

平成 13 年度未踏ソフトウェア創造事業
S-CODE: 細分割曲面理論による CAD データ
配信用コーディングシステム

S-CODE: A Coding System for Distribution of
CAD Data based on Subdivision Surface Scheme

鈴木 宏正¹⁾ 堀田 淳²⁾
Hiromasa Suzuki Jun Hotta

- 1) 東京大学大学院工学系研究科 精密機械工学専攻 (〒113-8656 東京都文京区本郷7丁目3番1号
E-mail: suzuki@cim.pe.u-tokyo.ac.jp)
- 2) 株式会社エリジオン (〒430-0927 静岡県浜松市旭町11番1号 プレスタワー 10階
E-mail: hotta@elysium.co.jp)

ABSTRACT. In digital manufacturing environment, CAD model must be transferred on the network. Usually CAD model is converted to polygon model. However, the polygon model is not a suitable way for distributing the CAD model on a network and also for rendering the model because of its large scale. We introduce a new system to convert CAD model to a subdivision surface model called S-CODE, which consists of S-Coder to accurately approximate and compress the CAD model, and S-Viewer to decompress and efficiently render it. The data in this system is not only compact and high accuracy with the original CAD data, but also very suitable for efficient rendering.

1 背景

もの作りの IT 化が進展する中で、様々な新しい設計ソリューションが利用されている。その中でも、ネットワークを利用して、地理的に分散している設計や製造、サプライヤーなどの技術者が、CAD データをリアルタイムに共有しながら、協調して作業を進められるシステムが注目されている。例えば、デジタルモックアップ (DMU) と呼ばれるシステムでは、自動車のエンジンルームのような膨大な数の部品が複雑に配置されたものに対して、各部品の 3 次元 CAD データを仮想空間に集積し、部品間の干渉や組立性などを、遠隔地の設計技術者と生産技術者が一緒に検討することができる。DMU では、複数の事業部や企業において相異なる CAD システムによって生成されたデータを扱う必要があるため、CAD データそのものを用いるのではなく、それを三角形分割 (tessellation) したポリゴンモデル (三角形メッシュ) を用いている。またこれは、CAD データの機密保持上から必要な場合もある。また、DMU に限らず、ネットワークを介して CAD モデルをビューイングすることは、製品ライフサイクルの多くの場面で要求されている。例えば、設計部門と製造部門との間の技術検討や、サプライヤーとの打ち合わせ、セールスやユーザ・サービス部門への情報提供などである。これは、CAD データを見るだけの「閲覧ユーザ」が、潜在的に多数いることを示している。しかし、閲覧の為だけに CAD システムそのものを利用させるのは、コスト的にも機能的にも過剰であり、また CAD データビューワのような表示専用システムも多数開発されているが、WEB で手軽に利用できるまでにはなっていない。また、機密保持上、社外のユーザがビューワで CAD ファイルを利用することは難しい場合も多い。

このように考えると、設計業務のための CAD データ

ベースに加えて、「CAD データを見るための仕組み」あるいは、「CAD モデルをブラウズするためのシステム」が必要であることが分かる。

2 目的

本事業の目的は、このような大多数の閲覧ユーザに対して、ネットワークを介して、CAD データの表示のための情報をリアルタイムに、効率良く配信するための環境を構築することである。そして、より圧縮率が高く形状品質の良い CAD データの新しいコーディング方式として、S-CODE システムを開発することである。

現在の DMU システムでは前述の通りポリゴンモデルを使用している (図 1 上)。しかし、次のような問題があり、本事業で目指す手軽な CAD データのビューイングは実現できていない。

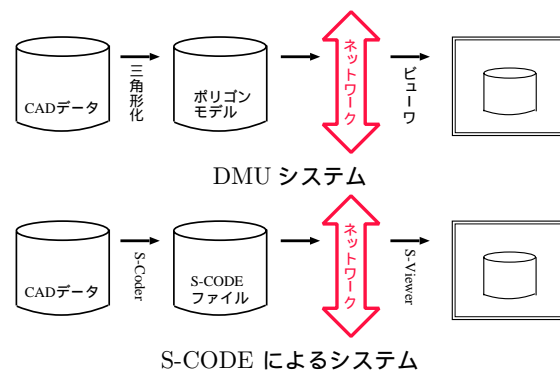


図 1: DMU システムと S-CODE によるシステム

- ポリゴンモデルの問題点

ポリゴンモデルはその名の如く多面体であり、元の CAD データの持つ曲面は近似されることになる。そのため、表示されるデータは、低品質で低精度のモデルとなり、現在の DMU では曲面品質などの評価を行うことはできない。

また、ある程度の曲面品質を保証するためには、曲面を微小な三角形に分割する必要があり、その結果、ポリゴンモデルのデータ量は膨大になる。このデータ量の問題はより深刻であり、現在、例えば DMU を行うには大型のグラフィックワークステーションが不可欠で、またネットワークを介した遠隔地での利用を考えると、高速なネットワークも必要になる。このような問題は、計算機技術やネットワーク技術の進歩によって近い将来解決できると期待されるが、理想的には、より多くの設計者が、特殊な装置を一切使わずに、手軽にデスクトップ PC の WEB ブラウザを使って CAD データを見ることが出来るような環境を想定すると、ポリゴンモデルのデータ量は、オーダー的に大きすぎると言わざるを得ない。

これに対して、メッシュのデータ圧縮技術の開発も進んでいる。例えば MPEG4 で採用されているような三角形メッシュの圧縮技術 [1] が考えられ、性質の良いメッシュ(きちんとした多面体、あるいは 2 多様体メッシュ)の場合には 1 三角形あたり数ビットまで圧縮することができる。しかし残念ながら、CAD システムから出力される三角形メッシュは、面と面の間にギャップが開いていたり、面が交差したりするような性質の悪いものがほとんどで、このような技術を適用することは難しい。そこで本研究では CAD モデルをポリゴンに変換するのではなく、もっとコンパクトで表現力のあるモデルに変換し、それを転送する。

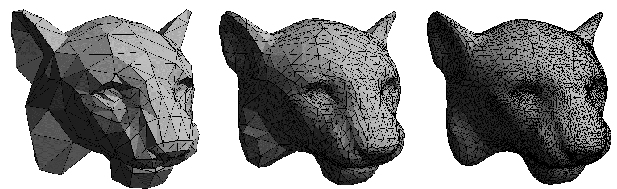
● 表示速度の問題点

一方、このような大規模なメッシュを表示することも問題である。三角形の数が増大すると、通常のデスクトップ PC やノート PC 等で表示することは困難である。これに対して最も効果的な方法は、必要な表示精度に応じて面の細かさを制御する LOD (Level of Detail、多重解像度) 法であろう。これは、例えば視点から遠方にある物体は粗いメッシュで表し、視点に近いメッシュは細かいメッシュで表すというもので、視点の変更に応じてメッシュの詳細度をダイナミックに変更するものである。三角形メッシュで LOD を実現する方法は数多くあるが、そのためには詳細度を制御するためデータ構造を生成する前処理が必要であり、膨大なメモリと計算を必要とする。

そこで本事業では、図 1 下のように、ポリゴン近似による方法を利用せず、別の表現方法を用いる。それは Subdivision Surface Modeling(細分割曲面モデリング) という形状理論でこれは上に挙げた二つの問題を解決する表現方法である。

細分割曲面は図 2 に示すように多面体を規則的に分割することを繰り返すことによって滑らかな曲面を生成する形状表現理論である [2],[3],[4],[5]。細分割曲面は、無限回の細分割によって極限曲面に収束する。その収束性は非常に良く、数回の細分割でも十分に滑らかな形状を生成することが出来る。更に、パラメトリック曲面に比べて、次のように優れた特徴がある。

- 曲面変形に対して連続性を保証可能: 制御点を変更しても、パッチ間の連続性(ほとんどの領域で C^2 連続)が保証される。
- 任意トポロジー: 四角形や三角形面だけでなく、位相幾何学的にはトーラス等と同様な曲面も表すことができる。



初期メッシュ 1 回細分割 2 回細分割

図 2: 細分割曲面

- LOD: 細分割の階数を制御することによって、詳細度の異なる多面体を生成できる。
- 連続部分も不連続部分も表現可能: 細分割処理を拡張することによって、ピン角やフィレットなども表現することができる。
- 処理の簡潔性: 細分割の計算は単純な線形計算で行うことができ、また、データ構造も規則的であるために、処理は非常に簡潔である。

3 手法の概要

本事業では図 1 に示した S-Coder と S-Viewer を開発した。

1) S-Coder

S-Coder の構成を図 3 に示す。まず、各種 CAD システムの CAD モデルに対応するために、CAD ファイルを変換して中間ファイルを生成する。そして、この中間ファイルから CAD モデルを近似する細分割曲面モデルを生成する。細分割曲面によって近似することによって、三角形メッシュで近似するよりもはるかに少ないデータ量でモデルを表現することができる。細分割曲面は、基本的には 3 角形面で構成されるポリゴンモデルとなるので、通常のポリゴン圧縮処理を適用し、更に圧縮を行う。これらの処理の結果を、S-Code ファイルとして出力する。

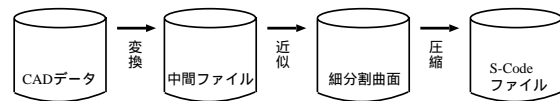


図 3: S-Coder の処理の流れ

2) S-Viewer

一方、このようにして生成された S-Code ファイルは、ネットワークを介して、ビューワへと転送される。図 4 に示すように、転送されるデータは、まず圧縮が伸張され、細分割曲面モデルとなる。このモデルは細分割曲面の初期メッシュであり、これを細分割することによって曲面(密な三角形メッシュ)を生成し、表示を行う。この際、LOD 処理として細分割のレベルや場所を調整するだけで、(付加的な処理やデータ無しでも)無駄なデータの生成を抑制し、表示効率を高めることができる。前節で示したように、細分割の計算は簡潔であり、プログラムもコンパクトなものとなる。

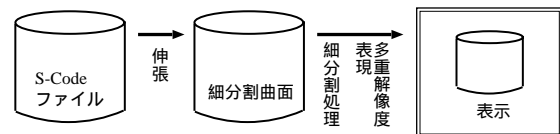


図 4: S-Viewer の処理の流れ

S-Coder の機能

● CAD ファイル変換

まず、DMU などでの利用を考えると、様々な CAD システムのデータを扱う必要が生じる。そのために本提案の母体である (株) エリジオン社のフレームワークによって、各種 CAD データを読み込み、標準的な境界表現のソリッド・サーフェスデータへ展開する。なお、この項目は開発項目に含まれていない。

● 細分割曲面近似機能

次に、このようにして読み込まれた CAD データから細分割曲面モデルを生成する。この機能は、本提案の重要な開発テーマであり独創的な部分である。これまでの細分割曲面はアニメーションキャラクターのモデリングに主として利用されており [4]、CAD で対象とする製品形状には十分に適用されていない。そのため、次のような機能を開発した。

- 解析曲面の近似変換機能: CAD データには、円筒面や円柱面などのような解析曲面が多数含まれている。これを細分割で近似する機能。
- パラメトリック曲面及びそのトリム曲面の変換: CAD データでは B-spline や NURBS 曲面などのようなパラメトリック曲面が利用されている。さらに、これらの曲面は、ある形に切り抜かれて (トリム) 利用されている。このようなパラメトリック曲面を細分割曲面に変換する機能

● 圧縮機能

以上により生成された細分割曲面の初期メッシュは、基本的にポリゴンモデルであることには変りがない。圧縮には、Edge Breaker アルゴリズム [6] を用いた。

S-Viewer の機能

● 伸張機能

S-Viewer は、すでに述べたように圧縮を伸張して初期メッシュを得る部分と、それを細分割して表示を行う部分からなる。伸張についても、Edge Breaker アルゴリズムに基づき実装した。

● 細分割処理によるメッシュ生成と LOD 機能

細分割曲面を表示するためには、初期メッシュを細分割して、小さい三角形から構成される三角形メッシュを作り出すことになる。幾何学的な計算は簡単であるが、単純に実装するとデータ量は非常に大きくなってしまふ。本事業では、細分割によって生成される三角形メッシュが規則的な配列をとることを利用して、データ量を削減する方式を開発した。また、このデータ構造を用いて、視点と形状との関係から、必要な部分だけを必要な詳細度で細分割する LOD を実現した。また、解析曲面に対しては、曲面の品質が重要となるため、細分割曲面の歪みを補正するようなメッシュの再構成アルゴリズムを適用した。

4 細分割曲面近似及び圧縮

4.1 三角形分割

まず中間ファイルに対して三角形分割を行う。この三角形メッシュは細分割曲面の初期メッシュを作るためのもので、大きな近似誤差をもつ粗めの分割が良い。ここでは、CAD モデルを構成する曲面パッチ単位で三角形分割を行い、今後はこれをメッシュ群 $M_{ti} \{i = 0, 1, \dots\}$ として、それぞれ個別に処理する。なお、以下のように三角形分割を行った。

- 三角形分割でできたメッシュの全ての頂点は元の曲面上に位置する。
- 隣接する曲面 M_{ti}, M_{tj} の境界曲線上にある頂点座標は全て同一である。

- 円筒面・円錐面・平面などの曲面は、境界曲線上とトリム曲線上にしか三角形分割によるメッシュの頂点はない。つまり曲面の内部には頂点が存在しない。
- 図 5 のように、 M_{ti} の粗さを近似距離 δ 、近似角度 θ で制御ができる。

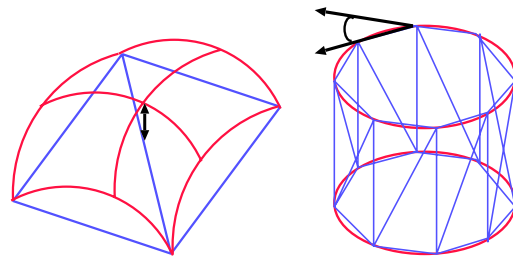


図 5: 近似距離、近似角度

4.2 細分割曲面近似

細分割規則

本事業では、扱う多角形が 3 角形で、高精度の近似を行うことを目的としているため (ほぼ C^2 連続)、1983 年に W. Boehm によって提案された Box スプライン曲面理論に基づく細分割方式 [2] を採用した。この方式は、さらに Loop によって一般化され、今日では Loop 細分割と呼ばれている [7]。Loop 細分割曲面は、図 6 のように、メッシュの各 3 角形を小 3 角形 4 つに分割することを繰り返すことで得られる曲面 (これを 4to1 細分割と呼ぶ) である。Loop 細分割の計算は 3 角形の各エッジに頂点を挿入し、3 角形を分割する位相分割ステップと細分割前のメッシュの頂点座標を移動する頂点移動ステップからなる。

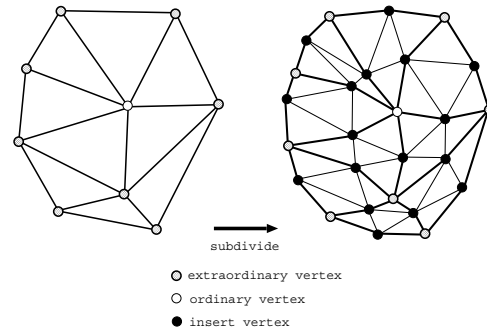


図 6: Loop 細分割

細分割の際、挿入する頂点及び頂点移動ステップの頂点の座標の算出にはそれぞれ周辺の頂点とのアフィン結合を用いるが、その重み付け (マスクと呼ぶ) を図 7 に示す (それぞれ左、右)。また、境界上の頂点はそれぞれ隣接する境界上の頂点のみを用いる (図 7 下)。

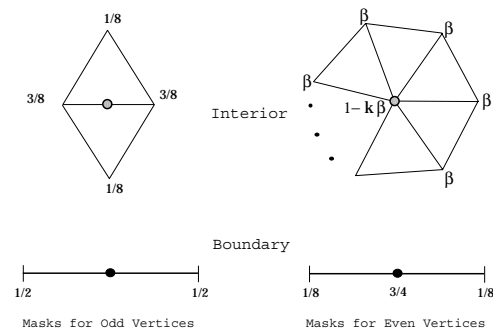


図 7: Loop 細分割のマスク

図中の k は頂点の価数 (頂点に接続しているエッジの数) を表し、 β は以下の式により求められる。

$$\beta = \begin{cases} \frac{3}{16} & (k=3) \\ \frac{1}{k} \left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4} \cos \frac{k}{2\pi} \right)^2 \right) & (k>3) \end{cases} \quad (1)$$

Loop 細分割手法によって生成される細分割曲面は正則頂点近傍 (Ordinary Vertex) (3 角形メッシュの場合には価数が 6 の頂点) では、 C^2 連続を満たし、それ以外の特異頂点近傍 (Extraordinary vertex) では C^1 連続を満たしている。

Loop 細分割手法では、一様に平滑化を行うため、鋭角領域を正確に表現することができない。そこで、接平面不連続な物体を正確にモデル化する新しい細分割規則が Hoppe 等によって提案された [8] (拡張 Loop 細分割曲面)。この細分割規則では、鋭角な領域を、crease (折り目)、corner (角)、dart (ダーツ) と呼ばれる特徴によって表現する。crease は、接線連続な曲線ではあるが、それに沿った曲面が C^1 ではなく C^0 であるような曲線である。corner は、3 本かそれ以上の crease が交差する点である。最後に dart は、crease の端に位置する曲面の内部点である。本事業では M_{ti} の内部に鋭角領域がないと仮定しているため、crease, dart は扱わず、CAD データの頂点を corner と指定する。そして、境界頂点に対して価数が 4 の場合かどうかで分類し、前者を regular vertex、後者を non-regular vertex とする。

細分割の際、内部の頂点における頂点座標は通常の Loop 細分割により求められる。境界頂点に対しては、頂点移動ステップの頂点座標の場合、corner であれば、頂点は移動せず、それ以外は Loop 細分割と同じである。挿入する境界頂点に対しては、両端の頂点のタイプの組合せにより決まる。

細分割極限点

初期メッシュに対し、拡張 Loop 細分割手法を繰り返し適用すると、制御メッシュ上の各頂点 v は極限曲面上の点へと収束していく。この点は細分割極限点 v^∞ と呼ばれ、この点は実際に細分割手法を無限回適用することなしに求めることができる。具体的には内部頂点に v における v^∞ は v と v に接続する頂点群 $v_j (j=0, 1, \dots)$ のアフィン結合によって決定される。

$$v^\infty = \alpha v + \chi \sum_{j=0}^{k-1} v_j \quad (2)$$

ここで式 (2) 中の χ は、接続している頂点の重みであり、 α 及び χ は細分割を行う際に用いられる β によって式 (3) のように表される。

$$\alpha = 1 - k\chi \\ \chi = \left(\frac{3}{8\beta} + k \right)^{-1} \quad (3)$$

細分割曲面近似

本事業では細分割極限点を用いた細分割曲面近似を行う [10],[11]。具体的には、細分割極限点がターゲットとなる曲面に近づくように細分割曲面の制御メッシュの各頂点座標位置を移動する。そのため、制御メッシュの各頂点に対してそれぞれターゲットとなる曲面上の対応する点が必要であり、近似を行うことにより、制御メッシュの各頂点における細分割極限点はこの曲面上の点に近づく。

近似処理において、入力複数のパラメトリック曲面 $P_i (i=0, 1, \dots)$ からなるボディと個々の曲面の三角形分割で得られた M_{ti} であり、出力は各パラメトリック曲面 P_i を近似する細分割曲面の制御メッシュ群 $M_{fi} (i=$

$0, 1, \dots)$ である。本手法では、まず各 P_i を三角形分割することによって生成された制御メッシュ M_{ti} に対してそれぞれ独立に近似を行う。近似は、境界上の頂点、内部頂点の順にそれぞれ適用する。そして、近似を行った制御メッシュ M_{fi} から得られる細分割曲面 S_i と P_i との誤差を測定する。誤差は、 M_{fi} の各頂点の細分割極限点と P_i 上の最近接点との距離を垂点計算によって計測する。

本手法で用いる細分割曲面近似は細分割極限点を用いたものであり、制御メッシュの頂点 v_i における細分割極限点 v_i^∞ を対応する入力パラメトリック曲面上の点 $SP(v_i)$ に近づけるように制御メッシュの各頂点座標を移動することで細分割曲面近似を実現する。具体的に図 8 を用いて説明すると、まず、制御メッシュ M_{ti} の各頂点は、初期状態では P_i 上に位置している (図 8(b))。そのため、 M_{ti} における各頂点 v_i の初期位置を $SP(v_i)$ とすることができ、また同時に、このとき M_{ti} の各頂点 v_i に対して、細分割曲面 S_i 上の細分割極限点 v_i^∞ を一意に求めることができる (図 8(c))。

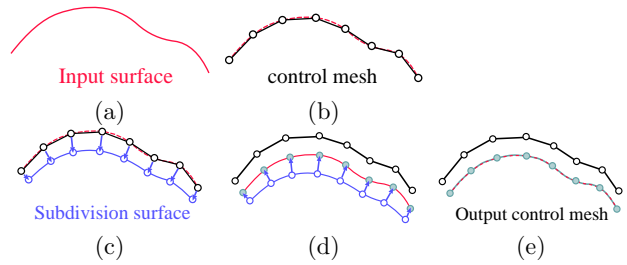


図 8: fitting

次に各 v_i^∞ が $SP(v_i)$ に移動するように v_i の各頂点座標位置を移動する (図 8(d))。各 v_i^∞ は v_i とそれに接続する頂点 v_j の荷重線形和で表されるので、条件を満たすためには $n \times n$ (n は頂点数) のマトリクスを解く必要がある (式 (4))。

$$\begin{cases} S & = Ax \\ V_i(New) & = V_i(Old) + x \end{cases} \quad (4)$$

ここで、 $v_i(New)$ は v_i の新しい頂点座標位置、 $v_i(Old)$ は v_i は v_i のもとの頂点座標位置をそれぞれ表しており、 S, A はそれぞれ以下の式 (5), (6) で表される。

$$S[i] = v_i^\infty - SP(v_i) \quad (5)$$

$$A[i][j] = \begin{cases} \alpha & \text{if}(i=j) \\ \chi & \text{else if}(edge(v_i, v_j)) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

ここで、 α および χ は細分割極限点の計算の際の重みに対応しており、マトリクス A の大きさは $n \times n$ であるため、式 (4) は非常に大規模な計算となるが、マトリクスの大部分が 0 であるので、実際はガウスザイデル法を用いる事で高速に解くことが可能である。

4.3 メッシュ圧縮

CAD データから細分割曲面に近似されたデータを、ネットワークを介してより高速に表示するためには、近似して得られた細分割初期メッシュ M_{fi} を更に圧縮する必要がある。具体的には、Edge breaker [6] を用いてまずメッシュデータの幾何形状を圧縮し、その後、zlib [9] を用いる事でさらなる圧縮を行う。Edge breaker は、隣接するエッジをたどり、一つずつ三角形を除去していくことによって三角形メッシュを Strip 化する手法であり、その際の取り除かれる三角形とメッシュとの関係をオペレーションコードとして生成し、その情報によって、接続関係を自動的に生成して圧縮を行う。そのため、Edge breaker は

座標値圧縮手法とは独立した幾何形状の圧縮手法であり、他の座標値圧縮手法と併用する事が可能であるため、今回は Edge breaker を用いた圧縮のあとに zlib を用いた圧縮を適用する。

本手法では、Edge breaker を、近似により得られた初期メッシュ群 M_{fi} それぞれに対して適用した。その際、Edge breaker は閉じたメッシュを対象としているため、dummy vertex により一時的に閉じた形状に変換し、そして伸張の際にもとのメッシュ M_{fi} に復元できる処理を行った。

Edge breaker により得られた圧縮データに対して、曲面の属性 (円筒面、円錐面、自由曲面など) や corner となる頂点 ID、また円筒面の軸方向ベクトルや円錐面の頂点座標などを付加した。これらは、後述する曲面の属性に応じた細分割に用いる。

5 細分割処理によるメッシュ生成と LOD 機能

S-CODE ファイルに対して伸張処理を行う事で復元された M_{fi} のそれぞれに対し、細分割処理を行う。その際、4to1 細分割を平面・円筒面・円錐面に適用すると、図9のようにメッシュの内部に歪みが生じてしまう。また、そのような頂点により必要以上の三角形を描画する事になるので、表示効率が悪くなる。

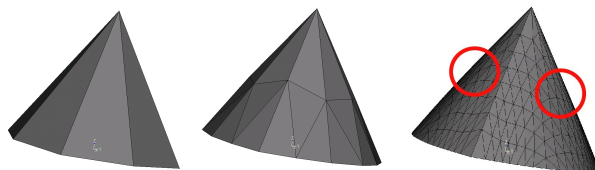


図 9: メッシュの内部の歪み

4.1 節で述べた通り、平面・円筒面・円錐面の制御メッシュの頂点は曲面パッチの境界上とトリム曲線上 (以後、これを特に境界と呼ぶ) にしか存在しない。以上より、表示精度と表示効率の改善のため、平面・円筒面・円錐面に対しては 4to1 細分割とは別の細分割を行う。今後平面・円筒面・円錐面を解析曲面とし、それ以外を自由曲面として、それぞれに対し、細分割・LOD を行った。なお、表示においては全て細分割極限点を用いた。

5.1 自由曲面の細分割と LOD

細分割

自由曲面の細分割については拡張 Loop 細分割手法 [8] による 4to1 細分割を用いた。4to1 細分割は次のような規則性を持つ。

- n 回細分割を行った時の頂点数、エッジ数、三角形数をそれぞれ $V(n), E(n), F(n)$ とすると、

$$\begin{aligned} V(n+1) &= V(n) + E(n) \\ E(n+1) &= 2E(n) + 3F(n) \\ F(n+1) &= 4F(n) \end{aligned} \quad (7)$$

- 細分割の際に新しく挿入される頂点の価数は、境界上にある場合は 4 であり、内部にある場合は 6 である。また、もとの頂点の価数は細分割を行ったも変わらない。

これらの規則性から、細分割後の頂点、エッジ、三角形の ID づけを規則的に割り当てる事で高速に細分割を行えるようにした。

LOD

規則的な ID づけにより LOD の際に、細分割処理の途中のメッシュの情報を保持する必要がない。つまり初期メッシュ M^0 から n 回細分割を行ったメッシュ M^n から m

回 ($m \leq n$) 細分割のメッシュ M^m の情報を容易に取得する事が可能となる。本 LOD 手法では、これを利用して初期メッシュ M^0 の 1 三角形単位で行う事とした。その手順は次の通りである。

- 1) M^0 に対し、最大回数 (n 回) 細分割を行う。この時、 M^0 と M^n のみを保持する
- 2) M^0 の三角形それぞれに対し、クリッピングを行い、クリップされていない三角形を 2 次元投影し、その大きさを求め、その三角形における表示レベル (m) を決定する
- 3) m に応じた三角形群を M^n から取得し、表示する

さらに、 M^0 の隣接する三角形間のレベルの違いにより生じる隙間については、レベルの高い方の三角形の頂点をレベルの低い方の三角形に合わせるよう、移動する頂点を予め求め、そのような場合に高速に取得を行えるようにした。図 10 に自由曲面の細分割と LOD の様子を示す。

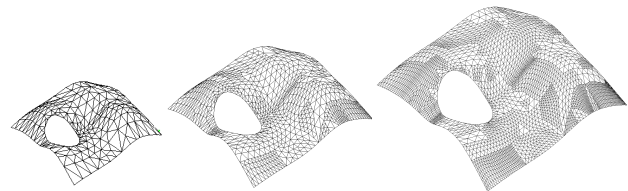


図 10: 自由曲面の細分割及び LOD の様子

5.2 解析曲面の細分割と LOD

細分割

解析曲面は面内に頂点を生成すると曲面が歪んでしまうので、境界上のエッジだけを分割することにした。制御メッシュの個々の三角形に対して分割するエッジの数により種類分けを行い、それぞれ別の方法で細分割を行った。細分割後の頂点座標は拡張 Loop 細分割手法 [8] により算出した。

平面やトリム曲線を含まない円錐面についての細分割はこの方法の適用で良いが、円筒面や円錐台、トリムを含む円錐面は図 11 のように、この方法でも形状が波うってしまう。そこで、これらについては細分割後にできるだけ稜線が軸方向を向くように Edge swap を行った (図 12)。

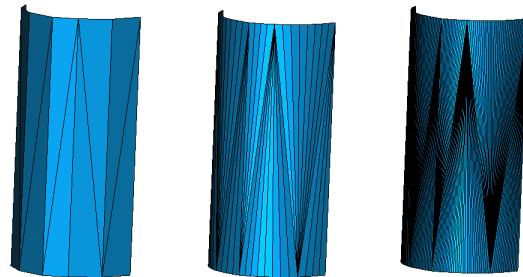


図 11: 内部に生じる歪み (円筒面)

具体的には次の 3 種類の場合に対して、Edge swap を適用した。なお、なるべく全ての内部エッジが軸方向を向くようにするため、Edge swap の評価には円筒面の場合は軸方向ベクトル、円錐面の場合はその頂点座標をそれぞれ用いた。

- 細分割の際に変更されない内部のエッジに対する Edge swap
- 一つの頂点にエッジが集中して集まる事を防ぐため、一つの頂点に均等にエッジが集まるように分配するための Edge swap

- 細分割の際に変更されない、分割エッジの数が0の三角形をなるべく軸方向に、かつ面積が小さくなるようにするための Edge swap

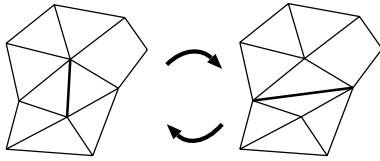


図 12: Edge swap

LOD については、非一様細分割を行うので、上で述べたデータ構造を用いることができない。また、自由曲面と同じ LOD を行うと、解析曲面の形状を正確に表現することができない。そのため、曲面バッチ単位で行うことにした。ただし、表示速度を向上するために、Triangle Strip を適用した。図 13 に円筒面と平面の細分割及び LOD の様子を示す。

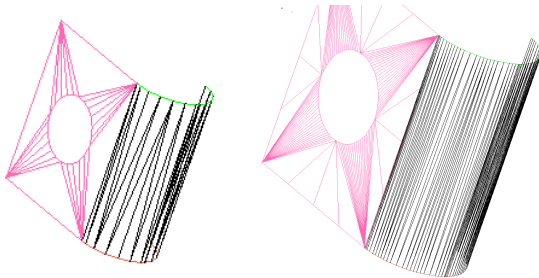


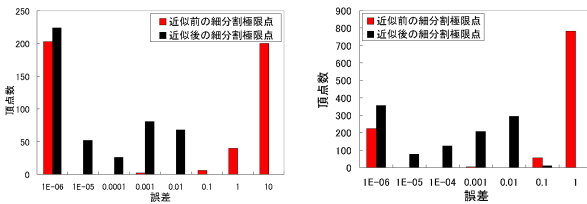
図 13: 円筒面と平面の細分割及び LOD の様子

6 考察

様々な CAD モデルに対し本手法を適用した。その結果を図 15 に示す。これらのモデルに対し近似の誤差、圧縮、形状や表示の品質について評価を行った。

6.1 誤差評価

本手法によって近似、生成された細分割曲面が、入力パラメトリック曲面に対してどの程度の誤差を持つか調べた。誤差の測定としては、まず近似前の制御メッシュ $M_{ti}(i = 0, 1, \dots)$ の各頂点における細分割極限点と入力曲面 $P_i(i = 0, 1, \dots)$ までの距離を測定し、次に近似後の制御メッシュ M_{fi} の各頂点における細分割極限点と P_i までの距離を測定した。図 14 にモデル Easy1 に対して、異なる粗さで三角形分割した M_{ti} の、近似前と近似後の誤差を示す (Easy1 は一辺が 10^2 オーダの大きさ)。



$\delta = 5.0, \theta = 24$ での近似 $\delta = 0.5, \theta = 18$ での近似

図 14: 近似誤差評価

この結果から、近似によって細分割曲面 (正確には細分割極限点) が入力曲面に十分近づいていることがわかる。

しかし、これは初期メッシュの各頂点における細分割極限点についてであって、細分割後に新たに挿入される頂点における細分割極限点 (つまり初期メッシュの極限点以外の領域) に対しての精度は保証していない。しかし、それ

らの極限点についても調べたところ、上で得た精度よりは良くないが、十分な高精度の結果となった。

6.2 圧縮の評価

本手法でモデルのデータがどの程度圧縮されているか評価を行った。評価は、入力 CAD データ (IGES データ) に対して本手法を適用することで得られる S-CODE ファイルを STL ファイル (これは $\delta = 0.1, \theta = 5$ で三角形分割したポリゴンデータ) と比較することによって行う。表 1 にそれらのファイルのサイズを示す。

モデル名	IGES	STL	S-CODE
Easy1 (0.5, 18)			9.53
Easy1 (5.0, 24)	130	1337	4.33
Tractor	13,899	85,746	474
Sports Car	11,221	73,414	256

表 1: 各モデルに対するサイズの比較 (単位は KB)

この表から本手法 (S-CODE) が、高い圧縮率を得ていることが分かる。

6.3 表示の品質についての評価

本手法により、自由曲面に対しては、高速な細分割処理や適用的な三角形の取得による高速かつレンダリングの質の高い LOD を行った。しかし、欠点として、隣接する三角形間のレベルの違いにより生じる隙間を埋めるために、移動する頂点全てを求めているため、多少時間を要してしまう。そして、隙間を埋めるために求めた頂点と、LOD の際に高速にマッピングを行うための情報を持つので、その分だけメモリの消費量が多くなってしまふ。

解析曲面に対しては、細分割に必要な領域に対してのみ (すなわち境界部分)、細分割を行うことで、よりコンパクトな表現が可能となり、またメッシュの再構築を行うことで、より整列したメッシュを生成することを可能とした。ただ、元の曲面が複雑にトリムされているような形状の場合は、本手法ではうまくいかない場合がある。

7 まとめ

本事業では、細分割曲面理論による CAD データ配信用コーディングシステム S-CODE の開発を行った。その機能として、S-Coder では細分割曲面近似機能、圧縮機能を、S-Viewer では伸張機能と、細分割処理によるメッシュ生成と LOD 機能を開発した。その際、解析曲面の細分割アルゴリズムを工夫することによって、円筒面や円錐面の歪みを極力低減できるようにし、また、大規模なメッシュを効率良く表現する内部データ構造の採用により、例題においてはストレスのない表示速度も実現できた。

また、CAD データに対する圧縮率に関しては、一応のレベルに達していると思われるが、さらに多くのデータに適用して評価・改良を行う必要がある。それは、圧縮の対象となる近似曲面のデータ量が近似精度と共に大きくなるために、トータルの圧縮率は、圧縮アルゴリズムそのものよりも、細分割曲面の近似方法によって影響を受けるからである。

その他に以下のような課題を残す。

- CAD モデルを構成する個々の曲面単位で扱っているため、曲面間で微小な隙間が生じてしまう。この隙間は、近似、細分割処理の際に生じてしまうもので、この隙間を埋める必要がある。
- 曲面単位で圧縮を行っている為、重複頂点などの無駄なデータを持つ。ボディ単位での圧縮により、より圧縮率を上げる必要がある。
- 表示の際、初期メッシュの 1 三角形単位で適用的な改良を行っているが、初期メッシュの三角形数が膨大になるとコストが過剰になるので、その改善が必要であ

る。また、より効率のよい表示のためにカリングを行う必要がある。

- 本事業ではきちんとした多面体か、あるいは2多様体のメッシュを対象としていたが、より頑強性の向上のために非多様体に対しても適用できるようにする必要がある。

事業化への展望

S-CODE ソフトウェア単体でも小規模な CAD データの閲覧サービスに利用できるが、それだけではユーザにとって利便性が低い。S-CODE を核とした次のようなアプリケーションやサービスを計画している。

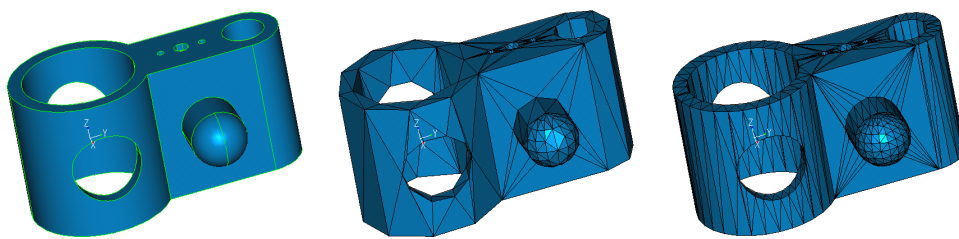
- VisDB(Visualization Data Base) システム
本開発に引き続き、S-CODE を核として、より製造支援に特化したものとして、PDM と S-CODE を組み合わせた VisDB の開発を計画している。多くの企業において、CAD データは PDM システムによって管理されており、ユーザは PDM を介して CAD データにアクセスすることができるようになっている。そこで、S-CODE を CAD データを閲覧するアプリとして位置付け、一方、ユーザに対しては WEB ブラウザ用の S-Viewer プラグインを開発して(無料で)提供する。こうすることによって、企業の情報インフラのなかに S-CODE をプラグインするだけで、広く CAD データの閲覧が可能になる。
- ASP (Application Service Provider) サービス
多くのユーザにデータを配信するためには、自前でサーバなどを管理する必要がある。ここでは、S-CODE システムを組み込んだサーバを立ち上げ、ユーザが CAD ファイルを転送すると、それを S-CODE ファイルに変換し、それを WEB サーバに登録し、その URL をユーザに知らせると言うビジネスモデルを考えている。これによってユーザは、CAD データを閲覧したいというユーザに、その URL を知らせるだけで、簡単に配信することができる。

参考文献

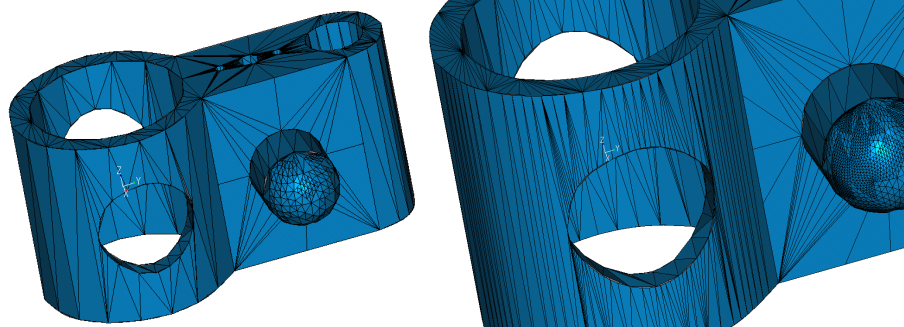
- [1] Gabriel Taubin, Andre Guezic, William Horn and Francis Lazarus. Progressive Forest Split Compression. *In Computer Graphics Proc. SIGGRAPH 98*. pp. 123-132, 1998.
- [2] Wolfgang Boehm: Subdivision multivariate splines, *Journal of Computer Aided Design*. Butterworth & Co Publishers, Ltd. Vol. 15, No. 6, Nov. 1983.
- [3] E. Catmull, J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *In Computer Aided Design*. pp. 350-355, 1978
- [4] Tony DeRose, Michael Kass, and Tien Truong. Subdivision Surface in character animation. *In Computer Graphics(Proc. SIGGRAPH 98)*.
- [5] D.Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. *In Computer Aided Design*, pp. 157-165, 1978.
- [6] William Horn Francis Lazarus J. Rossignac, Andre Guezic. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 1, pp. 47-61, January 1999.
- [7] Charles Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, University of Utah, August 1987.
- [8] Hugues Hoppe, Tony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John McDonald, Jean

Schweizer, and Werner Stuetzle. Piecewise smooth surface reconstruction. *In Computer Graphics (Proc. SIGGRAPH 94)*, pp. 295-302, 1994.

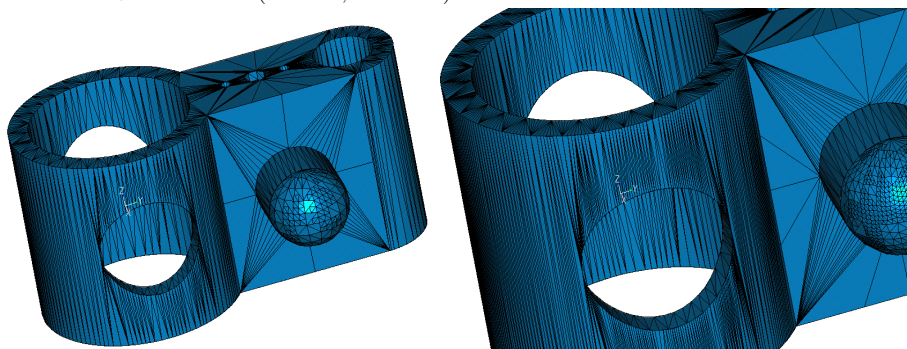
- [9] Mark Nelson, Jean-loup Gailly The Data Compression Book *M & T Books*, 1995
- [10] Hiromasa Suzuki, Shingo Takeuchi, Takashi Kanai, and Fumihiko Kimura. Subdivision surface fitting to a range of points. *In Proc. 7th Pacific Graphics International Conference (Pacific Graphics '99)*, pp. 158-167, 1999.
- [11] Shingo Takeuchi, Takashi Kanai, Hiromasa Suzuki, Kenji Shimada, and Fumihiko Kimura. Subdivision surface fitting with qem-based mesh simplification and reconstruction of approximated b-spline surfaces. *In Proc. 8th Pacific Graphics International Conference (Pacific Graphics '00)*, pp. 202-212, October 2000.



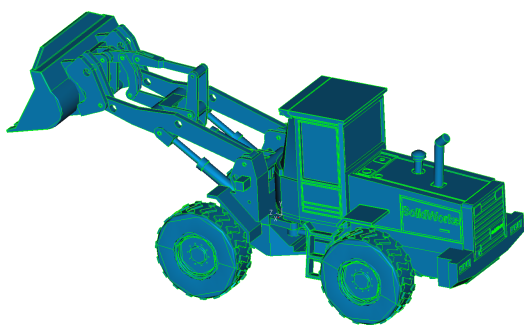
CAD データ (Easy1) 初期メッシュ ($\delta = 5.0, \theta = 24^\circ$) 初期メッシュ ($\delta = 0.5, \theta = 18^\circ$)



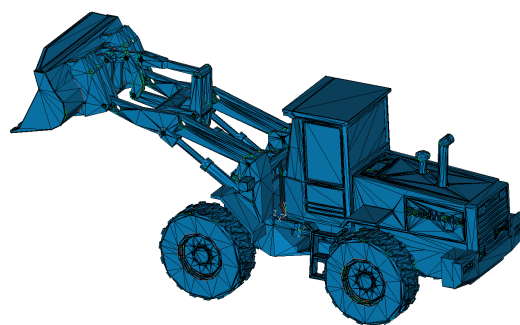
初期メッシュ ($\delta = 5.0, \theta = 24^\circ$) に対する細分割及び LOD の様子



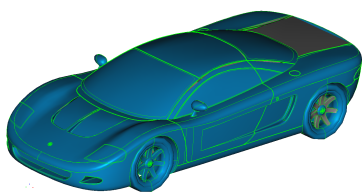
初期メッシュ ($\delta = 0.5, \theta = 18^\circ$) に対する細分割及び LOD の様子



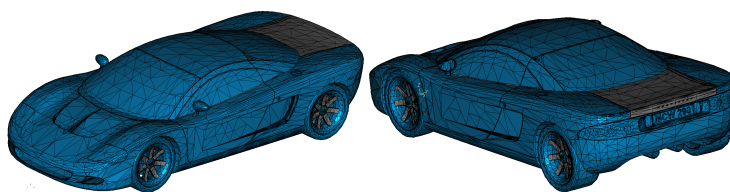
CAD データ (トラクター)



$\delta = 1, \theta = 20^\circ$ で近似、その細分割及び LOD の様子



CAD データ (スポーツカー)



$\delta = 1, \theta = 20^\circ$ で近似、その細分割及び LOD の様子

図 15: モデル Easy1、トラクター、スポーツカーに対する適用例