
要素指定と候補提示による 3 次元図形の描画

A Suggestive Interface for 3D Drawing

五十嵐 健夫 John F. Hughes*

Summary. This paper introduces a new type of interface for 3D drawings that improves the usability of gestural interfaces and augments typical command-based modeling systems. In our suggestive interface, the user gives hints about a desired operation to the system by highlighting related geometric components in the scene. The system then infers possible operations based on the hints and presents the results of these operations as small thumbnails. The user completes the editing operation simply by clicking on the desired thumbnail. The hinting mechanism lets the user specify geometric relations among graphical components in the scene, and the multiple thumbnail suggestions make it possible to define many operations with relatively few distinct hint patterns. The suggestive interface system is implemented as a set of suggestion engines working in parallel, and is easily extended by adding customized engines. Our prototype 3D drawing system, Chateau, shows that a suggestive interface can effectively support construction of various 3D drawings.

1 はじめに

既存の 3 次元モデリングシステム(CAD)は、熟練したユーザが複雑な図形を正確に記述するに合わせた用途に向けてデザインされており、インタフェースは概して複雑であり簡単な形状を生成するのに時間がかかることが多い。このような状況に対して、頭の中にあるアイデアを短時間で形として表現することを目的とした、手書きジェスチャー入力による 3 次元描画システムがいくつか提案されている[11][20]。これらのシステムでは、メニューやボタン操作をすることなく画面上に線を描いていくだけで、初心者でも比較的簡単に 3 次元形状を表現することができる。

ジェスチャー入力による 3 次元描画は、研究段階の成果としては一定の評価を受けているものの、実際に広く利用するためにはいくつかの問題点が課題として残されている。まず、既存のジェスチャー入力によるシステムは、非常にラフな形状表現に特化しており、接続性や合同性といった基本的な幾何学的関係を表現することが難しい。次に、限られた数の基本要素の組み合わせで表現できるジェスチャーの数には限りがあるため、数多くの操作を実装することが困難である。最後に、ジェスチャーの

* Takeo Igarashi, 東京大学情報理工学系研究科電子情報学専攻, John F. Hughes ブラウン大学計算機科学科

動作を確認するためには操作を完結しなければならないため、特に初心者の場合、結果を見てやり直すといった思考錯誤を繰り返す必要があり、学習が困難であるという問題がある。

本論文では、上記のような課題を解消する試みとして、ジェスチャー入力を拡張した新しいインタフェースを提案する。提案する手法では、まずユーザが画面内にある図形要素をクリックしてハイライトすることによって、システムに対して目的とする操作を示唆するヒントを与える。システムの方では、与えられたヒントに基づいて、可能性のある操作を複数推定し、それらを適用した結果を、複数の候補として縮小図にして提示する（図1）。その中にユーザの気に入ったものがあれば、縮小図をクリックすることで操作が適用され、気に入ったものがなければシステムの提案を無視して作業を継続する。

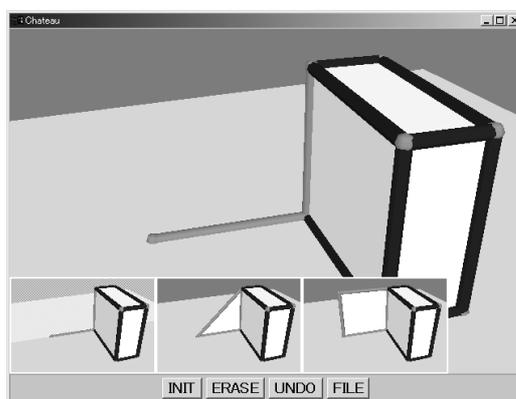


図1. プロトタイプシステムの画面例

ユーザが画面中の線分をハイライト（灰色の線分）すると、それに基づいてシステムが次の操作を推定し、その結果を 複数の候補として縮小図にして画面下部に並べて提示する。

本インタフェースは、ユーザの入力に対して、ジェスチャー入力のように既定の結果を即時に反映させるのではなく、複数の可能性を提示して、ユーザの確認を得ようとするものである。このような手法によって、前述の問題点を緩和することができる。まず、要素指定操作によって、既存の図形要素を入力の一部として自然な形で利用することができるので、対称性や合同性といった幾何学的な関係を表現することが容易になる。次に、同一の入力パターンに対して複数の操作を割り当てることができるので、より多くの操作を実装することが可能となる。最後に、操作の結果が縮小図として提示されるために、システムの動作に不慣れな場合でも学習が比較的容易となるメリットがある。

本稿では、提案するインタフェースに基づいて実装したプロトタイプシステム (Chateau) について紹介する。なお、カラー図版、デモプログラム、およびデモビデオについては <http://www.mtl.t.u-tokyo.ac.jp/~takeo/> から入手可能である。

2 関連研究

操作履歴に基づいて、次の操作を予測して提示する予測インタフェースがいくつか提案されている[5][17]。本論文で紹介する手法は、履歴に基づくのではなく画面上の静的な位置関係に基づいて次の操作を推定し提示する点で異なっている。

複数候補の提示については、文字認識や音声認識、日本語入力などで頻繁に用いられている[14][16]。コンピュータグラフィクス分野では、最適なパラメータをユーザの入力に基づいて探索するインタフェースとして利用されている[15][18]。

ユーザの操作内容から、幾何学的な制約を推定するシステムとして、Brair[7]やRockIt[12]などがある。また、ユーザの提示する例からレイアウトアルゴリズムを推定し、その結果を複数の候補として提示して確認を求めるシステムも存在する[8]。我々の手法は、関係を規定するものとしての制約を推定するのではなく、パターンマッチングによって画面内容を直接書き換えるものである。

本稿で提案する手法は、整形と予測に基づく幾何学的図形の描画システム Pegasus [9][10] を拡張したものである。Pegasus では、整形と予測時に全画面内を走査対象としていたために、図形の規模が大きくなってきたときに候補数が多くなりすぎるという問題点があったが、本手法ではユーザが走査すべき対象をヒントとして提示することによってこの問題を防いでいる。また3次元に拡張するにあたって、画面の混乱を防ぐために、画面内に直接候補を表示する代わりに、縮小図による表示を導入した。

3 ユーザインタフェース

ユーザの側から見た場合の本システムの動作について説明する。本システムには、通常の3次元描画システムに見られるような、円柱や直方体といった描画プリミティブの選択ボタンや、反転や移動といったコマンドメニューが存在せず、3次元表示画面上で線分を描くこと、既存の線分をクリックしてハイライト（あるいはその取り消し）すること、および提示された候補をクリックで選択すること、によって描画が進行する。現在の実装では、円弧や曲線はサポートされておらず、すべて直線と平面からなる3次元図形のみを対象としている。

3.1 基本操作

描画操作はすべてマウスの左ボタンで行われ、右ボタンはカメラ操作専用に使われる[19]。左ボタンでのドラッグ操作によって画面上に線が引かれる。描かれた2次元の線分は、画面上で対応する3次元要素の上に投影されて3次元の線分に変換される。具体的には、白い板で表現されたポリゴン、緑色の地面、および後で紹介する「描画面」の上に線を描くことができる。またドラッグ中にマウスを前後に繰り返し動かすことで線分を消去することができる。画面上の既存の3次元線分をクリックすることによって、当該線分をハイライトしたりハイライトを取り消したりすることができる。ダブルクリックすると、クリックされた線分につながっているすべての線分がハイライトされる。ポリゴンをクリックするとそのポリゴンを囲む全線分がハイライトされ、また何も無い部分をクリックするとすべてのハイライトが取り消される。なお、新たに描かれた線分は自動的にハイライトされる。ハイライト操作は、後で説明する

線分予測と縮小図による候補提示の他、線分描画時におけるスナップ動作[2]の制御にも使われる。すなわち、ある既存線分がハイライトされていると、描画中の線分はその線分と平行あるいは合同な形になるようにスナップする。スナップ動作については、ドラッグ中に他の要素に触れることでスナップ対象が動的に変化する。Drafting Assistant [1]に準じた機能も実装している。

3.2 線分の描画予測

本予測機能は、Pegasus システムに実装されていたものとほぼ同等であるが、画面内のすべての線分を走査して適合するものを探すのではなく、ハイライトされた線分のみを対象として予測の手がかりとする。すなわち、新しい線分がハイライトされると、その線分と合同な線分をハイライトされた線分の中から探し出し、その周囲の線分を新しい線分の周りにコピーしてくる(図 2)。予測結果はピンク色の線分として 3 次元画面内に直接提示される。気に入ったものがある場合には、クリックすることによって確定操作となるが、無視して次の操作に進むこともできる。本手法は、局所的に対称あるいは合同な形状を描くのに適している。なお、本論文では、図 2 以外の図については混乱を防ぐため予測機能部分は取り除いてある。

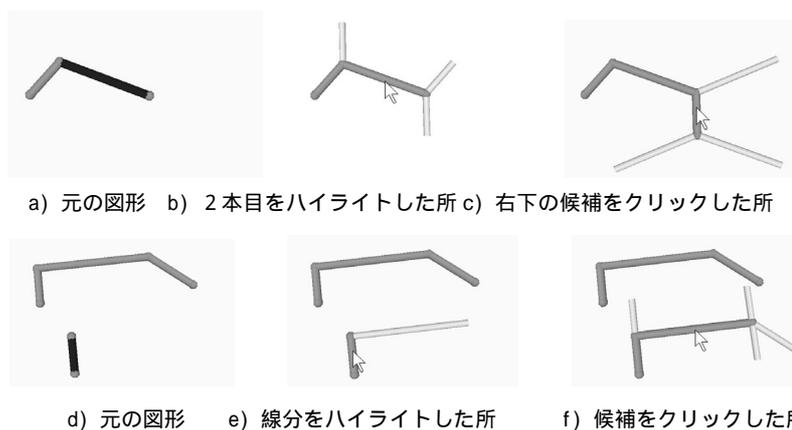


図 2. 線分の描画予測を利用した操作例.

3.3 縮小図による候補提示

ユーザの線分描画・消去操作、あるいはハイライトおよびその取り消し操作が起こると、システムは現在のハイライトの構成に応じて次の操作内容を推定し、縮小図として画面下部に並べて提示する。ユーザは気に入ったものがあればそれをクリックすることでメイン画面中の 3 次元図形へ反映させることができる。気に入ったものがなければ無視して次の操作に進むことができる。

図 3 に操作例を示す。まず、ユーザが画面上に 2 本線を引くと、それに応じて図 3a のような 3 候補が提示される。ユーザが一番左側の縮小図をクリックすると、それによって図 3b のような半透明の描画面が表示される。描画面は、既存の図形要素上

への描画だけでは実現できない、空中への線分描画に利用される[3]。ユーザがこの描画面上に線分を引くと、結果的に3本の線がハイライトされることになり、それに基づいて図3cのような3候補が提示される。ユーザが直方体を選んでクリックすると、図3dのような結果が得られる。ここで地面をクリックすることで、すべての線分のハイライトが取り消され、かつ描画面も消滅する(図3e)。最後に、直方体の上に線分を引くとそれに応じて、図3fのように切り落としを含む候補が提示される。

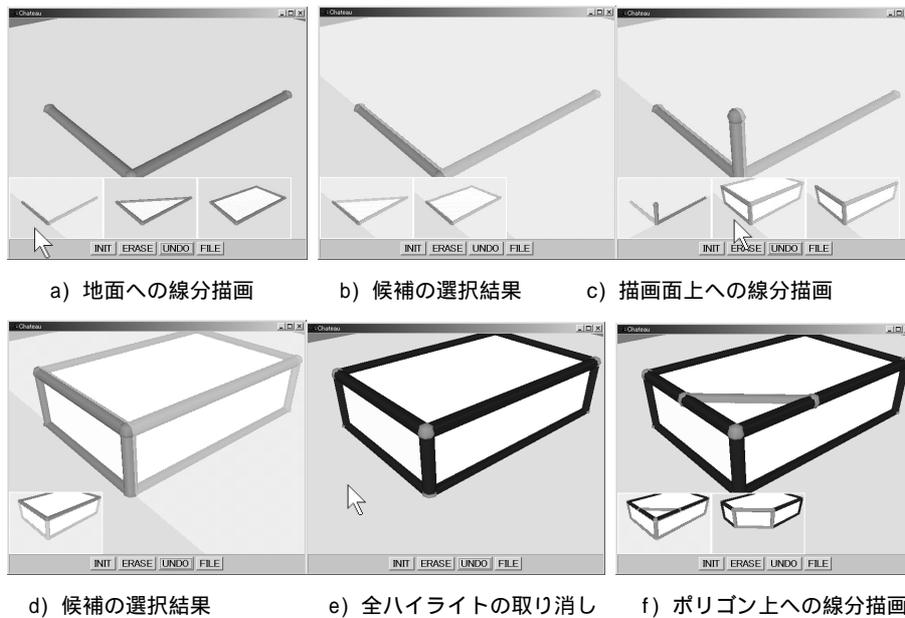


図3. 縮小図による候補生成を利用した操作例

各候補の生成は、対応する候補生成エンジンによって生成される。システムには複数の候補生成エンジンが実装されており、各エンジンはそれぞれ、現在のハイライトされた線分の構成と固有の入力パターンを照合し、適合した時に処理を実行して更新された3次元図形を縮小図にしてシステムに返す(図4)。各エンジンの動作はChimeraシステム[13]における図形的な探索と置換操作に相当するものといえるが、全体を走査するのではなくハイライトされた部分だけに注目する点で異なっている。

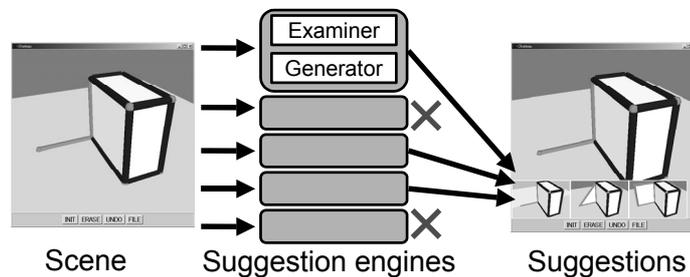


図4. 候補生成エンジンの動作。入力パターンに適合した場合に結果を返す。

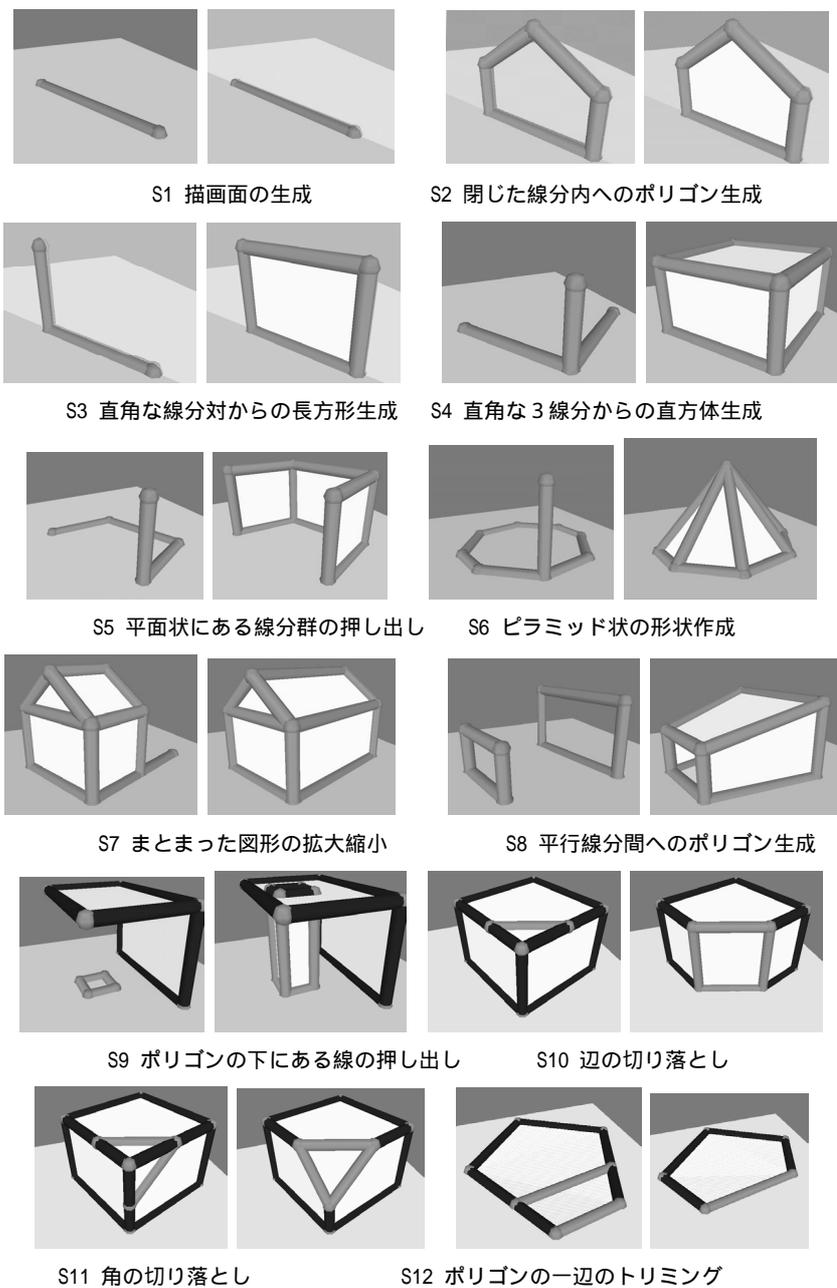


図5. 候補生成エンジンの一覧
左側が入力パターン、右側が処理の適用結果

A Suggestive Interface for 3D Drawing

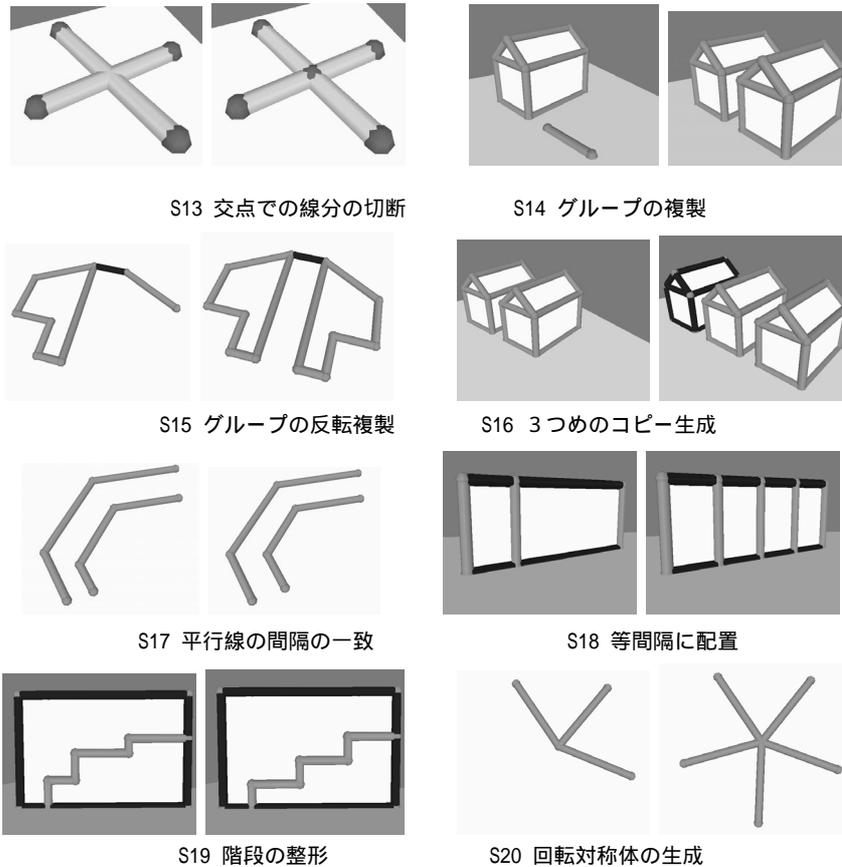


図 5. 候補生成エンジンの一覧(続)

図 5 に、現在実装されている候補生成エンジンの一覧を示す。最初の二つがもっとも基本的な機能を提供している。S1 は最後にハイライトされた線分を元に描画面を生成する。もし線分が既存のポリゴンや地面または描画面の上にあった場合には、当該線分を通りかつその面に垂直な面が描画面となる。線分がどの面上にもない場合には、その線分を通り地面に水平な描画面が設定される。S2 は、単一平面状にありかつ閉じている線分がハイライトされた場合にその内部にポリゴンを生成する。本システムで描画可能な図形はすべて上記の 2 つの基本エンジンのみで生成することができる。残りのエンジンは、手数のかかる操作を短縮する補助的な機能として動作する。

個々のエンジンの動作については、スペースの関係上詳しく述べられないので、基本的に図 5 中で絵として示すにとどめ、代表的なものについてのみここで説明する。S7 は最後にハイライトされた線が、ハイライトされたグループの中の一線分と一端を共有して重なっていた場合に、最後の線に沿うように拡大縮小を行う。S9 は、ある面上にあるまとまった線分がハイライトされた場合に、それらを延長した先に平行なポリゴンがあると、ハイライトされた線分をその面まで引き上げてポリゴンを生

成する。この操作は、テーブルの板を描いた後に足を描く場合等に用いられる。S10 と 11 は、ハイライトされた線分だけでなく、その線分のおかれている状況に基づいて切断操作を行う。S14 は、最後にハイライトされた線分が、他のハイライトされたグループの中の一線分と合同な場合に、最後にハイライトされた線分の周りにそのグループをコピーしてくる。S15 は、S14 とほぼ同様であるが、対応する線分同士が鏡像関係にある場合に動作する。S17 は、いくつかの平行線の対がハイライトされた場合に、それらの間隔が一定になるように整形を行う。S18 は、互いに合同な図形が一直線上に3つ並んでいた場合に、両端の図形の間を等間隔に埋めるようにコピーを複数生成する。中央の図形は、何等分するかを決定するヒントとして使われ、出力時には除去される。

これらのエンジンは、試験的に適当と思われるものを実装しただけであるが、実際のシステムでは、目的に応じて適切なエンジンのセットを開発して利用することを想定している。例えば、和風の建築物を描くときのエンジンのセットや水道管のモデルを作るときのセットなどが考えられる。

4 実装

プロトタイプシステム(Chateau)は、Microsoft J++および Direct3D を利用して実装されている。個々の候補生成エンジン(Java class)は、共通の interface を備え、3次元シーンの変更に応じて呼び出される listener 群として実装されている。個々のエンジンの実装は、共通の3次元図形操作 API を利用することで比較的簡単に実装することができる。典型的なエンジンの場合、ソースコードの長さは100-200行程度である。

5 ユーザテスト

プロトタイプシステムを利用して簡単な試用テストを行った。図6に、テストユーザによるモデリング例を示す。ユーザは、約30分程度チュートリアルの実行とテスト操作に費やし、その後1時間から2時間程度で、図のような3次元形状を生成した。テストユーザからのコメントでは、線分のハイライトと候補選択による操作自体は明白でわかりやすいものの、移動や拡大縮小といった操作は、従来のコマンドとドラッグ操作による方法があったほうがよいとのフィードバックを得た。

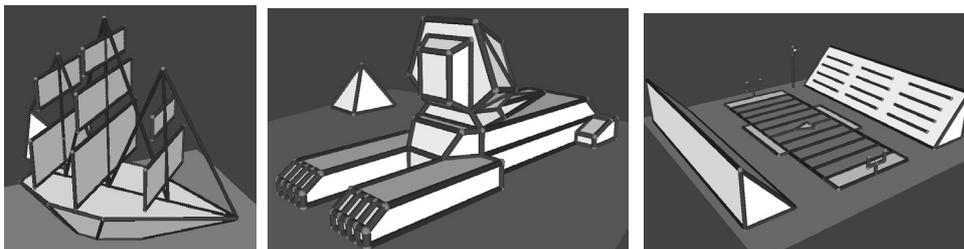


図6. テストユーザによる描画例

なお、現在の実装では、個々のエンジンが結果を返す際に、3次元形状全体を複製して編集するために、図形が大きくなると極端に動作が遅くなっている。速度向上のため、変更された差分だけを返すようにする必要がある。

6 議論

本手法の欠点として、メニューやボタンがないために、どのような操作が可能なのか一覧できないという点が挙げられる。これはジェスチャーに基づくインタフェースに共通する問題であり、とくに操作に慣れない間はマニュアルの形で図5のような一覧表を手元に置いておく必要がある。また、図形の位置や大きさの調節といった連続的なパラメータの入力ができないのも問題点のひとつである。この点に関しては、3DWidget[4]のような従来型のインタフェースと組み合わせて利用することが必要である。

スケーラビリティの問題については、図形そのものの規模に関するものと、エンジンの数に関するものが挙げられる。図形が大規模になったときには、ハイライトによって探索対象を限定することによってかなり対応できると考えられるが、効率的操作のためには、グループ化やロックをかけるといった機構を導入する必要がある。一方エンジンの数が増えた場合に大量の候補が生成されてしまうという問題については、現在のままでは対処できないので、自動的に優先順位をつけて提示したり、必要に応じて自由にエンジンのオン・オフを切り替えできるようにしたりする必要がある。なお、現在の実装では候補の提示順は各エンジンが実装されている順番に依存している。

今後の予定としては、自由曲面からなる3次元モデルの生成への応用[11]、例示プログラミング等[6]を利用したエンドユーザによるエンジンの作成、プレゼンテーション作成ツールなど他のアプリケーションへの応用などを考えている。

謝辞

本研究を進めるにあたって貴重なコメントをいただいたブラウン大学コンピュータグラフィクスグループの諸氏に感謝する。また、本システムの実装にあたって、カーネギーメロン大学 Stage3 研究グループによる3次元ライブラリ(Jalice)を利用させていただいた。

参考文献

- [1] Ashlar Vellum Products, Ashlar Inc., <http://www.ashlar.com/>
- [2] E.A. Bier and M.C. Stone. Snap Dragging. *Computer Graphics*, Vol. 20, No. 4, pp. 233-240, 1986.
- [3] J.M. Cohen, L. Markosian, R.C. Zeleznik, J.F. Hughes, and R. Barzel. An Interface for Sketching 3D Curves. 1999 Symposium on Interactive 3D Graphics, pp. 17-21, 1999.
- [4] D.B. Conner, S.S. Snibbe, K.P. Herndon, D.C. Robbins, R.C. Zeleznik, and A. van Dam. Three-Dimensional Widgets. 1992 Symposium on Interactive 3D Graphics, pp. 183-188, 1992.

- [5] A. Cypher, Eager: Programming Repetitive Tasks by Example. Proceedings of CHI'91, pp.33-39, 1991.
- [6] A. Cypher. Watch What I Do: Programming by Demonstration. Cambridge, MA: MIT Press. 1993.
- [7] M. Gleicher and A. Witkin. Drawing with Constraints. The Visual Computer, Vol. 11, No. 1, pp. 39-51, 1994.
- [8] S. Hudson and C. Hsi. A Synergistic Approach to Specifying Simple Number Independent Layouts by Example, Proceedings of INTERCHI'93, pp. 285-292, 1993.
- [9] T. Igarashi, S. Matsuoka, S. Kawachiya, and H. Tanaka. Interactive Beautification: A Technique for Rapid Geometric Design. Proceedings of UIST'97, pp. 105-114, 1997.
- [10] T. Igarashi, S. Matsuoka, S. Kawachiya, and H. Tanaka. Pegasus: A Drawing System for Rapid Geometric Design. CHI'98 Summary, pp. 24-25, 1998.
- [11] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. SIGGRAPH 99 Conference Proceedings, pp. 409-416, 1999.
- [12] S. Karsenty, J.A. Landay, and C. Weikart. Inferring Graphical Constraints with Rockit, Proceedings of HCI'92, 1992.
- [13] D. Kurlander and S. Feiner. Interactive Constraint-Based Search and Replace. Proceedings of CHI'92, pp. 609-618, 1992.
- [14] J. Mankoff, S.E. Hudson and G.D. Abowd. Interaction Techniques for Ambiguity Resolution in Recognition-based Interfaces. Proceedings of UIST '00, pp. 11-20, 2000.
- [15] J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. SIGGRAPH 97 Conference Proceedings, pp. 389-400, 1997.
- [16] T. Masui. An Efficient Text Input Method for Pen-based Computers. Proceedings of CHI'98, pp. 328-335, 1998.
- [17] D. Maulsby, I.H. Witten and K.A. Kittlitz. Metamouse: Specifying Graphical Procedures by Example. Proceedings of SIGGRAPH'89, pp. 127-136, 1989.
- [18] K. Sims. Artificial Evolution for Computer Graphics. SIGGRAPH 91 Conference Proceedings, pp. 319-328, 1991.
- [19] R.C. Zeleznik and A. Forsberg. UniCam - 2D Gestural Camera Controls for 3D Environments. Proceedings of 1999 Symposium on Interactive 3D Graphics, 1999.
- [20] R.C. Zeleznik, K.P. Herndon, and J.F. Hughes. SKETCH: An Interface for Sketching 3D Scenes. SIGGRAPH 96 Conference Proceedings, pp. 163-170, 1996.