

## 自然言語による3DCGモデラ “Do”

河合 善之<sup>†</sup>, 岡田 稔<sup>††</sup>

名古屋大学大学院工学研究科<sup>†</sup> 電子情報学専攻<sup>††</sup> 情報工学専攻

あらまし: 本論文では自然言語に基づいた3DCG幾何形状モデラ “Do” について述べる. 現在数多く利用されているCG作成システムでは, 肢体不自由等でキーボード, マウス等のハードウェアデバイスを使うことが困難なユーザ等がCGを作成することは容易ではない. そこで, 本モデラではモデリングのためのマン・マシンインタフェースを音声(自然言語)にすることにより, このようなユーザにもCGが作成できること等を目的としている. 本システムを利用することにより, CGの専門的知識の無いユーザや, 遠隔会議などコンピュータから遠隔地にいるユーザにもCGを作成することが可能となると考えられる.

## Natural-Language-Based 3DCG Modeler “Do”

Yoshiyuki KAWAI<sup>†</sup> and Minoru OKADA<sup>††</sup>

<sup>†</sup>Department of Information Electronics, <sup>††</sup>Department of Information Engineering,  
School of Engineering, Nagoya University

**Abstract:** In this paper, a new method to create a 3D shape using natural language is proposed. If it were possible to create 3DCG using natural language, inexpert users who do not have special knowledge of CG, and handicapped users who are unable to use hardware devices such as a mouse or a keyboard would be able to create a CG image. Generally, it is difficult for these users to create CG by using traditional CG modelers. So, we have developed “Do”: a 3D shape modeler utilizing natural language, that these users would be able to create CG.

### 1 はじめに

近年, 三次元コンピュータグラフィクス(3DCG)を作成する3Dモデラが数多く利用されている. しかし, これらのモデラでCGを作成する際には, CGに関する専門的知識が必要が多く, その専門的知識の無いユーザがCGを作成することは容易ではない. また既存のCG作成ソフトウェアの多くは, キーボードのようなキャラクタ入力デバイス, マウスのようなポインティングデバイスを利用することが必要となってくる. そのため肢体不自由等の理由で, これらのハードウェアデバイス等が利

用困難なユーザにとっても, CGを作成することは非常に困難であると言える. 自然言語をユーザインタフェースとして用いることができれば肢体不自由者, あるいは遠隔会議などのような, コンピュータから遠隔地にいるユーザにおいても音声はコンピュータに届けば, 容易に3DCGを作成することが可能となる.

そこで, 本論文では自然言語を志向して会話的に3Dモデルを作成することを目的とした3Dモデリングシステム “Do” について述べる.

このような自然言語を利用した過去の関連研究として, 2Dでは自然言語と手をポインタと

して利用したレイアウト技術 Put-That-There [1]をはじめとして、風景描写文から風景画像を生成する Picnyck [2] 等が報告されている。また 3D では仮想室内における家具配置作業のような室内レイアウトを考慮した 3D レイアウト技術 [3] に関する研究や、オブジェクト配置システム “Put” [4] がある。この Put は、スクリプトファイルを使用し動詞 *put* と *hang* によって、オブジェクトを配置するシステムである。

## 2 3DCG モデラ “Do”

### 2.1 システムの概要

3DCG モデラ “Do” では、図 1 のような流れで処理を行なう。本システムでは、音声スピーチは、既に音声認識を通してテキストに変換されたものと仮定している。また、自然言語として英語を目的とした。

テキストは過去の 3D シーン情報、即ち、今まで作成してきたオブジェクトの位置、サイズ、色等の情報を参照しながら構文解析部・インタプリタによって、我々の定義したニーモニックに変換される。また、モデリングの際に、新しくオブジェクトを呼び出す必要がある場合は、システム付属のオブジェクト辞書からオブジェクトデータを読み出し、それを使用する。

このニーモニックによって、3D オブジェクトを配置・変形しながら 3D モデルを作成していき、生成された 3D シーンをスクリーンに描画する。その際、新たに生成された 3D シーン情報を更新する。以上の処理を繰り返し、形状モデリングを行う。

### 2.2 ニーモニック

今回、“Do” のモデリング手法として CSG (Constructive Solid Geometry) を使用した。本システムを作成するにあたって、まず、CSG と自然言語を利用して 3D モデルを構築していく際に必要となる有益な英単語を集め、それらを分類した(表 1)。動詞については、同じような意味を有する動詞類に分類し、本システムにおける中間表現としてニーモニックを定義した(表 2)。ユーザが入力した自然言語は、2.3 で説明する構文解析部・インタプリタを通して、こ

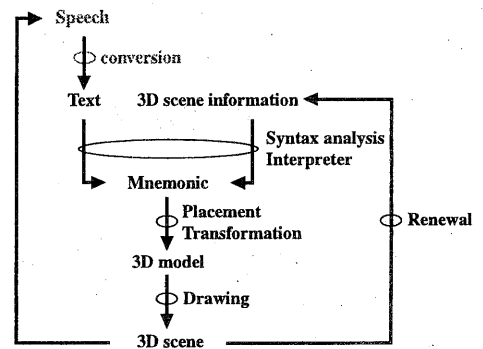


図 1: “Do” の処理の流れ  
Fig. 1: Overview of “Do”

のようなニーモニックに変換され、オブジェクトの配置、変形等の操作を行い、3D モデルを作成する。その後、スクリーン上に新しい 3D モデルが描画される。

### 2.3 構文解析部

構文解析部は lex & yacc[11][12] を用いて作成した。現在、システムでサポートしている自然言語の書式を表 3 に BNF で表記した。インタプリタでは、テキストを過去の 3D シーン情報を参照しながらニーモニックに変換する。表 3 の <sentence> はインタプリタによって、全てニーモニックに変換される。

現システムでは単文のみを解釈可能にしている。今後、より複雑な構文を解析できるように Parser をシステムに組み込む予定である。

### 2.4 度量・程度表現の解釈

インタプリタの作成の際に以下のような問題が存在する。例えば、“move the sphere 100cm left.” のように、100cm と正確な距離、位置をユーザが指定した際は、100cm 左にオブジェクト (the sphere) を移動するだけでよい。しかし、問題となるのは “move the sphere left.” のような曖昧な表現は、どの程度左にオブジェクトを移動すれば良いのかを、ユーザが意図した通りに判断することは容易ではない。今回 “Do” では、ある目標とする物体の位置 (Z 地点) に関して、基準となる距離  $x$  を定め、 $x$  によって移動等の操作を行った。 $x$  は次式により

表 1: 現在, “Do” で利用可能な英単語

Part of speech	Words	
	Mnemonic	
Verb	<b>PUT</b>	put, create, set, place, make
	<b>MOVE</b>	move, translate
	<b>ROTATE</b>	rotate, revolve
	<b>SCALE</b>	make ... smaller(larger, bigger), enlarge, shrink
	<b>PAINT</b>	paint, color, change the color of, alter the color of
	<b>DELETE</b>	delete, eliminate, erase, remove
	<b>UNITE</b>	merge, group, unite, take the union of
	<b>DISUNITE</b>	disunite, separate, ungroup
	<b>DIFFER</b>	cut, differ, take the difference of
	<b>INTERSECT</b>	intersect, take the intersection of
	<b>INVERT</b>	invert, take the inverse of
	<b>COPY</b>	copy, duplicate
	<b>NAME</b>	name, call
	<b>REGISTER</b>	register
	<b>UNDO</b>	undo
	<b>QUIT</b>	bye, exit, quit
Adjective	bigger, larger, smaller	
Adverb	right, left, up, down, in front, behind, back, center, little, more, to the right, to the left, to the behind, to the back, to the front, clockwise, anticlockwise, counterclockwise	
Preposition	above, behind, below, in front of, to the right of, to the left of, at, around	
Article	a, an, the	
Numeral	1, 2, 3, ...	
Conjunction	and	
Unit	dot(s), pixel(s), space(s)(スクリーン座標系) time(s), degree(s), mm, cm, m, km(ワールド座標系)	
Axis	x, y, z, x axis, y axis, z axis	
Color	red, green, blue, ...	
Object	primitive, newly created object, background, all	

表 2: Mnemonic

Mnemonic	Meaning
<b>PUT</b> <i>object dir(x, y, z)</i>	<i>object</i> を <i>dir</i> に置く操作
<b>MOVE</b> <i>object dir(x, y, z)</i>	スクリーン上の <i>object</i> を <i>dir</i> へ移動する操作
<b>ROTATE</b> <i>object θ(x, y, z)</i>	<i>object</i> を $\theta$ 度回転する操作
<b>SCALE</b> <i>object k(x, y, z)</i>	<i>object</i> を $k$ 倍スケールする操作
<b>PAINT</b> <i>object c(r, g, b)</i>	<i>object</i> を $c$ 色にペイントする操作
<b>DELETE</b> <i>object</i>	<i>object</i> をスクリーンから削除する操作
<b>UNITE</b> <i>object1 object2</i>	<i>object1</i> と <i>object2</i> をグループ化する操作
<b>DISUNITE</b> <i>object</i>	UNITE した <i>object</i> を解除する操作
<b>DIFFER</b> <i>object1 object2</i>	<i>object1</i> から <i>object2</i> を削る操作
<b>INTERSECT</b> <i>object1 object2</i>	<i>object1</i> と <i>object2</i> の積演算操作
<b>INVERT</b> <i>object</i>	<i>object</i> の否定演算操作
<b>COPY</b> <i>object</i>	<i>object</i> をスクリーン上に複製する操作
<b>NAME</b> <i>old-name new-name</i>	オブジェクト名を <i>old-name</i> から <i>new-name</i> に変更する操作
<b>REGISTER</b> <i>object</i>	<i>object</i> をプリミティブ辞書に新たに登録する操作
<b>UNDO</b>	前回の操作を <b>undo</b> する操作
<b>QUIT</b>	システムを終了する操作

表 3: Do で利用可能な文法 (BNF による)

```

<sentence> ::=
  <PUT> <obj.>
  | <PUT> <obj.> <adv.>
  | <PUT> <obj.> <prep.> <obj.>
  | <MOVE> <obj.> [<numeral> <unit>]+
  <adv.>
  | <MOVE> <obj.> <prep.> <obj.>
  | <ROTATE> <obj.> [<numeral> degree(s)]+
  [<direction>]+ [around <Axis>]+
  | <SCALE> <obj.> [<numeral> time(s)]+
  [on <Axis>]+
  | <PAINT> <obj.> <color>
  | <DELETE> <obj.> [<prep.> <obj.>]+
  | <UNITE> <obj.> and <obj.>
  | <DISUNITE> <obj.>
  | <DIFFER> <obj.> from <obj.>
  | <INTERSECT> <obj.> and <obj.>
  | <INVERT> <obj.>
  | <COPY> <obj.>
  | <NAME> <obj.> <obj.>
  | <REGISTER> <obj.>
  | <UNDO>
  | <QUIT>
<obj.> ::= [<art.>|<numeral>]+ [<color>]+
  <object>
<大文字表記> ::= 表 1: Verb(Mnemonic) 参照
<adv.> ::= 表 1: Adverb 参照
<adj.> ::= 表 1: Adjective 参照
<prep.> ::= 表 1: Preposition 参照
<numeral> ::= 表 1: Numeral 参照
<Axis> ::= 表 1: Axis 参照
<color> ::= 表 1: Color 参照
<art.> ::= 表 1: Article 参照
<object> ::= 表 1: Object 参照
<unit> ::= dot(s)|pixel(s)|space(s)|mm|cm|m|km
<direction> ::= clockwise | anticlockwise
  | counterclockwise

```

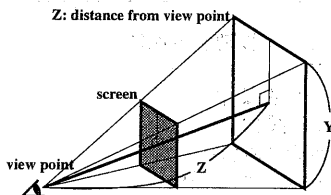


図 2: 移動距離の基準  
Fig. 2: Criterion for moving amount

定義した (図 2).

$$x = (Z \text{ 地点での基準となる距離}) = Y/10$$

つまり、スクリーン上で一定の距離だけ移動するように設定した。したがって、物体が視

点から遠くなると移動する距離も大きくなる。 $x$  を基準として、“move the sphere left little.” 等、“少し” といった表現については移動距離を  $x/10$  とし、“move the sphere left more.” 等、“もっと” といった表現には  $2x$  と解釈した。

前置詞、副詞の解釈でも、上記のような方法で位置決定を行った。そのため、前置詞が *above* であるのに、オブジェクト同士が交差してしまう矛盾が生じ得る。この問題は、Put[4] のように、オブジェクトにバウンディングボックスの情報を持たせ、オブジェクト同士の干渉検知を行うことで解決できる。今回、“Do” では前置詞 *on*, *in* についてはサポートしていないが、Put[4] と同様に、*on* では接地を、*in* ではオブジェクトを包含させれば良い。

## 2.5 ニーモニック変換例

テキストがニーモニックに変換される例を以下に示す。

<example1>

Text: Make a red sphere on the cube.  
Mnemonic: PUT *object dir(x, y, z)*

<example2>

Text: Make the sphere smaller.  
Mnemonic: SCALE *object k(x, y, z)*

この例で “*object*” は、オブジェクトデータにリンクしている。example1 では “a red sphere”, example2 では “the sphere” である。“*dir(x, y, z)*” は “on the cube” に相当する 3D 空間の座標値、“*k(x, y, z)*” は “the sphere” を拡大・縮小するための倍率である。

## 2.6 各操作の実現方法

本システムでは、オブジェクトのデータ構造として 2.2 で記述したように CSG を用いた。また、オブジェクトのデータとして、オブジェクトの名前、形状、色、位置情報をもつ。このことから、3D シーンに 3D オブジェクトが存在する様子を CSG で表現する必要がある。以下は、ニーモニックを CSG で表現している。

**PUT, DELETE:** 例えば、3つの物体 A, B, C が 3D 空間に存在している様子を CSG で表現すると、物体 A, B, C の和演算で表現できる。つまり、物体を置く操作 (PUT *object dir*) は CSG の和演算で実現できる。物体 C を 3D 空間から削除する操作 (DELETE C) は物体

A, B, C の CSG 木から葉 C を削除することにより実現できる。

**UNITE, INTERSECT, DIFFER, INVERT:** これらは一般の CSG 木の変形により実現できる。物体 A, B の CSG 演算後に生成された物体を D とすれば、移動、変形等の操作は物体 D を操作すればよい。また、変形後に物体 A, B を操作することも可能である。

**COPY:** オブジェクトの複製を指示した際は、ニーモニック COPY が使われる。COPY は複製したい CSG 木と、同じ構造をした CSG 木を新たに作成し、シーンに配置する。

**MOVE, SCALE, ROTATE:** オブジェクトの移動操作、スケーリング操作、回転操作はアフィン変換により行った。

**PAINT, NAME:** これらはオブジェクトの属性(色, 名前)を変えるだけでよい。

### 3 実験

#### 3.1 実験方法と結果

本システムを用いて、どの程度の 3D モデルを作成することができるのかを検証するために簡単な実験を行った。用いたプリミティブは“球”と“立方体”のみとし、球と立方体を配置・移動・変形し、3D モデルを作成した。その結果を図 3, 4 に示す。図 3 の左側には、シーンを作成する際に交わされたユーザとシステムとの会話である。右側はその生成結果である。また、同様な方法でユーザはシステムと会話し、図 4 のようなモデルも作成できる。表 4 にはユーザとシステムの会話に要した行数を記した。また、作成したオブジェクトはシステムのプリミティブ辞書に登録することができ、次回、同じオブジェクトを呼び出すことが可能である。

描画方法として Zバッファ法を用いた。512×512 の画像サイズで、これらの画像をシステムと会話しながら生成するのに 5~30 分程度かかった(表 4)。使用機材は Sun Ultra 30(300MHz) である。雪ダルマの場合、マウスのみでモデリング可能な Extreme3D で作成したところ約 10 分程度で同等なモデルが作成できた。しかし、POV-Ray のようなスクリプトを入力とするモデルで、このように短時間、かつ、インタラク

ティブにモデリングするのは困難である。

#### 3.2 考察

実験結果のユーザとシステムの会話を見ると、オブジェクト名を変更している箇所がいくつか存在する。これは、指示語(a, an, the 等)をシステムが認識していないため、ユーザが名前を変更することによって、過去のオブジェクト(同一のオブジェクト名)との区別を図ったためである。今後、指示語やオブジェクトデータ(大きさ, 色, 位置等)の参照によって、例えば、一番大きな球, 赤色の球, 左の球と言った具合に、同じ種類のオブジェクトでも指示語によって区別できるシステムにする予定である。

### 4 まとめ

本論文では、ハンディキャップを持つユーザ、CG の専門的知識の無いユーザ等にも CG が作成できることを目標とした自然言語に基づく 3D モデリングに関する一手法について述べた。

実験結果からも分かるように、自然言語をユーザインタフェースとして用いても、ある程度の 3D モデルは作成することが可能である。

今後の課題として、3.2 で述べたシステムの構築と共に、ユーザの意図をシステムが学習しながら理解していくシステムを作成、文献 [5][6][7] にみられるような非線形な変形操作の組み込みを考えている。

最後に、日本語が英語に翻訳可能なように、全世界の言葉も共通なことが言える。つまり、我々の定義したニーモニックに変換可能となる。そのインタプリタが作成可能であれば、マルチリンガルなシステムとなるであろう。

### 参考文献

- [1] R. A. Bolt, "Put-That-There: Voice and Gesture at the Graphics Interface", ACM-SIGGRAPH, Vol. 14, No. 3, pp. 262-270 (1980).
- [2] 西山晴彦, 大久保達真, 松下温, "Picnyck:風景描写文から風景画像の創造", 情報処理学会論文誌, Vol. 38, No. 5, pp. 997-1007 (1997).
- [3] 望月研二, 岸野文郎, "自然言語による 3次元画像へのアクセス", 信学技報, NLC92-30, PRU92-44, pp. 69-75 (1992).

Input-> put an orange sphere.  
 Input-> put an orange cube.  
 Input-> move the cube down.  
 Input-> move the cube 10cm down.  
 Input-> put an orange cube.  
 Change the past object(cube) name. -> c1  
 Input-> move the cube up.  
 Input-> move the cube 3cm up.  
 Input-> cut the c1 from sphere.  
 Input a name for the newly created object. -> test1  
 Input-> cut the cube from test1.  
 Input a name for the newly created object. -> test2  
 Input-> make the test2 1.2 times larger.  
 Input-> put an orange sphere.  
 Change the past object(sphere) name. -> s1  
 Input-> cut the sphere from test2.  
 Input a name for the newly created object -> ring  
 Input-> put a chocolate sphere.  
 Change the past object(sphere) name. -> s2  
 Input-> make the sphere smaller.  
 Input-> move the ring down little.  
 Input-> group the ring and sphere.  
 Input a name for the newly created object. -> saturn  
 Input-> make the saturn larger.

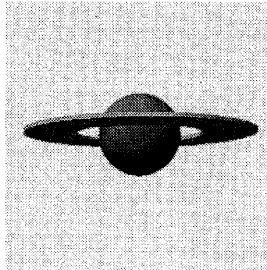
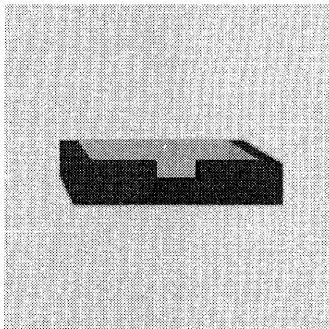


表 4: ユーザ入力に要した行数

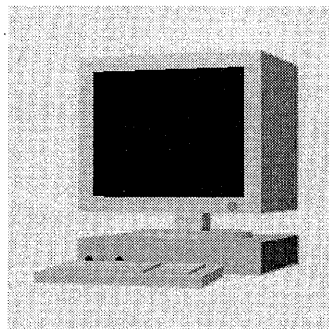
3D モデル	ユーザ入力	生成時間
土星	24 行	5 分
灰皿	42 行	8 分
コンピュータ	126 行	30 分
雪ダルマ	91 行	20 分

図 3: 実験 1:土星の例

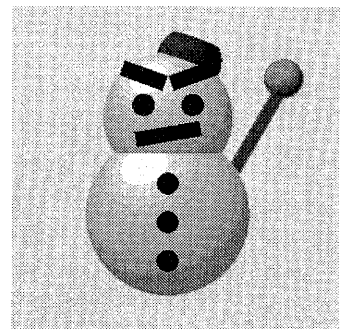
Fig. 3: An example of saturn



a. 灰皿



b. コンピュータ



c. 雪ダルマ

図 4: 実験 2:より複雑な画像生成例

Fig. 4: Examples of more complex shape

- [4] S. R. Clay and J. Wilhelms, "Put: Language-Based Interactive Manipulation of Objects", IEEE Computer Graphics and Applications, Vol. 16, No. 2, pp. 31-39 (1996).
- [5] A. H. Barr, "GLOBAL AND LOCAL DEFORMATIONS OF SOLID PRIMITIVES", Computer Graphics, Vol. 18, No. 3, pp. 21-30 (1984).
- [6] 福嶋茂信, 美濃導彦, 池田克夫, "画像情報と言語的指示を用いた曲面モデルの作成", 画像の認識・理解シンポジウム (MIRU'96), Vol.1, pp. 301-306 (1996).
- [7] 福嶋茂信, 美濃導彦, 池田克夫, "言語表現を利用する 3次元形状モデルの作成", 情処研報, Vol. 97, No. 79, pp. 55-60 (1997).
- [8] 山田篤, "自然言語による視覚表現と対称間の位置関係の対応に関する考察", 信学技報, NLC95-34, PRU95-139, pp. 27-32 (1995).
- [9] 佐藤義雄著, "入門グラフィックス", アスキー出版局 (1984).
- [10] 太田昌孝, 竹内あきら, 大口孝之, "応用グラフィックス", アスキー出版局 (1986).
- [11] J. R. Levine, Tony, Doug, "lex & yacc", アスキー出版局 (1996).
- [12] 五月女健治, "yacc/lex", テクノプレス (1996).