

マウス+左ボタンによる 3D/6DOF マニピュレーション

野中 秀俊[†]

本報告では、2D ポインティングデバイスによる 3D ユーザインタフェースを提案する。3 自由度の回転、2 自由度の平面の指定、2 自由度の平面上の移動という三つのフェーズによって、3D ダイレクトマニピュレーションが実現している。本方式は、マウス以外の 2D ポインティングデバイスや、携帯型マルチタッチタブレット PC、ペン入力携帯型ゲーム機などにも適用できる。

3D/6DOF Manipulation Using Mouse with Single Button

HIDETOSHI NONAKA[†]

In this paper, a 3D user interface using 2D pointing device is presented. 3D direct manipulation has been realized by using three phases: rotation, plane alignment, and moving along the plane. This method is applicable to most 2D pointing devices such as mouse, pen-tablet, pointing stick, trackball, touchpad, touch screen, multi-touch mobile tablet computer, hand-held game system with stylus, and so on.

1. はじめに

マウスなどの 2D ポインティングデバイスは、通常 2 自由度 (2DOF) 以下の操作に使用されている。建造物など、鉛直方向や水平面が固定されている仮想 3D 物体の回転にはマウスの x 軸方向移動を水平面上の回転や方位角に、 y 軸方向移動を仰角や伏角に対応させることにより、2DOF の自然な回転が得られる。

これに対し、3DOF の回転を実現するために、様々な方式が提案されている。[Hanson, 1992] は、マウスの x 軸方向移動を y 軸回りの正回転に、 y 軸方向移動を x 軸回りの負回転に対応させて回転軸を指定し、自由回転を逐次的に行う方式を提案している (Rolling Ball)。これにより z 軸回りの仮想 3D 物体の正回転はマウスの負回転によって実現される 1)。[Chen, 1988] らは、スライダ、メニュー、ボタンを使用した 3DOF 回転に加えて、仮想的なトラックボールを使用した 3DOF 回転を提案し、その操作性を評価している 2)。[Shoemaker, 1998] は、幾何学的な考察のもとに、仮想的な球面における弧と仮想 3D 物体の回転の対応を定式化し (Virtual Sphere)、ヒステリシスを伴わないマウス操作を実現している 3)。[Henriksen, 2004] らは、仮想的なトラックボール (Virtual Trackball) による回転操作方式を整理し、誤差の改善を行っている 4)。さらに、マウスホイールなどの付加的な入力を使用する方式や、モードを切り換えながら操作する方式、表示

ウィンドウと異なるウィンドウの操作による方式などが提案されている 5)–13)。また 3D/6DOF に特化した入力デバイスも数多く提案されている 14)–16)。

上に挙げた従来手法は、一定の評価を得て様々なアプリケーションで採用され普及しているが、いずれも以下の条件の一つ以上を満たしていない。

1. *Generality*. 現在普及している 2D ポインティング装置には、マウスだけでなく、トラックボール、タッチパッド、ペンタブレット、トラックポイント、タッチスクリーンなどがあり、それらを代用あるいは併用しているユーザが多いと思われる。これらのマウス以外の 2D ポインティングデバイスでも自然に操作できることが望ましい。
2. *Direct Manipulation*. 単一のデバイスによる仮想 3D 物体の選択・回転・移動が可能で、物体の特定の箇所をつかんで 3DOF の回転、3DOF の移動をさせる使用感が得られることが望ましい。例えば仮想球面や仮想トラックボールによる 3DOF 回転では、穴の開いた多面体の内側奥をつかんで逆回転させるような操作はできない。
3. *Modeless Interface*. 操作にモードがある場合、ユーザに対して、現在どのモードで操作しているかを短期記憶に保持するといった、認知的負荷を課すことは望ましくない。仮に操作にモードがあっても、タスクが終了した後に通常モードに復帰することが望ましい。例えば、マウスの左ボタン押下によるドラッグの場合、ドラッグが終了した後に、ドラッグモードからカーソル移動モードに復帰するため、ユーザへの負担は少ないと考えられる。

[†] 北海道大学

Hokkaido University

本報告では、上述の三つの条件を満たすような 3D/6DOF インタフェースを提案する。

2. 仮想 3D 物体の 3DOF 回転

2.1 回転軸および回転角の算出方法

仮想 3D 物体が適切にモデリングされ、表示されていることを仮定し、さらにユーザがその仮想 3D 物体の奥行きや立体感を適切に知覚できていることを仮定すれば、ユーザがスクリーンに表示されている物体上の 1 点をポイントしたときに、スクリーン上の 2 次元座標から、仮想 3D 空間の 3 次元座標を計算することができる。本報告では、式の導出過程は省略し、結果のみを示す。

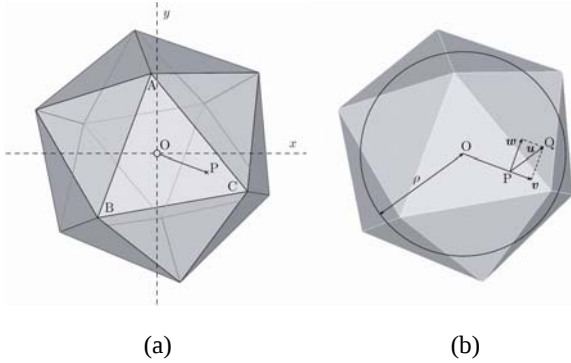


図1 仮想 3D 物体上の(a)ポインティングと(b)ドラッグ

以下、図 1(a) の例を用いて説明する。簡単のため、スクリーンの原点、仮想 3D 空間の原点、および回転の中心 O が同一であるものとし、右を x 軸、上を y 軸、手前を z 軸とする。ユーザが点 P をポイントしたとき、スクリーン上のマウスカーソルの座標を $\mathbf{p} = \overrightarrow{OP} = (p_x, p_y, 0)^t$ 、透視変換の視点位置の座標を $\mathbf{h} = (0, 0, h)^t$ 、点 P を含むポリゴン $\triangle ABC$ の法線ベクトルを $\mathbf{n} = (n_x, n_y, n_z)^t$ 、そのポリゴン内の任意の一点、例えば重心を $\mathbf{c} = (c_x, c_y, c_z)^t$ とすると、仮想 3D 空間における対応点 $\tilde{P}$ の座標 $\tilde{\mathbf{p}} = \overrightarrow{O\tilde{P}} = (\tilde{p}_x, \tilde{p}_y, \tilde{p}_z)^t$ は、式(1)で求められる。

$$\tilde{\mathbf{p}} = \frac{\mathbf{n}^t(\mathbf{c} - \mathbf{h})\mathbf{p} + \mathbf{n}^t(\mathbf{c} - \mathbf{p})\mathbf{h}}{\mathbf{n}^t(\mathbf{p} - \mathbf{h})} \quad (1)$$

また回転の中心 O と点 P との仮想 3D 空間における距離 ρ は、式(2)で与えられる。

$$\rho = \frac{1}{\mathbf{n}^t(\mathbf{p} - \mathbf{h})} \sqrt{(\mathbf{n}^t(\mathbf{c} - \mathbf{h}))^2 \mathbf{p}^t \mathbf{p} + (\mathbf{n}^t(\mathbf{c} - \mathbf{p}))^2 \mathbf{h}^t \mathbf{h}} \quad (2)$$

次に図 1(b) においてユーザが点 P から点 Q にドラッグしたものとする。

$\mathbf{q} = \overrightarrow{OQ} = (q_x, q_y, 0)^t$ 、 $\mathbf{u} = \overrightarrow{PQ} = (u_x, u_y, 0)^t$ とし、ベク

トル $\mathbf{u}$ を図に示すように $\mathbf{v} = \mathbf{p}^t \mathbf{p} \mathbf{u} / \mathbf{p}^t \mathbf{p}$ 、および $\mathbf{w} = \mathbf{u} - \mathbf{v}$ に分解する。これらを用いて、回転軸 $\mathbf{r} = (r_x, r_y, r_z)^t$ を以下のように求める。

If $0 \leq \mathbf{p}^t \mathbf{p} < \rho^2$:

$$r_x = -v_y, \quad r_y = \sqrt{1 - \mathbf{p}^t \mathbf{p} / \rho^2} w_y$$

$$r_y = v_x + \sqrt{1 - \mathbf{p}^t \mathbf{p} / \rho^2} w_x, \quad r_z = \frac{1}{\rho} \mathbf{p} \times \mathbf{w}$$

If $\rho^2 \leq \mathbf{p}^t \mathbf{p}$:

$$r_x = -v_y, \quad r_y = v_x, \quad r_z = \frac{1}{\sqrt{\mathbf{p}^t \mathbf{p}}} \mathbf{p} \times \mathbf{w}$$

$$(r_x, r_y, r_z) := (\text{sign}(p_z)r_x, \text{sign}(p_z)r_y, r_z)$$

さらに、回転行列 R を式(3)で表す。

$$\bar{\mathbf{r}} \equiv \mathbf{r} / \sqrt{\mathbf{r}^t \mathbf{r}}, \quad [\bar{\mathbf{r}} \times] \equiv \begin{pmatrix} 0 & -\bar{r}_z & \bar{r}_y \\ \bar{r}_z & 0 & -\bar{r}_x \\ -\bar{r}_y & \bar{r}_x & 0 \end{pmatrix}$$

$$R = \frac{1}{\sqrt{\mathbf{u}^t \mathbf{u} + \rho^2}} \left(\frac{\mathbf{u}^t \mathbf{u} \bar{\mathbf{r}} \bar{\mathbf{r}}^t}{\sqrt{\mathbf{u}^t \mathbf{u} + \rho^2} + \rho} + \sqrt{\mathbf{u}^t \mathbf{u}} [\bar{\mathbf{r}} \times] + \rho I \right) \quad (3)$$

原点と回転の中心が異なる場合は、回転の前後に平行移動を行うことにより、任意の点を中心とした回転が得られる。式(3)による回転には以下の性質がある。

1. マウスカーソルが仮想 3D 物体と重なっている状態では、仮想 3D 物体表面とマウスカーソルの位置関係がほぼ一定で、直接つかんでいる感覚が得られる。
2. マウスカーソルが仮想 3D 物体の像から離れても、仮想 3D 物体はマウスの動きにほぼ比例して回転し続ける。
3. マウスカーソルから原点までのスクリーン上の距離が ρ を超えたところでは、ベクトル $\mathbf{w}$ の成分が全て z 軸回転に対応する。

2.2 3DOF 回転に関するデモ 1

図 2 に回転立体パズルデモのスクリーンショットを示す。マウスカーソルは通常は矢印の形状になっている。マウスカーソルが回転パズルに重なると、右手人差指の形状に変化する。この状態で左ボタンを押下すると 100[20-220] ms 後に左掌の形状に変化する。変化のタイミングはデバイスの種類や個人差に応じて変更できる。それぞれのマウスカーソル形状を図 3 に示す。

マウスカーソルが左掌の状態でドラッグすることにより回転立体パズル全体が 3DOF で回転し、マウスカーソルが右人差指の状態でドラッグすることにより、パーツがドラッグ方向に 1DOF で回転する。以上の 2 通りのフェーズを使い分けることにより、パズルを操作することができる。

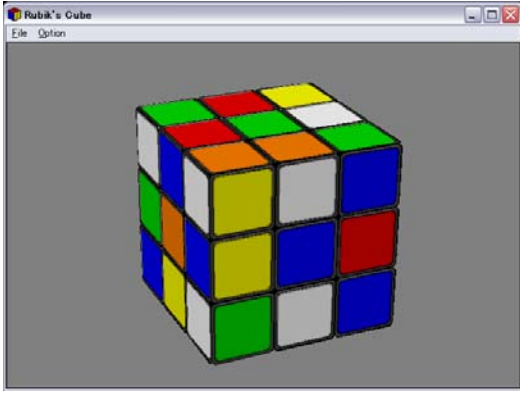


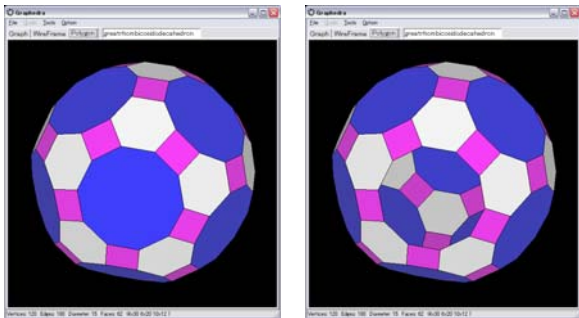
図2 3DOF 回転に関するデモ 1：回転立体パズル



図3 カーソル形状 (矢印, 右人差指, 左掌)

2.3 3DOF 回転に関するデモ 2

図 4 に準正多面体デモのスクリーンショットを示す。本アプリケーションは、様々な多面体、特に正多面体や準正多面体を、インタラクティブに作図できるソフトウェアで、当初は教育目的で開発したものである。例えば、図 4(a)において、カーソル形状が右人差指の状態が一番手前の面をクリックすることにより、その面が図 4(b)に示すように開放される。この状態で、開放部分の奥の多面体の内側の面を、左掌の状態で x 軸正の方向にドラッグすると、 y 軸回り負の方向、つまり逆方向に回転する。



(a)

(b)

図4 3DOF 回転に関するデモ 2：準正多面体

3. 仮想 3D 物体の 3DOF 移動

3.1 3DOF 移動の方法

マウスによる 3DOF 移動の方法としては、カメラコントロールによるシーンの回転と、スクリーン上における仮想 3D 物体とターゲットとの見かけ上の重なりを利用した、いわゆる “ray casting” の方法が一般的であると考えられる [17]。この手法では、シーン全体を頻繁に回転させる必要があり、また仮想 3D 物体とターゲットの間に障害物が存在する場合、直接移動を指定することができない。本報告では、カメラコントロールによるシーンの回転を行わなくても、奥行感や仮想 3D 物体とターゲットの相対位置関係を知覚でき、障害物をよけながら移動することが可能な方法を提案する。

まず 3D 空間内の任意の曲線が、区分的に振率 (torsion) 0 であるような曲線群で近似できることに着目する。これは 2 次元部分空間上の曲線を連結することと等価で、2 次元の曲線を 1 次元の曲線（すなわち直線）で近似することの自然な次元の拡張となっている (図 5)。

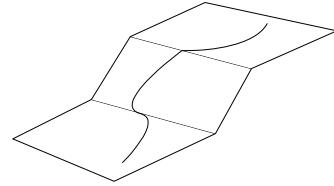


図5 2D 曲線による 3D 曲線の近似

マウスによる操作を、平面を指定するフェーズとその平面上を移動するフェーズに分け、それらを交互に繰り返すことによって、3 次元空間内の任意の軌道を近似的に入力することができる。

まず平面を指定するフェーズでは、マウスの x 軸移動を平面の空間軸の z 軸回りの負の回転 (roll) に、マウスの y 軸移動を x 軸回りの負の回転 (pitch) に対応させる。空間軸の回転の中心は、マウスでポイントした点に対応する 3 次元座標、すなわち前節における $\vec{p} = \overrightarrow{OP} = (\tilde{p}_x, \tilde{p}_y, \tilde{p}_z)^t$ とする。再び、ドラッグによるベクトルを $\vec{u} = \overrightarrow{PQ} = (u_x, u_y, 0)^t$ と表すと、座標系の 3 軸の単位ベクトル $\vec{e}_x, \vec{e}_y, \vec{e}_z$ は、適当に選んだ定数 μ を用いて式(4)で与えられる。

$$\theta = \mu \sqrt{u_x^2 + u_y^2}, \quad c = \cos \theta, \quad s = \sin \theta,$$

$$\vec{e}_x = \begin{pmatrix} \frac{(c-1)u_x u_y}{u_x^2 + u_y^2} & \frac{u_x^2 + c u_y^2}{u_x^2 + u_y^2} & \frac{-s u_y}{\sqrt{u_x^2 + u_y^2}} \end{pmatrix},$$

$$\mathbf{e}_y = \begin{pmatrix} \frac{cu_x^2 + u_y^2}{u_x^2 + u_y^2} \frac{(c-1)u_x u_y}{u_x^2 + u_y^2} \frac{-su_x}{\sqrt{u_x^2 + u_y^2}} \end{pmatrix},$$

$$\mathbf{e}_z = \mathbf{e}_x \times \mathbf{e}_y = \begin{pmatrix} \frac{su_x}{\sqrt{u_x^2 + u_y^2}} \frac{su_y}{\sqrt{u_x^2 + u_y^2}} c \end{pmatrix} \quad (4)$$

ドラッグによって回転する平面は、移動させる物体、ターゲット、及びその他の仮想 3D 物体との切断線 (cutting-plane line) として表示される。この切断線を用いて、移動させる物体とターゲットが同一平面上にあることや、その平面上にある障害物との位置関係などを確認することができる。また遠近感を知覚するための手掛かりにもなる。

平面を指定した後、マウスの左ボタンで Release & Press の操作をすると、指定した平面上を移動するフェーズに移行する。マウスの x , y 方向の移動は、仮想 3D 空間中の $\mathbf{e}_x$, $\mathbf{e}_y$ 方向の移動に対応させる。

平面を移動した後、Release & Press の操作をすると、再び平面指定のモードに移行する。このときの平面の回転の中心は移動先の座標とする。以下両フェーズを繰り返し、最後に左ボタンを離すと通常のカーソル移動に戻る。

3.2 3DOF 移動に関するデモ

図 6 に立体組立パズルデモのスクリーンショットを示す。このパズルを操作するためには、個々のピースの選択とそれぞれの 3DOF 回転だけでなく、3DOF 移動も必要である。まずピースをポイントすると回転のフェーズになる。回転後 Release & Press の操作をするか、あるいは直接ダブルクリックを行うことにより、平面指定のフェーズになる。さらに Release & Press の操作をすると平面移動のフェーズになる。以下 Release & Press の操作を行うごとに平面指定と平面移動のフェーズを交互に繰り返す。なおピース以外の場所のドラッグは、シーンの回転に対応させている。

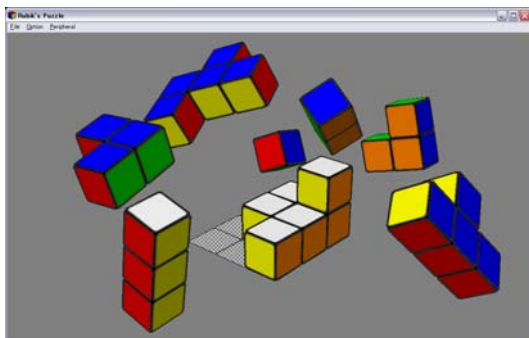


図6 3DOF 移動に関するデモ：立体組立パズル

4. まとめ

本稿では、2D ポインティングによる 3D/6DOF ダイレクトマニピュレーションの一方式を提案した。仮想 3D 物体の選択・回転・移動がシームレスに行えるため、例えば物体の数が n で、背景のドラッグをシーンの 2DOF 回転に対応させた場合、 $6n+2$ DOF の操作が可能となっている。応用範囲は仮想 3D 物体操作全般にわたると思われる。

参考文献

- 1) Hanson, A. J.: The Rolling Ball, *Graphics Gems III*, pp. 51-60, Academic Press (1992).
- 2) Chen, M., Mountford, S. J., and Sellen, A.: A study in Interactive 3-D Rotation Using 2-D Control Devices, *Computer Graphics*, **22** (4), pp. 121-129 (1988).
- 3) Shoemake, K.: ARCBALL: A User Interface for Specifying Three-Dimensional Orientation Using a Mouse, *Proc. of Graphics Interface '92*, pp. 151-156 (1992).
- 4) Henriksen, K., Sporning, J., and Hornbaek, K.: Virtual Trackballs Revisited, *IEEE Transactions on Visualization and Computer Graphics*, **10** (2), pp. 206-216 (2004).
- 5) 三次元マウスの装置, 特開 2008-282400(2008).
- 6) 複数の指で操作するセンサーを備えた三次元空間を操作するためのポインティングデバイス, 特開 2007-293853 (2007).
- 7) 三次元マウスの Z 軸座標入力機構, 特開平 09-288541 (1997).
- 8) 3 次元ポインティングデバイス装置と移動検出法, 特開平 09-288541 (1997).
- 9) 三次元マウス, 特開平 08-179883 (1996).
- 10) 3 次元座標値の入力が可能なマウス型入力装置, 特開平 05-282098 (1993).
- 11) 特開平 08-249490, 特開平 05-204589, etc.
- 12) Khan, A., Mordatch, I., Fitsmaurice, G., Matejka, J., and Kurtenbach G.: ViewCube: A 3D Orientation Indicator and Controller, *I3D 2008 Conference Proceedings: ACM Symposium on Interactive 3D Graphics*, pp. 17-25 (2008).
- 13) Cyber Sport: Orbita Mouse®, <http://www.orbitamouse.com/>
- 14) 3D connexion: SpaceNavigator®, etc. <http://www.3dconnexion.jp/>
- 15) Poupyrev, I., Weghorst, S., and Fels, S.: Non-Isomorphic 3D Rotational Techniques, *CHI Letters*, **2** (1), pp. 540-547 (2000).
- 16) Sundin, M., Fjeld, M.: Elastic 6 DOF Input, *International Journal of Human-Computer Interaction*, **25** (7), pp. 647-691 (2009).
- 17) Balakrishnan, R., and Kurtenbach, G.: Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces, *Proc. of CHI99*, pp. 56-63 (1999).