

メディアテクノロジーシリーズ 12

# Unityによる 3DCGプログラミング

— 基本原理と実践 —

藤代一成

【編】

石飛晶啓・杉田俊平・徳永恵太・青山昂平

【共著】

コロナ社

## メディアテクノロジーシリーズ 編集委員会

---

編集委員長	近藤 邦雄 (元東京工科大学, 工学博士)
編集幹事	伊藤 貴之 (お茶の水女子大学, 博士 (工学))
編集委員 (五十音順)	五十嵐悠紀 (お茶の水女子大学, 博士 (工学))
	稲見 昌彦 (東京大学, 博士 (工学))
	牛尼 剛聡 (九州大学, 博士 (工学))
	大淵 康成 (東京工科大学, 博士 (情報理工学))
	竹島由里子 (東京工科大学, 博士 (理学))
	鳴海 拓志 (東京大学, 博士 (工学))
	馬場 哲晃 (東京都立大学, 博士 (芸術工学))
	日浦 慎作 (兵庫県立大学, 博士 (工学))
	松村誠一郎 (東京工科大学, 博士 (学際情報学))
	三谷 純 (筑波大学, 博士 (工学))
	三宅陽一郎 (株式会社スクウェア・エニックス, 博士 (工学))
	宮下 芳明 (明治大学, 博士 (知識科学))

(2023 年 5 月現在)

## 編者・執筆者一覧

---

編者	藤代 一成	
執筆者	石飛 晶啓 (2 章)	杉田 俊平 (4 章, 5 章)
	徳永 恵太 (3 章)	青山 昂平 (1 章, 6 章)

---

## 刊行のことば

“Media Technology as an Extension of the Human Body and the Intelligence”

「メディアはメッセージである (The medium is the message)」というマクルーハン (Marshall McLuhan) の言葉は、多くの人々によって引用される大変有名な言葉である。情報科学や情報工学が発展し、メディア学が提唱されたことでメディアの重要性が認識されてきた。このような中で、マクルーハンのこの言葉は、つねに議論され、メディア学のあるべき姿を求めてきたといえる。

人間の知的コミュニケーションを助けることができるメディアは生きていくうえで欠かせない。このようなメディアは人と人との関係をより良くし、視野を広げ、新しい考え方に目を向けるきっかけを与えてくれる。

また、マクルーハンは「メディアはマッサージである (The medium is the massage)」ともいっている。マッサージは疲れた体をもみほぐし、心もリラックスさせるが、メディアは凝り固まった頭にさまざまな情報を与え、考え方を広げる可能性があるため、マッサージという言葉はメディアの特徴を表しているともいえるだろう。

さらにマクルーハンは“人間の身体を拡張するテクノロジー”としてメディアをとらえ、人間の感覚や身体的な能力を変化させ、社会との関わりについて述べている。現在、メディアは社会、生活のあらゆる場面に存在し、五感を通してさまざまな刺激を与え、多くの技術が社会生活を豊かにしている。つまり、この身体拡張に加え、人工知能技術の発展によって“知能拡張”がメディアテクノロジーの重要な役割を持つと考えられる。このために物理的な身体と情報や知識を扱う知能を融合した“人間の身体と知能を拡張するメディアテクノロジー”を提案・開発し、これらの技術を活用して社会の構造や仕組みを変革し、

## ii 刊行のことば

---

どのような人にとっても住みやすく、生活しやすい社会を目指すことが望まれている。

一方、大学におけるメディア学の教育は、東京工科大学が1999年にメディア学部を設置して以来、全国の大学でメディア関連の学部や学科が設置され文理芸分野を融合した多様な教育内容が提供されている。その体系化が期待されメディア学に関する教科書としてコロナ社から「メディア学大系」が発刊された。この第一巻の『改訂メディア学入門』には、メディアの基本モデルの構成として「情報の送り手、伝達対象となる情報の内容（コンテンツ）、伝達媒体となる情報の形式（コンテナ）、伝達形式としての情報の提示手段（コンペア）、情報の受け手」と書かれている。これからわかるようにメディアの基本モデルには文理芸に関連する多様な内容が含まれている。

メディア教育が本格的に開始され20年を過ぎるいま、多くの分野でメディア学のより高度で急速な展開が見られる。文理芸の融合による総合知によって人間生活や社会を理解し、より良い社会を築くことが必要である。

そこで、このメディア分野の研究に関わる大学生、大学院生、さらには社会人の学修のため「メディアテクノロジーシリーズ」を計画した。本シリーズは“人間の身体と知能を拡張するメディアテクノロジー”を基礎として、コンテンツ、コンテナ、コンペアに関する技術を扱う。そして各分野における基本的なメディア技術、最近の研究内容の位置づけや今後の展開、この分野の研究をするために必要な手法や技術を概観できるようにまとめた。本シリーズがメディア学で扱う対象や領域を発展させ、将来の社会や生活において必要なメディアテクノロジーの活用方法を見出す手助けとなることを期待する。

本シリーズの多様で広範囲なメディア学分野をカバーするために、電子情報通信学会、情報処理学会、人工知能学会、日本ソフトウェア科学会、日本バーチャルリアリティ学会、ヒューマンインタフェース学会、日本データベース学会、映像情報メディア学会、可視化情報学会、画像電子学会、日本音響学会、芸術科学会、日本図学会、日本デジタルゲーム学会、ADADA Japan などにおいて第一線で活躍している研究者の方々に編集委員をお願いし、各巻の執筆者選

定、目次構成、執筆内容など検討を重ねてきた。

本シリーズの読者が、新たなメディア分野を開拓する技術者、クリエイター、研究者となり、新たなメディア社会の構築のために活躍されることを期待するとともにメディアテクノロジーの発展によって世界の人達との交流が進み相互理解が促進され、平和な世界へ貢献できることを願っている。

2023年5月

編集委員長 近藤邦雄

編集幹事 伊藤貴之

## 表紙・カバーデザインについて

私たちは五感というメディアを介して世界を知覚し、自己の存在を認知することができます。メディア技術の進歩によって五感が拡張され続ける中、「人」はなにをもって「人」と呼べるのか、そんな根源的な問いに対する議論が絶えません。

本書の表紙・カバーデザインでは、二値化された五感が新しい機能や価値を再構築する様子をシンプルなストライプ柄によって表現しました。それぞれのストライプは5本のゆらぎを持った線によって描かれており、手描きのような印象を残しました。

しかし、この細かなゆらぎもプログラム制御によって生成されており、十分に細かく量子化された表現によって「デジタル」と「アナログ」それぞれの存在がゆらぐ様子を表しています。乱雑に描かれたストライプをよく観察してみてください。本書を手にとった皆さんであれば、きっともう一つ面白いことに気づくでしょう。

デザインを検討するにあたって、同じコンセプトに基づき、いくつかのグラフィックパターンを生成可能なウェブアプリケーションを準備しました。下記 URL にて公開していますので、あなただけのカバーを作ってみてください。読者の数だけカバーデザインが存在するのです。世界はあなたの五感を通じて存在しているのですから。

馬場哲晃

〈Cover Generator〉 ぜひお試しください

<https://tetsuakibaba.github.io/mtcg/>

(2023年5月現在)



# まえがき

Unity は、ゲームなどのインタラクティブなコンテンツ制作に向けて利用普及が急速に進んでいる、3次元コンピュータグラフィックス（3DCG）による統合開発環境の一つです。スクリプトを記述するには C# のプログラミングスキルを前提としますが、高品質でリアルタイムな描画の仕方が学びやすい環境として知られています。また、マルチプラットフォームに対応し、豊富なアセットが用意され、各種の VR/AR デバイス向けの開発サポートも充実しています。

本書は、慶應義塾大学理工学部情報工学科の藤代研究室で、2023 年度から卒業研究に配属された学部 4 年生に対して、春学期（前期）に集中的に実施してきた Unity プログラミングゼミの内容に基づいています。講師は研究室に所属する大学院生たちで、各自が自身の研究を進める間に培ってきた 3DCG プログラミングのコツをより広い読者層に伝授するために、そのゼミ資料を発展させ、本書を執筆しました。

実際、なにか CG 研究のアイデアを思いついたとき、それが本当に実現可能かどうかを確かめるには、独自のプログラムを即座に組めなければなりません。そのためには CG の基本的な仕組みを理解し、実装してみる最低限のコーディング力が必要となります。すでに、3DCG の理論書やプログラミング言語・アプリの指南書は多数出版されていますが、両者をブリッジし、今風のスタイルでコードを開発して自身の方針の良否を検証できるようになるためのコンパクトなコンパニオンはなかなか見つかりません。それが本書を出版した狙いです。本書は Unity をターゲットとしていますが、環境が替わっても 3DCG の基本原理は共通するものと信じて執筆されています。

メディアテクノロジーシリーズとメディア学大系には、すでに 7 冊の CG

に関連する書籍が出版され、今後も拡充される計画です。本書は、つぎの図に示すように、両シリーズの既刊書を補間する役割を担っているといえます。

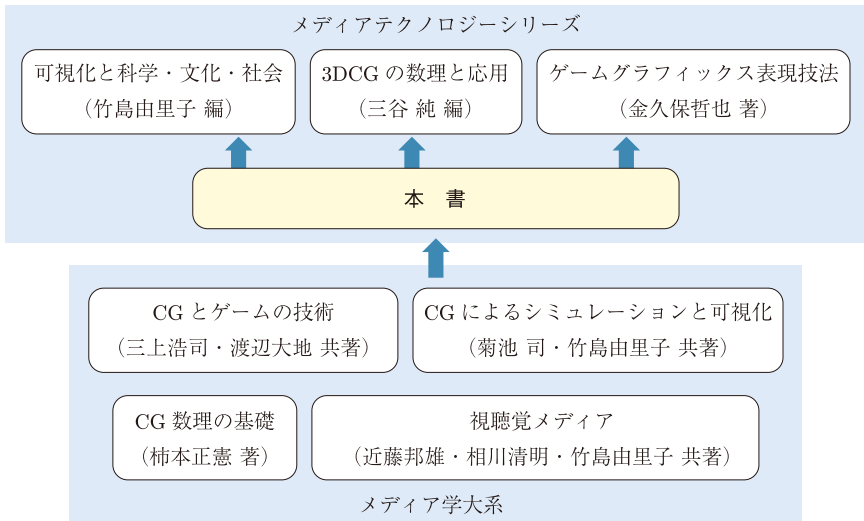


図 本書の位置付け

本書は以下のように全 6 章から構成され、3DCG の基本であるモデリングからレンダリングまでをひとつおろ網羅しています。

- 1 章 レンダリングパイプライン (青山)
- 2 章 ポリゴンメッシュ (石飛)
- 3 章 カメラ (徳永)
- 4 章 マテリアル (杉田)
- 5 章 ライティング (杉田)
- 6 章 GPGPU (青山)

1 章はモデリングからレンダリングまでの全体の処理の流れ—パイプラインについて説明しています。他章に比べやや抽象度が高いのですが、残りの章の位置付けや相互の関係を俯瞰するためにも飛ばさずに読んでほしいと思います。次章以降には、Unity を利用したプログラム例が豊富に掲載されています。2 章から 5 章の内容を順にこなしていけば、3DCG の実際を体験できる

はずです。さらに6章では、GPUの高速化機能をより幅広い関連処理に活用するGUGPUの概念を紹介し、パーティクルを用いた流体のビジュアルシミュレーションを題材に適用事例が掲載されています。

なお、コロナ社の専用ページには、口絵に掲載しきれなかったカラー図版や演習問題（難易度付、正答あり）が掲載されています<sup>注1)</sup>。本書に掲載されたプログラムは、Unity Version 6.2を前提として開発され、Windows上での動作を確認しています。上記の専用ページには、実際のコードを公開するGithubへのリンクも掲載されており、読者がUnityプログラミングを自習する際の便宜を図っています。

最後に、本書を出版する契機を作ってくくださった、メディアテクノロジーシリーズ編集委員長の近藤邦雄先生ならびに同シリーズ幹事の伊藤貴之先生に感謝いたします。また、原稿のとりまとめが予定よりも大幅に遅れたにもかかわらず、辛抱強く編集作業をご担当いただいたコロナ社の関係各位にも心からの謝意を伝えたいと思います。

本書が、読者の皆さんの3DCGプログラミングの入門に少しでも役立てば、編者としては望外の喜びです。

2026年2月

編者 藤代一成

---

注1) 本書の書籍詳細ページ (<https://www.coronasha.co.jp/np/isbn/9784339013825/>) からカラー図面などの補足情報がダウンロードできます。

注2) 本書で使用している会社名、製品名は一般に各社の登録商標です。本書では®やTMは省略しています。

注3) 本書で紹介しているURLは2026年2月現在のものです。



# 目 次

## 第 1 章

### レンダリングパイプライン

<b>1.1</b>	レンダリングパイプラインと本書の構成.....	1
<b>1.2</b>	モデルデータ.....	2
1.2.1	サーフェスモデル	3
1.2.2	ソリッドモデル	4
<b>1.3</b>	トポロジー.....	5
1.3.1	プリミティブ	5
1.3.2	頂点インデックスバッファ	5
1.3.3	プリミティブとトポロジー	6
<b>1.4</b>	レンダリング.....	7
1.4.1	パイプライン	8
1.4.2	ティアリング	10
1.4.3	垂直同期	10
1.4.4	ダブルバッファリング	11
<b>1.5</b>	レンダリングキュー.....	12
1.5.1	レンダーキュー	12
1.5.2	デプスソート	13
1.5.3	デプスバッファ	13
1.5.4	不透明メッシュ	14
1.5.5	半透明メッシュ	15
1.5.6	アルファブレンド	17
<b>1.6</b>	ま と め.....	18

## 第 2 章 ポリゴンメッシュ

<b>2.1</b>	<b>3次元空間に関する数学的基礎</b> .....	20
2.1.1	直交座標系	21
2.1.2	面の方向	22
<b>2.2</b>	<b>メッシュの基本的な生成方法</b> .....	22
2.2.1	Meshクラス	23
	コラム：レンダリングパイプラインと描画命令	24
2.2.2	ジオメトリ情報の設定	25
2.2.3	トポロジー情報の設定	26
2.2.4	メッシュの描画	28
2.2.5	属性情報の設定	30
<b>2.3</b>	<b>ループ処理の利用</b> .....	31
2.3.1	ジオメトリ情報の設定	32
2.3.2	トポロジー情報の設定	33
2.3.3	属性情報の設定	35
2.3.4	頂点移動による変形	36
<b>2.4</b>	<b>折れ曲がった面の表現</b> .....	38
2.4.1	上面・底面の作成	38
2.4.2	側面の作成	41
2.4.3	法線の設定	43
<b>2.5</b>	<b>再帰処理の利用</b> .....	45
2.5.1	L-System	45
2.5.2	再帰処理	48
2.5.3	タートルの定義	51
2.5.4	線による枝の描画	52
2.5.5	メッシュによる枝の描画	54
<b>2.6</b>	<b>ま と め</b> .....	57

# 第 3 章 カ メ ラ

<b>3.1 実世界のカメラ</b> .....	59
3.1.1 ピンホールカメラ	60
3.1.2 レンズカメラ	60
3.1.3 投影手法	62
<b>3.2 座標変換の基礎</b> .....	63
3.2.1 同次座標系	63
3.2.2 アフィン変換	65
<b>3.3 座標系の必要性</b> .....	66
3.3.1 カメラの役割	68
3.3.2 座標変換の流れ	69
<b>3.4 各座標系の意味</b> .....	70
3.4.1 ローカル座標系	70
3.4.2 ワールド座標系	70
3.4.3 ビュー座標系	71
3.4.4 クリップ座標系・NDC系	71
<b>3.5 座標変換と行列</b> .....	72
3.5.1 ローカル座標系からワールド座標系への変換	73
コラム：モデル行列の見方	73
3.5.2 ワールド座標系からビュー座標系への変換	74
3.5.3 ビュー座標系からクリップ座標系・NDC系への変換	75
コラム：NDC系？ NDC座標系？	77
3.5.4 投影行列の導出	77
<b>3.6 カメラ行列の編集と応用：立体映像</b> .....	79
3.6.1 RenderTexture	80
3.6.2 アナモルフォーシス	80
3.6.3 投影行列の操作	81

3.6.4	立体的に見える画像の生成	83
3.6.5	立体的に見える映像の生成	88
<b>3.7</b>	<b>ま と め</b> .....	<b>90</b>

## 第 4 章 マテリアル

<b>4.1</b>	<b>マテリアルの基本的な概念</b> .....	<b>91</b>
4.1.1	属 性	91
4.1.2	シェーディングモデル	92
<b>4.2</b>	<b>Unity のマテリアル</b> .....	<b>92</b>
4.2.1	シェーダ	93
4.2.2	プロパティ	94
4.2.3	レンダリングパス	94
	コラム：CGPROGRAM と HLSLPROGRAM の違い	94
<b>4.3</b>	<b>代表的なシェーディングモデルと実装</b> .....	<b>95</b>
4.3.1	基本コードの作成	95
	コラム：Unity の標準シェーダライブラリ	96
4.3.2	完全鏡面	97
4.3.3	拡散反射	97
	コラム：接空間の連続性	100
4.3.4	フォン反射	101
4.3.5	GGX (Ground Glass Unknown)	105
<b>4.4</b>	<b>マテリアルの効率的な描画手法</b> .....	<b>113</b>
4.4.1	プレフィルタリング	114
4.4.2	拡散反射におけるプレフィルタリング	115
4.4.3	GGX におけるプレフィルタリング	123
<b>4.5</b>	<b>ま と め</b> .....	<b>130</b>

# 第 5 章

## ライティング

<b>5.1 光源の種類と特性</b> .....	132
5.1.1 光源の分類	132
5.1.2 光源の照明特性	133
<b>5.2 照明計算の理論的取扱い</b> .....	135
5.2.1 点光源	135
5.2.2 平行光源	135
5.2.3 スポット光源	136
<b>5.3 基本的なライティング手法の実装</b> .....	136
5.3.1 ライティング計算の基本コード	136
5.3.2 平行光源のライティング	140
5.3.3 点光源のライティング	140
5.3.4 スポット光源のライティング	141
<b>5.4 シャドウイング</b> .....	144
5.4.1 シャドウイングとは	144
5.4.2 カメラと光源の双対性	145
5.4.3 シャドウマッピング	145
5.4.4 シャドウマッピングの実装	146
<b>5.5 レイトレーシングによる面光源ライティング</b> .....	150
5.5.1 面光源における照明計算	151
5.5.2 レイトレーシングの基礎知識	151
5.5.3 Unity におけるレイトレーシング	152
5.5.4 レイトレーシングによる面光源ライティングの実装	167
コラム：GPU レイトレーシングの歴史	169
<b>5.6 ま と め</b> .....	175

# 第 6 章

## GPGPU

<b>6.1</b>	<b>本章の前提</b> .....	177
<b>6.2</b>	<b>スレッドとワーブ</b> .....	178
<b>6.3</b>	<b>Compute Shader の使い方</b> .....	179
6.3.1	Compute Shader の書き方	180
6.3.2	Compute Shader の実行	181
6.3.3	サンプルコードの実行	184
<b>6.4</b>	<b>GPU アーキテクチャ</b> .....	185
6.4.1	GPC と TPC	185
6.4.2	SM	186
6.4.3	メモリ構造	187
<b>6.5</b>	<b>排他制御</b> .....	189
6.5.1	基本概念	189
6.5.2	アトミック操作	190
6.5.3	GPU のアトミック操作	191
6.5.4	Shared Memory	192
<b>6.6</b>	<b>コマンドバッファ</b> .....	195
<b>6.7</b>	<b>2D 流体シミュレーションへの応用</b> .....	197
6.7.1	流体シミュレーション	197
6.7.2	Compute Shader	199
6.7.3	C# による制御	204
6.7.4	計算結果の描画	205
6.7.5	結 果	206
<b>6.8</b>	<b>ま と め</b> .....	207
	<b>引用・参考文献</b> .....	209
	<b>索 引</b> .....	214

# 第 1 章

## レンダリングパイプライン

本章では、コンピュータグラフィックスの基本となるレンダリングパイプラインについて学ぶ。コンピュータグラフィックス（computer graphics, CG）とは、コンピュータを用いて画像を生成する技術をさし、大きく 2D と 3D の 2 種類に分類される。2DCG は平面上に存在する物体を描画する技術であり、3DCG は空間内に存在する物体を描画する技術である。本書ではおもに、3DCG に焦点を当てる。

本章では、レンダリングパイプラインを構成する主要な要素を学び、全体の仕組みを理解することを目的とする。まず、1.2 節では、3 次元物体を表現するための基本データ構造であるモデルデータについて説明する。つぎに、1.3 節では、モデルデータの頂点や稜線、面の接続関係を管理するためのトポロジーについて学ぶ。続いて、1.4 節では、GPU を用いてモデルデータを 2 次元画像に変換するレンダリング処理の流れを理解する。最後に、1.5 節では、描画順序を管理するためのレンダリングキューについて学び、レンダリングパイプライン全体を把握する。

### 1.1 レンダリングパイプラインと本書の構成

レンダリングとは、3 次元空間に存在する物体を 2 次元画像へと変換する処理である。図 1.1 に示すように、この処理は大きく 4 つのフェーズに分けられる。すなわち、3 次元物体であるメッシュの配置、カメラ座標系への変換、物体表面の性質を定義するマテリアルの生成、そしてマテリアルに基づいた描画による画像生成である。生成された画像はディスプレイへ転送され、ユーザはそれを視覚的に確認することができる。

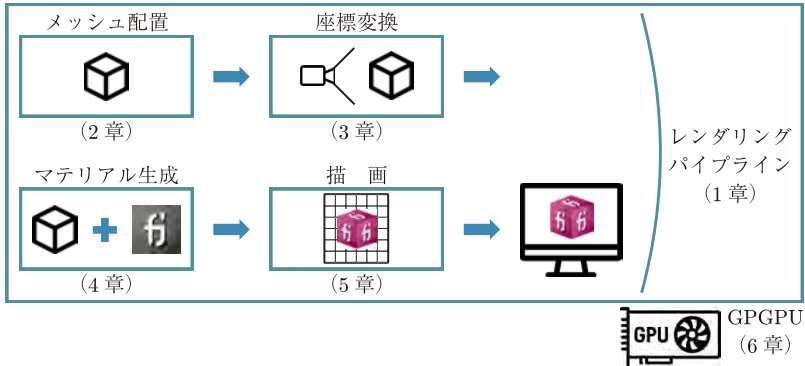


図 1.1 レンダリングパイプラインと各章の対応関係

本章では、これら4つのフェーズの全体像を整理し、レンダリングパイプラインとして体系的に示す。各フェーズの詳細な説明は2章から5章にかけて順に扱う。さらに、これらの処理を効率的に実行するための専用ハードウェアであるGPUと、GPUを用いた汎用計算手法については6章に譲る。

本書では、レンダリングの各フェーズについて理論と実装の双方の観点から解説する。より高度な応用に関心がある読者は、『Unity Graphics Programming Series』<sup>1)†</sup>や『Computer Graphics Gems JP』<sup>2)~4)</sup>を参照するとよい。また、基礎理論や仕組みをより深く理解したい読者には、CG-ARTS協会の『コンピュータグラフィックス』<sup>5)</sup>や『ゲーム制作者になるための3Dグラフィックス技術』<sup>6)</sup>などを読むとより理解が深まる。

## 1.2 モデルデータ

モデルデータ (model data) とは、レンダリング処理によって2次元画像に変換される3次元物体の立体形状データのことである。モデルデータには、大きくサーフェスモデルとソリッドモデルという2つの形式が存在する。

† 肩付き数字は、巻末の引用・参考文献を表している。

### 1.2.1 サーフェスモデル

サーフェスモデル (surface model) とは、図 1.2 に示すように、立体形状を表面のみで表現したモデルである。このモデルでは、表面には厚みがなく、質量や体積をもたず、表面積のみをもつ。サーフェスモデルの表現方法としては、ポリゴンモデル、曲面モデリング、細分割曲面などが挙げられる。

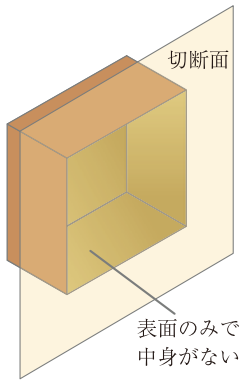


図 1.2 サーフェスモデル

〔1〕 ポリゴンモデル ポリゴンモデルは、頂点とそれらをつなぐ線分によって定義される多角形 (ポリゴン) で立体を表現する手法である。図 1.3 に示すように、ポリゴンの形状としては三角形が基本であるが、3 辺以上であれば何角形でも構わない。これらのポリゴンが集まって形成される立体をポリゴンメッシュ (polygon mesh) または単にメッシュ (mesh) とよぶ。これは最も一般的なモデルデータの一つであり、次節以降でもモデルデータにはポリゴンモデルを用いる。

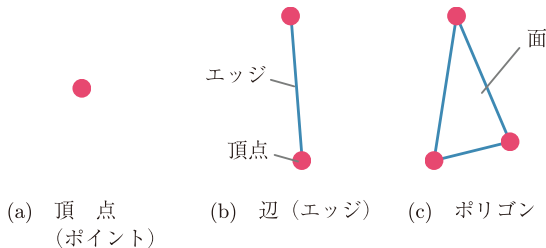


図 1.3 三角形によるポリゴンの構成

## 4 1. レンダリングパイプライン

〔2〕 **曲面モデリング** 曲面モデリング<sup>7)</sup>は、曲線を用いて立体形状を表現する方法である。Bézier 曲線や B-スプライン曲線など、さまざまな自由曲線 (free-form curve) を組み合わせて形状を定義し、曲線の制御点を操作することで立体形状を調整する。

〔3〕 **細分割曲面** 細分割曲面 (subdivision surface)<sup>8)</sup>は、ポリゴンメッシュを規則的に細分化することで曲面を表現する手法である。この手法はポリゴンメッシュと曲面モデリングを組み合わせたもので、再分割の度合いに応じて立体形状の滑らかさを調整できる。

### 1.2.2 ソリッドモデル

ソリッドモデル (solid model) とは、立体形状の内部まで表現したモデルである。図 1.4 に示すように、ソリッドモデルは体積をもち、その材料や密度から物性や質量を算出することができる。ソリッドモデルの表現方法の代表は、CSG と境界表現である。

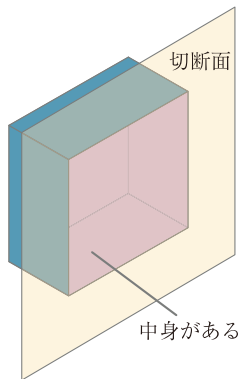


図 1.4 ソリッドモデル

〔1〕 **CSG** CSG (constructive solid geometry) は、立方体や円柱などの基本的な立体を組み合わせて形状を表現する手法である。複数の基本立体形状に対して論理演算を行うことで、より複雑な形状を作り出せる。

〔2〕 **境界表現** 境界表現 (boundary representation, **B-Reps**) は、複数の平面や曲面を組み合わせ、それぞれの面どうしを共有辺で接続するこ

とで形状を表現する手法である。単に面を組み合わせた状態ではサーフェスモデルであるが、面どうしの接続関係を定義して閉じた形状に変換することで、ソリッドモデルとして扱うことができる。

## 1.3 トポロジー

トポロジー (topology) とは、モデルデータ内の頂点や辺、面の接続関係を示す情報のことである。さらに、頂点インデックスを使用してメッシュの頂点情報を管理することで、重複を避けながら情報を再利用できるようになり、複雑なメッシュを効率的に描画することが可能となる。

### 1.3.1 プリミティブ

プリミティブ (primitive) とは、グラフィックスエンジンが描画可能な最小単位をさす。一般的には、図 1.5 に示すように、頂点、辺、ポリゴン (三角形) の3種類が使用され、すべてのモデルデータはこれらの幾何要素を組み合わせることで表現される。

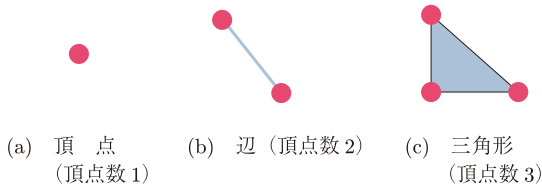


図 1.5 プリミティブ

### 1.3.2 頂点インデックスバッファ

プリミティブを用いてメッシュを表現する際、同じ座標に複数の頂点が存在することがある。そこで、図 1.6 に示すように、頂点情報を頂点インデックスによって辞書化し、再利用することで、データ量を削減する手法が一般的に用いられている。この際、頂点情報とインデックスのペアから作られる辞書

# 索引

<b>【あ】</b>	
アーティファクト	149
アクセスレイテンシ	188
アスペクト比	63
アトミック性	190
アトミック操作	190
アナモルフォーシス	80
アフィン変換	66
粗 さ	107
アルファブレンド	17
<b>【い】</b>	
位 置	28
隠面消去	13
<b>【お】</b>	
オイラー法	198
<b>【か】</b>	
カーネル	180
回 転	28
画 角	62
拡散反射	97
加速構造	152
カリング	9
完全鏡面	97
<b>【き】</b>	
境界表現	4
曲面モデリング	4
<b>【く】</b>	
グラム・シュミットの 正規直交化	99

グリッド	178
クリップ座標系	71
<b>【け】</b>	
原子性	190
<b>【こ】</b>	
格子法	198
光沢度	101
コマンドバッファ	195
コンピュータグラフィックス	1
<b>【さ】</b>	
サーフェスモデル	3
再 帰	48
サイクロプスの眼	89
細分割曲面	4
<b>【し】</b>	
シェーダ	8, 93
シェーディングモデル	92
ジオメトリ情報	23
軸境界ボックス	152
視錐台	75
シャドウアクネ	149
シャドウイング	144
シャドウマッピング	145
シャドウマップ	146
自由曲線	4
重点サンプリング	105
準モンテカルロ法	122
初期値	46

<b>【す】</b>	
垂直同期	10
スポット光源	134
スレッド	178
スレッドグループ	178
スレッドブロック	178
スワップチェーン	12
<b>【せ】</b>	
世 代	46
接空間座標系	99
占有率	179
<b>【そ】</b>	
双 対	145
双方向バストレーシング	152
双方向反射率分布関数	106
属性情報	23
ソフトシャドウ	175
ソリッドモデル	4
<b>【た】</b>	
多重重点的サンプリング	105
タートルグラフィックス	46
ダブルバッファリング	11
<b>【ち】</b>	
置換規則	46
頂 点	5
頂点インデックス	5
頂点インデックスバッファ	6
頂点パイプライン	9

<b>【て】</b>	
ティアリング	10
低い違い量列	121
定数	46
デプスソート	13
デプスバッファ	13
点光源	133

<b>【と】</b>	
投影行列	75
同次座標系	63
透視投影	62
トポロジー	5, 6
トポロジー情報	23

<b>【は】</b>	
バウンディングボックス	37
ハーフベクトル	107
排他制御	189
バストレッシング	152
ハードウェアレイ トレッシング	152
バックバッファ	11
バンク	189
バンクコンフリクト	189
半透明メッシュ	15

<b>【ひ】</b>	
ピクセルパイプライン	10
被写界深度	61
非同期コンピュータ	195
ビューイングパイプライン	8
ビュー行列	74
ビュー座標系	69

<b>【ふ】</b>	
フェンス	196
フォトンマッピング	152
フォン反射	101
物理ベースレンダリング	106
不透明メッシュ	14, 15
プリミティブ	5
フレームバッファ	9
フレネル反射率	107
プレフィルタリング	113
プロシージャルモデリング	
	45
プロパティ	94
フロントバッファ	11

<b>【へ】</b>	
平行光源	134
平行投影	62
ヘッドマウントディスプレイ	79
ペナルティ法	203
変数	46

<b>【ほ】</b>	
放射輝度	98
放射照度	98
ポリゴン	3
ポリゴンメッシュ	3
ポリゴンモデル	3

<b>【ま】</b>	
マテリアル	28, 91

<b>【め】</b>	
メッシュ	3, 28
面光源	134

<b>【も】</b>	
モデル行列	73
モデルデータ	2
モンテカルロ法	98

<b>【ら】</b>	
ラグランジュ法	198
ラスタライザ	9
ランバートモデル	97

<b>【り】</b>	
リサンブル重点サンプリング	105
粒子法	198

<b>【れ】</b>	
レイトレーシング	151
レイヤ	28
レンダーキュー	12
レンダーターゲット	7
レンダリング	7
レンダリングキュー	12
レンダリングパイプライン	8
レンダリングパス	94
レンダリング方程式	106

<b>【わ】</b>	
ワープ	178
ワープスケジューラ	179
ワールド座標系	69

<b>【A】</b>	
AABB	152

<b>【B】</b>	
BRDF	106
BVH	152
B-Reps	4

<b>【C】</b>	
Compute Queue	195
Constant Cache	188
Cookie テクスチャ	142

CSG	4	L1 Cache	188		
CSM	150	L2 Cache	188		<b>[S]</b>
CUDA Core	186			SBT	158
	<b>[E]</b>		<b>[M]</b>	Shared Memory	188
Early-Z	14	masking-shadowing 関数	107	SIMD	177
	<b>[F]</b>	microfacet	106	SIMT	178
far plane	75		<b>[N]</b>	SM	186
	<b>[G]</b>	NDC 系	69	SPH 法	198
GGX	107	near plane	75	spp	104
Global Memory	188		<b>[P]</b>	SRV	192
GPC	185				<b>[T]</b>
GPGPU	177	PBR	106	Thread Group Shared	
GPU アーキテクチャ	185	PCF	150	Memory	192
GPU 駆動レンダリング	208	Perlin ノイズ	36	Torrance-Sparrow	107
Graphics Queue	195		<b>[Q]</b>	TPC	186
	<b>[I]</b>	QMC	122	triangle strip	6
IBL	96		<b>[R]</b>		<b>[U]</b>
	<b>[L]</b>	Register file	189	UAV	192
L-System	45	RenderTexture	80, 182	UV 座標	9
L0 Cache	188	RHI	154		<b>[V]</b>
				VPL	134
				VSM	150

—— 編者・著者略歴 ——

藤代 一成 (ふじしろ いっせい)

1983年 筑波大学第三学群情報学類卒業  
1985年 筑波大学大学院博士課程工学研究科修士号取得退学 (電子・情報工学専攻)  
1985年 東京大学助手  
1988年 理学博士 (東京大学)  
1989年 筑波大学講師  
1991年 お茶の水女子大学助教授  
1998年 お茶の水女子大学教授  
2004年 東北大学教授  
2009年 慶應義塾大学教授  
現在に至る

石飛 晶啓 (いしとび あきのり)

2019年 慶應義塾大学理工学部情報工学科卒業  
2020年 慶應義塾大学大学院理工学研究科前期博士課程修了 (開放環境科学専攻情報工学専修)  
2023年 慶應義塾大学大学院理工学研究科後期博士課程修了 (開放環境科学専攻情報工学専修), 博士 (工学)  
2024年 デジタルコンテンツ制作会社勤務  
現在に至る

徳永 恵太 (とくなが けいた)

2020年 慶應義塾大学理工学部情報工学科卒業  
2022年 慶應義塾大学大学院理工学研究科前期博士課程修了 (開放環境科学専攻情報工学専修)  
2023年 慶應義塾大学大学院理工学研究科後期博士課程 (開放環境科学専攻情報工学専修), 在学中

杉田 俊平 (すぎた しゅんぺい)

2021年 慶應義塾大学理工学部電子工学科卒業  
2023年 慶應義塾大学大学院理工学研究科前期博士課程修了 (開放環境科学専攻情報工学専修)  
2023年 慶應義塾大学大学院理工学研究科後期博士課程 (開放環境科学専攻情報工学専修), 在学中

青山 昂平 (あおやま こうへい)

2013年 デジタルコンテンツ制作会社勤務, 在職中  
2023年 明治大学総合数理学部先端メディアサイエンス学科卒業  
2023年 慶應義塾大学大学院理工学研究科前期博士課程 (開放環境科学専攻情報工学専修), 在学中

# Unity による 3DCG プログラミング —基本原理と実践—

3D Computer Graphics Programming with Unity—Fundamentals and Practice—

© Fujishiro, Ishitobi, Sugita, Tokunaga, Aoyama 2026

2026 年 3 月 31 日 初版第 1 刷発行

★

検印省略

編 者 藤 代 一 成  
著 者 石 飛 晶 啓  
杉 田 俊 平  
徳 永 恵 太  
青 山 昂 平  
発 行 者 株式会社 コロナ社  
代表者 牛来真也  
印 刷 所 三美印刷株式会社  
製 本 所 株式会社 グリーン

112-0011 東京都文京区千石 4-46-10

発 行 所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844 ・ 電話 (03) 3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-01382-5 C3355 Printed in Japan

(松岡) D



**JCCOPY** <出版者著作権管理機構 委託出版物>

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-5244-5088、FAX 03-5244-5089、e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。