自然言語による 3DCG 作成における未知形状推定手法の比較

中畑 敦夫　伊藤 秀昭　福本 尚生　和久屋 寛　古川 達也
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（3DCG）を作成する際，通常はマウスやペンタブレットなどの入力デバイスが用いられるが，これらに加えて自然言語による指示が可能となれば操作が一層容易になる．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 3DCG 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，操作が容易になる．

そこで，筆者らは以前の研究 [1, 2, 3] で，世界中で広く使用されている 3DCG 作成ソフトウェアである Blender[4] に，自然言語による指示に基づいて描画を行う機能を追加した．その特色として，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，マウスやペンタブレットなどの入力デバイ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していることが挙げられる．

また，自然言語による入力が可能なシステム（例えば [5, 6]）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，筆者らのシステムでは，未知語が使用されても，未知語の形状を推定することによって，未知語の 3DCG を描画できるようにした．例えば，“add a red dice”という指示がシステムに入力されたとして，その中の単語 “dice”が未知語であった場合，未知語の形状推定によって，未知語 “dice”の形状を立方体などと推定し，その形状の赤い物体を画面に描画する．以前のシステム [1, 2] では，WordNet[7, 8, 9] という概念辞書を用いて，未知語の形状を推定する手法を用いていた．

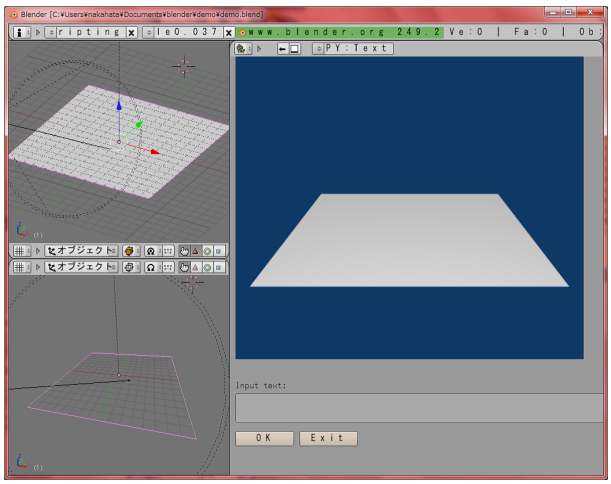
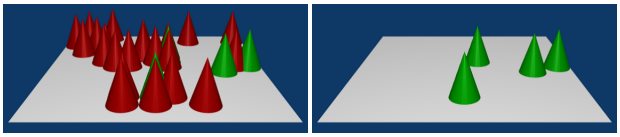


図 1: Blender の作業画面



(a) 入力前　(b) 入力後
図 2: “delete all the red cones”を入力した場合の動作例

しかし，WordNet を用いた手法では，設計者が各形状の具体例（例えば，立方体の例としてブロックや本など; 詳細は後述）をあらかじめ多数登録しておかなければうまく推定できず，手間がかかるという欠点があった．そこで本研究では，手間のかからない手法として，写真共有サイトである Flickr[10] の画像を用いて，未知語の形状を推定する手法を実装し，以前の手法との比較を行った．その結果，画像処理に若干時間がかかってしまうものの，以前の手法と同等の推定精度を得ることができたので以下に報告する．

2　システムの概要

以前の研究 [1, 2] で，3DCG 作成ソフトウェアである Blender 上で英語のテキスト入力に対する処理が可能となった．図 1 に作業画面を示す．図 2 はその一部（右上の 3DCG 表示部分）を抜き出したものであるが，

例えば，図 2(a) において，赤い円錐をすべて削除したい場合，“delete all the red cones”と入力すれば，図 2(b) のように指示通りに処理される．

しかし，“add a red dice”と入力した場合，未知語，すなわちシステムに登録されていない単語である “dice” が含まれるため処理に失敗する．これに対処するためには，システムに登録された単語を増やすという手段もあるが，その場合ユーザが使用する可能性のある単語をすべて事前に登録する必要があり実現困難である．そこで，筆者らのシステムでは，未知語の形状を推定する機能 [1, 2] を実装した．推定手法とその評価については次章で詳しく述べる．なお，推定が 100%うまくゆくとは限らないので，誤った場合にユーザとの対話によってこれを訂正する機能 [3] も実装している．

3 形状推定手法

本論文では，筆者らのシステムでこれまで使用していた WordNet を用いた手法と，今回新たに提案する，Flickr 画像を用いた手法とを比較する．以下で，それぞれの推定手法について説明する．

今回は，システムにとって既知の形状は “cube”および “cone”の二つだけであるとし，未知語が入力された場合にその形が “cube”あるいは “cone”のいずれであるかを推定させた．そのため以下ではそれに即して両手法を説明する．ただし，両手法とも既知の形状が三つ以上であっても適用可能である．

3.1 辞書による推定手法

辞書を用いた推定手法 [1, 2] では，概念辞書である WordNet を用いて形状推定を行った．手法のフローチャートを図 3 に示す．まず，ユーザからの指示文がシステムに入力されると，それを，あらかじめ与えておいた文脈自由文法により解析する．指示文に未知語が含まれていた場合に，形状推定処理が開始される．今回は，前述のように，“cone” と “cube” の形状が既知であるとし，未知語がそのどちらに近いかを推定した．

推定は，WordNet を用い，未知語が既知語（“cone” と “cube”）のどちらと概念的に近いかを計算し，近い方の既知語の形状を未知語の形状とみなした．

ここで，概念的な近さは，以下のように計算した（図 4）．まず，それぞれの既知語について，WordNet における代表的な意味概念を事前に選び，登録しておく．今回は，“cone ” に対しては “actor.n.01”など 74 個，“cube ” に対しては “block.n.01”など 73 個を登録した（ここで，“actor.n.01”は，WordNet において “actor” という単語の持つ，1 個目の名詞的意味概念を示す．

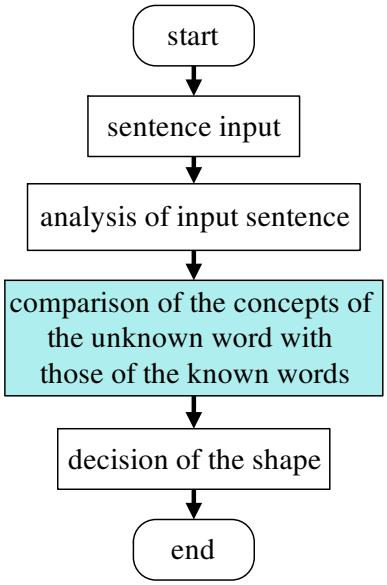


図 3: WordNet による推定手法の流れ図

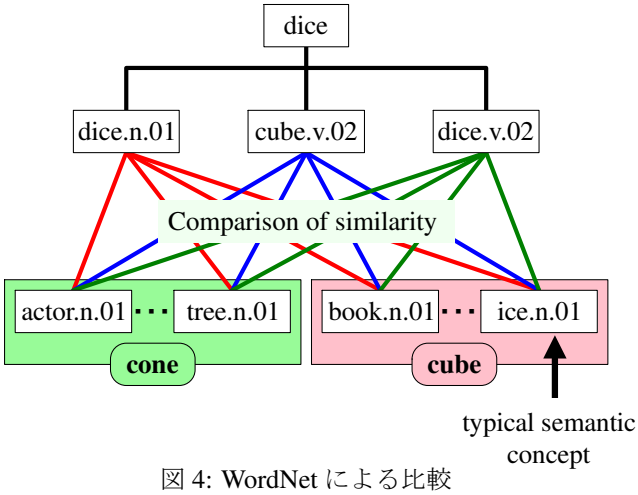


図 4: WordNet による比較

“block.n.01”は block の 1 個目の名詞的意味概念である）．以下，既知語 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，未知語の持つ概念の集合 (“dice.n.01”, “cube.v.02”など) を WordNet から求め，Y とする．そして，既知語 w と未知語との距離を，$\min_{x\in X(w),y\in Y} d(x;y)$ と計算する．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，今回は以下に示す Leacock & Chodorow の尺度 [12] を用いた．計算には NLTK[11] の lch_similarity 関数を用いた．

$$\mathrm{sim} = -\log \frac{p}{2D}$$

ここで，p は概念間の最短距離を示し，D は最深ノードの深さを示している．この尺度は，人間の判断基準と高い相関を示す尺度 [13] であり，本推定課題においても，他の尺度よりも高い推定精度 [2] が得られて

いる．

本手法では，上述のように，各既知語に対してその代表的な意味概念を登録しておく必要がある．そのため既知語を増やす場合に手間がかかってしまうという欠点がある．

3.2 画像による推定手法

画像を用いた推定手法 [14] では，著作権フリーの画像が多数登録されている Web サイトの一つである Flickr の画像を用いて形状推定を行った．手法のフローチャートを図 5 に示す．ユーザからの指示文を入力し，それを解析するところまでは WordNet を用いた手法と同じである．その後，Flickr から未知語の画像を 5 枚ダウンロードする．具体的には，Flickr 上でその未知語を検索語として画像検索をかけ，結果として表示された上位 5 枚をダウンロードした．次に，これらの画像を，既知の形状である “cone” と “cube” の画像と比較する．図 6 に比較方法の詳細を示す．まず，既知の形状である “cone”や “cube”の画像から SURF 特徴量 [15] を抽出し，各特徴点に対して 128 次元の特徴ベクトルを作成する．これを基準ベクトルと呼ぶことにする．ここで，SURF の閾値を調整し，“cone”と “cube”ともに特徴点が 5 個となるようにした．次に，Flickr で集めた画像からもそれぞれ特徴点を複数個抽出し（何個抽出されるかは画像ごとに異なり，図 6 では $N_1, N_2, ..., N_5$ 個と記している），それぞれの特徴点で 128 次元の特徴ベクトルを作成する．そして，各特徴ベクトルについて，最もユークリッド距離の近い

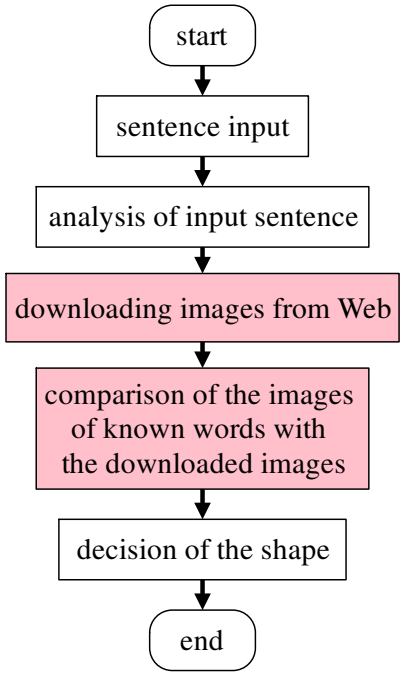


図 5: Flickr 画像を用いた推定手法の流れ図

基準ベクトルを求め，それが属する形状 (“cone” または “cube”) に一票を投じる．これをすべての特徴ベクトルについて行い，投票された票数が多かった方の形状を最終的な推定結果とする．

4 評価実験とその結果

本研究を知らない 2 名から，“cone”と “cube”の形状の単語を表 1 のようにそれぞれ 10 個ずつ集め，これらの計 20 個の単語のそれぞれについて，WordNet を

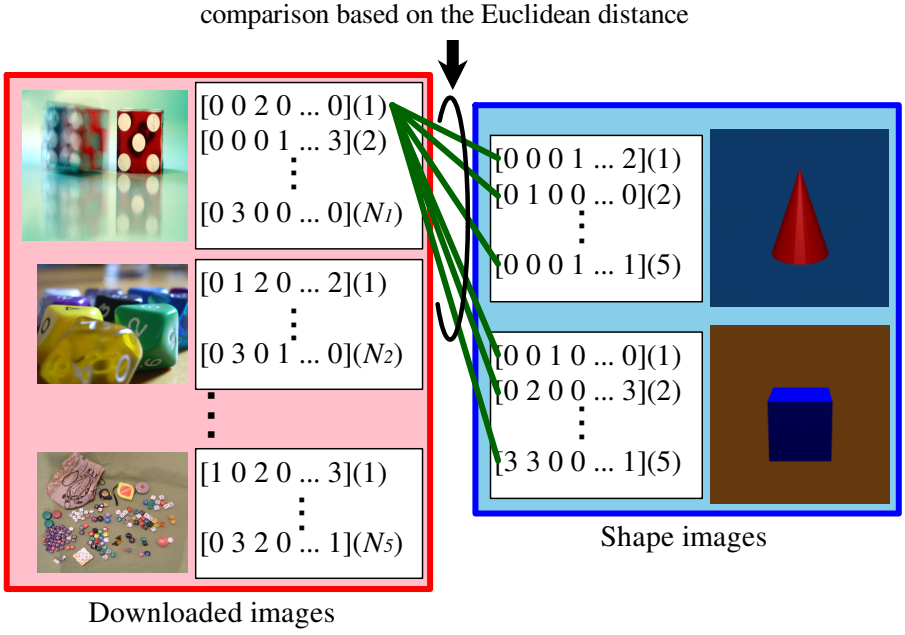


図 6: Flickr 画像による比較

表 1: 評価に用いた単語の一覧

	cone	cube
1	funnel	dice
2	pylon	skyscraper
3	fir	tofu
4	infundibulum	toaster
5	megaphone	oven
6	umbrella	refrigerator
7	bride	album
8	conch	bookshelf
9	spire	locker
10	cracker	cellphone

表 2: 比較結果

Method	Correct estimation	Time
WordNet	75 %	0.58 [s]
Flickr	75 %	25.49 [s]

表 3: 処理時間の内訳

	Download	Image processing	Total
Time	8.85 [s]	16.64 [s]	25.49 [s]

用いた手法および Flickr を用いた手法で形状推定を行い，正しく “cube” または “cone” と推定されるかどうかを調べた．

それぞれの手法の評価を正解率と処理時間で行った結果を表 2 に示す．ここで，処理時間は 1 単語あたりの平均処理時間を示している．実験には Intel Core i7 870@2.93GHz の PC を使用した．

それぞれの手法の正解率を比較すると，どちらも 75% で同じとなった．ただし，今回，Flickr から集める画像は 5 枚としたが，枚数の違いによって，形状推定の正解率は大きく変化したので，これについては今後の検討課題である．

また，処理時間については，従来の WordNet の手法のほうが短かかった．処理時間の内訳を表 3 に示す．ここでも，処理時間は 1 単語あたりの平均処理時間を示している．画像処理に長い時間がかかっていることがわかる．

5 おわりに

本研究では，未知語の形状推定手法について，WordNet を用いた従来手法と，Flickr を用いた新たな手法との比較を行った．その結果，Flickr を用いた手法でも WordNet を用いた手法と同程度の推定精度が得られることが判明した．Flickr を用いた手法では，現状では処理時間が比較的長くかかってしまうが，設計者にかかる負担が少ないという利点があるため，今後は処理時間を短くするよう改良してゆきたい．また，Flickr からダウンロードして比較する画像の枚数によって，形状推定精度が大きく変化するという現象も見られたので，今後は，枚数に関係なく，安定した正解率が得ることのできる手法の開発を行う予定である．

参考文献

[1] 中畑敦夫, 伊藤秀昭, 福本尚生, 和久屋寛, 古川達也. 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. 言語処理学会第 17 回年次大会論文集, pp. 200–203, 2011.

[2] A. Nakahata, H. Itoh, H. Fukumoto, H. Wakuya, and T. Furukawa. A natural-language-based 3D-CG system with shape estimation for unknown words. In Proceedings of the SICE Annual Conference 2011, SaB07-06, pp. 2847–2850, 2011.

[3] 中畑敦夫, 伊藤秀昭, 福本尚生, 和久屋寛, 古川達也. 自然言語を用いた 3 次元コンピュータグラフィックス作成における対話による未知語の形状推定. 平成 23 年度電子情報通信学会九州支部学生会講演会講演論文集, D-31, 2011.

[4] The Blender Foundation. http://www.blender.org/ (日本語版 http://blender.jp/)

[5] 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.

[6] B. Coyne and R. 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.

[7] WordNet. http://wordnet.princeton.edu/

[8] G. A. Miller.WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, No. 11, pp. 39–41, 1995.

[9] C. Fellbaum. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press, 1998.

[10] Flickr. http://www.flickr.com/ (2011/01/23 時点の画像データを使用)

[11] Natural Language Toolkit. http://www.nltk.org/

[12] C. Leacock and M. 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.

[13] A. Budanitsky and G. Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. Computational Linguistics, Vol. 32, Issue1, pp. 13–47, 2006.

[14] 黄瀬 浩一, 岩村 雅一. 3 日で作る高速特定物体認識システム. 情報処理学会誌 , Vol. 49, No. 9, pp. 1082–1089, 2008.

[15] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, pp. 404–417, 2006