

三次元情報入力における
ユーザーインターフェースについて
～タートル・グラフィクス手法の応用

辻堂仁規 江村恵里子

三菱電機株式会社 応用機器研究所

CAD/CAMの分野では、三次元情報を計算機上で効率良く扱うことが大きな課題の一つとなっているが、特にCAMにおいてはNC工作機械等による三次元曲面の自動切削を目的とした、使い易く、実用的な自動プログラミング・システムが望まれている。しかし現状では三次元情報が直接的に入力できない等いくつかの問題がある。私達はタートル・グラフィクスの手法を用い、三次元情報を簡単にかつ直接的に入力できることを目的とした三次元曲面入力システムを開発した。本報告では三次元曲面入力におけるタートル・グラフィクス手法のいくつかの利点を示し、さらにユーザーインターフェースを改善するためのいくつかの手法について述べた。

User Interface for
Entering the Three Dimensional Figures
with Turtle Graphics

Yoshinori TSUJIDO and Eriko EMURA

MITSUBISHI ELECTRIC CORP. Product Development Laboratory
1-1, Tsukaguchi-honmachi 8-chome, Amagasaki, Hyogo, 661 Japan

Information processing for three dimensional figures is one of the important problem in CAD/CAM field. Especially in CAM field, a practical and user-friendly automated programming tool for NC machining, expected to be developed. But now, conventional systems have some problems in data input of three dimensional figures. So we developed the new system for entering the three dimensional sculptured surfaces which utilize the Turtle Graphics method. In this paper, we show some merits of Turtle Graphics for this kind of input process, and some methods to improve user interface.

1. はじめに

CAD/CAMの分野では、三次元情報を計算機上で効率良く扱うことが大きな課題の一つとなっているが、最近ではその効率のネックとしてユーザーインターフェースが注目されるようになってきた。特にCAMにおいては対話型CNCが一般化してくるなどユーザーインターフェースの改善の必要性が明らかになっている。三次元曲面の自動切削を対象とした自動プログラミング・システム（以下自動プロシステムと略す）においても実用的なCAMツールとして、その実用性と共に使い勝手の改善が望まれているが、CAMツールという性格上Manufacturingの部分に比重が偏るせいか多分に専門的な言語が用いられているのが現状であり、そのユーザーインターフェースは決してよいとはいえない。さらに大きな問題点として現在使用されているこの種のシステムでは、そのほとんどが、三次元の直接的入力が行えないというユーザーインターフェースの制約から二次元平面上の曲線とこの組合せに基づいて生成される三次元曲面を扱っており、本来の曲面理論を持っているその自由度の高さを生かし切れていない点が挙げられる。その背景には従来の設計方式が二次元の「図面」の概念に強固に結び付いているという現実があり、さらには最近利用可能になってきた三次元ディスプレイを有効に利用したものがなく、従来の静的な二次元表現を主に用いることを前提としていたことなどから「人間にとっては二次元上で考えるのが自然であり、三次元を直接扱うのは難しい」と考えられていたように思われる。

私達はこういった観点から、これを改善し、できるだけ単純な入力により、できるだけ多くのことが記述できるようにすることにより、三次元情報を簡単にかつ直接的に入力できることを目的とした三次元曲面入力システムを開発した。この目的を達成するための一つの方法として私達はタートル・グラフィックスの手法を採用し、これを拡張利用したが、この手法はユーザーインターフェース面で有効な特長をいくつか持っており、使い易いシステムが構築できた。

本報告では私達が行なったタートル・グラフィックス手法の拡張及び曲面創成等について述べ、さらに本システムで試みたいいくつかの新しいプログラミング手法について説明を加えたいと思う。

2. タートル・グラフィックス手法について

タートル・グラフィックス手法は、MITのS.Papertらによって開発された「LOGO言語」のグラフィックス手法とし

て用いられているものである。[1] LOGO言語は子供の「自己発展的学習」を目的として開発されたものであり、あらかじめ利用方法の教育を受けてから使用するのではなく、どのように動くかを実際にやってみながらその利用方法を学んでいく、といった考え方に基いている。一般的にはインタプリタ形式として実現されており、BASIC言語のコマンド・モードのように入力即時実行のモードを持っている。

タートル・グラフィックスはLOGO言語の一部分でしかないが、上で述べたようなLOGO言語の基本思想が反映されており以下に述べるように、本システムの目標に有効ないくつかの特長を持っている。

タートル・グラフィックスの特長としては、

- ① タートルと呼ばれるグラフィック・カーソルを基準とした相対型のグラフィックスであり、人間が直観的に解り易い。
- ② コマンドが単純であり、数が少なく、かつ曲線等の情報がコンパクトに記述できる。
- ③ タートル自身が位置情報と共に姿勢情報を持つために、グラフィックス情報だけでなく、座標系の移動の記述ができる。[2]

等が挙げられるが、これらの特長はこの手法がロボット言語とも言われるように、基本的にタートルというObjectに命令的に「動き」を指示するための言語であることに拠っているものと思われる。

3. タートル・グラフィックスの拡張

タートル・グラフィックスの手法は非常にシンプルであり簡単にプログラムができるが、私達はこれをCAMのツールとして用い、三次元情報を直接的に入力できるようにするために、いくつかの拡張を行なった。

3.1 円弧プリミティブの追加

NC工作機械等による自動切削を目的とした自動プロシステムにおいては、円弧の記述機能が不可欠であるが、LOGOにおけるタートル・グラフィックスでは直接的に円弧を記述するコマンドを持っておらず、この機能を追加する必要があった。本システムにおいては、図3-1に示したように円弧の半径と回転角をパラメータに持つ2つの円弧コマンド 'LCURVE', 'RCURVE' を追加した。この拡張コマンドを用いれば図3-2に示したような滑らかに接続した曲線を簡潔に記述できる。

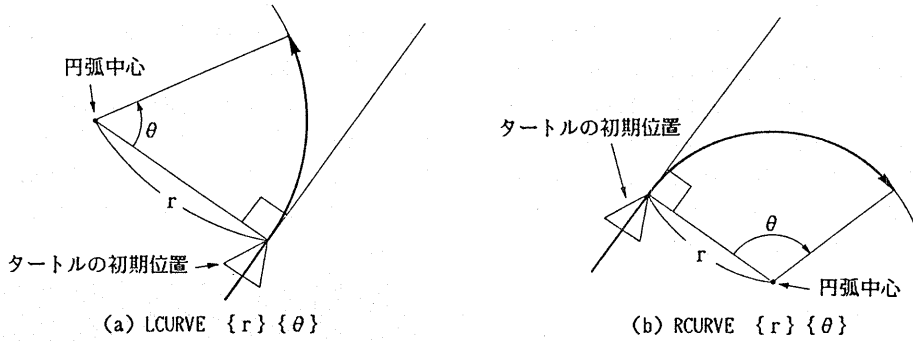


図3-1 円弧プリミティブ

3.2 三次元拡張のためのプリミティブ

三次元への拡張方法はいくつか考えられるが、本システムでは、タートルを三次元的なものと解釈し、その姿勢を示す3つの自由度を、飛行機やロボットの分野で用いられる 'ROLL', 'PITCH', 'YAW' によって規定し、現在のタートルからの姿勢の変動をこれらの3つのパラメータによって記述する方法を用いた。この3つのパラメータの内、'YAW' はそのまま既にある、'LEFT', 'RIGHT' に対応するので、拡張するのは、'ROLL' と 'PITCH' のみでよい。図3-3にこれらのパラメータのそれぞれの意味を示す。この拡張によって図3-4に示したような一平面上にない三次元空間曲線が直接的に簡単に記述できるようになる。

ここに示したような2つのコマンドのみの追加によって三次元への拡張が行なえ、そしてこれらのコマンドの有無によって二次元と三次元の切り分けができる点もタートル・グラフィクス手法の一つの大きな特長と言える。

3.3 仮想タートル概念の導入による座標系移動記述

従来の自動プロシシステム等で用いられてきた点列表記がその位置を表す3自由度の情報しか持たないのに対し、タートルはその位置と姿勢を表す6自由度の情報を持っている。そのためにタートル・グラフィクス記述は空間曲線等のグラフィクス情報だけでなく、座標系の移動をも表現できる。しかし実際のタートルの移動だけでは制約が大きく、限られた移動しかできないので私達は仮想タートルの概念を導入した。[2]そして PENDOWN、PENUP コマンドに対応する ON、OFF コマンドによって実タートルとの接続を制御できるようにし、任意の座標系の移動を記述できるようにした。

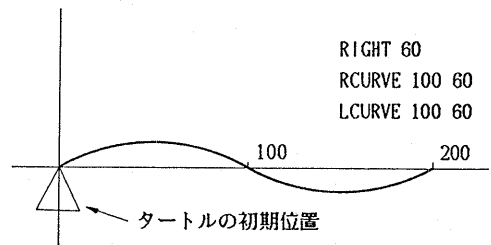


図3-2 円弧プリミティブの使用例

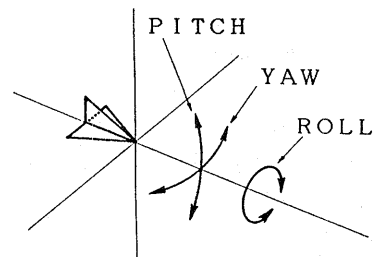


図3-3 三次元拡張プリミティブ

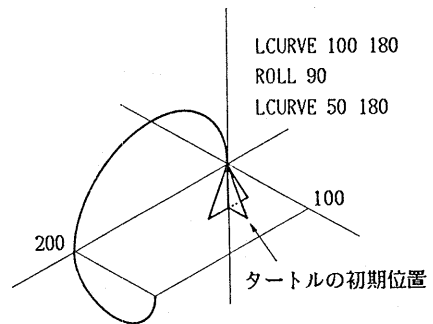


図3-4 三次元拡張プリミティブの使用例

3.4 重ね合わせ機能の追加

以上に述べてきたいくつかの拡張によって、自由度の高い三次元空間曲線や座標系移動の記述が可能となるが、単独のタイトルのみではその記述できる範囲が直線と円弧あるいは並進と回転のシーケンシャルな組合せに限られてしまう。そこで'SYNC'コマンドを追加することによって複数のタイトル・グラフィクス記述を同期させ、重ね合わせ(かけ合わせ)のことにができるようにする。例えば、

```
SYNC [  
  FORWARD 100 ;  
  ROLL 360  
]
```

という記述は100並進する間に360°回転するヘリカルな動きを表しているが、このように、この拡張によってより複雑な動きを記述できるようになる。

4. 曲面創成について

4.1 曲面創成の方法

以上に述べてきたように拡張タイトル・グラフィクス手法を用いれば、

① 三次元空間曲線 (Curved Line ; 以下CLと略す) と

② 座標系移動 (Coordinate Motion ; 以下CMと略す)

が記述できるが、曲面創成はこれらを組合せ従来のスイープ手法と同様な考え方によって行なう。即ち図4-1に示したようにCLがCMによって移動を受ける時に創成する曲面を創成曲面とする。図4-2~4-4に創成曲面の例を示す。図4-2と図4-3は同じCLによって創成された曲面であるが、一般的な曲面創成システムではこういった曲面の種類によって個々に平行移動、回転といった曲面のタイプを用

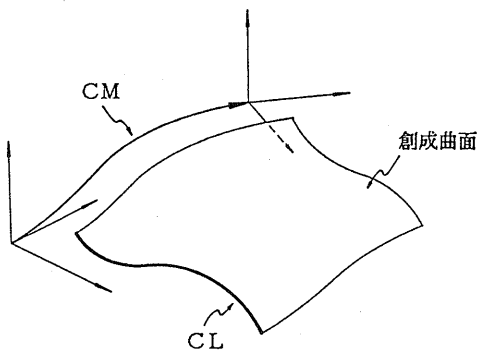


図4-1 曲面の創成

意し、これを使い分ける必要があったが、この手法では種々の座標系の移動が直接記述できるために曲面毎のタイプを特別に用意する必要はない。

線織面 (Ruled-surface) 等の複数のCLによって創成される曲面においては複数のCL間の補間が必要になってくるが、この手法によればCL空間とCM空間とを切り離して考えることが可能となるなど処理系の実現上からも好ましい面も持っている。

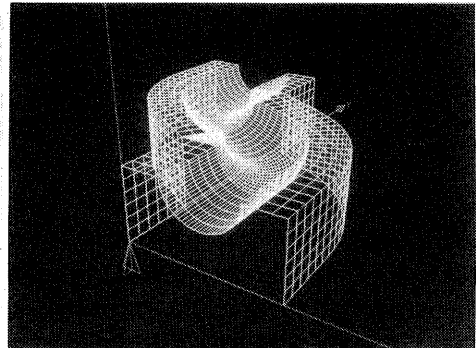


図4-2 創成曲面の例(1)

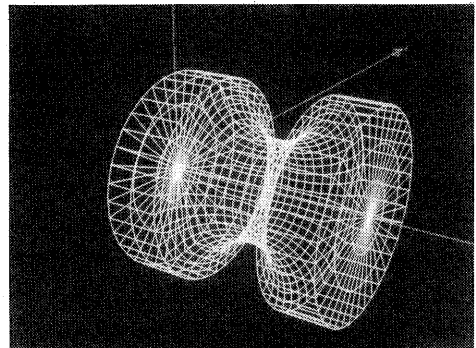


図4-3 創成曲面の例(2)

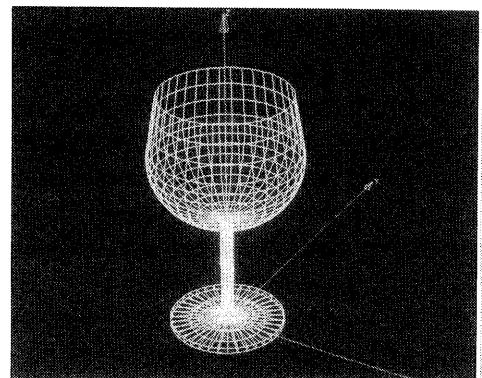


図4-4 創成曲面の例(3)

4.2 曲面記述能力について

上述したように従来の一般的なシステムでは曲面のタイプを指定する必要があったが、タートル・グラフィクス手法によればその記述の中にその意味を埋め込むことができる。例えば、図4-5(a)に示したような一見すると同じCL1、CL2、CMの組合せに対して、同図(b)、(c)、(d)のような異なった曲面を生成することができる。以下にそれぞれのプログラムを記す。

```
(b) CL1:PITCH 90
      RCURVE 100 180;
      CL2:RCURVE 100 180;
      CM :FORWARD 320;
```

```
(c) CL1:PITCH 90
      RCURVE 100 180;
      CL2:(CL1に同じ又は指定しない)
      CM :FORWARD 320
          PITCH 90;
```

```
(d) CL1:PITCH 90
      RCURVE 100 180;
      CL2:(CL1に同じ又は指定しない)
      CM :SYNC [
          FORWARD 100;
          PITCH 90
      ];
```

(b) は一般的な線織面であり、(c) は並進と回転の組合せ、(d) は重ね合わせの機能を用いたものである。

このように曲面のタイプ毎の特別な創成機能を用意しなくてもここで示したような多様な曲面が得られる点が大きな特長である。

5. プログラミング環境

本システムはモノクロのビットマップ・ディスプレイを持ったワークステーションと、これにDMA結合された三次元グラフィクス・ディスプレイによって構成されており、この上でいくつかの新しい考え方に基づいたプログラミング手法を実現したのでここではこれについて述べる。

5.1 入力のマウスへの集中化

一般的なCAD/CAMシステムの入力はテキストやデータを入力するためのキーボードと、対話的に位置情報等を入力するためのタブレット等のポインティング・デバイ

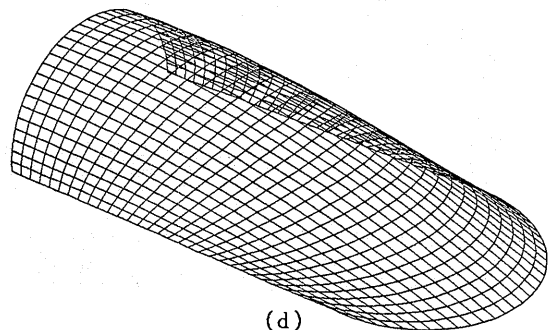
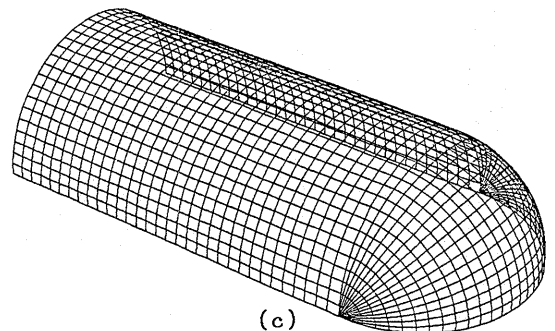
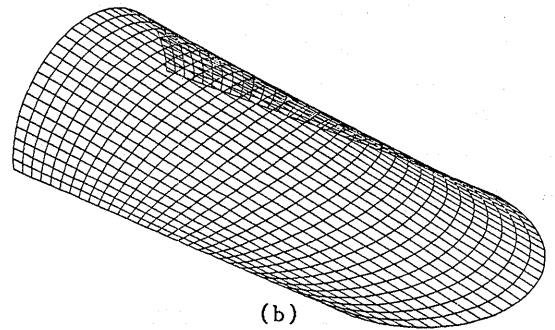
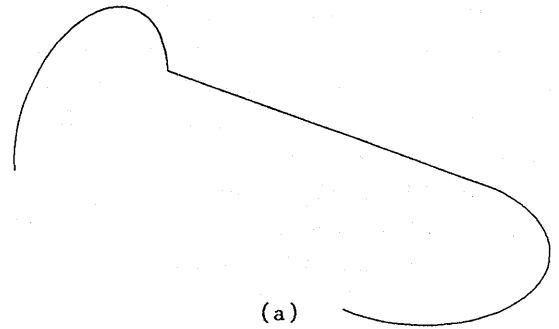


図4-5 曲面の記述能力

スを併用しているものが多い。そのためにテキスト（あるいはデータ）入力とポインティング入力が混在し、その切り換えが頻繁に起こる場合には手の移動が多くなり、入力の効率が悪かった。これを解決するために、足マウス[3]、頭の動きによるポインティング等の手法がいくつか研究されているが、まだ実用には至っていない。

私達はこれを解決するために

- メニューによるコマンドの選択的入力
- アイコンによるデータの選択
- 数値メジャーによるパラメータの入力

等の手法によりマウスで全ての入力を行なえるようにした。

私達は視点変換の機能も一つのマウスボタンに割付け、いつでも対話的な視点変換が行なえるようにしたが、三次元情報の直接的な入力をサポートするためにはこのような操作が違和感なくかつ迅速に行なえることが一つの重要なポイントであり、デバイスを持ち換えることなくこれらの操作が行なえる点は都合が良い。

5.2 コマンド・モードのみのプログラム入力

オペレータにとって自分の入力したプログラムの正当性が容易に確認できれば非常に助かるが、私達はこれをサポートする方法として、コマンド・モード（入力即時実行モード）のみのプログラム入力手法を用いた。以下にこれについて説明する。

プログラミングはワーク・ステーションのビットマップ・ディスプレイに表示されたプログラミング画面（図5-1）上で行なう。

① まずカーソルを自分の好きな空いた領域に設定する。カーソルの移動は初期化も兼ねており、これによってタートルがHOME位置に戻る。

② ポップアップ・メニューによるコマンドの選択と、数値メジャーを用いたパラメータの入力によってタートル・グラフィクスのプログラムを作成してゆく。この時、各入力が行われる毎に即時実行され三次元グラフィクス・ディスプレイ上のタートルにその動きが反映される。オペレータは入力毎にそのタートルの動きを確認しながらプログラミング作業を続けることができる。

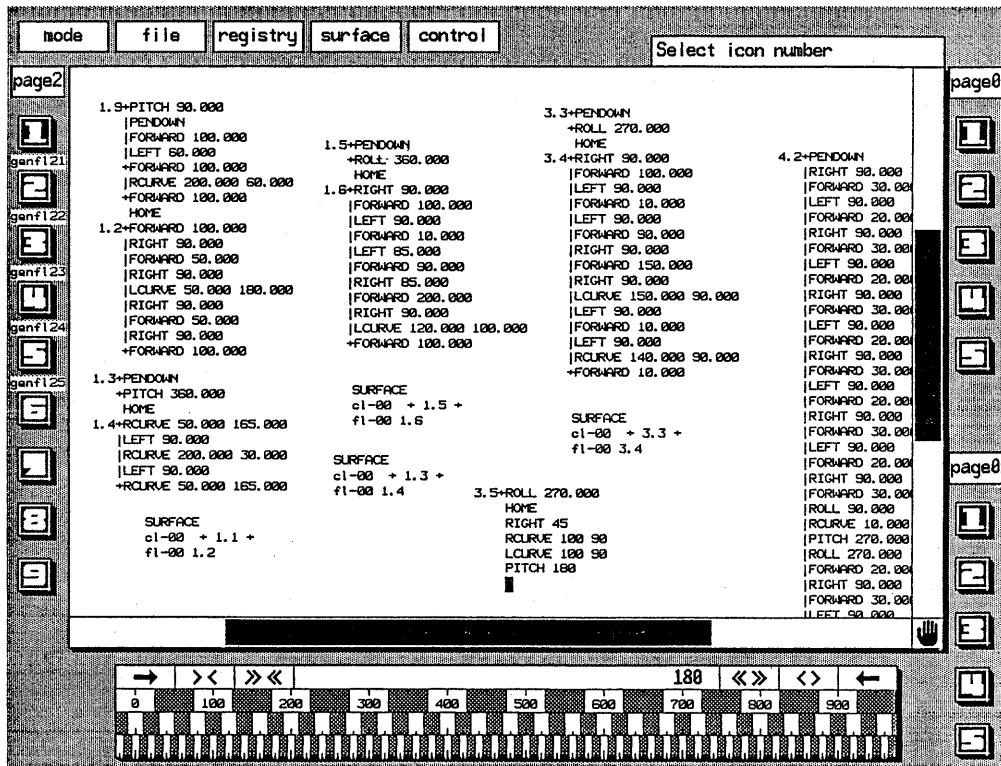


図5-1 プログラミング画面

このようにして入力されたプログラムは、次のカーソル移動が起こるまではその表示位置と関連付けられた一かたまりのプログラムとして記憶されている。

①、②を繰り返すことにより複数のプログラムを画面上の任意の位置に作成することができるが、これらのプログラムは、いつでも開始行と終了行を指定してその一部分を登録あるいは参照、引用などができるようになっている。

この手法によれば、オペレータはプログラミング作業を意識せずに紙の上にメモ書きをする感覚で、しかもその入力を確認しながら入力作業を続けることができる。

6. まとめ

本報告では三次元曲面入力におけるタートル・グラフィクス手法のいくつかの利点を示し、さらにユーザーインターフェースの良いシステムを実現するためのいくつかの手法について述べた。三次元情報の直接的な入力にはある程度の慣れが必要なのは事実であるが、タートル・グラフィクス手法の簡便性、グラフィクスによる即時フィードバックなどの効果により、簡単なマウスの操作でかなり複雑な曲面が数分から十数分で生成できるようになり、そのユーザーインターフェースの改善の効果が実証できたと思う。

ユーザーインターフェースに関連したソフトウェアはその対話性の向上のために複雑なものになる傾向があるが、今後はこういった部分へのAI的手法の利用等も考えてみたい。

<参考文献>

- [1] 祐安重夫、Logo、ラジオ技術社、(1984)
- [2] 辻堂他、タートル・グラフィクス手法の三次元曲面入力への応用、情処第35回全国大会、(1987 予定)
- [3] 武藤他、足マウスの指示動作特性と文書入力作業におけるキー操作との協調性について、第1回ヒューマン・インターフェース・ソフトウェア論文集、(1985)
- [4] H.Abelson & A.A.diSessa, Turtle Geometry, The MIT Press, (1980)

討 論

3. 三次元情報入力におけるユーザインタフェースについて

辻堂（三菱）

川合：仮想タートルの仕組みがよく分からなかったのですが。

辻堂：OFFコマンドを行うと仮想タートルが取り残されて実タートルだけが動き、ONコマンドで再び仮想タートルも動くようになります。このとき実タートルと仮想タートルの相対的位置が保存されて動きます。

福井：実タートルと仮想タートルは名前が逆のような気がします。

辻堂：実際に見えるタートルが実タートルで、仮想タートルは見えません。

福井、川合：仮想タートルはどういうときに必要なのですか。

辻堂：空間曲線を創成する場合に、タートルが一方向だけにしか動けないと制約が大きいので、座標系を動かすために仮想タートルを用います。

鎌田：最近のLOGOにはマルチタートルの機能があって、複数のタートルを同時に動かすことができます。複数のタートルをゴムで結んで動かしてその軌跡を利用するという方法はありますか。

辻堂：それも良いかもしれませんね。

ただ、あまり多くの機能を入れると分かりにくくなると思われれます。

川合：座標系の移動と描画を仮想タートルと実タートルで区別するのは分かりにくくはありませんか。座標系の変換は別に設定した方が分かりやすいかもしれませんね。

原田：機能をコンパクトにするのは良いですが、使いやすさがユーザインタフェースでは大切だと思います。

辻堂：この方法は使いやすいかどうか分かりませんが、結果がすぐに画面にフィードバックされるのでその点では良いと思います。

坂下：これをはじめて使う人は、どの程度で自分の思いどおりの形状を作れるのでしょうか。

辻堂：使いやすさの評価は今後の課題です。

それほど知らない人でもわりあい簡単に作ることができます。

川越：線をタートルと見なしてはどうですか。

辻堂：CL（三次元空間曲線）がいわば線タートルに対応しています。

川越：CLはもとのタートルという概念を使っているから、仮想タートルが必要になって分かりにくいのではないのでしょうか。

辻堂：線タートルといっても・・・

川合：それは亀（タートル）ではなく蛇ですね。（笑い）