

リアルタイムアニメーションで 均一なフレーム速度を達成する表示アルゴリズム

永塚 仁夫 大野 義夫

慶應義塾大学 理工学部

リアルタイムアニメーションで均一なフレーム速度を達成する表示アルゴリズムについて述べる。静止画像を単純に連続して生成するリアルタイムアニメーションシステムでは、光景の複雑さによってフレーム速度が変動してしまう。そこで、指定された目標フレーム時間において各フレームを生成し、さらに極力画質は高く維持するというアルゴリズムを提案する。このアルゴリズムはフレーム生成時間を拘束条件としながら、段階的簡略化により多重表現された各モデルに対して最適な細部レベルやシェイディングアルゴリズムなどの画質属性を選択する。

Display Algorithm to Obtain Uniform Frame Rates for Realtime Animation

Norio Nagatsuka, Yoshio Ohno

Faculty of Science and Technology, Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama-shi, Kanagawa 223 Japan

We describe a display algorithm to obtain uniform frame rates for realtime animation. Frame rates of realtime animation system will change depending on the complexity of the scene. Therefore, we propose a display algorithm to generate the "best" image for each frame within a specified target frame time. It chooses the most detailed level of shape description and the most sophisticated shading algorithm for each object, on the condition that the frame can be generated within the target frame time.

1 はじめに

リアルタイムアニメーションはユーザからの入力に対応したインタラクティブな画像を即座に表示できたり、対象物や視点の動きなどの変更が手軽に行えるという利点をもつ。しかしその反面、描画する光景が複雑になると1フレームの作成に時間がかかり、フレーム速度が遅くなるという問題が生じる。バーチャルリアリティにおいて、フレーム速度が変化すると、そのシステムのインタラクティブな感覚は減少してしまう。

フレーム速度を低下させない方法として、同一モデルを段階的簡略化によって階層的に多重表現し、細部レベルを各モデルごとに設定するという手法がある。この手法は生成画像の画質を一部犠牲にして描画するポリゴン数を減らし、フレーム速度を向上させる。その細部レベルの選択方法としては、ディスプレイ上でのモデルの大きさによって静的に選択する手法 [2]、フレーム時間を予測しながら選択する手法 [1]、視線制御によるもの [3] などがある。

本論文では、リアルタイムアニメーションでフレーム速度を均一に維持することを最重要課題とするアルゴリズムを述べる。このアルゴリズムは、あらかじめ指定された目標フレーム時間で一画像を生成することを目的とする。そのために、各描画対象物体の細部のレベルとシェイディングアルゴリズムを、毎フレームそれぞれ個別に設定する。本手法のアプローチは、基本的にはフレーム時間を予測しながら選択する手法 [1] と同様であるが、各物体に対して細部レベルだけではなくシェイディングアルゴリズムも設定を行うことによって、より均一なフレーム速度と画質の劣化抑制を達成する。また、リアルタイム処理に適したオーバーヘッドの少ないシンプルなものでありながら、有効に機能する手法となることを目的とする。

2 描画時間の予測

1フレームの生成にかかる実際の描画時間は、ワークステーションの種類など複数の要素に依存し、その対応関係は非常に複雑である。しかしながら、

使用するシステムが処理時間に関して最も依存している各要素を用いて、近似的な予測をすることは可能である。

今回の測定に用いたシステムで最も時間を要するのは、描画対象となる各ピクセルに対する処理 (Zバッファリング、輝度計算、表示) である。したがって、ある物体を描画するのに必要な時間は、その物体が画面上で占めるピクセルの数と線形であると仮定し、次式のように定義する。

$$\text{物体描画時間} = A_x * \text{ピクセル数} + B_x$$

ここで、 A_x, B_x は各細部レベル、シェイディングアルゴリズムとマシンに特有な一定係数であり、実験的に決定される。

さらに、1フレームの生成時間は描画対象となるすべての物体描画時間の累計となる。各フレームにおいて描画対象となる物体のセットを S とすると、本アルゴリズムの目的は、

$$\Sigma_S \text{物体描画時間} \approx \text{目標フレーム時間}$$

を達成することである。

3 画質の評価

画質を正確に評価するには、人間の認知力や理解力、視覚の特性等までも考慮しなければならないが、これらを正確にモデル化するのは非常に困難である。また、正確にモデル化できたとしても複雑になり、計算機上でリアルタイム処理するには不向きである。したがって、本手法では前節の描画時間の予測同様、幾つかの重要な要素だけを用いて近似的に評価を行う。

3.1 画質判断の要素

まず、画質判断の要素を物体の重要度と画質属性の2種類に大きく分類する。物体の重要度とはその物体が画面上でユーザに対してどれだけ重要であるかを示すもので、各要素として以下のものが挙げられる。

- 物体の画面上での大きさ
- 画面中心からの距離

一方、画質属性は画質を決定する基準となるもので、対象物要素集合が含む要素でもある。以下のものが挙げられる。

- 細部レベル
- シェイディングアルゴリズム

3.2 画質評価手法

今回のシステムでは、物体の画面上での大きさ、細部レベル、シェイディングアルゴリズムの3つを画質評価要素とし、その順に視覚への影響が大きいと仮定した。図1に画質評価の概念を示す。大きさを最優先で評価し、次に細部レベルを優先して評価する階層的な評価手法となっている。

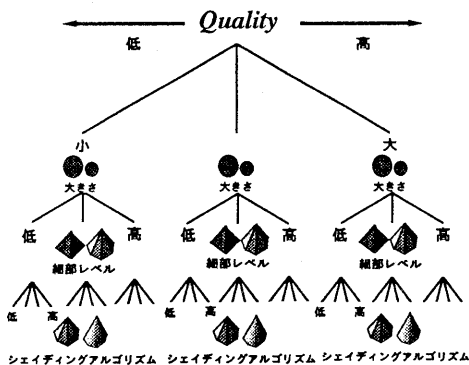


図1: 画質評価の構造

4 最適化表示アルゴリズム

本節では前節で定義した画質評価の概念を用いて、各フレームで全ての描画対象物体に対して細部レベル、シェイディングアルゴリズムをそれぞれ選択する最適化表示アルゴリズムを与える。その流れ図を図2に示す。

本論文の画質評価手法は、その構成要素の中で“物体の画面上での大きさ”への依存度が最も大き

い。そこで、描画対象物体のスクリーン上での大きさ（小ささ）の順に、画質属性の細部レベルとシェイディングアルゴリズムを操作、選択する。実際のアルゴリズムとしては、まず現在のフレームで選択されている描画対象物体のセット S の描画時間を予測し、予想フレーム時間を見積もる。

予想フレーム時間が目標フレーム時間より大きくなってしまった場合には、最も画面上で小さい物体のシェイディングアルゴリズムのレベルを1つ低くし、更新された画質属性を用いて予想フレーム時間を再予測する。この作業を予想フレーム時間が目標フレーム時間より小さくなるまで繰り返す。終了時の画質属性によって各物体を描画すれば、目標フレーム時間でフレームが生成できる。シェイディングアルゴリズムが最低になっても終了しない場合には、細部レベルを1つ低くしシェイディングアルゴリズムを最高レベルにして繰り返す。さらに細部レベルとシェイディングアルゴリズムの両方が最低になっても終了しない場合には、次に小さい物体に対して同様の処理を施す。

反対に、最初の予想フレーム時間が目標フレーム時間より小さい場合には、画面上で最も大きい物体から順に、シェイディングアルゴリズムと細部レベルのレベルを高めていく。そして、予想フレーム時間が目標フレーム時間より大きくなったときの、その1つ手前の画質属性を用いてフレーム生成を行う。

5 適用例

5.1 テスト方法と測定結果

ワークステーション上に簡単なフライトシミュレータを作り、本論文で提案した最適化表示アルゴリズムを用いた場合と用いない場合について、同一の飛行進路を設定してテストを行った。

物体として設定した山の細部レベルは3段階とし、シェイディングアルゴリズムは輝度計算なし、フラットシェイディング、Gouraudシェイディング、Phongシェイディングの4段階を設けた、

図3、図4にテストコースに沿った各観察者視点

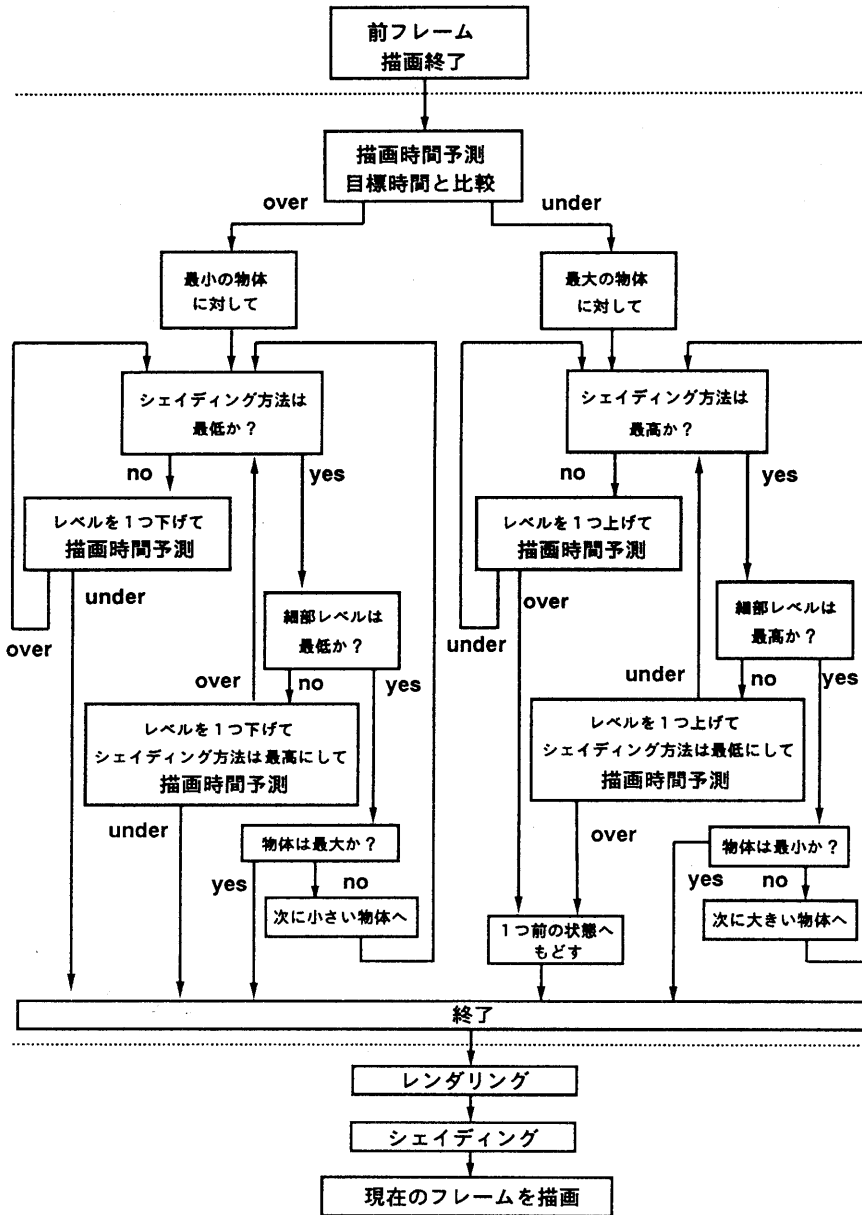


図 2: 最適化表示アルゴリズム

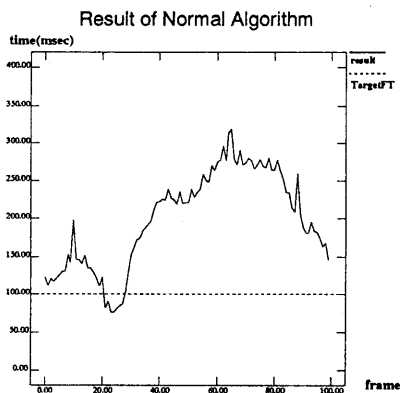


図 3: 簡略化なし

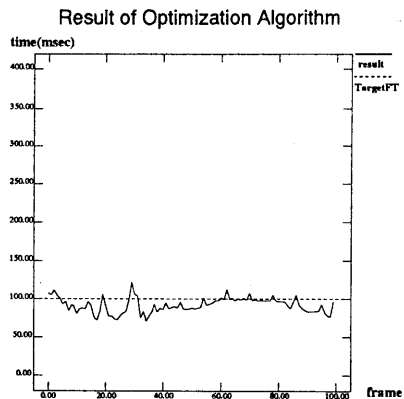


図 4: 最適化表示

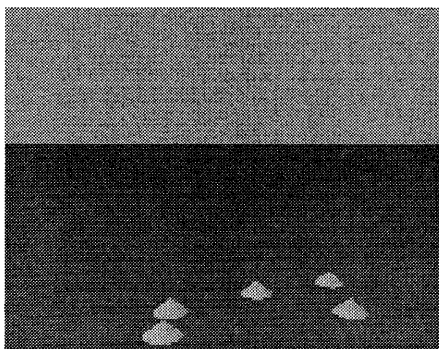


図 5: 簡略化なし

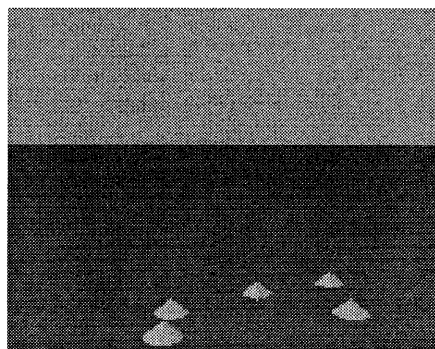


図 6: 最適化表示

に対するフレーム生成時間と目標フレーム時間との比較を示す。図3は簡略化をせずに使用できる最高画質で全フレームを作成した場合であるが、各フレームごとにフレーム時間が大きく異なりフレーム速度が均一にならない。一方、図4は本アルゴリズムによって最適な簡略化を施した結果であり、各フレーム時間を目標フレーム時間近辺に維持し、均一なフレーム速度を達成できている。

5.2 結果の検討

本論文で提案した最適化表示アルゴリズムを用いて生成したフレーム時間は、用いない場合と比較してより均一なものとなる。

最適化表示アルゴリズムは、均一でインタラクティブなフレーム速度を維持するために画質を調整しながら、各観察者視点に対して目標フレーム時間内で極力“ベスト”な画像を描こうとする。簡略化なしを用いた場合、および最適化表示アルゴリズムを用いて簡略化を施した場合における10フレーム目の画像の比較を図5、図6に示す。図5を描画するには0.199秒を要するが、図6では0.080秒ですんでいる。画質に関する比較をすると、図6では小さい山2つに最も低い細部レベルが用いられているのが少し目立つ。今回は画質評価の要素としてスクリーン上の位置(中心からの距離)を考慮していないので、目立つ位置に描画される物体でも低い画質属性が選択されてしまった。しかしながら、これ

らはスクリーン上で小さい山から順に適用されている。また、最高の細部レベルと次のレベルとの差や、シェイディングアルゴリズムによる差異は画面上ではほとんど感じられない。結果として、最適化表示アルゴリズムを用いても、簡略化をしなかった画像に遜色ない結果が得られる。

6 まとめと今後の課題

本論文ではリアルタイムCGアニメーションにおける、速くて均一なフレーム速度のための表示アルゴリズムについて述べた。そのアルゴリズムは、目標時間で1フレームを作り出すために、各可視性物体を描写するのに用いる細部レベルとシェイディングアルゴリズムを選択する。インタラクティブな視覚化アプリケーションにおいて実験を行い、同一物体に対する異なる表現が同じように見え、表現間の移行が非常に目立つものでない限りは、画質の変化はフレーム時間の変化よりもユーザを不快にさせないことを確認した。したがって、本アルゴリズムは「ユーザがいかにインタラクティブな感覚を感じられるか」を主眼としたシステムを構築する場合に重要な役割を果たす。

テストにより、本論文で提案した最適化表示アルゴリズムがモデルの視覚化において、画質の顕著な差異はほとんどなしに、均一なフレーム速度を達成できることが確かめられた。

今後の課題としては以下に挙げるような項目を考えている。

- レンダリングアルゴリズム等の画質属性およびその選択肢の追加。
- 計算機の負荷状態を常時調査し、時間予測アルゴリズムのパラメータを更新してより正確な予測を可能にする。
- 画質属性の選択時に、前フレームまでの画質属性の変化の履歴も考慮する。
- 段階的簡略化モデルのための形状データ自動生成アルゴリズム。

参考文献

- [1] Funkhouser, Thomas A., and Sequin, Carlo H., "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments," *ACM SIGGRAPH '93 Proceedings*, pp. 247 - 254, 1993.
- [2] 北州克寛, 遊佐洋子, "リアルタイム景観シミュレータのための形状のグルーピングと多重表現に基づく描画時間の短縮," 電子情報通信学会論文誌, D-II, Vol. J77-D-II No. 2, pp. 311 - 320, 1994.
- [3] 大島登志一, 内山普二, 山本裕之, 田村秀行, "視線制御を用いた階層型ポリゴンデータの適応的実時間表示法," *グラフィクスとCADシンポジウム論文集*, Vol.94, No. 7, pp. 49 - 59, 1994.