C 2 1 9 マルチプラットフォーム可視化APIの設計と開発

小野 謙二 (理化学研究所), 吉川 広幸 (富士通長野システムズエンジニアリング (株))

Design and Development of Multi-Platform Visualization API

Kenji ONO and Hiroyuki YOSHIKAWA

ABSTRACT

This paper reports a newly designed visualization API (Application programming Interface) for a Cartesian and octree data structure which allows us to render and visualize the data efficiently. It is designed that this visualization API consists of a window frame class, a scene-graph class and a visualization function class. The window frame class is constructed from wxWidgets, which is open source GUI system, so as to be operated on major platforms such as Linux, Windows and MacOS X. As concerns the scene-graph class, we developed an original API instead of the standard scene-graph library like VTK because of seeking of the rendering efficiency and supporting the special data structure. Based on this API, a rendering application was developed and evaluated. This data viewer will be delivered from RIKEN.

Keywords: Rendering, Visualization, Simulation, Application Programming Interface

1. はじめに

近年の計算機の能力の増大により、PCや小規模な計算機システムでも中規模程度のシミュレーションが可能になっている。PCとグラフィックスカードの性能向上は扱えるデータ量の増加をもたらし、3Dグラフィックスアプリケーション上で手軽にデータを可視化する基盤要素は整いつつある。可視化システムを稼働させるプラットフォームとしては、ハードウェアとOSから分類して、Windows, Linux, そしてMacOS Xが主要な候補となるだろう。エンドユーザから見た場合に、どのプラットフォームでも同様に動作する可視化システムは、データの可搬性も含めて、利便性が高い。

可視化システムにおいて、扱うべきデータ構造の選択はシステム開発および動作効率に大きく影響する因子である。提案のAPIでは、汎用性と効率が高い直交等間隔格子で定義されるボクセルデータと局所的に細分化が可能な木構造であるOctreeデータ構造を対象にする。

本稿では、マルチプラットフォームで稼働する可視化APIの概念設計について報告する。

2. 設計概念

可視化を始めとして、CADやモデリングなどのアプリケーションは、工業製品の設計過程には必須のツールであり、3Dグラフィックス機能を最大限利用するソフトウェアである。3Dグラフィックスの基本部分はどのアプリケーションでも共通に利用できるため、共通する機能を纏めて独立したライブラリとし、この上にア

Fig. 1 Components of present API

プリケーションを構築する方針とする。

3Dグラフィックスアプリケーションでは、描画機能とともに、3次元空間に存在する物体を描画するのに必要な場の情報をデータベース化したシーングラフが重要な機能となる。シーングラフの実装は、表示データを木構造で管理し、階層的な操作を可能とする方法が一般的である。シーングラフを導入したグラフィックスライブラリは、OpenGL Performer, OpenInventor, OpenSG, VTK等、多数存在する。特にVTK¹⁾では、シーングラフ機能に加え可視化のAPIも実装しており、オープンソースでもあるため利用者が多い。しかしながら、汎用的に作成しているため、動作効率に関しては高くない。提案のAPI開発では、描画対象とするデータ構造として、陰関数やOctreeデータ構造等、既存のシーングラフライブラリではサポートされていないデータ構造を扱う点と、動作効率の点を考慮して、シーングラフは独自に実装した。

対象とするプラットフォームは主要な3種、つまりWindows, Linux(X11), MacOS Xを考えている。これら全てのプラットフォームをサポートするため、グラフィックスライブラリにはOpenGL, ウィンドウGUIライブラリとしてwxWidget²⁾を利用し、アプリケーションフレームワークを実装した。APIの構成は、図1のようである。

3. API設計

Visualization APIはC++のクラスライブラリとして実装し、下記の3種類のクラス構成とした。

- (1) ウィンドウフレームクラス
- (2) シーングラフクラス
- (3) 可視化機能クラス

3.1 ウィンドウフレームクラス

ウィンドウフレームクラスは、下記の4つの機能を提供する。

1) DrawArea

描画領域機能の基底クラスで、レンダリングコンテキストとScreenを保持し、サイズ変更、再描画、キー入力、マウス操作のイベントを受け付ける。

2) Screen

ビューポート機能を実装するクラスで、DrawArea内のビューポート情報を管理し、参照するカメラから見たシーンの描画を実行する。

3) Action

イベントハンドラの基底クラスである。各イベントに対する実際の動作は、このクラスを継承するユーザ独自のクラスを作成し、イベントクラスに登録することによりイベント駆動型のアプリケーションを構成する。

4) DrawAreaWx

wxWidgetsを用いて実装したDrawAreaクラスである。

3.2 シーングラフクラス

1) ベース

シーングラフ構成要素の基底クラスで、幾何変換マトリックスとグラフの連結機能を実装する。

2) ノード

描画オブジェクトの基本クラスであり、ID、名前、頂点、法線、色、ディスプレイリストの管理機能を実装する。

3) グループ

グルーピング機能を実装したノードクラス。子供として複数のノードを持ち、シーングラフの枝別れを表現する。

4) シーン

シーングラフのルートにあたるグループクラス。

5) カメラ

視界(カメラ)機能の基本クラス。投影情報と視点データを管理し、シーンを参照する。

3.3 可視化機能クラス

1) データクラス

数値データに対応するクラス群で、ファイル入出力とデータの管理を行う。ボクセルデータクラス、Octreeデータクラス、ジオメトリ(WaveFront OBJ, STL)等のクラスが存在する。グループクラスを用いて実装されており、次の2)のメソッドクラスは子ノードとして登録される。

2) メソッドクラス

各データクラスに対応する可視化機能を実装するクラス群である。スライス表示、流線表示、データプローブ等のクラスが存在する。ノードクラスとwxWidgetのウィジェットを用いて実装される。

4. サンプルビューア

図2に評価用として作成したデータビューアのスクリーンイメージを示す。同時に5つのウィンドウを表示し、4つの異なるオブジェクトを表示している。マルチウィンドウ・マルチビューの機能により、効率的にシミュレーション結果の比較が可能となる。評価版の可視化機能として、分布図、等高線、ベクトル、流線と幾何形状の表示がある。

Fig. 2 Sample viewer with function of multi-window/view function

5. おわりに

マルチプラットフォーム上で稼働する3Dグラフィックスアプリケーションを作成するために、描画関連の機能を纏めたAPI群を設計し、直交等間隔とOctreeデータ構造に対応したデータビューアを開発した。このデータビューアは、理化学研究所から配布予定である。

参考文献

- 1) <http://public.kitware.com/VTK/>
- 2) <http://www.wxwidgets.org/>