

# ストリーミングメッシュを用いた 大規模点群からの平滑化メッシュ生成手法

東京大学 ○村上健治, 増田 宏

Generation of Smooth Surfaces from Large-Scale Point-Clouds using Streaming Mesh  
The University of Tokyo: Hiroshi Masuda, Kenji Murakami

As-built 3D models are useful for simulating maintenance processes of manufacturing facilities. However, 3D scanners for large facilities produce very noisy point-clouds with a lot of outliers. In this paper, we introduce a method for stably eliminating outliers and smoothing very noisy mesh models using streaming meshes.

## 1. はじめに

生産設備などでは、度重なる改修や更新によって現状が図面やCADと異なっていることが多いため、設備の保守や点検を行うに当たっては、現物の測定によるアズビルトモデリングが有効である。近年開発された位相差方式レーザスキャナにより、短時間で大量の3次元形状測定が可能となったが、測定点群の膨大なデータ量、大きなノイズなどに対する処理が課題となっている。

本研究では、位相差方式レーザスキャナによる測定点群が持つ空間的コヒーレンスを利用した処理を行う。メッシュデータの記述にストリーミングメッシュを用い、点群とメッシュに対してストリーミング処理を行うことによって、大規模データでもメモリ効率と計算効率の優れた処理ができることを示す。

## 2. ストリーミング処理

### 2. 1. 計測点群のコヒーレンス

位相差方式レーザスキャナは、緯度と経度の2方向に沿ってレーザ光を回転させながら距離の計測と記録を行う。そのため、ファイル中での計測点の並び順と空間内の計測点の並び順の間にコヒーレンスが存在する。計測点群を図1に示したように経度と緯度をx, y軸にとって平面に展開すると、平面画像での上部から下部に向かって、計測点群が列ごとにほぼ整列して記録されている。

ただし、この順序関係は厳密なものではなく、計測点の欠落や誤差の補正により、空間内の順序とファイル中の順序にずれが発生するため、データの順序だけでは高品質なメッシュは生成できない。本研究では、ファイル内の順序に従って図2に示すようにある幅を持った帯状のデータを取り出し、ドローネー分割を用いて積層式にメッシュ生成していく。実メモリには、図2のように3本の帯状データだけを保持し、帯状データのメモリを逐次開放していく。生成されたメッシュはファイルに格納する。

このように、ファイルの先頭から順次読み込んで処理していく操作は、ランダムアクセスに比べて処理時間が著しく短くて済む。こうした処理をここではストリーミング処理と呼ぶ。

### 2. 2. ストリーミングメッシュフォーマット

メッシュフォーマットであるvml, offなどにおいては、初めに全ての頂点座標値(x, y, z)が記述され、これらを全て読み込んだ後

に、三角形を構成する頂点インデックス(i, j, k)により接続関係が指定される。しかし、大規模なメッシュモデルにおいては、頂点座標をすべてメモリ上に展開するのはメモリ効率が悪い。



図1. ファイル内でのデータの並び順

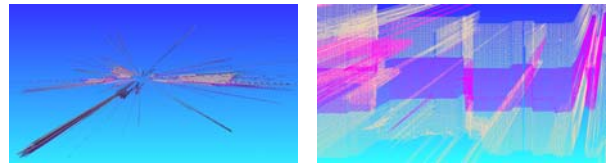


図2. コヒーレンスを利用したメッシュ生成 (右: 拡大図)

そこで、本研究では、図3に示すようなストリーミングメッシュフォーマット(以後、SMFと略す)でメッシュデータを記述し、逐次処理が行えるようにする。この形式では、頂点座標と頂点インデックスは混在して記述されており、ファイルの読み込み順に逐次形状を復元することができる。また、頂点は以後に参照されなくなる場合には頂点インデックスを負の値にして、メモリ解放のタイミングが明示的に示される(これをfinalization tagと呼ぶ)。SMFを利用することにより、点群やメッシュに関する処理をストリーミング形式で行うことを考える。

```
v(1.0, 0.0, 0.0)
v(1.0, 2.0, 1.0)
v(1.5, 2.0, 1.4)
f(1, 2, 3)
v(1.5, 2.0, 1.4)
f(1, 2, 4)
f(-1, 3, 4)
```

図3. ストリーミングメッシュフォーマットの例

### 2. 3. ストリーミングメッシュにおける近傍処理

大規模メッシュの平滑化をストリーミング処理で行う際、近傍頂点の情報が必要となる。特に、位相差方式の装置で取得された点群は誤差が大きく、我々の実験では、十分な平滑化には100~200個の近傍頂点を要した。そのため、近傍探索で利用される可能性のある頂点は、finalization tagの到着後も保持されている必要がある。

そこでSMFでの逐次復元においては、バッファ領域を設けて、メモリの開放を遅延させる。バッファの大きさは、図1の横一列の頂点数  $W$  に依存し、頂点の処理に  $N$  個の近傍を必要とするときは、概ね、前後にそれぞれ  $W\sqrt{N}/2$  個の頂点が保持されていければよい。ここでは余裕を持たせるために適当な  $\alpha$  を定め、 $W(\sqrt{N} + \alpha)/2$  個の頂点を保持した。SMFメッシュでの近傍処理は以下のように行う。

- (1)  $W(\sqrt{N} + \alpha)/2$  個の頂点を格納できるバッファを用意する。
- (2) 頂点を受け取ったら、頂点リストに格納する。
- (3) finalized tag が来たらその頂点をバッファに格納する。
- (4) 頂点リストにおいて、処理すべき頂点の前後の頂点数が  $W(\sqrt{N} + \alpha)/2$  個を超えたら、頂点リストからデータを順次取り出して近傍抽出を含む処理を行う。

- (5) バッファ内のデータが  $W(\sqrt{N} + \alpha)/2$  個を超えたら、先頭データのメモリを開放する。また、頂点リストからも削除する。
- この手順によって、平滑化処理に限らず、近傍探索に必要な頂点数を常に保持しながら、out-of-coreな環境で省メモリに形状処理が行える。さらに、データはHDDより重複せずただ一度のみ読み出されるため効率的である。

### 2. 4. パイプライン処理

ストリーミング処理が可能なプログラムでは、図4に示すように、パイプラインで繋ぐことにより、大規模メッシュ処理を限られたメモリで行うことができる。処理の途中結果をファイルに格納することなく、ファイル入出力はパイプラインの入り口と出口だけで行う。ここでは、異常値除去とメッシュ平滑化の機能を、それぞれ5~10個のコマンドをパイプラインとして接続することで実装している。コマンドには、点群からのメッシュ生成、フォーマット変換、フィルタ処理などが含まれる。

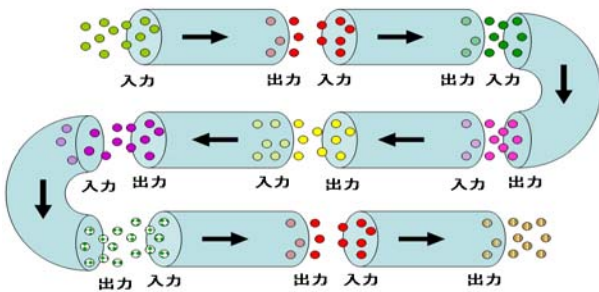


図4. パイプライン処理

### 3. 実験

平滑化処理を、これまでに述べたストリーミング処理を用いて実装し、約5000万頂点のファイルに対して、一括処理を行った。図5は結果の例である。上段の図は、全体のデータを示している。この点群を処理することにより、約3000万ポリゴンの平滑化メッシュが生成された。中段の図は、上段の四角で示された一部を切り出して

表示したものである。下段の図は、中段の図の一部をさらに拡大したものである。この図の示す通り、ストリーミング処理を用いることで、詳細度の高いメッシュを限られたメモリを用いて作成することが可能である。この処理に要した計算時間を2.40GHzのQuad Core CPUを用いて計算時間を計測したところ、点群を入力してから平滑化メッシュの生成まで、CPU Timeで約13分であった。マルチコアによる並列処理を用いているため、実験ではCore数にほぼ反比例して計算時間が短縮できている。

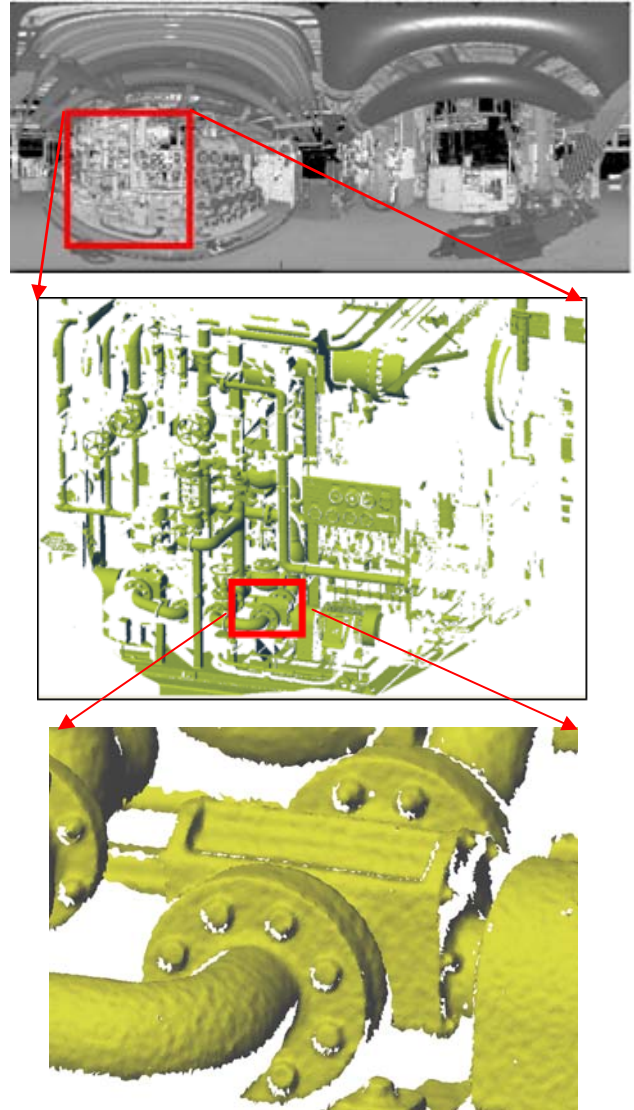


図5. ストリーミング処理による平滑化メッシュの生成

### 4. まとめ

本研究では、計測点群のもつコヒーレンスを利用し、ストリーミング処理を行うことで大規模点群を一括処理で平滑化メッシュに変換する方法を示した。

今後は、メッシュ簡略化やセグメンテーションなどの各種処理を、パイプライン処理として実装していくことを考えている。

### 参考文献

- [1] 増田, 村上, 藤井: 既設プラントモデリングのための点群データ平滑化と曲面生成, 2007年度精密工学会秋季大会.
- [2] M. Isenburg, P. Lindstrom: *Streaming mesh*, Visualization'05, pp. 231-238, 2005.