

大規模森林表示のための テクスチャ座標ビルボード生成に関する研究

赤木康宏[†] 北嶋克寛[†]

本稿では数万本以上の樹木を対象とする大規模自然景観表示を可能にするための、新たな樹木描画高速化手法を提案している。樹木形状の簡略化および表示高速化に関してはビルボードを用いたアプローチが有効であり、これまで多くの手法が提案されている。本研究では、葉の配置（テクスチャ座標）パターンを動的に生成するために、ビルボードに用いるテクスチャに対して葉の密度情報を格納することで、個々の葉の情報を省略し、メモリ負荷の少ないビルボード描画を実現する。提案手法を実装し、実験により画像の品質劣化が小さいことを検証する。

A Generation method of Texture Coordinate Texture for Rendering a Large-scale Forest.

YASUHIRO AKAGI[†] KATSUHIRO KITAJIMA[†]

In this study, we propose a rendering technique for a large-scale forest which has tens of thousands of trees. For the rendering of forests, billboard-based techniques are typically used in many studies. In our approach, we aim to reduce texture memories by using a "Density Texture" of leaves. The Density Texture has density values of leaves in each Texel on a billboard and dynamically generates patterns of texture coordinate. We implement this method and verify the quality of rendering, a texture size and frame rates to render large-scale forests.

1. はじめに

本稿では数十種、数万本の樹木から構成される大規模自然景観表示を実現するための樹木描画高速化手法について述べる。近年、GISを用いた地理情報表示に関する様々な研究およびサービスが実現している。特に、都市景観表示においては3次元CGによる建物表現が可能となり、その利用価値が高まっている。一方、自然景観表示に関しては、地形の起伏の表現は可能であるものの、自然物の表現では航空・衛星写真を用いることがほとんどであり、形状を立体的に表現することは少ない。特に、自然景観を構成する主要な要素である樹木はその形状の複雑さから、葉や枝の個々の形状を表現した場合、表示コストが非常に高くなるという問題があり、自然景観表示を困難なものにしている。本稿では、ビルボードを用いることで樹木形状表示に要するメモリ使用量および描画コストの削減をおこなう新たな手法の提案をおこなう。

2. ビルボードによる樹木表示

ビルボードによる樹木表現は、樹木の表示コストを削減する典型的な解決方法¹⁾であり、複数枚のビルボードを組み合わせた代替形状による表現が多く用いられる。これらの手法では、複数のビルボードに対して樹木形状を投影し、レンダリングすることで各ビルボードに対応する画像データを生成する。この画像をテクスチャとして各ビルボードに適用することで樹木形状を少数の平面で表現している。

2.1 ビルボード生成に関する研究

樹木を近似するために、効率よくビルボードを自動生成する手法として Décoret らの研究²⁾がある。Décoret らは Billboard Clouds と呼ばれる、対象となる形状を近似するための平面の集合を指定した枚数で生成する手法を開発した。この手法では、ビルボード生成に関する一連の処理の自動化が可能であり、表示コスト（平面の枚数）の制御が容易であるという特徴がある。しかし、各ビルボードに用いるテクスチャの解像度が表示品質へ大きな影響を与えるので、近距離からの表示を行う際には高解像度のテクスチャを用意する必要があり、大量の樹木を表示する際にはメモリ使用量が問題となる。そこで、Garcia らは Indirect Texturing（間接テクスチャ）と呼ばれる手法³⁾によって、メモリ使用量の削減を実現した。

* † 東京農工大学
Tokyo University of Agriculture and Technology

2.2 Indirect Texturingによる樹木表示

この手法では樹木が同形状をした葉の集合から構成されている点に着目し、ビルボード用のテクスチャに通常の葉をレンダリングした画像を用いず、各テクセルへ葉の有無および回転量を保存した画像 (Indirect Texture) を用いる。そして、ビルボードの各ピクセルを描画する際には、まず Indirect Texture の値を読み取り、そこに埋め込まれた情報から、高解像度の葉のテクスチャを参照することで、最終的な表示色を決定する。本手法では、Indirect Texture の解像度を落としても、表示品質の劣化を抑えることが可能であり、メモリ使用量を削減できるという特徴がある。しかし、各テクセルが1枚の葉の有無を表現しているのみなので、葉の重なりを表現できないという問題がある。そこで、Garcia らの研究⁴⁾では、樹木を非常に近距離で表示したときであっても葉の重なりを正しく表示するために、Indirect Texture と葉のテクスチャとの間に、葉の重なり状態を記憶するためのテクスチャを追加することでこの問題を解決した。この手法では、近距離での樹木表現の品質を、通常のポリゴン表示に対してほとんど劣化させることなく表示が可能であるが、1本の樹木に対して約1~3MByteのテクスチャメモリを使用するので、多種の樹木を表示する場面への応用は難しい。そこで、中~遠景の表示に適した効率のよい Indirect Texture 生成手法を提案する。

3. 密度テクスチャを用いた樹木描画

本節では、よりメモリ負荷の少ない Indirect Texture の一種である密度テクスチャの生成手法について述べる。

3.1 樹木表現用のビルボードの特徴

典型的な樹木表現用のビルボード (図1) は同形状の葉が重なり合い敷き詰められた画像となることがほとんどである。それぞれの葉は、葉を表現するためのポリゴンに対して葉の写真などをテクスチャとして適用することで描画される。この時、1つの葉形状に対して、1枚のテクスチャが対応するという特徴がある。先ほど述べた Garcia らの手法においても、この特徴を利用し、対応関係を表現するテクスチャの生成を行っている。

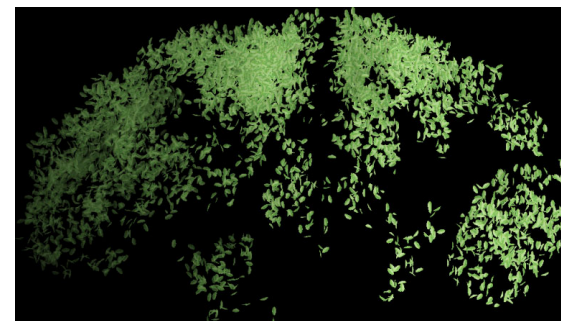


図1 典型的な樹木ビルボード

本研究では、図1で示した樹木の葉を、テクスチャマッピングせずに、テクスチャ座標を色情報 (u座標を赤、v座標を緑) として出力し (図2)、情報の圧縮を行う。図2に示した画像は、テクスチャ座標を色情報として描画したものである。以降、同様の方法で描画する画像をUVテクスチャと呼ぶ。UVテクスチャは単調に変化するグラデーションが連続した画像であるという特徴がある。このような画像になる原因は、描画のもととなる葉の形状が平面に近く、その上に1枚のテクスチャをゆがめることなく配置していることに由来する。各グラデーションの画素値はテクスチャ座標に由来するので、0から255への単調な変化、もしくは重なりなどにより一定の区間が切り出された画像となるという特徴がある。次節では、この特徴を利用し情報の圧縮を行うための手法について述べる。

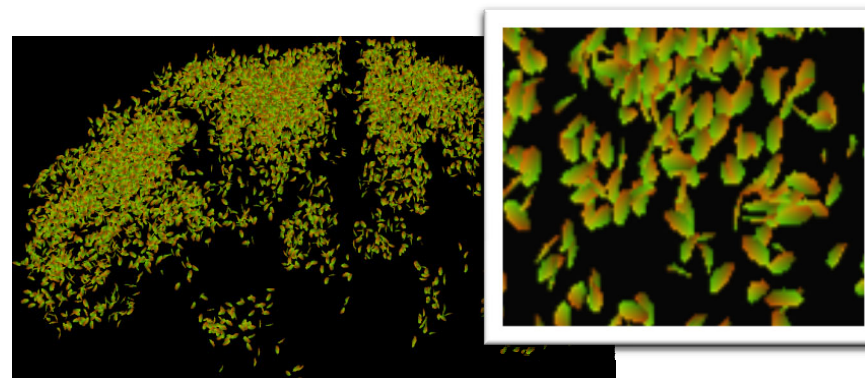


図2 UVテクスチャ

3.2 UVテクスチャの解析

3.1節で述べた、UVテクスチャの特徴を利用し、のこぎり波との相関をとることで、任意のピクセル位置で、周辺の葉の配置状態（葉の有無・密度等）を解析する。画像圧縮および信号処理の分野においては、フーリエ変換（三角関数との相関を調べる）が広く用いられているが、UVテクスチャの場合では、単調な勾配から画像が構成されているので、のこぎり波による解析が有効である。本研究で用いるのこぎり波は次の式 1 で表す。

$$f(x) = \frac{x}{w} - \left\lfloor \frac{x}{w} \right\rfloor \quad w: \text{周期 (密度)} \quad \dots \text{式 1}$$

w はのこぎり波の周期を表す係数であり、UV 画像を解析する際には、任意の地点において、周囲のピクセル値と式(1)の相関をとる。実験では周囲 16 ピクセルに対して解析をおこなっているが、この範囲は画像の圧縮率に応じて設定する。この時、葉の取りうる範囲で周期を変化させることで、最も相関の高い周期を検出する。本研究では、この周期 W は、周囲の葉の繰り返し度合い、つまり密度の情報を表している情報と考える。密度の解析は UV 画像の横方向および縦方向のそれぞれについておこない、密度 w_x および w_y の検出をおこなう。そして、各密度 w は周辺の葉の配置状態を近似しているものとして、画像の圧縮に用いる。

3.3 密度テクスチャの生成

本節では、3.2 節で述べた手法により得られた、各地点での葉の密度情報を保存する、密度テクスチャの生成について述べる。葉の密度情報は w_x および w_y の 2 次元の実数値であるので、これを 0~255 の値に量子化し密度テクスチャの赤成分および緑成分として保存する。密度テクスチャは、UVテクスチャに対しておこなった解析の範囲（のこぎり波との相関をとる範囲）の半分のサイズの間隔でサンプリングをおこなう。実験では、解析の範囲が 16 ピクセルであるので、密度テクスチャは 8 ピクセル毎にサンプリングをおこない、画像のサイズは 1/64 となる。また、ビルボードへの詳細な陰影付けを可能にするために、法線の情報も併せて密度テクスチャへ格納する。具体的には、密度テクスチャの青成分およびアルファ成分に法線の x 成分および y 成分を量子化し格納する。残りの z 成分については、法線ベクトルが単位ベクトルであるという性質から z 成分の絶対値を求め、ビルボードの向きから方向を求めることで、描画時に情報を復元する。図 3 に生成した密度テクスチャの例を示す。

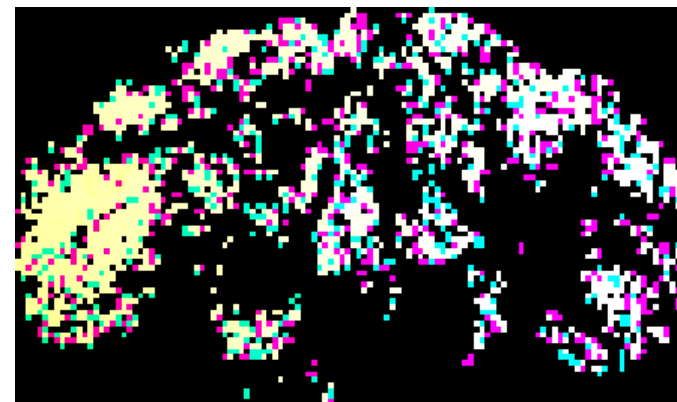


図 3 密度テクスチャ

3.4 密度テクスチャの統合

3.4 節で述べた密度テクスチャはビルボードの各矩形ごとに生成する必要がある。描画処理をおこなう際には、各樹木毎に複数の密度テクスチャを入れ替える必要があり、手間がかかる。そこで、1 つの樹木で用いられる複数の密度テクスチャを 1 枚の画像に統合し、個々のビルボードのテクスチャ座標を調整することで、テクスチャ入れ替えにかかる負荷を軽減させる。

3.5 ビルボード描画

本節では 3.1 ~ 3.3 節で述べた手法により生成した密度テクスチャを用いたビルボード描画について述べる。描画に関わる主要な処理は Pixel Shader によりおこなわれる。処理手順を以下に示す。

- (1) ビルボード上でのテクスチャ座標から、密度テクスチャの色情報（葉の状態色と呼ぶ）を得る。（サンプリング）
- (2) 葉の状態色の赤および緑成分から密度の情報を取り出し、のこぎり波の関数（式 1）に与える。このとき、位相 x はビルボードのテクスチャ座標とする。（隣接するピクセルでの整合性を保つため）
- (3) のこぎり波関数より得られた値をテクスチャ座標として、葉の色情報をもつテクスチャから出力用の色情報を得る。
- (4) 葉の状態色の青およびアルファ成分より、法線ベクトルの復元をおこない、出力用の色情報に対して陰影付けをおこない、出力する。

以上の処理により、密度テクスチャに基づく葉の描画をおこなうことができる。通常のビルボード描画と比較した場合、

- ① テクスチャを2回参照する
- ② のこぎり波関数を参照する
- ③ 法線の復元をおこなう

という点において負荷が増大している。②および③の処理は単純な演算であるので影響は小さい。①に関しては、処理時間を増大させる要因となる可能性があり、テクスチャサイズの低減効果とのトレードオフの関係にある。

4. 実験及び考察

本章では、3章で述べた一連の手法を実装したシステムを用いて、実験により描画品質およびテクスチャ容量の削減効果、描画速度等の検証を行う。実験には38952枚の葉をもつ樹木形状データを用いた。実行環境を表1に示す。

表1 実験環境

CPU	Core2duo E6700
RAM	2GByte
GPU	GeForce 8800GT
OS	Windows Vista

実験の前処理として、ビルボードの生成およびUV画像の生成、密度テクスチャの生成を行った。なお、ビルボードは樹木形状を近似した24個の平面をから構成されるものを用いた。1つの樹木を表現するための統合された密度テクスチャの解像度は各ビルボードの解像度を128×128pixelとし、横6枚、縦4枚を並べた768×512pixelとした。

4.1 描画品質の比較

まず、密度テクスチャを用いた樹木の描画品質を比較するために、次の3種類の方法で樹木の描画をおこなった。

- (1) 樹木形状データ（ポリゴン）を通常描画
- (2) 高解像度（1024×1024pixel×24枚）で生成した典型的なビルボードテクスチャによる描画（ビルボードのポリゴンは(3)と共通）
- (3) 密度テクスチャを用いた描画

それぞれの結果について図4に示す。まず、(2)の結果より、本実験で用いたビルボードは適切に元の樹木形状を近似していることが確認できる。葉を平面へ投影しているために、輪郭部でのばらつき、および葉の密度の変化が多少みられるものの、十分な品質を保っている。本稿で提案する手法(3)の結果では、輪郭形状がよく保存されていることがわかる。(3)の手法では、密度の解析をおこなっているため、(2)で見られたばらつきが生じている地点では十分な密度が検出されず、葉の存在しない領域として扱われたためと考えられる。また、葉の粗密については若干の増加がみられるものの、各部分の状態は表現できている。一方、葉の並び方が規則的であるので、いくつかの領域で縞状のパターンが見えるという問題もある。これは、(3)の手法が葉の姿勢を扱っていないために生じる現象である。濃度テクスチャとは別に葉の回転情報を与えることで解決できるが、その情報を取得するために別のテクスチャを参照することは処理への負荷が高いという問題もあり、本手法では陰影を生成するための法線情報を優先し、回転情報を送ることはしなかった。また、(3)の手法によって描画した樹木を近くから見た場合、図5のように樹木の周辺分ではテクスチャの乱れが生じた。これは、密度情報から復元したのこぎり波状のテクスチャ座標が途中で途切れてしまっているために生じる問題手あり、近距離の表示においても品質を保つためには別の手法をLODの考え方に従い利用する必要がある。

4.2 テクスチャサイズの比較

次に、従来のIndirect Textureを用いた手法とのテクスチャサイズの比較をおこなう。Garciaらの研究では、23660枚の葉をもつ樹木に対してビルボード用テクスチャを生成し、DXT5形式で圧縮することで、3,119,734byteのメモリ占有量がある。実験で用いた濃度テクスチャを同様にDXT5形式で圧縮した結果、393,344byteとなり、十分に小さなメモリ占有量で済んでいることが確認できた。

4.3 描画速度

最後に、大規模自然景観表示を想定したシーンにおける描画速度の検証をおこなう。ここでは、これまでの実験で用いた樹木と同程度の葉をもち、形状が異なる樹木を5種類用意し、それぞれに濃度テクスチャの生成をおこなう。これを数千本単位でランダムに配置することでシーンを構成した（図6・4万本を表示した際の結果）。計測結果を表2および図7に示す。

表 2 樹木数と描画速度

樹木数(本)	5000	10000	20000	30000	40000	45000
FPS	66.7	21.3	7.1	4.3	3.0	2.4

この結果から、約 1 万本までの樹木数であれば十分な速度で描画でき、4 万本を超えるようなシーンにおいても 1 秒以内で描画が可能であることを確認した。2 万本以上の本数を描画する際にも十分な応答性を得るためには、図 6 の結果からもわかるように、詳細な形状が見えなくなる遠方の樹木を構成するビルボードの枚数を動的に減らすなどの工夫により、速度の向上を図る必要がある。

5. おわりに

本研究では大規模自然景観表示を実現するために、密度テクスチャを用いた樹木描画高速化手法を提案し、次のような成果を得た。

- (1) 樹木ビルボードに用いられる典型的なテクスチャに対して、のこぎり波を用いた密度情報の検出手法を適用し、テクスチャ容量の少ない密度テクスチャの生成をおこなった。
- (2) 密度テクスチャを用いた樹木のビルボード表示を実装し、描画品質の劣化が実用上問題のない程度に抑えられていることを確認した。
- (3) 0.5~4.5 万本の樹木を含む大規模自然景観に対して描画速度の計測をおこない、1 万本程度の樹木数であれば、応答性の高いシステムが構築できることが確認できた。

さらに品質および応答性の高い大規模自然景観表示をおこなうためには、表示範囲に応じて複数の簡略化手法を適宜組み合わせる必要がある。

謝辞

本研究の一部は、文部科学省科学研究費補助金 若手研究 (B) 課題番号 21700095「地理空間情報システムを利用した大規模仮想自然環境表示技術の構築」の支援を受けたものである。

参考文献

- 1) Stephan Behrendt, Carsten Colditz, Oliver Franzke, Johannes Kopf, and Oliver Deussen. : Realistic real-time rendering of landscapes using billboard clouds. *Comput. Graph. Forum*, Vol.24, No.3, pp.507–516 (2005).
- 2) Décoret, X., Durand, F., Sillion, F. X., and Dorsey, J. 2003. Billboard clouds for extreme model simplification. *ACM Trans. Graph.* 22, No.3, pp.689–696 (2003).
- 3) I.Garcia, G.Patow, M.Sbert, and L.Szirmay-Kalos. Tree Rendering with Billboard Clouds. *Third Hungarian Conference on Computer Graphics and Geometry 2005*, (2005).
- 4) I.Garcia, G.Patow, M.Sbert, and L.Szirmay-Kalos. Multi-layered indirect texturing for tree rendering. *Eurographics Workshop on Natural Phenomena 2007*, (2007).

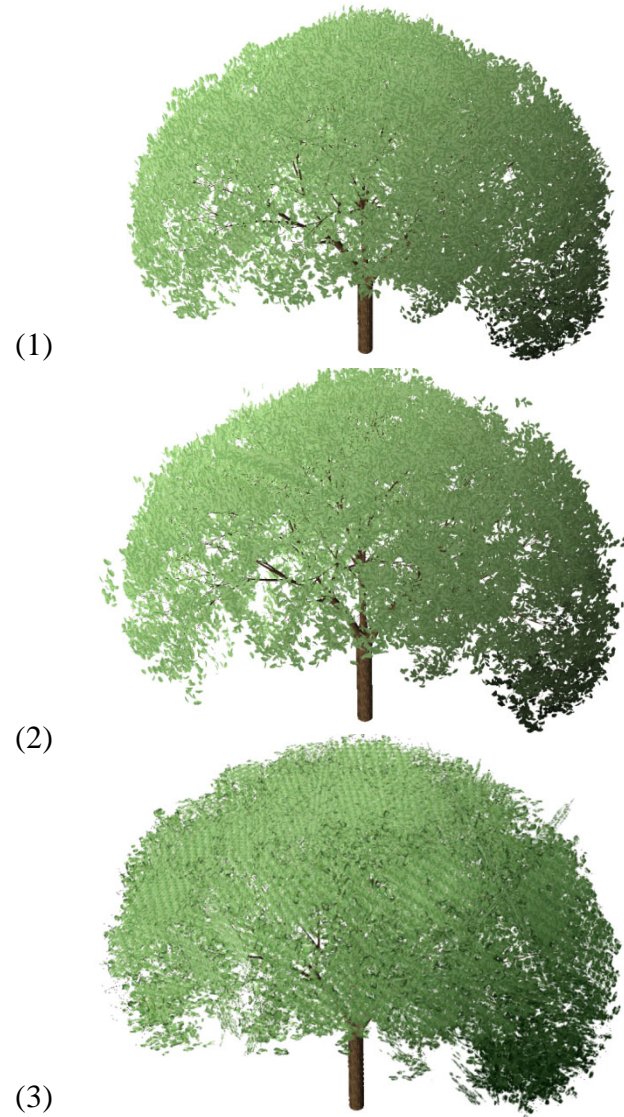


図 4 描画品質の比較



図 5 周辺部の葉の描画結果

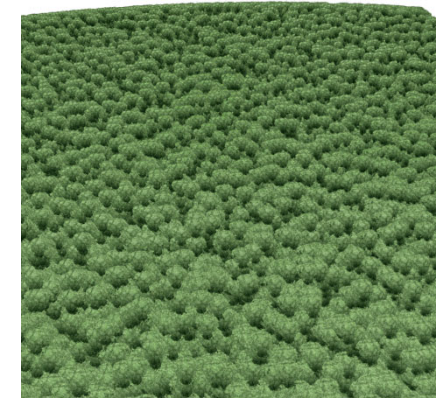


図 6 大規模森林表示

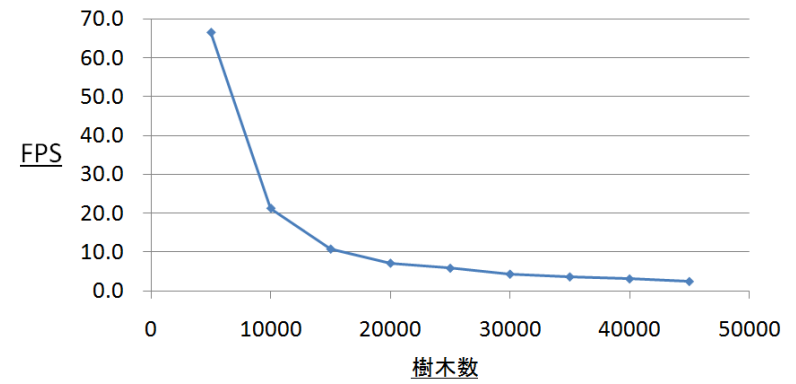


図 7 樹木数と描画速度