

## 幾何形状と画像を用いた草のリアルタイムアニメーションの高速化

馬場園 大地\* 中本 啓子† 小池 崇文\*

\* 法政大学情報科学部 † 法政大学大学院情報科学研究科

## 1. はじめに

本研究では、幾何形状で表現した草（以下草形状）と画像によって表現した草（以下草画像）を用いて、画質を劣化させずに風の影響下における草原のリアルタイムアニメーションの生成速度を高速化する手法を提案する。CGにおいて、高速化のために処理を軽くした場合、その分画質は低下する。しかし、例えばVRでの利用を考えると、画質を劣化させずに生成速度を高速化する手法が必要である。

そこで、生成する草の位置とカメラの距離に応じて草形状と草画像を切り替えることで、見た目の劣化を抑えた高速化を行う。また、画質向上のため、物理現象による草の動きを再現する。カメラ付近に生成する草形状では、重力、草の復元力、風、他物体との衝突による草の動きを再現する。遠方に生成する草画像では、草画像の濃淡や周波数成分を変更し、風による草の動きを再現する。本手法で画質を劣化せずに高速化を実現できれば、より複雑な処理を追加した草のリアルタイムアニメーションを生成することができる。

## 2. 関連研究

草の生成手法には、草画像による手法と草形状による手法がある。Orthmannらは、草画像を変形させ、他物体と草が衝突した時のアニメーションを生成した[1]。またJahrmannらは、草形状を用いて複数の物理現象による草の動きを再現したアニメーションをリアルタイムで生成した[2]。

CGでは、遠方のモデルのポリゴン数を削減して処理の高速化を行うことが多い。しかし、草のような単体オブジェクトではポリゴン数が少なく、オブジェクト全体ではポリゴン数が多いシーンでは、ポリゴン数を大きく削れず、処理時間を大きく減らすことができない。Fanらは、生成する草形状の量を調整することで高速化を実現した[3]。また、Boulangerらは、草画像と草形状を両方用いることで高速化を実現した[4]。この手法では、カメラ付近の草は草形状、遠方の草は草画像、中間位置の草はその両方で生成し、画質の劣化を抑えている。

本研究は、Jahrmannらの手法に草画像を組み込むことで高速化を行う。一方で、Boulangerらの手法のように、草形状と草画像が混在する領域を作らず、草形状の生成量を減らすことで、より高速な処理を実現する。また、混在領域を削除したことで発生する見た目の違和感は、草画像において風による草の動きを再現することで解決する。

## 3. 風影響下の草の画像特性

風影響下の草原の画像と風の影響を受けていない草原の画像では、濃淡と周波数特性に差がある。草画像において、風の影響による濃淡と周波数特性の変化を実現することで、風による草の動きを再現できる。これにより、草画像が現実に近い見た目となり、草形状と草画像の境界がわかりにくくなるため、草画像を用いたことによる見た目の劣化を抑える。

## 3.1. 濃淡の変化

風の影響を受けて草が倒れることで、風の影響を受けていない草と比べて色の濃い部分が生まれる。色の濃い部分が風向きの方に徐々に移動していくことで、風による動きが確認できる。一定の方向に一定の間隔で風が吹いている場合、風

による濃淡の変化には周期性が存在する。周期的な変化の再現には複雑な処理を必要としないため、草画像に対して画像処理で風の動きを再現しても、十分に処理の高速化を行える。画像処理では、風の影響を受けていない草画像に対して色の濃い部分を作り、一定のフレーム毎に色の濃い部分を移動させることで風による草の濃淡変化を再現する。

## 3.2. 周波数特性

風影響下の草原の画像と風の影響を受けていない草原の画像を用い、それぞれのフーリエ変換画像を作成して周波数特性の解析を行った。

図1(A)は風の影響を受けていない草原を上から見た時の画像で、図1(B)がそのフーリエ変換画像である。図1(C)は右下から左上方向への風の影響下にある草原の画像で、図1(D)がそのフーリエ変換画像である。図1(B)より、風の影響がない時は各周波数成分が均等に存在していることが確認できる。また図1(D)より、風向きに直交する向きの周波数成分のうち、高周波数成分が減少していることが確認できる。このことから、風向きに直交する向きの高周波数成分を減少させることで、風による草画像の周波数特性の変化を再現できる。

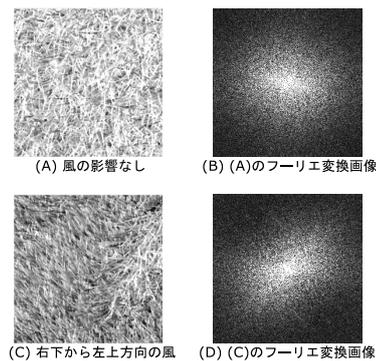


図1. 草画像の周波数特性の解析。

## 4. 提案手法

本手法は平らな地面に草原を生成することを想定し、以下の手順で処理を行う。草画像と草形状で草を生成していくにあたり、図2のように境界となる位置を決める。その境界を元に、地面となる平面を2枚生成する。その地面の一方に、Jahrmannらの手法を用いて草形状を生成する。図3のように、生成した草形状を上から鉛直下向きで草画像を作成し、草画像に対して風向きに対応した周波数成分を減少させる処理を行う。また、草画像に対して風による濃淡の変化を再現する処理を行う。最後に、それらの草画像を用いて、もう一方の地面に草を生成していく。

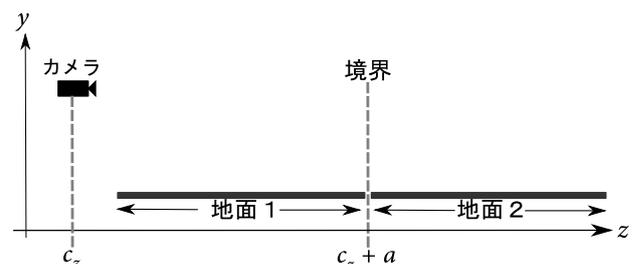


図2. 境界と地面の生成。

Speed-up of Real-Time Grass Animation by Geometrical Object and Image Processing

Daichi Babazono, Keiko Nakamoto, and Takafumi Koike  
Faculty of Computer and Information Sciences, Hosei University  
3-7-2 Kajino-cho, Koganei-shi, Tokyo, Japan

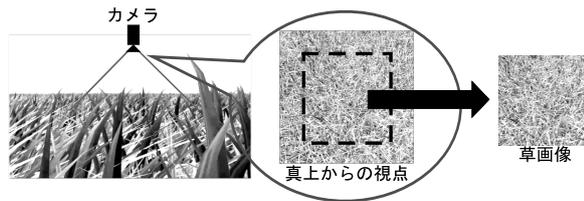


図 3. 草画像の生成方法.

#### 4.1. 草画像の周波数成分の変更処理

風向きに直交する向きの高周波数成分を減少させるため、草画像のフーリエ変換画像に対してローパスフィルター（以下 LPF）をかける。本手法では、 $\sigma = 1$  のガウシアンフィルターを LPF として用いる。

高周波成分を減少させる処理は以下の手順で行う。まず、風の影響を受けていない草画像に対し LPF をかけたフーリエ変換画像を用意する。次に、図 4 のように、風向きに応じた高周波成分を減らしたフーリエ変換画像を部分的に切り取る。そして切り取った画像を、元の草画像のフーリエ変換画像の対応部分に置き換える。最後に、置き換え後のフーリエ変換画像に対し逆フーリエ変換を行うことで、任意の高周波成分を減少させた草画像を生成する。

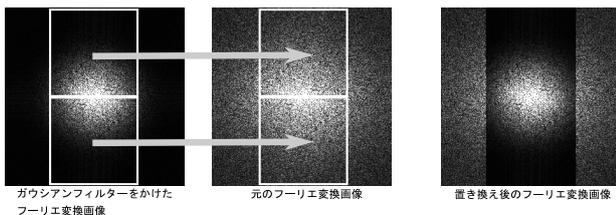


図 4. 高周波成分を減少させた草画像の生成. 図では、下から上方向への風向きを想定している。

#### 4.2. 草画像の濃淡の変更処理

濃淡の変化は、作成した草画像 1 枚と明度の情報を持つ画像（以下明度画像）複数枚を用いて行う。明度画像は、図 5 (B) のように、草画像と同じサイズで、草画像の明度を変更したい部分のみ明度を変更したグレースケール画像である。明度画像の各画素の座標とその画素値を元に、図 5(C) のように明度を部分的に下げた草画像を作成する。

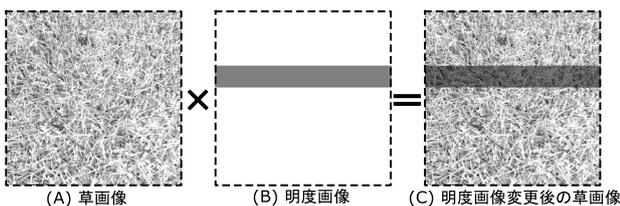


図 5. 草画像の明度変更の例

元の草画像を  $G$ 、画像処理に用いる明度画像を  $V$  とし、 $V$  を用いて  $G$  の明度を  $n$  倍した草画像  $G'$  を生成することを考える。この時、 $G'$  における座標  $(i, j)$  にある画素値  $G'(i, j)$  は式 (1) によって計算される。全ての画素において計算を行うことで、画像全体で明度を変更していくことができる。

$$\begin{aligned} G'(i, j) &= G(i, j) \times n \\ n &= V(i, j)/255 \end{aligned} \quad (1)$$

$G(i, j)$  は  $G$  における座標  $(i, j)$  の画素値、 $V(i, j)$  は  $V$  における座標  $(i, j)$  の画素値である。明度画像の各画素は 1 ~ 255 の値を持つようにし、 $n$  の値を  $(0, 1]$  の範囲に収めることで、画像処理前後で明度が大きくならないようにする。ま

た、フレーム毎に用いる明度画像を変更して画像処理を行い、草画像の明度を下げる部分を移動させていく。

## 5. 実験

提案手法と既存手法で、草原のリアルタイムアニメーション生成速度の比較を行う。また、提案手法と既存手法のアニメーション画質を比較し、劣化が起きていないか調べる。

### 5.1. 実験方法

実験では、Jahrmann らの手法 [2] と提案手法で比較を行う。同じ広さの地面を生成し、その上に各手法で草を生成する。また、各手法における風向きと草形状の位置、高さ、幅、向きは同じ条件とする。風向きを画面手前から奥方向とし、300 フレーム描画を行って、平均 FPS を比較する。シーンの作成は、C++ と OpenGL を用いて行った。また、CPU は Intel Core i7-7700、GPU は NVIDIA GeForce GTX 1060 6GB を使用した。

### 5.2. 結果

図 6 に描画結果を示す。既存手法での平均 FPS は 43.4 で、提案手法では 65.0 であり、描画速度は約 1.5 倍となった。また、図 6 のようにカメラから草画像まで距離があると、境界がわかりにくくなり、見た目の劣化を抑えることができている。

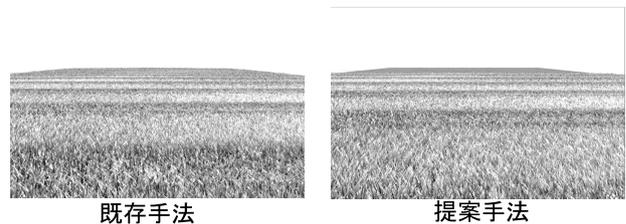


図 6. 描画されたシーン.

### 5.3. 考察

草画像までの距離を近くした場合、境界がわかりやすくなり、草画像の濃淡移動と草形状の動きとの差が目立つようになる。これは、草形状において発生する濃淡変化の間隔を考慮できていないことが原因と考えられる。図 6 の既存手法より、濃淡変化の間隔は画面奥になるほど狭くなる。一方、提案手法の草画像は、一定の大きさの草画像を敷き詰める方法で草を生成しているため、濃淡変化の間隔が常にテクスチャ 1 枚分の幅となっている。これを改善するには、カメラの位置に応じて、草画像における濃淡変化の間隔を変更する必要があると考えられる。

## 6. まとめ

本研究では、草原のリアルタイムアニメーションの高速化を目指し、草形状と草画像を組み合わせた手法を提案した。また、アニメーションの画質の劣化を抑えるため、草画像の濃淡や周波数成分を変更する処理を行い、草画像において風による草の動きを再現した。草形状のみのアニメーションを生成する既存手法と比較を行ったところ、1 フレームの生成速度が上がった。このことから、提案手法は草のリアルタイムアニメーションの高速化において有効であった。

## 参考文献

- [1] J. Orthman et al., "GPU-based Responsive Grass," *J. WSCG*, vol. 17, no. 1-3, pp. 65-72, 2009.
- [2] K. Jahrmann and M. Wimmer, "Responsive Real-Time Grass Rendering for General 3D Scenes," in *Proc. the 21st ACM SIGGRAPH Symposium on i3D*, pp. 1-10, 2017.
- [3] Z. Fan et al., "Simulation and Rendering for Millions of Grass Blades," in *Proc. the 19th ACM SIGGRAPH Symposium on i3D*, pp. 55-60, 2015.
- [4] K. Boulanger et al., "Rendering Grass in Real Time with Dynamic Lighting," *IEEE Comput. Graph. Appl.*, vol. 29, 1, pp. 32-41, 2009.