\ & \text{else} & n_{detach} = \left\lceil \frac{d - d_{min}}{d_{move}} \right\rceil \end{aligned} \quad (2)$$



(a) 2 階層目で分解 (b) 1 階層目で分解 (c) 0 階層目で分解

図 5 分解される階層の切り替え (矢印は着目している階層を示す)



図 6 1 回のクリック

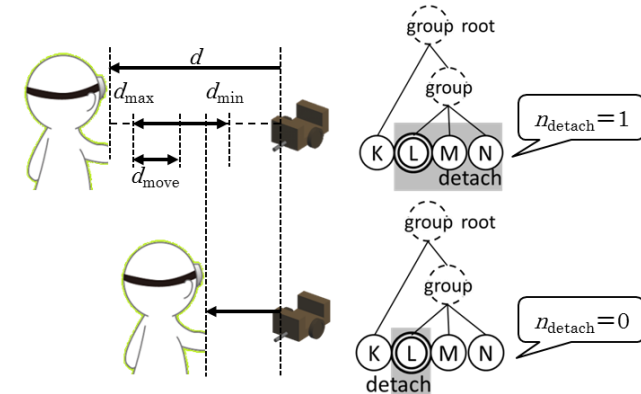


図 7 ユーザとパーツとの距離を用いた階層の変更 ($m_{level} = 2$ の場合)

(c) 両手を用いた分解

必要な部分を片方の手で選択し、不要な部分をもう片方の手で押さえる、という実物体を扱うような直観的な操作を実現する。以降、必要な部分を選択する手を右手、もう片方を左手とした場合を例にとり、その処理手順を説明する (図 8)。

(1) 右手でつまんだパーツの実体ノードから、ルートノードに向かって探索を行う。現在着目しているノード (図 8a 中の **current node**) の親が左手で触れたパーツの実体ノードを含むグループノードと一致した場合 (同 **common parent node**) に探索を終了する (図 8a)。

(2) 現在着目しているグループノード以下の実体ノードを全て分解状態にする (図 8b)。着目しているノードが実体ノードの場合はその実体ノードのみを分解状態にする。

4.2 実装

図 9 に分解操作中の様子を示す。分解状態のパーツを示す視覚フィードバックとして該当するパーツのみポリゴンの稜線を表示している。図 9a はクリック回数を用いた分解の様子であり、1 回クリックをすることで、1 段階下の階層が分解可能になる。図 9b はユーザからパーツまでの距離を用いた分解操作の様子を示している。ユーザが接近する事で、より細かいパーツの分解が可能になっている。図 9c は両手を用いた分解の様子である。右手でつまんだ場合に分解状態となるパーツでも、左手で触れることで除外することができる。

4.3 運用

このシステムを数名に椅子に着席させた状態で体験させ、コメントの収集を行った。運用には 5 段階の階層化がされている単純な形状のモデル (図 10a) と、10 段階の階層化がされた複雑な形状のモデル (図 10b) の 2 種類を用いた。体験者には 3 種類の分解手法を体験させている間、自由に発言させた。

4.4 考察

コメントの内容と得られた知見を下記に示す。

(a) クリック回数を用いた分解

段階的に値を変える確実な入力手法として提案したが、「モデルが複雑でパーツが細くなると、同じパーツを続けてクリックすることが難しい」というコメントが得られた。間違えて別のパーツをクリックしてしまったり、選択がモデルから外れてしまった場合は、クリック回数が初期値に戻ってしまうため、操作が煩雑になる。解決策としてはパーツごとにクリック回数を記憶しておくことが挙げられる。

(b) ユーザからパーツまでの距離を用いた分解

クリック回数を用いる手法と比べると、目的の階層に素早く辿り着くことができるが、モデルの階層が多くなるほど、 d_{move} が小さくなるため、意図した階層の選択が難しくなる、というコメントが得られた。具体的には d_{move} が 0.5 cm 程度になると意図した階層を選択することが困難であるという知見が得られた。

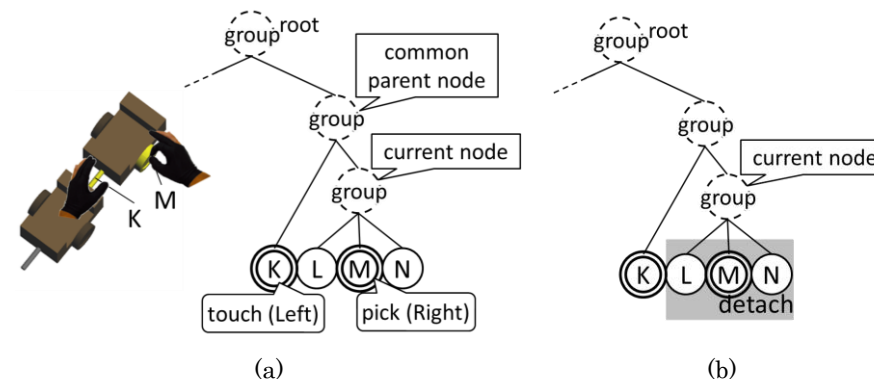
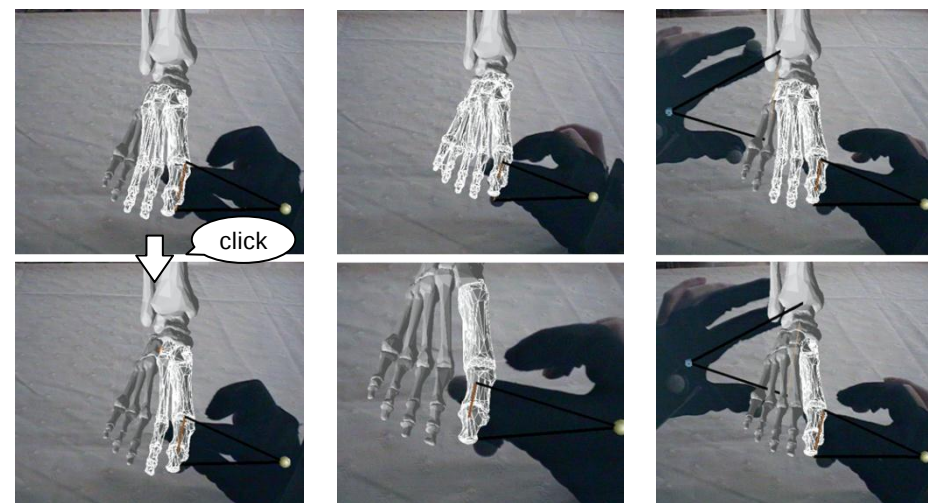


図 8 両手を用いた分解



(a) クリックを用いた分解

(b) 距離を用いた分解

(c) 両手による分解

図 9 分解の様子

これに関しては、 d_{max} と d_{min} を変更する事で解決することができるが、 $|d_{max}-d_{min}|$ が大きくなりすぎると、ユーザが頭部を動かさなければならない距離が長くなり、疲労につながるという問題が出る。研究室内で $|d_{max}-d_{min}|$ を様々に変更したところ、着席した状態では「 $|d_{max}-d_{min}|$ が 20 cm 程度までであれば無理なく動かせる」というコ

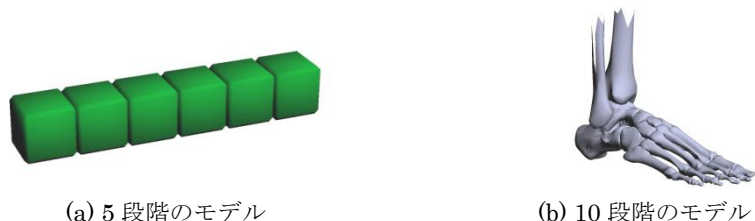


図 10 運用に使用したモデル

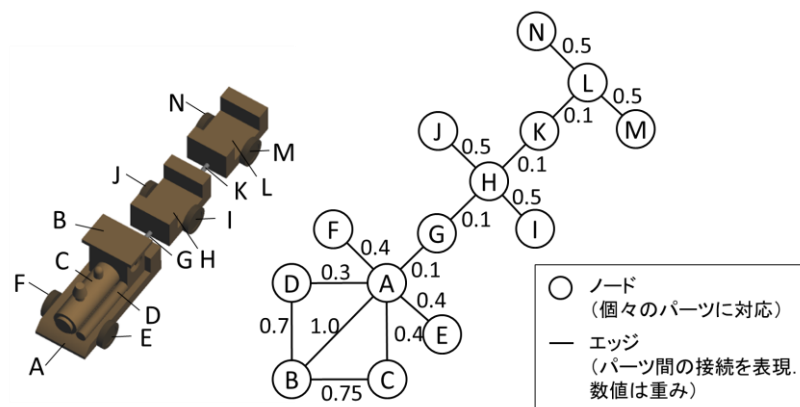


図 11 モデルとグラフ

メントが得られた。

(c) 両手を用いた分解

現実世界と近いイメージで分解が行えるというコメントが得られたが、実際のグループ化・階層化の状態と、ユーザがモデルから想像したグループ化・階層化とが異なる場合、他の分解手法と比べギャップが大きい、といった問題がある。

5. 自動グループ化・階層化

5.1 パーツの情報を用いた自動グループ化・階層化

3章で述べたシステムにおいて、グループ化・階層化に用いるツリーを自動的に生成する手法について考える。ここでは、パーツ同士の接触関係に基づいて、パーツをノード、接続関係をエッジとするグラフを用意する。エッジの重みは、例えばモデルの設計者が「この部分の接続を他の箇所より強固にしたい」といった場合にはそのパ

ラメータを考慮する、もしくは、ユーザが利用するパラメータを選択できるものとするなど、柔軟な設計を考えている。重みとして用いるパラメータとしてはパーツの素材や、パーツ同士の関連性など多数挙げられるが、本研究ではシンプルなパラメータとして、パーツ間の接触面積を用いる。例として、汽車のモデルを用いた場合のグラフを図 11 に示す。

Newman 法は、同一クラスタ内ではエッジを多く、異なるクラスタ間ではエッジを少なくするようにノードをクラスタリングしていくアルゴリズムである[11]。Newman 法では、クラスタリングの精度を Q で定義し、この値が大きくなるようにクラスタリングを行う。

$$Q = \sum_i (e_{ii} - a_i^2) \quad (3)$$

ただし、

$$e_{ij} = \frac{\text{クラスタ}i\text{とクラスタ}j\text{間のエッジの重みの和}}{\text{全エッジの重みの和}} \quad (4)$$

$$a_i = \sum_k e_{ik}$$

である。

Newman 法では、はじめにノードと同じ数だけクラスタを作成する。そしてグラフ内で Q の値が最も大きく増加する 2 つのクラスタを統合する。クラスタ i とクラスタ j 間のクラスタを統合したときの ΔQ は以下の式で求めることができる。

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j) \quad (5)$$

以上の計算式を踏まえ、グラフの中で最も高い ΔQ が得られるクラスタの組み合わせを探しだし、ツリーを形成していく。 Q の最大値が得られた時点で、残されているすべてのクラスタを多分木としてつなぐ。この手法を用い、図 11 の汽車モデルから作成したツリーを図 12 に示す。

しかしこの手法では、図 11 のパーツ E、パーツ F などは同等の情報を持っているにもかかわらず、作成されたツリーでは階層が異なっており、分解される順番に差が発生する。この問題を解決するために、本研究で用いるクラスタリング手法は、最も高い ΔQ が得られる組み合わせが複数存在し、それらの組み合わせの一方が他の組み合わせの一方と一致している場合、それらすべてのクラスタを同時に結合し、多分木のツリーを生成する。図 13 に提案手法で作成したツリーを示す。条件が同じパーツ同士が同じ階層にあることがわかる。

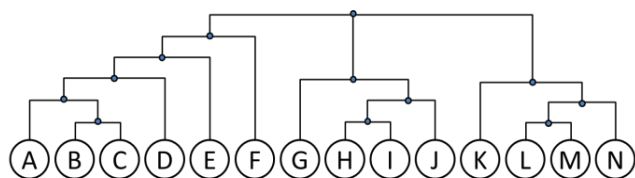


図 12 Newman 法によるツリー

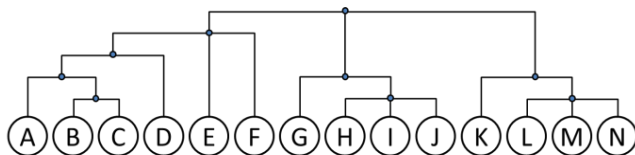


図 13 提案手法によるツリー

図 14 は、多数のパーツからなる複雑なモデルの各パーツをナンバリングし、グラフ化したものである。これを提案手法でクラスタリングし、作成したツリーを図 15 に示す。

5.2 グループ化・階層化のタイミング

自動グループ化・階層化を行うタイミングについて考える。分解作業を行うと、モデルのパーツ同士の接触関係とツリーの間に矛盾が生じる場合がある。例えば図 16 に示されるように、パーツ群 G～J のグループを分解した場合、実際のモデルでは、パーツ群 A～F とパーツ群 K～N は接触していないが、ツリー上では同じグループに属していることになっている。

このような矛盾を防ぐために、分解操作が終了する度に、変形したグラフ（図 17）に対して再度クラスタリングを行う。実装の結果を図 18 に示す。モデルとツリーに矛盾がないことがわかる。

5.3 考察

図 14 の足骨モデルは、パーツ数が 30、グラフのエッジ本数は 38 本となっており、実行時のツリー作成時の fps は 55 前後とストレスなく操作が可能である。しかし、一般的なグラフクラスタリングがオフラインで行われるということを考慮すると、本研究のように、分解する度にリアルタイムでクラスタリングを行うことは、必ずしもよいとは言えない。パーツ数やエッジ本数によっては処理に時間がかかってしまうという問題が考えられるためである。

今後はより計算速度の速いクラスタリング手法を用いることや、ツリーを生成するのではなく、グラフのみを用いて、エッジの重みに応じて自動グループ化・階層化を

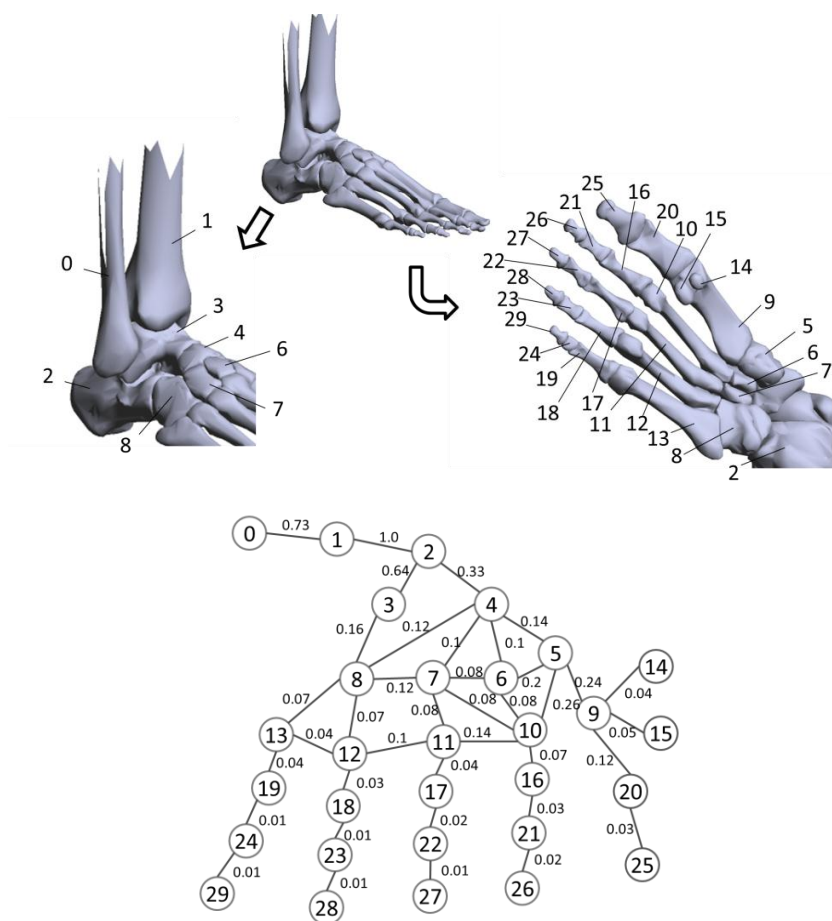


図 14 複雑なモデルと、接続関係に基づいて作成したグラフ

行うことが考えられる。

6. むすび

本研究は MR 空間において、(1) グループ化・階層化された多数のパーツからなる

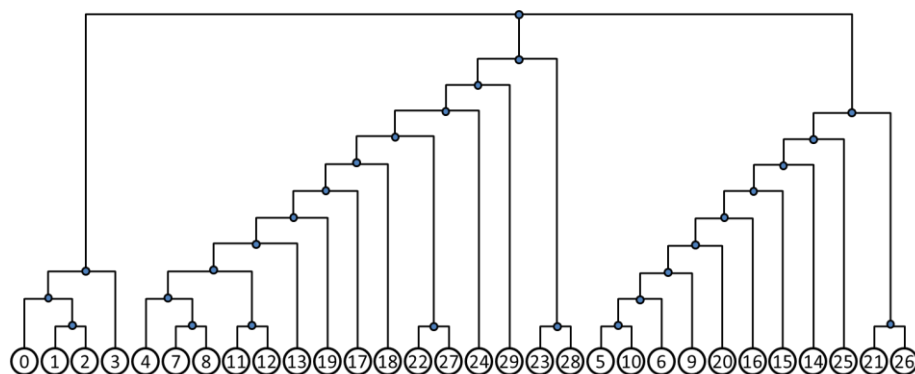


図 15 複雑なモデルのツリー生成結果

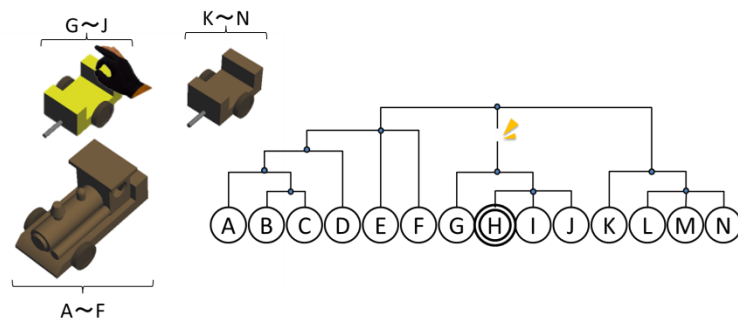


図 16 モデルとツリーが矛盾する場合

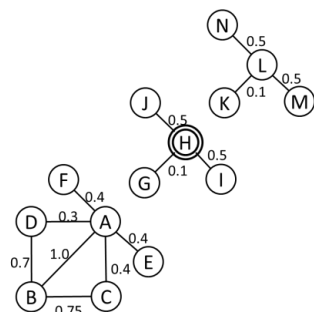


図 17 分解後のグラフ

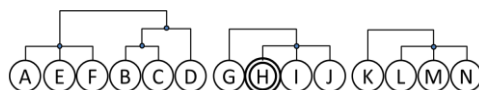


図 18 グラフごとのツリー生成結果

複雑なモデルを、ユーザが意図した階層で分解可能な入力手法と、(2) 各パーツの情報に基づいた自動的なグループ化・階層化の実現方法を提案した。

(1) では 3 種類の入力手法を提案し、その運用を行い、それぞれの手法について考察を行った。(2) では、パーツ同士の接続関係をグラフに見立て、Newman 法をもとに、本研究の用途に沿ったクラスタリング手法の提案を行い、その実装結果において考察を行った。

今後の発展として、(1) については、定量的な評価とユーザフィードバックをもとにした手法の改良が挙げられる。(2) については、ユーザスタディを行うことや他の自動グループ化・階層化手法の検討を予定している。

参考文献

- 1) 大槻麻衣, 大下勉, 木村朝子, 柴田史久, 田村秀行: “3D 空間における仮想オブジェクトの分解・観察に適した操作法の提案と実装”, 日本バーチャリアリティ学会論文誌, Vol. 16, No. 2, pp. 227 - 237, 2011.
- 2) 大西克彦, 北村喜文, 正城敏博, 岸野文郎: “両手操作による樹木モデルの部分形状選択の効率化”, ヒューマンインタフェース学会論文誌, Vol. 4, No. 1, pp. 9 - 15, 2002.
- 3) Wigdor, D., Benko, H., Pella, J., Lombardo, J., and Williams, S: “Rock & Rails: Extending multi-touch interactions with shape gestures to enable precise spatial manipulations,” Proc. of CHI '11, pp. 1581 - 1590, 2011.
- 4) Watanabe, N., Washida, M., and Igarashi, T: “Bubble clusters: An interface for manipulating spatial aggregation of graphical objects,” Proc. UIST '07, pp. 173 - 182, 2007.
- 5) W. Stuerzlinger, and G. Smith: “Efficient manipulation of object groups in virtual environments,” Proc. VR '02, pp. 251 - 258, 2002.
- 6) J. Y. Oh, W. Stuerzlinger, and D. Dadgari: “Group selection techniques for efficient 3D modeling,” Proc. 3DUI '06, pp. 95 - 102, 2006.
- 7) 上田達也, 安倍広多, 石橋勇人, 松浦敏雄: “P2P によるインターネットノードの階層的クラスタリング手法の評価”, マルチメディア, 分散, 協調とモバイル (DICOMO 2005) シンポジウム論文集, Vol. 2005, No. 6, pp. 77 - 80, 2005.
- 8) 戸田浩之, 北川博之, 藤村 考, 片岡 良治, 奥雅博: “グラフ分析を利用した文書集合からの話題構造マイニング”, 電子情報通信学会論文誌, Vol. J90-D, No. 2, pp. 292 - 310, 2007.
- 9) 安倍満, 小沢慎治: “撮影環境にロバストな交通流計測アルゴリズム”, 電子情報通信学会論文誌, Vol. J88-A, No. 8, pp. 983 - 993, 2005.
- 10) M. E. J. Newman: “Fast algorithm for detecting community structure in networks,” Phys. Rev. E 69, 066133, 2004.
- 11) 田淵史郎, 鍛冶伸裕, 吉永直樹, 喜連川優: “大規模コーパスからの語義のマイニング”, 日本データベース学会論文誌, Vol. 8, No. 1, pp. 77 - 82, 2009.