

## 離散グラフの選択的詳細化に基づく多面体上の近似最短経路算出とその応用

金井 崇 鈴木 宏正†

理化学研究所 素形材工学研究室

〒351-0198 埼玉県和光市広沢 2-1

† 東京大学大学院工学系研究科 精密機械工学専攻

E-mail: kanai@postman.riken.go.jp

多面体モデル上の近似最短経路を計算するための、新しいアルゴリズムについて提案する。本手法は主にダイクストラの手法を用い、多面体の離散グラフの選択的詳細化に基づく。このアルゴリズムは近似手法であるが、高速である、実装が簡単である、近似精度が高い、数値的に頑健である、などいくつかの重要な利点を持つ。また、本近似アルゴリズムと、非凸多面体の正確な最短経路を計算することのできる拡張 Chen & Han (ECH) アルゴリズムに対し、計算精度と計算時間に関する比較を行った。その結果、我々の例題では、ECH アルゴリズムに比べて 0.4% 以内の近似精度に収まる経路が、おおよそ 100-1000 倍の速さで求めることができた。さらに、本近似アルゴリズムの形状モデリングに対する二つの応用についても議論する。

### Approximate Shortest Path on a Polyhedral Surface Based on Selective Refinement of the Discrete Graph and Its Applications

TAKASHI KANAI HIROMASA SUZUKI†

RIKEN, Materials Fabrication Lab.

2-1 Hirosawa, Wako-shi Saitama, 351-0198, Japan.

†University of Tokyo, Dept. Precision Engineering

E-mail: kanai@postman.riken.go.jp

A new algorithm is proposed for calculating the approximate shortest path on a polyhedral surface. The method mainly uses Dijkstra's algorithm and is based on selective refinement of the discrete graph of a polyhedron. Although the algorithm is an approximation, it has the significant advantages of being fast, easy to implement, high approximation accuracy, and numerically robust. The approximation accuracy and computation time are compared between this approximation algorithm and the extended Chen & Han (ECH) algorithm that can calculate the exact shortest path for non-convex polyhedra. The approximation algorithm can calculate shortest paths within 0.4% accuracy to roughly 100-1000 times faster than the ECH algorithm in our examples. Two applications are discussed of the approximation algorithm to geometric modeling.

#### 1 はじめに

多面体、特に三角形面から構成されるメッシュは、コンピュータグラフィックス (CG) や、関連する分野における基本的な表現形式である。近年では、レンジスキャナや GPS などの発展により、大規模な(多くの面を持つ)多面体が比較的安易に生成されるようになった。例えば CG の分野では、高密度測定点群から多面体を迅速に生成するための研究 [3] や、これらの多面体を利用するための研究が、現在でも活発に行われている。また地理情報システム (GIS) の分野では、TINs (Triangular Irregular Networks) と呼ばれる地形を表すモデルが一般的に用いられている。

一方、このような多面体上の最短経路問題は、計算幾何学における基本的な問題であり、ロボティクスや

地理情報などの分野への応用を考える上で重要な技術の一つである。我々の目的は、多面体上の最短経路を計算するための効率的なアルゴリズムを考え、それを形状モデリングに応用することである。二次元平面内のポリゴン、三次元多面体、そして三次元空間内での最短経路問題に関しては、これまでに多くの研究がある。しかしながら、特に(非凸の場合を含む)多面体に関する、正確な最短経路算出に関するアルゴリズムは [14, 2], 多くの計算時間および空間量を消費するので、大規模な多面体を扱うのに実用的ではない。

そこで我々は、近似最短経路を求める問題 [9, 13] に注目する。本報告で提案するアルゴリズムは、[9] と同様ダイクストラのアルゴリズムを基本とする。本アルゴリズムの特徴は、多面体の離散グラフを選択的に詳細化する、すなわち、ダイクストラのアルゴリズム

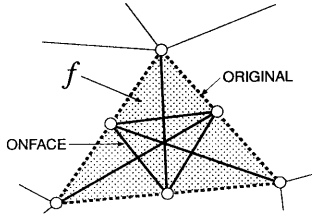


図 1: エッジの分類

ムを繰り返し計算することで、経路の存在する範囲を絞り込んでゆく。本アルゴリズムはいくつかの利点を持つ。例えば、高速、実装が簡単、近似精度が高い、より少ない空間量で経路を求めることが可能、などが挙げられる。これらの持つ利点を実証するために、正確な最短経路算出アルゴリズムである Chen & Han のアルゴリズム [2] との比較を行っている。

また、我々の手法のモデリングへの応用についても議論する。我々の提案する手法はあくまで近似手法であり、正確な経路（またはその長さ）を求める必要のある応用には不向きである。だが、我々の方法を効率的に利用できる二つの例を示す。一つが、最短経路算出の領域境界指定への応用である。このような境界は最短経路である必要ではなく、むしろ滑らかな線が、高速かつ効率的に求まることが重要である。またもう一つが、それほど厳密に値を知る必要のないような応用でのインタラクティブな測地線計測である。

## 2 近似最短経路の算出

多面体  $M$  が与えられたとき、 $M$  上の二つの頂点  $V_S, V_D$  間の最短経路  $\mathcal{L}$  を、

$$\mathcal{L} = \{v_1, v_2, \dots, v_n, e_1, e_2, \dots, e_{n-1}\}, \quad (1)$$

$$V_S = v_1, \quad V_D = v_n, \quad e_i = \{v_i, v_{i+1}\},$$

と定義する。ここで、 $v_i$  と  $e_i$  はそれぞれ  $M$  の面上の点もしくはエッジを示す。また、 $l = |\mathcal{L}|$  を最短経路  $\mathcal{L}$  の長さとする。このとき、

$$l = |\mathcal{L}| = |e_1| + |e_2| + \dots + |e_{n-1}|, \quad (2)$$

である。ここで  $|e_i|$  は、エッジ  $e_i$  の両端の頂点  $v_i, v_{i+1}$  間のユークリッド距離である。

### 2.1 経路計算のための前処理

$\mathcal{L}$  を求めるための前処理として、 $M$  から離散グラフ  $G^0(v, e)$  を算出する。 $G^0$  は、 $M$  の頂点とエッジの他に、付加的な頂点とエッジから構成される。

ここで、以後のアルゴリズムの中で、エッジを付加するかどうかを判定するため、 $G^0$  のエッジを分類する。図 1 のように、 $M$  のエッジ、およびそれを分

割したものを ORIGINAL とし、 $M$  のエッジではなく、面上にのっているエッジを ONFACE とする。

$G^0$  を生成するために、まず  $M$  の頂点とエッジを  $G^0$  に加える。そして、付加的な頂点とエッジを次のように生成する： $M$  のエッジ上にいくつかのSteiner Point (SP) と呼ばれる中間頂点を  $G^0$  に加える。一つのエッジ上に生成するSteiner Point の数は、エッジの長さでパラメータ  $\gamma$  によって決まる。エッジ  $e$  の長さを  $|e|$  とすると、 $(|e| \bmod \gamma) - 1$  をそのエッジにおけるSteiner Point の数とする。 $\gamma$  が小さいほどより多くのSteiner Point が生成される。ただこのままだと、 $\gamma$  はスケール依存の変数になる。 $\gamma$  をなるべく  $M$  の大きさには依存せず定義したい。そこで、 $M$  全体が単位立方体内に収まるようスケール変換し、エッジの長さを正規化する。 $\gamma$  は、経路の近似精度の向上と、計算時間、空間量の増大とのトレードオフを決めるためのユーザ変数となる。

生成された SP の頂点と  $M$  の頂点を用いて、中間エッジを生成する。 $G^0$  の頂点ペア  $\langle v_a, v_b \rangle$  が次の条件 1 のどちらかを満たすとき、 $e = \{v_a, v_b\}$  となるエッジを  $G^0$  に加える：

#### 条件 1

1.  $\langle v_a, v_b \rangle$  はある面の異なるエッジ上にある (分割) 頂点である
2.  $\langle v_a, v_b \rangle$  は同じエッジ上にある隣接した (分割) 頂点である

### 2.2 近似最短経路算出アルゴリズム

二頂点  $V_S, V_D$  間の最短経路  $\mathcal{L}$  は、 $G^0$  を入力として以下のアルゴリズムによって求める。アルゴリズムは、ループカウンタ  $i$  を 0 から一つずつ進めるごとに、次の STEP 1 - STEP 4 を繰り返す。

#### STEP 1 $G^i$ 内の最短経路の算出

ダイクストラのアルゴリズムにより、グラフ  $G^i$  の最短経路を算出する。算出された経路の長さを  $l^i$  とする。最短経路は  $G^i$  の頂点とエッジで構成される。

#### STEP 2 最短経路からの新しいグラフ $G^{i+1}$ の抽出

最短経路  $\mathcal{L}^i$  を辿っていき、その頂点を  $G^{i+1}$  に加える。また、経路の頂点を含むエッジのうち、ORIGINAL であるエッジと接続するもう一方の頂点を  $G^{i+1}$  に加える。 $G^{i+1}$  の頂点ペア  $\langle v_a, v_b \rangle$  のうち、エッジ  $e = \{v_a, v_b\} \in G^i$  が存在すれば、そのエッジを  $G^{i+1}$  に加える。

#### STEP 3 グラフ $G^{i+1}$ への頂点とエッジの付加

$G^{i+1}$  のエッジに対し、 $e$  が ORIGINAL ならば、エッジ上に  $m$  個の SP を  $G^{i+1}$  に付加する。 $G^{i+1}$

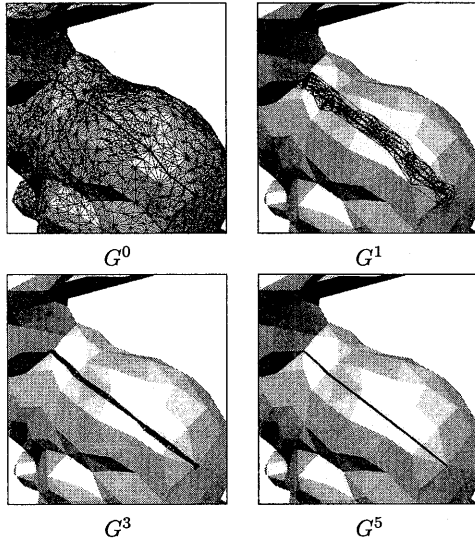


図 2: アルゴリズムの収束

の頂点ペア  $\langle v_a, v_b \rangle$  に対し、条件 1 のどちらかを満たし、かつ  $e = \{v_a, v_b\} \notin G^{i+1}$  となるときに  $e$  を  $G^{i+1}$  に追加する。このとき、1. を満たすものは ONFACE, 2. を満たすものは ORIGINAL とする。また 2. のとき、 $G^{i+1}$  の分割されたもとのエッジは消去する。

#### STEP 4 グラフの更新

$G^i \leftarrow G^{i+1}$  とする。

STEP 1 が終了した時点で、現在のグラフ  $G^i$  とその一つ前のグラフ  $G^{i-1}$  における経路の長さ  $l^i, l^{i-1}$  を比較し、 $|l^i - l^{i-1}| < \epsilon$  となったところで処理を終了する。2.1 節で述べた  $G^0$  への SP とエッジの追加は、一つのエッジに対する SP の数を除いて、上のアルゴリズムの STEP 3 に相当する。また  $m$  は、ユーザが決めるべきもう一つの変数であり、収束回数と計算時間、空間量とのトレードオフとなる。 $m$  を、エッジ毎に変化させることも可能であるが、 $G^0$  の生成の段階でエッジの長さはほぼ均一になっているので、我々はすべてのエッジに対し固定量を選択している。

本アルゴリズムの特徴を表すために、図 2 には、兎のモデル (頂点数 525, エッジ数 1527) のうちのある二点間の経路を求めたときの様子を示している。なおこの例では、 $\gamma = 0.04, m = 1$  と設定し、計 7 回の繰り返し計算で収束した。図 2 はそれぞれ  $G^0, G^1, G^3, G^5$  の各段階での STEP 1 終了後の様子を示している。図の細線に STEP 2 と STEP 3 により生成、追加された離散グラフ  $G$  を、太線は各段階で算出されたグラフ上の最短経路を示している。

	$v$	$e$	$l^i$	$T$ (sec.)
$G^0$	2,859	16,055	0.52200	0.083
$G^1$	79	361	0.51408	0.017
$G^2$	75	365	0.51277	0.017
$G^3$	76	371	0.51252	0.017
$G^4$	78	393	0.51237	0.017
$G^5$	78	393	0.51234	0.017
$G^6$	70	313	0.51234	0.017

表 1: 本アルゴリズムの各離散グラフにおける要素数、最短経路の長さ、計算時間

図より、グラフの存在する領域が次第に狭められ、最後には一本の折れ線になっていく様子がわかる。

表 1 には、アルゴリズムの各収束段階における要素数 (頂点とエッジの数)、各グラフ  $G^i$  における最短経路の長さ  $l^i$ 、および各グラフでの STEP 1 - STEP 4 の合計の計算時間を示している。要素数は、最初の  $G^0$  だけ圧倒的に数が多いが、その他の段階では大体ほぼ一定数のエッジの探索 (本例では 300 から 400 の間) で収まっている。本アルゴリズムの処理にかかる計算時間は、ダイクストラのアルゴリズムによる  $G^0$  上の経路算出に大きく依存する。我々は、 $O(n \log n)$  の計算時間で処理できる、半順序木による優先度付き待ち行列に基づく方法 [1] を用いた。このアルゴリズムはグラフのエッジの長さを評価するので、グラフ  $G^0$  のエッジの数、ひいては  $\gamma$  の大きさが計算量に大きく影響する。

本手法の実装において、 $G$  を表すための次のようなシンプルな構造を用意している:  $G$  を表すデータ構造は、 $M$  の頂点や SP を表すノードと、 $M$  の (分割) エッジや付加されたエッジを表すリンクからなる。それぞれのノードは、三次元座標値と、そのノードに接続するリンクへのポインタを持つ。リンクは、両端のノードへのポインタや、エッジの長さ、エッジの分類 (ORIGINAL もしくは ONFACE) を表すフラグを持つ。ノードやリンクが  $M$  の頂点やエッジから生成されたものであれば、それらへのポインタも持つ。また、リンクが ONFACE ならば、その背後にある  $M$  の面へのポインタを持つ。

### 3 実験結果と評価

我々の近似最短経路算出アルゴリズムを二つの観点から評価する。一つは変数を変化させたときの近似精度と計算時間の評価であり、もう一つが、計算精度の評価である。我々は Chen & Han の正確な最短経路算出アルゴリズム [2] による結果を評価の基準にした。ただこの手法は、非凸多面体の場合経路が算出できなくなる場合がある。そこで、我々は非凸多面体でも適用できるよう拡張し実装した。以後この手法を拡張 Chen & Han (ECH) アルゴリズムと呼ぶことにする。

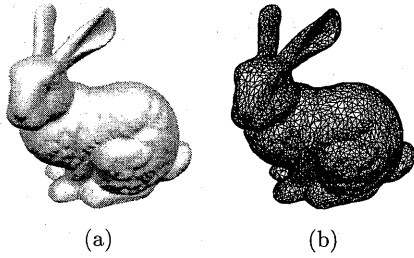


図 3: (a) 兎のモデル (頂点数 35,947, 面数 69,451)  
(b) 簡略化モデル (頂点数 5,047, 面数 9,999)

### 3.1 近似精度と計算時間

まずは、ユーザ指定の変数  $\gamma$  と  $m$  の違いに対する近似精度と計算時間への影響を調べる。例題として、図 3(a) に示される兎のモデルを簡略化したものを用いた。簡略化手法は、Garland らの手法 [4] を用いている。簡略化モデルを用いたのは、我々の ECH アルゴリズムの実装が、経路を求めるために空間量を莫大に消費するからである。図 3(b) に面の数が 9,999 の簡略化モデルを示す。

評価に必要なデータは次のようにして得た: まず、図 3(b) のモデルの頂点群からランダムに 1000 組の頂点ペアを選んだ。これらの組のそれぞれの頂点ペア間の最短経路とその長さを、ECH アルゴリズムと、何種類かの  $\gamma$  と  $m$  を与えた場合の本アルゴリズムにより求めた。また、実装はすべてグラフィックスワークステーション (MIPS R10000 175MHz) 上で行った。

図 4(a) に、近似精度を比較したグラフを示す。グラフの縦軸には 1000 組の頂点ペアに対する本アルゴリズムでの経路の平均長さの、ECH アルゴリズムでの長さに対する割合を示す。図 4(b) には、計算時間を比較したグラフを示す。グラフの縦軸は経路の平均計算時間を示している。また、双方のグラフの横軸は、 $\gamma$  に相当する値を示している。ここではわかりにくい  $\gamma$  の代わりに、 $\gamma$  によって導きだされる一つのエッジに対する SP の数を横軸にしている。

図 4(a) より、近似経路は正確な経路に対して 0.4% 以内に収まっていることがわかる。 $\gamma$  を小さくすればとるほど正確解に近づく。また  $m$  は近似精度の若干の向上に寄与するのがわかる。また図 4(b) より、ECH アルゴリズムの計算時間 (170.245 Sec.) よりも劇的に高速であるのがわかる。 $m$  を大きくすることにより、あるいは  $\gamma$  を小さくすることにより、計算時間は増える。特に  $\gamma$  の計算時間に与える影響は  $m$  よりも大きい。異なる  $\gamma$  に対する同じ経路の計算時間は、 $O(n \log n)$  のオーダーで増加している。

図 4(c) に、本アルゴリズムによる経路の長さ、計算時間の関係を表すグラフを示している。横軸に経路の平均長さを、縦軸に平均計算時間を示す。このグラフでは、計算結果を経路の長さによって 5 つ

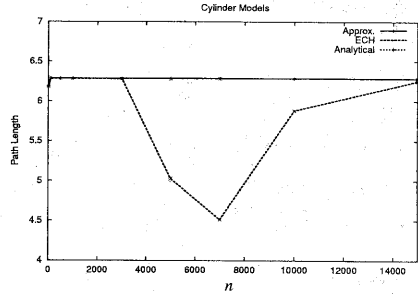


図 5: 本アルゴリズムと ECH アルゴリズム、解析解による計算精度の比較

のグループに分類している。各グループの長さの間隔を 0.2 とし、グループごとの重み付き平均長さをプロットしている。図 4(c) は本アルゴリズムの特徴を良く表している。すなわち、経路が長くなるほど計算時間が増える。異なる経路の長さに対する計算時間は、本手法が経路の近傍のグラフに対するダイクストラのアルゴリズムを繰り返すことから、 $O(k \log k)$  のオーダーで増加する ( $k$  は経路を通過するエッジの数)。図 4(d) には、 $m$  の値を変化させたときの、本アルゴリズムの収束回数を示している。 $m$  を増やすほど、収束回数が減っているのがわかる。

総じて本例では、0.4% の近似誤差内に収まる経路が、正確な解を算出する ECH アルゴリズムの約 100-1000 倍高速に求められることがわかる。ただ、 $\gamma$  や  $m$  は、近似精度や計算時間の向上にある程度寄与するが、それほど劇的な変化はないのと、逆に計算時間が増えることから、計算効率の向上につながるとは必ずしも言えない。

またさらに、面の数がそれぞれ 1000, 5000 程度に簡略化したモデルに対しても試してみたが、同様の結果が得られたので、本報告では割愛する。

### 3.2 計算精度

次に我々は、本アルゴリズムの ECH アルゴリズムに対する一つの優位性を表すため、正  $n$  角柱の周りの長さを計測した実験データを提供する。正  $n$  角柱の周りの長さ  $d$  は  $d = 2n \sin \frac{\pi}{n}$  である。 $n$  を変化した場合の、解析解、ECH アルゴリズム、本アルゴリズムによる経路の長さを計算し比較した。

図 5 に、 $n$  として 0 から 15,000 までのいくつかの値をサンプルし、解析解、ECH アルゴリズム、本アルゴリズムによる  $d$  を計算した結果をグラフ表示している。ECH アルゴリズムは  $n$  が 3,000 を超えたあたりから正確な値を計算できなくなるのに対し、本アルゴリズムでは、 $n$  が 15,000 になっても解析解と一致している。これはアルゴリズムにおける経路の長さの計算方法の違いが原因である。ECH アルゴリズムは、面をエッジを軸にして回転し、二つの面が同一平

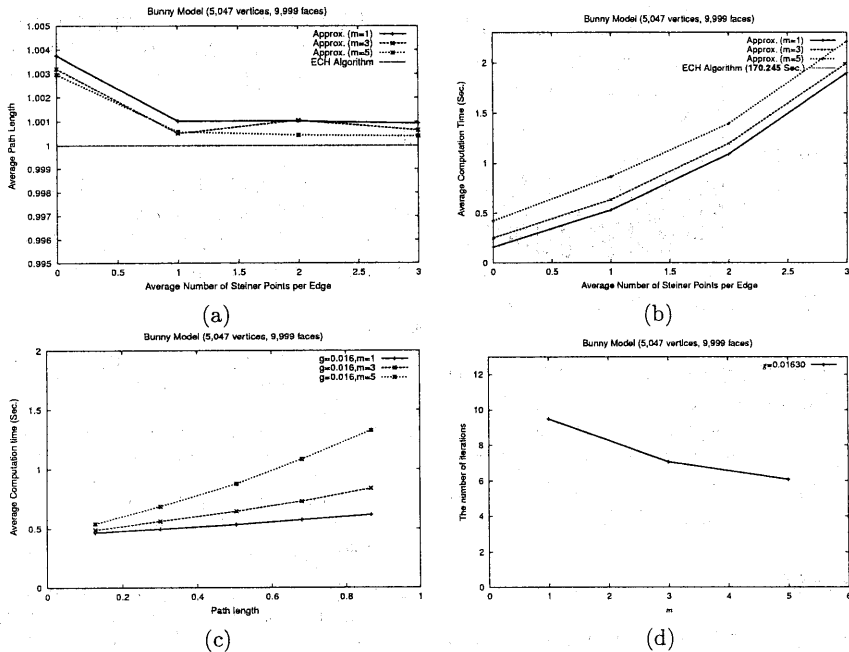


図 4: (a)-(b) 本アルゴリズムと ECH アルゴリズムの近似精度と計算時間での比較 (c) 本アルゴリズムの経路の長さとの関係 (d)  $m$  と収束回数との関係

面にしてから、経路の存在範囲を計算していくことを基本とする。そのため、面の回転計算による数値誤差が蓄積する。本アルゴリズムは単にエッジの長さの足し算を用いており、結果として、計算精度の良いアルゴリズムであることが実証されている。

## 4 応用

本節では、本アルゴリズムのモデリングへの二つの応用について議論する。一つは多面体の局所的な領域指定を実現するためのツールとしての利用であり、もう一つは、測地線距離の計測としての利用である。

### 4.1 多面体の領域指定

特にレンジデータから生成した密な多面体に対し、モデリングに必要なある一部の領域を指定するという操作は、モデリング、レンダリング、それにアニメーションのために必要である。例えば、テクスチャなどの属性の付加 [12] やパラメトリック曲面への変換 [7]、メッシュの局所的変形 [10, 8, 11]、モーフィング [5, 6] などが挙げられる。ここでは本アルゴリズムを、インタラクティブな領域の境界指定を実現するためのツールとして利用することを考える。上記のようなモデリングにおける領域の境界は、滑らかな線になっていることが求められる。境界が歪んでいると、応用によっては得られる結果の質に悪い影響を及ぼす。

さらに、このような操作はしばしばユーザによる試行錯誤の連続であるため、インタラクティブに処理できる高速かつ効率的な方法が求められる。

上記のような領域指定の操作において本アルゴリズムを利用する場合、まず前処理として  $G^0$  を計算する。この  $G^0$  は、領域の境界の入力・修正の間常に保持しておくことで、効率的な経路の計算が可能である。我々のアルゴリズムでは二点を指定することで一本の線を入力することが可能であるが、複数の経路により一本の線を構成することも可能である。すなわち、何点かの制御点を定義し、隣接する制御点間を最短経路により接続することで、一本の境界線を構成できる。図 6 に、馬のモデル (頂点数 19,851, 面数 39,698) の腹付近に星型の領域を指定した例を示す。図 6 の球は入力・修正した制御点を示している。各制御点の入力・修正につき、隣接する二つの経路の修正が必要であるが、これらの計算はいずれも高速であり、インタラクティブな操作が可能である。

### 4.2 測地線距離の計測

最短経路算出は、ロボティクス、GIS などの分野で重要な技術の一つである。これらの応用では、その経路や長さは正確であることが必要とされることが多い。そのような場合、正確な経路から少しでもかけ離れていると、応用での結果に悪い影響を及ぼす可能性がある。他方、近似解でも十分に満足な応用も確か

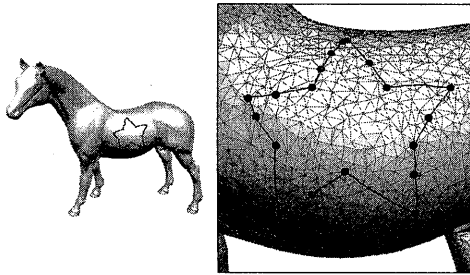


図 6: 星型領域の指定のための境界の生成

に存在する。例えば医用分野において、CT や MRI などの医療画像から得られた脳の三次元モデルの中で、脳腫瘍の大きさを迅速に測定したい、などという場合が想定できる。本近似アルゴリズムは、インタラクティブ性を必要とし、かつ、必ずしも正確な解が必要でない場合への応用には適していると考えられる。

## 5 結論と展望

本報告では、離散グラフの選択的詳細化による多面体上の近似最短経路算出法を提案した。また、近似精度、計算時間、それに計算精度の観点から本アルゴリズムと ECH アルゴリズムを比較した。さらに本アルゴリズムを用いた二つの応用について議論した。本手法は以下の利点を持つ：

- 実装が単純である。頂点とエッジにもとづくグラフ構造を繰り返し使用するだけである。
- 高速である。計算時間はダイクストラのアルゴリズムに依存するが、 $O(n \log n)$ での計算が可能である。最近 Thorup はダイクストラのアルゴリズムを  $O(n)$ で処理する方法を提案した [15] が、このようなアルゴリズムも検討課題である。
- 正確な経路に比べ、高い近似誤差を持つ。本文中の例では、正確な経路の長さに比べて 0.4% の近似誤差以内に収まっている。
- 単純なエッジの長さの和算により経路の長さが求まる。このため数値誤差の影響を受けにくいアルゴリズムである。
- 本文中では示さなかったが、重み付き最短経路の算出も可能である。

反面、以下の欠点も持つ。これらの課題の克服が今後の課題である。

- 規則的な多面体（例えば平面上の格子状に並んでいるポリゴン）に対する計算に弱い。これらの例ではより  $\gamma$  を小さくする必要がある。

- 計算時間が経路の長さに依存するため、同じ多面体でも一定時間での応答を期待できない。このことは、応用によっては大きな欠点となる。

## 謝辞

この研究の一部は(株)アルモニコスより支援を受けた。ここに感謝の意を表す。兎のモデルはスタンフォード大のグラフィックス研究室のデータを、馬のモデルは Cyberware Inc. のデータを使用した。

## 参考文献

- [1] AHO, A. V., HOPCROFT, J. E. and ULLMAN, J. D. *Data structures and algorithms*, Addison-Wesley (1983).
- [2] CHEN, J. and HAN, Y. Shortest Paths on a Polyhedron, Proc. 6th ACM Symp. on Computational Geometry (1990).
- [3] CURLESS, B. and LEVOY, M. A Volumetric Method for Building Complex Models from Range Images, Computer Graphics (Proc. SIGGRAPH 96) (1996).
- [4] GARLAND, M. and HECKBERT, P. S. Surface Simplification Using Quadric Error Metrics, Computer Graphics (Proc. SIGGRAPH 97) (1997).
- [5] KANAI, T., SUZUKI, H. and KIMURA, F. Metamorphosis of Arbitrary Triangular Meshes with User-Specified Correspondence, IEEE Computer Graphics and Applications, to appear.
- [6] KANAI, T., SUZUKI, H., MITANI, J. and KIMURA, F. Interactive Mesh Fusion Based on Local 3D Metamorphosis, Proc. Graphics Interface '99 (1999).
- [7] KRISHNAMURTHY, V. and LEVOY, M. Fitting Smooth Surfaces to Dense Polygon Meshes, Computer Graphics (Proc. SIGGRAPH 96) (1996).
- [8] KURIYAMA, S. and KANEKO, T. Discrete Parametrization for Deforming Arbitrary Meshes, Proc. Graphics Interface '99 (1999).
- [9] LANTHIER, M. A., MAHESHWARI, A. and SACK, J.-R. Approximating Weighted Shortest Paths on Polyhedral Surfaces, Proc. 13th ACM Symp. on Computational Geometry (1997).
- [10] LEE, A. W. F., SWELDENS, W., SCHRÖDER, P., COWSAR, L. and DOBKIN, D. MAPS: Multiresolution Adaptive Parameterization of Surfaces, Computer Graphics (Proc. SIGGRAPH 98) (1998).
- [11] LEE, S. Interactive Multiresolution Editing of Arbitrary Meshes, *Computer Graphics Forum (Eurographics 99)*, 18, 3 (1999), 73-82.
- [12] MAILLOT, J., YAHIA, H. and VERROUST, A. Interactive Texture Mapping, Computer Graphics (Proc. SIGGRAPH 93) (1993).
- [13] MATA, C. S. and MITCHELL, J. S. B. A New Algorithm for Computing Shortest Paths in Weighted Planar Subdivisions, Proc. 13th ACM Symp. on Computational Geometry (1997).
- [14] MITCHELL, J. S. B., MOUNT, D. M. and PAPPADIMITRIOU, C. H. The Discrete Geodesic Problem, *SIAM J. Computing*, 16, 4 (1987), 647-668.
- [15] THORUP, M. Undirected Single Source Shortest Paths in Linear Time, Proc. 38th Symp. on Foundations of Computer Sciences (1997).