

3次元ポリゴンモデルの展開図作成

三谷純 鈴木宏正[†] 木村文彦[†]

東京大学大学院工学系研究科 情報工学専攻

[†] 東京大学大学院工学系研究科 精密機械工学専攻

E-mail: {mitani, suzuki, kimura}@cim.pe.u-tokyo.ac.jp

本論文では、計算機上に存在するポリゴンモデルデータに対し、その3次元モデルを紙で効率良く組み立てることを目的とした展開図作成手法を提案する。具体的には、展開図を切り取る際に切断の必要がある稜線の総延長、展開図に外接する長方形の面積、および展開図の部品数を組み立て易さの評価に用いる。本手法は、モデルを連続した1枚の展開図に展開することを目的とするのではなく、組み立て易さを考慮した展開図を高速に作成することを目的としている。本論文では面と稜線の間接関係を表すグラフに対し、深さ優先探索と欲張りアルゴリズムを用いた2種類のアルゴリズムを提案し、4通りのコスト設定方法により、作成される展開図の評価を行った。本アルゴリズムを用いた展開図の作成と、組み立ての支援を行うアプリケーションを作成し、生成された展開図をもとに紙模型を作成したので、具体例とともに報告する。

A Method for Generating Developments of Triangular Polyhedral Models

JUN MITANI HIROMASA SUZUKI[†] FUMIHIKO KIMURA[†]

Dept. Information Engineering, University of Tokyo.

[†] Dept. Precision Engineering, University of Tokyo.

E-mail: {mitani, suzuki, kimura}@cim.pe.u-tokyo.ac.jp

This paper describes a method for generating developments of polyhedral models, which are easily constructed with paper sheet. The easiness is evaluated in terms of the total length of edges which must be cut, the area of a rectangle circumscribing the development and the number of parts of development. Our goal is not for generating the optimal development but for generating an acceptable development quickly. Basically, generating a development corresponds to finding a spanning tree of *face-edge graph* of the polyhedral model. We propose two algorithms for traversing the *face-edge graph*, and four methods for setting costs representing the easiness of manual construction. We implemented an application which generates developments using this method and supports a user to construct paper models.

1 はじめに

計算機内に存在する3次元形状データに対して、実際に手で触れることのできる実体模型を作り、現実世界で知覚したいという要求は強い。一方、古くから紙工作の一種として、立体の展開図から紙模型を組み立てることが行われている。

本論文では、計算機上に存在するポリゴンモデルを、紙で効率良く組み立てることを目的とした展開図作成手法を提案する。

展開図の作成は、人間が展開後の形状を想像しながら試行錯誤を繰り返して行うことが多いが[1]、この作業には時間と労力が非常にかかる。衣服の裁断図を計算機を用いて生成する研究[2]もあるが、これは同一平面上

に乘らない四角形パッチから構成される展開図を出力するため、柔軟性のある布や生地には適するが、紙には応用できない。また、衣服の形状のみを対象としている。多面体モデルの展開図を作成する研究もなされているが[3]、これは多面体を1枚の展開図に展開することを目的とした全探索を行っているため、本研究で目的とするような面数の多いモデルでは探索の数が爆発的に増加してしまい、実時間で展開図を求めることが難しい。

本手法では、図1に示すような、面数が数100前後の三角形メッシュモデルの展開図作成を対象とする。本手法は以下の利点をもつ。

- 計算機によって展開図の作成が完全に自動化される。
- 曲面を三角形メッシュで近似することで、任意の形状を扱える。このため、一般に人間が展開図を考案するのが困難である自由形状モデルを扱うことができる。
- 実際に組み立て易さを考慮した評価を行いながら展開図を作成するため、組み立てに適した展開図を作成できる。なお、本手法では、展開図の切り取りの際に切断が必要な稜線の総延長、展開図に外接する長方形の面積、および展開図の部品数を組み立て易さの評価に用いた。
- 必ずしも1枚の展開図に展開することを目的とせず、局所的なコストの設定と評価による、インクリメンタルな展開図作成アルゴリズムを用いる。そのため、面の数が多いモデルでも実時間で評価の良い展開図を作成することができる。

本論文では、展開図生成のためのアルゴリズムを示し、さらに展開図の作成と、組み立ての支援を行うアプリケーションを作成したので、具体例と共に報告する。

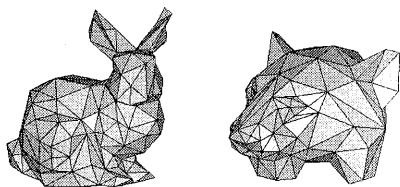


図1: 三角形メッシュモデル

2 展開図と面接続グラフ

2.1 展開図の定義

曲面(多角形を含む)の表面を剛体変換によって一平面に移した形を、その曲面の展開図という。剛体変換では曲面のガウス曲率は変わらず、平面のガウス曲率はゼロであるから、平面上に剛体変換できる曲面、すなわち

展開可能な曲面はガウス曲率がゼロである曲面に限られる。これは柱面、錘面、および接線曲面(空間曲線への接線群で構成される曲面)の3種のみである[4]。

展開可能でない曲面に関しては、近似展開を行うことになる。一般に3角形メッシュによって多くの曲面を多角形近似することができるため、本研究では曲面の展開は扱わずに、3角形メッシュから成る多面体モデルのみを対象とする。

多面体モデルの中には、展開した時に自己干渉を起し、連結した1つの展開図に展開できないものがある。このような場合、展開図を複数に分割する必要がある。1つの多面体モデルが K 個の部分展開図に展開される時、本論文では「展開図の部品数は K である」と言うことにする。なお、多面体は連結なものとす。

2.2 3角形メッシュモデルの展開図作成方法

多面体の展開図は、ある面を始点として、それに接続する面を、順次同一平面上に展開していくことで作成される。多面体が3角形メッシュモデルである場合、新しい面を展開するには、展開済みの面と接続する稜線がすでにこの平面上にあるため、残り1つの頂点の位置がわかればよい。面を構成する稜線はつねに3本であり、それぞれの長さが既知であるため、新しい頂点の座標値を求めることができる。

また、展開された面が互いに干渉してはならないため、新しく展開される面を構成する辺が、展開済みの面を構成する辺と展開後に交わるか否かを判定する。交わる辺が存在する場合は、面の干渉が起るため、その面は展開できない。どの辺とも交わらない場合は展開可能である。

2.3 面グラフ

多面体の面を節点、稜線を枝とした、面の接続を表すグラフを面グラフと呼ぶことにする。図2(左)に示す6面体の面グラフは図2(右)のように表すことができる。球と同位相の3角形メッシュモデルの場合、節点には必ず3本の枝が接続する。また、多面体が連結であることを仮定しているので、すべての節点が異なる節点と連結であり、面グラフは連結グラフとなる。

部品数が1である展開図(図3(左))の面グラフは、すべての面が閉路を持たずに稜線によってつながっているため、図3(右)の様に、展開前の面グラフの極大木となる。

展開前のモデルに対応する面グラフの極大木を求め、それを元に展開図を作成することが可能であるが、展開平面上で面が干渉する場合があるため、必ずしも部品数が1の展開図に展開できるとは限らない。

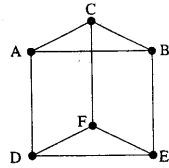
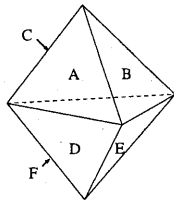


図2: 多面体とそれに対応する面グラフ

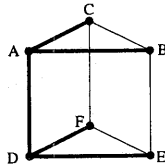
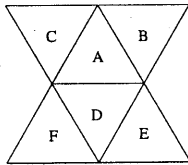


図3: 展開図とそれに対応する面グラフ

2.4 極大木の数

一般に、面グラフから生成できる極大木は複数存在する。これらの極大木をすべて調べることで、多面体が一平面上に展開可能であるかを判定できるとともに、最も評価のよい展開図を求めることが可能である。しかし、モデルの面数が増えるに従って極大木の数は爆発的に増加するため、本研究で対象とするような面数の多いモデルに対しては現実的ではない。本研究では、全ての極大木を調べずに、局所的なコスト評価で、なるべく全体的に評価のよい展開図を作成することを考える。その際に用いるアルゴリズムを次章で述べる。

3 展開図作成アルゴリズム

3.1 面グラフ探索アルゴリズム

展開済の面から、次に展開する面を決定し、インクリメンタルに展開図を作成して行くためのアルゴリズムとして、次の2つを考案した。

3.1.1 コスト評価を用いた深さ優先探索

適当な面を開始点とし、深さ優先探索で面グラフを巡り、順次、面を展開しながら展開図を作成していく。すべての面を巡ったところで展開図の作成が完了する。

1つの面に隣接する面は3つあり、1つはすでに展開済みの面であるから、次に展開する候補の面は最大で2つあることになる(開始点では3つある)。ここで、展開する候補の面とは、展開したときに展開済の面と干渉しない面のことである。この様子を図4(左)に示す。ここで、色塗りの三角形が探索の始点であり、斜線の三角形が1つ前に展開された面を表す。次に展開される候補の三角形が破線で記されている。各面の重心を結ぶ線分が面

グラフの稜線に対応する。

展開する候補の面が2つある場合はコストの小さい方を、1つの場合はそれ自身を展開し、0である場合はバックトラックする。探索をさかのぼっても展開可能な面が存在しない場合は、まだ展開されていない面の中から新しく開始点となる面を選択し、展開図の作成を続ける。この場合、展開図の部品数が1増えたことになる。

2つの候補があるときにコストの小さい方を選ぶことで、コスト評価を行わない単純な深さ優先探索の展開よりもよい結果を得ることができる。計算量は、面の数を n とすると、 $O(n)$ にすぎない。コストの設定については3.3節で述べる。

3.1.2 コスト評価を用いた「欲張り」アルゴリズム

展開済みの面に隣接して、次に展開することが可能な面すべてに対してコスト計算を行い、最もコストの小さい面を次に展開することで展開図を作成していく。この様子を図4(右)に示す。図の見方は前節で述べたとおりである。

重みつきグラフに対して、その時点でコストが最小な枝を1本ずつ追加していくことでコスト最小の極大木を作成するプリムのアルゴリズム [5] と同等の手法で枝を延ばしていく。よって、一見するとコストが最小の最適解が見つかるように思えるが、展開平面上での干渉によって面が展開できない場合、グラフの枝が突然消える(動的にグラフが変化する)ため、最適解を見つけることはできない。

しかし、深さ優先探索の手法よりも、次に展開可能な候補の数が圧倒的に多いため、次節に示すように、よりよい解を得られる。計算量は $O(n^2)$ である。

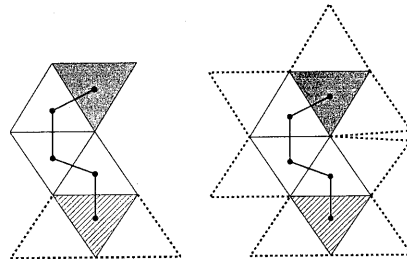


図4: 深さ優先探索(左)と欲張りアルゴリズム(右)

3.2 組み立て易さを表す評価値

本研究では、展開図の組み立てやすさを評価するために、工作の際に切断を行う稜線の総延長(L)、展開図に外接する長方形の面積(S)、部品数(N)、の値を用い、それぞれ値が小さいほど組み立てやすい展開図であるとしました。これらの値を用いる理由は以下の通りである。

- 切断を行う稜線の総延長(L)

切断を行う稜線の総延長が短いほど、展開図の切り取りの労力と、組み立て作業の際の貼りあわせの労力を共に減らすことができる。

● 展開図に外接する長方形の面積 (S)

不必要に展開図が広がらず、小さくまとまっているほど工作用紙の無駄を減らすことができる。

● 部品数 (N)

部品数が少ないほど組み立てやすい。部品数 1 の展開図に展開できることが理想である。

3.3 コストの設定

本手法では局所的にコストの小さい面を選択し、順次展開を進めていくことで、すべての展開図を調べなくとも、ある程度評価の良い展開図を作成することをねらう。そのためには、次に展開する候補の面に対して適切なコストを設定する必要がある。以下に本手法で用いた局所的なコスト設定方法を示す。

● L を小さくするためのコスト (C_L) の設定

長い辺はなるべく面グラフの枝 (つまり切断しない稜線) にした方がよいので、次に展開する候補の面が長い辺によって展開済みの面と接続していた方がよい。そこで、候補の面と展開済みの面とを接続する稜線の長さの逆数をそのコスト C_L とする。

● S を小さくするためのコスト (C_S) の設定

候補の面を展開することによる面積の増加率をコストとすると、一方向に展開図が延びてしまうため、候補の面を展開した後における、展開図に外接する四角形の縦と横の長さの和をコスト C_S とする。

● N を小さくするためのコスト (C_N) の設定

展開図の部品数がいくつになるかは、すべての面の展開が完了するまで不明であり、展開の途中では部品数を評価することができない。そのため、経験則に基づいてコストの設定を行った。

まず、部品数を減らすためには、面の干渉を減らす必要があるため、展開済みの面が局所的に集中せず、幅広い展開図が作成される方がよいと考えられる。そこで、展開される面が、展開済みの面全体の重心から離れている方がよいとする。

また、小さな部品が最後に多数残らない様にするため、できるだけ開始点となる面の近くに存在する面から順に展開していくようにする。

さらに、展開後の面の干渉を減らすには、一般に谷折りの部分を切断した方がよい [6]。そこで、候補の面と展開済みの面とを接続する稜線が谷折りであるものはコストを大きくする。

以上より、 C_N は以下のように定めた。

$$C_N = (\text{展開図の重心と候補の面の重心との距離の逆数}) \times (\text{開始点となる面の重心と候補の面の重心との距離}) \quad \{\times 2: \text{稜線が谷折の場合}\}$$

4 結果

前章で述べたアルゴリズムを図 1(右) のトラの頭のモデル (面数 200、辺数 300、頂点数 102) に適用した結果を表 1 に示す。

探索アルゴリズムには前節で述べた、コスト評価を用いた深さ優先探索 (表中「深さ」) と、コスト評価を用いた欲張りアルゴリズム (表中「欲張」) を使い、候補の面に設定するコストには、 C_L 、 C_S 、 C_N および、次式で表される C^* を用いた。

$$C^* = C_L/\overline{C_L} + C_S/\overline{C_S} + C_N/\overline{C_N}$$

ここで、 $\overline{C_L}$ 、 $\overline{C_S}$ 、 $\overline{C_N}$ はそれぞれ、展開候補の面の C_L 、 C_S 、 C_N の平均である。

表中の L, S, N の値は、3.2 で定めた各評価値であり、値が小さい程組み立て易い展開図であることになる。なお、L と S については、コスト評価を行わない「手法 A」に対する相対値を記した。

探索の開始点となる面によって展開図が異なるが、今回はランダムに選択した一つの面を、各手法で共通の開始点として使用した。

手法	コスト	探索法	L	S	N	計算回数
A	なし	深さ	1	1	11	297
B	C_L	深さ	0.85	0.50	13	294
B'		欲張	0.71	0.58	8	6601
C	C_S	深さ	1.03	0.57	17	294
C'		欲張	1.00	0.49	12	8569
D	C_N	深さ	0.92	1.14	5	299
D'		欲張	0.93	1.27	2	4769
E	C^*	深さ	0.92	0.69	8	297
E'		欲張	0.77	0.50	4	7356

表 1: 各手法によって生成された展開図の評価値

結果の表 1 より、局所的なコスト設定によって、それぞれの評価値を小さくすることが可能であることがわかる。

つまり、 C_L の設定を行った手法 B、B' によって、実際に評価値 L が小さくなっていることが確認できる。同様に、 C_S の設定を行った手法 C、C' によって、評価値 S が小さくなっている。また、 C_N の設定を行った手法 D、D' によって、評価値 N が小さくなっているが、部品数を 1 にすることはできなかった。

各コストを総合的に用いた欲張りアルゴリズムの手法 E' が、各評価値を小さくすることができ、全体的により展開図を作成できたと判断できる。

深さ優先探索に比べ、欲張りアルゴリズムの計算回数は、およそ 20 倍であるが、その分よい結果が出ている。それでも計算回数はわずか数千のオーダーであり、展開図は数秒で作成することができた。展開図の作成が高速であるため、コストの設定方法を変えて何度も試行する事が可能である。開始点となる面を変更したり、手法 E' については、各コストに対する重み付けを変更するなどして、試行することができる。

比較のために、手法 A と手法 E' によって生成された展開図を図 5 および図 6 に示す。なお、各部品のレイアウトは手作業で行った。これらの図から、手法 A では部品数が多く、不必要に広がった展開図になっているが、手法 E' では部品数も少なく、まとまった展開図になっていることがわかる。また、表の結果より、切断すべき稜線の長さも 2 割ほど少なくなっていることがわかる。

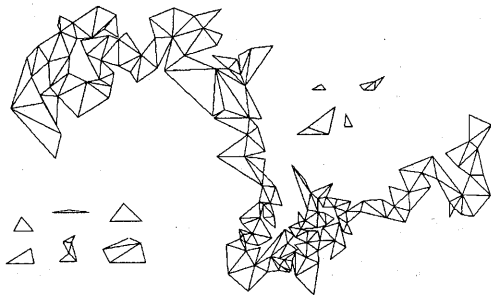


図 5: トラの頭のモデルに手法 A を適用した例

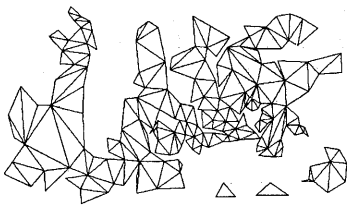


図 6: トラの頭のモデルに手法 E' を適用した例

実験としてこれらのアルゴリズムを簡単な幾何形状に適用したところ、手法 C' を用いると、秩序的な展開図が作成された。この例を図 7、図 8 に示す。

このように、メッシュのサイズが規則的に変化する形状に対しては、稜線の長さを基準に展開図を作成することで、人間が想像する展開図に近い、規則的な展開図を作成することができた。この他に、円錐や円柱のポリゴンモデルに対しても整った展開図が出力された。

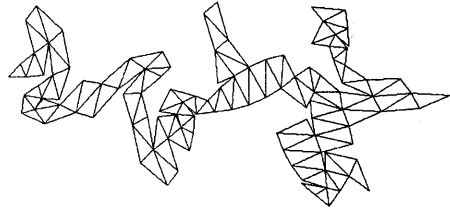


図 7: 球体のポリゴンモデルに手法 A を適用した例

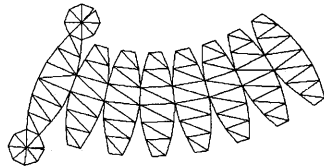


図 8: 球体のポリゴンモデルに手法 B' を適用した例

5 本手法の適用例

5.1 システムの実装

本論文で紹介したアルゴリズムを使用し、展開図を自動的に作成するアプリケーションを作成した(図 9)。

このアプリケーションは obj 形式のポリゴンモデルを読み込み、展開図を ps 形式で出力する。組み立て時の利便のために、接続するべき稜線のペアには同一の識別番号が記され、山折りの稜線は実線で、谷折りの稜線は破線で記される。

オプションで、モデルの展開図を作成している様子をリアルタイムで見ることができるほか、展開図作成の開始点となる面をユーザが指定できる。

また組み立てる際に山折りになる稜線と谷折りになる稜線を異なる色で 3D 表示したり、切断されている稜線を誇張表示する [7](図 10) ことにより、実際に紙工作を行う際の作業を支援する機能も含まれる。

5.2 ウサギのモデルへの適用

本研究の目的である自由形状モデルの展開図作成の例題として、図 1(左)のウサギのモデルの展開図を作成し、実際にそれをもとに紙模型を作成した。

ウサギの首や耳のつけねのように、大きくくびれたところがある場合、そこでパーツに分けた方が組み立てやすくなるため、前処理として手作業でメッシュモデルを複数のパーツへ分け、手法 E' を適用した。

紙面の都合上、作成された頭と耳の展開図のみを図 11 に示す。

この展開図をもとに、作成されたウサギのモデル(面数 300)を図 12 に示す。完成後のサイズは手のひらに収

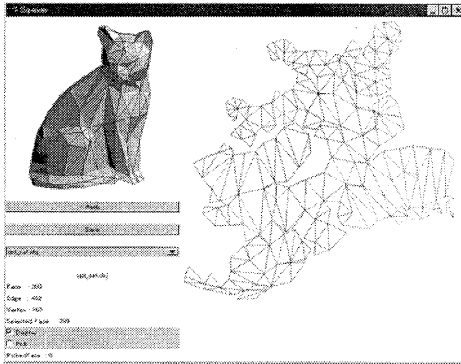


図 9: 実装したアプリケーション

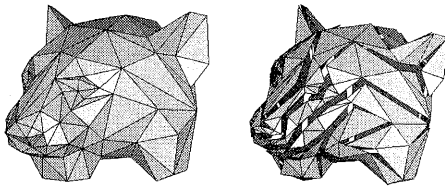


図 10: 切断稜線の誇張表現

まる程度で、展開図の切り取りと組み立てに要した作業時間は5時間程度であった。

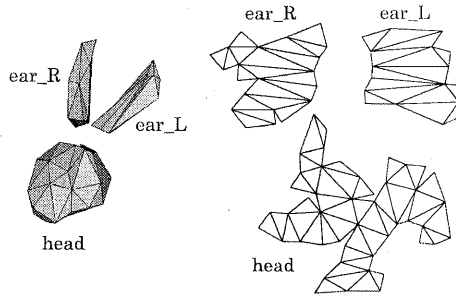


図 11: パーツに分けたウサギのモデルの展開図

6 結論と展望

三角形メッシュモデルに対し、局所的なコスト評価を用いて、組み立て易さを考慮しながら高速に展開図を作成するアルゴリズムを提案した。

また、実際に本手法を使用して作成した展開図をもとに紙模型を組み立て、その有効性を示した。

課題としては、ウサギのモデルのように、大きくくびれた箇所がある場合、展開図作成の上でも、実際に組み立てる上でも、複数のパーツに分けたほうがよいことは

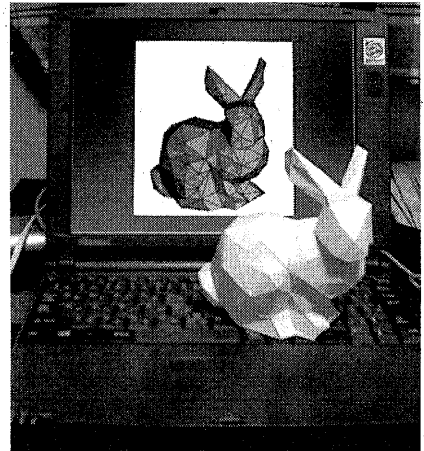


図 12: 作成した模型

明らかであるため、モデルのパーツ分けの自動化が挙げられる。本手法では部品数が増えると、細切れに小さな部品が生じやすいという問題があるため、なるべく小さな部品を生じさせないようにする工夫が必要である。また、実装したアプリケーションでは、糊代部分を考慮していないため、糊を使用して面の貼りあわせを行う場合は、新たに糊代部分を生成するための工夫が必要となる。

参考文献

- [1] YAMAHA MOTOR. ペーパークラフト総集編. <http://www.yamaha-motor.co.jp/papercraft/index.html>.
- [2] 堤江美子. 人体形状のモデル化・類型化とその被服設計への応用. グラフィックスと CAD 研究会 46-14, pp. 95-102. 情報処理学会, 1990.
- [3] 田村雄一郎, 高橋利治, 山口富士夫. ソリッドモデルの展開図作成に関する研究. 1992 年度精密機械工学会秋期大会学術講演会講演論文集, pp. 439-440. 精密機械工学会, 1992.
- [4] 磯田浩, 鈴木賢次郎. 図学入門. 東京大学出版会, 1986.
- [5] A.V. エイホ, J.E. ホップクロフト, J.D. ウルフマン. データ構造とアルゴリズム. 培風館, 1992.
- [6] 島田静雄. 幾何モデリングの演習. 埼玉大学情報システム工学科近藤研究室 <http://www.ke.ics.saitama-u.ac.jp/kondo/lect/geomap/05-1.html>, 1996.
- [7] G.Taubin and J.Rossignac. Geometry compression through topological surgery. In *SIGGRAPH'98 course notes No.21 3D Geometry Compression and IBM Research Technical Report RC-20340*, January 1996, 1998.