

Blenderを用いた自然言語による 3次元コンピュータグラフィックス

中畑 敦夫† 伊藤 秀昭†† 福本 尚生†† 和久屋 寛†† 古川 達也††

†佐賀大学 理工学部電気電子工学科

††佐賀大学 大学院工学系研究科

{nakahata, hideaki, fukumoto, wakuya, tach}@ace.ec.saga-u.ac.jp

1 はじめに

3次元コンピュータグラフィックス(3D-CG)を作成する際、ユーザインタフェースとしてマウスやペンタブレットなどがよく用いられるが、これに加えて自然言語を使用することができれば便利である。例えば、「画面上の全ての赤い物体を消す」という処理は、赤い物体が多数存在する場合、マウスで一つずつ選択して削除するよりも、「画面上の全ての赤い物体を消す」という指示そのものをコンピュータが理解してくれた方が作業量が減る。また、肢体不自由者が3D-CG作成用ソフトウェアを使用する場合、マウスなどではなく音声で操作できることが望ましいが、その際にも、日常的に使用している自然言語を用いることができれば操作が一層容易になる。

そこで本研究では、世界中で広く使用されている3D-CG作成用ソフトウェアであるBlender [1]に、自然言語による指示を受け付ける機能を追加した。自然言語による3D-CGの作成については、Do [2]やWordsEye [3]をはじめとして多くの研究があるが、その大半は、自然言語だけが使用可能な独自のCG作成システムを提案している。それに対し、本研究では、マウスなどによって高度な描画を行うことが可能なソフトウェアであるBlenderに、自然言語による指示機能を付加して操作性を高めるというアプローチをとり、マウスと自然言語のそれぞれの長所を活かすことができるようにする。Blenderを用いた3D-CG作成については、先駆的な試作システムの報告 [4]はあるものの、実用的なものは、未だ開発されていない。また、物語文からアニメーションを作成しBlenderを用いて表示するという研究 [5, 6]は存在するが、直接的にBlender上でのCG作成を目的としたものは従前には無かった。

また、自然言語による入力可能なシステムの多くでは、固定されたある範囲の語彙しか入力文中で使用

できないが、本研究では、未知の単語が使用されても、システムが自動的に辞書を調べて対応できるようにした。辞書としては、フリーな辞書であるOPTED [7]およびWordNet [8, 9]を用いた。

以下では、今回開発したシステムの概要を述べ、動作結果を報告する。

2 自然言語を用いた3D-CG作成

2.1 Blender

本研究ではBlenderを用いて3D-CGを作成する。Blenderは、フリーでありながら、モデリング(3次元形状の作成)から、レンダリング(画像化)までを高いレベルで行うことができるソフトウェアである。世界中のユーザに利用されており、活発に改良も加えられている。入力のためのユーザインタフェースとしては、マウスとキーボードを用いるのが基本であるが、本研究では、これを改良し、自然言語による入力もできるようにした。

2.2 NLTK

入力された自然言語を解析するために、NLTK (Natural Language Toolkit) [10]を使用した。NLTKは、自然言語処理のためのオープンソースのライブラリである。また、NLTKはPython言語で書かれており、Blenderには、Pythonスクリプトを実行させる機能が備わっているため、Blenderからの利用が容易である。

2.3 動作概要

今回開発したシステムの動作の流れを図1に示す。

入力は、将来的には音声による入力も可能とする予定であるが、今回は英語でのテキスト入力だけに限定

した。入力されたテキスト（例：“Add a red cube.”）は、まずNLTKの形態素解析および構文解析ルーチンにより解析される。あらかじめ、特定の語彙からなる文脈自由文法(図2)を与えておき、最初はこれを用いて解析する。図2において、意味解析を容易にするために、形容詞はAdj1とAdj2に分けた。名詞も同様である。文法中の品詞の一覧は表1の通りである。解析にはNLTKのparse_cfg, RecursiveDescentParser, およびnbest_parse関数を用いた。

もし解析が成功した場合、次の意味解析と3Dモデルの作成へと進み、入力文の内容に応じて3DモデルがBlender上に生成される。例えば、“Add a red cube.”という入力文の場合、赤い立方体が仮想空間内に追加される。この部分の処理は、Blenderに適切な処理(この場合は立方体の追加)を行わせるための自作関数をPythonで作成することによって実現させた。その後、レンダリングが行われ、作成したCG画像が画面に表示される。

また、形態素解析・構文解析に失敗した場合、入力文の中に未知単語が含まれていないかをチェックする。辞書の中に未知単語があった場合は、これを図2に示す文法に追加して、形態素解析・構文解析をやり直す。詳細は2.4節で説明する。それ以後は、上述の解析が成功した場合と同じ処理が行われる。

入力文の処理が全て終わると、次の入力文が入力されるのを待つ。以降は同じ処理を繰り返す。

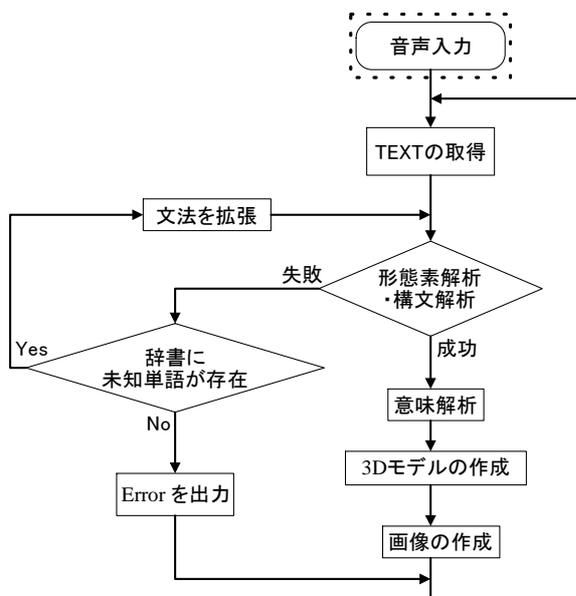


図1: 流れ図

文脈自由文法

```

S -> V NP|V NP PP|V Adj2 NP
V -> "add"|"delete"
NP -> Det N1|Det Adj1 N1
PP -> P Det N2
P -> "in"|"at"|"on"
Det -> "a"|"an"|"the"
Adj1 -> "red"|"green"|"blue"
Adj2 -> "all"
N1 -> "cone"|"cube"
N2 -> "left"|"right"
  
```

図2: 最初の文法

表1: 品詞一覧

品詞	意味	品詞	意味
S	文	Det	限定詞
V	動詞	Adj1	形容詞
NP	名詞句	Adj2	形容詞
PP	前置詞句	N1	名詞
P	前置詞	N2	名詞

2.4 未知単語の追加

前節で説明したように、最初は図2の文法を用いて入力文を解析するが、入力文に未知単語が含まれていた場合(例：“Add a red dice.”), 解析に失敗する。この場合でも、できるだけユーザの意図通りに動作するように、以下のようにして未知単語に対処できるようにした。

まず、入力文から未知単語を抜き出す(例：“Add a red dice.”の中では“dice”が未知単語)。次に、英英辞書のOPTED(The Online Plain Text English Dictionary)[7]を用いて、その品詞を調べる。今回は、簡単のために、名詞だけを対象とすることとし、未知単語がOPTEDに名詞として掲載されている場合にだけ、以下の処理に進むものとした。

次に、図2の文法のN1の行に、未知単語を追加する。そして、この新たな文法を用いて形態素解析・構文解析を行う。解析に成功すれば、次の意味解析に進む。

意味解析では、その未知単語(例：“dice”)がどのような形状の物体か推測する必要がある。今回は、概念辞書のWordNet[8,9]を用いて、これを推測した。具体的には、未知単語が物体に関する既知の単語(“cube”と“cone”)のどちらかと概念的に近いかを計算し、近い

方の既知単語の形状を未知単語の形状とした。

ここで、概念的な近さは、以下のように計算した。まず、それぞれの既知単語について、WordNetにおける代表的な意味概念を適当に選び、登録しておく。今回は、“cone”に対しては tree.n.01 など 74 個、“cube”に対しては block.n.03 など 73 個を登録した（ここで、tree.n.01 は、WordNetにおいて tree という単語の持つ、1 個目の名詞的意味概念を示す。block.n.03 は block の 3 個目の名詞的意味概念である）。以下、既知単語 w に対して、この代表的意味概念の集合を $X(w)$ と書くことにする。次に、未知単語の持つ概念の集合を WordNet から求め、 Y とする。そして、既知単語 w と未知単語との距離を、 $\min_{x \in X(w), y \in Y} d(x, y)$ と計算した。ここで、 $d(x, y)$ は二つの概念 x と y の距離であり、様々な定義が提案されているが、今回は Leacock & Chodorow の尺度 [11] を用いた。この尺度は、人間の判断基準と高い相関を示す尺度 [12] である。計算には NLTK の lch_similarity 関数を用いた。

3 動作結果

3.1 基本動作

本システムを起動した初期画面は図 3 のようになっている。画面の左側は、マウスなどで編集可能な 3D 空間である。右側下方に、テキストを入力する領域がある。右側上方にはレンダリング結果が表示される。

ここで、例えば、テキスト入力欄に

例文 1: Add a red cube in the left.

と入力すると、図 4 のように、指示通りの結果が表示される。ここで、“left” という指定は漠然性があるが、今回は左側のランダムな場所に配置されるものとした。

また、図 5(a) の状況で、

例文 2: Delete all the red cones.

と入力すると、図 5(b) のようになる。図 5(a) には、赤いコーンと緑のコーンが多数存在するが、例文 2 により、赤いコーンが全て削除されている。これは、マウスよりも自然言語入力の方が便利である場合の例となっている。赤いコーンの数が多く、近くに緑のコーンが存在している場合には、マウスを用いて短時間に赤いコーンを全て削除することは難しい。

3.2 未知単語への対処

前述の通り、今回は、物体の名詞としては“cube”と“cone”だけが既知である。しかし、例えば

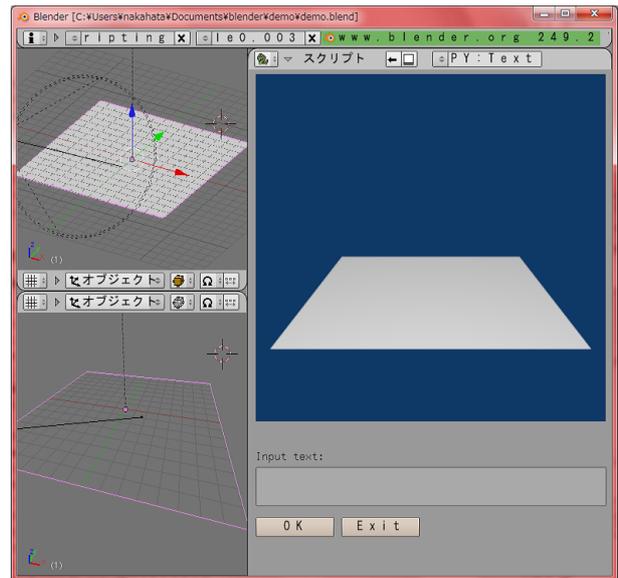


図 3: 基本画面

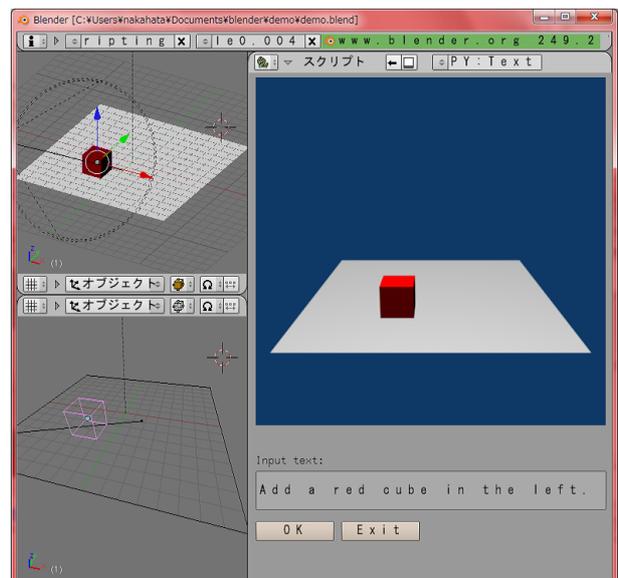
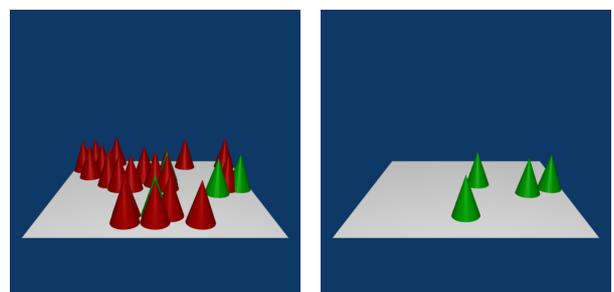


図 4: 例文 1 の結果



(a) 例文 2 の入力前 (b) 例文 2 の入力後

図 5: 例文 2 の効果

例文 3: Add a red dice in the left.

と入力すると、2.4 節で説明した仕組みによって、未知単語の“dice”の形状が“cube”または“cone”と推測され、どちらかの形状で 3D モデルが追加される。

推測がどれくらい正確であるかを調べるために、“dice”を含む 20 個の単語 (表 2) について推測結果を調べた。これらの 20 個の単語は、システムの内容を知らない成人 2 名に、cube と cone に類似する形状の物体を答えてもらうことによって収集した。ただし、英訳は著者らの二人 (中畑, 伊藤) が行った。ここで、収集された単語が、2.4 節で各既知単語の代表的概念として与えた概念のいずれかを持つ場合は、正解となることが明らかであるため表 2 から除外してある。

表 2 において、上側の 10 個は“cube”, 下側の 10 個は“cone”と推測されるのが正解の単語である。結果は“cube”が正解率 90%, “cone”は正解率 60% となり、全体では正解率 75% であった。

表 2: 推測結果とその正誤

未知単語	意味	推測結果	判定
dice	サイコロ	cube	○
skyscraper	高層ビル	cube	○
tofu	豆腐	cube	○
toaster	トースター	cone	×
oven	オーブン	cube	○
refrigerator	冷蔵庫	cube	○
album	アルバム	cube	○
bookshelf	本棚	cube	○
locker	ロッカー	cube	○
cellphone	携帯電話	cube	○
funnel	じょうご	cube	×
pylon	鉄塔	cube	×
fir	もみの木	cone	○
infundibulum	漏斗	cone	○
megaphone	メガホン	cone	○
umbrella	傘	cube	×
bride	花嫁	cone	○
conch	巻き貝	cone	○
spire	尖塔	cube	×
cracker	クラッカー	cone	○

4 おわりに

本研究では、Blender に、自然言語のテキスト入力により 3D-CG を作成する機能を追加した。現状では

対応できる動詞や名詞の数が少なく、文法も限られているため、あまり多様な入力文を受け付けることができないが、基本文法を拡張することによって、使用できる文を増やすことは可能であり、今後の課題である。

また、未知の名詞が入力された場合に、文法を自動的に拡張し、形状を推測する機能を作成したため、ある程度柔軟に対応することもできるようになった。しかし推測性能が十分高くはないので、これを改善することも今後の課題である。

今後の課題としては、上記に挙げたもののほかに、音声入力や、日本語による入力を可能にすることが挙げられる。

参考文献

- [1] The Blender Foundation. <http://www.blender.org/> (日本語版 <http://blender.jp/>)
- [2] 河合善之, 岡田稔. 自然言語による幾何形状モデリングと画像合成の一手法. 情報処理学会論文誌, Vol. 42, No. 5, pp. 1161–1168, 2001.
- [3] Bob Coyne and Richard Sproat. WordsEye: an automatic text-to-scene conversion system. In SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 487–496, 2001.
- [4] Peter Gorniak and Deb Roy. BISHOP | BLENDER: Spatially Grounded Language Understanding in 3D Modelling Software. In Proceedings of the NAACL 2004.
- [5] Kevin Glass and Shaun Bangay. Automating the creation of 3D animation from annotated fiction text. In IADIS-CGV '08: Proceedings of the IADIS International conference on Computer Graphics and Visualization, pp. 3–10, 2008.
- [6] Patrick Ye. Natural Language Understanding in Controlled Virtual Environments. PhD thesis, University of Melbourne, 2009.
- [7] The Online Plain Text English Dictionary. <http://www.mso.anu.edu.au/~ralph/OPTED/>
- [8] George A. Miller. WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, No. 11, pp. 39–41, 1995.
- [9] Christiane Fellbaum. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press, 1998.
- [10] Natural Language Toolkit. <http://www.nltk.org/>
- [11] Claudia Leacock and Martin Chodorow. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum (ed.) WordNet: An electronic lexical database. pp. 265–83, Cambridge MA: MIT Press, 1998.
- [12] Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. Computational Linguistics, Vol. 32, Issue 1, pp. 13–47, 2006.