

ウェーブレット変換と4分木を利用した 大規模3次元地形モデルの動的表示のための詳細度制御法

北海道大学大学院工学研究科 ○橋田 祐樹, 金井 理, 岸浪 建史

要旨

大規模な3次元地形モデルを安価なPC上で動的にワークスルー表示するため、ウェーブレット変換及び4分木を利用した詳細度制御手法を新たに提案し、実際の地形モデル表示に適用し、その有効性を確認した。

1. はじめに

近年、DEM やレーザープロファイラ測定データなどの大規模3次元地形モデルデータが容易に入手可能となり、通常のPCプラットフォームでそれらのデータに対する動的ワークスルー表示を行いたいといった要求が高まっている。しかし、それらのデータは一般に莫大なポリゴン量となり、そのままではPC上で動的表示が行えないという問題点がある。

そこで、本研究ではPC上でDEMデータに対する動的表示を可能とする詳細度制御手法の開発を目的とする。類似研究としては[1]があるが、本研究はDEMデータのウェーブレット変換と4分木を用いた階層データ構造を活用し、地形・視点およびポリゴン総数に依存した詳細度(LOD)制御を同時に行えること、また差分データ管理手法による表示の高速化を実現している点に特徴がある。

2. システムの機能概要

本研究で提案するシステム概要を図1に示す。まず、DEMデータを入力し、初期4分木を生成する。初期4分木は各ノードが解像度の異なる三角形に対応している。また、三角形の各頂点の高さ値に対しLazyウェーブレット変換が行われる。次にユーザの指定する視野錐台情報および目標表示ポリゴン数に応じて、地形依存・視点依存・表示ポリゴン総数LOD制御を行い、表示三角形集合であるfrontを更新する。そしてfrontに対し差分データ管理手法を適用し、LOD制御されたDEMデータの表示を実時間で行う。

3. DEMのウェーブレット変換と4分木生成

3.1 ウェーブレット変換

DEMは、地表面上の標高を2次元の規則格子点上の高さ値配列として表したデータである。

本研究では、DEMの多重解像度表現を行うためにウェーブレット変換を用いる。ウェーブレット変換を用いるとオリジナルデータは、低解像度の近似と、ウェーブレット係数により表現される詳細部とに分解される。本研究ではLazyウェーブレットを用いた変換を行う[2]。ここで、 j をメッシュの解像度レベル、 $Z^j(m,n)$ と $w^j(m,n)$ をそれぞれレベル j のメッシュにおける位置が (m,n) である格子点の高さ値とウェーブレット係数とすると、DEMデータに対するレベル $j+1$ から j へのLazyウェーブレット変換は図2のように式(1),(2)によって決定される。

$$Z^j(m,n) = Z^{j+1}(m,n) \tag{1}$$

$$w^j(m,n+I^{j+1}) = Z^{j+1}(m,n+I^{j+1}) - \{Z^j(m,n) + Z^j(m,n+2I^{j+1})\} / 2 \tag{2}$$

ここで $I^j = 2^{j-1}$ 、 J は最高解像度レベルである。変換後に残る頂点には式(1)を適用し、残らない頂点には式(2)を適用して、ウェーブレット係数と置き換える。

3.2 4分木生成

DEMのウェーブレット変換データを統一かつ階層的に管理するために、4分木構造を導入する。この木の

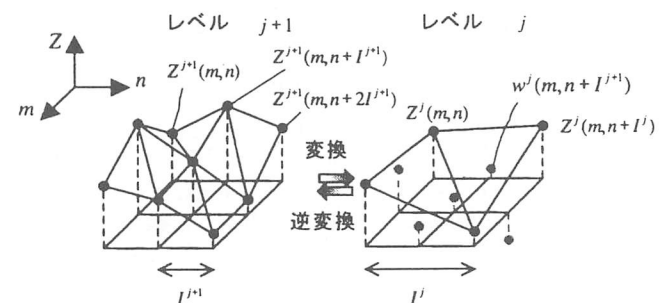
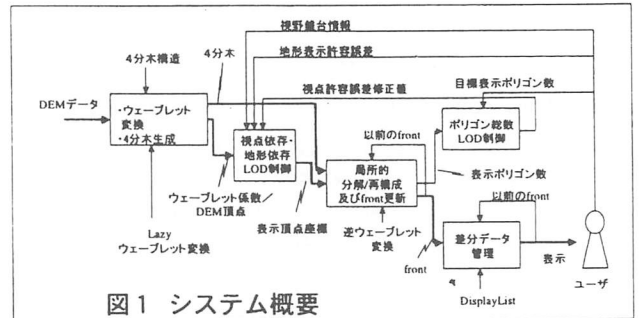


図2 DEMデータのLazyウェーブレット変換

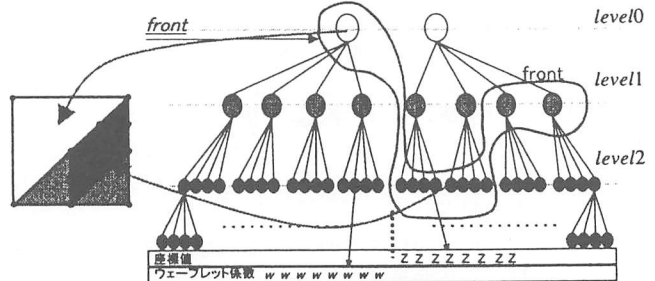


図3 4分木構造

レベル j のノードは、図3のように解像度レベル j のメッシュの三角形に対応している。また、このノードは三角形の各頂点もしくはウェーブレット係数へのポインタをもつ。

まず最高解像度レベル J の三角形を4分木に登録する。次にレベル $J-1$ の三角形を、その三角形に含まれるレベル J の4つの三角形を子として、木に登録する。この時、子頂点のうち親頂点ではない点においては式(2)の変換を行い、ウェーブレット係数に置き換える。この操作をレベル0まで行い、根のノードを初期front(表示三角形集合)にする事で、4分木が完成する。この結果、frontより下のレベルの頂点は全てウェーブレット係数として表されている(図3)。

4. LOD制御

LOD (Level of Detail) 制御とは、モデルの解像度を目的にあうように変化させる制御である。本研究では、

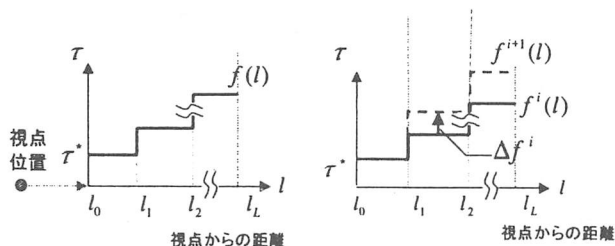


図4 階段型関数 $f(l)$ 図5 $f(l)$ の修正

地形の表示許容誤差を満たす地形依存型 LOD 制御、視点位置からの距離に依存する視点依存型 LOD 制御、表示ポリゴンの総数を一定化する表示ポリゴン総数 LOD 制御の3種類の LOD 制御を、ウェーブレット変換と4分木構造を利用して同時に行う。

4. 1 地形依存型 LOD 制御

ウェーブレット係数の絶対値とユーザの指定する地形表示許容誤差 τ^* の大小比較により、その点を表示に使用するかどうかを決定する。 $|w^j(m, n)| \geq \tau^*$ であれば、その格子点で逆変換により高さを再構成して表示と判定し、 $|w^j(m, n)| < \tau^*$ であれば、変換を行い、高さをウェーブレット係数に置き換え、非表示と判定する。

4. 2 視点依存型 LOD 制御

視点依存型 LOD 制御では、地形許容誤差 τ^* を最小とする視点距離 l に依存した階段型の関数 $f(l)$ (図4) と $|w^j(m, n)|$ との大小比較により頂点の表示判定を行う。今、視野錐台内において視点からの最近頂点までの距離を l_0 、最遠頂点までの距離を l_L と置き、その間を L 個の区間に分割する。 $f(l)$ は以下の性質をもつ。 $f(l)$ は各区間内では一定で、 $f(l_0) = \tau^*$ であり、 $f(l_0) < f(l_1) < \dots < f(l_L)$ である。 $f(l)$ を4.1節の τ^* と同様に考え、 $|w^j(m, n)|$ との大小比較により、頂点の表示・非表示を決定する。

4. 3 表示ポリゴン総数 LOD 制御

$f(l)$ を変化させる事で、表示ポリゴン総数 LOD 制御が可能となる。時刻 i における関数 $f(l)$ を $f^i(l)$ と定義する。まず、使用する PC のポリゴン表示性能 R (ポリゴン/s) と、目標とする表示速度 v (frame/s) を用いて、ポリゴン総数の目標値 $N_T = R/v$ (個) を決定する。次に時刻 i における表示ポリゴン数 N_C^i に基づき、目標値との差 $\Delta N^i = N_C^i - N_T$ を算出する。これを用いて、時刻 $i+1$ における関数 $f^{i+1}(l)$ の各区間の修正値 Δf^i を $\Delta f^i = C_l \times \Delta N^i$ を用いて計算する。 C_l は視点からの距離に対する重み付け係数で、 $C_1 < C_2 < \dots < C_L$ である。図5のように、 $f^{i+1}(l) = f^i(l) + \Delta f^i$ と更新し、次の時刻の頂点表示判定に用いる。

5. ウェーブレット変換・逆変換による front の変更

以上の LOD 制御により、表示対象と判定された頂点を追加するには、4分木を用いてその頂点を含む三角形集合を特定し、その集合が現 front になれば、その上位にある front 三角形を現 front から削除し、最初に特定した集合を front に加える。追加頂点に対しては、ウェーブレット逆変換を行う。頂点を削除する場合は、反対の操作を行い、ウェーブレット変換を行う。

6. 差分データ管理

LOD 制御に加え、さらに動的表示速度の向上をはかるため、front に含まれる三角形の情報を持つリストを OpenGL の DisplayList を用いて管理する。

図6のように、時刻 i における front を $Fr(i)$ と表すと、 $Fr(i-1) - Fr(i)$ に属する三角形は DisplayList から削除し、 $Fr(i) - Fr(i-1)$ に属する三角形は DisplayList

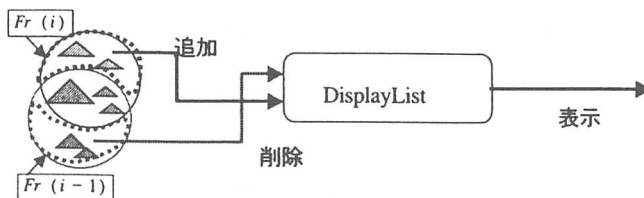
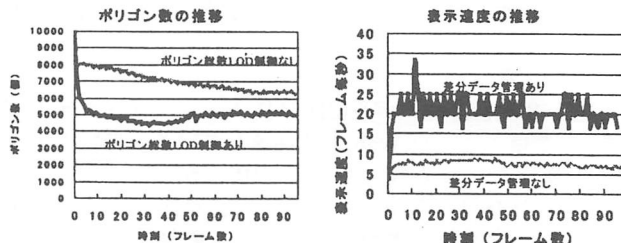


図6 差分データ管理の流れ



(a) ポリゴン総数 LOD 制御 (b) 差分データ管理
図7 LOD 制御と差分データ管理の効果

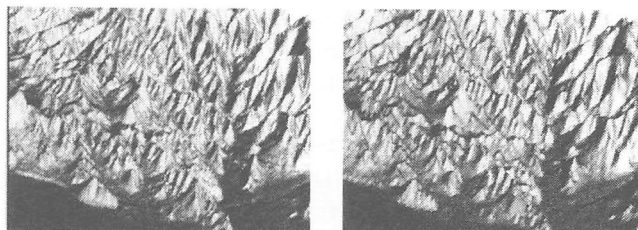


図8 最高解像度の表示(左)と LOD 制御による表示(右)

に追加する。これにより、DisplayList の中には時刻 i において表示すべき三角形のみ含まれることになる。ここで $Fr(i) \cap Fr(i-1)$ の集合に含まれる三角形は、時刻 $i-1$ での DisplayList を再利用できるため、頂点座標値計算を行う必要がなく、処理速度を高めることができる。

7. 実行結果

表示用 PC として、CPU が PIV 1.8GHz、GPU が RIVI TNT2 64(32MB) を用いた。DEM データは札幌南西部の 50m メッシュ DEM、頂点数 257×257、総三角形数 131072 個を使用した。

まず、ポリゴン総数 LOD 制御の有無によるポリゴン数の推移を比較した結果を図7(a)に示す。ここで設定は、 $N_T = 5000$ 、 $C_l = l-1$ 、 $L = 5$ とした。提案手法が有効に機能していることが分かる。

次に差分データ管理の有無による表示速度の比較により、図7(b)の結果を得た。差分データ管理手法を用いた時は、表示速度が3倍程度となることが確認された。

また図8では、最高解像度の表示(三角形数 131,072)と地形・視点 LOD 制御を行った表示(三角形数 31,592)の比較を示す。LOD 制御を行うと、三角形数が4分の1程度に減少するが、視覚的に差異がないことが分かる。

8. おわりに

本研究では、大規模3次元地形モデルを動的かつ効率的に表示するために、4分木構造およびウェーブレット係数の閾値処理を利用した LOD 制御と、高速表示のための差分データ管理手法を提案し、有効性を確認した。

[参考文献]

[1] D. Cline et al.: Terrain Decimation through Quadtree Morphing, IEEE Trans. visualization and computer graphics Vol. 7-1 (2001) pp.62-69
[2] E. Stollnitz et al.: Wavelets for computer graphics, Morgan Kaufmann (1996), pp.102-105