

汎用高速 3D ライブラリ:GPUPPUR

—リアルタイム 3D の高速化、汎用化に挑戦—

1. 背景

3DCG には主にリアルタイム CG とオフライン CG の 2 つの分野がある。リアルタイム CG では、3D ゲーム等でのユーザの入力等の影響を受けて変化するような映像を扱っている。その為 1 秒間に 30 回以上画面を更新してスムーズな表示を行なえるよう常に 3D レンダリングの計算を行なっている。リアルタイム CG を使ったソフトの多くは一般的なユーザの PC で動作する必要があることと、短時間に計算を行なう必要があることから、あまり複雑な処理を行なうことができない。

一方オフライン CG では、映画等のように一定の映像を表示する。一度画像を作ってしまうと、映画館やテレビ等で 3D に関する計算をせず映像をそのまま表示することができる。そのため、CG のレンダリングを長時間行なうことや、値段の高いコンピュータを使って計算を行なうことができるためにリアルタイム CG よりもはるかに複雑な計算を行なうことができる。

現在、リアルタイム CG には主にラスタライズ、オフライン CG には主にレイトレーシング法が用いられている。ゲームで表示される 3D グラフィックスはどんどん複雑でリアルなものへと発展している。また、CPU や GPU も発展している。このまま発展してゆくと、現在のラスタライズ法からレイトレーシング法が使われるようになるのではないかと考えられる。現在ゲームで使われている量の三角形はレイトレーシング法よりもラスタライズ法のほうが速く処理を行えるが、三角形の量が増加していくといつかはレイトレーシング法のほうが速く処理を行えるようになると考えられる。なぜなら、ラスタライズ法では処理時間が三角形の数に比例するのに対してレイトレーシング法では処理時間が三角形の数に対数的に増加するからである。

2. 目的

ラスタライズ法もレイトレーシング法も扱うことのできる 3D レンダリングソフト、GPUPPUR を開発した。

GPUPPUR の目標を以下に示す。

- ・ ライブラリ利用者の創造性を発揮することができ、独創的な映像を生み出すことができる。
- ・ カスタマイズや特殊な機能を使わない限りは、様々なプラットフォームへ簡単に移植すること、様々なハードウェアに対応することを可能にする。
- ・ カスタマイズすることによって、特定のハードウェア、OS 上でしか動かすことができないが、ハードウェアの能力をフルに発揮できるアプリケーションの開発を可能にする。

3. 開発の内容

3.1 GPUPPUR と GPUPPUR 関連プログラムの構成

GPUPPUR は複数の方法を使って 3D を表示することができる。本プロジェクト

ではレイトレーシング法を使う GPUPPURay とラスタライズ法を使う GPUPPURas を開発した。GPUPPURay、GPUPPURas にはそれぞれ OpenGLを使ったものとDirect3Dを使った実装を開発した。それらの実装はすべて PhysX を扱うクラスを含んでいる。

画面に画像を表示するためには通常ウィンドウを作る必要がある。GPUPPUR の開発、テストを行なうために、煩雑なウィンドウ管理やマウス、キーボードからの入力処理を抽象化するためのライブラリとして GPUPPURUT を開発した。これは OpenGLにおける GLUT、Direct3Dにおける DXUT に相当する。GPUPPURUT はそれらのライブラリよりもまだ機能は少ないが、それらのライブラリの代わりとして使用することもできる。

3D プログラミングで必要となる n 次元ベクトルや n 行 m 列の行列を扱うテンプレートクラス、テンプレート関数(VectorNd, matrixRxC)を開発した(n, m は自然数で 0 を含まない)。それらの機能は OpenGL や Direct3D の関数を使用するときにも使うことができるように作られている。

GPUPPUR のテストプログラム兼デモプログラム、gpuppur_tester を開発した。gpuppur_tester では GPUPPUR を使って 3D 映像を表示するなどして、GPUPPUR の機能が正しく動作するかを検証できるように作られている。表示する物体を変更したり、動き方を設定するには直接 gpuppur_tester のプログラムを変更し再コンパイルする必要がある。将来的には外部ファイルに表示する物体のデータなどをすべて保存し、プログラムの再コンパイル無しに設定を変更できるようにする予定である。

以上の構成をわかりやすく図示したものを図 1 に示す。

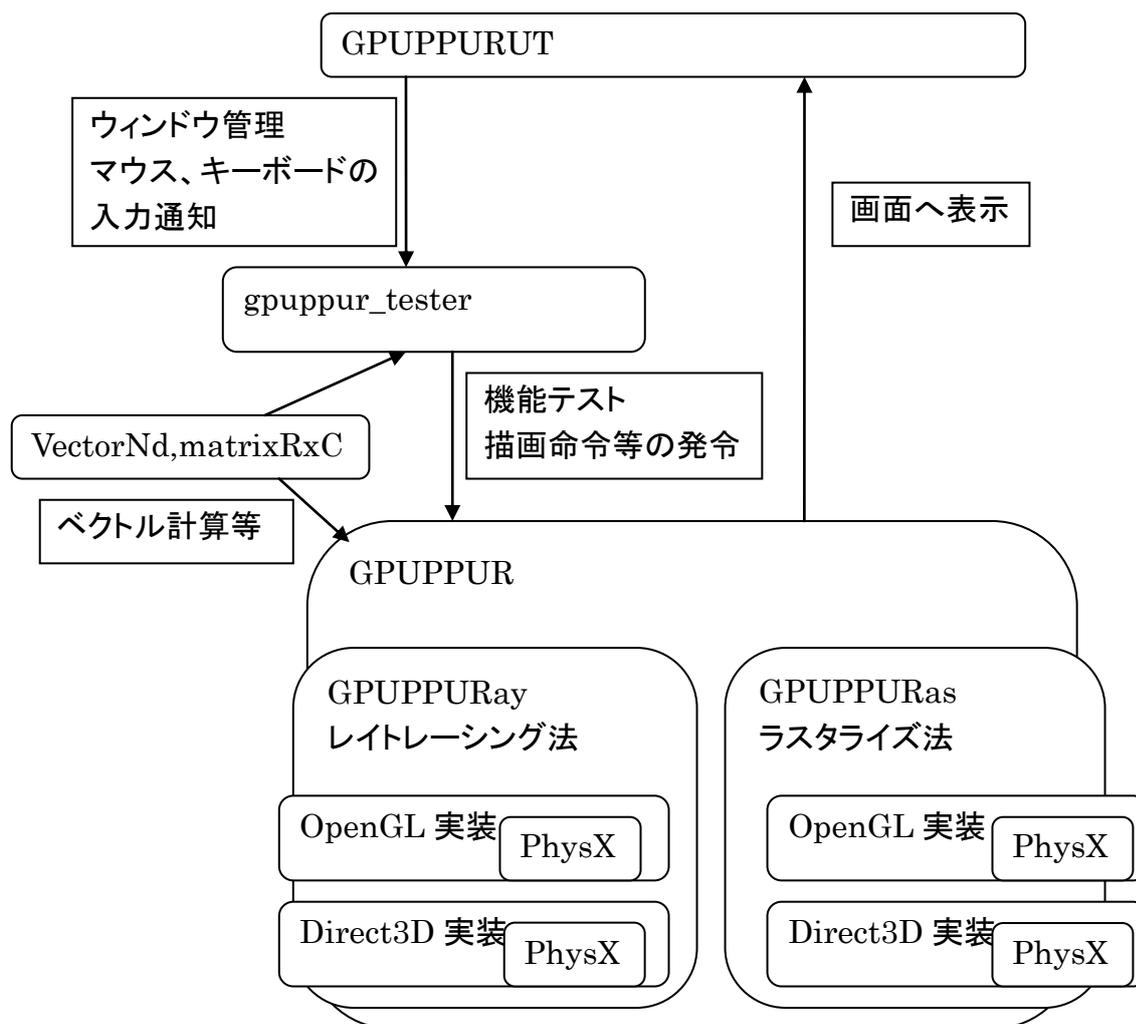


図 1 GPUPPUR とその関連プログラムの構成

4. 従来の技術(または機能)との相違

GPUPPUR はレイトレーシング法とラスタライズ法の両方を使うことができる。リアルタイム 3D でよく使われている OpenGL や Direct3D はラスタライズ法のみを使うようように設計されており、ラスタライズ法に適したインターフェースを持つ。だがそのインターフェースはレイトレーシング法にはあまり適していない。GPUPPUR ではレイトレーシング法でもラスタライズ法でも同様のインターフェースから機能を利用することができ両方の機能に適したものとなっている。

5. 期待される効果

レイトレーシング法とラスタライズ法を組み合わせることによって、レイトレーシング法で効率良く表現できる技法と、ラスタライズ法で効率良く表現できる技法を同時に使用することができる。よって、様々な表現技法をより効率的に処理することができると考えられる。そのことによって、斬新な映像をリアルタイムに表示できるようになるだろう。

6. 普及(または活用)の見通し

現在はまだ GPUPPUR は実用的なライブラリにはなっていない。もし、実用化することができれば、3D ゲーム、3DCAD、3D モデリングソフト、シミュレーション等のリアルタイム 3D が使われている分野で活用することができるだろう。

GPUPPUR 開発時に作成したベクトル、行列計算クラスや GPUPPURUT はこれから 3D プログラミングを始めるときに役立つかもしれない。そのためにはさらに充実したドキュメントを作成する必要がある。

7. 開発者名(所属)

松本 知大(デジタルファッション株式会社 技術開発部)

(参考)開発者URL

<http://d.hatena.ne.jp/gpuppur/>

GPUPPUR の開発者のブログ。およそ月に一回の更新。

<http://sourceforge.jp/projects/gpuppur/>

Sourceforge.jp の GPUPPUR のプロジェクトページ。GPUPPUR のソースコードをここで入手できる。

<http://www.comp.pu-toyama.ac.jp/gpuppur/>

元々、AGEIA 社に GPUPPUR の存在を知ってもらうために作成したサイト。GPUPPUR の説明やデモ等がある。ほとんど更新していない。