

3ZF-05

## プログラミング基礎教育のための図形言語の 3D 拡張

古川 孝太郎<sup>1</sup>, 糸山 克寿<sup>1</sup>, 吉井 和佳<sup>1</sup>, 奥乃 博<sup>1</sup><sup>1</sup> 京都大学 大学院情報学研究科 知能情報学専攻

## 1. はじめに

本研究は、京都大学工学部情報学科の開講科目「アルゴリズムとデータ構造入門」[1]のカリキュラムを改善し、3D 図形、特にフラクタル図形の構成と造型を通じた、手続き・データ抽象化への理解の促進を目的とする。プログラミング言語 Scheme 上で 2D 図形を構成的、再帰的に描画可能なシステムである図形言語 [2] を取り扱い、これに 3D 図形の構成と描画、3D プリンタによる造型のための変換機能を加える。実現したシステムを講義で取り入れ、受講生から作品の提出を受けて、それを造型したモデルとともにフィードバックを行い、アンケートによる評価を加える。

「アルゴリズムとデータ構造入門」は主にプログラミングの初学者を対象とし、手続き・データ抽象化等といった基礎を扱う講義である教科書として SICP (Structure and Interpretation of Computer Programming) [2] を使い、Scheme によるプログラミングの演習を通して学習を進める。図形言語はこのカリキュラムに含まれるもので、プログラムの記述と図形の部分との対応が容易であるため、手続きの呼出しを視覚的に捉え、再帰呼出しや階層化等の概念を直観的に学ぶことができる。

図形を構成的に作成する試みは Scheme をはじめとする Lisp 系言語で早くからなされてきた。Actor/Scriptor Animation System [3] は 3D アニメーションを構成するシステムである。図形の移動や回転、画面のズーム等の映像上の効果を表す、独立性の高い手続きの組合せにより、複雑な映像を表すプログラムの構成を容易にしている。本システムもこのプリミティブな手続きの組合せにより複雑な図形を構成する設計思想を採用する。

本研究は、図形言語の 3D 拡張とともに、その上で作成した図形を 3D プリンタによって造型するためのインタフェースも提供する。3D プリンタは可塑性樹脂を熱で溶かし平面形状を積層しながら立体形状を造型するデバイスであり、ラピッドプロトタイピングに有用である。従来の講義では平面図形の描画しかできなかったが、作成した 3D 図形を実際に手に取ることのできるモデルとして実体化し学生へフィードバックを与えることで、そのプログラムの構造への理解を深めるとともに講義内容の積極的な学習を促進する。

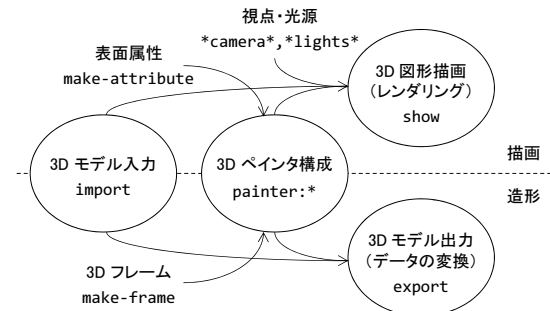


図 1: 3D 図形言語の概要

## 2. 設計

3D 図形言語は主として手続き・データ抽象化をはじめとした基礎的な概念の学習というプログラミング教育に用いられる。その機能として SICP による図形言語の拡張としての 3D 図形の構築と描画の仕組みを持ち、構成された 3D 図形を 3D プリンタにより造型可能なデータ形式への変換を可能とすることが求められる。非機能的な面は特に使用性や可搬性について、SICP によるカリキュラムに沿った理解と学習が可能であるような図形言語の思想に沿って設計されること、Scheme というプログラミング学習のために適した言語の特長を生かしていること、ユーザであるプログラミング学習者の興味をひき積極的に利用しようという意識を持たせること、異なる計算機環境においても利用が容易であること、実用に不自由ない程度のユーティリティやサンプルが提供されることが必要である。これらの要求に従い、次の機能を持つよう本システムを設計する。

- 本システムは 3D ペインタを構成する手続きを提供する。2D 図形言語においてペインタの評価が図形の描画を意味するのと同様に、3D 図形言語における 3D ペインタを評価することで 3D 図形を描画できる。3D ペインタはその評価時に、あらかじめ定義された視点と光源の情報に基づき、視点から見た場合の 3D 図形の見え方を計算して像を求め、ウィンドウ上にそれを描画する。
- 本システムは 3D フレームを構成する手続きを提供する。2D 図形言語においてフレームがアフィン変換を表す三次正方行列に等価であるのと同様に、3D 図形言語における 3D フレームはアフィン変換を表す四次正方行列に相当する情報を持つ。3D フレームの 3D ペインタへの適用は、3D ペインタの保持する 3D 図形への 3D フレームの表す変換の適用である。

謝辞: 本研究の一部は、科研費 24220006 の支援を受けた。

- 本システムは 3D ペインタに 3D フレームを適用し、新たな 3D ペインタを構成する手続きを提供する。
- 本システムは複数の 3D ペインタを合成し、新たな 3D ペインタを構成する手続きを提供する。
- 本システムは 3D 図形の出力手続きを提供する。出力手続きは 3D ペインタのデータ形式を変換してファイルに書き出す機能を持つ。出力先のデータ形式は、3D プリント可能な形式へ変換する目的で OpenSCAD スクリプト形式を選択できるものとしテキストファイルとしてこれを生成する。
- 本システムは図形定義のための空間座標系を持ち、その中に存在する図形をウィンドウという平面座標系に投影するために必要な視点と光源を定義する手続きを提供する。
- 本システムは Scheme の言語拡張として構成され、Scheme の機能を包含する。すなわち本システムは Scheme の文法に従い、新たに定義される手続きのほか、Scheme のもともと提供する手続きは全て利用可能であるものとする。

### 3. 実装

我々は 3D 図形言語における 3D 図形の構成、描画、変換という三つの機能をそれぞれ 3D ペインタ及び 3D フレームの構成、レンダリング、データ形式の変換を実現する Scheme 上の手続きとして実装する。3D ペインタは閉包内に 3D 図形の内部表現を持ち、評価時にはそれに 3D フレームの表す変換を適用してレンダリング手続きへ渡す手続きとして構成される。この時 3D 図形を変換の適用された履歴と適用後のポリゴンの集合という二種類のデータとして保持することで、描画と出力のそれぞれに対応する。レンダリング手続きは投影、クリッピング、シェーディング、陰面・陰線消去、及びこれらを経て得られた処理済みのポリゴンの描画という五つの段階によって、3D 座標上におけるポリゴンの集合を 2D 図形言語における描画機能へと還元する。データ形式の変換手続きは 3D 図形の内部表現のうち変換履歴を表す木構造を再帰的に辿りながら、木構造の各節の意味する変換や合成といった操作、および葉の意味するプリミティブな図形を、それぞれに対応する OpenSCAD スクリプト上の関数に変換する。

### 4. 評価

評価のため 3D 図形言語を用いて独創的な 3D 図形を創作する随意課題を設け、作品とアンケートの提出を受け付けた。提出作品を 3D プリンタを用いて造型し、寸評とともに学生に与えてフィードバックとした。

提出を受けた作品のうち、図 2a に示す独自のフラクタル図形と Hirbert 曲線 [4] を紹介する。前者は最小単位を立方体とし、親子関係として親の立方体の六つ

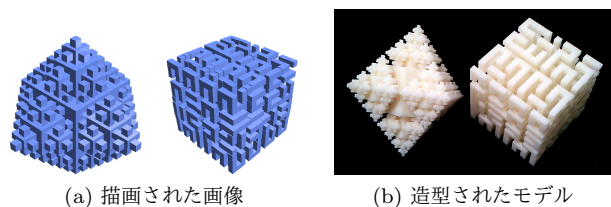


図 2: 提出を受けた 3D 図形作品

の面の中心にそれぞれ子の立方体を接して配置する構造を取る。すなわち再帰度が 0 の場合は立方体であり、1 の場合は大小の立方体がそれぞれ 1 個と 6 個、2 の場合は大中小の立方体がそれぞれ 1 個、6 個、30 個というように、立方体が増加していく。後者は空間充填曲線というフラクタル図形の一つであり、立体的なコの字型の要素が再帰的に呼び出され、それぞれの間が接続された構造を持つ。それぞれを造型した結果を図 2b に示す。

またアンケートにより 3D 図形言語とその随意課題に対する感想を調査したところ「おもしろく、ぜひ挑戦したい」といった多数の好意的な感想を受け取った。以上のことから本システムの利用によってより学生の興味をひき、講義及びプログラミング学習に対する積極性を増すことと、プログラミングのための基礎となる技術力の向上が期待できる。

### 5. おわりに

本研究はプログラミング基礎教育の教材として、図形言語を拡張し 3D 図形の構成・描画と 3D プリンタによる造型を可能とするシステム 3D 図形言語を開発した。オリジナルの図形言語からの連続的な学習を重視して設計し、構成的な 3D 図形作成の仕組みを取り入れ、3D 図形の実体化によるプログラムの構造に対する視覚的かつ直観的な理解を可能とした。また本システムを実際に講義で使用し、課題の提出やアンケートによって学生からの良好な反応を受け取って有効性を確認した。3D 図形言語のソースコードは Github 上<sup>1</sup>、講義に用いたスライドは SlideShare 上<sup>2</sup>で公開しており、誰でも閲覧し利用することが可能である。

### 参考文献

- [1] 奥乃 博, 馬谷 誠二, 糸山 克寿, Syllabus - 京大工学部, <http://www.t.kyoto-u.ac.jp/syllabus-s/?mode=subject&lang=ja&year=2013&b=6&c=91150> (April 2013).
- [2] H. Ableson, G. J. Sussman, J. Sussman, Structure and Interpretation of Computer Programs, 2nd Edition, The MIT press, Cambridge, MA, USA, 1996.
- [3] C. W. Reynolds, Computer Animation with Scripts and Actors, SIGGRAPH Comput. Graph. 16 (3) (1982) 289–296.
- [4] D. Hilbert, Über die stetige Abbildung einer Linie auf ein Flächenstück., Math. Ann. 38 (3) (1891) 459–460.

<sup>1</sup><https://github.com/vi-iv/jakld-3dgc>

<sup>2</sup><http://www.slideshare.net/vi-iv/jakld-3dgc>