

OpenGL を用いたリアルタイム立体視光シミュレータ

北川 智也

Tomoya KITAGAWA

1 はじめに

我々の研究室では、利用者各人に任意の光環境を提供する知的照明システムの研究を行っている。これをシステム導入前のシミュレーションや、研究室外でのデモンストレーションに用いることから、コンピュータグラフィックス (以下、CG) を用いた知的照明システムの光環境シミュレータの作成も行っている。本研究では、知的照明システムの光環境 CG をリアルタイムにレンダリングし、仮想空間内を自由に歩き回れるようなシミュレータの作成を行う。リアルタイムレンダリングとは、外部からの入力を随時 CG に反映し動画の様に表示することを指す。1 秒間にレンダリングできる CG の数で描画速度を測り (Frames Per Second), 30fps 以上であればリアルタイムレンダリングと言える。しかし CG において光の計算は大きな負荷となる。そのためリアルタイム性を保持するには光源の数を制限したり、近似的な計算を行う必要がある。本研究では、知的照明システムの様に多数の照明を用いた CG のリアルタイムレンダリングを OpenGL と GLSL を用いて実現する。また近年、ホログラフや 3D ディスプレイを用いた、立体視による CG のリアリティ向上に関する研究も盛んに行われている¹⁾。本研究でも 3D ディスプレイを用いた CG の立体視化を行うことでリアリティを高める。

2 OpenGL

2.1 OpenGL とは

OpenGL (Open Graphics Library) は Silicon Graphics 社が開発したグラフィックスハードウェア用の API である。3 次元 CG の描画に必要なオブジェクト生成や陰影処理、計算手順などの関数が用意されている²⁾。利点としてクロスプラットフォームな API であるため多くの OS で実行可能な点や、オープンソースである点などが挙げられる。また、非常に高速なシーン描画が行えるため、リアルタイムレンダリングも可能である。

2.2 GLSL

1 章で述べたように、OpenGL ではリアルタイムレンダリングを行うため、光源の数に制限を設けている。従って、OpenGL で用意された機能だけでは知的照明システムのような多数の照明を使用した光環境を表現できない。そこでプログラマブルシェーダを用いてレンダリング機能を拡張する。

シェーダとは、CG を画面に表示するための座標変換方法を定義したレンダリングパイプラインである。コンピュータで設定されたシーンに関するデータをシェーダ

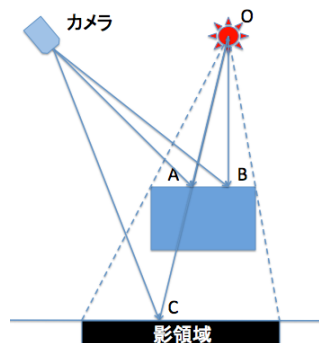


Fig.1 シャドウマップ法のイメージ

内で設定された演算に従って変換し、画面に表示する。従来はグラフィックスハードウェアに組み込まれたものであったが、次第に CG の開発目的が多様化し、ハードウェアに固定された機能だけでは非効率的であることからシェーダの設定を開発者が自由に行えるようになった。

GLSL (OpenGL Shading Language) はプログラマブルシェーダの 1 種であり、OpenGL で使用する機能を記述することができる。GLSL はオブジェクトの頂点位置や距離などの座標や空間情報に関する計算を行うパーテックスシェーダと、オブジェクトの色や光の反射などをピクセルごとに計算するフラグメントシェーダからなる。本研究ではフラグメントシェーダでピクセルごとに複数の光源を考慮した色づけを行う。

3 シャドウイング

3.1 CG における影

CG において影の描写はリアリティ向上に必要不可欠である。しかし、影の描写は光源や物体の位置を考慮した複雑な計算であり、大きな処理時間を要する。従って、リアルタイムレンダリングで影を描写するためには効率的な計算方法が必要であり、多数の手法が提案されている^{3) 4)}。本実験では、リアルタイムレンダリング用の影付け手法としてシャドウマップ法を採用する。

3.2 シャドウマップ法

シャドウマップ法⁵⁾ はリアルタイムレンダリング用に提案された CG の影付け手法の 1 つである。物体の形や光源の位置を考慮した影を付けられることや、物体が自分自身に落とすセルフシャドウを描写できるなど、従来手法の欠点を克服でき、さらに高速性も保持されることから注目を集めている。

具体的な処理を Fig. 1 を用いて説明する、はじめに光源を視点として目的のシーンを描画する。このとき、光源から最も近い物体までの距離を各ピクセル方向に対し

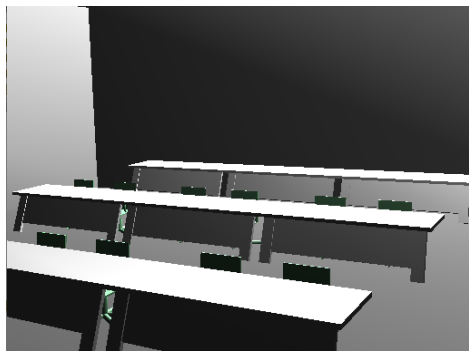


Fig.2 実験に用いたシーン

て計測し、シャドウマップと呼ばれるバッファに格納する。図中の点 A と C のように同一方向に存在する点は OA の距離のみ保存される。次にカメラからのシーンを描画する。このとき、目標のピクセルから光源までの距離とシャドウマップに保存されている距離を比較し、その大小によってそのピクセルが影であるかどうかを判定する。OA や OB の距離はシャドウマップに格納された距離と等しいため影領域ではないという判断がされる。一方 OC の距離はシャドウマップに格納された距離 OA よりも大きいため、点 C は影領域であると判定される。

4 立体視

人間の目は左右に 2 つ水平に並んで付いているため、左目と右目では得られる視覚情報が異なる。この両目から得られる情報の違い(視差)によって物体までの距離を認識し、立体的にとらえることができる。現在一般的に使用されている 3D ディスプレイや映画ではこのような視差画像を出力し、右目と左目に異なる情報を与えることで立体視としている。本研究でも 2 枚の視差画像を用いて立体視シミュレータを生成する。目的のシーンを水平に少し距離を空けて並べた 2 つのカメラによって描画し、得られた 2 つシーンを同時に出力することで立体視化する。

5 実験

5.1 実験概要

3.2 節、4 章で述べたことをふまえ、Fig. 2 に示すような部屋のモデルをリアルタイムにレンダリングが可能かどうかの描画速度の計測を行った。本実験で用いた実行環境は Table 1 の通りである。モデルのポリゴン数は約 12000 である。このモデルを立体視するためにカメラを 2 つ用いて 2 種類の画像を描画する。リアルタイムレンダリングは 30fps 以上とされるため、カメラを 2 つ用い

Table1 実験環境

OS	Mac OS X 10.7.4
CPU	Intel Corei5-2435M 2.4GHz*2cores
GPU	Intel HD Graphics 3000
OpenGL version	2.1 APPLE-7.18.18
GLSL version	1.20

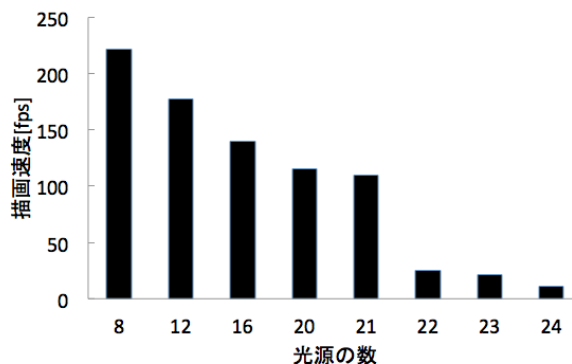


Fig.3 20 秒間の平均描画速度

る場合は全体の速度が 60fps 以上でリアルタイムレンダリングが可能と言える。実験では光源の数を変化させながら、20 秒間のシーンレンダリングの平均描写速度を求めた。

5.2 実験結果

結果を Fig. 3 に示す。光源数が 21 個までの場合、目標の 60fps を越えており、リアルタイムレンダリングが出来ていると言える。しかし、実際の知的照明システムにはさらに多くの光源を用いるため、プログラムの効率化によって光源をさらに増やした場合でもリアルタイム性を保持することが必要である。

6 今後の展望

5.2 節で述べた様に、知的照明システムのリアルタイム光環境シミュレーションを行うためにはプログラムの効率化が必要である。提案手法としてまず、シェーダを用いた描画演算の最適化が挙げられる。シェーダは CG の座標変換や計算精度を自由に記述できる。従ってこれらの値を見直すことで効率化を図る。

さらに、CG 描写の正確さを場合にに応じて変化させるといった手法も挙げられる。知的照明システムの仮想空間内を自由に動き回ることに関して、大切なのは静止中であり、移動中の CG 描写の正確さは重視されない。よって、移動中は影の描写をなくしたり、光源の数を減らすなど、処理量を減らすことで効率化を図る。

参考文献

- 1) 中嶋正之. CG 領域の動向, 2011.
<http://ci.nii.ac.jp/naid/110008762600/>.
- 2) Mark Segal, Kurt Akeley. *OpenGL Graphics System: A Specification*. 2012.
- 3) F.Crow. Shadow algorithms for computer graphics. *SIGGRAPH'77*, pp. 242-248, 1977.
- 4) C.Everitt. Projective texture mapping. *Nvidia technical report*, 2001.
- 5) L. Williams. Casting curved shadows on curved surfaces. *SIGGRAPH'78*, pp. 270-274, 1978.