

ジオメトリ処理

杉浦 拓海

● 線形代数との関係

近年コンピュータやスマートフォンのグラフィック性能の目覚ましい向上により、3次元コンピュータ・グラフィックス(3DCG)を用いたアプリや映像作品が身近にあふれています。一般の読者になじみがあるのは、今年アカデミー賞を獲得した実写映画でも用いられている視覚効果(VFX)や、3Dのオープンワールド系スマートフォン・ゲームなどでしょうか、これらの表現方法としてCG技術が使われています。IT系エンジニアの読者にはBlenderやUnity, Unreal Engineなどになじみがあるのではないのでしょうか。

3DCGとは主に3次元のオブジェクトを2次元の画面上に表示する技術のことを指します。この3次元から2次元へ投影する処理過程ではベクトルや行列などの線形代数が多用されます。本稿では3DCGの仕組みを知る第1歩として、3次元のオブジェクトとして最もシンプルな3次元点群を、2次元画像として描画する処理の流れを学んでいきます。実際に動作するスクリプトを通して、3DCGにおいて線形代数がどのような場面で応用されているのか解説します。

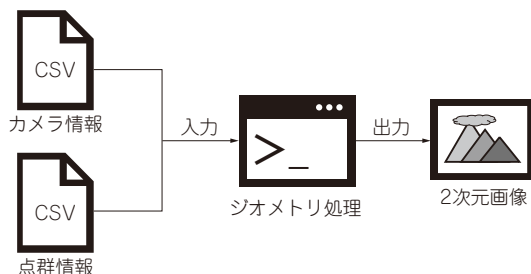


図1 今回作成するスクリプトのイメージ

● 作成するスクリプトのイメージ

今回作成するスクリプトのイメージを図1に示します。入力として点群の情報(CSVファイル)と、カメラの位置や姿勢などの情報を持つCSVを読み込み、カメラに映る点群を画像として出力するPythonスクリプトを作成します。点群/カメラCSV内の位置情報や姿勢情報を書き換えることで、任意の位置・角度から点群をレンダリングすることができます。

7-1 3DCG作成過程の概要

3次元オブジェクトの情報を入力として、2次元の画面上に表示する処理のことをレンダリングと呼び、レンダリングの処理過程をレンダリング・パイプラインと呼びます(入力データに対して、一連の処理を経て出力を行う過程を、一般的にパイプラインと呼ぶ)。

3次元情報から2次元画像を作る流れ

レンダリング・パイプラインの処理内容はフレーム・ワークごとに異なりますが、1つの三角面オブジェクトを画像として描画するまでに大まかに図2のような処理が行われています。

▶①ジオメトリ処理(MVP変換とビューポート変換)

ジオメトリ処理ではオブジェクトの各頂点が2次元の出力画像上のどの位置に映るのかを求めます。本稿

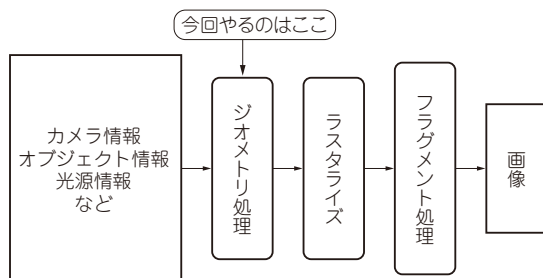


図2 レンダリング処理全体の流れ(レンダリング・パイプライン)