

サーフェルによる連続体 / 不連続体の統一的表现

榎山 奈保 藤代 一成[†]

お茶の水女子大学 大学院人間文化研究科

[†] お茶の水女子大学 理学部 情報科学科

〒112-8610 東京都文京区大塚 2-1-1

naho@imv.is.ocha.ac.jp, [†]fuji@is.ocha.ac.jp

コンピュータグラフィックスにおいて、連続体と不連続体をと共に表現する場面は頻繁に現れる。これらを統一的に表現できる方法としてメタボールが知られているが、不連続部分の粒子感を表現するためには、大量のプリミティブの指定が必要になるだけでなく、等濃度面を抽出する際に生じるポリゴンパッチの個数が無視できなくなる。2000年に Pfister, 他は、複雑な形状の連続体を点をベースとしてサンプリングした上で、それを中心として接平面に平行な微小円盤上にテクスチャをマッピングすることで、インタラクティブなフレームレートでレンダリングする方法である、サーフェルを提案した。本論文ではこの方法でプリミティブの主要構成要素が点であることに着目し、連続体と不連続体を統一的にレンダリングできるようにする手法を提案する。

A Unified Representation of Continuous / Discontinuous Objects Using Surfels

Naho Kashiya Issei Fujishiro[†]

Graduate School of Humanities and Sciences

[†] Department of Information Sciences

Ochanomizu University

2-1-1 Otsuka, Bunkyo-Ku, Tokyo 112-8610, Japan

naho@imv.is.ocha.ac.jp, [†]fuji@is.ocha.ac.jp

In many synthesized scenes, simultaneous rendering of continuous and discontinuous objects is frequently required. When employing metaballs, we need to specify a large number of primitives in discontinuous portions, resulting in a highly-complex polygonal patch structure of iso-density surfaces. In 2000, H. Pfister, et al. proposed a novel modeling technique, called surfels, which relies on a point-based approximation to render complex-shaped continuous objects at an interactive frame rates. In this paper, we describe an initial attempt to extend surfels so that it can serve as a unified representation framework for both types of objects.

1 はじめに

近年、3D コンピュータグラフィックス (CG) は一般にも普及し、あらゆるところでその成果を目にするようになってきた。なかでもシミュレーションゲームなどではインタラクティブなグラフィックス表示を要求するものが数多く出ている。比較的平坦な形状をもつオブジェクトの表現であれば、視覚的複雑さは、テクスチャの利用によって容易に増すことができる [1]。一方、複雑な形状のオブジェクトは、通常膨大な個数のポリゴンパッチへ分割され、時空間計算量の増加を招いている。しかし、パッチへの分割に伴う相互の接続情報は、必ずしもレンダリングに必要な情報ではない。

2000年に Pfister, 他はこの事実に注目し、複雑な形状のオブジェクトをインタラクティブなフレームレートでレンダリングする方法として、サーフェル (Surfels: Surface Elements) を提案した [2]。サーフェルはオブジェクトの表面をサンプリングして得られる点に、深さや法線などの情報をもたせるとともに、点を中心とした円盤 (disk) をオブジェクトの接平面にそって配置し、その上にテクスチャをマッピングする方法である。サーフェルは点をプリミティブの主要構成要素とするため、接続情報を必要としない分、計算量的に有利である。なお、円盤の半径はすべてのプリミティブに共通であり、形状を覆う際に意図せずに行えるすきまをなくすように設定することで、連続体を表現している。

ところで、水や炎など連続体と不連続体を両方扱う場面を、サーフェイスとパーティクルなどそれぞれ異なる手法で表現すると、レンダリング方法も異なるため、それらを組み合わせる必要が生じる。メタボールは両者を統一的に表現する手法として知られるが [3]、粒子感を出さず場合、数多くのプリミティブを必要とし、抽出される等濃度面の表現に大量のパッチが利用される。またボリューム表現によっても、伝達関数の工夫で同様の視覚効果を上げることが可能であるが、その設定はたいへんむずかしい。

そこで本研究では、サーフェルのプリミティブが点を基本としていることに注目し、個々の点に異なる半径情報をもたせることによって、表面にすきまのない連続体と同時に、互いに離れて存在する不連続体も表現できるような拡張を提案する。

次節ではまず、サーフェルのオリジナルアルゴリズムの概要を紹介する。そして第3節でサーフェルの拡張方針と可能となる特長について述べる。第4節では、拡張されたサーフェルを用いた碎破へのモデリング例を示す。そして第5節で本論文をまとめ、今後の課題にふれる。

2 サーフェルの概要

サーフェルは主として、サンプリングとレンダリングの二つのフェーズに分けられる。本節はサーフェルにおけるサンプリング方法とその階層的データ構造について述べ、次にレンダリングの基本方針を示す。そして最後に簡単な具体例をあげる。

2.1 サンプリング

図1に、サーフェルにおけるサンプリングの概念図を2次元的に示す。サンプリングは視点と独立に、直交する三つの座標軸に平行な方向にそって行う。例えば x 軸と平行な方向で見ると、 y, z 軸の各方向とも h_0 の間隔で格子点ごとに視線をとばす。サンプリングを行う視線

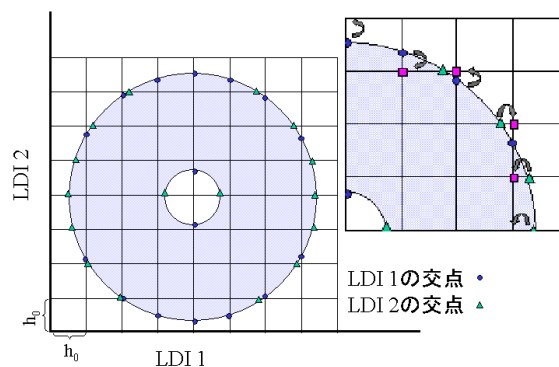


図1: LDC と格子点への移動

とオブジェクトが交わらないとき、その格子点の情報は空にする。視線とオブジェクトが交わるとき、その格子点に x 軸方向の深さ情報とともにオブジェクトサーフェスの法線を格納する。また、レンダリングに要する計算時間を短縮し、高品質な描画を可能にするため、この段階でサンプリングされたオブジェクトのテクスチャを円盤ごとにマッピングする。同様に直交する他二方向からも視線をとばし、すべての交点でオブジェクトの情報を抽出する。これらの情報を含む格子点が構成する yz, zx, xy 平面をそれぞれ LDI (Layered Depth Image) 1, LDI2, LDI3, これらの面から構成される立方体を LDC (Layered Depth Cube) とよぶ。このサンプリングは、ポリゴンパッチへの分割と異なり、接続情報をもたない。

なお、深さ方向の値が格子点上にない点は、データ削減のため最も近い格子点へ移動させ、格子点上では格子点に最も近い点の情報をとる方針 (3-to-1 reduction) が選択可能となっており、本論文ではこの方法を用いるものとする。

サンプリングで得られた格子点の情報はすべて、八分木構造をとる LDC 木という階層的なデータ構造に格納される。LDC 木は最下層をレベル 0 とし、 h_0 の間隔でサンプリングした情報を格納する。レベルが上がるごとに 2 倍の間隔でサンプリングした情報を格納するため、レベル i では $2^i h_0$ の間隔でサンプリングした情報を格納する。しかし、重複した格納はせず、レベル i にはレベル $i-1$ へのリンクを格納し、順に参照することで情報を得る。上の層から順に情報を見ていき、あるレベルで訪問しているノードが空であれば、対応する部分木にはサーフェスのプリミティブが存在しないのでスキップすることができる。階層的に格納することで、速度と画質のどちらを優先するかを選択が可能になっている。また、与えられたビューイングパラメタに対してピクセル間隔より細かい間隔のサーフェルは表示しないようにするためにも、このような階層的データ構造は有効である。

2.2 レンダリング

LDC 木に格納されたデータを用いてレンダリングを行う。レベルが i が設定されたとき、LDC 木の一番上の層から情報を探し、レベル i までのサーフェルの情報を得、レンダリングを行う。個々のサーフェルは連結情報をもたないため、表面にすきまが生じないようにオブジェクトを描画する必要がある。よって、描画されるサーフェルの中で、隣接しているサーフェル同士の距離を算出し、すきまのない半径の大きさを計算する。レベル i で隣り合う任意の一組のサーフェルの中心間の最大距離 s_{max}^i は $\sqrt{3}h_i$ となる。そのため、半径 r^i は $s_{max}^i/2$ に設定すれば十分である。すべての円盤で半径は一定なので、2次元では例えば図 2 のように描画される。

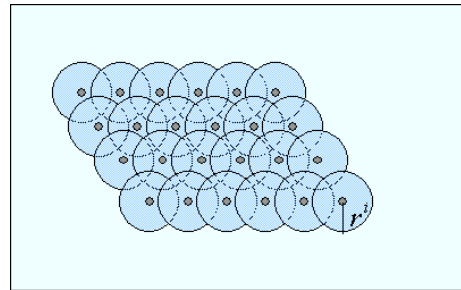


図 2: サーフェルの概念図

前処理で得たテクスチャを用いて、サーフェルにマッピングする。視点から見えない範囲は表示されないため、カリングする。また、Zバッファを用いて、投影されるサーフェルが重なって描写することを避ける。また、法線情報を利用して、通常のサーフェスレンダリングを実行することができる。

2.3 具体例

具体例としてスギ・ヒノキのような、円錐状の稠密な葉の分布をもつ立ち木によって遮蔽されたために、根元付近にだけ雪が降り積もらなかった雪原を、サーフェルを利用して表現することを考える (図 3)。縦横 48 × 64

の格子点をもつ 2 次元平面に，中心 (48, 44)，半径 16 の半円を配置する．レンダリングには OpenGL を使用し，C 言語で実装した．実装環境には SGI 社製の O2 (CPU:R10000, Clock: 195 MHz, RAM: 256 Mbytes) を用いた．

本来は，雪表面の微細構造を反映したテクスチャをマッピングすることもできる．しかし，本論文ではサーフェルのプリミティブが点

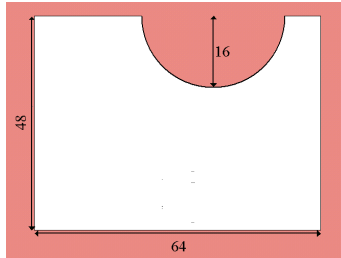


図 3: 雪原の 2 次元空間

あることに注目しただけであるため，簡単化するためにテクスチャは用いず，それぞれのサーフェルにすべて同じ色で，白を格納した．また，法線には摂動を加えた．三角形ストリップを用いて各プリミティブの円盤を表現し，スムーズシェーディングを利用した．さらに，遠近感をつけるため透視投影で画像を表示した．バンプマッピングを用いたため，同一色を用いているにも関わらず，光を照り返しによって，色の変化が描写された．木の周囲に積もる雪原を表現した結果を図 4 で示す．

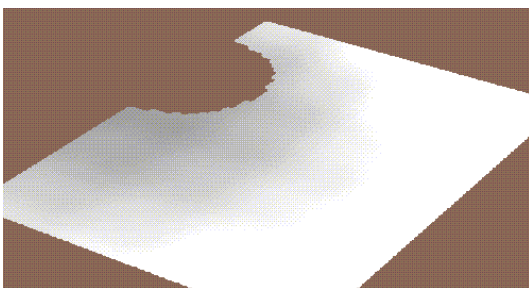


図 4: サーフェルによる雪原の表現

3 サーフェルの拡張

本節では不連続体も表現できるようにサーフェルの拡張を試みる．

3.1 サンプリング

サーフェルでそれぞれのプリミティブに格納するデータに，固有の半径情報を与えるような拡張を考える．上述したように，オリジナルのサーフェルでは，表面を覆うようにすべての円盤に対して共通の半径が与えられている．よって，連続体として表現されるプリミティブには，サーフェルと同様の半径を設定する．しかし，不連続体として表現されるプリミティブには，その半径より小さい半径を設定し，円盤と円盤の間にすきまをつくる．これによって個々の円盤の輪郭が描画されるようになり，不連続な点の表現ができる．

サンプリングでは，サーフェルと同じように，直交する三方向から視線をとばし，視線がオブジェクトと交わる点における情報を格納する．しかしここでは，プリミティブのデータとして，位置と法線の他に，半径も格納する．

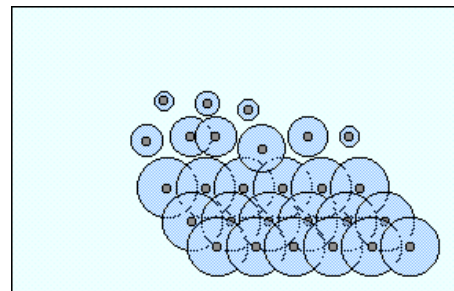


図 5: 固有半径をもつサーフェルの概念図

サーフェルの拡張のイメージ図を 2 次元で表すと図 5 のようになる．各半径は，オリジナルのサーフェルで設定される値以下の正数とする．そこで，半径を格納するとき，不連続体である範囲を明確にし，半径をどのように設定するかという規則が必要となる．また，格子点上に不連続な点を描画すると規則的に見えてしまうため，半径が小さく設定されている円盤は，摂動を与えて，格子点上からずらして描画する．

3.2 レンダリング

個々のサーフェルに与えられた位置や法線の情報を読み込み，設定されている半径でレンダリングを行う．プリミティブの主要構成要素が点であるという構成はオリジナルから不変なため，接続情報をもたず，不連続体を扱っても，時間計算量の増大を招くことはない．範囲外のオブジェクトのカリングもオリジナルのサーフェルと同様であるが，不連続体ですきまがあき，背面部分が見えるため，背面の消去は行わない．

3.3 拡張の具体例

図6に，図3を拡張して表現した例を示す．雪の描画では，先行研究として文献[4]があげられるが，そこではポリゴンパッチで表現しているため，やはり細かい描写をするには多くのパッチが必要となっている．また，パーティクルを使用することで細かい雪を表現している．ここでは，これらの統一的な表現を試みた．レベルを i とすると，元のサーフェルのプリミティブにおける円盤の半径 r_i は $\sqrt{3}h_0/2$ である．ここで，根元の中心からの距離を d とすると，個々に与えられる円盤の半径は，

$$\begin{cases} d \leq 10 \text{ のとき} & \text{NULL} \\ 10 < d < 18 \text{ のとき} & d^4/18^4 r_i \times e \\ 18 \leq d \text{ のとき} & r_i \end{cases}$$

のように設定した．ここで， e は，区間 $[0.5, 1]$ の一様乱数である．また，不連続な点として見える円盤の位置は，規則的に見えないよう，元の位置を中心に区間 $[-0.75, 0.75]$ の範囲で動くよう乱数を用いて配置している．このように設定されたサーフェルの情報は半径が異なることで，表現されるものは結果として異なるが，統一的なレンダリングを行うことができる．実際は，木の周辺における雪は不連続で，そこから離れたところでは連続な表面となるはずである．しかし，図4では，半径が一定なので，木の周囲の雪は明確に境界線がはっきりしている．それに対して，境界前後の範囲に，小さくラン

ダムな半径をもたせることによって，図6では境界線がぼけ，実際の吹き込みに近い表現ができています．

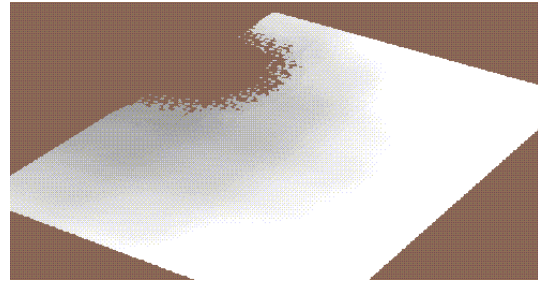


図6: 拡張されたサーフェルによる雪原の表現

4 適用例

連続体と不連続体が同じ場面に出てくる代表的な例として，海における砕波が考えられる．砕波はうねりやさざなみのような連続体である部分に加え，しぶきという不連続体がある[5]．そのため先行研究である Fournier，他は，Fournier，他はサーフェスによってうねりを，パーティクルによってしぶきをそれぞれ表現した[6]．図7にサーフェルの拡張によって，これまで異なる枠組みで表現されてきた連続なうねりと不連続なしぶきを統一的に表現した試みの結果を示す．

うねりの形状はトロコイドを用いた．また，しぶきはトロコイドの面において，高さが一定値より高いところとし，高さの3乗に比例し，雪と同様の摂動係数 e を利用し，半径を小さくした．色は，海の深い部分にいくほど暗く，また，浅いしぶき部分に近づくほど明るくなるよう，徐々に色の変化を加えている．不連続な点では，同じ高さに同じ色を設定すると不自然さが残るため，色にも摂動を加えた．しぶき部分は白く光るため，半径が一定値よりも小さくなったときは白く表示するようにする．

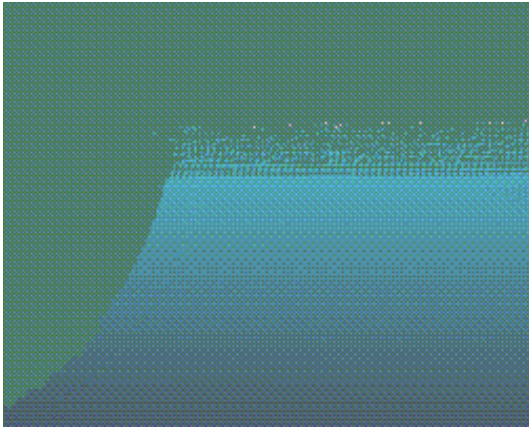


図 7: 拡張されたサーフェルによるしぶきを含む砕波の表現

5 結論

本論文では，サーフェルを拡張し，個々のプリミティブごとに，固有の半径情報ももたせることで，連続体と不連続体を統一的にレンダリングする方式を提案した．これによって容易に境界をあいまいに表現することが可能となった．

また，従来の砕波の表現ではサーフェイスとパーティクルのように異なる手法で表現するものが多かったが，うねりとしぶき統一的に表現することができた．

しかし，うねりの形状はきわめて単純化している上，しぶきの範囲の特定法も高さだけの考慮なので，より改良できると考えている．また，砕波においては，波の面速度よりも水粒子の速度の方が早いときにしぶきが発生するなど，より物理の法則に基づいた設定が可能なので検討を重ねていきたい．

サーフェルは，円盤にテクスチャを張り付けるため，拡張方法として，テクスチャを制御しているいろいろな効果を出すことができる．また，テクスチャに密度を加えることで，2Dにおけるメタボールのような役割をできるプリミティブを作ることによって，より境界の不明瞭な表現が可能になるとも考えている．

参考文献

- [1] P. E. Debevec, C. J. Taylor and J. Malik : “ Modeling and Rendering Architecture From Photographs: A Hybrid Geometry- and Image-Based Approach, ” In *Computer Graphics Proceedings, Annual Conference Series*, 1996, ACM SIGGRAPH, pp. 11–20.
- [2] H. Pfister, M. Zwicker, J. v. Baar and M. Gross : “ Surfels: Surface Elements as Rendering Primitives, ” In *Computer Graphics Proceedings, Annual Conference Series*, 2000, ACM SIGGRAPH, pp. 335–342.
- [3] 西村 仁志, 平井 誠, 河合 利幸, 河田 亨, 白川 功 : “分布関数による物体モデリングと画像生成の一手法”, 電子通信学会論文誌, Vol. J68-D, No 4, pp. 718–725, 1985.
- [4] P. Fearing : “ Computer Modeling of Fallen Snow, ” In *Computer Graphics Proceedings, Annual Conference Series*, 2000, ACM SIGGRAPH, pp. 37–46.
- [5] 檜山 奈保, 藤代 一成, 竹島 由里子 : “砕波 = うねり + さざなみ + しぶき”, 第 62 回情報処理学会全国大会, 7E-04, 2001 年 3 月
- [6] A. Fournier and W. Reeves : “ A Simple Model of Ocean Waves, ” *Computer Graphics*, Vol. 20, No. 3, pp. 75–84, August 1986.