Vol.2009-CG-137 No.8
2009/11/5

鞍点特徴を利用した3次元メッシュ展開

葉　斐† Saw Seow Hui† 高橋 成雄†

3次元メッシュ形状を2次元の展開図に変換する時，展開図を2次元平面上で自己交差をもたないようなかたちで得るためには，与えられた3次元メッシュを切り開く方法に注意を払う必要がある．本研究報告では，3次元メッシュを鞍点特徴に沿って切り開くことで，そのような展開図における自己交差を最小化する手法を提案する．3次元メッシュの鞍点は隣接稜線間に挟まれる角度の和が360度より大きいため，その鞍点に隣接する面を二つのセグメントに分断する必要がある．鞍点に接続する二本のエッジを展開図の境界線に対応する切断線の候補として抽出して，自己交差を最小化する．

Unfoldings for Triangulated Models based on saddle features

Fei Ye† Saw Seow Hui† Shigeo Takahashi†

When we unfold a 3D mesh into a 2D plane, it is necessary to take into account the way or the method of cutting or breaking its edges in 3D mesh so that it can be unfolded into a 2D plane without self-intersection. In this research, we proposed a cutting approach along the edges in 3D mesh based on the saddle point feature and it effectively minimizes the self-intersection. We are required to subdivide the adjacency faces that correspond to the saddle point into 2 segments because saddle point always yields to more than 360 degree when summing up the angles between the adjacent ridgelines in the 3D mesh. To achieve this, we cut along the 2 edges that connect to the obtained saddle points into unfolded borders.

†東京大学
The University of Tokyo

1. 背景

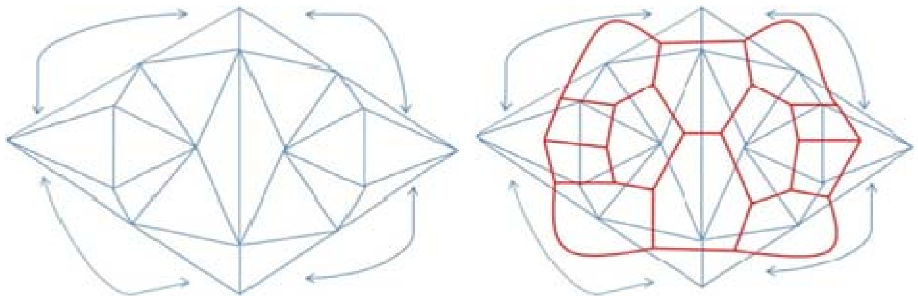
３次元多面体形状を平面に展開した図は，対応する境界線を貼り合わせることで，平面から立体的な形状を組み上げることができる．そのため，単なる趣味や道楽の対象としてのみならず，建築模型物のプレゼンテーションなどへの活用や，教育分野での教材としての活用など，幅広い分野で用いられている．

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，この作業には時間と労力が非常にかかる．また，展開する際にどの稜線を切断し，どの稜線を接続したまま残すかを決定する必要があり，この決定方法に拠って展開図の形は異なったものになる．本稿では，パーツ数が少ない展開図を作成することを目標として，3 次元メッシュから自己交差の少ない展開図を生成するための新しい手法を提案する．

2. 関連研究

3次元空間上の曲面および多面体を平面に展開する手法は古くから研究されている．平面に展開できる曲面は可展面と呼ばれるものに限られ，それ以外の曲面を展開するには，それらを可展面の集合体で近似する必要がある．従来，展開図作成の対象となる多面体形状は, 三角形メッシュとして表現されることが多い. 展開図の作り方とは，ある面を出発点として，その面に3次元上で接続する面を1つずつ配置してゆく．既に展開済みの面と自己交差が起きる場合には，自己交差が起きない別の面を選択して展開を続ける(逐次法) [1]．しかし，展開が終了する前に自己交差を起こさずに配置できる面が存在しなくなった場合には，まだ展開していない面を 1 つ選択し，その面を新たな展開図パーツの出発点として展開作業をやり直していく．そのため，パーツの個数が多くなりがちで，結果として貼り合わせる展開図の境界部分が必要以上に長くなるため，組み立てに時間がかかる展開図となる．

Mitaniら[2]は，三角形メッシュを簡単に分割できるように各パーツ(例えば，ウサギでは頭，胴体，脚）に分割したのち，分割したパーツごとに三角形ストリップ(三角形を蛇腹折りのように連続的につなげた帯状形状)へ変換することで, 展開図を作成する手法を提案した．この手法では，元メッシュの特徴を最大限に保持するため，展開図のパーツ数は多く存在し, 辺同士のマッチングがとりにくいという問題点が存在する．Massarwiら[3]は，三角形メッシュを円筒に近似できるような形にパーツごとに分割したあと，自己交差が起きない円弧状のパーツの展開図を作成する手法を提案した．この手法では，円筒に近似できる形を探索するため，凸凹が多いメッシュの場合，パーツ数は膨大な数となる．



(a)三角形メッシュ(青)　(b) 双対グラフ(赤)

(c) 完全マッチング(緑)　(d) 三角形ストリップ

図 1

また, Mitaniら[4]はまず多面体の全ての面をバラバラに展開平面に配置し. その後, ３次元空間で接続している隣接面を展開平面上でも接続し，徐々にパーツを大きくまとめあげていく手法(集約法)を提案した．しかし，すべての展開図のパターンから最適なものを見つけ出すことは計算量がかかる上，パーツ数の問題も考量してない．

Tachiら[5]は三角形メッシュから折紙ができるように展開して, 自己交差が起きない展開図を生成する手法を提案した．この手法は，対象となる三角形メッシュの形は変わらないが，折紙ができるよう展開図に折り目と折り面を追加するため，結果として面数がかなり増える．そのため，組み立てには長い時間がかかってしまう．

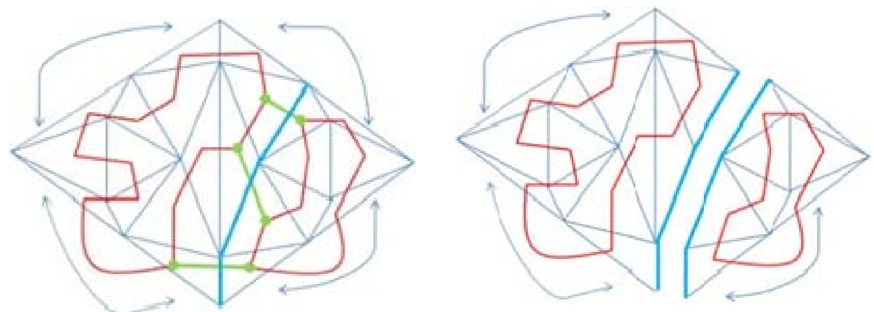
そこで本稿では，元三角形メッシュを変化させることなく，展開図のパーツ数が多くなる原因であると考えられる，自己交差の起きる箇所を減らすことを目的とした展開図作成手法を提案する．

3. 提案手法

三角形メッシュから複数の三角形ストリップを求める際に，鞍点を利用して，自己交差の少ない切断方法を探索する．三角形ストリップ同士の結合を行いながら，パーツ数が少ない展開図を生成する．

3.1 三角形ストリップ

本手法は，Gopiら[6]が提案した，三角形メッシュを単一三角形ストリップに展開する手法を利用していく．まず，与えられた三角形メッシュ(図1(a))上の三角形の重心をノードとし, それらのノードを結ぶことで三角形メッシュの双対グラフ(図1(b))を構築し，得られた双対グラフのノードに対する完全マッチング(図1(c))を求める．ここで，完全マッチングとは，そのノード集合が双対グラフのノード集合と等しくなっていてかつ，エッジ集合は互いに端点を共有しない部分グラフとして定義される．Gopiらの手法では，三角形メッシュにおいて完全マッチングに含まれる双対エッジと交差するエッジ(図2(a))が，展開図の境界線に対応する切断線(図2(b)).



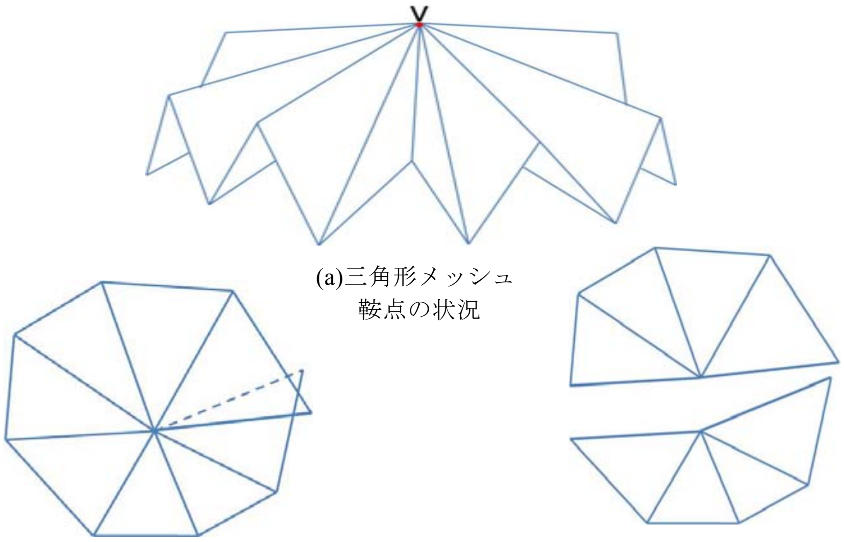
(a) 完全マッチング(緑)と交差するエッジ(青)　(b) 切断線(青)

図 2

3.2 自己交差数最小化のための鞍点の利用方法

オリジナル三角形メッシュのすべての頂点ごとに，接している角の和を計算して，360 度より大きい頂点を調べ，そのような頂点を鞍点と判定する．もし，三角形メッシュ上のノードに接している角の和が 360 度を超える(図 3(a))場合，平面に展開すると自己交差が起きる(図 3(b))ので二つのセグメントに分ける必要がある(図 3(c))ため鞍点を抽出する．

実際のアルゴリズムとしては，最初にすべての鞍点を抽出する．抽出した鞍点は角



(a)三角形メッシュ
鞍点の状況

(b)鞍点の隣接する面
を平面に展開する

(c)二つのセグメント
に分ける

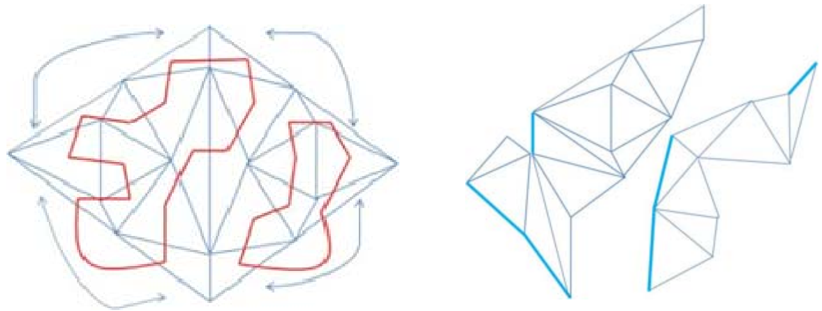
図 3

度の和が大きい順にソートする.そこで,各鞍点から二つの切断線の候補を取得する.さらに，この切断線のエッジと交差するエッジを双対グラフの完全マッチングのエッジとする．完全マッチングを生成できない場合は，完全マッチングを生成できるように別の切断線を選択して続ける．すべての鞍点から二つの切断線を取り終わったら，残りのエッジは完全マッチングを生成するまで計算する．今回は鞍点から二つの切断線が取れない場合は検討しない.

以上のように，この手法ではメッシュの鞍点から切り開くことで，展開図の自己交差を減らすことができる.実際には,双対グラフから完全マッチングのエッジを削除し,残った双対グラフのエッジをたどりながら面を順次つなげて展開することで，メッシュを三角形ストリップに分解していく.

3.3 展開図

三角形ストリップを展開する時は，ある面を始点とし，それに接続する面を，三角形ストリップ(図 4(a))の順番で同一平面上に展開していく(図 4(b)).



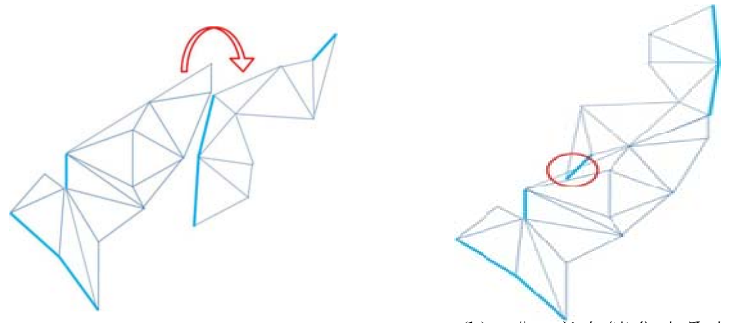
(a) 三角形ストリップ　　(b) 展開図

図 4

3.4 展開図パーツの結合

以上で，自己交差が起きない展開図パーツを生成した．そこで，パーツ数の個数を減らすために，展開図パーツの結合を行う.

結合する際に，展開図の切断線を調べると，切断線の端点がメッシュ上の鞍点の場合は，結合した場合二つのセグメントに分けていない状況に戻る．そのため，展開図の切断線の両端点ともがメッシュ上の鞍点ではないエッジを利用して，パーツとパーツを繋げる．しかし，パーツとパーツの形を考慮しないと，パーツ通しで自己交差が起きる可能性がある(図 5)．さらに，パーツは三角形ストリップを展開する始面に拠って形が変わるため，パーツ数が多い場合は計算量が膨大になる.

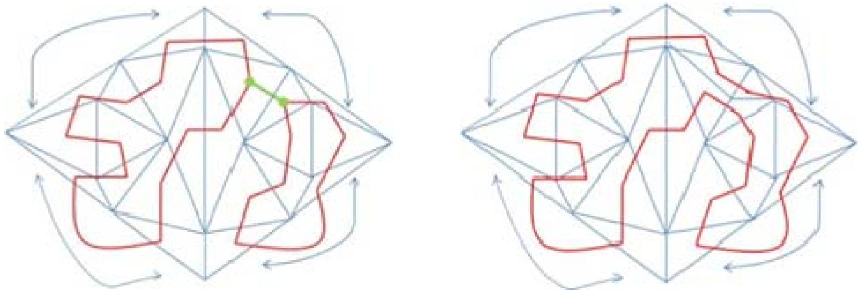


(a) 展開図

(b) パーツを結合すると
自己交差が起きる

図 5

我々は Gopi ら[6]が提案した，三角形メッシュを単一三角形ストリップにする手法を利用する．複数のストリップを生成する時，三角形メッシュの三角形面に分割処理を施してひとつのストリップに併合する(図 6).



(a) 分割処理する面　(b) 単一三角形ストリップ

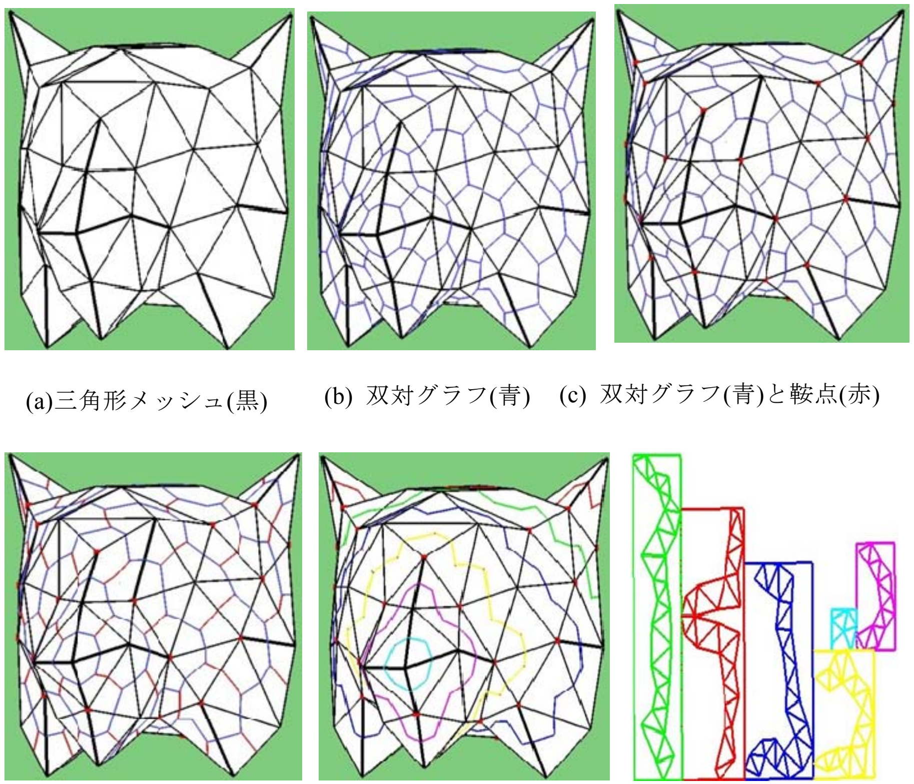
図 6　複数の三角形ストリップを単一三角形ストリップにする

4. 結果

本節では，提案手法を用いて虎のデータから得られる結果を示す．実装環境としては，メモリ2GB, Intel Core2DuoE6550(2.33GHz, 2MB cache) を搭載したデスクトップPCである．

図7 (a) は，今回利用する虎のデータ(黒い線)であり，データの頂点数は58で，面数は112 である．図7 (b) は，元メッシュから得られた双対グラフ(青い線)．さらに，図7 (c)で見てとれるようにメッシュ上全ての鞍点(赤い点)を抽出する．図7 (d) は，その鞍点を囲んでいるエッジは最低二本以上を切断線として，完全マッチング(赤い線)を生成する．図7 (e) で双対グラフから完全マッチングのエッジを削除し，六個の自己交差が起きない三角形ストリップを生成した．図7 (f)はこの六個の三角形ストリップを，主成分分析を用いることで長方形領域に収まるように向きを調整したのち，2次元平面上に敷き詰めて配置した[7].

そこで，三角形メッシュの三角形面に分割処理を施して複数の三角形ストリップを一つにまとめたい．今回の虎のデータでは，赤色の三角形ストリップと緑色の三角形ストリップが，一つの三角形ストリップになると自己交差が起きる，緑以外の三角形ストリップと赤色の三角形ストリップを繋げると，自己交差が起きない三角形ストリップを生成する事はできる．

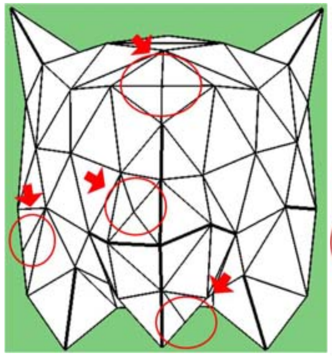


(a)三角形メッシュ(黒)　(b) 双対グラフ(青)　(c) 双対グラフ(青)と鞍点(赤)

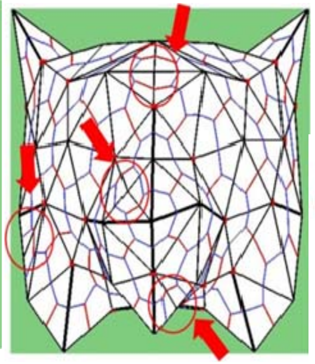
(d) 完全マッチング(赤)　(e) 三角形ストリップ　(f) 展開図

図 7

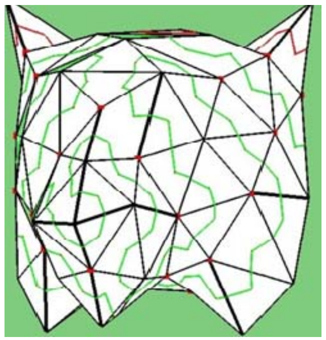
図8 (a)は，三角形メッシュの4か所を分割処理した結果．図8 (b)は，分割処理した三角形メッシュの鞍点から二つの切断線を取った完全マッチングの結果．図8 (c)は，三角形ストリップを合併することで，二個の三角形ストリップになった結果である．図8 (d)は，二個の三角形ストリップを，主成分分析を用いて2次元平面上に配置した結果．図8 (e)は，二個の三角形ストリップを結合して，虎データから一つだけのパーツとして作り出された展開図.



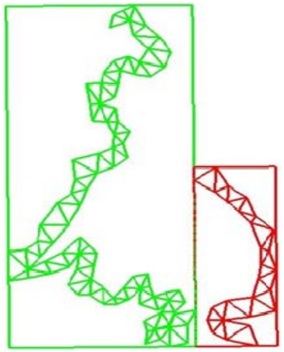
(a) 分割処理した三角形メッシュ



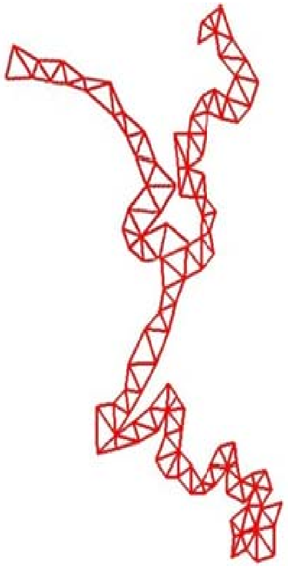
(b) 完全マッチング



(c) 三角形ストリップ



(d) 展開図



(e) 結合後の展開図

図 8

5. 結論と今後の課題

結果より，本手法が三角形メッシュデータから自己交差のない 1 つのみのパーツとして展開図を作りだせることを示した．

今後の課題として，鞍点から二つの切断線が取れない場合を検討したい．また，鞍点から二つの切断線を取るアルゴリズムを最適化して，より効率的に，展開図のパーツの合併を行う．最後に，展開図を実際に手で組み立てる時間を計ることによる展開図の評価をしていく．

参考文献

[1] J. Mitani and H. Suzuki, “A Method for Generating Developments of Triangular Polyhedral Models,” *IPSJ SIG Notes, Vol. 99, No. 70,* pp.13-18, 1999.

[2] J. Mitani and H. Suzuki, “Making Papercraft Toys from Meshes using Strip-based Approximate Unfolding,” *ACM Transactions on Graphics, Vol. 23, No.3*, pp. 259-263, 2004.

[3] F. Massarwi, C. Gotsman, and G. Elber, “Papercraft models using generalized cylinders,” *in PG ’07: Proc. the 15th Pacific Conference on Computer Graphics and Applications,* pp. 148–157, 2007.

[4] J. Mitani and H. Suzuki, “A face−gathering approach for generating unfolded pattern of polyhedrons,” Journal of graphic science of Japan, Vol. 39, No.4, pp. 3-9, 2005.

[5] T.Tachi: "Origamizing Polyhedral Surfaces," *IEEE Transactions on Visualization and Computer Graphics,* 2009.

[6]M. Gopi and D. Eppstein, “Single-Strip Triangulation of Manifolds with Arbitrary Topology,” *Computer Graphics Forum, Vol. 23, No. 3,* pp. 371-379, 2004.

[7] T. Igarashi and D. Cosgrove, “Adaptive Unwrapping for Interactive Texture Painting,” *Symposium on Interactive 3D Graphics,* pp. 209–216,2001